國 立 交 通 大 學

電子工程學系 電子研究所

碩 士 論 文

應用於序列傳輸系統

之 10-Gbps 離散時間適應性等化器

A 10-Gbps Discrete-Time Adaptive Equalizer

for Serial Link System

研 究 生：許馥淳

指導教授：周世傑 博士

中 華 民 國 九 十 八 年 九 月

應用於序列傳輸系統之 10-Gbps 離散時間適應性等化器

A 10-Gbps Discrete-Time Adaptive Equalizer for Serial Link System

研 究 生： 許馥淳　　Student：Fu-Chun Hsu

指導教授： 周世傑 博士　　Advisor：Prof. Shyh-Jye Jou

國 立 交 通 大 學

電子工程學系 電子研究所碩士班

碩 士 論 文

A Thesis

Submitted to Department of Electronics Engineering &
Institute of Electronics
College of Electrical and Computer Engineering
National Chiao Tung University
in Partial Fulfillment of the Requirements
for the Degree of Master of Science
in
Electronics Engineering
September 2009
Hsinchu, Taiwan, Republic of China

中 華 民 國 九 十 八 年 九 月

應用於序列傳輸系統之 10-Gbps 離散時間適應性等化器

研究生：許馥淳　　指導教授：周世傑博士

國立交通大學

電子工程學系電子研究所碩士班

摘　要

隨著積體電路製程技術的進步，晶片的操作速度變得越來越快，晶片間傳遞資料的高頻寬輸出入介面愈來愈重要。各種高速序列傳輸技術廣泛的使用在許多高效能的電子產品中。為了讓訊號經過傳輸通道衰減後可以維持一定的品質，等化器在高速序列傳輸系統中扮演了重要的角色。而由於通道的特性會因為環境而改變，適應性等化器較適合於長時間的使用。

在本論文中，我們提出了一個操作在 10 Gbps 2-tap 的離散時間適應性決策回授等化器。在等化器系統的前端，我們設計了一個可變增益的放大器，來調整輸入訊號的振幅到下一級電路接受的範圍。另外，我們設計了一個高速、電流模式的加法器，用來消除 post-cursor ISI。接著，我們提出一個叫跳躍式係數更新的方案，還有一個混合訊號的積分器，來實現係數更新的機制。跳躍式係數更新方案可以藉由降低係數更新的速度，來減少功率的消耗。混合訊號積分器是由一個四位元的上數下數計數器和一個電荷幫浦組成，用來獲取更好的輸出表現和較小的面積。我們使用了延遲 sign-sign LMS 演算法來做係數的收斂。提出的等

化器是用 65 奈米互補式金氧半導體製程來設計。模擬結果位於等化器輸出的信號眼圖可以開至正負 200 mV，而緩衝器的輸出端可以將信號眼圖張開到規格所定的正負 300 mV。在等化器輸出的峰對峰值抖動大約 31ps。而係數的收斂時間約為 20000 個位元時間。電路總面積為 510 × 510 μm^2，而核心電路面積是 115 × 95 μm^2。在 1.2V 的操作電壓下，電路總功率為 40.63 mW，其中等化器系統佔了 11.18 mW。

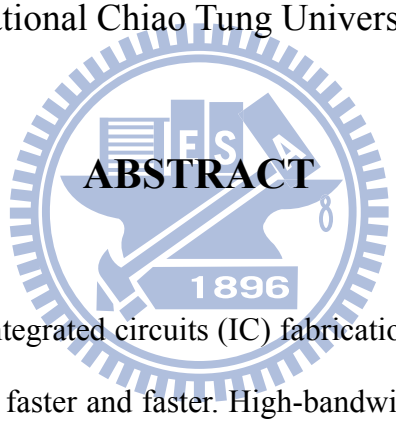
A 10-Gbps Discrete-Time Adaptive Equalizer for Serial Link System

Student: Fu-Chun Hsu Advisor: Prof. Shyh-Jye Jou

Department of Electronics Engineering

Institute of Electronics

National Chiao Tung University



ABSTRACT

With the advance of integrated circuits (IC) fabrication technology, the operation speed of chips is becoming faster and faster. High-bandwidth I/Os have found a great demand for transferring data between chips. Many high-speed serial link transmission technologies are developed and are widely used for high performance modern electronic products. In order to maintain the signal quality that will be attenuated by communication channel, the equalizer becomes an important component in the high-speed serial link system. Since the characteristics of channel may vary due to the environment, adaptive equalizer is much preferable for long-time usage.

In this thesis, we propose a 2-tap discrete-time adaptive decision-feedback equalizer that operates at 10 Gbps. We design a variable gain amplifier in the front of the proposed equalizer system to adjust the swing of input signal in the range for the following stage. A high-speed current-mode summer is designed to cancel the

post-cursor ISI. A coefficient updating scheme called hopping and a mixed-signal integrator are presented to realize the mechanism of coefficients adaptation. The hopping update scheme can reduce the power consumption by slowing down the operation speed in the coefficients adaptation. The mixed-signal integrator consists of a 4-bit up/down counter and a charge pump to acquire a good performance and small area. We use the delayed sign-sign LMS algorithm to do the convergence of coefficients. The proposed equalizer is designed in a 65-nm CMOS technology. The simulation result shows that the data eye in the output of equalizer is about ±200 mV_{pp}, and the data eye in the output of buffer stage can reach ±300 mV_{pp} that meets our specification. The peak-to-peak jitter at the equalizer output is about 31ps. The convergence time of coefficients is about 20000 bits time. Total area of our proposed equalizer including pads is 510 × 510 μm^2 while the core area is 115 × 95 μm^2. The total power consumption is 40.63 mW while the equalizer system consumes 11.18 mW under 1.2V power supply.

誌　　謝

終於到了論文完成的這一天，真的很開心。首先要感謝我的指導教授周世傑老師，提供非常完善的研究環境與資源，另外在每次的 meeting 報告時所給予的糾正及建議，都讓我獲益良多，研究也才能這麼順利。因為老師的諄諄教誨，讓我在研究的領域裡，不斷的精進，也才能完成論文，非常感謝老師。

再來要感謝育群學長，在我研究過程中，給予我相當大的幫助，尤其是觀念的解釋，還有研究細節的指導。由於學長的幫助，我的研究才不至於停滯不前，非常謝謝學長。

接著要感謝實驗室的學長跟同學們：jackie、mike、小胖、大大、小肥、明賢、志宇、代暘、明銓，大家一起在實驗室度過很多快樂時光，一起研究、互相學習。另外還有學弟妹：以樂、銘謙、祥譽、雅雪、為凱。有了大家，研究的日子很快就在歡樂聲中度過了。

最後要感謝我的父母，總是給予我關心和鼓勵，讓我更可以全力專心在研究上，最終也才能完成這個學位與論文。因為你們的栽培，才有今日的我，謝謝你們。

Contents

List of Figures

List of Tables

Chapter 1
Introduction

1.1 The Challenge for High-Speed Serial Link

With the advances in integrated circuits (IC) fabrication technology, the speed of on-chip data processing as well as the integration level will continue to increase. High-speed inputs and outputs (I/Os) become an important role to transfer large amounts of data in many applications, including computer-to-peripheral connections, local-area networks, inter-chip communications and so on. In modern interconnect systems, high-speed serial links have replaced parallel data buses and become the dominant I/Os for data transmission. A typical serial link block diagram is shown in Fig. 1.1. The serial link is composed of three primary components: a transmitter, a channel and a receiver. We will discuss these three blocks in chapter 2.

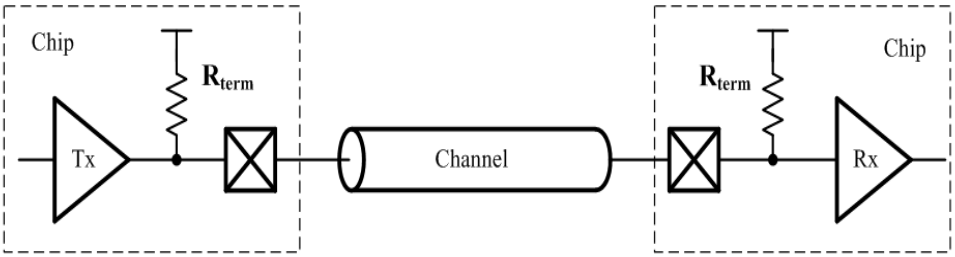


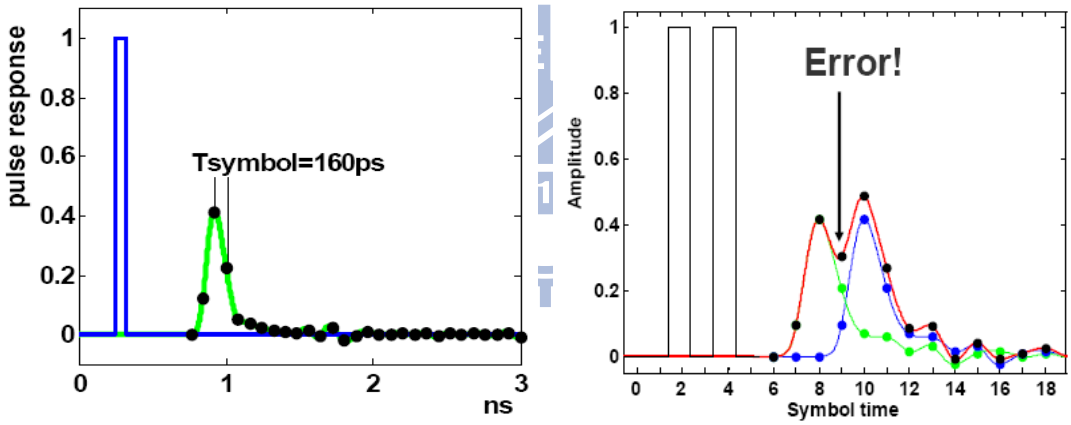
Figure 1.1 A typical serial link block diagram.

Table 1.1[1] shows the data rates of industrial standards about high-speed serial links. As the speed of data rates increase, the design of high-speed serial links becomes a big challenge for maintaining signal integrity since the off-chip bandwidth

Table 1.1 Industrial data-rate standards of high-speed serial link.

Serial ATA	1.5/3/6 Gb/s
PCI-Express	5 Gb/s
USB 3.0 (High Speed)	5 Gb/s
IEEE 802.3ae	10 Gb/s

scales at a much lower rate compared to the on-chip bandwidth. The band-limited channel causes intersymbol interference (ISI) and severely degrades the transmitted data. Fig. 1.2 illustrates the phenomenon of ISI. In Fig. 1.2(a), a nice shot pulse gets spread out due to a dispersive channel. When two consecutive pulses are fed into the channel, the output is the summation of the two spread out waveforms, as shown in Fig. 1.2(b). This will cause bit error in the detection of receiver.



(a) (b)

Figure 1.2 Illustration of ISI phenomenon.[a] (a) An impulse and its response. (b) A series of impulse and their response.

[1]Reference: "Serial ATA II Electrical Specification Revision 1.0, 26 May 2004.", "PCI Express Base Specification Revision 2.0, 27 Feb. 2009.", "Universal Serial Bus 3.0 Specification Revision 1.0, 12 Nov. 2008.", "IEEE Std. 802.3ae: IEEE standard for 10 Gbps Ethernet."

[a]This figure is imaged from the tutorial "Lecture #4 Communication Techniques: Equalization & Modulation in Advanced Topics in Circuit Design: High-Speed Electrical Interface," 2004 by Jared Zerbe.

1.2 Motivation and Challenge of High-Speed Equalizer

The task of equalization is to recover the transmitted data from distortion due to the effect of ISI. As the data rate increases up to multi-Gbps, the impact of ISI on the signal becomes very severe, imposing difficulty on the design of equalizer. To compensate for the signal loss, some approaches have been proposed in high-speed serial links. Pre-emphasis in the transmitter [1]-[4], equalizers in the receiver [5]-[8], or a combination of the two [9]-[12], are employed to do equalization for high data rates. Since the characteristics of channel are not known in advance and they may vary due to variations in environment, adaptive equalizers are preferred in applications for long-time using. Decision-feedback equalizer (DFE) with coefficients adaptation (ADFE) in the receiver has better performance to cancel ISI than adaptive linear equalizer. However, for high data rate application, its design challenge is very high due to signal feedback loops in data path and coefficients updating loops.

In this thesis, the proposed 2-tap adaptive decision-feedback equalizer can cancel post-cursor ISI at data rate of 10 Gb/s. We employ a hopping coefficient update scheme to reduce power consumption and a mixed-signal integrator to save area and enhance the performance. The proposed equalizer is designed in 65-nm CMOS technology.

1.3 Thesis Organization

This thesis is organized as follows:

Chapter 2 begins with the brief introduction of the serial-link system. Then, a

general description of the transmitter, receiver and the role of equalizer in this system is carried out. Then, fundamental concepts of transmission are reviewed, including eye diagram, bit error rate, and so on. Finally, channel modeling will be described.

In chapter 3, we discuss the concept of equalization. Two kinds of equalizers, continuous-time equalizer and discrete-time equalizer, depending on the domain of signal processing will be introduced. Then, a brief derivation of the sign-sign LMS algorithm will be given.

Chapter 4 begins with the overall architecture of our proposed equalizer. We will discuss the key components used in the proposed system. Then, an updating scheme called hopping and the operation of a mixed-signal integrator will be presented. Finally, behavioral simulations of our equalizer on different conditions will be shown.

In chapter 5, the implementation of our proposed equalizer will be presented. We will begin with the operation of our equalizer system. Next, circuit design details, simulation results and chip layout are given. Finally, measurement setup consideration is also discussed.

In chapter 6, a brief conclusion and future work are given.

Chapter 2
Overview of Equalizer for Serial Link System

High-speed serial links have become an indispensable interface in the applications of inter-chip communication. In this chapter, we will give a short overview of serial link system. Next, some basic concepts about signal processing will be given. Finally, we will describe the method of channel modeling.

2.1 Overview of Serial Link System

Fig. 1.1 is a typical block diagram of serial link. The serial link comprises three primary components: a transmitter, a channel and a receiver. The transmitter converts digital bits into a signal stream that is propagated on the channel to the receiver. The receiver transfers this analog signal into binary data. Fig. 2.1 illustrates the signaling in a serial link.

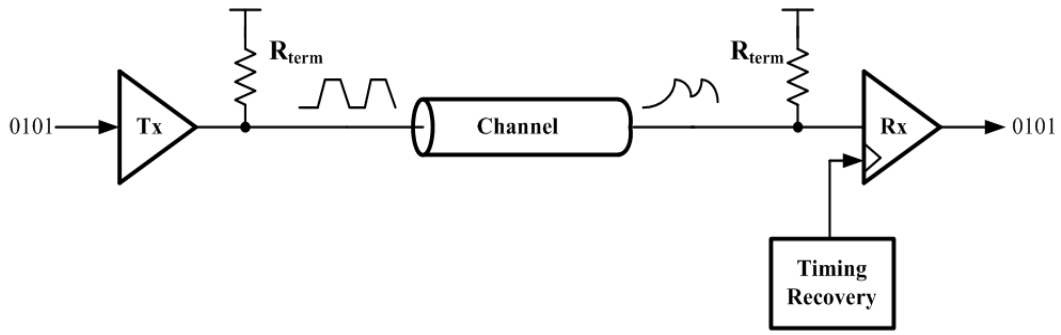


Figure 2.1 Signaling in a typical serial link.

A transmitter sends the data as analog symbols. The analog value represents a single bit, known as non-return-to-zero (NRZ). For electrical systems, different levels have different voltages. The duration of each ONE or ZERO is the bit time. The difficulty in a transmitter design is to maintain clean signal levels while transmitting data in high data rates.

In electrical systems, differential signaling is often used. It is a method of transmitting information electrically by means of two complementary signals sent on two separate wires. An important advantage of differential signaling over single-ended signaling is higher immunity to environmental noise. Since the impacts of common-mode noise on differential signals are opposite, the common-mode noise will be rejected by using differential signaling.

The channel is the medium where the data is propagated. This medium would be an optical fiber, a coaxial transmission line, a printed-circuit board (PCB) trace or the chip package. The medium can attenuate or filter the signal and introduce noise.

To recover the bits from the signal, the analog waveform is sampled and equalized. The receiver-end must be able to resolve small inputs at high data rates to recover high-speed signals. An additional circuit, the timing-recovery circuit, properly places the sampling strobe for the receiver to amplify and sample the waveform correctly.

