4Gbps CMOS Backplane Receiver

with Adaptive Blind DFE

By

Slobodan Milijevic

A thesis submitted to the Faculty of Graduate Studies and Research

in partial fulfillment of the requirements for the degree of

Master of Applied Science

Department of Electronics

Carleton University

Ottawa, Ontario, Canada

September 2006

Your file Votre référence
ISBN: 978-0-494-23342-9
Our file Notre référence
ISBN: 978-0-494-23342-9

Abstract

This thesis presents a serial backplane receiver with adaptive blind decision feedback equalization (DFE), designed in 0.35 μm TSMC process, which can operate up to 4Gbps over 1.2 m long FR-4 (typical isolation material used for making Printed Circuit Boards — PCB) based PCB channel, which includes discontinuities due to the packaging and backplane connectors.

To maximize data rate that can be supported by the receiver, the DFE is achieved in look-ahead manner where each input symbol is sampled with two biased comparators — one biased high as if the previous symbol was low and the other biased low as if the previous symbol was high.

The biased comparator is implemented by adding two bias transistors to the sense amplifier based flip-flop (sense amplifier followed by an SR-latch), also know as Strong-Arm flip-flop. The inherent input hysteresis of the Strong-Arm flip-flop was reduced by an order of magnitude with a simple modification of the standard SR-latch.

DFE coefficient calculation is not performed on every consecutive received sample, which significantly reduces the design complexity and power consumption. Adaptation algorithm is not only used to adjust DFE coefficient, but also to compensate for attenuation of the transmission line.

Acknowledgements

I would like to thank my research supervisor Prof. Dr. Tad Kwasniewski who proved to be an invaluable source of advice, guidance and support during the course of this thesis.

I am grateful to my wife and our three children for their encouragement, patience and love. Their presence gave me the strength to finish my part time graduate studies while having a full time job with Zarlink Semiconductor.

I am also thankful to Zarlink Semiconductor for financially supporting my graduate studies.

The information used in this thesis comes in part from the research program of Dr. Tad Kwasniewski and his associates in the VLSI in Communications group. The research results appearing in this thesis represent an integral part of the ongoing research program. All research results in this thesis including tables, graphs and figures but excluding the narrative portions of the thesis are effectively incorporated into the research program and can be used by Dr. Kwasniewski and his associates for educational and research purposes, including publication in open literature with appropriate credits. The matters of intellectual property may be pursued co-operatively with Carleton University and Dr. Kwasniewski, where and when appropriate.

Table of Contents

List of Tables

List of Figures

List of Abbreviations

AC	Alternating Current
ATCA	Advanced Telecom Computing Architecture
BERT	Bit Error Rate Test
BGA	Ball Grid Array
CMOS	Complementary Metal Oxide Semiconductor
D/A	Digital to Analog Converter
DC	Direct Current
DFE	Decision Feedback Equalization
DLL	Delayed Locked Loop
EMI	Electro Magnetic Interference
FIFO	First in, First out
FIR	Finite Impulse Response
FO-4	Fanout of four
IC	Integrated Circuit
ISI	Inter Symbol Interference
MSE	Mean Square Error
PAM	Pulse Amplitude Modulation
PCB	Printed Circuit Board
SR Latch	Set Reset Latch
TDM	Time Division Multiplexing
XO	Crystal Oscillator

CHAPTER 1

Introduction

1.1 Motivation

In the past fifteen years, the on-chip clock frequency has increased by more than 20 times thanks to improvements in semiconductor manufacturing. At the same time, PCB technology did not improve very much. The maximum number of signal and power layers did increase, buried and blind vias were introduced which helped improve routing density and somewhat signal integrity, but the signals are still transmitted over the printed circuit board (PCB) traces with the same bandwidth — copper traces separated with a dielectric material (usually FR-4 due to its low cost).

Therefore, the major bottleneck in digital systems is becoming the inter-chip and inter-board communication over the PCB. At the multi-gigahertz data rates, the inter symbol interference (ISI) due to the skin effects, and the dielectric losses in the PCB, become the major problem for reliable digital transmission. The higher frequency components of a transmitted signal are attenuated much more than the lower ones causing dispersion of symbols in time. ISI is usually mitigated, by placing a pre-emphasis [3] and [4] or adaptive pre-emphasis [2] filter at the transmitter. The pre-emphasis and adaptive pre-emphasis are almost exclusively used for multi-gigahertz data rates, rather than the receive equalization because a pre-emphasis filter can be implemented relatively simple in the analog

domain by summing the currents from the filter taps in an output pad. However, the pre-emphasis has two major shortcomings:

- Because of the limited swing of the output drivers, the pre-emphasis is performed not by amplifying the high frequency components of the transmitted signal but rather by attenuating the lower frequency components. This in turn reduces the power of the transmitted signal as well as the maximum eye opening at the receiver.
- Adaptive pre-emphasis needs a reliable return channel (from the receiver to the transmitter) because the coefficients for pre-emphasis filter are calculated at the receiver side.

These two problems can be solved with an equalizer at the receiver side. This thesis presents a method for mitigating ISI with an adaptive blind Decision Feedback Equalization (DFE), which adaptively adjusts not only DFE coefficient but also the threshold level of the blind adaptation engine to compensate for attenuation of the of the transmission line.

This thesis also shows that the adaptive coefficients calculation does not have to be done on every consecutive received sample, but rather can be done on every Nth sample where N is the ratio of demultiplexing in the receiver front-end. This in turn reduces the size and power consumption of the adaptive coefficient calculation engine by factor N. This is a significant reduction considering that the coefficient calculation engine in [2] consume 60% of the total power of the transceiver.

1.2 Objective

The objectives of this thesis is to design a high-speed backplane receiver with adaptive blind DFE in 0.35um TSMC process which can operate at multiple giga bit per second rate.

1.3 Contributions

Previous solutions of multiple gigabit per second receivers [2], [3] used pre-emphasis and adaptive pre-emphasis in order to mitigate ISI. In this thesis we have shown how the problem with ISI can be solved with adaptive blind DFE where DFE is done in look-ahead way by biasing high-speed comparators. The primary contributions of this thesis are:

- Look ahead DFE implementation with biased comparators
- The high-speed comparator, implemented by modifying Strong-Arm [15] high-speed flip-flop — biasing transistors were added to the sense amplifier and the comparator input hysteresis was solved by modifying the standard SR latch.
- Blind adaptation algorithm which not only adjusts DFE coefficient, but also the threshold level of the blind adaptation algorithm in order to compensate for attenuation of the transmission line.
- Finding that the adaptive coefficient calculation does not have to be done on every consecutive received symbol, which reduces the power consumption and the size of the adaptation engine by N where N is the level of demultiplexing in the receiver. It also reduces the maximum speed requirements of the logic used in the adaptation engine by N.

Some of contributions of this thesis have been published in [1].

1.4 Thesis organization

The background material on transmission lines, backplane transceivers and equalization and pre-emphasis used in backplane transceivers is introduced in Chapter 2. The receiver architecture, implementation and simulation results of major blocks are presented in Chapter 3. Simulation results of the complete receiver are shown in Chapter 4. Finally, the conclusion and some ideas for future improvements of the receiver are presented in Chapter 5.

CHAPTER 2

Background Theory

2.1 Introduction

This chapter provides background material necessary for understanding high speed serial digital transmission over printed circuit board (PCB) based backplane.

First, we present common methods of interboard communication over the backplane: point to multi-point (shared bus architecture) and point to point (star and mesh architecture).

This is followed by the section which covers basics of the transmission line theory. In this section we also derive the voltage and current equations, define the characteristic impedance and the reflection coefficient, and examine common methods for terminating transmission lines.

Next, we describe differential and single ended communication with their corresponding advantages and disadvantages as well as the basic drivers (transmitters) used for digital transmission over PCBs as well as the transmitter and receivers architectures used in multi-giga bit transmission over the backplane. Then, we provide more emphasis on the transmitter design because the receiver design will be covered in more detail in the next chapter.

Further, we explain how PCB transmission line bandwidth affects the transmission of multi-giga bit per second digital signals. Specifically, we show how the transmission

line bandwidth limitation, caused by the skin effect in copper traces and the loss in dielectric material give rise to ISI, which adversely affects receiver's symbol detection ability due to the closure of the eye-pattern.

Finally, we explain common methods for mitigating ISI in multi-giga bit per second transmission over the backplane.

2.2 Interboard communication in telecom/datacom systems

Typical telecom/datacom system consist of multiple cards that communicate over a common backplane as shown in Figure 2.1. The communication over the backplane can be based on point to multipoint and point to point communication.

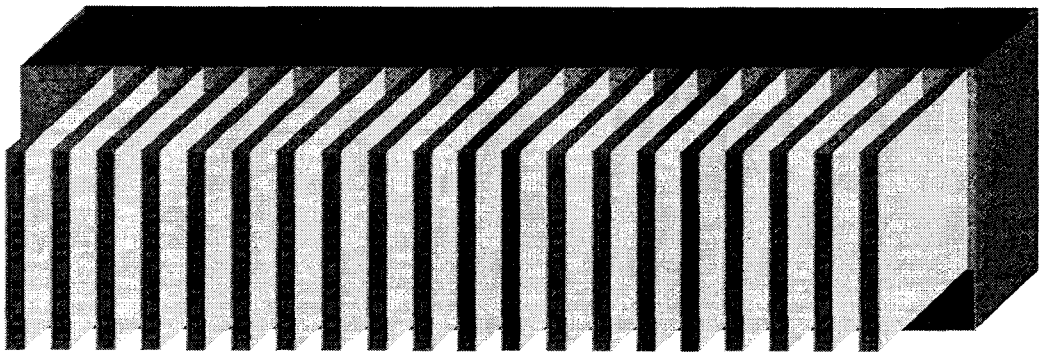

Figure 2.1: Typical telecom/datacom system

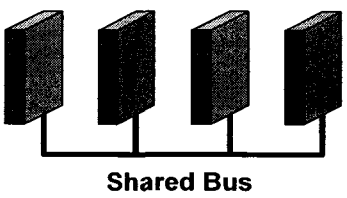

Figure 2.2: Shared Bus backplane architecture

The point to multipoint communication is used for shared bus (Figure 2.2) where each trace on the backplane is shared among all cards.

The shared bus is used in systems where data rate on the backplane is below 100Mbps per one differential link (two traces on the backplane per link). Typically, the shared bus is used in Time Division Multiplex (TDM) systems where each serial link carries multiple voice channels. Common data rates in TDM are 2.048Mbps, 8.192Mbps, 16.384Mbps, 32.768Mbps and 65.536Mbps.

Major drawbacks of the Shared Bus architecture are relatively low data rates and the single point of failure — if only one card fails where failure shortens all backplane traces to the power of ground rail, then the whole system (chassis) fails.

Because of the relatively low maximum data rate, and because of the single point of failure, the shared bus architecture is not used often in the systems requiring the high data throughput and the high reliability. These high performance systems use more often point to point communication over the backplane with either dual star or mesh topology as shown in Figure 2.3.

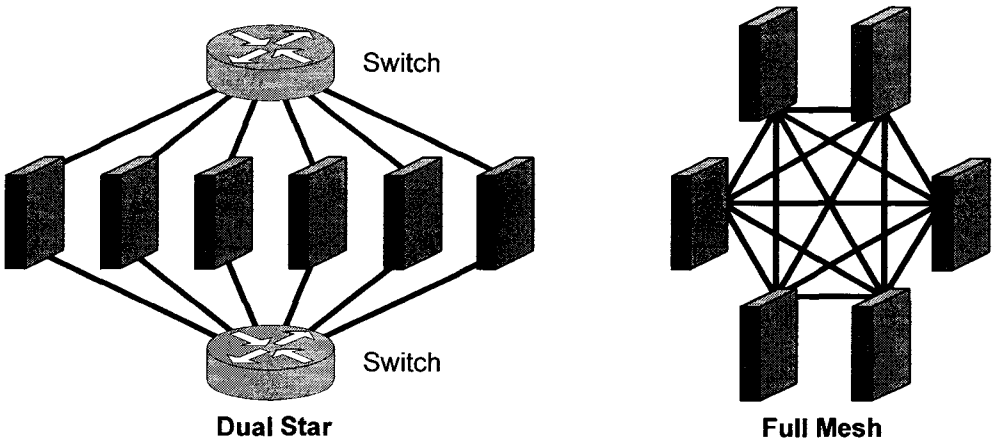

Figure 2.3: Dual Star and Full Mesh backplane architecture

In the Star architecture, all cards in the system (chassis) are connected to a switch card with point-to-point links. Hence, the communication between any two cards is the system is done via the switch card. To prevent a single point of failure — if the switch card fails, whole system fails — additional switch card is added for redundancy purposes (Dual Star architecture). Besides, redundancy protection, this additional switching card can be used to double the system bandwidth during the normal operation (when both switching card are working properly).

In the Full Mesh architecture, any card in the system (chassis) is connected to all the other cards in the system via point-to-point links. Full Mesh architecture provides full redundancy protection because system can operate (though with the lower performance) as long as there are two working cards in the system. However, drawback of the Full Mesh architecture is that the complexity (cost) of each card increases if the system needs to accommodate more cards because if the maximum number of cards is N than, each card has to provide (N-1) point-to-point interfaces to connect with remaining cards. The other problem with this architecture is that the number of links on the backplane increases rapidly with the maximum number of cards system is supposed to accommodate. If the maximum number of cards is N than backplane need to have (N-1)*N/2 point-to-point serial links.

An example of a system that supports both Dual Star and Full Mesh architectures is PICMG 3.X (ATCA) [27], which is collection of standards for systems based on packed switched fabrics where communication over the backplane is done with point-to-point serial links that can run at multiple giga bit per second rates.

2.3 Transmission lines

Infinitesimal section of a transmission lines are modeled in the circuit theory as a four terminal networks with two series parameters per unit length (resistance R and inductance L) and two parallel (shunt) parameters per unit length (conductance G and capacitance C) as shown in Figure 2.4.

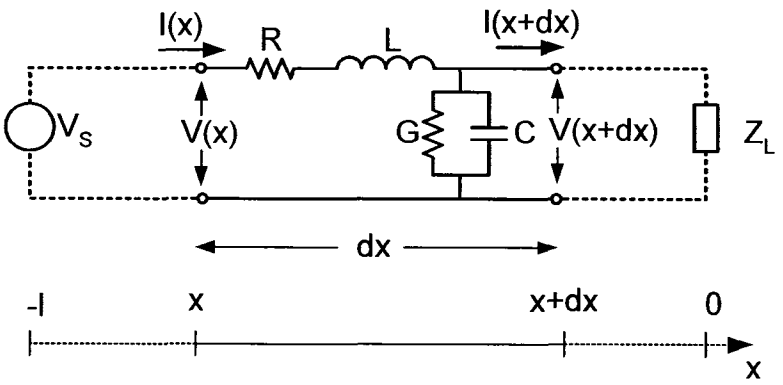

Figure 2.4: Model of infinitesimal section of a transmission line

In order to simplify equations, we will derive the voltage and current equations for sinusoidal excitation. However, these equation can be extended to cover any arbitrary excitation using Fourier series.

The voltage and current can be derived from [26]:

$$\frac{dV}{dx} = -(R + j\omega L)I \tag{2.1}$$

and

$$\frac{dI}{dx} = -(G + j\omega C)V \tag{2.2}$$

By taking the first derivative of (2.1) with respect to distance x, and by substituting (2.2) into the resulting equation, we obtain

$$\frac{d^2V}{dx^2} = ZYV \tag{2.3}$$

where $Z = R + j\omega L$ and $Y = G + j\omega C$. Similarly, we obtain

$$\frac{d^2I}{dx^2} = ZYI \tag{2.4}$$

Equations (2.3) and (2.4) are well known differential equations for wave propagation which specify voltage and current anywhere along the transmission line. Solutions to these two differential equation are

$$V = V_{in}e^{-\rho x} + V_{ref}e^{\rho x} \tag{2.5}$$

$$I = I_{in}e^{-\rho x} + I_{ref}e^{\rho x} \tag{2.6}$$

where $V_{in}e^{-\rho x}$, $V_{ref}e^{\rho x}$ and $I_{in}e^{-\rho x}$, $I_{ref}e^{\rho x}$ are incident and reflected voltage and current waves respectively and ρ is short form of $\rho = (ZY)^{1/2}$, usually called the propagation constant. V_{in}, V_{ref}, I_{in} and I_{ref} are constants, whose values can be determined from the known voltages and currents at two different points on the transmission line. Relationship between the voltage and current constants can be shown by taking the first derivative of (2.5) with respect to x

$$\frac{dV}{dx} = -V_{in}\rho e^{-\rho x} + V_{ref}\rho e^{\rho x} \tag{2.7}$$

and by substituting (2.1) into (2.7) we obtain

$$-ZI = -V_{in}\rho e^{-\rho x} + V_{ref}\rho e^{\rho x} \tag{2.8}$$

or

$$I = \frac{V_{in}}{Z}\rho e^{-\rho x} - \frac{V_{ref}}{Z}\rho e^{\rho x} \tag{2.9}$$

By comparing (2.6) and (2.9) we have

$$I_{in} = \frac{V_{in}}{Z}\rho = \frac{V_{in}}{(Z/Y)^{1/2}} \tag{2.10}$$

$$I_{ref} = -\frac{V_{ref}}{Z}\rho = -\frac{V_{ref}}{(Z/Y)^{1/2}} \tag{2.11}$$

where $(Z/Y)^{1/2}$ is called the characteristic impedance of the transmission line [26].

$$Z_0 = (Z/Y)^{1/2} = \left(\frac{R+j\omega L}{G+j\omega C}\right)^{1/2} \tag{2.12}$$

The characteristic impedance is the ratio between the voltage and current at any point along a transmission line with infinite length. The infinite length means that the voltage wave goes only forward and that there is no reflected voltage wave. We might think that the characteristic impedance has no meaning in practice since length of actual transmission lines is always finite. However, magnitude of the reflected wave will depend on the impedance of the load at the end of a transmission line. The incident wave will be absorbed completely for load impedance that is equal to the transmission line characteristic impedance. For such load termination, the transmission line will behave as an infinite transmission line.

Let us now define the reflection coefficient Γ as a ratio between the reflected and incident wave

$$\Gamma(x) = \frac{V_{ref}e^{\rho x}}{V_{in}e^{-\rho x}} = \frac{V_{ref}}{V_{in}}e^{2\rho x} \tag{2.13}$$

The reflection coefficient at the load ($x = 0$) is

$$\Gamma_L = \frac{V_{ref}}{V_{in}} \tag{2.14}$$

and now impedance along the line can be expressed as

$$Z(x) = \frac{V(x)}{I(x)} = Z_0\frac{e^{-\rho x} + \Gamma_L e^{\rho x}}{e^{-\rho x} - \Gamma_L e^{\rho x}} \tag{2.15}$$

At $x = 0, Z(0) = Z_L$, and (2.15) becomes

$$Z_L = Z_0 \frac{1+\Gamma_L}{1-\Gamma_L} \tag{2.16}$$

Finally, from (2.16) we have reflection coefficient at the load

$$\Gamma_L = \frac{Z_L - Z_0}{Z_L + Z_0} \tag{2.17}$$

Now, if the load impedance is equal to the characteristic impedance of the transmission line than reflection wave is equal to zero and the energy of the incident wave is absorbed into the load. Contrary, if the transmission line is open ($Z_L = \infty$), whole incident wave gets reflected ($V_{in} = V_{ref}|_{x=0}$).

2.3.1 Transmission line terminations

Preserving signal integrity is very important for reliable digital signal transmission over PCB traces. The signal integrity problems such as ringing, overshoot, undershoot, and non-monotonicity of signal's edges are typically by-product of reflected signal waves caused by discontinuities (such as PCB vias, stubs, connectors ...) and improper termination of the transmission line. As we have shown in the previous section, the reflected wave at the end of the transmission line can be canceled if the line is terminated with the load equal to line's characteristic impedance. Although very beneficial for signal integrity, terminating the transmission line with the load equal to the characteristic impedance has negative effect on the power consumption because of the power dissipated in the load. The power consumption in the load can be unacceptably high, because typical characteristic impedance of a PCB trace is 50 Ohm or lower. We might ask, why not to use PCB traces with higher impedance; 1 KOhm for instance. Unfortunately, this is not possible because much higher characteristic impedance would require much narrower traces, which cannot be manufactured with standard PCB process.

As a matter of fact, majority of PCB connections in practice today do not use load termination equal to the characteristic impedance. The reliable transmission is achieved

here by transmitting digital signals with relatively low data rates (this implies slow rise and fall times), by making traces as short as possible, or by using series, AC or Thevenin termination [23] as shown in Figure 2.5.

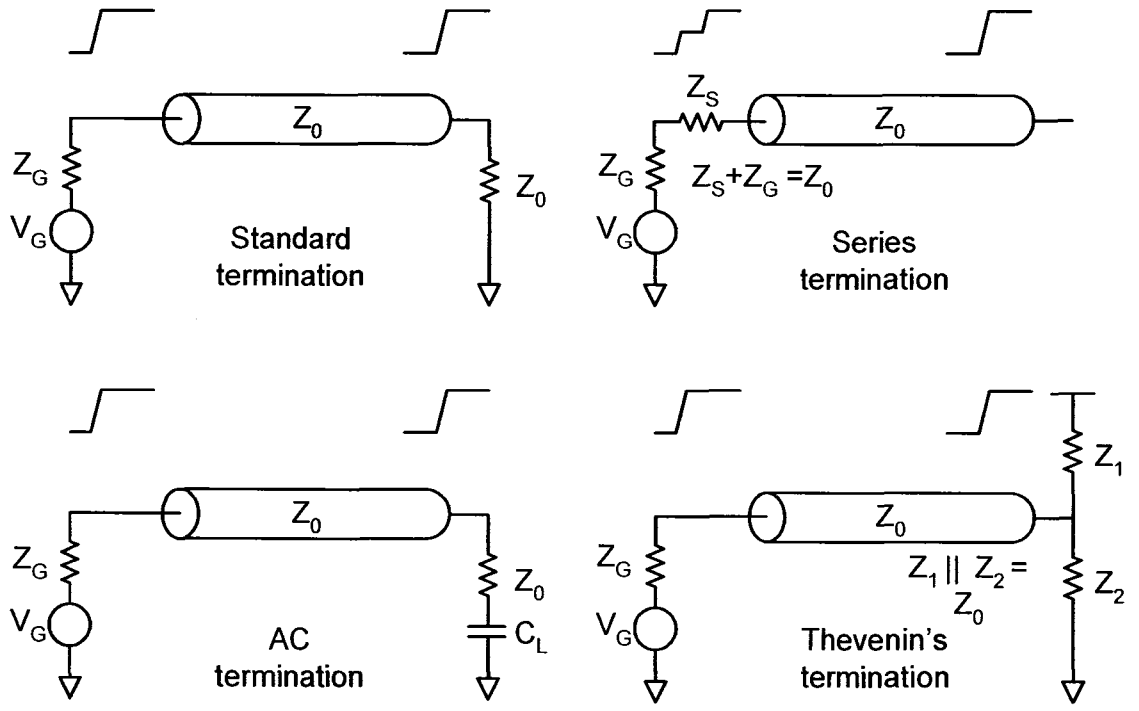

Figure 2.5: Common PCB transmission line terminations

Series termination is not really a termination per se. By making series termination plus driver output impedance equal to the line characteristic impedance, we can reduce incident wave by half. When this wave arrives at the end of the unterminated line, it gets reflected, and resulting signal goes all the way to the power rail. When the reflected wave arrives at the source it gets absorbed in the matching load (driver output impedance plus series termination).

The AC termination adds a capacitor in series with load to cancel out DC power consumption. This method however, increases AC power consumption, which is propor-

tional to the capacitive load and also reduces the rise and fall time when used with weak drivers.

The Thevenin or split termination adds two resistors whose equivalent load is equal to the characteristic impedance. While this method reduces the current drawn from the driver (by 2 for Z1 = Z2), it does not reduce the overall power consumption because power is burned here not only when the driver drives high, but also when it drives low.

2.4 Transmitting and receiving circuits

2.4.1 Differential vs. single ended signaling

Communication between a transmitter and receiver can be done with the differential or with single ended signalling [23], [24]. As can be seen in Figure 2.6, the single ended signaling — also called unbalanced signaling — requires only one transmission line (trace).

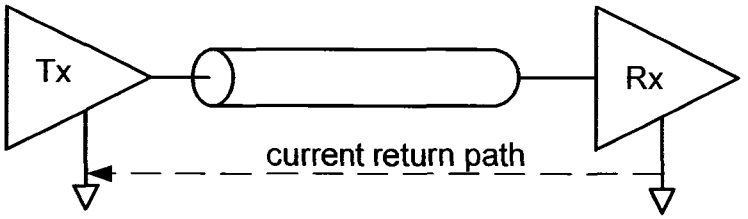

Figure 2.6: Single ended signaling

The return current is assumed to go via a zero impedance ground plane shared between the transmitter and receiver. The name unbalanced comes from the fact that the forward and return currents take paths with different (unbalanced) characteristics.

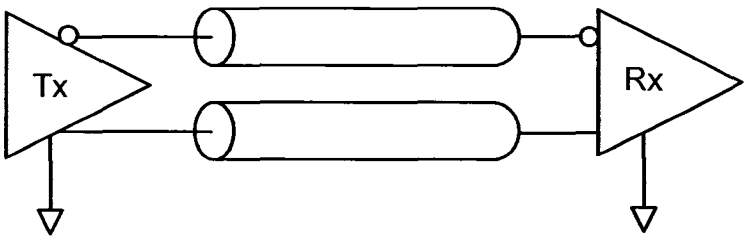

Figure 2.7: Differential signaling

Differential signaling — also called balanced signaling — requires two identical traces between the transmitter and receiver as shown in Figure 2.7. The differential signal is generated by applying voltages (currents) that are equal in amplitude, but opposite in polarity onto the traces. A common ground plane for return current is not needed because the currents in two traces always go in the opposite direction. The receiving device detects either zero or one based on the polarity of the voltage difference between the traces. Hence the name differential signaling. The name balanced signaling comes from the requirement that two traces need to be identical (balanced).

Although differential signaling requires twice the number of traces compared to the single ended signaling, it has a number of advantages which are outlined below:

- Differential signaling is not affected by the common mode noise (power supply noise, crosstalk...) that equally affect (same amplitude and polarity) both conductors. The common mode noise is canceled at the receiver because the signal is detected by taking the difference between voltages at two traces.
- It reduces EMI emission because EMI generated by one conductor in differential pair is in part canceled by the emission generated in the other conductor. The degree of EMI reduction is based on the distance and balance between the conductors.

- Differential signaling can deliver two times larger signal amplitude than the single ended signaling.

Because of these advantages, differential signaling is almost exclusively used for transmission over the backplane except for very low data rates (typically below 8Mbps per link).

2.4.2 Voltage vs. current drivers

Depending on their output impedance, line drivers can be divided to voltage and current drivers. A simple voltage and current drivers are shown in Figure 2.8 and Figure 2.9 respectively.

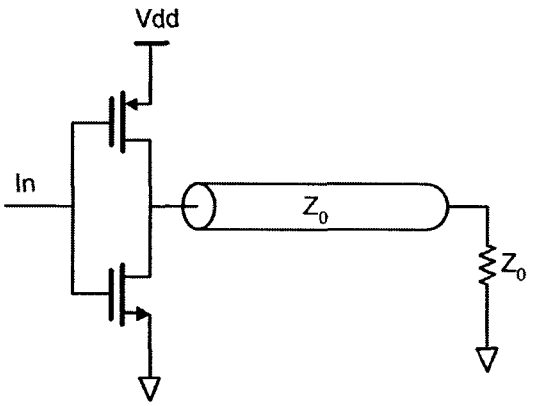

Figure 2.8: Voltage driver

The simple voltage driver is just a CMOS inverter while the current mode driver is usually implemented as a differential current steering circuit. The current driver shown in Figure 2.9 drives a differential line but it can be also used to drive a single-ended line by terminating the complementary output ($\overline{out}$) with the termination voltage (Vt). The current mode driver behaves as a high-impedance current source where the output signal swing is adjusted simply by changing the bias voltage (Vbias).

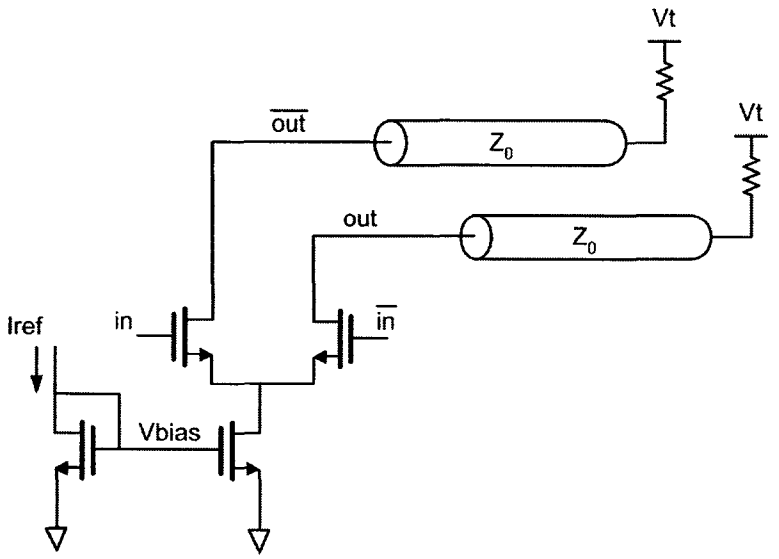

Figure 2.9: Current driver

The voltage mode driver behaves as a low-impedance voltage source where the output signal swing cannot be easily adjusted — this can be done only by changing the supply voltage of the output driver. Hence, the output swing of a typical voltage mode driver is from rail to rail. Because of this, transmission lines driven with the voltage mode drivers are generally not terminated with matching termination to reduce the power consumption and the size of the driver.

For instance, if a 50 ohm transmission line is terminated with the matching 50 ohm resistor to the ground, then DC power consumption (assuming 3.3V power supply) of the driver, when it drives high voltage level, would be $(3.3V)^2/50ohm = 0.218W$!

Although, the voltage mode drivers do not consume DC power when they drive unterminated transmission lines, they do consume AC power, which can be expressed as

$$P_{AC} = fC_L(V_{dd})^2 \tag{2.18}$$

where f is switching frequency, C_L is capacitive load of the driver and V_{dd} is driver's power supply voltage. The current drawn from the power supply is equal to zero, except

during the transitions from low to high (charging the load capacitance). The charge and discharge of the load capacitance generates noise spikes on the die which contains the driver. Because of the noise injection, power consumption proportional to switching frequency, and difficulty, the voltage mode drivers are usually used for lower data rates on short PCB traces.

Current drivers draw only DC current because source coupled pair steer the current form the current source trough the one or the other leg. This significantly reduces the AC component of the power supply noise.

Because of the low noise generation and high noise immunity, current mode drivers can transmit reliably data with lower voltage swing (typically 100 mV to 800 mV) which in turn translates into lower power consumption and higher maximum speed of operation. Transmitters used in multi-gigabit per second use current differential drivers, because of their high speed and noise immunity.

2.4.3 Transmitter for multi-gigabit per second link

The maximum on-chip clock frequency is usually expressed in the number of propagation delays of an inverter that drives four times higher capacitive load than its input capacitance. The delay of such inverter is labeled as FO-4 (fanout of 4) and the maximum on-chip clock speed is reciprocal of M x FO-4 where M is usually taken to be between 4 and 8. For 0.35 μm technology, the maximum 6 x FO-4 delay over all voltage and temperature corners is about 700 ps so that the maximum clock frequency is 1.4 GHz. Because the maximum on-chip clock frequency is lower than the serial data rate, some sort of multiplexing is required at the transmitter output. The multiplexing can be done at the output pad [2], [3], by turning on/off output drivers in the round-robin fashion as shown in Figure 2.10 or by multiplexing data just before the output current driver [12] as shown in Figure 2.11.

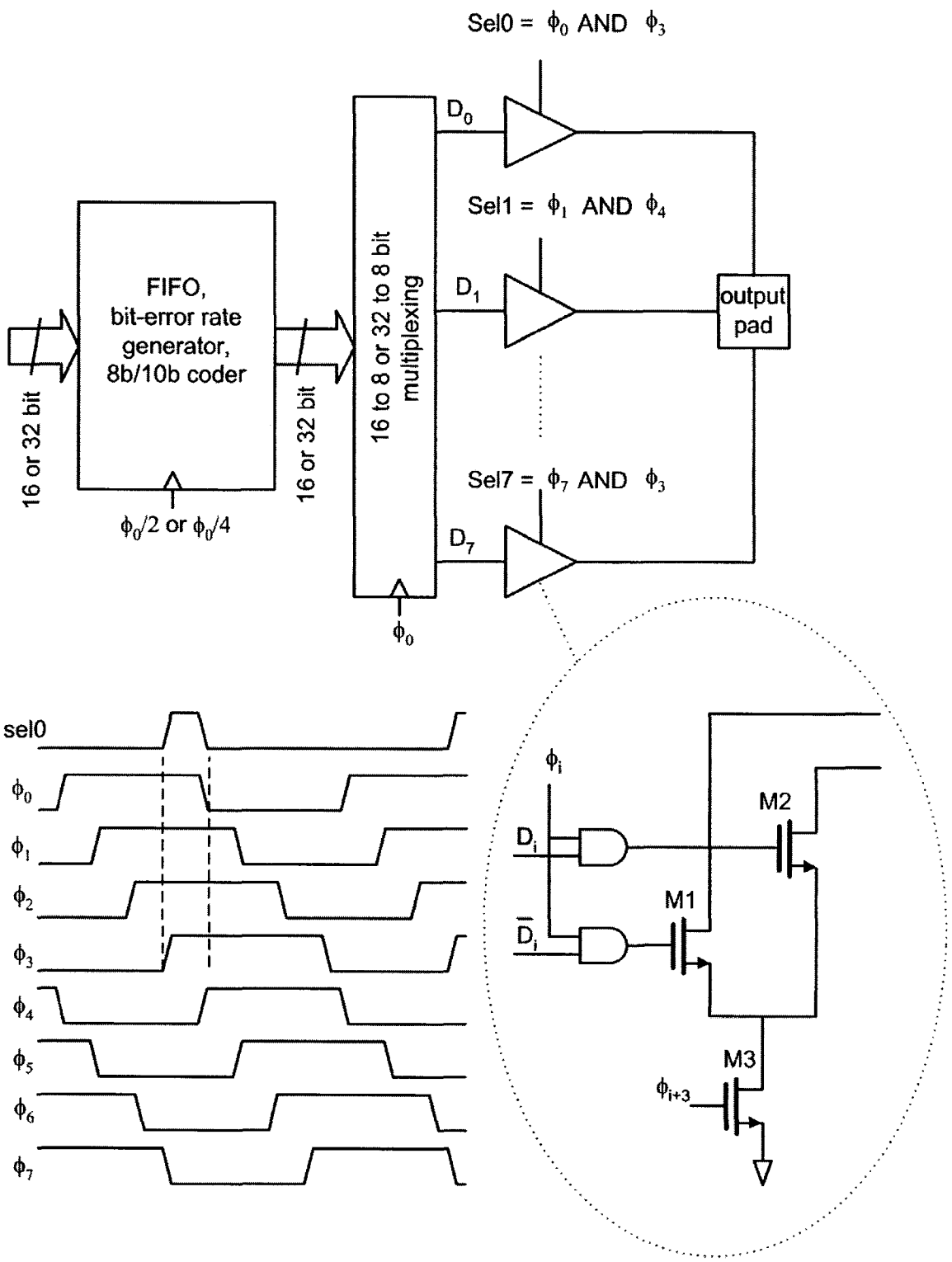

Figure 2.10: Typical transmitter with external multiplexing buffer

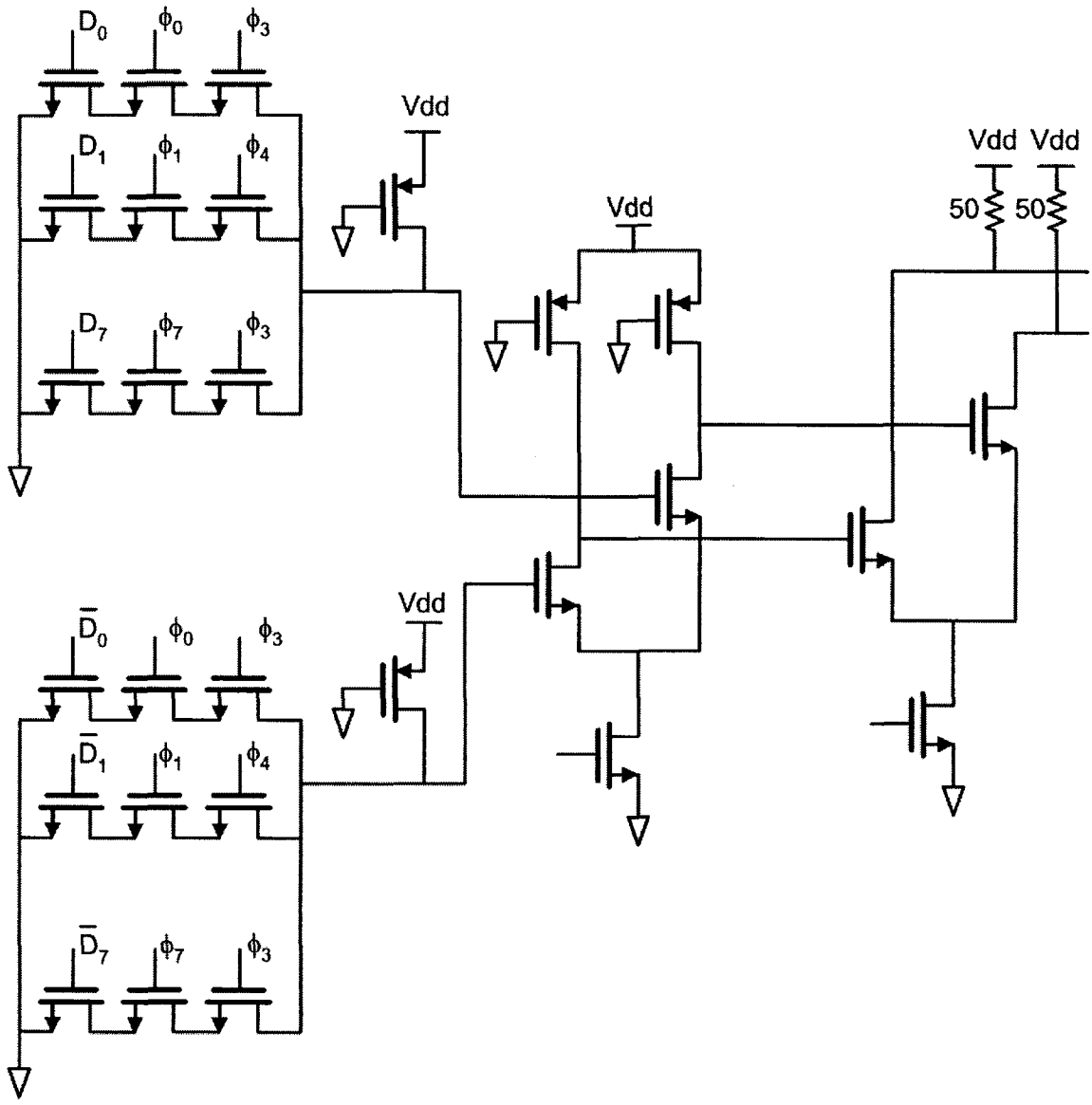

Figure 2.11: Output buffer with internal multiplexing

The first method is very simple to implement and has being used in CMOS based transmitters with rates of up to 10 Gbps, even in 0.4 μm CMOS process [3]. This translates to delays of less than FO-4 for 0.4 μm CMOS process. However, this solution takes large space on the die because the output buffer has to be reproduced N times where N is the ratio of multiplexing. The second solution proposed in [12], requires only one output

buffer because the multiplexing is done internally. This method, takes considerably less space on the die and consumes less power. However, its major drawback is the maximum speed limitation. The transmitter described in [12], implemented in 0.25 μm CMOS technology, operates at maximum 4 Gbps. This is about 2 x FO-4 for 0.25 μm CMOS.

The remaining digital circuit is the same regardless of the multiplexing scheme as shown in Figure 2.10. Digital data are fed to the internal FIFO via relatively low speed parallel links (16 or 32 bit wide). From the FIFO, data are fed to some sort of coder (typically 8b/10b), which is used to introduce enough transitions into the transmitted data as well as to cancel DC component. After the coder, data are multiplexed down (32 or 16 bit to 8 bit) and fed to the transmit multiplexer (8 to 1 bit). For the bit-error rate test purposes, the transmitter can bypass the FIFO data and inject locally generated pseudo-random data.

2.4.4 Receivers for multi-gigabit per second links

Received multi-gigabit per second serial signal is first demultiplexed inside the receiver to reduce the required on-chip clock frequency [2], [12], [13]. The ratio of demultiplexing is usually from 1 to 8 or higher. The incoming serial data stream is sampled with slicers, each clocked by a different phase of the on-chip clock as shown in Figure 2.12. The output of N slicers clocked with eight different phases (ϕ_0, ϕ_1, ..., ϕ_7) of the recovered clock are further re-synchronized to only one phase (ϕ_0), which simplifies design and implementation of remaining digital circuitry. To further reduce requirements of the internal circuitry such as the speed of the memory (FIFO) and the speed of the output drivers used in parallel port, additional demultiplexing might be performed — from 8 to 16 or from 8 to 32 bit.

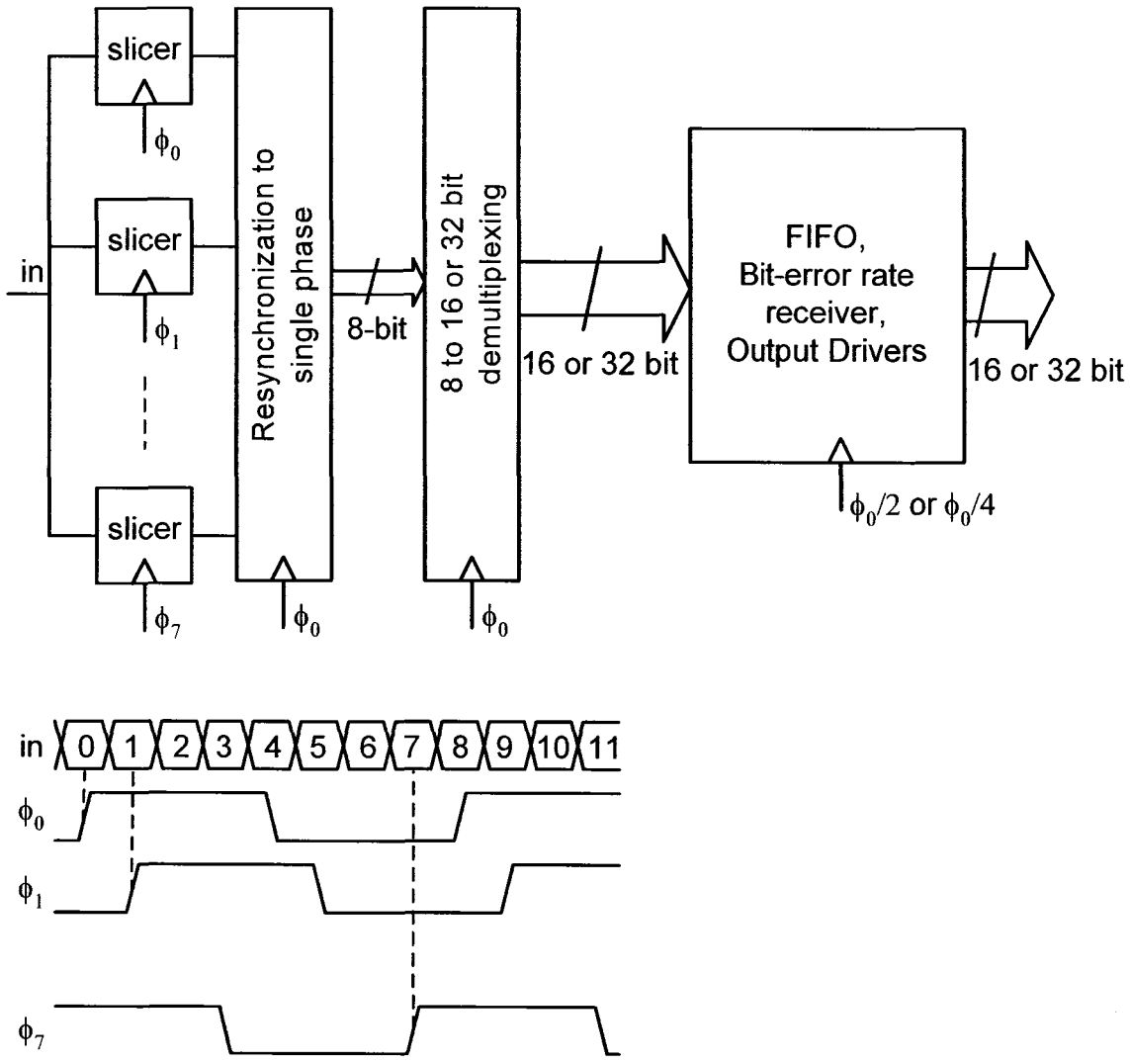

Figure 2.12: Block diagram of a generic multi-giga bit per second receiver

To simplify the block diagram, Figure 2.12 shows that each received symbol is sampled only once. However, the received symbols are sampled two times or more in order to recover timing (phase and frequency) of the received data stream. It also shows only binary transmission, but the same block diagram is applicable for multilevel pulse amplitude modulation. For instance, if the PAM-4 is used for the transmission, each slicer would have at least three comparators.

2.5 Skin Effect and Dielectric loss

As shown in the section Section 2.3, transmission lines cannot be treated as ideal conductors because they have nonzero resistance, conductance, capacitance, inductance and delay. Moreover, some of these parameters are function of the frequency such as resistance, inductance and conductance. Only the transmission line capacitance can be considered to be a constant over the frequency. Resistance of the conductor increases with frequency, because high-frequency current flows mostly on the surface of the conductor (hence the name "skin effect") with current density dropping off exponentially towards the center of the conductor (2.19) [24].

$$J = \exp\left(-\frac{d}{\delta}\right) \tag{2.19}$$

where d is the distance from the surface towards the center of the conductor and δ is the skin depth, defined as the distance from the surface where the current density has fallen by $\exp(-1)$ from its nominal value. δ is given by

$$\delta = (\pi f \mu \sigma)^{-1/2} \tag{2.20}$$

where f is the frequency, μ is the magnetic permeability of the conducting material and σ is the conductivity of the conducting material.

This is shown graphically for a stripline in Figure 2.13 where shaded areas in the inner trace and the outer plates depict current densities — darker the area, the higher the current density. Although the current density drops exponentially, we usually approximate this by assuming that the current flows uniformly up to the skin depth and zero towards the center of the conductor. Hence, the resistance of a strip line [24] can be approximated with

$$R = R_{DC} + R(f) = R_{DC} + R_{AC} = \frac{1}{wt\sigma} + \frac{1}{2w\rho\sigma} = \frac{1}{wt\sigma} + \frac{(\pi\mu/\sigma)^{1/2}}{2w}\sqrt{f} \tag{2.21}$$

where w is the width and t is the thickness of the stripline.

For instance a 0.2 mm wide and 18 μm thick strip line has DC resistance of 4.6 Ω/m. At 1 GHz the effective resistance is 28 Ω/m.

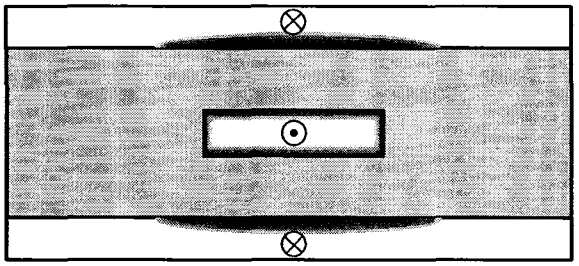

Figure 2.13: Skin effect in stripline

Dielectric materials used as insulators in transmission lines have negligible conductance at low frequencies. However, at higher frequencies, good insulators such as glass or plastic may consume significant energy in alternating fields because of dielectric hysteresis which is analogous to magnetic hysteresis in ferromagnetic materials.

Lets assume that an alternating voltage source V is connected to two copper planes with area S, separated by a dielectric material with conductivity σ, permittivity ε, and thickness d. Then the current through the dielectric material can be expressed with [24]

$$I = V\frac{S}{d}(\sigma + j\omega\varepsilon) \tag{2.22}$$

Here we have two current components: one in-phase with the voltage $V\frac{S}{d}\sigma$ and the other in-quadrature $V\frac{S}{d}j\omega\varepsilon$. The first one shows how much dielectric behaves as a resistor (burns the power) and the other how much it behaves as a capacitor.

However, at high frequency alternating fields [24], the dielectric permittivity becomes complex and (2.22) becomes

$$I = V\frac{S}{d}\sigma + j\omega(\varepsilon' + j\varepsilon'') = V\frac{S}{d}[(\sigma + \omega\varepsilon'') + j\omega\varepsilon'] = V\frac{S}{d}(\sigma' + j\omega\varepsilon') \tag{2.23}$$

where $\varepsilon = \varepsilon' + j\varepsilon''$, ε' is real or lossless part of ε and ε'' is imaginary part or lossy part of ε. For a good dielectric material ε'' is very small so that for low frequencies $\omega\varepsilon''$ is negligible. However for higher frequencies $\omega\varepsilon''$ cannot be ignored and power losses due to dielectric hysteresis can be significant.

Quality of dielectric material is usually specified with the loss tangent defined as

$$\tan\delta = \frac{\sigma + \omega\varepsilon''}{\omega\varepsilon'} \tag{2.24}$$

The lower the value of the loss tangent the better the dielectric material.

Because of the loss in the dielectric material, the distributed conductance G, used in transmission line models is also a function of frequency. As can be seeing from (2.22), the loss in dielectric material is directly proportional to the frequency. Hence, the distributed conductance in the transmission line model [24] can be written as

$$G = G_{DC} + G(f) = G_{DC} + G_{AC} \tag{2.25}$$

and the distributed conductance for a strip line in Figure 2.13 can be written as

$$G = \frac{w}{d}\sigma + \frac{w}{d}(2\pi\varepsilon'') \cdot f \tag{2.26}$$

It should be noted that we ignored fringing fields in deriving simple (2.21) and (2.26) equations. To get accurate values of distributed parameters, one should use 2-D electromagnetic simulator which uses numerical methods. The transmission line simulation models used in this thesis have been derived with Linpar 2-D electromagnetic simulator [25].

The distributed inductance L in transmission line modes is also a function of frequency. The inductance of a transmission line depends on magnetic flux inside and outside of the conductor and can be expressed as $L_{tot} = L_{int} + L_{ext}$. The internal inductance is proportional to the cross section of the conductor though which current flows. At higher frequencies — due to skin effects — this cross section becomes negligible and so does the

internal inductance. The external inductance is not a function of frequency. Hence, for higher frequencies (f > 10MHz), the distributed inductance can be considered constant (not a function of frequency). This is specifically true for simulation of the transmission lines used for multi-giga bit per second transmission. However, for simulations of long twisted pair lines used for transmission of modest data rates where signal energy is concentrated below 10MHz such as in X-DSL, the frequency dependence of the distributed inductance cannot be ignored.