Fig. 2.2 shows a typical serial link with equalizers. The equalizer in the transmitter side can be called pre-emphasis. The topology of this pre-emphasis is usually a feed-forward equalizer (FFE). For a channel that changes with time, the pre-emphasis circuit would need updated information from the receiver. The equalizer set at the receiver side is usually a decision-feedback equalizer (DFE). Since the characteristics of channel may vary due to environment, the coefficients of DFE need to update adaptively. This type of DFE is called adaptive DFE (ADFE). When FFE

and DFE are used together optimally, the strengths of both can be cooperated to bear on the channel and maximum performance is obtained.

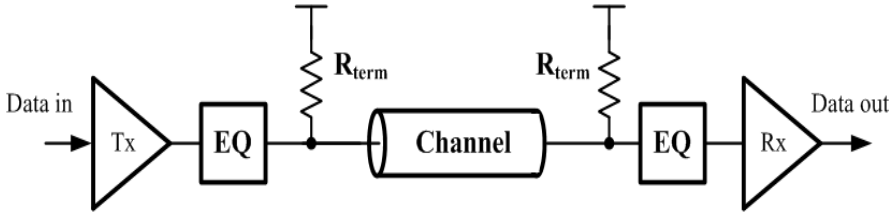


Figure 2.2 Typical serial link with equalizers.

Fig. 2.3 shows the equalizer in the receiver side. There are two FIR filters in Fig. 2.3: feed-forward filter (FFF) and feed-back filter (FBF). FFF is designed to ease the pre-cursor ISI, and FBF is designed to cancel the remained post-cursor ISI. Many literatures call Fig. 2.3 as a DFE. In this thesis, we define DFE as the dashed area in Fig. 2.3 and define FFF as FFE. We will discuss FFE and DFE in Section 3.2.

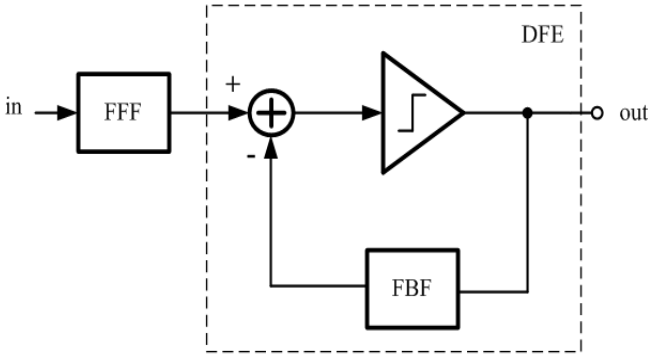


Figure 2.3 Block diagram of equalizer in receiver side.

2.2 Basic Concepts

2.2.1 NRZ Data

"Non-return to zero" (NRZ) data is a data format used in many high-speed communication systems. As shown in Fig. 2.4, Each logical bit of NRZ data specifies

one of two signal levels. This type of waveform is called NRZ to distinguish it from "return to zero" (RZ) data. Each bit of RZ data consists of two sections: the first section represents the bit value, and the second section is always equal to a logical zero. In other words, every two bits are separated by a redundant zero symbol. Because the property of RZ data, it needs about twice as much bandwidth as NRZ data does. This is the reason why NRZ data is much suitable for high-speed applications.

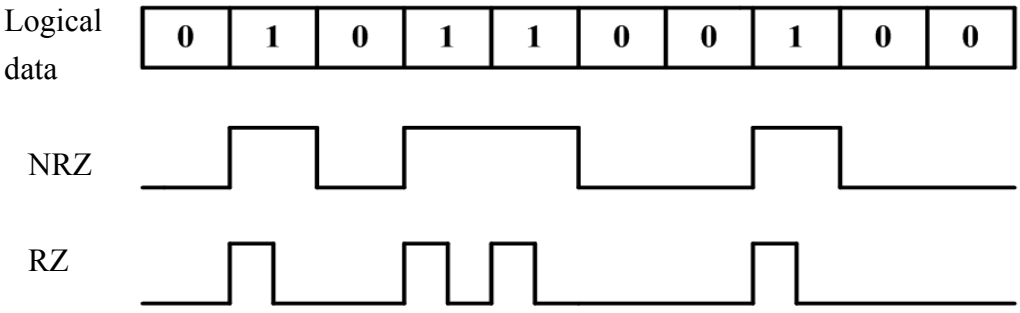


Figure 2.4 NRZ and RZ data.

2.2.2 Pseudo-Random Binary Sequence

Most wire-line communication systems employ binary amplitude modulation. A random binary sequence (RBS) comprises logical ONEs and ZEROs that usually occur with equal probabilities. In Fig. 2.5, if each bit period is T_b seconds, then the bit rate, R_b, is equal to $1/T_b$ bits per second. Fig. 2.5 also reveals that the ONEs and ZEROs assume equal and opposite values, thereby yielding a zero average.

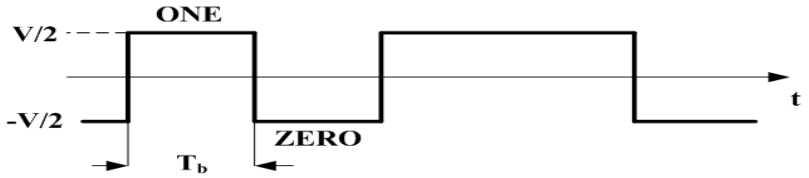


Figure 2.5 Random binary sequence.

In simulation and chip measurement, it is difficult to generate absolutely RBS. Hence it is common to utilize "pseudo-random" binary sequence (PRBS). Each PRBS is a repetition of a pattern, containing a RBS of a number of bits (Fig. 2.6).

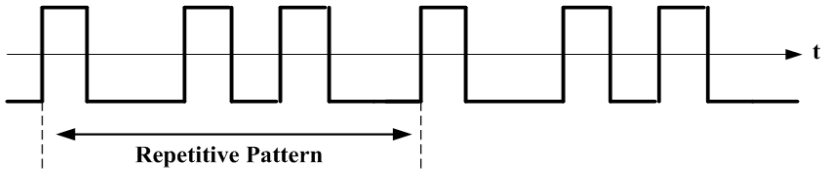


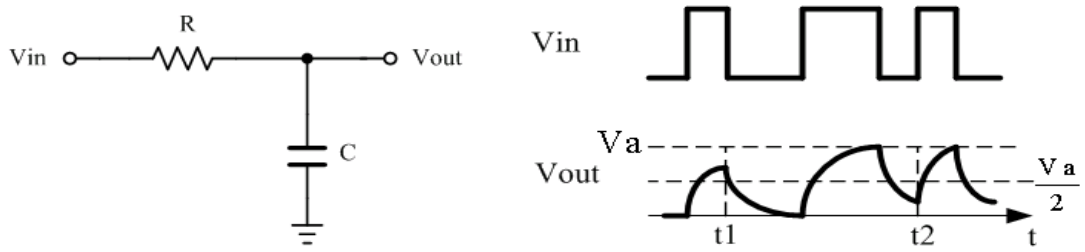
Figure 2.6 Pseudo-random binary sequence.

A PRBS of length $2^m - 1$ means the pattern repeats every $2^m - 1$ bits, where m is a positive integer. For example, if there is a PRBS of length $2^3 - 1$, the sequence will repeat every 7 bits, and the maximum run length is 3. Maximum run length is the maximum number of consecutive ONEs or ZEROs in a pattern. The random nature of data implies that a binary sequence may contain arbitrarily long maximum run length. The long runs produce difficulties in the design of receiver circuits. For instance, in CDR design, the low data transitions during a long run may cause the oscillator to drift and hence generate jitter. Therefore, random data may be encoded to limit the maximum run length. 8-bit/10-bit (8B/10B) coding [13] is a coding algorithm that converts a sequence of 8 bits to a 10-bit word to guarantee a maximum run length of 5 bits. Although the data rate increases by 25%, many aspects of transceiver design can be relaxed.

2.2.3 Intersymbol Interference

Intersymbol Interference (ISI) has been a serious limitation on data rates that can be sent through a communication channel. ISI generally refers to the interference that occurs between the current received bit and other previously received bits. The

interference indicates that the portions of previously received bits add to the bit that is being received. This phenomenon can be seen by the following example.



(a) (b)

Figure 2.7 (a) A low-pass filter. (b) Effect of low-pass filtering on random data.

Fig. 2.7(a) is a RC low-pass filter, when a random binary sequence passes through it, the high-frequency components are attenuated, as illustrated in Fig. 2.7(b). For a single ONE followed by a ZERO, the output does not reach the peak, but for two consecutive ONEs, it does. The output voltage levels corresponding to ONEs and ZEROs vary with time, making it difficult to define a decision threshold voltage. For example, if the threshold is set at Va/2, the voltages at $t=t_1$, and $t=t_2$ are very susceptible to noise and the decision circuit may make a wrong decision of bit. Such undesirable phenomenon is called "intersymbol interference" (ISI). ISI produces degradation in system performance and is the major source of bit errors.

In fact, there are two factors that ISI is occurred by the low-pass nature of communication channel. One is the bandwidth of channel, and the other is the density of transitions in the data stream. For the factor of bandwidth, we can see Fig. 2.7 again. The rising edge of output voltage before t=t1 can be expressed as

$$V_{out}(t)=V_a(1-e^{-t/RC}). \tag{2.1}$$

Therefore, the larger the RC, i.e. narrower bandwidth, the lower output voltage level can reach. For the factor of the transition density of input pattern, Fig. 2.8 is an example to show the problem.

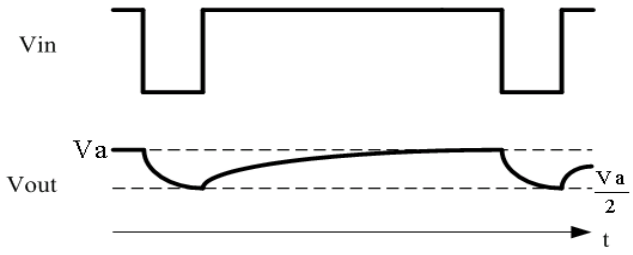
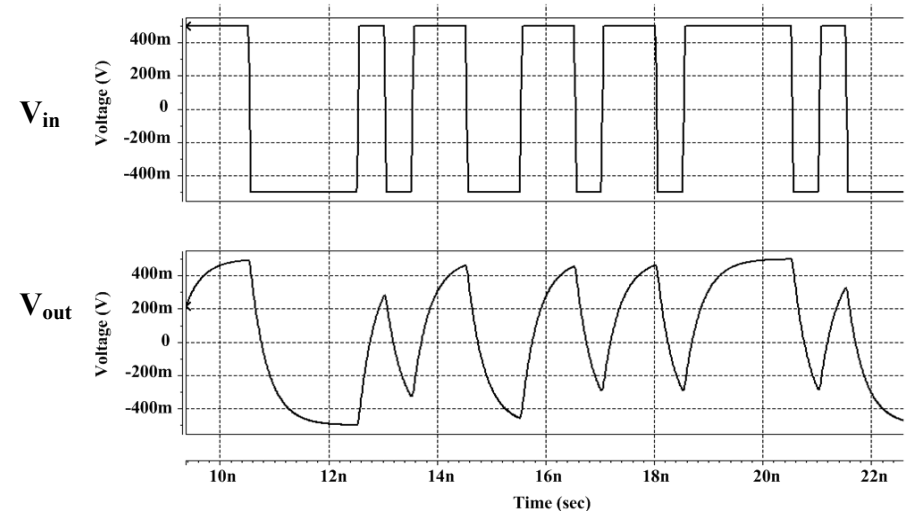


Figure 2.8 Response of long run passes a moderate limited bandwidth channel.

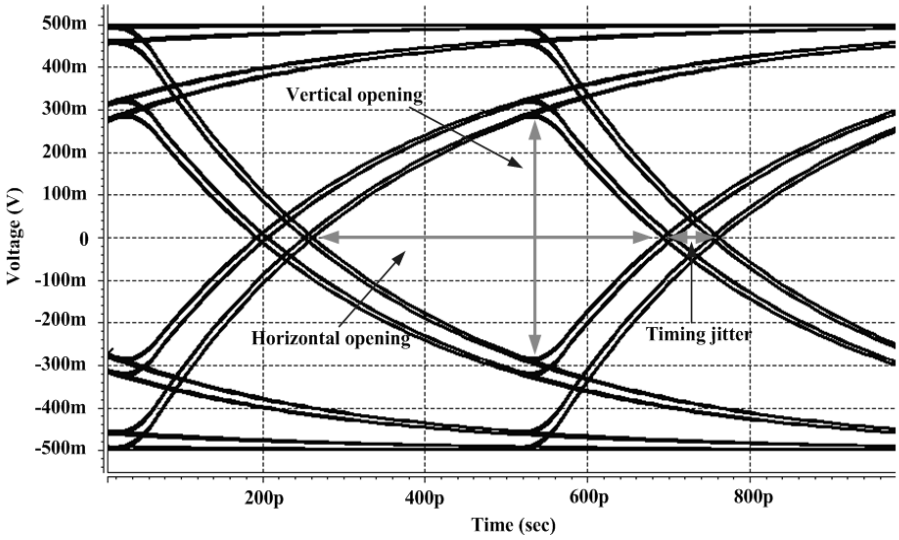
In Fig. 2.8 assume the bandwidth of channel is moderate limited, when more consecutive ONEs are followed by a single ZERO, the output voltage which is supposed to be a ZERO may be far from logical ZERO. It is obvious that the longer the identical series of ONEs before a ZERO, the worse the effect of ISI on the amplitude of the ZERO bit and vice versa. Since the longer the input stays at constant amplitude, the more the channel charges to the amplitude which will make it more difficult to discharge when the input switches.

2.2.4 Eye Diagram

As discussed so far, ISI is a function of the bit patterns being sent across the channel. If the input pattern is very long and random, it becomes a very difficult task to find out the effect of ISI on both the amplitude and duration of the received bits. A common method for visualizing the nonidealities in random data is the “eye diagram.” An eye diagram is created by capturing the output waveform which is divided into a short interval, e.g., two bits wide, and overlaid on top of each other. As an example, considering a 2-Gb/s random binary sequence is fed into a first-order low-pass filter having a －3-dB bandwidth of 500 MHz. As depicted in Fig. 2.9, we superimpose all 1-ns intervals to obtain the eye diagram. The two important parameters in an eye diagram are the vertical and horizontal openings of the eye. The vertical eye opening



(a)



(b)

Figure 2.9 An illustration of eye diagram construction. (a) input and output waveform. (b) eye diagram.

represents the minimum amplitude the received bits can have. On the other hand, the horizontal eye opening defines the time interval over which the data can be sampled without error caused by ISI. In Fig. 2.9(b), we can also observe that the zero crossings experience some deviation from their ideal position. This is called "timing jitter." As the amount of timing jitter increases, the horizontal eye opening decreases. Therefore, when data rates increase, timing jitter can become a significant problem and cause bit errors in data transmission systems.

2.2.5 Bit Error Rate

The general definition of "bit error rate (BER)" is

$$\text{BER}=\frac{\text{number of bit errors}}{\text{total number of bits received}}. \tag{2.2}$$

In other words, the bit error rate (BER) is the percentages of bits that have errors relative to the total number of bits received in a transmission. It is unavoidable that the noise will exist in data transmission. The noise added to the signal degrades both the amplitude and the time resolution, closing the eye and increasing the BER.

Assume the noise amplitude n(t) has a Gaussian distribution with zero mean and a root mean square (rms) value of σ_n . Thus, we could write the PDF of n(t) as:

$$P_n=\frac{1}{\sigma_n\sqrt{2\pi}}\exp\left(\frac{-n^2}{2\sigma_n^2}\right). \tag{2.3}$$

Next, if ONEs and ZEROs of the input sequence x(t) occur with equal probabilities, the PDF of x(t) contains two impulses at $x=-V_a$ and $x=+V_a$ (assume the nominal values of ONEs and ZEROs are $+V_a$ and $-V_a$, respectively), and each has a weight of 1/2. Having PDF of x(t) and n(t), we could obtain the amplitude distribution of x(t)+n(t), as shown in Fig. 2.10.