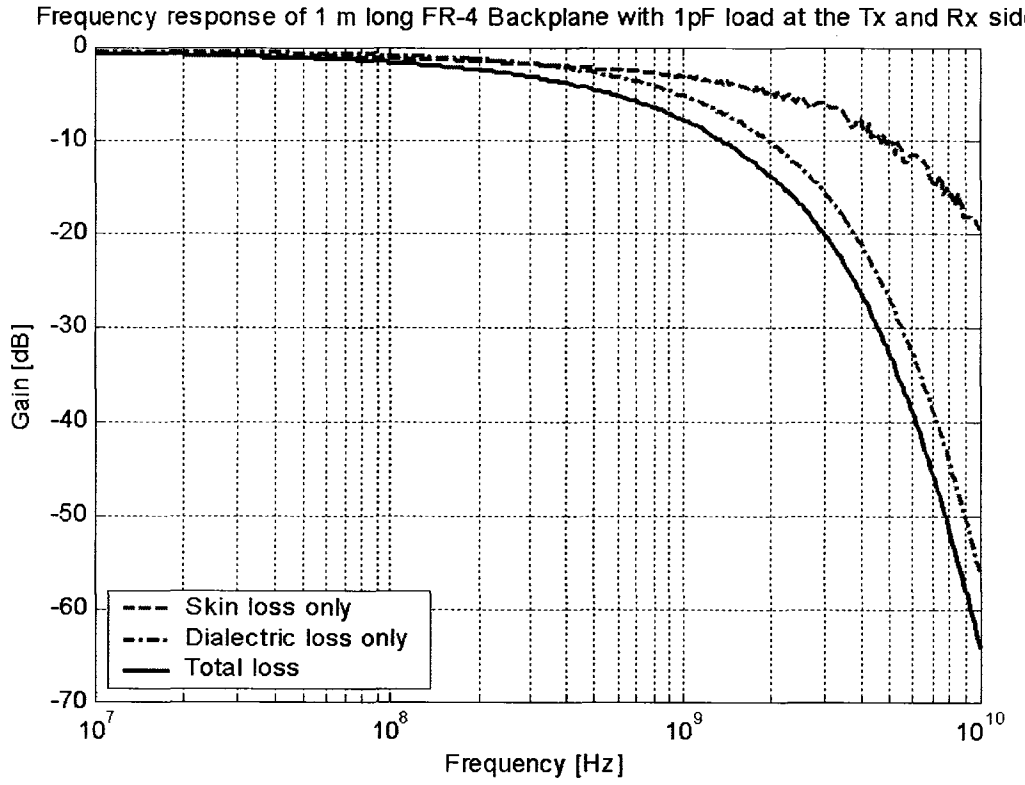

Figure 2.14: Frequency response of the 1 m long FR-4 backplane

Figure 2.14 shows frequency response of 1 meter long FR-4 differential trace where the attenuation due to the skin effect and the dielectric loss is separated. The transmission line is modeled and simulated with 2-D electromagnetic simulator and HSPICE as described in CHAPTER 4. We see that attenuation due to the dielectric loss is more severe (it is propor-

tional to the frequency as opposed to square root of frequency (2.25) vs. (2.21)). However, the attenuation due to skin effects starts at lower frequencies because the constant R_{AC} is larger than the constant G_{AC}.

The time response of this channel to a 250 ps wide pulse is illustrated in Figure 2.15.

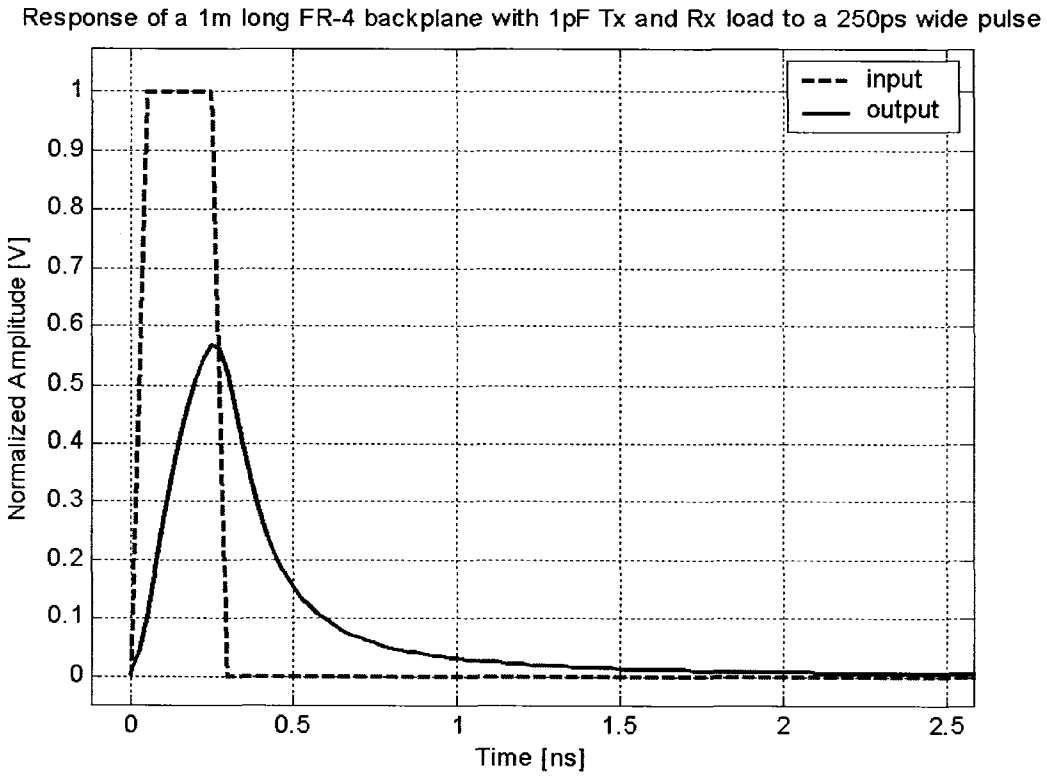

Figure 2.15: Time response of the 1m long FR-4 backplane

2.6 Inter Symbol Interference

ISI is a dominant impairment in multi gigabit transmission over PCB backplanes. It is caused by frequency dependent attenuation and frequency dependent delay in the transmission lines.

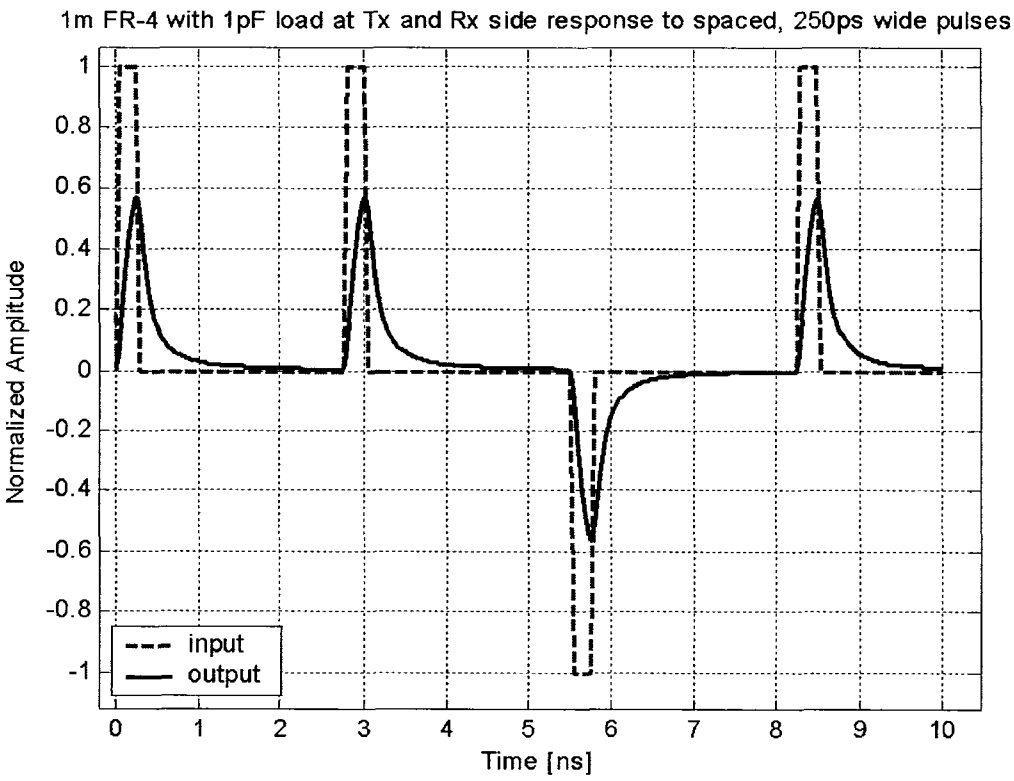

Figure 2.16: Illustration of ISI

Figure 2.16 illustrates transmission of 250 ps wide pulses over the transmission line model described in the previous section. Solid line represents received symbols, and the dashed, transmitted symbols. If we increase the rate of transmission (reduce the time between the transmitted pulses) as shown in Figure 2.17, we can see that received pulses (thin dash-dot lines) overlap with each other and that the resulting received signal (solid line) is obtained by adding all four received symbols (1101). We can see that the third received symbol ("0") is seriously affected by the previous two symbols because it has very little noise margin (distance from the zero crossing).

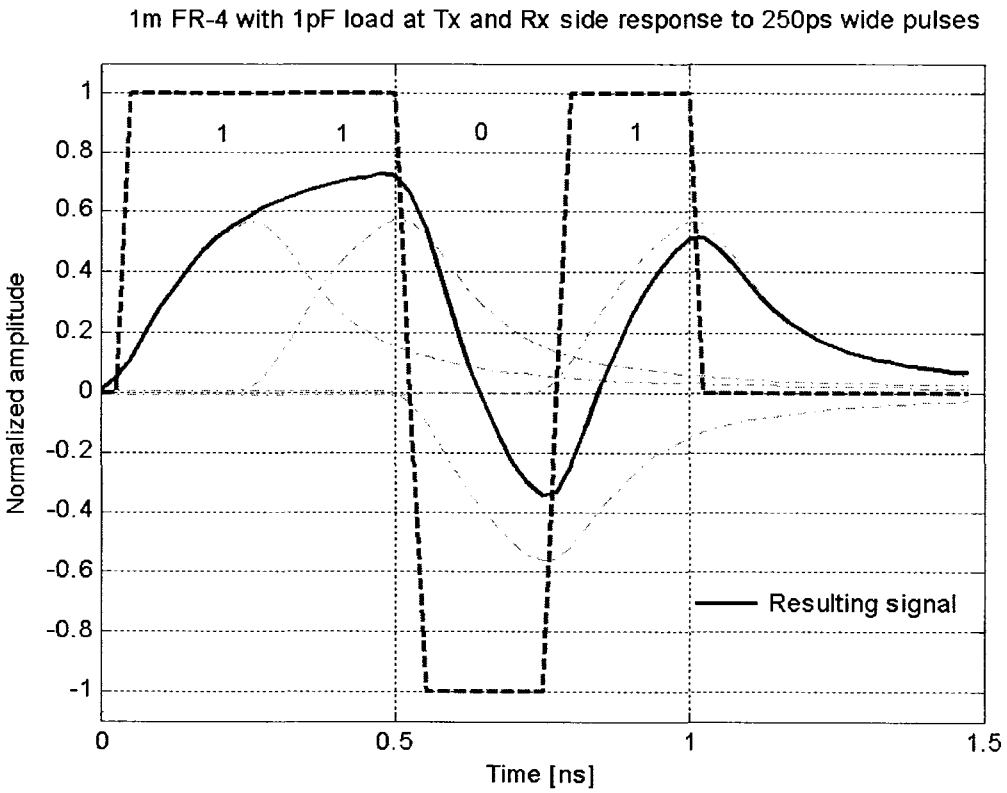

Figure 2.17: Illustration of ISI cont.

If all received pulses are overlapped one over the other we can get an eye-diagram which can help us to visually quantify severity of the ISI as shown in Figure 2.18.

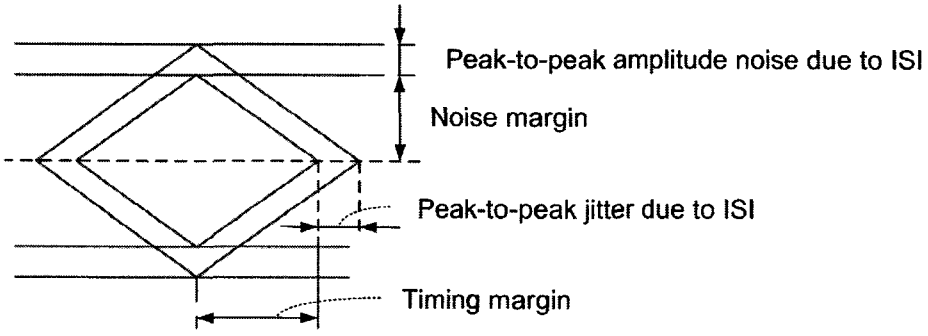

Figure 2.18: Eye-diagram for binary transmission

Peak to peak amplitude noise caused by ISI closes the vertical eye-opening and reduces the available additive noise margin. At the same time, the jitter caused by ISI closes the eye-opening horizontally end reduces the available jitter margin of the sampling clock.

Example of the eye-opening diagram at the end of the channel described in Section 2.5 if shown in Figure 2.19.

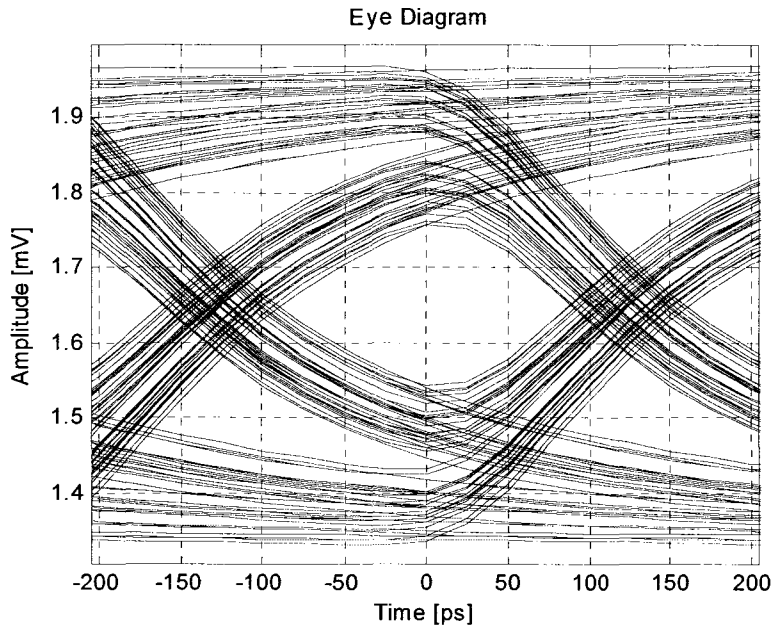

Figure 2.19: Eye-opening for 1m long FR-4 backplane

Although the transmission line modeling in analog domain, described in the previous section, is used extensively in the circuit simulation (HSPICE), it is also convenient for analysis purposes to model transmission lines in digital domain, because filters used to mitigate ISI are typically implemented in digital domain. A transmission line can be viewed in digital domain as a low pass filter with impulse response $h(n)$.

The received signal y_n can be written as

$$y_n = \sum_{k=0}^{\infty} x_k h_{n-k} + n_n \tag{2.27}$$

where x_k is transmitted sequence, h_k is channel impulse response, and n_n is additive noise. If we rewrite (2.27)

$$y_n = x_n h_0 + \sum_{\substack{k=0 \\ k \neq n}}^{\infty} x_k h_{n-k} + n_n \tag{2.28}$$

we get $x_n h_0$ desired transmitted data x_n, scaled by factor h_0, the term

$$\sum_{\substack{k=0 \\ k \neq n}}^{\infty} x_k h_{n-k} \tag{2.29}$$

which represent intersymbol interference and n_n is additive gaussian noise.

There are several different ways to quantify severity of the ISI and quality of different methods for mitigating ISI. The most common are: Probability of error and signal to noise ratio. However, in this thesis we use the eye-opening, which is a very simple, frequently used method, for quantifying ISI and residual ISI in multi-gigabit backplane applications. Quantifying ISI with eye-opening has being described in [21]. Percentage of eye-opening can be calculated from

$$EyeOpening = \frac{h_0 - (N-1)\sum_{k \neq 0} |h_k|}{h_0} \cdot 100\% \tag{2.30}$$

where h_0 is central (cursor) component, $h_k, k \neq 0$ are pre-cursor and post-cursor ISI components and N is the number of PAM levels. Negative *EyeOpening* means that the eye is completely closed. As as example, the eye opening for 1 m long FR-4 backplane with 1pF capacitive loads is 33% open for PAM-2, but it is fully closed for PAM-4 (-1%). This is expected because PAM-4 (PAM-N) has three (N-1) times lower noise margin than PAM-2 for the same symbol rate.

2.6.1 Equalization and Pre-emphasis

ISI is typically mitigated with a filter with characteristics close to the inverse characteristic of the transmission line so that the magnitude of the equivalent system (transmission line in series with the filter) is constant with frequency. How close this filter is to the inverse characteristic of the transmission line will depend on the compromise between noise enhancement, ISI and realizability of the inverse filter. For instance, if a transmission line has zeros in frequency domain, then its inverse does not exists, because inverse filter would need to have infinite gain at these frequencies.

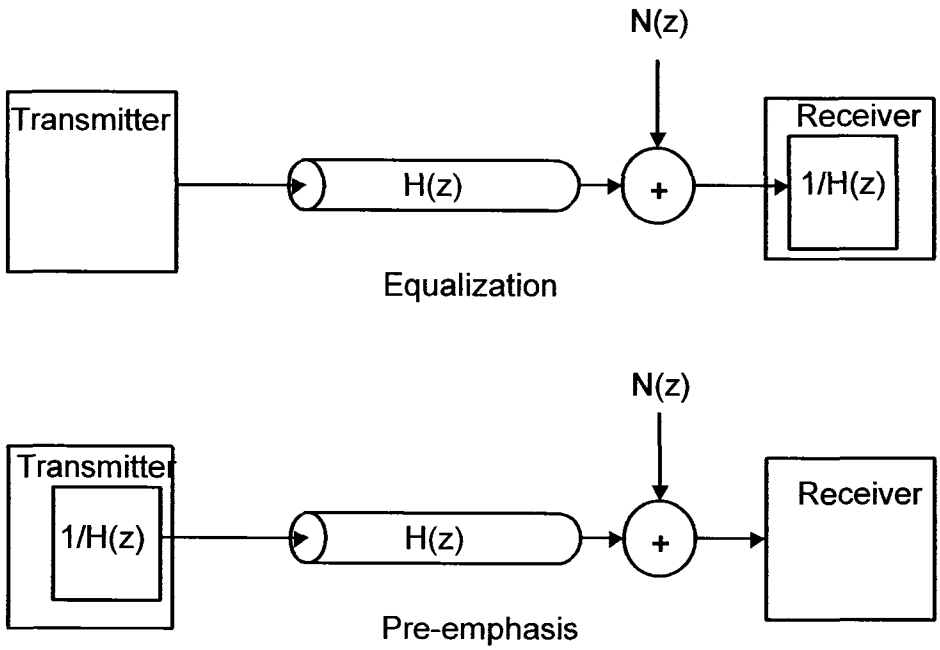

Figure 2.20: Equalization and pre-emphasis

If this filter is placed at the transmitter, it is called a pre-emphasis filter, and if it is placed at the receiver it is called an equalization filter (Figure 2.20).

Both of these methods have their own advantages and disadvantages. If we ignore implementation complexity (which depends on applications), pre-emphasis filter has

advantage that it does not amplify the noise because filtering is done on noise free signal before it is transmitted over the transmission medium as can be seen in Figure 2.21.

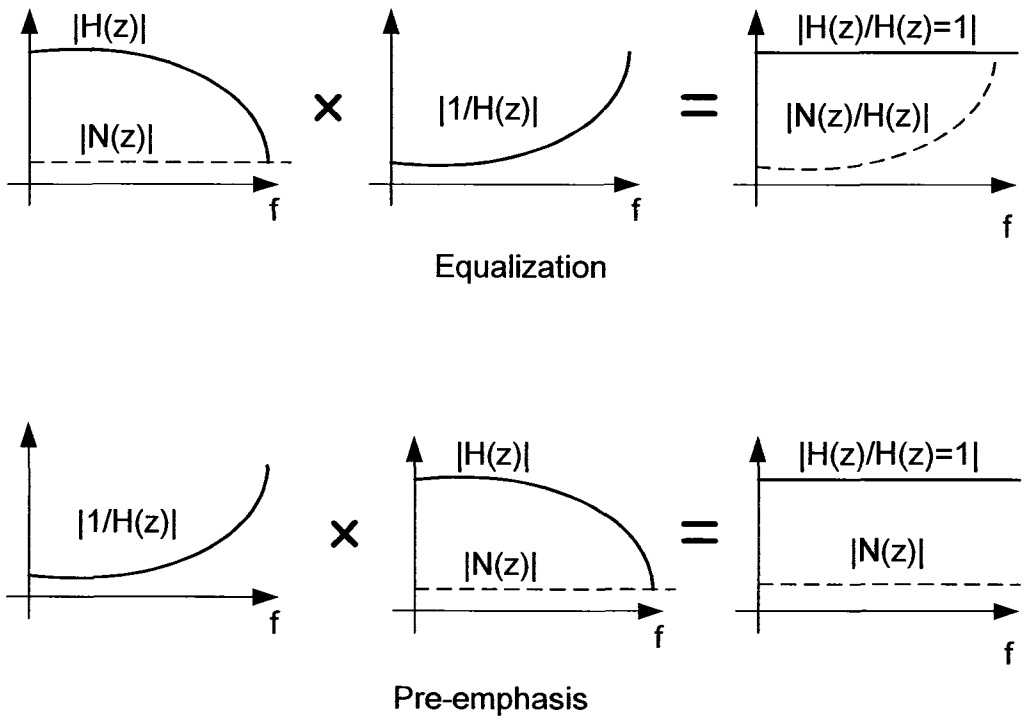

Figure 2.21: Noise performance for equalization and pre-emphasis

At the same time it has two major disadvantages

- Because of the limited swing of the output drivers, the pre-emphasis is performed not by amplifying the high frequency components of the transmitted signal but rather by attenuating the lower frequency components. This in turn reduces the power of the transmitted signal as well as the maximum eye opening at the receiver.
- Adaptive pre-emphasis needs a reliable return channel (from the receiver to the transmitter) because the coefficients for pre-emphasis filter are calculated at the receiver side.

Pre-emphasis have been used for years in broadcast systems (TV for instance) because it is much cheaper to add filter at single broadcast transmitter, rather than to add it at every receiver.

Recently, pre-emphasis has started to be used in multi-gigabit digital transmission over backplanes, because it can be implemented relatively simple by summing currents from different FIR filter taps in the output pad as shown in Figure 2.22.

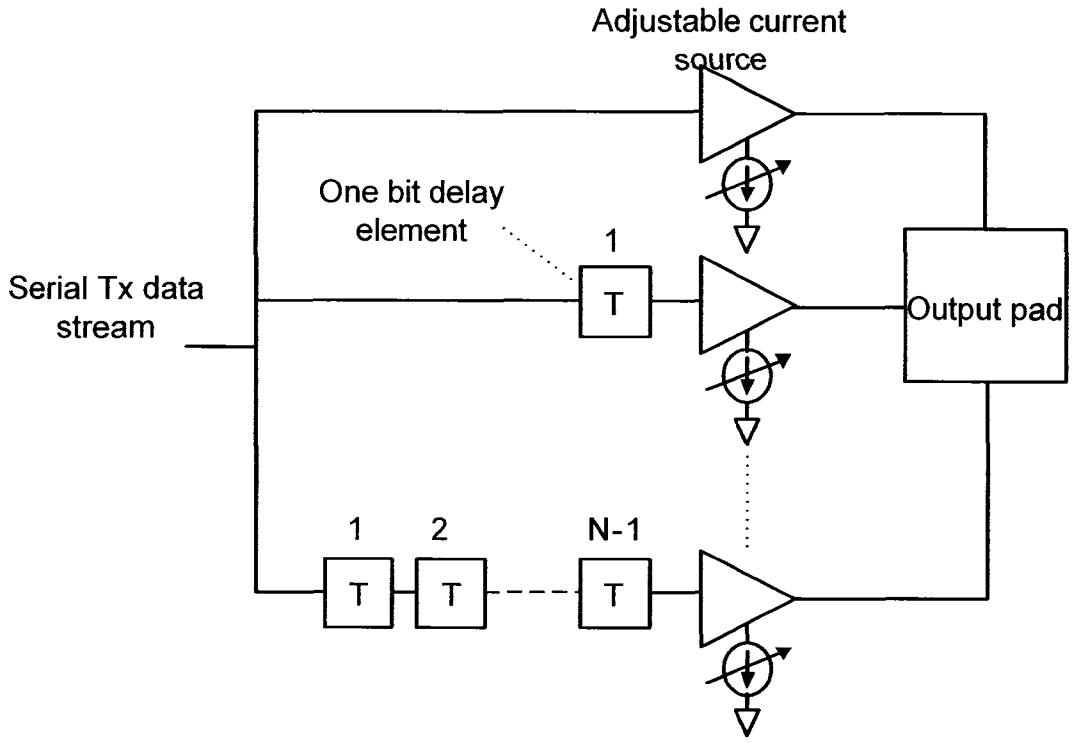

Figure 2.22: Block diagram of an N tap pre-emphasis filter

Above figure shows a Tx pre-emphasis filter with N taps but the typical number for high-speed transmitters is between 2 and 5. The gain of pre-emphasis filter taps are adjusted by adjusting the current of the output buffers.

An equivalent linear equalization filter is very difficult to implement at multi-giga bit per second rates so that the receiver equalization is not used in the practice.

Optimum filter (either pre-emphasis or equalization) coefficients can be found from the following:

If we write the equation (2.27) in matrix form [20], we get

$$\boldsymbol{Y}_n = \boldsymbol{X}_n\boldsymbol{H} + \boldsymbol{N}_n \tag{2.31}$$

or

$$\boldsymbol{Y}_n = \begin{bmatrix} y_n \\ y_{n-1} \\ \cdots \\ y_{n-f+1} \end{bmatrix} = \begin{bmatrix} h_0 & h_1 & \cdots & h_v & 0 & \cdots & 0 & 0 \\ 0 & h_0 & h_1 & \cdots & h_v & 0 & 0 & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & 0 & 0 & h_0 & h_1 & \cdots & h_v \end{bmatrix} \begin{bmatrix} x_n \\ x_{n-1} \\ \cdots \\ x_{n-f-v+1} \end{bmatrix} + \begin{bmatrix} n_n \\ n_{n-1} \\ \cdots \\ n_{n-f-v+1} \end{bmatrix} \tag{2.32}$$

$\boldsymbol{Y}_n$ is the channel output where v is the length of the channel response and f is number of consecutively transmitted symbols, $\boldsymbol{H}$ is the channel response matrix with size $(f) \times (f+v)$, $\boldsymbol{X}_n$ is the transmitted data sequence and $\boldsymbol{N}_n$ is Gaussian noise.

From the system theory — ignoring the noise — we know that there is no difference in the output values if the filter is placed before (pre-emphasis) or after (equalization) transmission line. Assuming that we have an equalization filter with coefficients given with vector $\boldsymbol{w}$ (this will help in next section where we talk about decision feedback equalization), the output of the filter is

$$z_n = \boldsymbol{w}\boldsymbol{Y}_n \tag{2.33}$$

The optimum coefficients $\boldsymbol{w}$ can be obtained by minimizing Mean Squared Error (MSE)

$$\sigma_{MSE} = E\left\{|e_n|^2\right\} = E\left\{|x_{n-\Delta} - z_n|^2\right\} \tag{2.34}$$

where E is expected value, $x_{n-\Delta}$ is desired received symbol and Δ is the channel delay. From the orthogonality principle [20], the MSE is minimized when the expected value of the product between the error and the received symbols is equal to zero

$$E\left\{e_n \boldsymbol{Y}_n^T\right\} = 0 \tag{2.35}$$

In other words, the channel will be optimally equalized in MSE sense, when the received error $e_n = x_{n-\Delta} - z_n$ is uncorrelated with the channel output $\boldsymbol{Y}_n$. Hence [20],

$$E\left\{x_{n-\Delta}\boldsymbol{Y}_n^T\right\} - \boldsymbol{w}E\left\{\boldsymbol{Y}_n\boldsymbol{Y}_n^T\right\} = \boldsymbol{R}_{x\boldsymbol{Y}^T} - \boldsymbol{w}\boldsymbol{R}_{\boldsymbol{Y}\boldsymbol{Y}^T} = 0 \tag{2.36}$$

where $\boldsymbol{R}_{x\boldsymbol{Y}^T}$ is cross correlation between the input and the output of the channel and $\boldsymbol{R}_{\boldsymbol{Y}\boldsymbol{Y}^T}$ is autocorrelation of the channel output. From (2.36) we have

$$\boldsymbol{w} = \boldsymbol{R}_{x\boldsymbol{Y}^T}(\boldsymbol{R}_{\boldsymbol{Y}\boldsymbol{Y}^T})^{-1} \tag{2.37}$$

where,

$$\boldsymbol{R}_{x\boldsymbol{Y}^T} = E\left\{x_{n-\Delta}\boldsymbol{Y}_n^T\right\} = E\left\{x_{n-\Delta}\boldsymbol{X}_n^T\right\}\boldsymbol{H}^T = \left[0 \ \ldots \ 0 \ \varepsilon_x \ 0 \ \ldots \ 0\right]\boldsymbol{H}^T = \varepsilon_x 1_\Delta \boldsymbol{H} \tag{2.38}$$

where ε_x is the energy of the input symbol and $1_\Delta \boldsymbol{H}$ is short form of

$$1_\Delta \boldsymbol{H} = \left[0 \ \ldots \ 0 \ h_v \ \ldots \ h_0 \ 0 \ \ldots \ 0\right] \tag{2.39}$$

and

$$\boldsymbol{R}_{\boldsymbol{Y}\boldsymbol{Y}^T} = E\left\{\boldsymbol{Y}_n\boldsymbol{Y}_n^T\right\} = \boldsymbol{H}E\left\{\boldsymbol{X}_n\boldsymbol{X}_n^T\right\}\boldsymbol{H}^T = \varepsilon_x \boldsymbol{H}\boldsymbol{H}^T \tag{2.40}$$

because

$$E\left\{\boldsymbol{X}_n\boldsymbol{X}_n^T\right\} = E\left\{\begin{bmatrix} x_n \\ x_{n-1} \\ \ldots \\ x_{n-f-v+1} \end{bmatrix}\begin{bmatrix} x_n \ x_{n-1} \ \cdots \ x_{n-f-v+1} \end{bmatrix}\right\} = \varepsilon_x \begin{bmatrix} 1 & 0 & \ldots & 0 \\ 0 & 1 & 1 & 0 \\ \ldots & \ldots & \ldots & \ldots \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.41}$$

We can now rewrite the equation (2.37)

$$\boldsymbol{w} = 1_\Delta \boldsymbol{H}(\boldsymbol{H}\boldsymbol{H}^T)^{-1} \tag{2.42}$$

From this equation we can calculate optimum pre-emphasis or equalization filter coefficients for given number of filter taps. This equation as well as the equation (2.30) can be used to estimate minimum number of filter taps for specific channel. More taps will always better suppress ISI. However, the number of taps is usually chosen as a compromise between circuit complexity (power consumption, cost) and performance. For instance, Figure 2.23 shows relationship between the eye-opening and the number of filter taps for PAM-2 and PAM-4 transmitted over 1m long FR-4 channel with 1pF capacitive load at the input and the output.

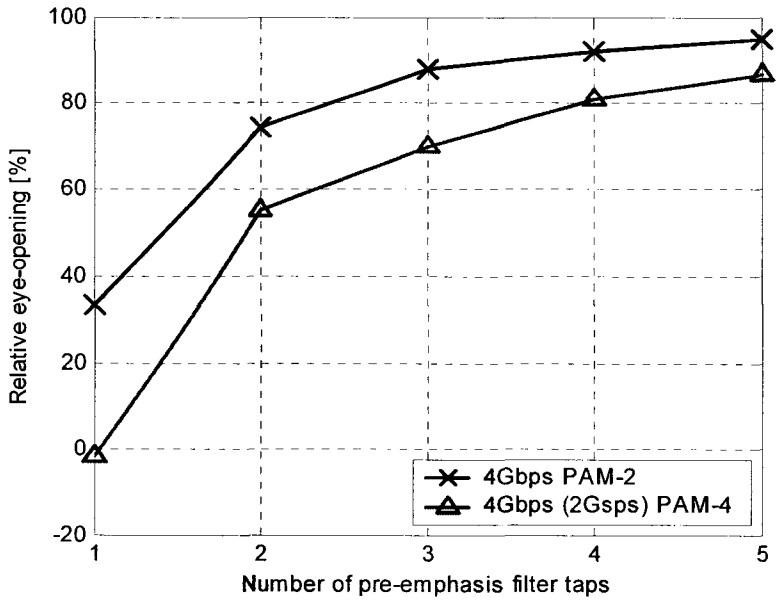

Figure 2.23: Eye-opening vs. number of pre-emphasis filter taps

We can see that for 4Gbps transmission (4Gsps for PAM-2 and 2Gsps for PAM-4), PAM-2 have wider relative eye-opening regardless of the number of pre-emphasis filter taps. This can be explained by noting that although the symbol rate for PAM-4 is two times lower than for PAM-2 (2Gsps vs. 4Gsps in our case), its noise margin is three times lower as well.

Another form of equalization filter called decision feedback equalization filter can be implemented in digital domain, which is subject of this thesis. Following section briefly describes basics of DFE, while implementation detail is given in the Chapter 3.

2.6.2 Decision Feedback Equalization

DFE is a non-linear equalization method where previously detected symbols are used to cancel ISI in the present symbol as shown in Figure 2.24. In general, the DFE filter is preceded with a linear equalization filter (dashed area) which is used to cancel all pre-cursor ISI components because DFE filter can cancel only post-cursor ISI components. This linear filter is not the must in multi-giga bit per second transmission over FR-4 back-planes because pre-cursor ISI components are relatively small in these applications.

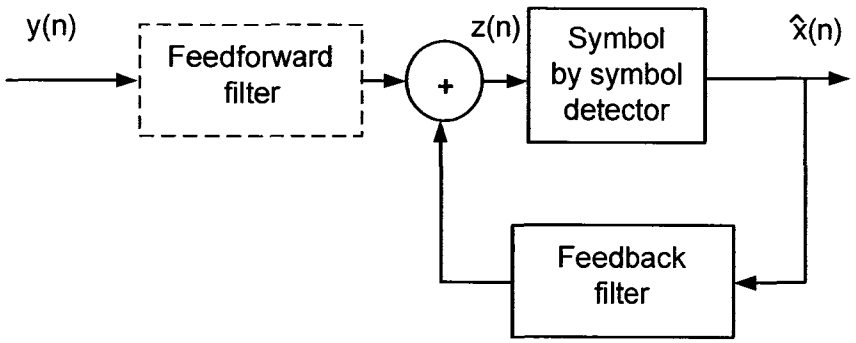

Figure 2.24: Decision Feedback Equalization

It should be pointed out that pre-cursor components are relatively negligible only for PAM-2. However, the pre-cursor components may not be possible to ignore in PAM-4, which is also used in backplane applications, because PAM-4 has three times lower noise margin than PAM-2. In this case, the feedforward filter can be moved to the transmitter side where it is easy to implement (pre-emphasis filter). This implementation would still require passing the coefficients from receiver to transmitter in order to adaptively train pre-emphasis feedforward filter.

We will now derive equations for optimum DFE filter coefficients for both feedback and feedforward filter. This derivation is very similar to one for linear pre-emphasis/ equalization filter derived in previous section with addition of feedback filter.

MSE for DFE [20] can be written as

$$\sigma_{MSE} = E\left\{|e_n|^2\right\} = E\left\{|x_{n-\Delta} - z_n - \boldsymbol{b}_n \boldsymbol{x}_{n-\Delta+1}|^2\right\} \tag{2.43}$$

where $\boldsymbol{b}_n$ is a vector containing feedback filter coefficients $\boldsymbol{b}_n = \left[b_0\ b_1\ \ldots\ b_b\right]$ and $\boldsymbol{x}_{n-\Delta+1}$ is a column vector containing detected symbols in the feedback path.

To make the derivation identical to the one in previous section we define.

$$\tilde{\boldsymbol{w}}_n = [\boldsymbol{w}_n | -\boldsymbol{b}_n] \tag{2.44}$$

$$\tilde{\boldsymbol{Y}}_n = \begin{bmatrix} \boldsymbol{Y}_n \\ \boldsymbol{x}_{n-\Delta+1} \end{bmatrix} \tag{2.45}$$

Now derivation for MSE is analogous to (2.36)

$$E\left\{x_{n-\Delta}\tilde{\boldsymbol{Y}}_n^T\right\} - \tilde{\boldsymbol{w}} E\{\tilde{\boldsymbol{Y}}_n \tilde{\boldsymbol{Y}}_n^T\} = \boldsymbol{R}_{x\tilde{Y}^T} - \tilde{\boldsymbol{w}} \boldsymbol{R}_{\tilde{Y}\tilde{Y}^T} = 0 \tag{2.46}$$

From this equation we have

$$\tilde{\boldsymbol{w}} = \boldsymbol{R}_{x\tilde{Y}^T} (\boldsymbol{R}_{\tilde{Y}\tilde{Y}^T})^{-1} \tag{2.47}$$

where [20]

$$\boldsymbol{R}_{x\tilde{Y}^T} = E\left\{x_{n-\Delta}\tilde{\boldsymbol{Y}}_n^T\right\} = \left[\boldsymbol{R}_{xY^T}\ 0\right] = \left[\varepsilon_x 1_\Delta \boldsymbol{H}\ 0\right] \tag{2.48}$$

$$\boldsymbol{R}_{\tilde{Y}\tilde{Y}^T} = E\{\tilde{\boldsymbol{Y}}_n \tilde{\boldsymbol{Y}}_n^T\} = \begin{bmatrix} \boldsymbol{R}_{YY^T} & E\{\boldsymbol{Y}_n \boldsymbol{x}_{n-\Delta+1}\} \\ E\left\{\boldsymbol{x}_{n-\Delta+1} \boldsymbol{Y}^T{}_n\right\} & \varepsilon_x \boldsymbol{I}_b \end{bmatrix} = \begin{bmatrix} \boldsymbol{R}_{YY^T} & \varepsilon_x \boldsymbol{H} \boldsymbol{J}_\Delta \\ \varepsilon_x \boldsymbol{H} \boldsymbol{J}_\Delta & \varepsilon_x \boldsymbol{I}_b \end{bmatrix} \tag{2.49}$$

I_b is an identity matrix and J_Δ is matrix with dimensions $(f+v)\times b$ whose elements are ones and zeros which has $\Delta+1$ upper rows equal to zero and an identity matrix with dimensions $min(b, f+v-\Delta-1)$.

If we include equations (2.48) and (2.49) into equation (2.47) we have

$$\begin{bmatrix} w & -b \end{bmatrix} \begin{bmatrix} R_{YY^T} & \varepsilon_x HJ_\Delta \\ \varepsilon_x HJ_\Delta & \varepsilon_x I_b \end{bmatrix} = \begin{bmatrix} \varepsilon_x 1_\Delta H & 0 \end{bmatrix} \tag{2.50}$$

which is essentially two equations with two unknown variables.

$$wR_{YY^T} - b(\varepsilon_x HJ_\Delta) = \varepsilon_x 1_\Delta H \tag{2.51}$$

$$w(\varepsilon_x HJ_\Delta) - b\varepsilon_x = 0 \tag{2.52}$$

Solving these two equations with respect to w and b we get the optimum coefficient values for the feedforward and feedback filters respectively.

$$w = 1_\Delta^T H^T (HH^T - HJ_\Delta J_\Delta^T)^{-1} \tag{2.53}$$

$$b = wHJ_\Delta = 1_\Delta^T H^T (HH^T - HJ_\Delta J_\Delta^T)^{-1} HJ_\Delta \tag{2.54}$$

From equation (2.54) we see that if a Decision Feedback Equalizer contains only feedback filter, then (2.54) reduces to $b = HJ_\Delta$.

2.7 Adaptive Pre-emphasis and Equalization

Although the coefficient calculation methods presented in previous section provide us with optimum coefficient values, they are usually used only for analysis purposes and not for real applications, because they require knowledge of the impulse response of the channel. However, impulse response may not be known apriori and even it is known, the channel characteristics may change with the time due to temperature and voltage variations as well as aging. This section will describe adaptive methods used for pre-emphasis coefficient calculation in backplane transceiver applications.

Generally, adaptive pre-emphasis/equalization methods use gradient iterative method for calculating the filter coefficients. This method relies on the fact that the MSE is a positive convex function of pre-emphasis/equalization filter coefficients with only one minimum value (no local minimums) [21].

$$\sigma_{MSE} = E\left\{|e_n|^2\right\} = E\left\{|x_{n-\Delta} - \boldsymbol{w}\boldsymbol{Y}_n|^2\right\} \tag{2.55}$$

Initially, we set a filter coefficients to some arbitrary value $\boldsymbol{w} = \boldsymbol{w}_o$, which corresponds to point on the MSE surface possibly far from the minimum value. Now, we calculate the gradient of MSE ($\partial\sigma_{MSE}/\partial\boldsymbol{w}|_{\boldsymbol{w}_0}$), which is a vector that points to direction of the maximum change of the MSE function. We than move to the opposite direction of the calculated vector towards the minimum of the MSE, by calculating the new filter coefficient $\boldsymbol{w} = \boldsymbol{w}_1$. This procedure is iteratively repeated until we reach the minimum value. Iterative equation is given with equation (2.56) [21]

$$\boldsymbol{w}_{k+1} = \boldsymbol{w}_k - \Delta\left(\frac{\partial\sigma_{MSE}}{\partial\boldsymbol{w}}\bigg|_{\boldsymbol{w}_k}\right) \tag{2.56}$$

where Δ is a positive coefficient. Δ needs to be selected small enough to ensure convergence of the iterative algorithm and to minimize the error once the algorithm converges to the minimum value but also big enough to satisfy convergence time for a particular application.

Gradient of the MSE is [21]

$$\frac{\partial\sigma_{MSE}}{\partial\boldsymbol{w}} = \frac{\partial}{\partial\boldsymbol{w}}\left(E\left\{|x_{n-\Delta} - \boldsymbol{w}\boldsymbol{Y}_n|^2\right\}\right) = E\left\{\frac{\partial}{\partial\boldsymbol{w}}(|x_{n-\Delta} - \boldsymbol{w}\boldsymbol{Y}_n|^2)\right\} = -2E\{e_n\boldsymbol{Y}_n\} \tag{2.57}$$

This can be further simplified by replacing the expected value of the function with the function itself

$$\frac{\partial\sigma_{MSE}}{\partial\boldsymbol{w}} = -2e_n\boldsymbol{Y}_n \tag{2.58}$$

Now the iterative equation simplifies to [21]

$$\boldsymbol{w}_{n+1} = \boldsymbol{w}_n + \Delta e_n \boldsymbol{Y}_n \tag{2.59}$$

where multiplier 2 was included (absorbed) into the coefficient Δ.

There are number of different variations of this algorithm [21]. For high speed applications one could take just polarity of the error $sign(e_n)$ or just polarity of the channel output $sign(\boldsymbol{Y}_n)$ as shown below

$$\boldsymbol{w}_{n+1} = \boldsymbol{w}_n + \Delta sign(e_n)\boldsymbol{Y}_n \tag{2.60}$$

$$\boldsymbol{w}_{n+1} = \boldsymbol{w}_n + \Delta e_n sign(\boldsymbol{Y}_n) \tag{2.61}$$

or polarity of both the error $sign(e_n)$ and the channel output [21]

$$\boldsymbol{w}_{n+1} = \boldsymbol{w}_n + \Delta sign(e_n) sign(\boldsymbol{Y}_n) \tag{2.62}$$

These algorithms are very easy to implement and can run at very high data rate, but they have slower convergence. This is specially true for the last algorithm (called sign-sign iteration algorithm), because we use only sign of the gradient vector and ignore its amplitude. Hence, more steps are needed to reach to the minimum MSE.

The accuracy of convergence might be improved (estimation noise reduced) if the gradient is averaged over number of received samples [21] before the new coefficient values are calculated. For instance, if the gradient is averaged over N consecutive samples we have

$$\boldsymbol{w}_{N(n+1)} = \boldsymbol{w}_{Nn} + \Delta\left(\frac{1}{N}\sum_{k=0}^{N-1}(e_{Nn+k}\boldsymbol{Y}_{Nn+k})\right) \tag{2.63}$$

The drawback of averaging gradient algorithm is a slow convergence time. We will see in the next section how gradient averaging can be combined with sign-sign algorithm for adaptive transceiver pre-emphasis used in backplane applications.

2.7.1 Adaptive pre-emphasis used in backplane transceivers

Block diagram of the adaptive pre-emphasis is shown in Figure 2.25 where two transceivers communicate via the backplane. Figure 2.25 shows adaptive pre-emphasis only in one direction (from left to right) for simplicity, but the actual transceivers are usually identical and both contain adaptive pre-emphasis FIR filter as well as convergence engine for calculating the coefficients for pre-emphasis filters.

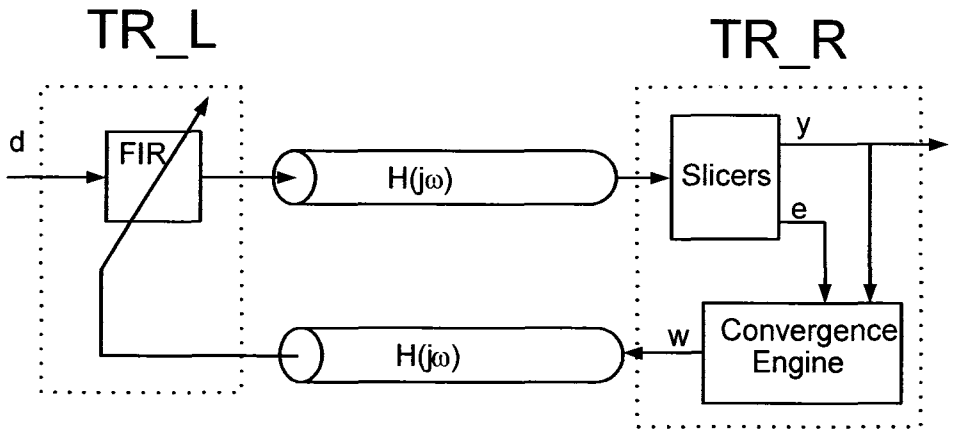

Figure 2.25: Block diagram of adaptive pre-emphasis

Initially, one side (lets assume it is the left side as shown in Figure 2.25) starts to transmit data at the nominal (multi-giga bit per second) data rate and the right side receives data distorted by the channel, and calculates the first iteration of pre-emphasis filter coefficients. Calculated coefficients are than transmitted from the right to the left side via a reliable return link. The return channel usually has the same characteristic as the transmit one but the data rate is much lower so that the channel does not distort the signal. This procedure is repeated until the pre-emphasis filter is fully tuned. Now, roles between the left and right side are reversed and the right side starts to transmit data at the nominal rate. The left side calculates the coefficients and transmits them back to the right side. After both sides finish tuning corresponding pre-emphasis filters, they can start communi-

cating with nominal speed in both direction. Coefficients calculation engines on both sides continue to calculate coefficients in order to track changes in the transmission path due to temperature, voltage and aging. The only difference is that the coefficient are now transmitted at the nominal rate because links are equalized (reliable). Coefficient transmission takes only a fraction of the available bandwidth and the remaining bandwidth is used for data transmission.

Adaptive pre-emphasis for backplane transceiver was first reported in [2]. This paper presented a backplane transceiver that uses PAM-4 and can operate up to 5Gbps (2.5Gsps) over FR-4 backplane. The pre-emphasis was achieved with 4-tap FIR. The filter has being implemented by summing the currents from four taps in the output pad. One tap is used for pre-cursor component, one for cursor and two taps for post-cursor components.

Coefficient calculation engine uses sign-sing averaging algorithm

$$w_{N(n+1)} = w_{Nn} + sign\left(\sum_{k=0}^{N-1} (sign(e_{Nn+k})sign(Y_{Nn+k}))\right) \tag{2.64}$$

The sign-sign algorithm with gradient averaging was used because multiplication can be done at very high rate because multiplier is simply an X-OR gate. The averaging was added not to increase the stability, but to reduce the data rate at which coefficient are relayed back to the transmitter.

Block diagram of the coefficient calculation engine is shown in Figure 2.26.

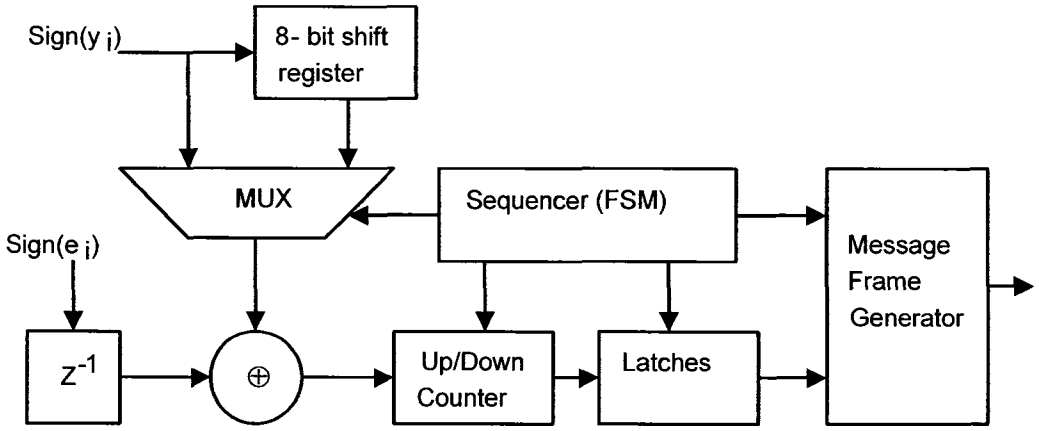

Figure 2.26: Block diagram of the coefficient calculation engine

Coefficients are calculated sequentially. For instance, the pre-cursor component is calculated by multiplying (XOR-ing) the latest y_i sample (before the shift register) with the error e_i delayed by one sample. Results of multiplication for consecutive errors e_i and inputs y_i are integrated (averaged) with an Up/Down counter. After N samples the averaged value is latched and stored in the message frame generator. Now the calculation of the cursor component can start by passing to the multiplier one bit delayed version of the y_i. After all four coefficients are calculated, the message frame generator package them in a frame and sends the frame back to the transmitter.

CHAPTER 3

Receiver Implementation

3.1 Introduction

This chapter describes implementation of a 4Gbps backplane receiver with adaptive blind DFE. First, we provide high level block diagram of the overall receiver and then explain the operation of the receiver's front-end with underlying math. Next, each block of the receiver' front-end is explained separately, starting from the DFE slicers, which are used to sample incoming data stream and perform DFE. Detailed schematics of the biased comparator used in DFE slicers is presented. This is followed with detailed description of the adaptive coefficient calculation engine. Finally, we conclude with description of the clock recovery unit, which is used to align the phase of the reference clock with the center of the symbol cell where the noise margin has a maximum value.

3.2 High-level Receiver Architecture

High-level receiver architecture block diagram is shown in Figure 3.1. The input serial stream is first equalized and demultiplexed to ease requirement for high-speed on chip clock. The demultiplexer takes snapshots of the eight consecutive data samples (sampled at the middle of the symbol cells) and eight corresponding transition samples (sampled at the edge of the symbol cells) and feeds them to synchronization block where they are used to tune the phase of the reference clock so that the incoming serial stream is sam-

pled at the middle of the symbol cells. Data samples are demultiplexed from 8 to 16 bit format to further relax the on-chip clock requirements.

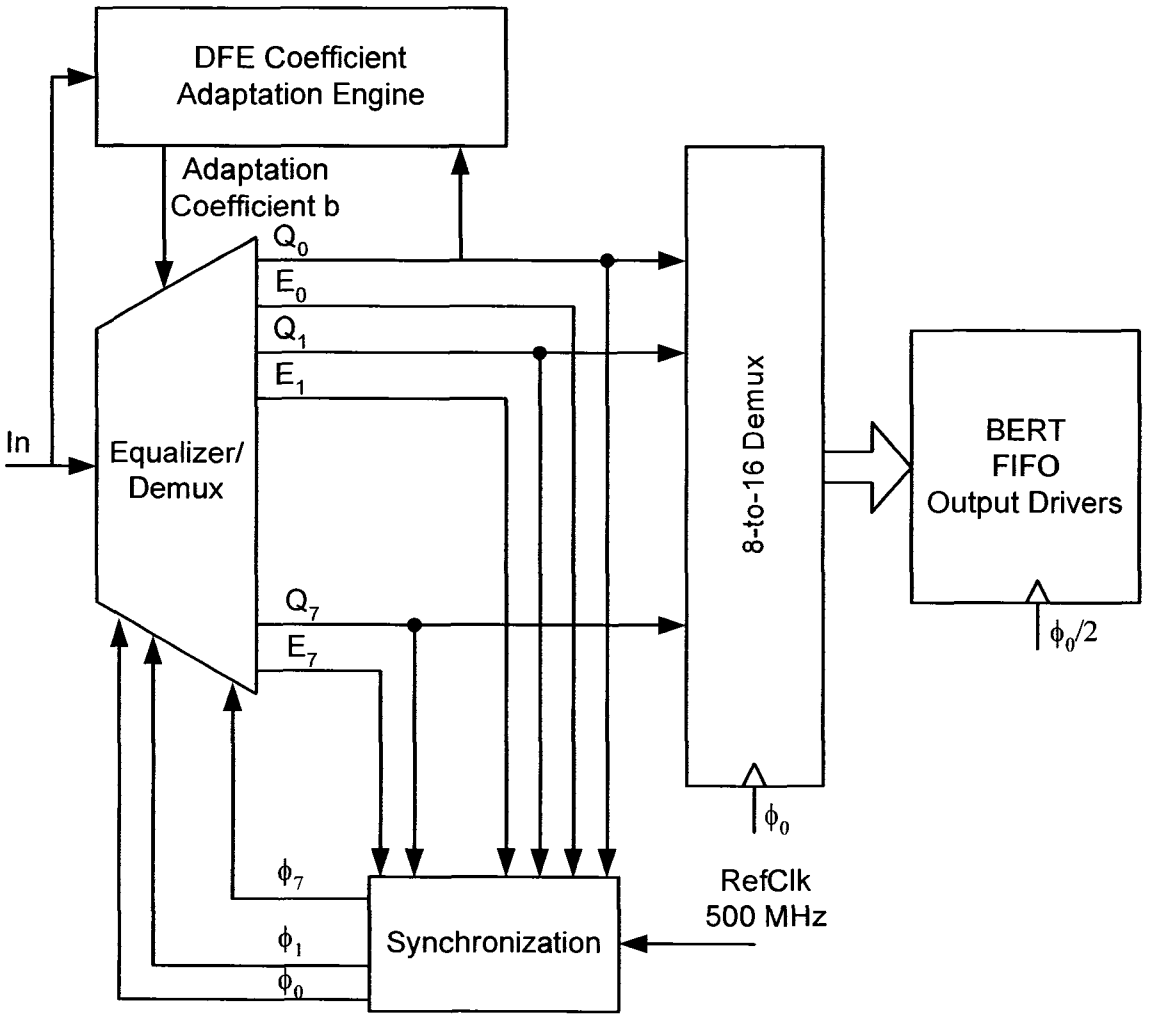

Figure 3.1: High-level block diagram of the receiver

Only one out of eight demultiplexed samples is used for DFE coefficient calculation, which is performed in adaptation engine. The adaptation engine constantly calculates DFE coefficient as well as the cost function coefficient which is used to compensate for the transmission line attenuation. The DFE coefficient is passed to the demultiplexer where it is used to cancel post-cursor ISI component.