As illustrated in Fig. 2.10, x(t)+n(t) shows a PDF consisting of two Gaussian distributions centered around $+V_a$ and $-V_a$. The shaded area represents the error

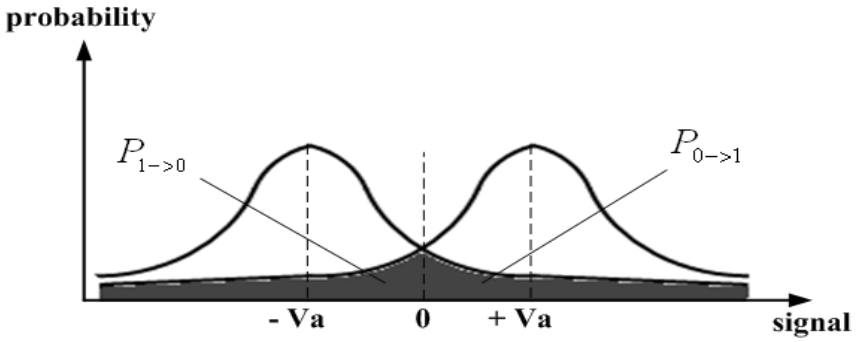


Figure 2.10 PDF of signal plus noise.

samples and the BER can be expressed as:

$$\text{BER}=(P_{0->1}+P_{1->0}), \tag{2.4}$$

where $P_{0->1}$ and $P_{1->0}$ represent the probability of "receiving a ONE while actually a ZERO is transmitted" and "receiving a ZERO while actually a ONE is transmitted", respectively.

The probabilities of $P_{0->1}$ and $P_{1->0}$ can be derived as [14]:

$$P_{0->1}=\frac{1}{2}\int_{0}^{+\infty}\frac{1}{\sigma_n\sqrt{2\pi}}\exp\left(\frac{-(u+V_a)^2}{2\sigma_n{}^2}\right)du=\frac{1}{4}erfc\left(\frac{V_a}{\sqrt{2}\sigma_n}\right) \tag{2.5}$$

$$P_{1->0}=\frac{1}{2}\int_{-\infty}^{0}\frac{1}{\sigma_n\sqrt{2\pi}}\exp\left(\frac{-(u-V_a)^2}{2\sigma_n{}^2}\right)du=\frac{1}{4}erfc\left(\frac{V_a}{\sqrt{2}\sigma_n}\right), \tag{2.6}$$

where the complementary error function erfc(t) is defined as:

$$erfc(t)=\frac{2}{\sqrt{\pi}}\int_{t}^{\infty}e^{-z^2}dz. \tag{2.7}$$

where $z=\frac{u+V_a}{\sqrt{2}\sigma_n}$.

Consequently, the BER is given by

$$\text{BER}=\frac{1}{2}\text{erfc}\left(\frac{V_a}{\sqrt{2}\sigma_n}\right). \tag{2.8}$$

Table 2.1 is the different BERs for different confidence intervals of root mean square value σ_n. For serial data transmission schemes, the required BER performance is generally less than about 10^{-12}, which according to this table translates to a necessary minimum margin of over $\pm 7\sigma_n$.

In equalizer design, the timing jitter will affect the BER. The causes of jitter can be categorized into two types: deterministic jitter and random jitter. Deterministic jitter is often called a systematic jitter since it is generated by the system. Clock timing jitter and data signal jitter are deterministic jitter and they are predictable and bounded. We usually use peak-to-peak value to measure deterministic jitter. Random

jitter, also called Gaussian jitter, is an unpredictable electronic timing noise such as thermal noise. Root-mean-square value is often used to measure the random jitter. Hence, if the random jitter is severe, the value of σ_n is large, and the BER is large according to EQ. 2.8.

Table 2.1 BERs for different confidence intervals.

interval	BER
$\pm 1\sigma_n$	0.159
$\pm 2\sigma_n$	2.28×10^{-2}
$\pm 3\sigma_n$	1.35×10^{-3}
$\pm 4\sigma_n$	1.84×10^{-5}
$\pm 5\sigma_n$	2.14×10^{-7}
$\pm 6\sigma_n$	9.87×10^{-10}
$\pm 7\sigma_n$	1.28×10^{-12}
$\pm 8\sigma_n$	6.22×10^{-16}
$\pm 9\sigma_n$	1.13×10^{-19}

2.3 Channel Model

In previous sections, we have roughly introduced the effect of ISI on the quality of signal in data transmission. Since ISI arises from some imperfections of channels, in this section, we will discuss the characteristics of channel and build the channel model for our equalizer circuit design.

There are several types of channels utilized in high-speed interconnects, primarily based on the target application. These channels can be roughly classified into three categories. First, for chip-to-chip communication on a printed circuit board (PCB), short copper traces are used. Second, for systems such as local-area network requiring high-speed connection between two computers, coaxial cable or optical

fiber is used as the transmission medium. Finally, copper traces along with backplane connectors are employed for high-speed board-to-board communication systems such as routers. In this section, we focus on the PCB traces used for chip-to-chip communication.

The channel is the entire path from the output of the transmitter circuit to the input of the receiver circuit. This includes the connections from the chip I/O pads to the package pin on both sides and the PCB trace that connects them. The termination resistor is used to match the characteristic impedance of the channel for minimizing the signal reflection. As we know, a signal can continue to propagate along a transmission line as long as the impedance remains constant. Changes in the impedance along the line will cause part of the signal energy be reflected which then propagates in the opposite direction. If the signal is reflected again, the second reflection would interfere with the signal that is now transmitted. Hence, it appears as a signal-dependent noise [15].

The source of these reflections occurs at the two ends of transmission line if the energy being propagated is not dissipated. Using a line-termination resistor whose value matches the impedance of the line is the most common solution. Then the voltage and current would follow the Ohm's law, and reflection will be eliminated. The resistance of the line-termination resistor is usually 50Ω or 100Ω. Error in matching the termination resistor can still cause reflection. The reflection coefficient can be calculated by [16]:

$$\Gamma = \frac{Z_L - Z_0}{Z_L + Z_0} \tag{2.9}$$

where Γ is the reflection coefficient; Z_L is the load impedance; and Z_0 is the channel characteristic impedance.

The channel can be viewed as a transmission line. A transmission line is an

interconnect whose length is a significant fraction of the wavelength of interest. An ideal transmission line can be modeled as shown in Fig. 2.11. As depicted in Fig. 2.11, a distributed LC ladder forms the ideal transmission line model. L_0 and C_0 denote the inductance and capacitance of the line per unit length, respectively. The characteristic impedance of the line is equal to

$$Z_0 = \sqrt{\frac{L_0}{C_0}}. \tag{2.10}$$

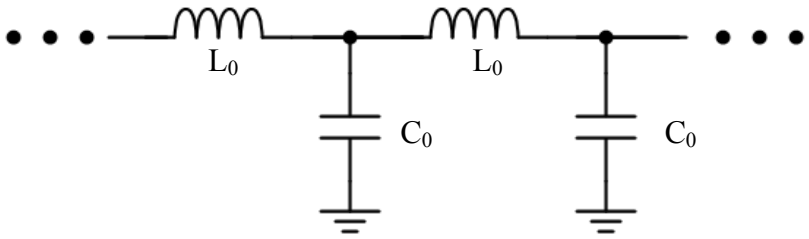


Figure 2.11 LC model of a transmission line.

Since there is no resistance of conductors in the model of transmission line, the energy of signal does not dissipate, and this is called "lossless." If a lossless uniform line is matched at the far end, a step signal propagates and reaches the load with no distortion or attenuation while only exhibiting a frequency-independent delay. A lossless transmission line possesses an infinite bandwidth and a linear phase shift. Unfortunately, there is no this kind of material in practical applications. Essentially, channels must suffer from substantial resistance, and the resulting loss must be taken into account.

The circuit for this signaling medium should be modeled as distributed energy-storage elements, inductance and capacitance, and with some loss due to the series resistor and conductive component of the dielectric. A series of lumped RLGC elements form the lossy channel model, as shown in Fig. 2.12.

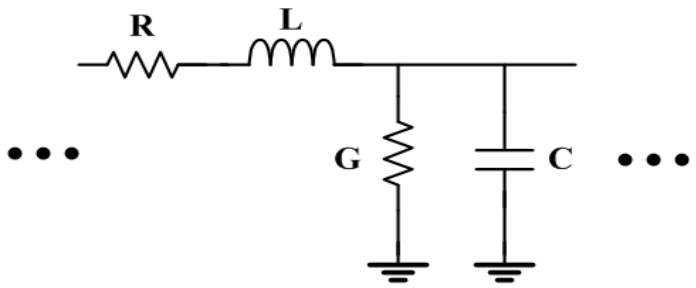


Figure 2.12 Lumped RLGC model of transmission line.

In Fig. 2.12, the RLGC is per-unit-length. In high-speed chip-to-chip design, the frequency-dependent attenuation of channel must be taken into account for a reliable communication. Actually, the dominant sources of loss in channels are due to skin effect and dielectric loss. At high frequencies, current flows closer to the surface of the conductor, reducing the area of current flow and increasing the resistive loss. This is known as skin effect. Since the dielectric material for PCB is not a perfect insulator, it leads to a DC loss associated with the resistive drop between the signal conductor and reference plane. If the frequency of signal increases, the dielectric loss would be worsened. Here, the resistor R embodies both low-frequency and high-frequency conductor losses while conductor G models the dielectric loss. The characteristic impedance of the lumped RLGC model is equal to [17]:

$$Z_0 = \sqrt{\frac{R+jwL}{G+jwC}}. \tag{2.11}$$

When the frequency of signal increases, the current propagates closer to the surface of the conductor with a limited depth. This is called skin depth (δ) and it can be expressed as [18]:

$$\delta = \frac{1}{\sqrt{\pi \cdot f \cdot \mu \cdot \sigma}}, \tag{2.12}$$

where f is the frequency, μ is the permeability ($4\pi \times 10^{-9}$ H/m for copper), and σ is the conductivity of the conductor. If the frequency (f_S) is above the point where δ is half of the conductor thickness t, then the resistance R will be frequency-dependent and

must be considered. Here f_S is given by

$$f_S = \frac{1}{(\frac{t}{2})^2 \pi \times \mu \times \sigma} . \quad (2.13)$$

Above this frequency, the resistance of the conductor can be calculated as

$$R(f) = R_{DC} \sqrt{\frac{f}{f_S}} . \quad (2.14)$$

An equation for the loss due to skin effect for a transmission line of length L is given by

$$A_S(f) = \frac{R_{DC}}{2Z_0} \sqrt{\frac{f}{f_S}} \quad (2.15)$$

Moreover, the attenuation due to dielectric loss is given by

$$A_D(f) = \frac{\pi \sqrt{\varepsilon_r} \cdot f \cdot \tan \delta_D}{c}, \quad (2.16)$$

where ε_r is the relative dielectric constant of the dielectric, $\tan \delta_D$ is the loss tangent of the dielectric, and c is the speed of light [18]. The loss due to skin effect is proportional to the square root of f and typically dominates the total loss at low frequencies. On the other hand, dielectric loss is proportional to f, and therefore, determines the total loss at high frequencies.

Fig. 2.13 and Fig. 2.14 shows the frequency and impulse responses of 6-inch backplane channel model by using MATLAB, respectively. Note that this model is from practical measurement of backplane channel. We can observe that when frequency is at 5GHz, the attenuation is about 15dB. Fig. 2.14 reveals the impact of ISI. The highest tone is the main cursor. The right-hand side of main cursor is called post-cursor ISI and the left-hand side of main cursor is called pre-cursor ISI. We will discuss the pre- and post-cursor ISI in Sub-section 3.2.1. We can observe that the first and second post-cursor ISI dominate the overall ISI and this information gives us a

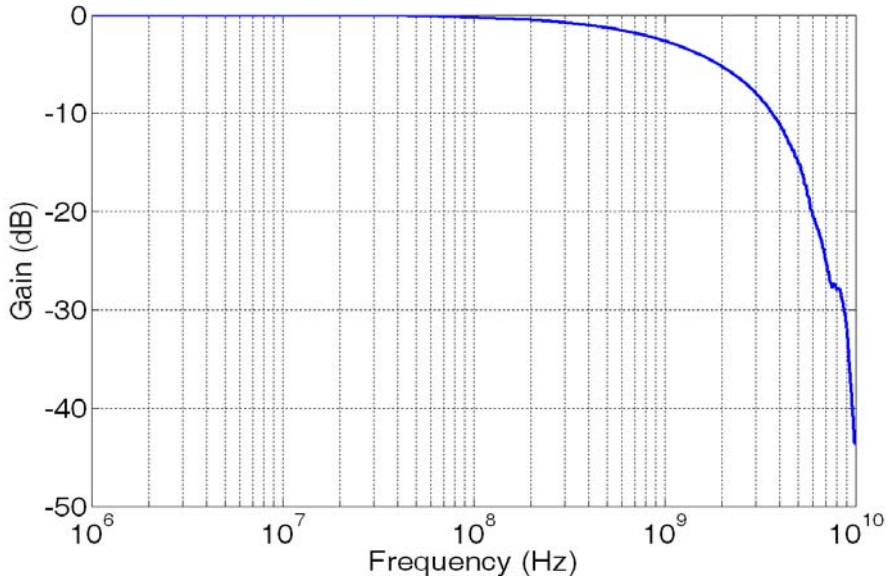


Figure 2.13 Frequency response of 6-inch backplane channel.

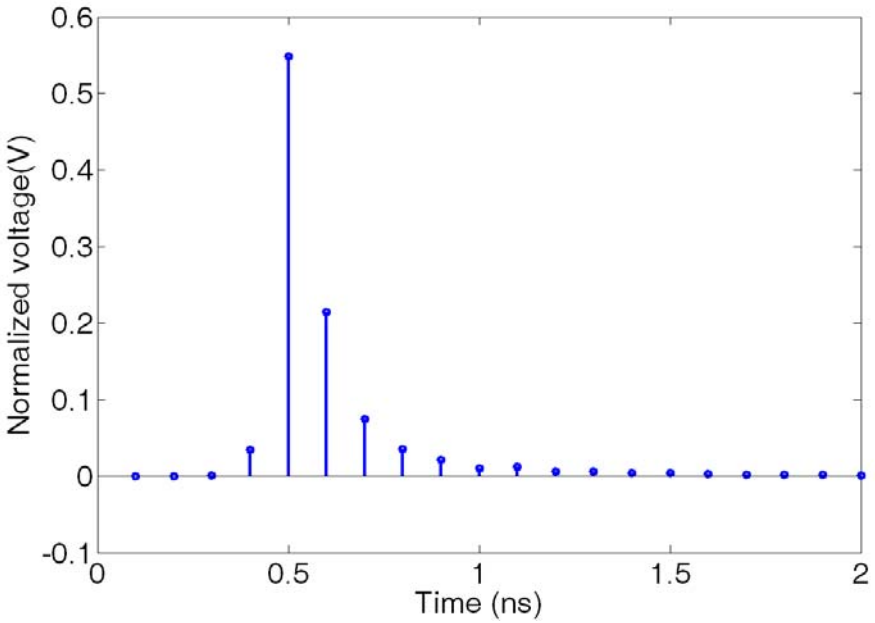


Figure 2.14 Impulse response of 6-inch backplane channel.

guideline to design our equalizer.

Although having the channel model in MATLAB, we still have to create the model for HSPICE. With the help of the built-in field solver in HSPICE, a set of RLGC elements can be calculated according to the frequency response and the material properties. The channel model in HSPICE is called a W-element transmission line. HSPICE models the skin effect resistance of the conductor and the dielectric

conductance of the insulator using the following equations [19],

$$R(f) = R_0 + R_S \cdot \sqrt{f} \ , \tag{2.17}$$

$$G(f) = G_0 + G_D \cdot f \ . \tag{2.18}$$

where R_0 is the DC resistance in O/m, R_s is the skin effect parameter in O/m$\sqrt{\text{Hz}}$, G_0 is the DC conductance of the dielectric in S/m, and G_D is the dielectric loss parameter in S/m·Hz.

Fig. 2.15 shows the frequency response of channel model in HSPICE. Because the channel model created in HSPICE needs to fit the model in MATLAB, we have to compare the difference between both. Fig. 2.16 illustrates errors in frequency response between the two models. Obviously, it shows the accuracy is within ± 1dB between 1MHz and 5GHz. Hence, the channel model in HSPICE is accurate enough for us to do the equalization design.