Before we describe each part of the receiver in detail we should mention that all high speed mixed signal circuits (Equalizer/Demux, DFE error generation circuit) are designed at the transistor level, while lower speed circuits (DFE coefficient integrator, 8-to-10 Demux, BERT, FIFO) are designed at the behavioral level because they can be implemented with standard cell components. This approach significantly increased the speed of HSPICE simulation. The synchronization (clock recovery) circuit was also implemented at the behavioral level because the synchronization was not the main topic of this thesis.

3.3 Receiver Front-end Architecture

Block diagram of the receiver front-end is shown in Figure 3.2. The receiver front-end circuit makes decision about the received symbol based on the level of the received analog signal at the sampling instance and the previously detected symbol. Essentially, it subtracts the weighted previously received symbol from the present symbol and then makes decision by comparing the resulting value with the threshold level.

The received data is demultiplexed with eight DFE slicers to reduce the on-chip clock and data rate.

As we saw in the previous Chapter, the received signal y_n can be written as

$$y_n = \sum_{k=0}^{L} h_k x_{n-k} + n_n \tag{3.1}$$

where x_n is transmitted sequence, h_n is channel impulse response, and n_n is additive noise that will be neglected for a moment. The signal equalized with one tap DFE is

$$z_n = y_n - b\hat{x}_{n-\Delta-1} \tag{3.2}$$

where b is the feedback coefficient, $\hat{x}_{n-\Delta-1}$ is the previously detected symbol and Δ is the channel delay. If there are no errors in reception (the channel is equalized), the detected symbols are equal to the received symbols, $\hat{x}_{n-\Delta-1} = x_{n-\Delta-1}$ and each DFE slicer output Q_m $m = \{0, 1, ..., 7\}$ sources $\hat{x}_{8n-m-\Delta-1} = x_{8n-m-\Delta-1}$.

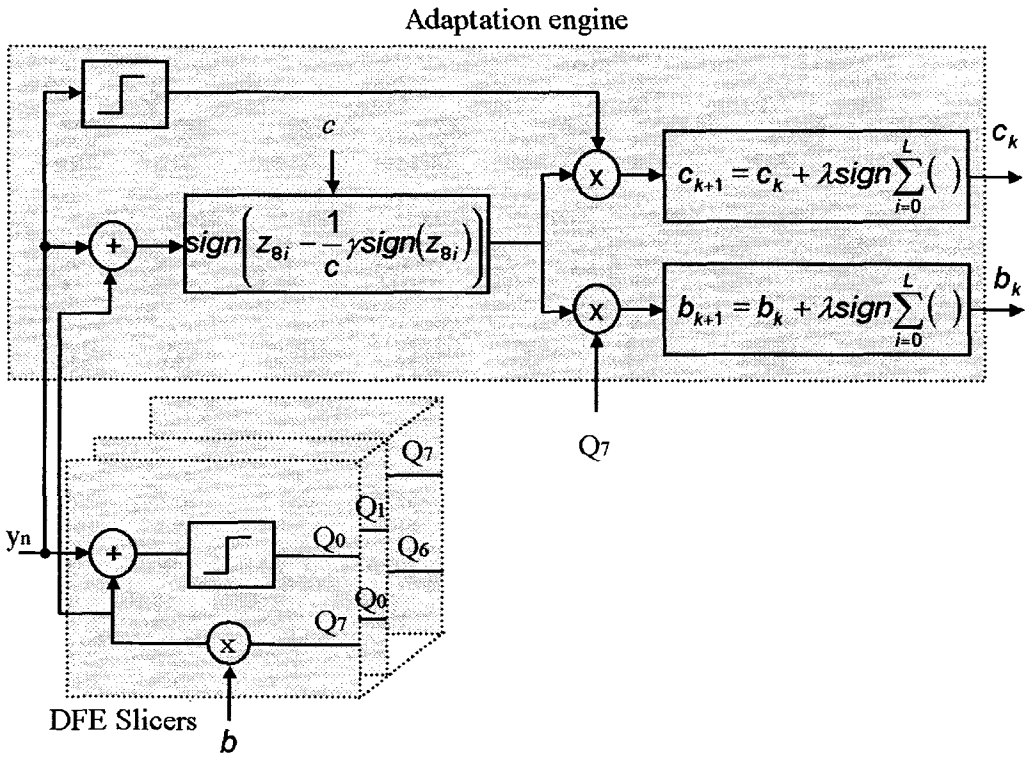

Figure 3.2: Block diagram showing DFE slicers and adaptation engine

If the known training signal where available at the receiver, the feedback equalizer would be initially fed with the actual transmitted symbols $x_{n-\Delta-1}$, until the equalizer coefficient convergence is achieved. At this point, the equalizer would switch to the decision directed mode where its feedback equalizer would be fed with previously detected symbols $\hat{x}_{n-\Delta-1}$. In this case, the equalizer coefficients are adjusted by minimizing the mean square error cost function (3.3) with an iterative procedure (3.4).

$$MSE = E\left\{(z_n - x_{n-\Delta})^2\right\} \tag{3.3}$$

$$b_{k+1} = b_k - \lambda \frac{d(MSE)}{db_k} \tag{3.4}$$

Because the known transmitted sequence is not available, we will minimize a cost function similar to the one introduced by Sato [17], mathematically analyzed by Benveniste et al. [18] and generalized by Godard [19], but with some modification to compensate for non-existence of feed-forward filter. Sato's algorithm is based on minimizing the cost function (3.5) where the constant γ is defined by (3.6) [19].

$$J = E\left\{(\gamma \cdot sign(z_n) - z_n)^2\right\} = E\left\{(|z_n| - \gamma)^2\right\} \tag{3.5}$$

$$\gamma = \frac{E[x_n^{\,2}]}{E[|x_n|]} \tag{3.6}$$

For binary (PAM-2) transmission, the constant γ (assuming that $x_n = \pm 1$), and the cost function (3.5) becomes identical to the cost function for decision directed mode. However, this cost function needs to be modified if the equalizer has only feedback filter because even if the feedback filter cancels all ISI, it cannot compensate for attenuation in the transmission line — attenuation of the cursor component of the channel impulse response. Hence, the equalized signal cannot be equal to the transmitted signal ($z_n = x_n$), but rather to an attenuated version of it ($z_n = x_n/c$), where $c \geq 1$. This attenuation can be compensated for by adjusting the transmitter gain, which requires a return channel from the receiver to the transmitter, or by having a programmable gain amplifier at the receiver, which is very difficult to implement at multi gigabit per second data rates.

Rather than adjusting the gain of the input signal, we adjust γ by replacing it with

$$\tilde{\gamma} = \frac{\gamma}{c} \tag{3.7}$$

Now the minimum of the Sato's cost function can be obtained by solving

$$\frac{dJ(b)}{db} = E\left\{-x_{n-\Delta-1} z_n |z_n|^{-1} (|z_n| - \tilde{\gamma})\right\} = 0 \tag{3.8}$$

If the ISI is fully canceled, then $z_n = x_{n-\Delta}/c$ and the expression in the parenthesis is equal to zero. This applies only to PAM-2. For PAM-4 and higher, the expression in parenthesis is not zero. However, equation (3.8) is still satisfied when ISI is fully canceled, because the transmitted samples are assumed to be independent and identically distributed (iid) random variables, so that $E[x_{n-1-\Delta} x_{n-\Delta}] = 0$. The feedback filter coefficients are calculated with an iterative procedure where the expectation term is dropped [19].

$$b_{k+1} = b_k - \lambda \hat{x}_{n-\Delta-1} z_n |z_n|^{-1} (|z_n| - \tilde{\gamma}) = b_k + \lambda x_{n-\Delta-1} (z_n - \tilde{\gamma} \cdot sign(z_n)) \tag{3.9}$$

Transmission line attenuation c, can be estimated with a similar iterative procedure

$$c_{k+1} = c_k + \lambda z_n (z_n - \tilde{\gamma} \cdot sign(z_n)) \tag{3.10}$$

where c is equivalent to coefficient of a single tap feed forward filter – single tap filter is essentially a programmable gain amplifier. Although we do not have an actual single tap filter (programmable gain amplifier), we calculate its coefficient and use it to adjust $\tilde{\gamma}$.

To simplify the implementation and increase the maximum speed of the circuit, which calculates the coefficient values (3.9) and (3.10), we use the sign-sign algorithm with gradient averaging.

$$b_{k+1} = b_k + \lambda \cdot sign\left[\sum_{i=0}^{L} sign(\hat{x}_{Ni-\Delta-1}) \cdot sign(z_{Ni} - \tilde{\gamma} \cdot sign(z_{Ni}))\right] \tag{3.11}$$

$$c_{k+1} = c_k + \lambda \cdot sign\left[\sum_{i=0}^{L} sign(\hat{x}_{Ni-\Delta}) \cdot sign(z_{Ni} - \tilde{\gamma} \cdot sign(z_{Ni}))\right] \quad (3.12)$$

As an be seen from (3.11) and (3.12), the gradient is averaged over L samples. The main reason for this is not only to increase the stability of the adaptation algorithm, but also to ease the design requirements of the coefficient generation circuit (D/A converter), which can run at a much slower speed than the symbol rate. The gradient averaging reduces the speed of convergence, but this is not an issue because the characteristics of the targeted transmission medium (backplane PCB) change very slowly over the time.

For instance, it takes about 50,000 received symbols or 12.5μs for DFE coefficient to converge from zero to the optimum value (see Figure 4.9 in Chapter 4), whereas changes in the transmission medium due to the ambient temperature changes and aging take minutes and days respectively.

It should be noted that for the coefficient calculation we do not have to take every consecutive received symbol ($N = 1$) as it is done in [2]. We can take spaced snapshots of the received symbol sequence because if x_n is an iid sequence, than x_{Nn} is also an iid, where N is the ratio of demultiplexing in the receiver front-end. By doing this we can significantly reduce the size and complexity of the receiver's adaptation circuit as well as the size of receiver's front end. Drawback of doing this is longer convergence time (by factor N), but this is not an issue for backplane applications as explained previously. The validity of this approach is shown in Figure 4.4 and Figure 4.9 in Chapter 4.

For instance, it is reported in [2] that 60% of power is consumed by adaptation circuit and that the receiver front-end uses 20 slicers out of 50 for the error detection. If this circuit were implemented as it is proposed here, the power consumption of the adaptation circuit and the number of slicers used for the error detection would be reduced by factor five, because the receiver in [2] uses 1-to-5 demultiplexing.

3.3.1 DFE Slicers

As can be seen in Figure 3.3, received 4Gbps serial data stream is de-multiplexed with eight identical blocks each clocked with a different phase ($\phi_0, \phi_1, ..., \phi_7$) of the 500MHz clock. Each block has two biased comparators, a multiplexer and a latch. Two comparators are used to perform look-ahead equalization before the previous detected symbol is known. One comparator is biased high to cancel inter-symbol interference if the previous detected symbol was low, and the other comparator is biased low to cancel inter-symbol interference if the previous detected symbol was high. The multiplexer selects the output of the one or the other comparator based on previously detected symbol. The latch stores received symbol Q_i and uses it to select the correct symbol Q_{i+1} in the next block. The level of biasing (equalization) is controlled with voltage b.

The maximum serial data rate, this circuit can accept is equal to the reciprocal value of:

$$t_{max} = t_{muxdel} + t_{latchdel} + t_{stlatch} \tag{3.13}$$

where t_{muxdel} is the maximum propagation delay of the multiplexer, $t_{latchdel}$ is the maximum propagation delay of the latch and $t_{stlatch}$ is the minimum setup time of the latch.

Block diagram in Figure 3.3 shows the single ended connections for simplicity. However, all circuits in the receiver front-end are differential.

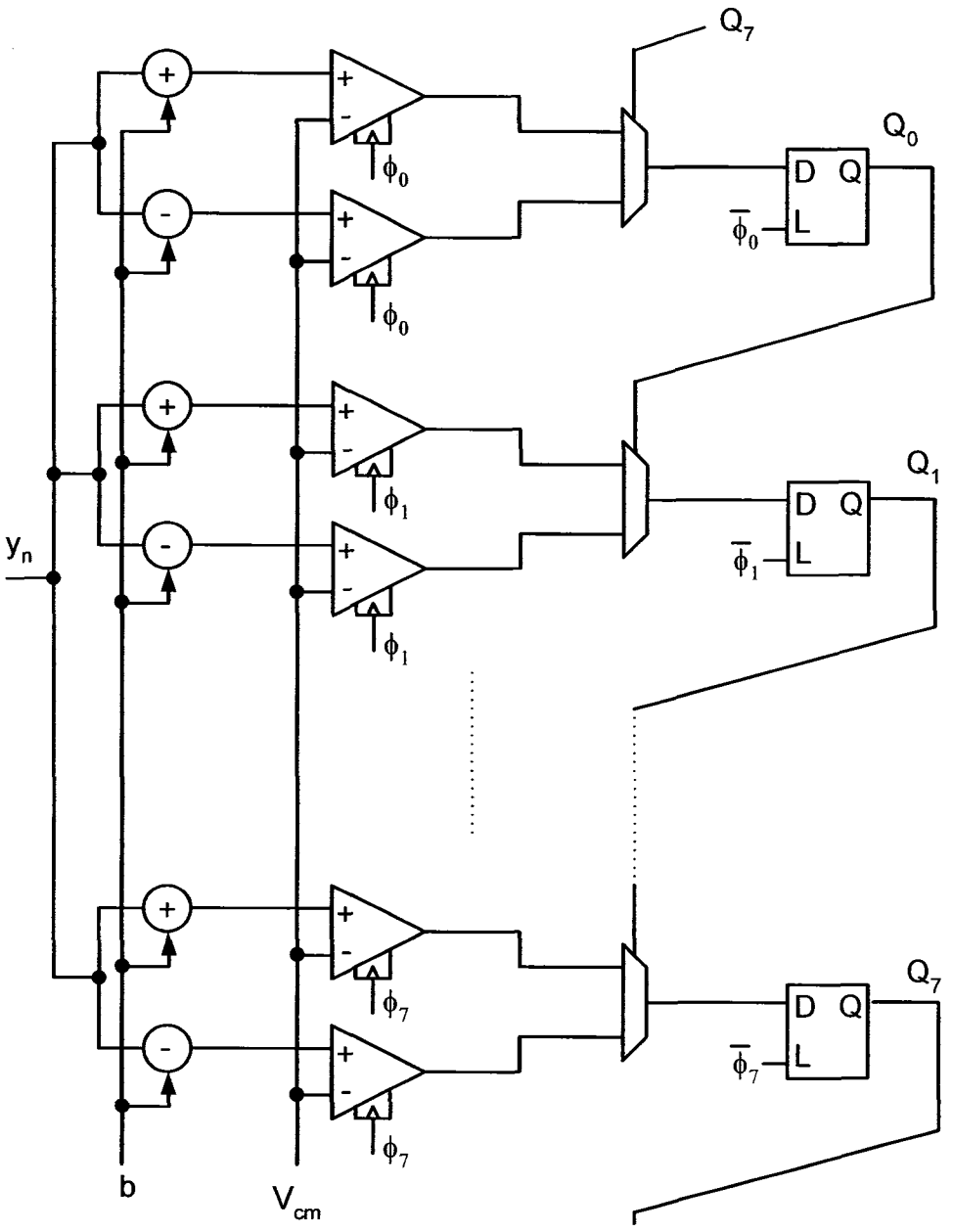

Figure 3.3: Block diagram of the receiver's demultiplexer

A transistor level schematic of the biased comparator is shown in Figure 3.4. The comparator is based on the high-speed latch used in Strong-Arm processor [15] with addition of the comparator bias transistors M10 and M11. When the clock is low, the output

nodes of the sense amplifier are pulled high with transistors M2 and M3. Transistors M5 and M8 are used to cancel out any voltage difference between the output nodes S and R and between nodes N1 and N2 clearing any memory from the previous latched value.

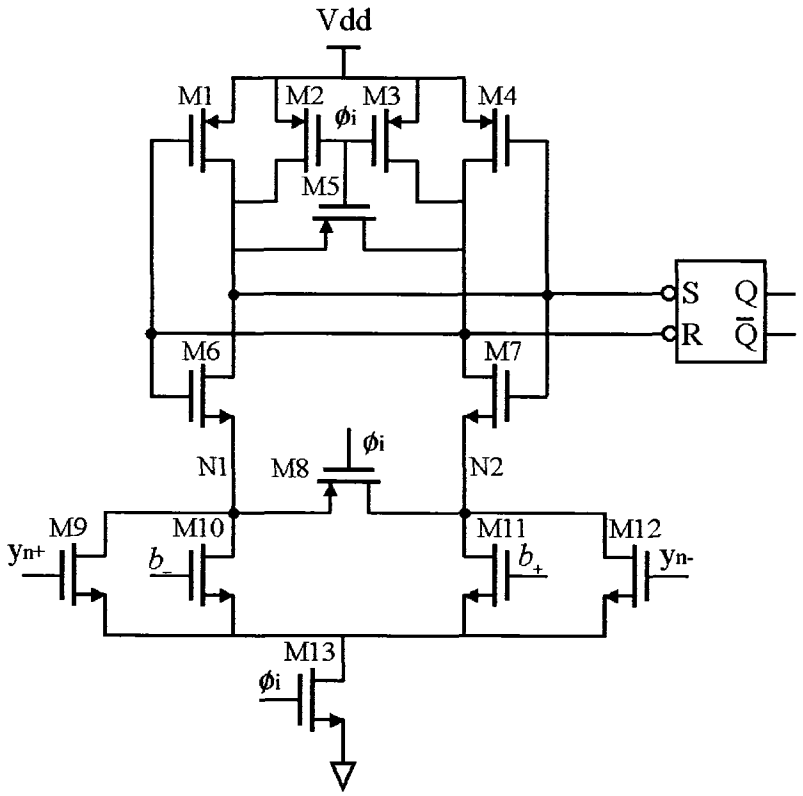

Figure 3.4: Biased comparator in DFE

During this time, the output of SR Latch does not change, holding the previous latched value. On the rising edge of the clock, transistors M2, M3, M5 and M8 are turned off, and transistor M13 is turned on, pulling down the nodes N1 and N2 via input transistors M9 and M12 and via DFE bias transistors M10 and M11. Depending on the input signal and DFE bias voltage at the rising edge, one of the nodes N1 or N2 will be pulled down faster. As soon as the voltage on one of the nodes goes below Vdd – Vt, let's assume it is N1, the transistor M6 above the node N1 will start conducting and this will trigger regenerative process in cross connected inverters (M1, M6) and (M4, M7). At the end of

regenerative process, the output of the inverter (M1, M6) will be pulled low and the output of (M4, M7) will be pulled high.

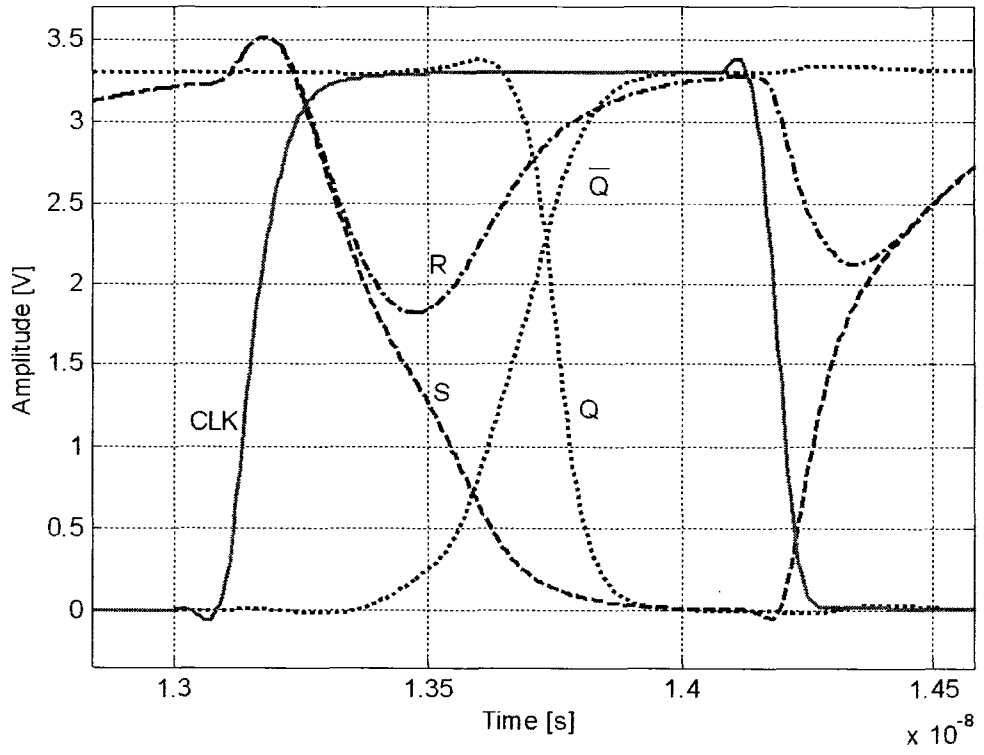

Figure 3.5: Biased Comparator Waveform

The high-speed latch reported in [15], has an always-on nmos transistor (gate tied to Vdd) between nodes N1 and N2. This transistor is used to prevent either node N1 or N2 from floating during to clock high cycle. However, this transistor is needed only if the change at the input voltage is such to turn-off transistors M9 or M11, which is typically case in digital circuits where the voltage at the input changes from rail-to-rail. In our case, transistors M9 and M11 are always on because the received voltage swing is only 500mV or lower.

It is very important for SR latch not to present data dependent load to the sense amplifier. The conventional SR latch (Figure 3.7) cannot be used in this application

because it presents data dependent load (capacitance) — its input capacitance is different for states 1 and 0. If used, it would give a hysteresis of 30 mVpp to the comparator as shown in Figure 3.6.

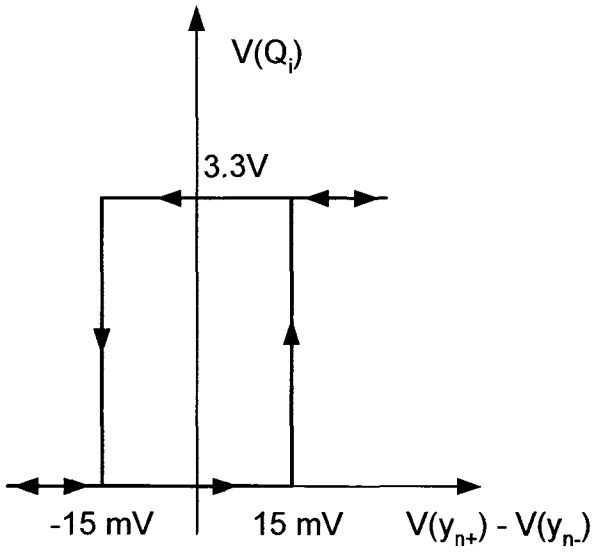

Figure 3.6: Hysteresis of the comparator with standard SR latch

Two different solutions have being used to overcome this problem. First one, described in [12] and [13], adds additional sense amplifier timed with inverted version of the clock, in order to increase the gain by cascading two sense amplifiers, which in turn reduces the hysteresis. Essentially, sense amplifier timed at the positive edge is followed by a sense amplifier timed at the negative edge, which is in turn followed by a standard SR latch.

The other solution, described in [14], adds inverters between sense amplifier and the SR latch made with two NOR gates. However, both of these solutions add extra logic which increases the power consumption and the area of the receiver. The second solution reduces the maximum speed of operation as well, because inverters add extra delay and NOR gate based SR latch is inherently slower than an SR Latch with NAND gates —

NOR gate has two PMOS transistors in series so the pull up time is slower than in the case of NAND gate based SR Latch.

In this thesis we use a simple modification of the standard SR latch to solve the comparator hysteresis problem. Let's analyze first the operation of a standard SR latch. In the standard SR latch, the S and R inputs are connected to NMOS transistors (transistors M3 and M7 in Figure 3.7) whose drains are tied to SR latch outputs Q and $\overline{Q}$.

First, assume that the output Q is high ($\overline{Q}$ is low) and that S and R are high (no change at the output nodes Q and $\overline{Q}$. Now if the R goes low, it will pull up node $\overline{Q}$ by turning on transistor M5 and by turning off transistor M7. $\overline{Q}$ high will pull down node Q by turning off transistor M2 and by turning on transistor M4.

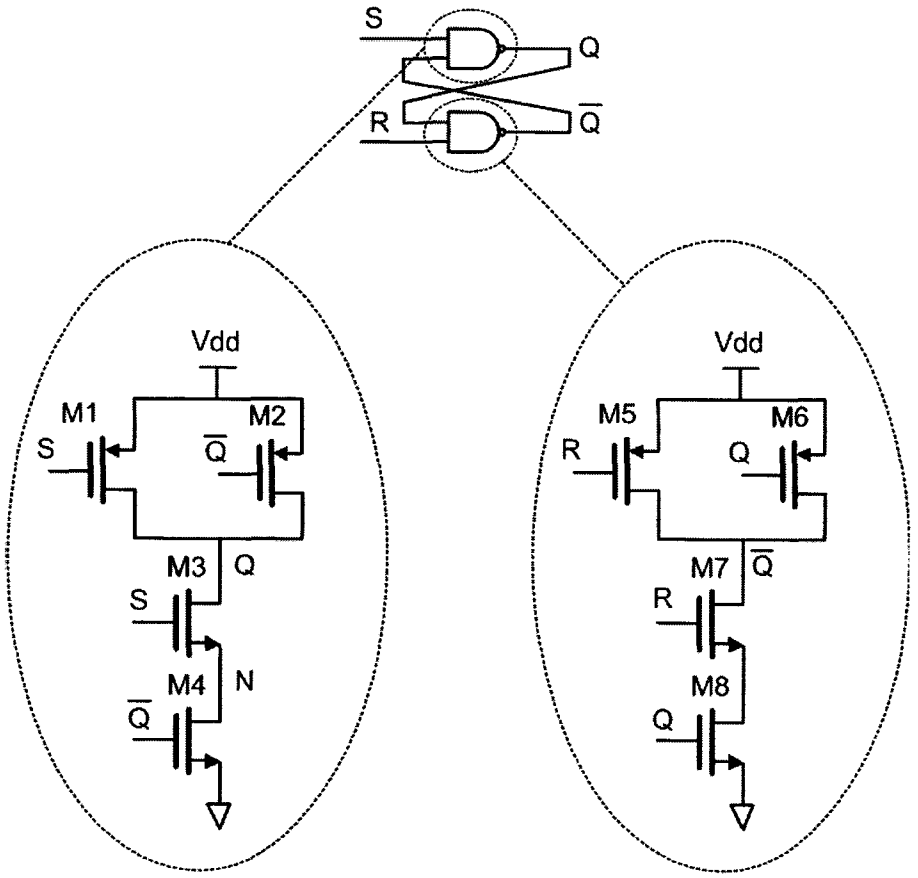

Figure 3.7: Standard SR latch

To understand why inputs S and R see the different capacitive load during the rearrangement, we should first look at the gate capacitance plots for NMOS and PMOS transistors shown in Figure 3.8 and Figure 3.9 respectively.

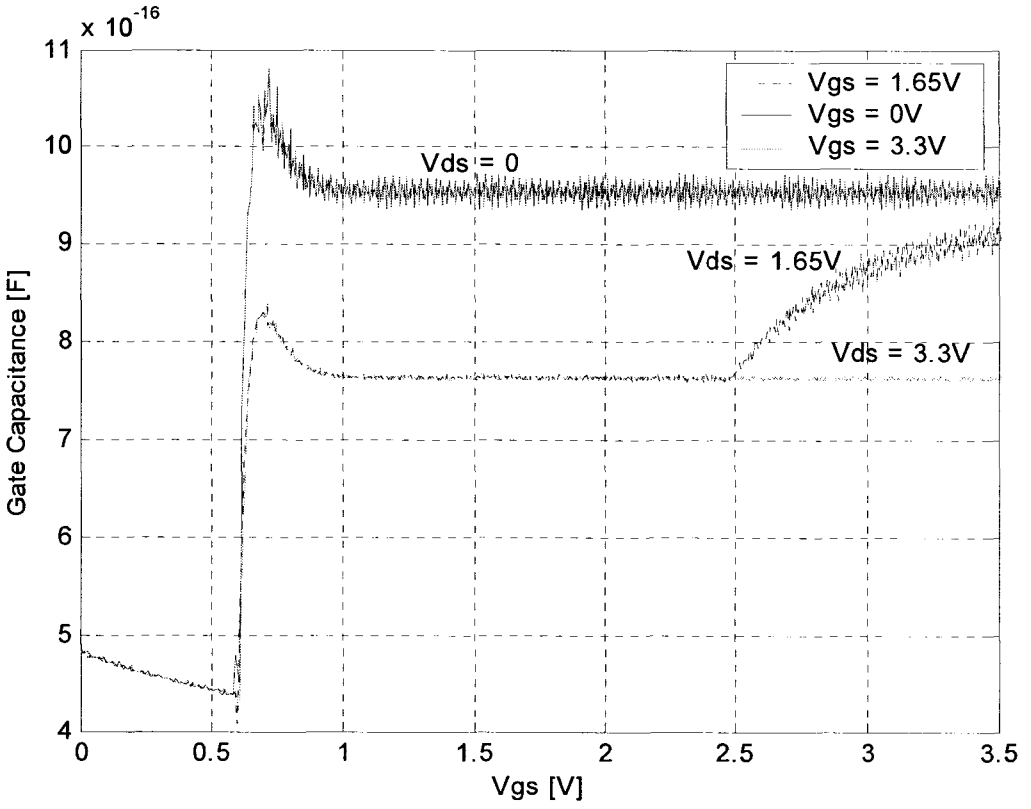

Figure 3.8: NMOS Gate Capacitance

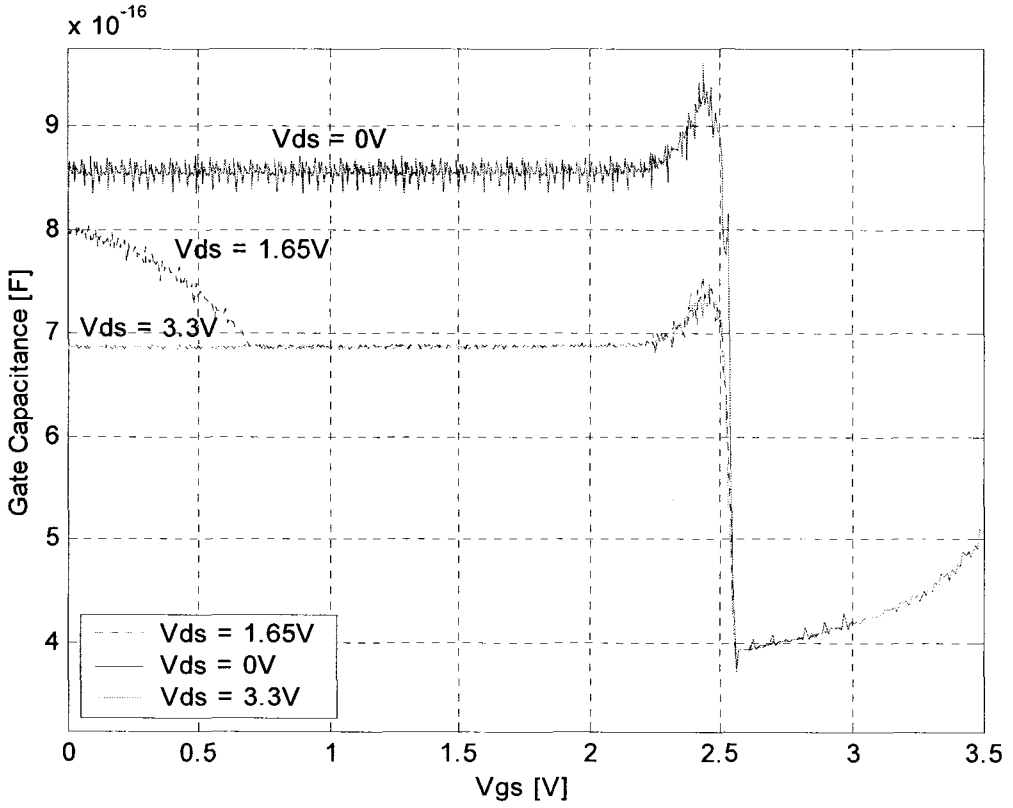

Figure 3.9: PMOS Gate Capacitance

From Figure 3.8 and Figure 3.9 we can see that when NMOS and PMOS transistors are off, their corresponding gate capacitances are about 50% lower, and their values do not depend on the voltage between the drain and source. Also, we see that NMOS transistor has about 10% higher gate capacitance under the same condition.

With these approximations in mind, we can now try to estimate the initial input capacitances of the standard SR latch.

Assume that initially Q is high ($\overline{Q}$ is low) and that S and R are high. Transistors M1 and M5 have the same gate capacitance because they both are turned off. Transistor M7 is turned on and its Vds = 0. However, transistor M3 is not really turned on, although its gate is tied high because its source is disconnected from the ground by transistor M4.

The source of the transistor M3 (node N in Figure 3.7) is at the voltage slightly above Vdd-Vtn, where Vtn is NMOS transistor threshold voltage. This keeps transistor M3 off in sub-threshold region.

Finally, we can estimate the initial input capacitances for S and R inputs of an SR latch.

Cs = Cpmos_off + Cnmos_off = 0.9 x Cnmos_off + Cnmos_off = 1.9 x Cnmos_off

Cr = Cpmos_off + Cnmos_on = 0.9 x Cnmos_off + 2 x Cnmos_off = 2.9 x Cnmos_off.

Based on this simple analysis, we see that when Q is high ($\overline{Q}$ is low), the R input has about 53% higher capacitance than the S input. Analogously, when Q is low ($\overline{Q}$ is high), the situation is reversed — capacitance of S input is about 53% higher than the capacitance of the R input.

This can be shown with HSPICE simulation by noting that for instance for S input

$$C_s(V_s) = \frac{dQ_s}{dV_s} = \frac{I_s dt}{dV_s} = \frac{I_s}{\frac{dV_s}{dt}} \tag{3.14}$$

and the capacitance equation for R input is analogous.

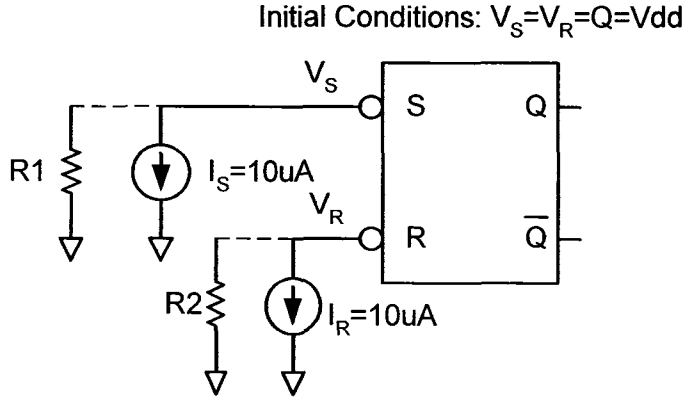

Figure 3.10: SR Latch input capacitance test bench

Figure 3.10 shows HSPICE test bench for simulating input capacitance of the standard SR latch with initial conditions Q is high ($\overline{Q}$ is low), and $V_S = V_R = V_{dd}$ (Similar test bench was used to obtain gate capacitances for NMOS and PMOS transistors shown in Figure 3.8 and Figure 3.9). Inputs S and R are initially charged to Vdd and then discharged either trough current sources or resistors as shown in Figure 3.10. Both methods give the same results for input voltages of interest $V_S, V_R > 1V$. However, as V_S and V_R approach zero value, their slopes becomes more flat and their derivative in (3.14), as well as the current trough resistors R1 and R2 drops to very low value causing errors in calculation due to finite precision. Hence, discharging them through the current sources is preferable if we are interested in capacitance for all values of the input voltages.

HSPICE uses quasi-static approximation for MOS charge and capacitance models, which are valid as long as the input signal rise or fall times is not shorter than

$$t_{r,f} > 20\tau_t \tag{3.15}$$

where $t_{r,f}$ is rise or fall time, and τ_t is the transit time for carriers that leave the source and arrive at the drain [28]. The transit time τ_t can be calculated with

$$\tau_t = \frac{L_{eff}}{v_{SAT}} \tag{3.16}$$

where L_{eff} is effective channel length and v_{SAT} is carrier saturation velocity.

For an NMOS transistor with 0.35 μm channel length and $v_{SAT} \approx 10^7$ cm/m, the transit time is τ_t = 3.5 ps. Hence, the transitory input rise or fall time $t_{r,f}$ needs to be longer than 70 ps [28] for quasi-static approximation to be valid. Hence, the current sunk by the current source (or resistor R1 and R2 values) should be selected accordingly, so that the rise and fall times are not shorter than calculated.

The HSPICE simulation results of a standard SR latch input capacitances with respect to the input voltage are presented in Figure 3.11.

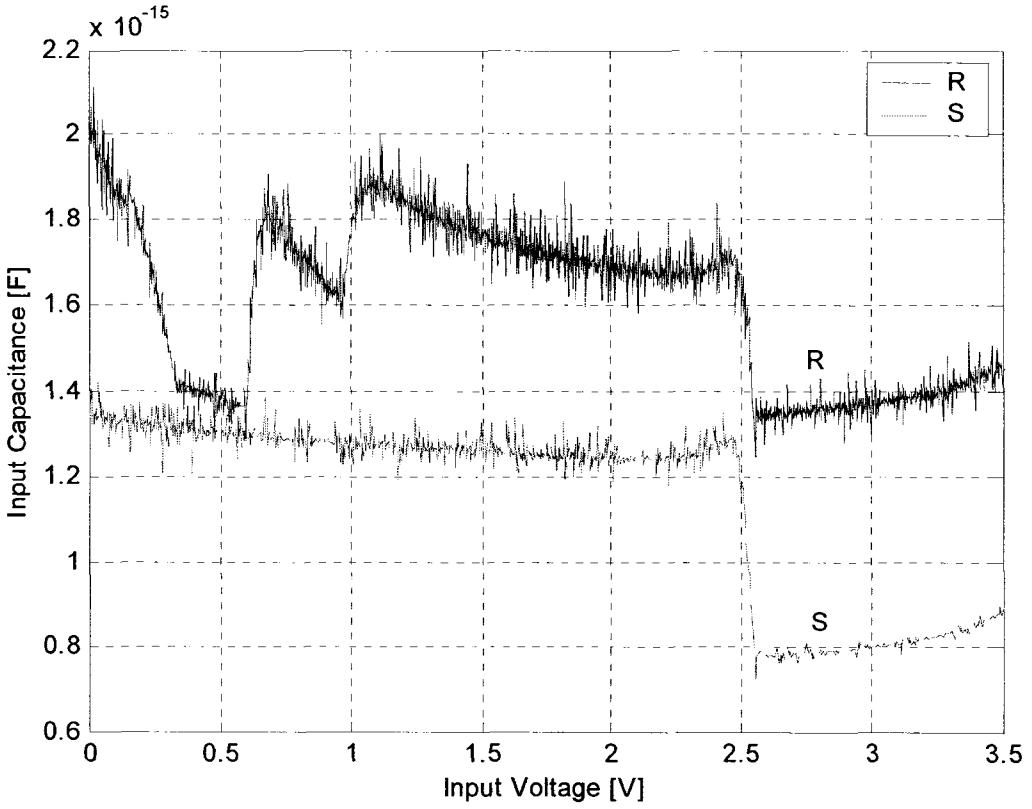

Figure 3.11: SR Latch input capacitance when Q is high ($\overline{Q}$ is low)

Simulation result shows that initial capacitance (for Vdd = 3.3V) differs between S and R inputs by 63%, which is relatively close to the previous, by hand estimated difference of 53%. The major discrepancy between HSPICE and the hand calculation is caused by the fact that we assumed that input capacitance is constant when transistor is turned off $0 < |V_{GS}| < V_t$. However, a MOS transistor capacitance decreases as V_{GS} goes from 0 to V_t because thickness of depletion region increases as V_{GS} approach V_t. In vicinity of V_t it is about 20% lower than for $V_{GS} = 0$ as can be seen in Figure 3.8. This means that the input capacitance of transitory M3 was overestimated in hand calculation.

Now, we will describe how with very simple modification of the standard SR latch, we can get a latch that does not present data dependent load to the high speed comparator.

If the inputs R and Q in Figure 3.7, connected to M3 and M4 respectively, are swapped as well as the inputs S and $\overline{Q}$ connected to transistors M7 and M8, we get a modified SR latch shown in Figure 3.12. Now, both S and R inputs see the same initial capacitance because n-transistors M4 and M8 are turned on (S and R are high) and their $V_{DS} = 0$, and because p-transistors M1 and M5 are turned off (their V_{DS} is not equal, but their input capacitance is equal because both of them are off).

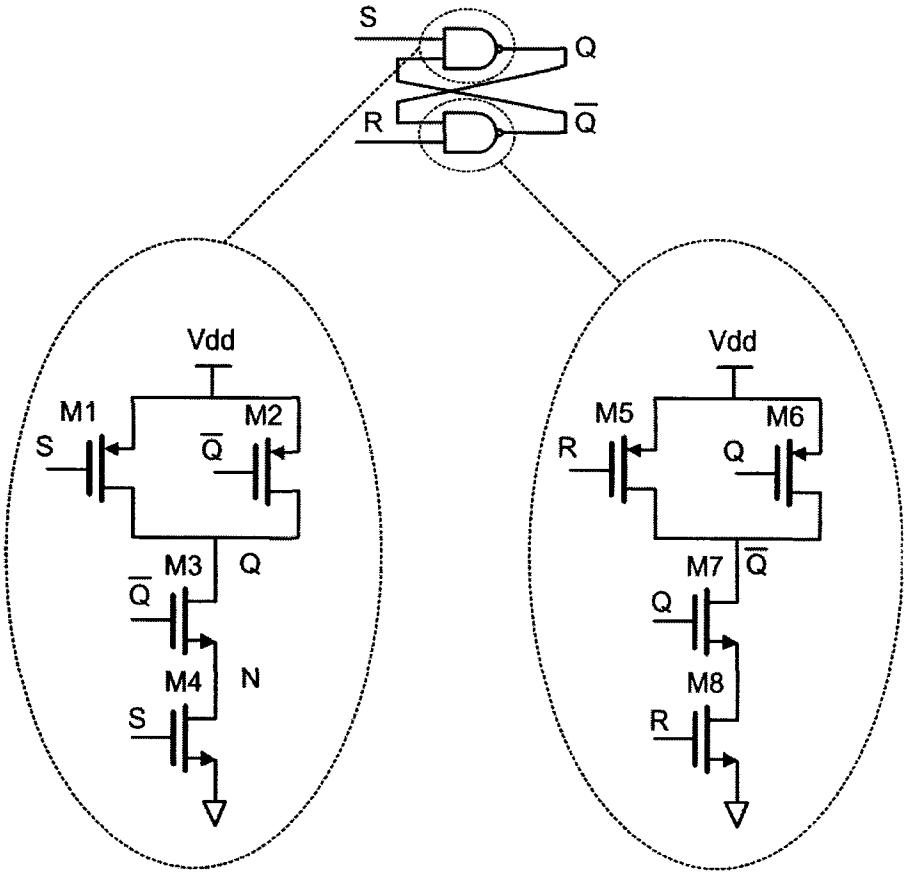

Figure 3.12: Modified SR Latch

The input capacitance of the modified SR Latch is shown in Figure 3.13. Initially, when both S and R are high, the capacitance is equal. The input capacitance will stay equal as long as p-transistors M1 and M5 are off $Vdd - V_{tp} < V_S, V_R < Vdd$.

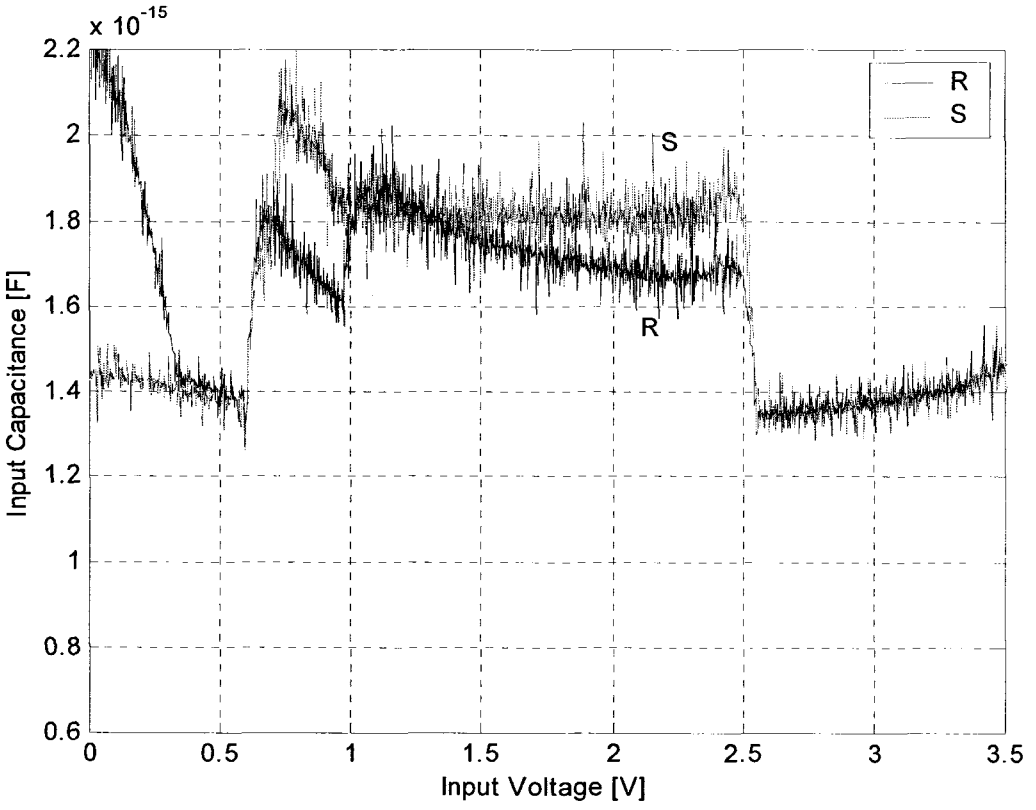

Figure 3.13: Modified SR Latch input capacitance

When transistors M1 and M5 start to conduct $V_S, V_R > V_{dd} - V_{tp}$, their gate capacitances will start to differ because their corresponding V_{DS} will be different. This difference is not important in our application because during the evaluation phase (clock goes high) of the sense amplifier (Figure 3.4), the S and R inputs will sag by typically less than V_{tp} before the high-speed comparator makes decision — either node S or R will go low and the other will pull back to the high voltage level as shown in Figure 3.5.

However, when the input differential voltage is very low (less than 2 mVpp), the sense amplifier will sag S and R by more than V_{tp} before it makes decision. For such low input values, the difference in the SR Latch input capacitance will be dominant in the decision process. We should point out that the polarity of the difference between S and R input capacitances for $V_S, V_R < V_{dd} - V_{tp}$ is different between the standard and modified SR Latch. While the R input on the standard SR Latch has higher input capacitance, the S input on the modified SR Latch has higher input capacitance (For the case when Q is high ($\overline{Q}$ is low) and $V_S, V_R < V_{dd} - V_{tp}$). The consequence of this is that while the standard SR latch has input hysteresis of 30mVpp, the modified latch has negative hysteresis of 2mVpp. So if the differential input stays in the 2mVpp range the output will toggle. This is not concern in our application because the sensitivity of the input comparator is not expected to be lower than 2mVpp.

It should be noted that pulling both S and R inputs all the way to the ground forces the SR latch to an undefined state. In the case of the standard (Figure 3.7) and modified (Figure 3.12) SR latch both outputs Q and $\overline{Q}$ would go high. Hence, the input capacitance plots below the SR Latch input threshold can be ignored because an SR Latch would never go to an undefined state under normal conditions. However, the input capacitance for standard and modified SR Latch over all input voltages is shown in Figure 3.11 and Figure 3.13 just for completeness.

3.3.2 Multiplexer and latch

As explained in previous section, each DFE block contains a multiplexer and latch. They have to be designed to with minimum possible delay because the maximum serial data rate, this circuit can accept is equal to the reciprocal value of

$$t_{max} = t_{muxdel} + t_{latchdel} + t_{stlatch} \tag{3.17}$$

where t_{muxdel} is the maximum propagation delay of the multiplexer, $t_{latchdel}$ is the maximum propagation delay of the latch and $t_{stlatch}$ is the minimum setup time of the latch. The maximum simulated data rate for 0.35μm TSMC technology is 4.5Gbps.

The schematics of the multiplexer and latch are shown in Figure 3.14. The multiplexer is realized with pass-gate logic for minimum delay. The latch is implemented with two cross-coupled inverters and four extra transistors used to toggle the state of the latch. It is very similar to implementation of an SRAM cell. Transistor sizing is very important for proper operation of the latch.

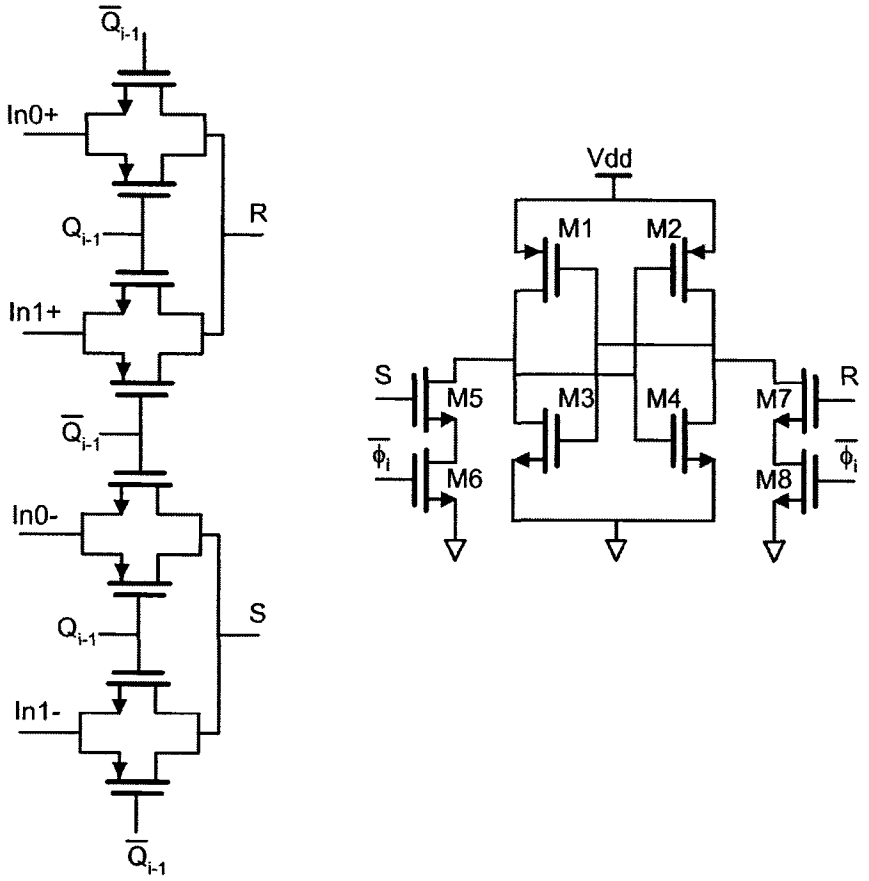

Figure 3.14: Implementation of the multiplexer and latch

3.3.3 Adaptation Engine

Figure 3.15 shows detailed block diagram of the circuit that calculates error $e_{Ni} = sign(z_{Ni} - \tilde{\gamma} \cdot sign(z_{Ni}))$. The input signal is simultaneously equalized with one tap look-ahead DFE and compared with either positive $V_{cm} + \tilde{\gamma}$ or negative $V_{cm} - \tilde{\gamma}$ limit, depending on the received symbol being either positive or negative respectively.