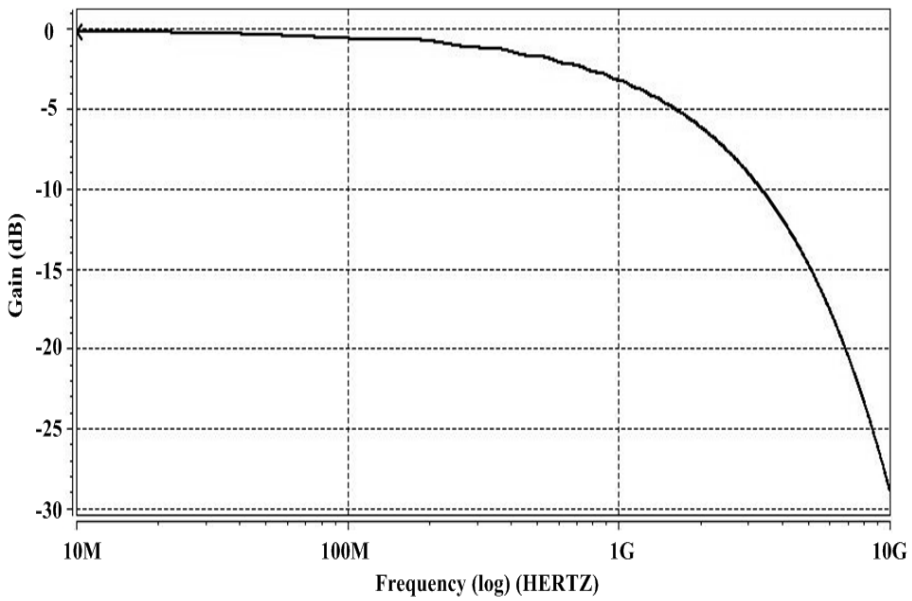


Figure 2.15 Frequency response of channel model in HSPICE.

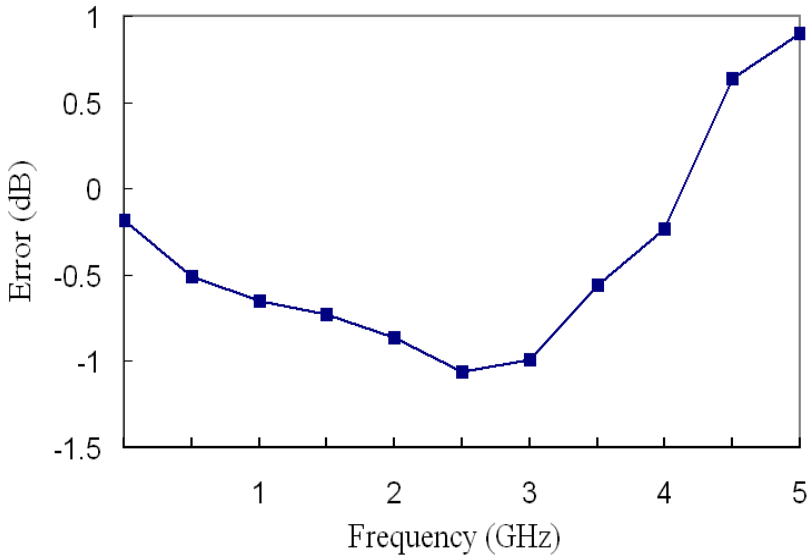


Figure 2.16 Error in curve fitting between MATLAB and HSPICE.

Chapter 3
Equalization Basis

Equalization is a kind of signal processing. The equalizer deals with the data received from the channel output and then transfers the equalized data to the circuit after it. Equalization method can be categorized into two types depending on the domain of signal processing: a continuous-time equalizer and a discrete-time equalizer. In Section 3.1 and 3.2, we will give a brief introduction of these two categories and focus on the decision-feedback equalizer. In Section 3.3, we will introduce an adaptive algorithm: sign-sign LMS algorithm. This algorithm will play an important role in the design of our equalizer system.

3.1 Continuous-Time Equalization

The continuous-time equalizers do equalization without the timing information. The signal processed by continuous-time equalizer is not digitized. They always do equalization in the frequency domain. Since the characteristic of channel is usually a low-pass filter type, continuous-time equalizers are like high-pass filters to compensate or to equalize the frequency response of the channel. Hence, we can view the continuous-time equalizer as a high-pass filter. This equalizer has a trend of increasing gain in high frequency to compensate the gain loss of the channel. Using continuous-time equalizer between the channel and receiver, we expect the high pass response can just compensate the loss of channel at high frequency to flat or to

equalize the total effective response. If we can not equalize the response, at least we can make frequency band nearly flat. Fig. 3.1 roughly illustrates the goal of this kind of equalizer.

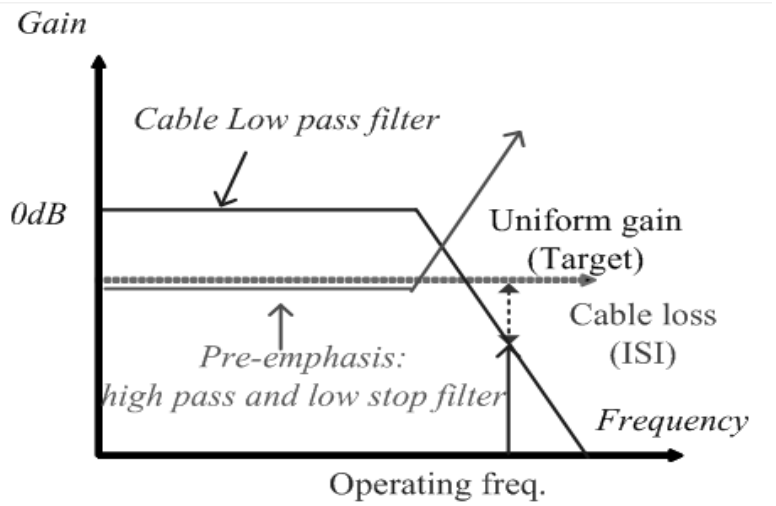


Figure 3.1 Channel response and equalizer response.

However, any circuit has its poles and zeros. That means to reach infinite high gain at the infinite high frequency is impossible. Gain amount of frequency response will fall after an limited frequency range. Fig. 3.2 illustrates the practical frequency response of continuous-time equalizers. In this figure, we use piece-wise to sketch a roughly Bode plot. We assume there is a zero at a relative low frequency in the equalizer circuit. The response will increase in 20 dB/dec when there is a zero. The

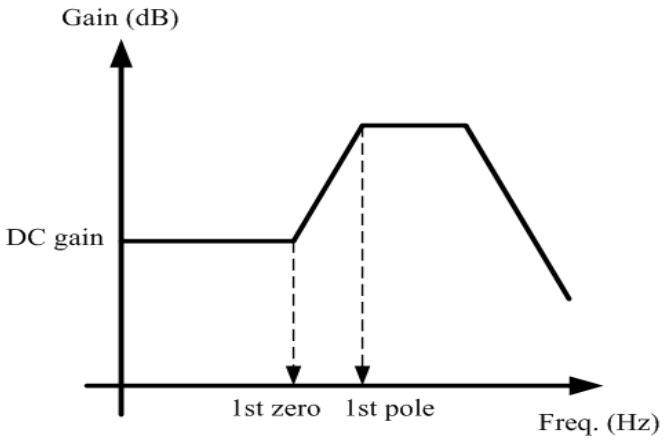


Figure 3.2 Frequency response of continuous-time equalizers.

gain keeps rising with the frequency increasing until reaching the first pole. The first pole cancels the rising frequency of the zero and flats the frequency response. As the frequency keeps increasing, the gain will drop dramatically due to the dominant pole introduced by the circuit itself.

Obviously, in order to compensate the loss of channel, continuous-time equalizers create the zero to produce a gain pulse in high frequency part. Allocating the first zero and first pole at proper frequency, we can move the gain pulse to the band we focus on. Although the effective response is not flat through the whole frequency, the equalizer extends the flat part toward the range of data transmission frequency.

A continuous-time equalizer is truly a simple one tap continuous-time circuit with high-frequency gain boosting transfer function that effectively flattens the channel response. The equalizer can be implemented by passive components or active components. As an example, the required frequency shaping can be achieved by a simple RC network as shown in Fig. 3.3. The resistor attenuates the low-frequency signals while the capacitor allows the high-frequency signal content, thus resulting in high frequency gain boosting. The transfer function and the pole zero frequencies are given by:

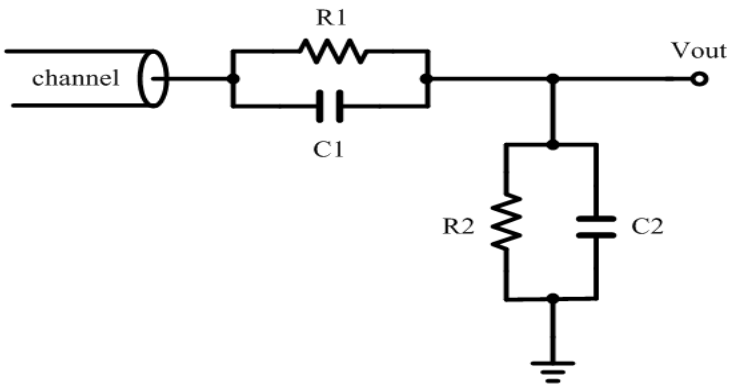


Figure 3.3 Continuous-time passive equalizer.

$$H(s) = \frac{R_2}{R_1 + R_2} \frac{1 + sR_1C_1}{1 + \frac{R_1R_2}{R_1 + R_2}(C_1 + C_2)s} \tag{3.1}$$

$$w_z = \frac{1}{R_1C_1} \tag{3.2}$$

$$w_p = \frac{1}{\frac{R_1R_2}{R_1 + R_2}(C_1 + C_2)} \tag{3.3}$$

The gain-boost factor is proportional to the ratio of zero and pole frequency ω_p/ω_z, so reasonable amounts of equalization can be achieved by choosing appropriate component values that set the required gain-boosting. There are two main drawbacks with simple passive RC equalizers. First, the RC network introduces large impedance discontinuity at the channel and equalizer. Employing inductors for impedance matching networks can be used to prevent the discontinuity. However, the large inductors make this approach less suitable for on-chip integration. Second, this method can not improve SNR since equalization is performed by attenuating low-frequency signal spectrum. Due to these reasons, this technique has limited use in high-speed serial links.

It is desirable to have a gain greater than one at all frequencies to maximize the benefit from receiver-side equalization. Therefore, equalizers using active circuit elements rather than passive components are required to achieve gains greater than one. Since parallel RC combination introduces a zero in the transfer function, it is possible to degenerate the transistors in a differential pair such that their effective transconductance increases at high frequencies [6][7][20]. Shown in Fig. 3.4, such an arrangement employs both capacitive and resistive degeneration. We express the equivalent transconductance as

$$G_m = \frac{g_m}{1 + g_m(\frac{R_S}{2} \| \frac{1}{2C_S s})} = \frac{g_m(R_S C_S s + 1)}{R_S C_S s + 1 + g_m R_S / 2} \tag{3.4}$$

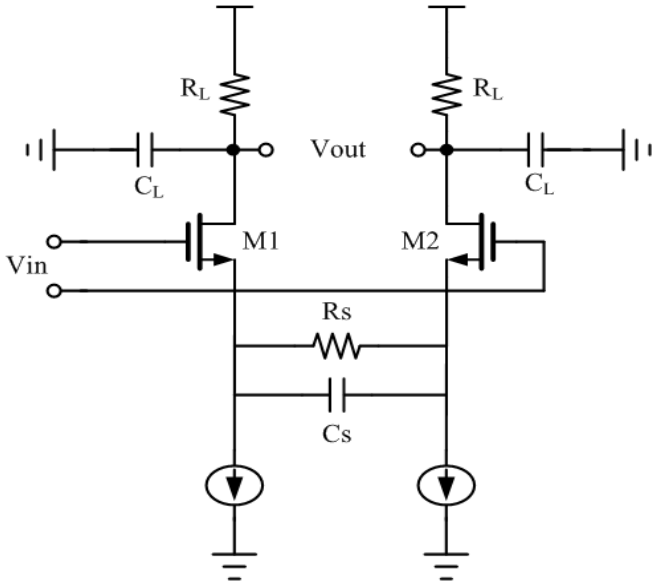


Figure 3.4 Continuous-time equalizer using capacitive degeneration.

where g_m is the MOS (M1,M2) tranconductance.

Hence,

$$w_z = \frac{1}{R_S C_S} \tag{3.5}$$

$$w_{p1} = \frac{1 + g_m R_S / 2}{R_S C_S} \tag{3.6}$$

$$w_{p2} = \frac{1}{R_L C_L} \tag{3.7}$$

$$DC\_gain = \frac{g_m R_L}{1 + g_m R_S / 2} \tag{3.8}$$

By designing the zero frequency to be lower than the dominant pole, considerable high frequency gain boosting can be achieved and it will looks like the frequency response in Fig. 3.2. However, the maximum gain boosting achieved by this method is limited by the bandwidth of the amplifier due to the load capacitance. There are several other broadband techniques for equalizing filters proposed, like inductive peaking [20], Cherry Hopper amplifier [7], or combination of inductive peaking and Cherry Hopper amplifier [21]. The goal of these circuits is to widely extend the effective gain boosting in higher frequencies.

Unlike discrete-time equalizers (we will discuss in the Section 3.2), which need

sampling clock to perform equalization, continuous-time equalizers just provide high-frequency boost to equalize the band. However, the gain-peaking transfer function of the equalizer amplifies the high frequency noise that degrades the noise margin. Moreover, the adaptation mechanism of continuous-time equalizers is more complex than that of discrete-time ones. These are disadvantages of continuous-time equalizers.

3.2 Discrete-Time Equalization

Discrete-time signal processing is for signals that are defined only at discrete points in time. The processing contains the concept of timing index and the order of data becomes a key parameter in operation. Discrete-time equalizer has the same idea of equalizing signals in discrete data points. However, the essential meaning of the discrete-time equalizers is to cancel the intersymbol-interference (ISI) induced by imperfect characteristics of channels at the discrete data points. The discrete-time equalizers can be categorized into two depending on canceling different parts of ISI: a feed-forward equalizer and a decision-feedback equalizer. We will discuss these two kinds of discrete-time equalizers in the followings.

3.2.1 Pre-and Post-Cursor ISI

Before introducing discrete-time equalizers, we investigate ISI again but this time we will look ISI in different point of view. We have known ISI can be thought of as the effect of past and future symbols on the current symbols. Assume that the impulse response of channel is h(t) and the transmitted signal is x(t), and the signal in

the channel output, y(t), is the convolution of x(t) and h(t), y(t)=x(t)*h(t). Hence, the data received at timing index n can be expressed as

$$y[n] = x[n]h[n] + \sum_{\substack{m=-\infty \\ m\neq 0}}^{\infty} x[m]h[n-m] = x[n]h[n] + ISI = \hat{y}[n] + ISI \tag{3.9}$$

We can write y[n] in another form:

$$\begin{aligned} y[n] = c_{-m}\hat{y}[n+m] + \ldots + c_{-2}\hat{y}[n+2] + c_{-1}\hat{y}[n+1] + c_0\hat{y}[n] \\ + c_1\hat{y}[n-1] + c_2\hat{y}[n-2] + \ldots c_k\hat{y}[n-k] \end{aligned} \tag{3.10}$$

The equation shows the current symbol y[n] (with some factor c_0) along with some factor ($c_m \ldots c_k$) of k past symbols and m future symbols respectively. The roll-off of the rising edge of a pulse can be treated to have an effect on the next bit (bit on the left in a pulse train) and is called pre-cursor ISI. The roll-off of the falling edge of the pulse can be treated to have an effect on the previous bit (bit to right in pulse train) and is called post-cursor ISI. Fig. 3.5 illustrates a discrete-time representation of an impulse response with pre- and post-cursor ISI. In a practical system, it is reasonable to assume that the ISI affects a finite number of symbols. Hence we can assume c_n=0 for n<-m and n>k, where m and k are finite, positive integers.

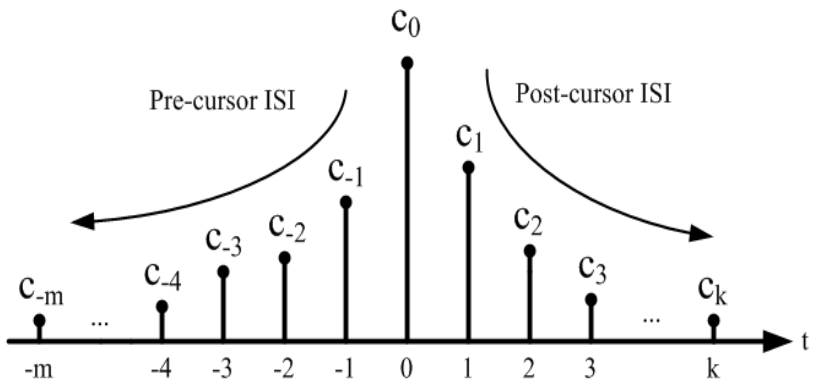


Figure 3.5 Discrete-time representation of pre- and post-cursor ISI.