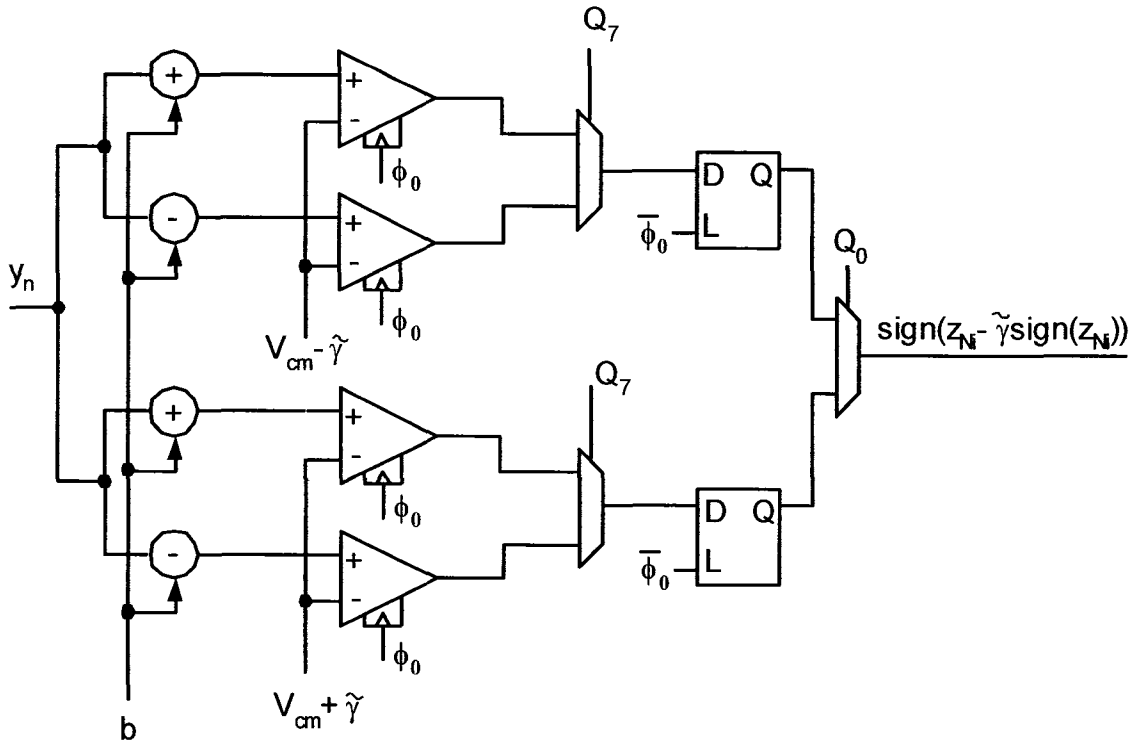

Figure 3.15: Block diagram of the error calculation circuit

The gradient is generated by multiplying (with and XOR gate) the error e_{Ni} with the previous received symbol (Q_{n-1}), and by integrating the resulting value with an Up/ Down counter, as shown in Figure 3.16. The sign (most significant bit) of the counter CNT-1 is used to increment/decrement the counter CNT-2, once every L clock periods. The value stored in CNT-2 is used to generate the bias voltage for DFE slicers via the low speed 7-bit D/A converter.

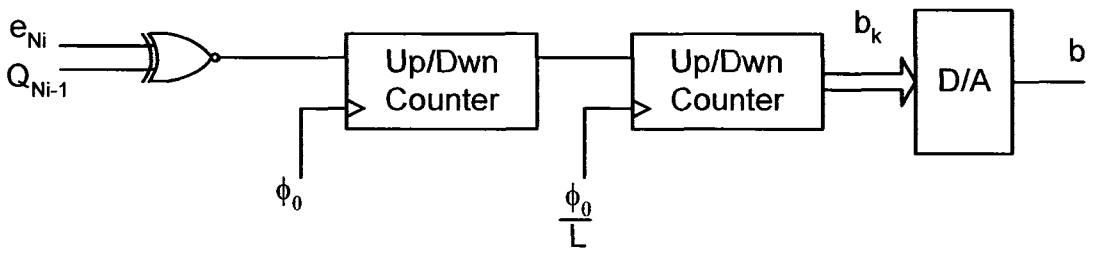

Figure 3.16: Block diagram of DFE coefficient calculation circuit

A similar circuit is used to calculate $\tilde{\gamma} = 1/c$. By noting that for practical applications, $1 \le c \le 4$ we can approximate $\tilde{\gamma} = 1/c$ with a linear function by replacing it with a first degree Taylor polynomial around $c = 2.0$, which turns out to be $\tilde{\gamma} = 1 - 0.25c$.

As mentioned before, the error calculation is not performed on every consecutive received symbol as in [2], but on every eighth received symbol. This reduces design requirements on adaptation circuit and significantly reduces the power consumption.

3.4 Re-synchronizing received data to a single clock phase

Section 3.3.1 showed how demultiplexing of the received serial stream was done with eight slicers, each timed with a different phase of the recovered clock. To simplify the design and implementation of the remaining circuits, demultiplexed data needs to be re-synchronized to a single phase of the internal clock as shown in Figure 3.17. This was achieved by first sampling the output of eight transparent latches in Figure 3.3, with eight flip-flops clocked by eight phases of the internal clock. Next, the outputs of these eight flip-flops are split into two groups: one sampled with ϕ_0 and the other with ϕ_4. Each group is stored in corresponding four bit register. Finally, the outputs of two four-bit registers are re-synchronized to only one clock phase ϕ_0.

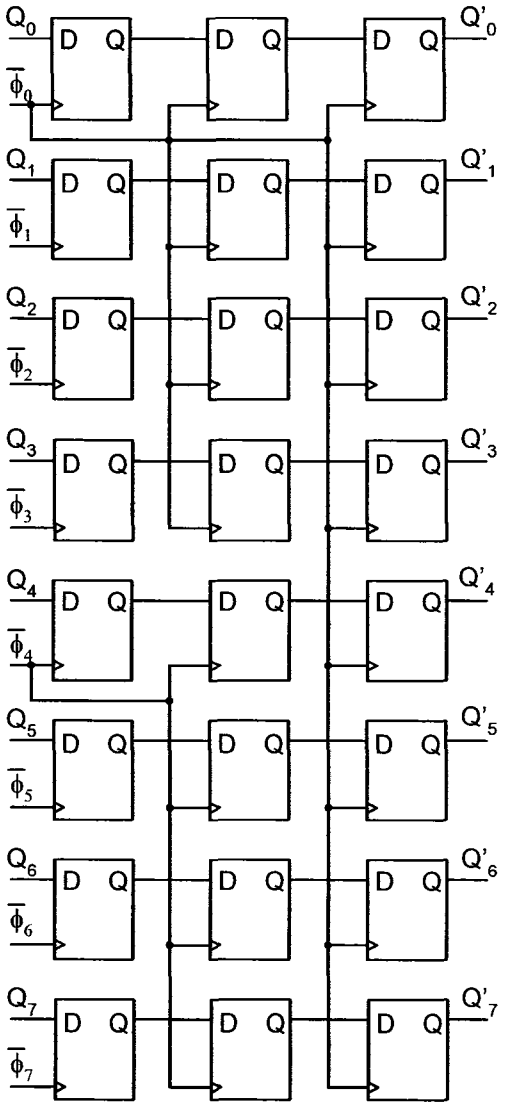

Figure 3.17: Re-synchronizing to a single clock phase

3.5 Clock Data Recovery

Previous sections described how the received data is sampled and equalized, assuming that the clock aligned to bit edges is available all the time. In this section, we describe the clock recovery method implemented in this thesis.

Backplane serial transceivers generally do not transmit high speed clock because this would require extra pair of traces for each serial link. In addition, it would be difficult

to guarantee that the data and the clock have the same delay and attenuation. Because of this, backplane serial transceivers extract the clock from the received data stream. Depending on the synchronization between the transmit and receive device we can categorize the clock recovery in two different methods.

- Phase only recovery — when the transit and receive device are synchronous. In many telecom/datacom systems all cards connected to the backplane are synchronous to same clock usually generated from the dedicated timing card in the system. Having all devices in the system synchronized to the same clock greatly simplified the design (it is much easier to design with single than with multiple clock domains) and improves performance (delay trough the system gets reduces because less or no buffering is needed).
- Phase and frequency recovery — when the transmit and receive devices are plesio-synchronous. This means that the transmit and receive device are not driven by the same clock, but with two different clocks whose frequency difference is very small. For instance, if the transmit device is timed with a crystal oscillator XO1 whose accuracy +/-25 ppm (parts per million) and the receive device is timed with XO2 whose accuracy is +/-30 ppm then we can say that the transmit and receive device are plesio-synchronous. This method is more difficult to implement, but its main advantage is that it does not need jitter attenuators, because a clock generated with XO placed very close to the device has less jitter than the clock which is distributed throughout the system. The most frequently used method for clock recovery in backplane transceivers is one described in [16].

Although the second method is more powerful because it works well with synchronized as well as with plesio-synchronous system, we used first method in this thesis because it is simple to implement and because the clock recovery was not the main focus of this thesis.

In order to simplify diagram in the Figure 3.3, we have shown how the demultiplexing is done in the receiver without showing the clock recovery circuit. However, the

clock recovery is integral part of the receiver and its block diagram is shown in Figure 3.18. The clock recovery circuits is comprised of clock recovery (CR) slicers, phase detector, integrator and two DLLs. As mentioned in Section 3.2, all clock recovery circuits are implemented behaviorally except the clock recovery slicers which are implemented at the transistor level. The input serial stream is oversampled by two, where DFE slicers sample data at the middle of the bit cell and the clock recovery slicers sample the data at the bit cell transitions as shown in Figure 3.18. The major difference between DFE slicers and the clock recovery slicers is that each clock recovery slicer contains only one non-biased comparator whereas each DFE slicers have two contra biased comparators. DFE and clock recovery slices are timed with 16 equally spaced phases of a 500MHz input clock generated with DLL2. DLL1 is used to shift these clocks left or right so that DFE slicers and clock recovery slicers sample input stream at proper instances — DFE slicers at the middle, and the clock recovery slicers at the transition of bit cells. Delay of the DLL1 is adjusted based on data sampled with DFE and clock recovery slicers as shown in Figure 3.19 and Table 3.1.

TABLE 3.1: Generation of Early/Late signals

Sampled Data (Q_0 E_0 Q_1)	Sampling Point
110 or 001	Early
100 or 011	Late
The other values	do not change

DLL1 block contains only a charge pump and a voltage controlled delay line, while its phase detector is actually formed with DFE and CR slicers, Early/Late decoders, and the Majority vote circuits. Detailed schematics of the DLL1 block, implemented with behavioral HSPICE, is shown in Figure 3.20. The delay of the delay line is adjusted by varying the capacitive load (voltage controlled capacitances) seeing by delay line inverters. DLL2 is implemented similarly with addition of a standard phase detector.

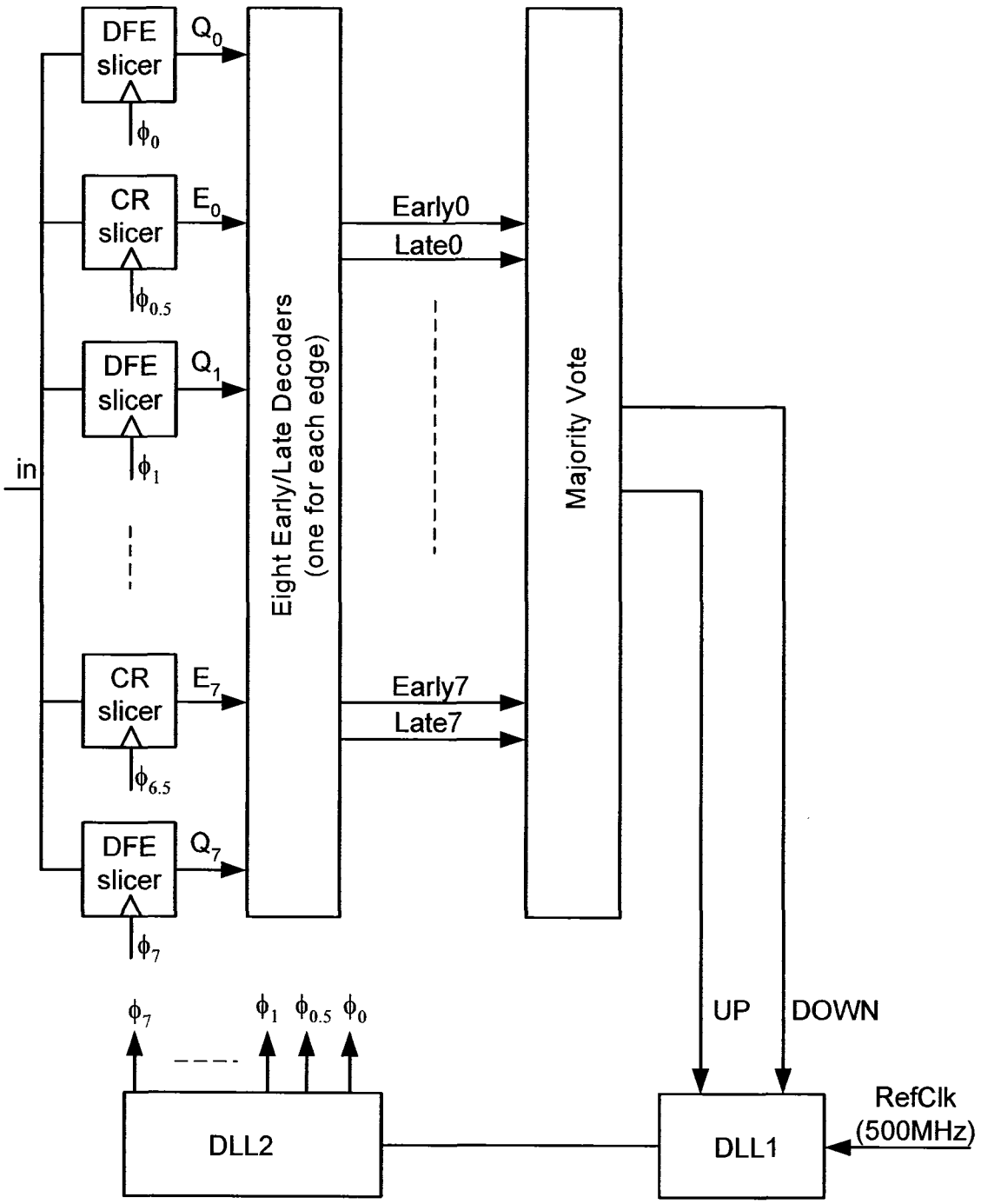

Figure 3.18: Block Diagram of Clock Recovery Circuit

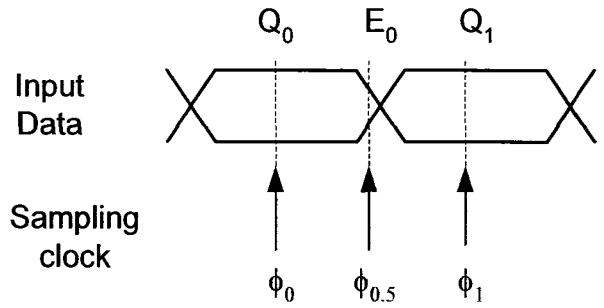

Figure 3.19: Sampling Clocks

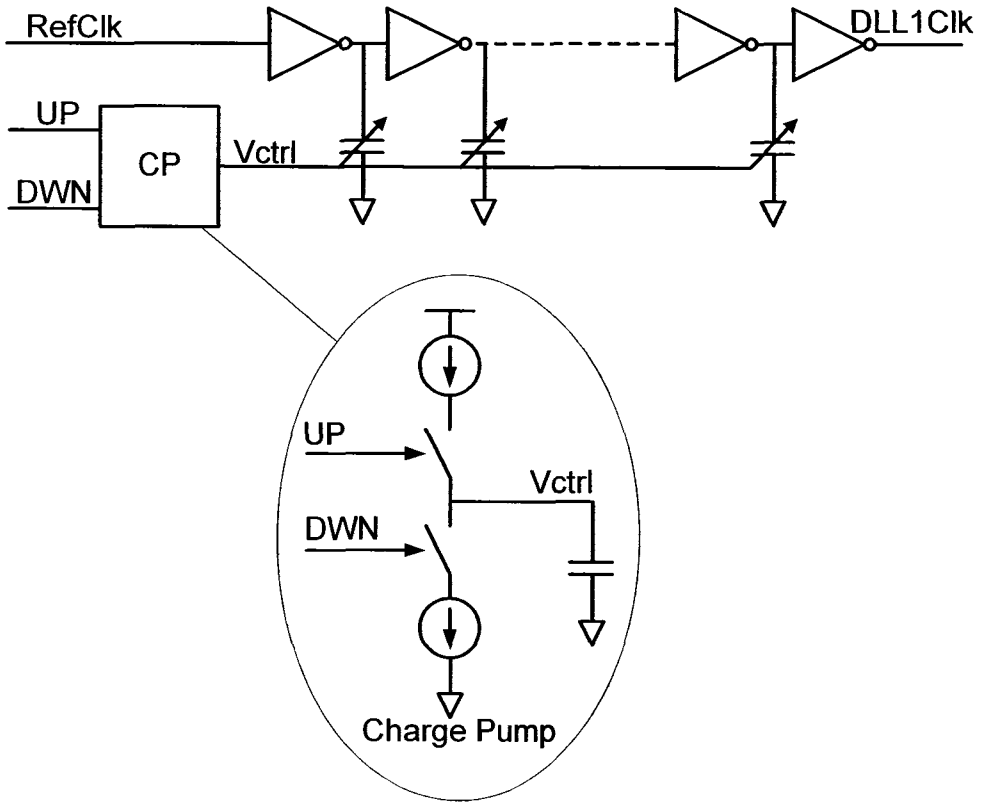

Figure 3.20: Schematic of DLL1

Simulated clock adjustment is shown in Figure 3.21. Initially, the internal clocks are not phase aligned with incoming serial stream. Once the DLL1 gets adjusted, we can see that the peak-to-peak value of the residual jitter is 18 ps.

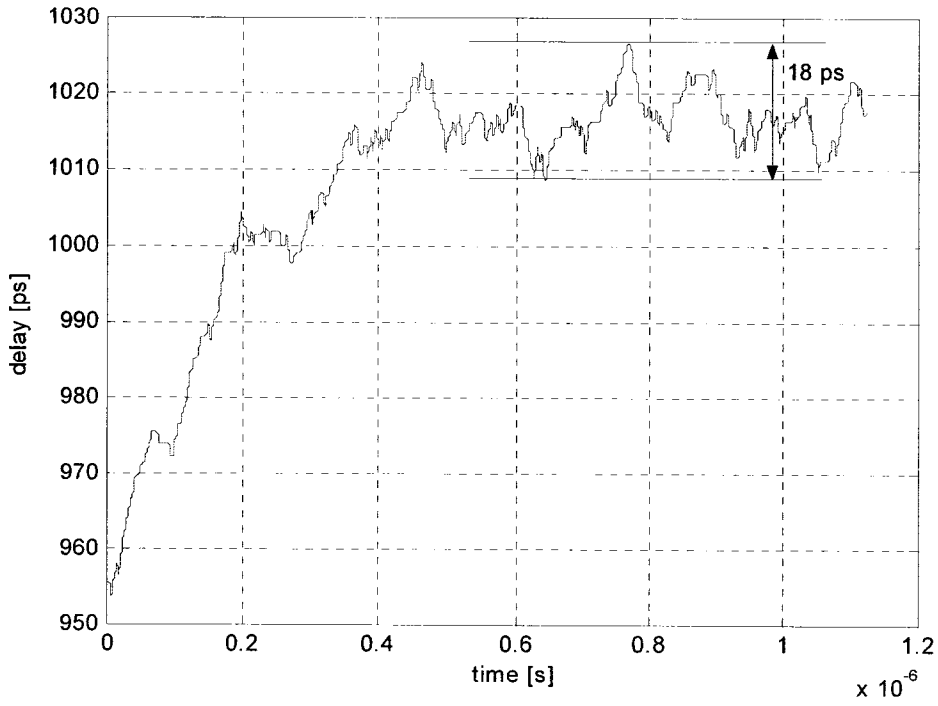

Figure 3.21: DLL locking and residual jitter

CHAPTER 4

Simulation Results

4.1 Introduction

The previous chapter provided architectural and implementation details of the receiver with simulation results of some critical blocks. In this chapter we present simulation results for complete receiver. We start with the transmission line model used in this thesis, which besides PCB traces includes the IC packaging and backplane connectors models. Next, we show effectiveness of the receiver by comparing the eye-diagram with and without DFE engaged. Because DFE is accomplished by biasing comparators, the eye-diagrams can only be obtained from behavioral simulation. In order to confirm that the circuit simulation results closely match the behavioral simulation, we conclude this chapter by comparing the bit-error rate of the behavioral and circuit simulation during the convergence process.

4.2 Transmission line simulation model

The simulation test bench, shown in Figure 4.1, include 1m long trace on a FR-4 backplane, two 0.1m traces on the FR-4 transmit and receiver boards, two backplane connectors and Tx and Rx device package models. The width of all traces is 0.2mm and the thickness is 18 μm. A low cost Ball Grid Array (BGA) package model have been used with the worst case parameters 2 pF ball capacitance and 5 μH bonding wire inductance.

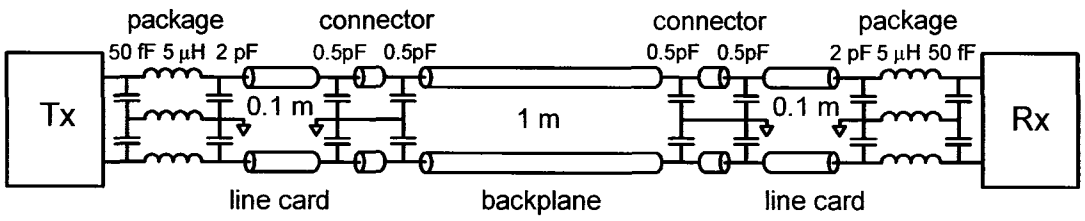

Figure 4.1: Transmission line simulation model

Frequency response of the simulation model is shown in Figure 4.2. If we compare the frequency response of the simulation model, with the frequency response of 1m long FR-4 PCB (Chapter 2, Figure 2.14) we can see that the attenuation of the simulation model is significantly more severe, which is primarily influenced by IC packaging and backplane connector models, but also by the fact that the total PCB trace length in the simulation model used in this thesis is 1.2 m (1 m backplane and 2 x 0.1 m line card traces).

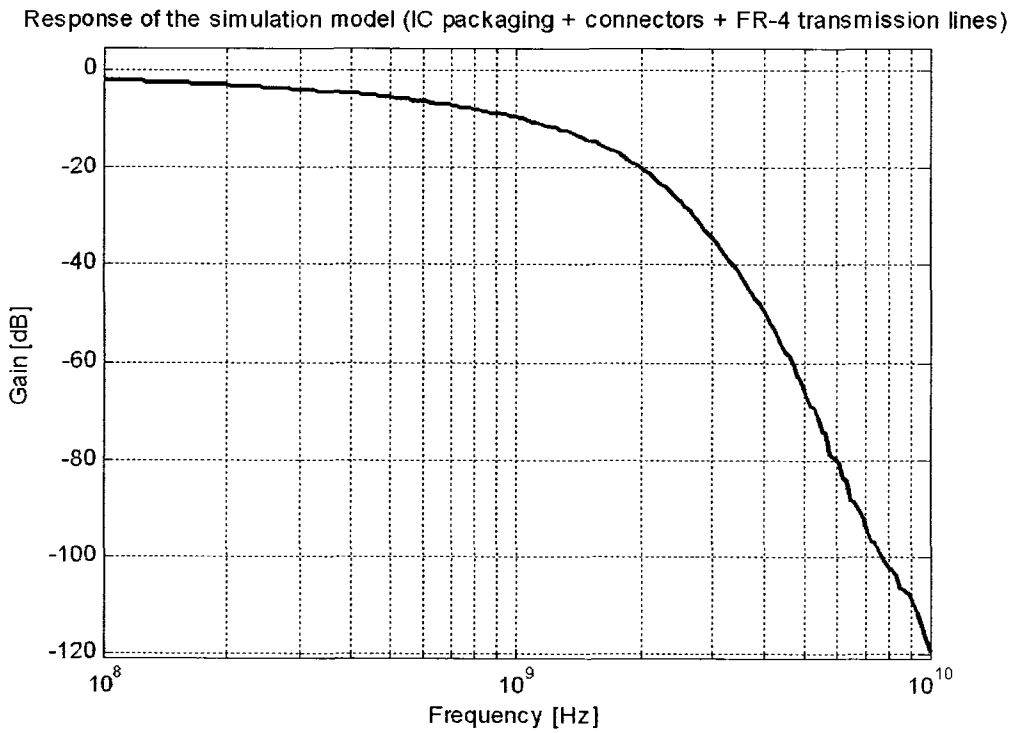

Figure 4.2: Frequency response of the simulation model

Transmission lines are modeled and simulated using HSPICE's W element. Figure 4.3 shows cross section of the transmission line whose parameters (R, L, G, C) required by HSPICE simulator have been calculated with 2-D electromagnetic simulator Linpar from [25].