3.2.2 Feed-Forward Equalizer

The simplest discrete-time equalization technique that has been used over the years is linear feed-forward equalization. Feed-forward equalization typically involves the use of a linear transversal finite impulse response (FIR) filter as shown in Fig. 3.6. Here we use a 3-tap FFE as an example. The FIR filter consists of adjustable tap coefficients, C_{-1}, C_0, C_1, with a time delay, D between adjacent taps. The output of the FFE is a summation of the input signal with delayed versions of itself. Depending on the relative values of C_{-1}, C_0, and C_1, the FFE can be used to cancel pre- and post-cursor ISI, only pre-cursor ISI, or only post-cursor ISI. The tap coefficients C_{-1}, C_0, and C_1 are calculated using algorithms designed to meet certain system criteria. We can write the equation of output $\hat{y}_D$ of the FFE as:

$$\hat{y}_D[n] = C_{-1}y[n] + C_0y[n-1] + C_1y[n-2] \tag{3.11}$$

where y[n] is the output signal of channel. Since y[n] can be expressed as x[n]*h[n], where x[n] is the transmitted signal and h[n] is the impulse response of channel, EQ. 3.11 can be re-written as:

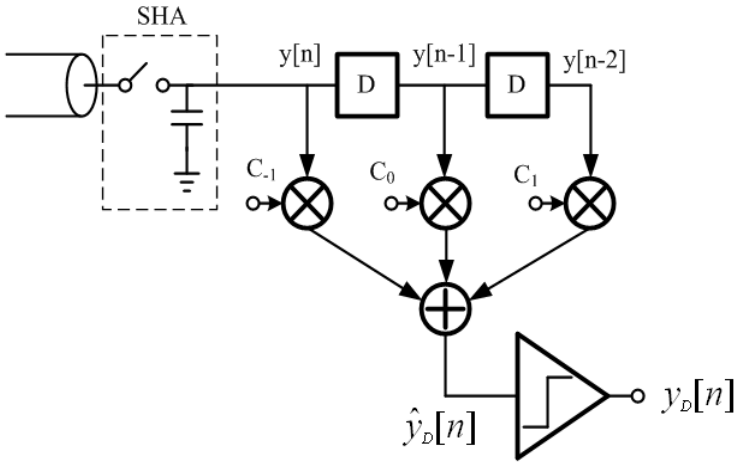


Figure 3.6 A linear feed-forward equalizer (FFE) topology.

$$\hat{Y}_D(z) = X(z)H(z)(C_{-1} + C_0 z^{-1} + C_1 z^{-2}) \tag{3.12}$$

Since $C_{-1} + C_0 z^{-1} + C_1 z^{-2}$ is the z transform response of equalizer, we write it as E(z). Hence, if the output $\hat{y}_D[n]$ is perfectly equal to input x[n], e.g. there is no error occurred, the equation about equalizer and channel responses is

$$E(z) = \frac{1}{H(z)} \tag{3.13}$$

EQ. (3.13) is an illustration of zero-forcing (ZF) equalization. ZF equalization tries to cancel all the ISI and results in a transfer function that is the exact inverse of the channel response. While this might sounds like an ideal solution, ZF equalization implies gain in the frequency range where the channel response is small. Because channels are usually low-pass filters in nature, this type of equalizers will be high-pass response. Any additive noise in that frequency range is also amplified. So in noisy channels, the ZF-FFE can result in very poor SNR at the output.

In fact, Fig. 3.6 is an analog FIR equalizer. An analog delay chain is required to implement the analog FIR. This analog delay can be implemented using a replica delay line whose delay is locked to a delay locked loop or a phase locked loop operating at data rate [8]. However, there are some difficulties occurred in the FIR analog equalizer. First, the settling time of the front-end sample-and-hold (SHA) limits the overall operating speed. Second, the sampled signal experiences attenuation due to the limited bandwidth of the delay elements. Finally at high data rates, the precise generation of analog delay elements consumes large power. One alternative way to generate the analog delay is by using multi-phase ($\phi_1 \sim \phi_n$) clocks. By paralleling several data path, the circuit can slow down its operation frequency. The conceptual block diagram of a parallelized architecture is shown in Fig. 3.7.

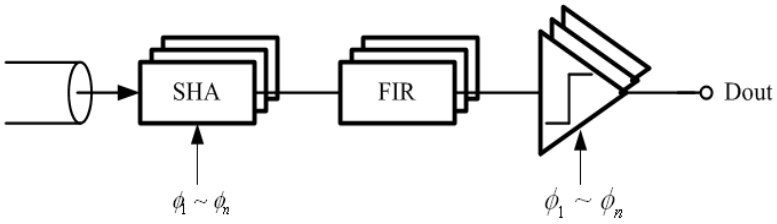


Figure 3.7 A parallelized analog FIR equalizer.

FFE has been shown to be effective on channels where the ISI is not severe. As mentioned earlier, the FFE amplifies noise on channels, resulting in poor SNR. Therefore, non-linear equalizers such as the decision-feedback equalizer (DFE) have been explored and we will discuss it in the next sub-section.

3.2.3 Decision-Feedback Equalizer

Decision-feedback equalization [22] is a non-linear equalization that employs previous decisions to eliminate the ISI caused by previous symbols on the current symbol. It was first introduced by M. E. Austin in 1967, who introduced a decision-theory approach to solve the problem of digital communication over known dispersive channels. This work was the first to describe a method to utilize the information of past decisions to make corrections to current symbols and thereby cancel post-cursor ISI. Today the DFE is used extensively to combat ISI in different dispersive channels and finds many applications in high-speed communication systems.

A typical decision-feedback equalizer is shown in Fig. 3.8. Here we use one tap of DFE as an example. The signal, $\hat{y}_D$, is the summation of received data and the previous decision bit multiplied by a constant C_1. The slicer is a decision-making circuit that compares y_D and a reference V_{ref} and then generates a binary signal. If y_D

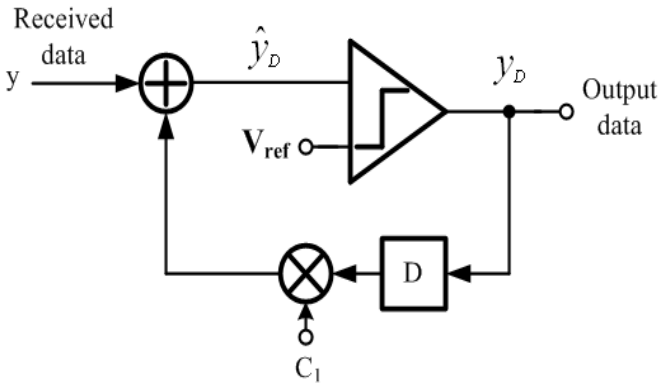


Figure 3.8 Typical block diagram of DFE with one tap.

$\geqq V_{ref}$, the output of slicer is ONE and it is ZERO when $y_D < V_{ref}$.

Now we will explain why the DFE can cancel the post-cursor ISI. We can write the transfer function for the simple one-tap DFE as

$$\hat{y}_D[n] = y[n] + C_1 \,\mathrm{sgn}(\hat{y}_D[n-1]) \tag{3.14}$$

where y[n] is the signal level being received, $\hat{y}_D[n]$ is the DFE corrected signal, C_1 is the first DFE feedback tap weight, and sgn() produces a 1 for $\hat{y}_D[n-1] \geqq 0$ and -1 otherwise. For the case of zero-error detection, EQ. (3.14) can be analyzed in a linear sense by replacing sgn ($\hat{y}_D[n-1]$) with the transmitted data x[n-1] and y[n] with the convolution of sequence x[n] with the channel response h[n]. The resulting difference equation and corresponding z transform are

$$\begin{aligned} \hat{y}_D[n] &= (x[n] * h[n]) + C_1 x[n-1] \\ \hat{Y}_D(z) &= X(z)(H(z) + C_1 z^{-1}) \end{aligned} \tag{3.15}$$

To illustrate the operation of the DFE, assume a low-pass channel-response function

$$H(z) = h[0] + h[1]z^{-1} \tag{3.16}$$

with h[0] normalized to 1 and h[1] positive, resulting in reduced gain at high frequencies. In this case

$$\hat{Y}_D(z) = X(z)(1 + h[1]z^{-1} + C_1 z^{-1}) \tag{3.17}$$

To cancel the post-cursor ISI, C_1 is set to the value –h[1], which realizes a fully

equalized channel $\hat{Y}_D(z) = X(z)$.

The biggest bottleneck of the simple DFE is its operation speed. Before the next bit comes, the feedback signal must be ready at the input of adder. This implies that the total propagation delay consumed by the slicer, the flip-flop, multiplier, setup time of flip-flop and some operating margin, needs to be less than one symbol period. This loop is a critical path in the DFE. As data rate increases to several multi-Gbps, it seems that the critical path delay can not meet the stringent timing constraint. For example, if the data rate is 10 Gb/s, the total latency of the critical path must be less than 100 ps. Unless the propagation delay of silicon processing elements improves greatly and overcomes this difficulty, high data rates will limit the use of this direct type of DFE.

In order to overcome the feedback loop latency challenge imposed by the limitations of the clocked topology, designers have come up with some novel design techniques to implement multi-Gbps DFEs in standard CMOS. A common approach to reduce the critical path delay is to use a "look-ahead" architecture, also referred to as loop-unrolled DFE or speculative DFE [23]. The architecture is shown in Fig. 3.9. The basic concept behind this technique is that for a NRZ signal, every symbol is a 1 or a -1. The two threshold value V_1 and V_0 are the first tap weight, C_1, multiplied by 1 and -1, respectively. Instead of feeding back the slicer decision for the first tap, a look-ahead DFE makes two decisions with two slicers where each slicer assumes the previous bit is a -1 and 1. The received data value is selected from these two slicer outputs based on the previous data value with a multiplexer. The n^{th} decision y_D can be expressed as

$$y_D[n] = y^1{}_D[n]y_D[n-1] + y^0{}_D[n]\overline{y_D[n-1]} \tag{3.18}$$

Similarly, the $(n-1)^{th}$ decision can be written as

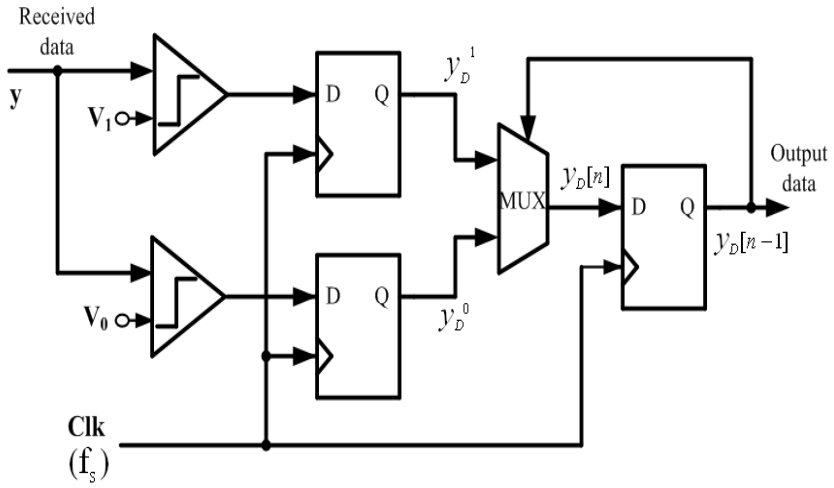


Figure 3.9 Look-ahead one-tap DFE.

$$y_D[n-1] = y^1{}_D[n-1]y_D[n-2] + y^0{}_D[n-1]\overline{y_D[n-2]} \tag{3.19}$$

Substituting EQ. (3.19) in EQ. (3.18), a_n can be expressed as

$$\begin{aligned} y_D[n] &= y^1{}_D[n]\left(y^1{}_D[n-1]y_D[n-2] + y^0{}_D[n-1]\overline{y_D[n-2]}\right) \\ &\quad + y^0{}_D[n]\left(\overline{y^1{}_D[n-1]y_D[n-2] + y^0{}_D[n-1]\overline{y_D[n-2]}}\right) \\ &= \left(y^1{}_D[n]y^1{}_D[n-1] + y^0{}_D[n]\overline{y^1{}_D[n-1]}\right)y_D[n-2] \\ &\quad + \left(y^1{}_D[n]y^0{}_D[n-1] + y^0{}_D[n]\overline{y^0{}_D[n-1]}\right)\overline{y_D[n-2]} \\ &= f_1[n]y_D[n-2] + f_2[n]\overline{y_D[n-2]} \end{aligned} \tag{3.20}$$

EQ. 3.20 shows that $y_D[n]$ is now a function of $y_D[n-2]$ and not $y_D[n-1]$. This implies that the critical timing path has been extended from T_{bit} to $2T_{bit}$, where T_{bit} is one bit period. The extra delay available can be used to reduce the clock rate to half the data rate. The half-rate architecture of look-ahead DFE is shown in Fig. 3.10. Obviously, this approach has the advantage of lower clock rate but the hardware and area increase tremendously.

The look-ahead technique is typically limited to only one tap because of an exponential increase in the number of slicers with the number of taps. As a result, the second and higher order taps of the DFE do not apply look-ahead and are often fed back directly. We call this dynamic feedback technique [9][10]. A lot of creative

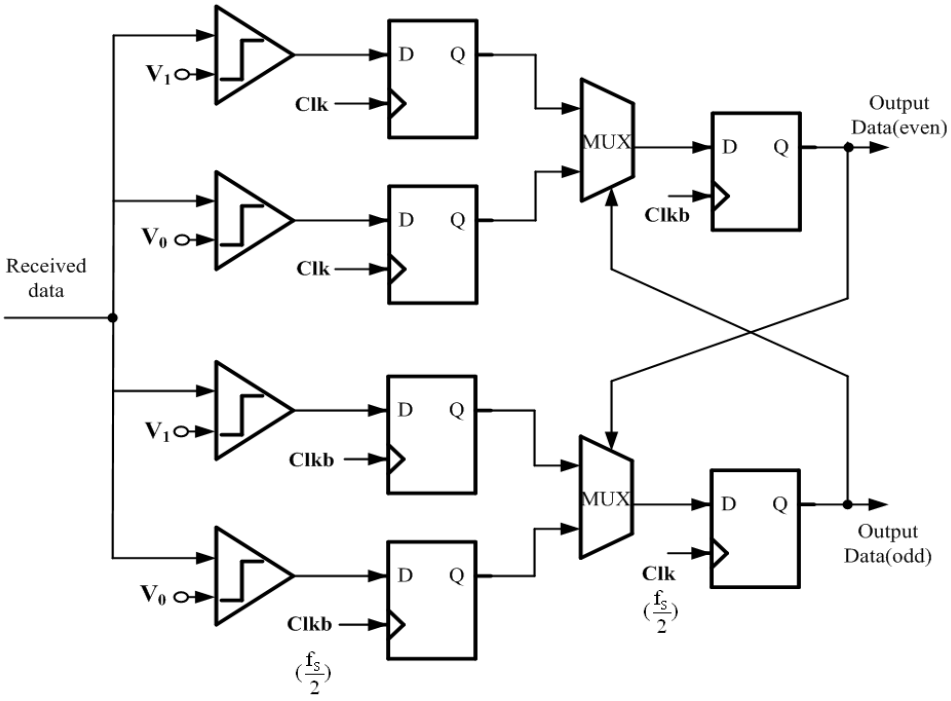


Figure 3.10 Half-rate look-ahead DFE.

inventions about dealing the timing constraint, higher data rates and lower power consumption, have been proposed [24][25].

The problem of noise enhancement can be completely eliminated by using DFE since the DFE just utilizes the previous decision to do the equalization without boosting the high-frequency noise. There are two design issues with the DFE design. First, the effectiveness of ISI cancellation is based on the assumption that all previous decisions are correct and therefore if decisions are incorrect, the ISI will be worse. The problem is referred to as error propagation. However, in the case of serial-links with required BER < 10^{-12}, error propagation does not degrade the performance. Second, the DFE can cancel only post-cursor ISI. If the channel response is very severe, including pre- and post-cursor ISI, a separate FFE is required.

3.3 Sign-Sign LMS Algorithm

The sign-sign least-mean-square (sign-sign LMS) algorithm is a simplification of least-mean-square (LMS) algorithm [26]. In this section, we will give a short derivation of LMS algorithm and point out the relationship between LMS and sign-sign LMS algorithms. Finally, the concept of adaptive equalizer will be described.

First, we introduce the method of steepest descent. Fig. 3.11 shows a N-tap FIR Wiener filter. In the figure, d[n] is the desired response or the correct output that we expect. We define four parameters as follows:

tap-weight vector: $W=[W_0\ W_1 \ldots W_{N-1}]^T$,

signal input: $X[n]=[x[n]\ x[n-1] \ldots x[n-N+1]]^T$,

filter output: $y[n]=W^T X[n]$,

error signal: $e[n]=d[n]-y[n]$.

The cost function is defined as the mean-square error

$$F_C=E[e^2[n]]=E[d^2[n]]-2W^T P+W^T RW$$

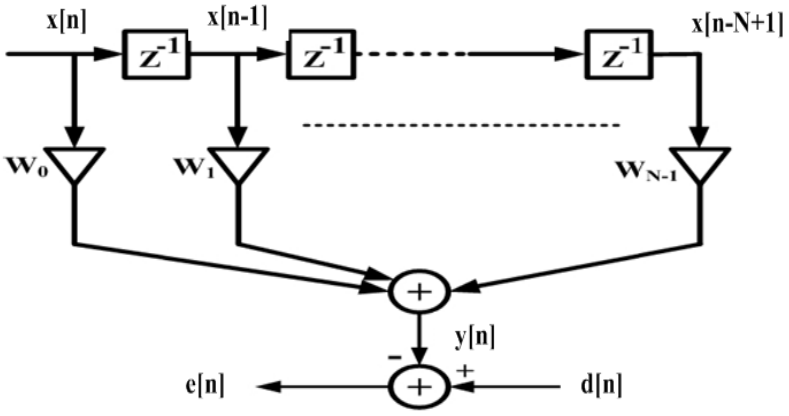


Figure 3.11 N-tap FIR Wiener filter.

where $R=E\left[X[n]X^T[n]\right]$ is autocorrelation of the filter input. F_C has a single global minimum obtained by solving the Wiener-Hopf equation

$$RW_{opt} = P$$

if R and P are available, where W_{opt} is the optimal tap-weight vector. However, we can use iterative method rather than solving the Wiener-Hopf equation directly. Starting with an initial guess for W_{opt}, say W[0], a recursive search method that may require many iterations to converge to W_{opt} is used.

The method of steepest-descent is a gradient-based method. Using the initial or present guess, we compute the gradient vector and evaluate the next guess of the tap-weight vector by making a change in the initial guess in a direction opposite to that of the gradient vector. Here, we define the gradient of F_C as

$$\nabla F_C = 2RW - 2P \tag{3.21}$$

With an initial guess of W[0] at n=0 the tap-weight vector at the k-th iteration is denoted as W[k]. We can write down the recursive equation that is used to update W[k]:

$$W[k+1] = W[k] - \mu \nabla_k F_C \tag{3.22}$$

where $\mu > 0$ is called the step-size, $\nabla_k F_C$ denotes the gradient vector ∇_k calculated at the point $W_{opt.}=W[k]$.

If we substitute EQ. (3.21) into EQ. (3.22), we get

$$W[k+1] = W[k] - 2\mu(RW[k] - P) \tag{3.23}$$

By EQ. (3.23), W[k] will converge to the optimum solution W_{opt} and the convergence speed is dependent on the step-size μ.

The LMS algorithm is a stochastic implementation of the steepest-descent algorithm. It simply replaces the cost function $F_C = E\left[e^2[n]\right]$ by its instantaneous

estimate $\hat{F}_C = e^2[n]$. By substituting the simplified cost function into EQ. (3.22), we obtain

$$W[n+1] = W[n] - \mu \nabla e^2[n] \tag{3.24}$$

where $\nabla = \left[\frac{\partial}{\partial W_0} \quad \frac{\partial}{\partial W_1} \quad \ldots \quad \frac{\partial}{\partial W_{N-1}} \right]^T$. We can expand the i-th element of the gradient vector $\nabla e^2[n]$ as

$$\frac{\partial e^2[n]}{\partial W_i} = 2e[n]\frac{\partial [n]}{\partial W_i} = -2e[n]\frac{\partial y[n]}{\partial W_i} = -2e[n]x[n-i]$$

then, we will get

$$\nabla e^2[n] = -2e[n]X[n] \tag{3.25}$$

where $X[n] = \left[x[n] \quad x[n-1] \quad \ldots \quad x[n-N+1] \right]^T$.

Finally, we can get the LMS algorithm by substituting EQ. (3.25) into EQ. (3.24)

$$W[n+1] = W[n] + \mu \cdot e[n] \cdot X[n] \tag{3.26}$$

Here, we merge the constant 2 into the step size μ.

EQ. (3.26) describes the relationship between the new tap weight and the current tap weight of LMS algorithm. The sign-sign LMS algorithm is a simplicity of the LMS algorithm. Using the sign of e(n) and X[n] instead of the actual value of them simplifies the tap weight calculation. The sign-sign LMS algorithm is

$$W[n+1] = W[n] + \mu \cdot sign\left(e[n]\right) \cdot sign\left(X[n]\right) \tag{3.27}$$

where the sign() is a function that gets the sign of input. Since sign-sign LMS algorithm is a simplicity of the LMS algorithm, there is a shortcoming about sign-sign LMS algorithm. When coefficients converge, the error in sign-sign LMS algorithm will be larger than that in LMS algorithm. However, the values of sign error and data imply that the complexity of circuit design can be largely reduced.

In a practical transmission system, the exact channel characteristics are not

known a priori, and they may vary due to the variations of materials or environment. Hence, the equalizer coefficients need to update adaptively. The conceptual block diagram illustrating the operation of an adaptive one-tap DFE is shown in Fig. 3.12 [27]. The coefficient update processor uses an algorithm like LMS or sign-sign LMS to automatically adjust the coefficients by measuring the equalizer performance so as to improve the performance on an average.

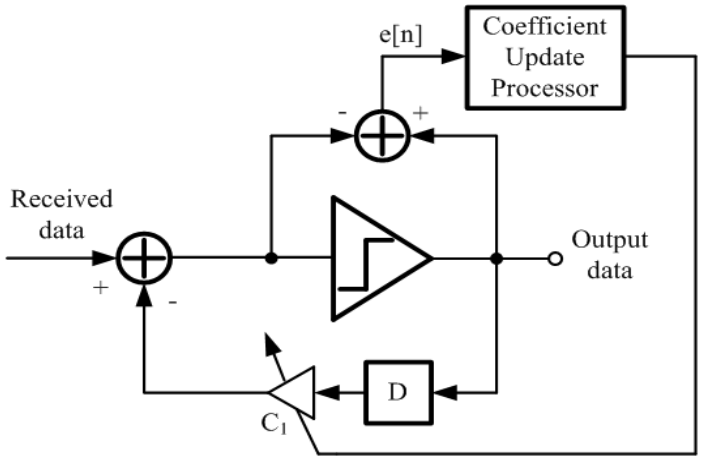


Figure 3.12 An adaptive one-tap DFE.

Chapter 4
Discrete-Time Adaptive Decision-Feedback Equalizer

In Chapter 2 and Chapter 3, we have generally introduced the basic concepts, channel model, and the classification of equalizers. In this Chapter, we will demonstrate our proposed architecture of equalizer and explain the behaviors of these blocks. Finally, some behavioral simulation results will be presented.

4.1 DT-ADFE Architecture

Chapter 3 has illustrated the category of receiver equalizers depending on the domain of signal processing: a continuous-time equalizer and a discrete-time equalizer. Since continuous-time equalizers introduce noise enhancement in high frequency band and adaptive method in continuous-time equalizers is difficult to implement, we choose our equalizer type as discrete-time version.

Fig. 4.1 is the block diagram of discrete-time adaptive decision feedback equalizer. We will discuss in details of the proposed architecture. As depicted in Fig. 4.1, the architecture is a typical model of ADFE. Discrete-time ADFE needs a clock to sample the data and make the decision. The clock rate and then the bit rate depends on the feedback latency. If the latency is more than one bit period in a full-rate architecture, it must be considered to lower the clock rate and employ look-ahead architecture and so on. The discrete-time equalizer has benefits in speed as it can

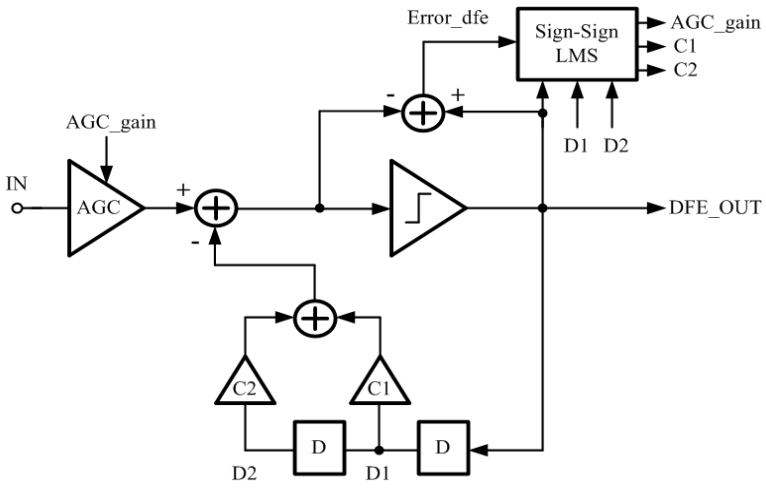


Figure 4.1 DT-ADFE architecture.

exploit parallelism. Since our date rate is set as 10 Gb/s, our clock rate is 10GHz. Here we assume the latency within the first tap in the feedback loop is less than one bit time and it will be proved in Chapter 5.

Fig. 4.1 also shows that there are two taps in our DFE model. Actually, we need to examine the channel impulse response in advance and then decide how many taps enough to cancel dominant post-cursor ISI. From Section 2.3, we know that in our channel model the first two post-cursor ISI dominate the all post-cursor ISI and hence two taps are adequate in our equalizer design.

Our DT-ADFE architecture is composed of the following blocks: automatic gain control(AGC), adder or subtractor, slicer, delay element, multiplier, and sign-sign LMS engine. We have known adder or subtractor, slicer, delay element and multiplier are essential blocks in DFE design. Because we need to design an adaptive DFE, the equalizer coefficients are set adaptively, and an algorithm must be included to help the coefficients to converge. An important advantage of using the sign-sign LMS algorithm is the simplicity of implementation for the multiplication operation. Moreover, the function of AGC is to adjust the value of input data to approach the logical level, 1 or -1. We can view AGC as a gain controller which multiplies the

input data with an adaptive scalar gain. For example, if an input data level is 0.5 and the gain of AGC is 2, then the output is 1. AGC can help coefficients to converge successfully. Without the AGC, the DT-ADFE could not work normally.

4.2 Mixed-Signal Integrator with Hopping Update Scheme

4.2.1 Hopping Coefficients Update Scheme

Section 4.1 has introduced overall blocks in our DT-ADFE architecture. Among these blocks, the performance of sign-sign LMS engine has the dominant effect to the quality of the DT-ADFE. Thus, if we can come up with some novel methods to improve one aspect of the performance metrics, it will make a difference in our equalizer design.

The equation of power consumed by a digital circuit can be expressed as:

$$P = C \cdot VDD^2 \cdot f \cdot p_t \tag{4.1}$$

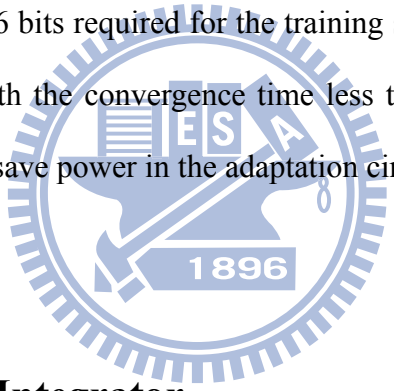
where C is the output loading capacitance, VDD is the voltage of power supply, f is the circuit operation frequency, p_t is the probability of switching activity.

Typical adaptive mechanism for adaptive equalizer is to update the weights of coefficients per bit of data. This method takes each information into consideration and the coefficients of equalizer converge to stable state in a very short time. However, this means the adaptation circuit in the equalizer needs to operate at a frequency as high as the data rate. With the increasing of data rate, this mechanism will produce a large power consumption for adaptation circuit.

Obviously, we can save power consumption by slowing down the clock rate in

the adaptation circuit. In fact, the transmitted data are often independent between each other and we do not need to worry about losing the correlation information if we ignore some bits. From the theory of sign-sign LMS algorithm, we know that the convergence is only dependent on the step size μ. Therefore, we can choose some bits from the received data as a new received sequence for finding the characteristics of the channel. For example, we can choose the bits with time indexes that are equal to a multiple of eight as a new sequence to update the coefficients of DFE. This method is called "hopping" [28].

Although the convergence time of coefficients will increase when we use the hopping scheme, this is not a problem in high-speed transmission. In the specification of USB 3.0, there are 65536 bits required for the training sequence. Thus, we can use hopping update scheme with the convergence time less than the required condition. Since hopping scheme can save power in the adaptation circuit, it is worth to do in our equalizer design.



4.2.2 Mixed-Signal Integrator

Section 3.3 has talked about sign-sign LMS algorithm. The sign-sign LMS coefficient updating algorithm is given by:

$$C_k[n+1] = C_k[n] + \mu \cdot sign\left(e[n]\right) \cdot sign\left(y_D[n-k]\right) \tag{4.2}$$

where C_k [n+1] represents k^{th} coefficient at (n+1) update, μ is the step size, e[n] is the error in the equalizer output, and y_D is the decision of ADFE slicer output. Since the product of $sign\left(e[n]\right) \times sign\left(y_D[n-k]\right)$ is always a logical value 1 or -1, the coefficients always add a "μ" or subtract a "μ". From the view of circuit design, the operation is like to count up or count down and sends the information to a mechanism

for transferring it into a relative analog value which is corresponding to the step size μ. Obviously, we need a counter that could count up and count down. We also need a circuit that could receive the information from the counter and then generate a voltage (or current) represents the coefficient. In fact, the implementation of sign-sign LMS algorithm is equivalent to a discrete-time integrator. Fig. 4.2 shows the equivalent discrete-time integrators [29]. In a traditional mixed-signal DFE implementation, a

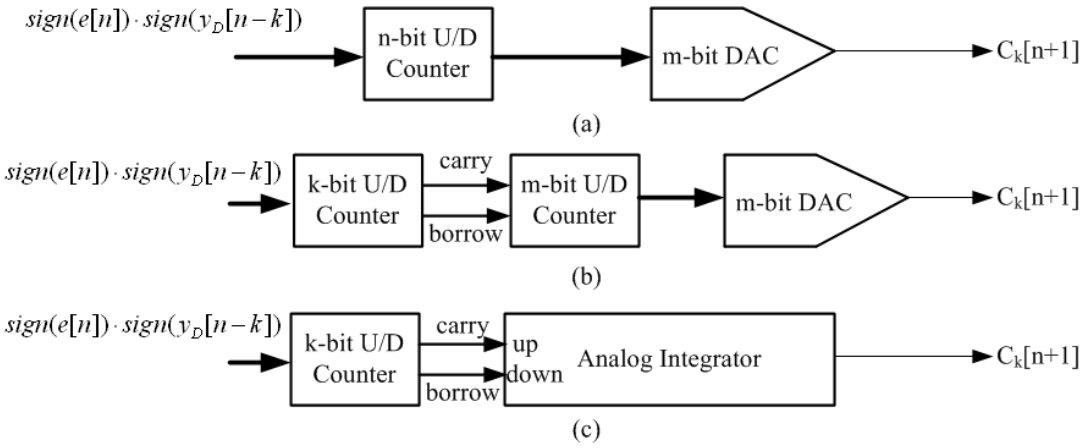


Figure 4.2 Discrete-time integrators, (a) n-bit counter and m-bit DAC. (b) cascaded counters and m-bit DAC. (c) k-bit counter and analog integrator.

n-bit Up/Down counter followed by a m-bit DAC is used to build the integrator, where n>m, as shown in Fig. 4.2(a). The number of n is usually large such as 10 and the number of m is the most significant bits of n. This arrangement is set to realize the small value of step size μ while ease the difficulty of DAC design. The integer input is a binary signal with value ±1. This signal determines whether the counter should increment or decrement its current value according to EQ. (4.2). The m-bit DAC generates an output voltage (or current) based on the m most significant bits of the n-bit counter. This output voltage is the analog coefficient $C_k[n+1]$. The n-bit counter can be realized as a cascade of two smaller counters: a k-bit Up/Down counter followed a m-bit Up/Down counter, where k+m=n, as shown in Fig. 4.2(b). For example, n is equal to 10, k is equal to 4 and m is equal to 6. When the 4-bit pre-counter overflows, it counts from positive full-scale (+7) to 0 and outputs *Carry*

=1, causing the 6-bit counter to increment by 1. When it underflows, it counts from negative full-scale (-7) to 0 and outputs *Borrow* =1, causing the 6-bit counter to decrement by 1. Otherwise, the 6-bit counter remains unchanged. The advantage of Fig. 4.2(b) over Fig. 4.2(a) is that the input of DAC does not need to change often since the k-bit up/down counter will reset to 0 when it overflows or underflows. In an adaptive equalizer system, we hope the coefficients will be stable when the coefficients converge. When the step size is small enough, the stable coefficients will lead to low MSE and BER. Section 4.3 will show the simulation results.