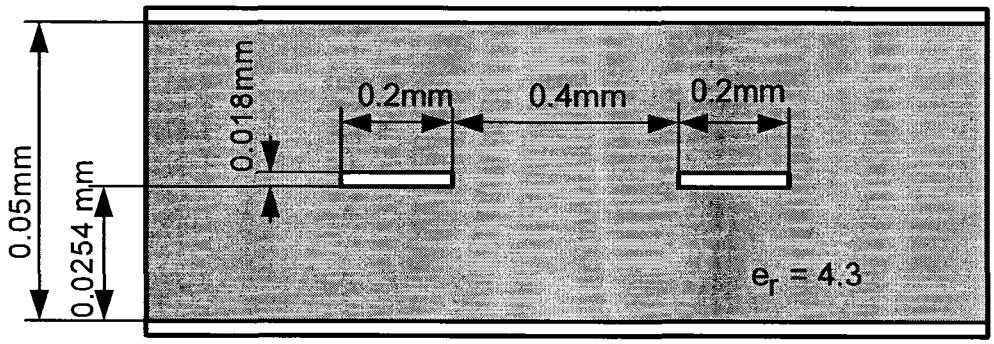

Figure 4.3: Cross section of transmission line used in the simulation

For the differential stripline shown in Figure 4.3, assuming that dielectric is FR4 (ε_r=4.3), copper conductance $\sigma = 5.98 \cdot 10^7 \left[\frac{S}{m}\right]$ and dielectric loss $\tan\delta = 0.025$, Linpar generates following RLGC matrices.

$$L = \begin{bmatrix} 3.484e-07 & 1.118e-08 \\ 1.118e-08 & 3.484e-07 \end{bmatrix} \left[\frac{H}{m}\right]$$

$$C = \begin{bmatrix} 1.374e-10 & -4.411e-12 \\ -4.411e-12 & 1.374e-10 \end{bmatrix} \left[\frac{F}{m}\right]$$

$$R = \begin{bmatrix} 2.891e+01 & 7.229e-01 \\ 7.229e-01 & 2.891e+01 \end{bmatrix} \left[\frac{\Omega}{m}\right] \qquad (4.1)$$

$$G = \begin{bmatrix} 2.158e-02 & -6.929e-04 \\ -6.929e-04 & 2.158e-02 \end{bmatrix} \left[\frac{S}{m}\right]$$

While L and C matrices can be loaded directly into HSPICE (C and L are not frequency dependent), R and G need to be split into DC and frequency dependent AC components

$$R = R_{DC} + \frac{R_{AC}}{\sqrt{f}} \tag{4.2}$$

$$G = G_{DC} + \frac{G_{AC}}{f} \tag{4.3}$$

Frequency dependent components R_{AC} and G_{AC} can be easily derived from (4.2) and (4.3) since we have R and G from Linpar, $G_{DC} = 0$, and for copper traces

$$R_{DC} = \frac{1}{\sigma w t} \tag{4.4}$$

where w is the width, and t is the thickens of the copper trace.

After simple calculation we get the matrices suitable for HSPICE W model. Following attachment contains corresponding HSPICE W model for stripline shown in Figure 4.3.

```
* Created January 14, 04 using Linpar
* based on differential stripline w = 0.2 mm, t = 0.018 mm, h = 0.0254 mm
* H = 0.05 mm, s = 0.4 mm, er = 4.3 (FR-4) sigma = 5.98e07 siemens/m,
* tan(d)= 0.025

* N = number of lines
*********
2

* Lo = inductance matrix
***************
348e-9
11.18e-9    348e-9

* Co = Capacitance matrix
**************
137.4e-12
-4.411e-12  137.4e-12

* Ro = Resistance matrix
```

```
**************
4.628
0      4.628

* Go = conductance matrix
*******************
0
0      0

* Rs = skin effect matrix
***************
0.000914
2.28e-5      0.000914

* Gd = dielectric loss matrix
***************
21.58e-12
-69.29e-14   21.58e-12
```

4.3 Simulation Results

Figure 4.4 shows that the presented blind equalization algorithm converges to the same coefficient values as the equalization with the known sequence for different lengths of the backplane. It is interesting to note that DFE coefficient is not monotonically increasing with the line length. This is because the absolute value of the postcursor ISI component, canceled by DFE, is affected not only by the ISI, but also by the line attenuation. However, the relative value of the postcursor component (h_1/h_0) does monotonically increase with the line length.

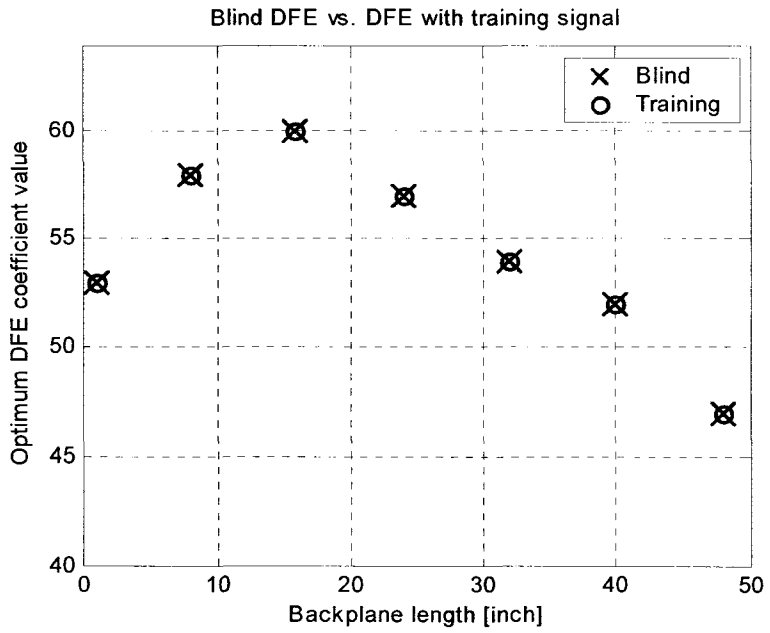

Figure 4.4: Optimum DFE coefficient for blind DFE and DFE with training signal

An eye-diagram is usually used to show performance of an equalizer. In our case, an eye-diagram can be generated only with behavioral simulation because in the actual circuit, the equalization is achieved by biasing high-speed comparator.

Eye-diagram at the input of receiver is shown in Figure 4.5, and the received symbol rate eye-diagram without equalization is shown in Figure 4.6. Figure 4.7 illustrates equalized symbol rate eye-diagram after DFE converges.

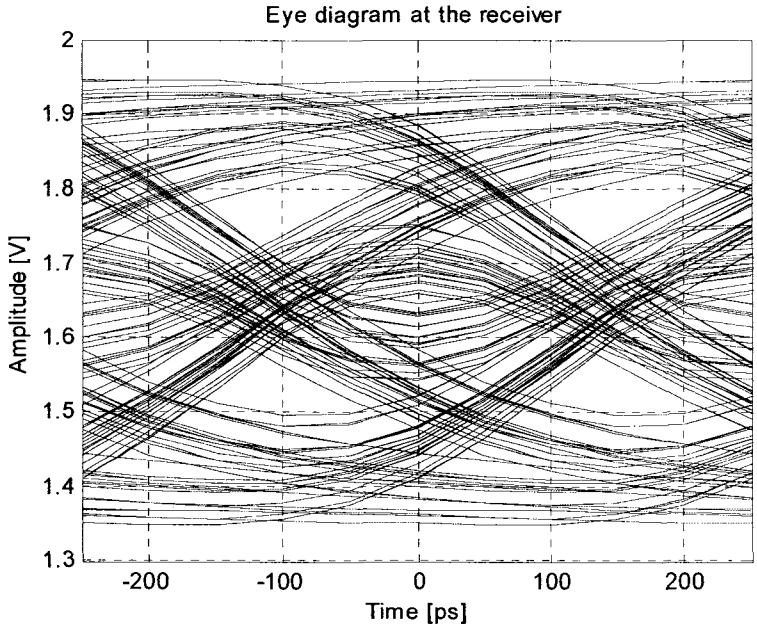

Figure 4.5: Eye-diagram at the input of the receiver

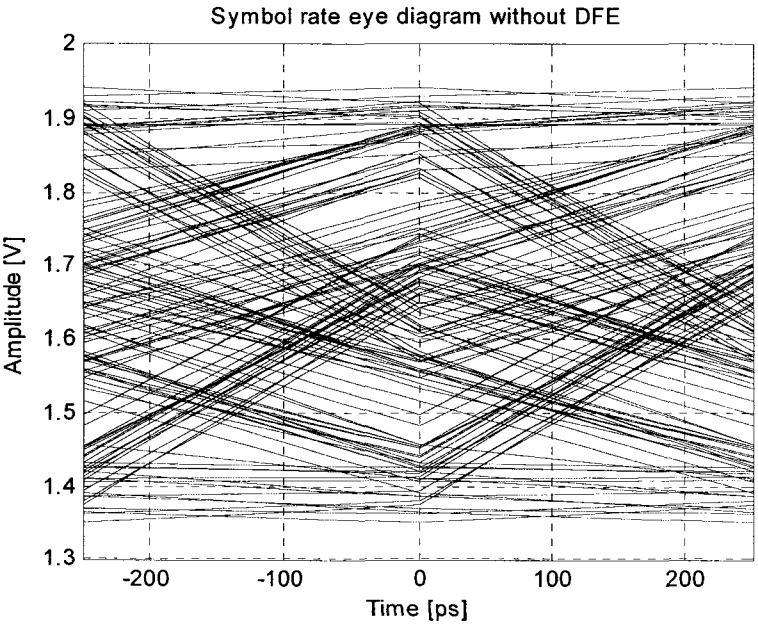

Figure 4.6: Symbol-rate sampled eye-diagram without DFE

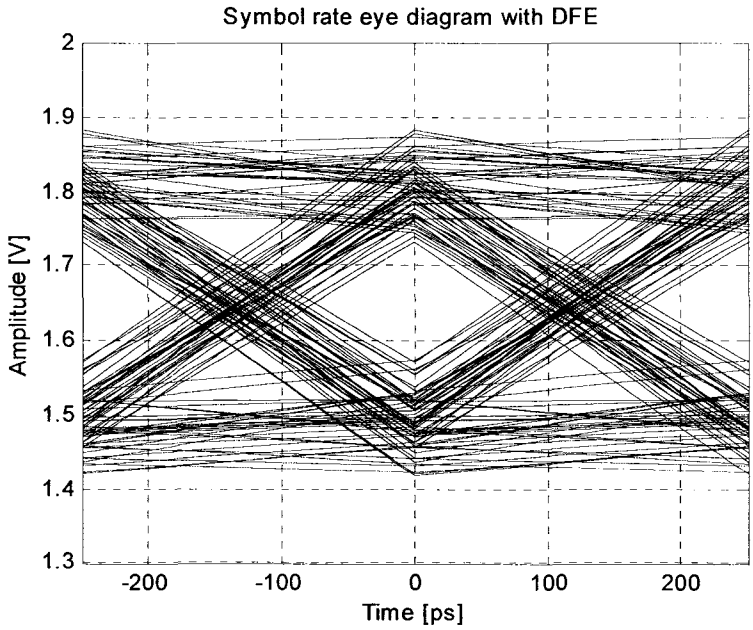

Figure 4.7: Symbol-rate sampled eye-diagram with DFE

The performance of the equalizer is verified by comparing the bit error rate between the behavioral and circuit simulations during the equalizer convergence as shown in Figure 4.8. Each iteration is 1024 symbols long and received data is $2^{16}-1$ pseudo-random pattern. The slight difference in the bit error rate is caused by difference in sampling instance between the behavioral and circuit model. The behavioral simulation uses ideal sampling whereas the circuit simulation is affected by the intrinsic jitter in the clock recovery circuit.

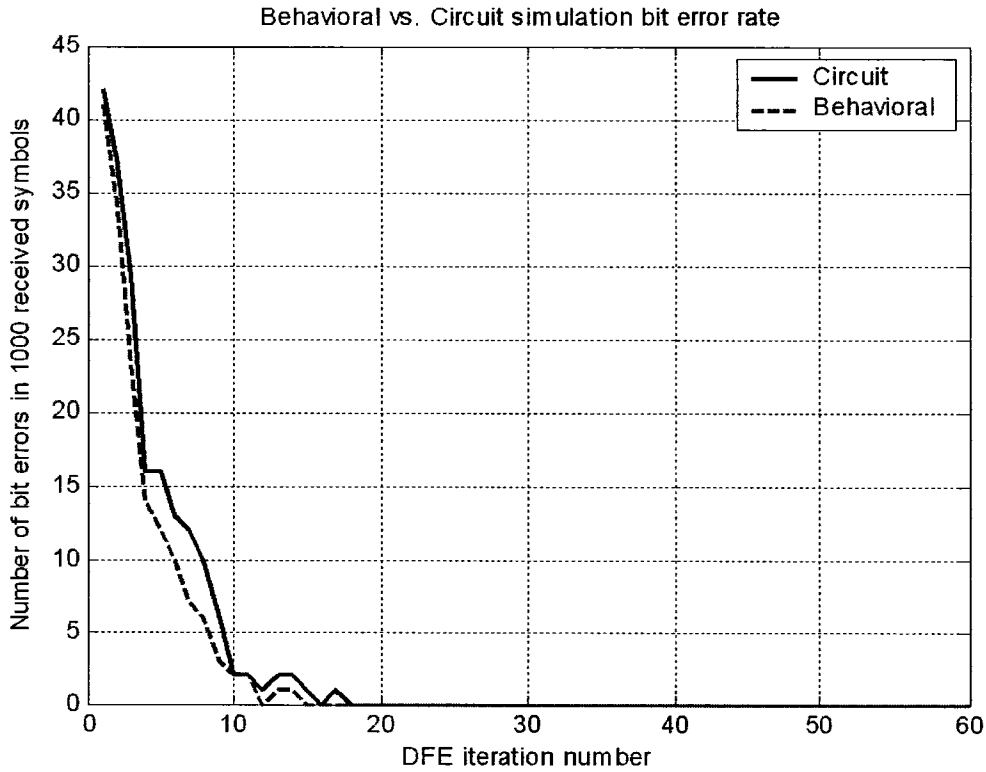

Figure 4.8: Bit-error rate during convergence (Behavioral vs. circuit simulation)

Figure 4.9 shows circuit simulation bit-error rate vs. DFE coefficient value during the convergence of the DFE adaptive algorithm. We can see how bit-error rate drops as DFE converges from zero to the optimum over the number of iteration.

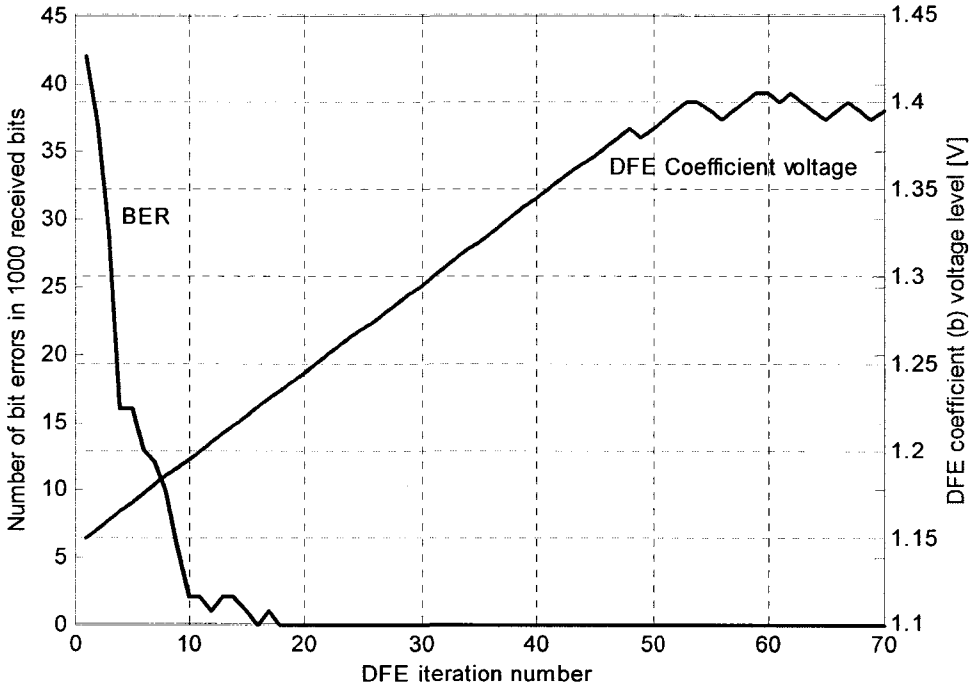

Figure 4.9: Bit-error rate drop during DFE convergence

4.3.1 Comparison between adaptive pre-emphasis and DFE

Figure 4.10 shows how the eye-diagram for the same channel would look like if the two-tap pre-emphasis were used, as opposed to the one-tap DFE. Although, the pre-emphasis eye-diagram appears to have higher vertical opening, it is actually just cleaner. From the Figure 4.7 and Figure 4.10 we can see that the absolute value of the vertical eye opening is just slightly better for the pre-emphasis. The two-tap pre-emphasis does cancel more ISI than the one-tap DFE, but the signal with pre-emphasis has lower peak-to-peak amplitude at the receiver because the pre-emphasis is done by attenuating the low frequency components of the signal (rather than busting high frequency components) due to limited swing of the transmit driver.

The two-tap pre-emphasis though, has obvious advantage as far as the jitter margin (horizontal eye-opening) is concerned.

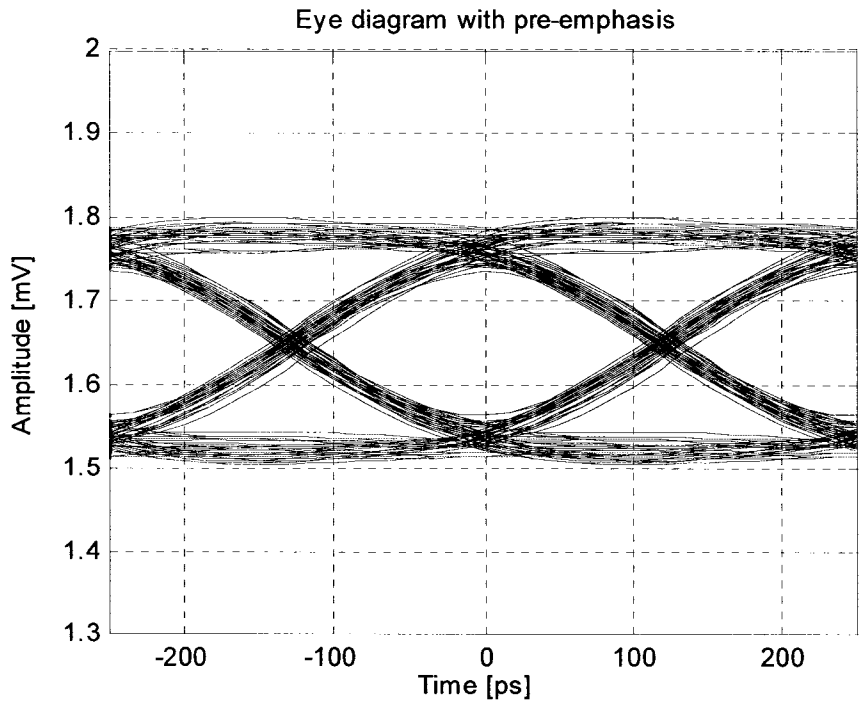

Figure 4.10: Eye-diagram with two-tap pre-emphasis

Although, the performance of the one-tap DFE is inferior when compared with the two-tap pre-emphasis, its major advantage is the fact that it does not need reliable return channel for adaptive tuning as the pre-emphasis. In addition, receivers with adaptive DFE have more flexibility because they could potentially work with transmitters from different semiconductor vendors, whereas adaptive pre-emphasis requires both parts to be from the same vendor (unless some sort of adaptive pre-emphasis becomes an industry standard).

Performance of the DFE could be further improved is we use two-tap DFE or if we use fixed pre-emphasis at the transmitter with one-tap DFE. Fixed pre-emphasis could be

set to cancel ISI for the best channel (shortest backplane link), and the one-tap DFE would take care of ISI for the longer backplane links.

CHAPTER 5

Conclusions and Future Work

5.1 Conclusions

This thesis has shown implementation of a 4 Gbps backplane receiver in 0.35 μm TSMC process. The receiver mitigates ISI with adaptive blind DFE, which is performed by biasing high-speed comparators. Each input symbol is simultaneously sampled with two comparators — one biased high as if the previous symbol was low and the other biased low as if the previous symbol was high. This way we perform look-ahead equalization before previous sampled symbol is known, which significantly improves maximum speed of operation. The high-speed comparator was implemented by modifying Strong-Arm high-speed flip-flop — biasing transistors were added to the sense amplifier and the input hysteresis was solved by modifying standard SR latch used in Strong-Arm flip-flop.

We have shown that the adaptation coefficient does not have to be calculated on every received symbol, which significantly reduces circuit complexity, area and power consumption. Since, DFE implemented in this thesis does not contain feedforward equalizer or adjustable input gain amplifier', we have developed a circuit which adaptively adjusts the cost function in order to compensate for attenuation of the transmission line.

5.2 Future Work

The maximum data rate of the receiver described in this thesis is limited by delay trough the multiplexer and the latch, which was described in Section 3.3.2. However, the receiver in this thesis was not implemented in the latest (fastest) process, but in relatively old 0.35 μm TSMC process. Hence, the simplest way to increase the maximum data rate would be to implement the receiver in 0.18 μm or 0.13 μm technology.

DFE implemented in this thesis had only one feedback tap, which was shown to be enough for 4Gbps data rates over the very long backplane. However, an additional tap, would further improve the eye-opening for 4Gpbs data rates and would enable the receiver to operate at even higher data rates over the same backplane.

To further improve the performance, adaptive DFE could be used with fixed pre-emphasis where pre-emphasis filter coefficients would be preset to cancel best case ISI (ISI caused by the shortest backplane link), and DFE would take care for remaining ISI for longer links.

The clock data recovery circuit implemented in this thesis does not support plesio-synchronous operation. The transmitter and receiver has to be fed by the same clock. Although the clock can run at much lower frequency than the data, and does not have to be phase aligned with the data, it still needs to be filtered from jitter which is caused by distributing the clock over the long traces. To support plesio-synchronous operation, a clock data recovery circuit similar to one described in [16] could be used. This circuit uses a local oscillator which can be placed very close to the receiver. The jitter on such reference clock would generally be lower than the jitter on the clock distributed over the backplane.

References

1. S. Milijevic and T. Kwasniewski, " 4Gbit/s receiver with adaptive blind DFE", IEE Electronics Letters, vol. 25, No. 25, pp. 1373-1374, December 2005.
2. J. T. Stonick et al., "An Adaptive PAM-4 5 Gb/s Backplane Transceiver in 0.25 μm CMOS", IEEE Journal of Solid-State Circuits, vol. 38, pp. 436-443, March 2003.
3. R. Farjad-Rad et al., "A 0.4 –um CMOS 10-Gb/s 4-PAM pre-emphasis serial link transmitter," IEEE Journal of Solid-State Circuits, vol. 34, pp. 580-585, May 1999.
4. W.J. Dally and J. Poulton, " Transmitter equalization for 4Gb/s signaling," in Proceeding of Hot Interconnect Symposium, pp. 29-39, August 1996.
5. M. Li et al., "FIR filter optimization as preemphasis of high-speed backplane data transmission," International Conference on Communications, Circuits and Systems, vol. 2, pp. 773-776, June 2004.
6. J.H.R. Schrader et al., "CMOS transmitter using pulse-width modulation pre-emphasis achieving 33 dB loss compensation at 5-Gb/s", Simposium on VLSI Circuits Digest of Technical Papers, pp. 388-391, June 2005.
7. P. Westergaard et al., "A 1.5V 20/30 Gb/s CMOS backplane driver with digital pre-emphasis," Custom Integrated Circuits Conference, 2004. Proceedings of the IEEE 2004, pp. 23-26, October 2004.
8. D.J. Foley and M.P. Flynn, "A low-power 8-PAM serial transceiver in 0.5-μm digital CMOS," IEEE Journal of Solid-State Circuits, vol. 37, pp. 310-316, March 2002.

9. Y.S. Sohn et al., "A 1.2 Gbps CMOS DFE receiver with the extended sampling time window for application to the SSTL channel," Symposium on VLSI Circuits Digest of Technical Papers, pp. 92-93, 2002.

10. J.M. Khoury and K.R. Lakshmikumar, "High speed serial transceivers for data communication systems," IEEE Communications Magazine, vol. 39, pp. 160-169, July 2001.

11. R. Payne et al., "A 6.25-Gb/s Binary Transceiver in 0.13-m CMOS for Serial Data Transmission Across High Loss Legacy Backplane Channels," IEEE Journal of Solid-State Circuits, vol. 40, pp. 2646-2657, December 2005.

12. Ming-Yu Edward Lee et al. "Low-Power Area-Efficient High-Speed I/O Circuit Techniques", IEEE Journal of Solid-State Circuits, vol, 35, pp. 1591-1599, November 2000.

13. Ming-Yu Edward Lee, "An Efficient I/O and Clock Recovery Design for Terabit Integrated Circuit", PhD Thesis, Stanford University, August 2001.

14. Chih-Kong Ken Yang, " Design of High-Speed Serial Links in CMOS", PhD Thesis, Stanford University, December 1998.

15. J. Montanaro et al., "A 160MHz, 32b, 0.5W CMOS RISC microprocessor," IEEE Journal of Solid-State Circuits, vol. 31, pp. 1703-1714, Nov 1996.

16. S. Sidiropoulos and M. Horowitz, "A Semidigital Dual Delay-Locked Loop", IEEE Journal of Solid-State Circuits, vol. 32, pp. 1683-1692, November 1997.

17. Y. Sato, "A method of self-recovering equalization for multilevel amplitude modulation systems," IEEE Trans. Communication, vol COM-23, pp. 679-682, June 1975.

18. A. Benveniste et al., "Robust identification of a nonminimum phase system: Blind adjustment of a linear equalizer in data communication," IEEE Transaction on Automatic Control, vol. AC-25 pp. 385-398, June 1980.

19. D. N. Godard, "Self recovering equalization and carrier tracking in two-dimensional data communication systems," IEEE Transaction on Communication, vol. COM-28, pp. 1867-1875, November 1980.

20. J. Cioffi, Digital Communications, course notes, http://www.stanford.edu/class/ee379a.

21. J. Proakis, Digital Communications, Fourth edition, McGraw- Hill, August 2000.

22. B. Sklar, Digital Communications: Fundamentals and Applications, Second edition, Prentice Hall, January 2001.

23. W. J. Dally and J. Poulton, Digital Systems Engineering, Cambridge University Press, 1998.

24. H. W. Johnson, High-Speed Signal Propagation—Advanced Black Magic, Prentice Hall, 2003.

25. A. R. Djordjevic, Matrix Parameters of Multiconductor Transmission Lines, Artech House, 1988.

26. W. H. Hayt and J. A. Buck, "Engineering Electromagnetics", McGraw-Hill, 2001

27. PICMG 3.X, collection of standards, AdvancedTCA Specifications for Next Generation Telecommunications Equipment, www.picmg.org.

28. Daniel Foty, MOSFET Modeling with SPICE: Principles and Practice, Prentice Hall, 1997.