If there is an analog discrete-time integrator that could replace the m-bit counter and m-bit DAC, the power dissipation and die area will be greatly saved, as shown in Fig. 4.2(c).

An smart implementation of this analog integrator is a charge pump. Fig. 4.3 shows the mixed-signal integrator which consists of a 4-bit Up/Down counter and a charge pump. Coefficient $C_k[n+1]$ changes when either switch closes, as controlled by *Carry* and *Borrow*. When *Carry* and *Borrow* are both low, loading capacitor C_L keeps the coefficient constant. An equation about step size μ and charge pump voltage is

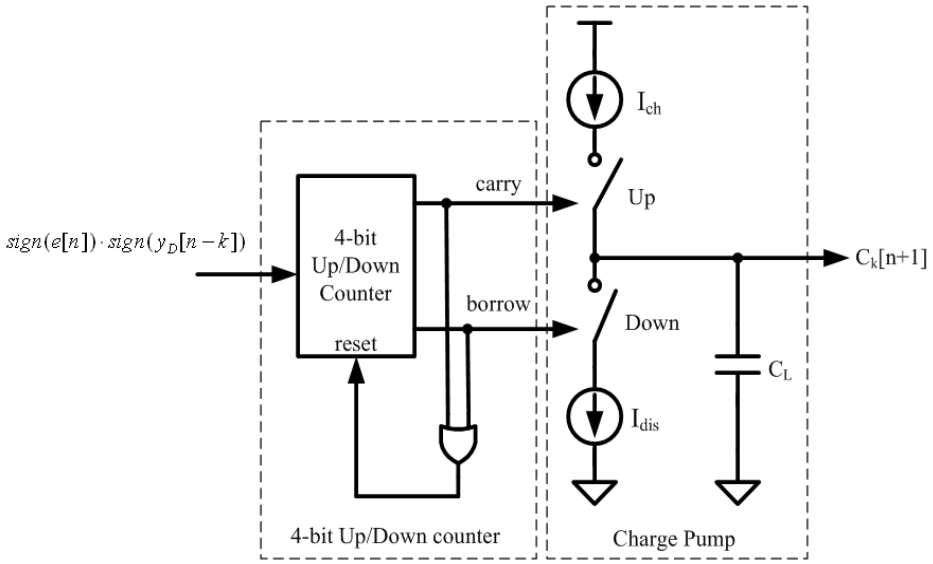


Figure 4.3 Mixed-signal integrator.

$$\mu \propto \Delta V = \frac{I \cdot \Delta t}{C_L} \tag{4.3}$$

where I is the average current of charge pump, Δt is a turn-on time. If the charge current matches the discharge current, we could combine EQ. (4.2) and EQ. (4.3) into EQ. (4.4):

$$C_k[n+1] = C_k[n] + (Carry - Borrow)\frac{I \cdot \Delta t}{C_L} \tag{4.4}$$

where the result of (*Carry-Borrow*) can be either +1, -1, or 0. In fact, having the Up/Down counter, we can alleviate the stringent design of charge pump and a precise voltage at the charge pump output is not a requirement. Since an adaptive equalizer will monitor the error, if there is a small current offset in the charge pump, the equalizer system will automatically correct the accumulation of current offset in a long run. When the counter reaches its maximum value (+7) or minimum value (-7), the counter is reset to its mid-scale value, 0. As a result, the charge pump will spend most of the time with both switches open. Either the Up or Down switch can be enabled at most once every 7 clock cycles. Hence, the problem of current mismatch is not a big issue in charge pump design.

In high-speed circuit implementation, it is difficult to update the coefficients of the equalizer as the same as symbol rate. If we add the hopping coefficients update scheme to the mixed-signal integrator, sign-sign LMS algorithm will be replaced by delayed sign-sign LMS algorithm. The delay sign-sign LMS algorithm is given by

$$C_k[n+1] = C_k[n] + \mu \cdot sign\left(e[n-D]\right) \cdot sign\left(y_D[n-k-D]\right) \tag{4.5}$$

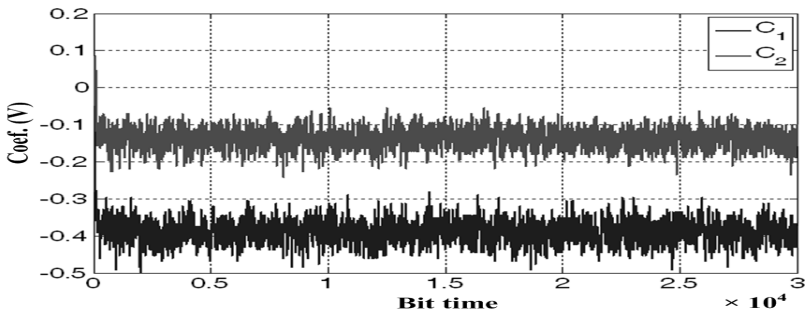
where D is the number of delay bits time. Since hopping scheme slows down the clock rate in the adaptation circuit, a sequence to adapt the coefficients is per D bits of original received data. Therefore, the current coefficient is adapted by the error and decision terms with D bits before. We call this method as “delayed sign-sign LMS with hopping update scheme.” We will employ this algorithm in our proposed

DT-ADFE design.

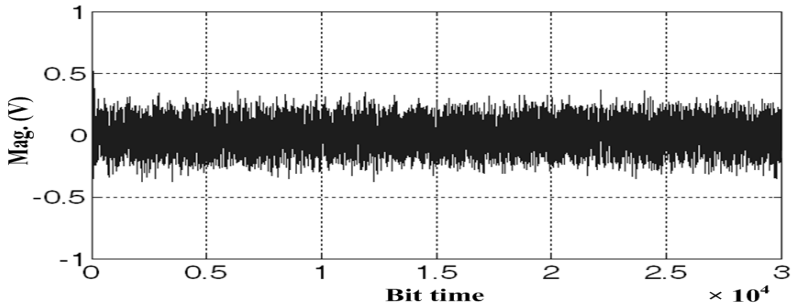
4.3 Behavioral Simulation Results

Section 4.1 and 4.2 have described our DT-ADFE architecture and the "delayed sign-sign LMS with hopping update scheme". In this section, we will simulate our equalizer by MATLAB under different conditions and investigate the performance from the behavioral results.

In all the simulations, a 10 Gb/s pseudo random binary sequence (PRBS) is generated and passed through the channel presented in Section 2.3. We choose the step size μ as 2^{-7} that is small enough to make the sign-sign LMS algorithm converge. Our simulations take three different conditions into consideration: hopping update scheme, hopping update scheme with 3-bit Up/Down counter, and hopping update scheme with 4-bit Up/Down counter. The hopping update scheme uses the operation frequency of 1, 1/4, 1/8, 1/16 of the data rate. Fig. 4.4 shows the coefficients and the amount of error with time of the equalizer with data rate updating. The error is defined as the value difference between slicer input and slicer output. From the plot of the coefficients, we can observe the convergence situation of equalizer weights. Fig. 4.5 shows the simulation results with quarter of data rate adaptation. The results of hopping with eighth and sixteenth of data rate adaptation are shown in Fig. 4.6 and Fig. 4.7, respectively.

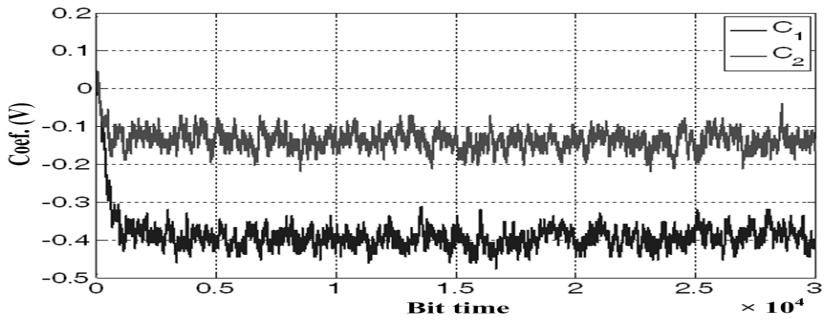


(a)

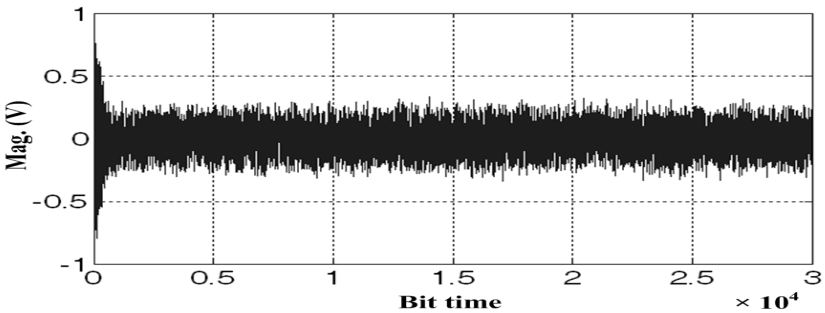


(b)

Figure 4.4 (a) DFE coefficients and (b) error of hopping update scheme with data rate.

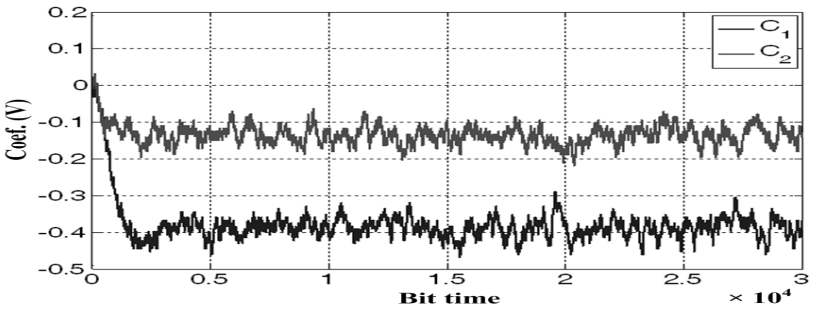


(a)

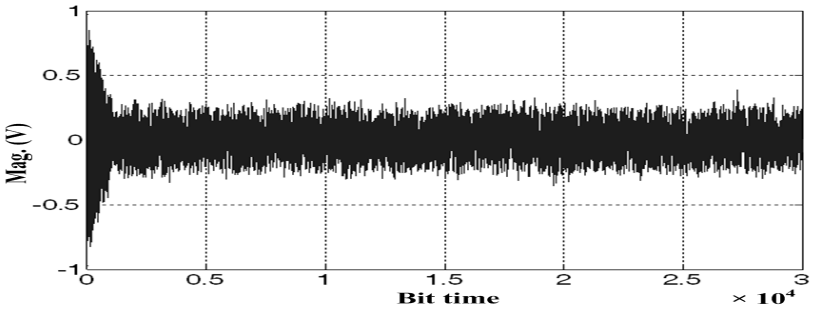


(b)

Figure 4.5 (a) DFE coefficients and (b) error of hopping update scheme with 1/4 data rate.

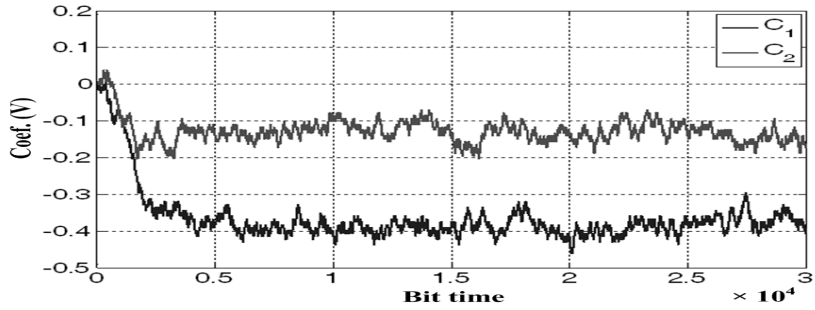


(a)

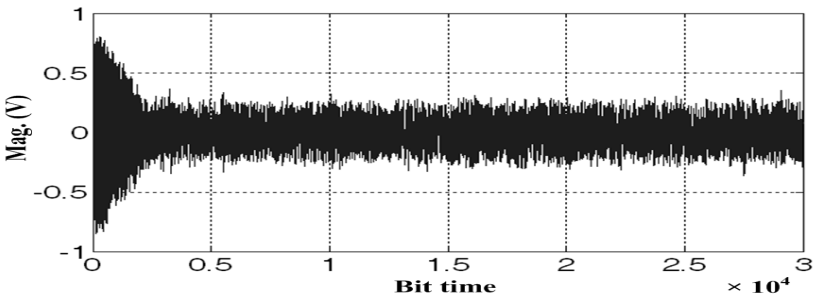


(b)

Figure 4.6 (a) DFE coefficients and (b) error of hopping update scheme with 1/8 data rate.



(a)



(b)

Figure 4.7 (a) DFE coefficients and (b) error of hopping update scheme with 1/16 data rate.

From Fig. 4.4 to Fig. 4.7, we can find the convergence time increases with the reduction of the adaptation frequency. This phenomenon is what we expected in section 4.2. Table 4.1 lists the mean square errors (MSE) and convergence times of these four different update frequencies. It is very interesting that MSE increases with lower adaptation frequency as the convergence time does. Therefore, lower power consumption is at the cost of higher MSE and slower convergence time.

Table 4.1 MSE and convergence time for hopping update scheme at four different frequencies.

adaptation frequency (data rate)	1	1/4	1/8	1/16
MSE	0.01232	0.01252	0.01263	0.01268
Convergence time (bit)	300	1450	1850	2230

Fig. 4.8 shows the histogram of error of hopping update scheme with data rate when coefficients converge. If the samples are many enough, the histogram of error will be like Gaussian distribution. The standard deviation of error is 0.111. Hence, the seven times of standard deviation is 0.777. Since 0.777 is less than 1, the BER of the equalizer is less than 10^{-12} according to Table 2.1.

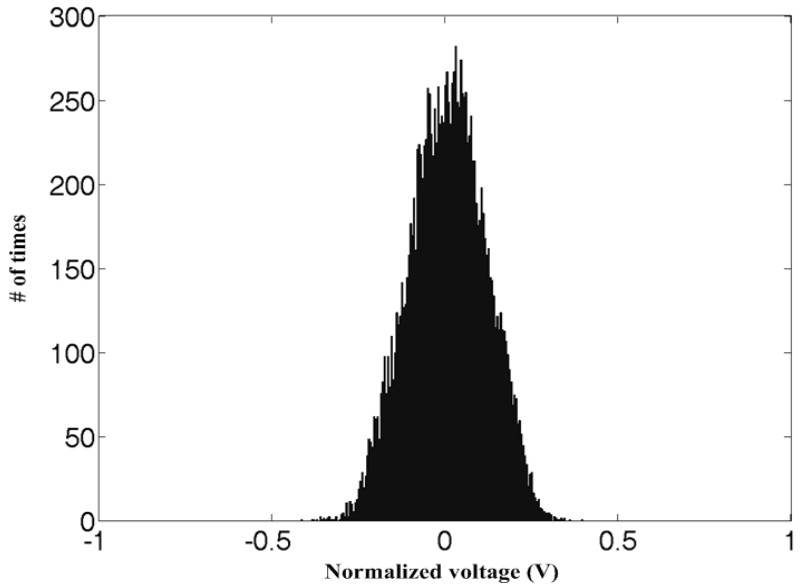
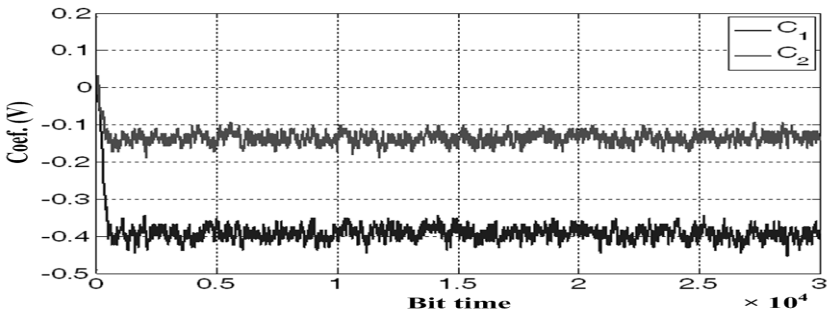
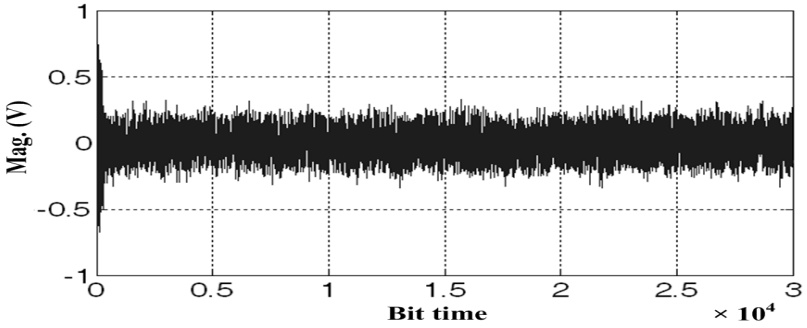


Figure 4.8 Histogram of error.

We have seen the results about the effect of hopping update scheme. Now, we add an Up/Down counter in the adaptation block to observe the performance. In simulation, we simulate the behavior of an Up/Down counter. For example, if the Up/Down counter has 3 bits, the positive full-scale is 3 and the negative full-scale is -3. When the output of counter is 3 and the next product of error sign and decision sign comes a **1**, the counter overflows and returns to 0, and the coefficients update. Also, when the output of counter is -3 and the next product of error sign and decision sign comes a **-1**, the counter underflows and returns to 0, and the coefficients update. If the bit counts of the Up/Down counter is 4-bit, then the maximum is 7 and minimum is -7 and the behavior is the same as a 3-bit counter. From Fig. 4.9 to Fig. 4.12, the simulation results of 3-bit Up/Down counter with hopping update scheme at 1, 1/4, 1/8, 1/16 date rate are shown, respectively. We list the MSE and convergence time of these four cases in Table 4.2.

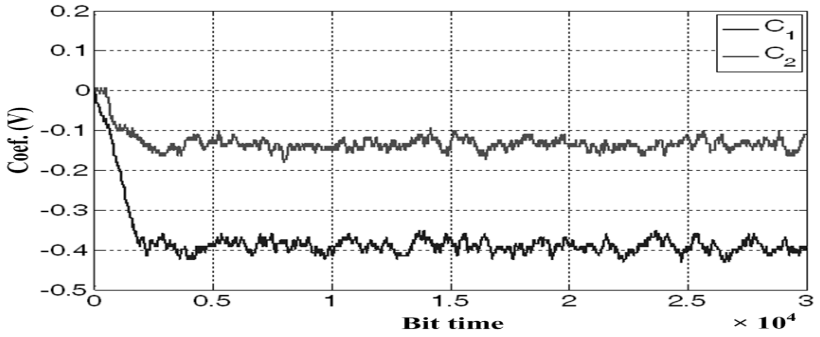


(a)

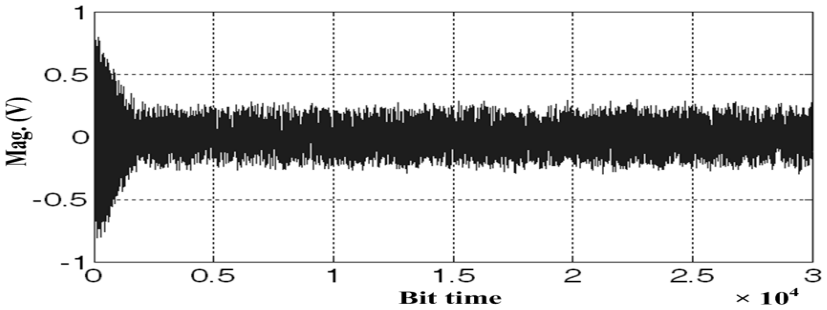


(b)

Figure 4.9 (a) DFE coefficients and (b) error of hopping update scheme with data rate and 3-bit up/down counter.

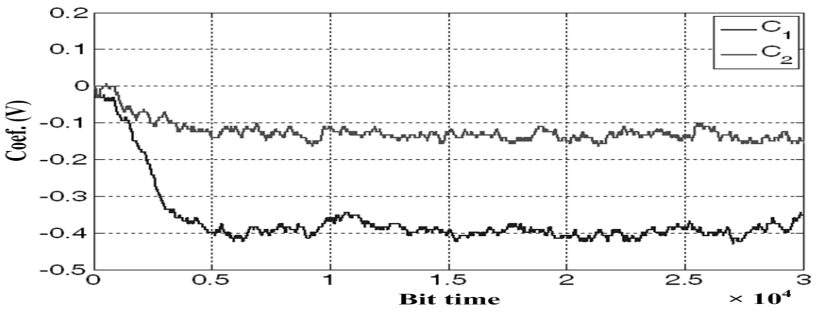


(a)

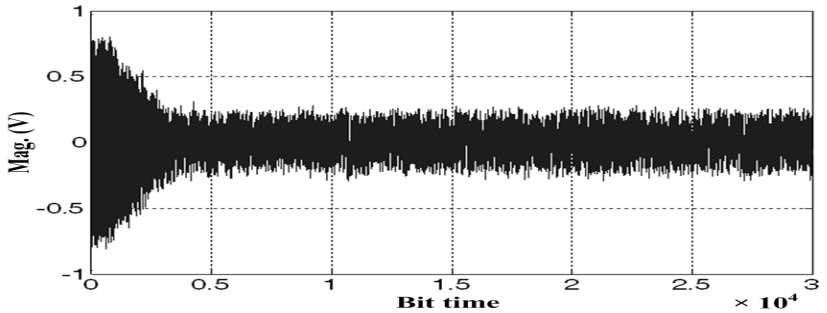


(b)

Figure 4.10 (a) DFE coefficients and (b) error of hopping update scheme with 1/4 data rate and 3-bit up/down counter.

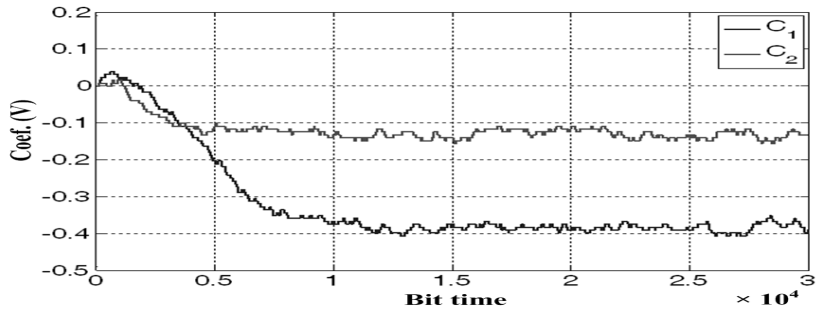


(a)

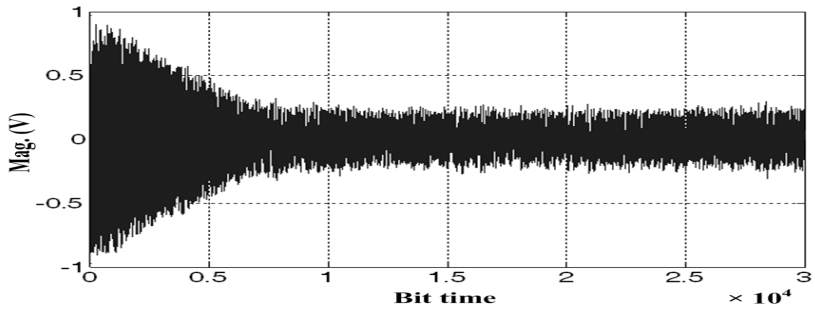


(b)

Figure 4.11 (a) DFE coefficients and (b) error of hopping update scheme with 1/8 data rate and 3-bit up/down counter.



(a)



(b)

Figure 4.12 (a) DFE coefficients and (b) error of hopping update scheme with 1/16 data rate and 3-bit up/down counter.

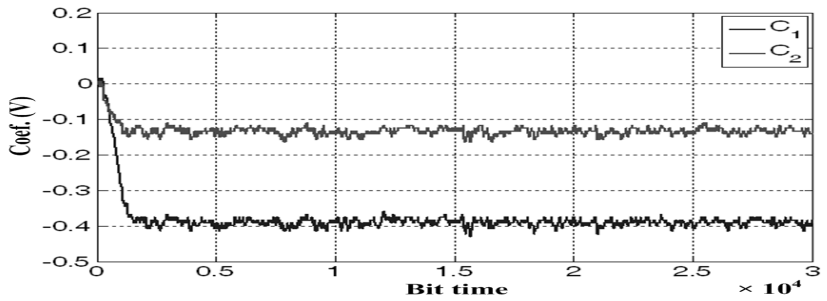
Table 4.2 MSE and convergence time for 3-bit Up/Down counter with hopping update scheme at four different frequencies.

adaptation frequency (data rate)	1	1/4	1/8	1/16
MSE	0.01143	0.01140	0.01121	0.01132
Convergence time (bit)	650	2300	4130	8200

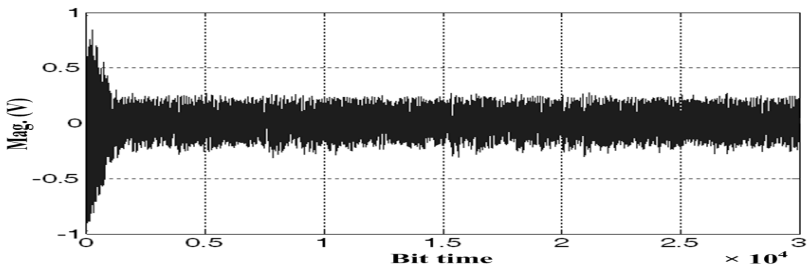
We can find that the convergence times are longer than those of only hopping update scheme. This is easy to be anticipated. Moreover, the MSE is lower as compared with that of only hopping update scheme. Because when the coefficients converge, they will be stable with the same value at most of the time, the variance of error changes steadily and the MSE reduces. However, with Up/Down counter the MSE does not increase as adaptation frequency decreases. Since the coefficients converge, the Up/Down counter is like a barrier to keep the coefficients stable, and the adaptation frequency becomes less impact to influence the change of coefficients.

In other words, Up/Down counter dominates the magnitude of MSE. Lower MSE implies bigger eye diagram and lower BER. From this point of view, the addition of Up/Down counter is beneficial to coefficient adaptation circuit design.

The coefficients and error results of equalizer with 4-bit Up/Down counter and hopping update scheme at four different updating frequencies are shown in Fig. 4.13 to Fig. 4.16. MSE and convergence time of these four cases are listed in Table 4.3. Again, we can find the convergence times are longer than those in Table 4.2, but the MSE is lower than that of with 3-bit Up/Down counter. The convergence times are all smaller than 65536 bits time of USB 3.0 specification. The information provides a good guideline for us to choose the suitable type of adaptation algorithms when doing performance trade-off. For the hopping coefficients update scheme, we can reduce the power consumption of coefficients update block. For Up/Down counter, we can obtain the advantage of lower MSE and alleviate the difficult design of charge pump. On this basis, we will implement our DT-ADFE in the next chapter.

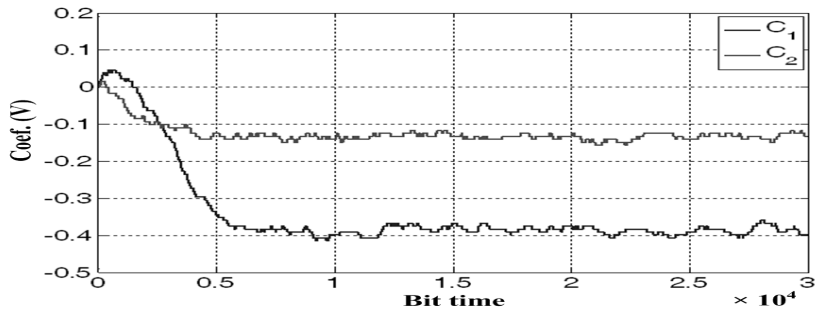


(a)

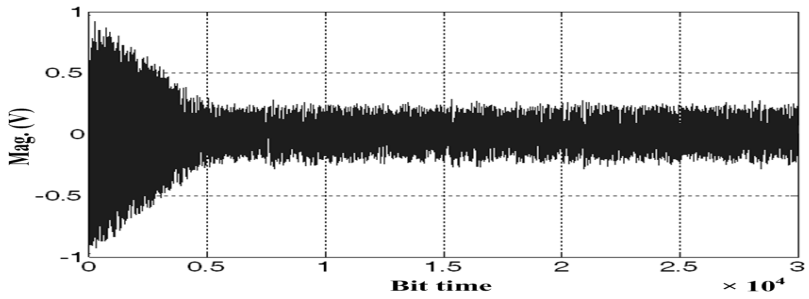


(b)

Figure 4.13 (a) DFE coefficients and (b) error of hopping update scheme with data rate and 4-bit up/down counter.

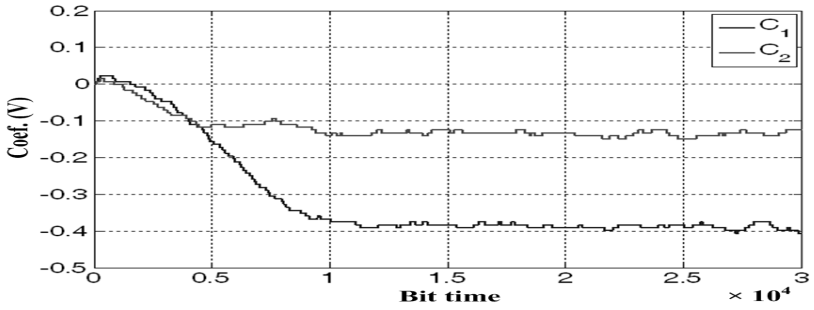


(a)

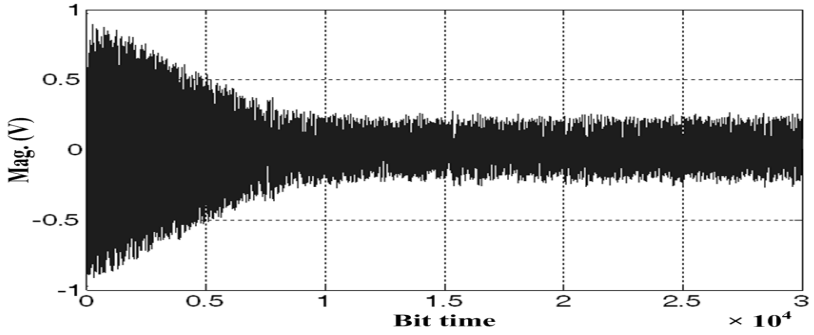


(b)

Figure 4.14 (a) DFE coefficients and (b) error of hopping update scheme with 1/4 data rate and 4-bit up/down counter.

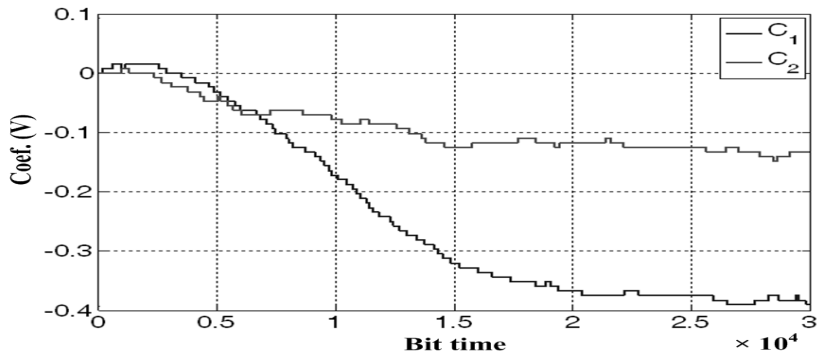


(a)

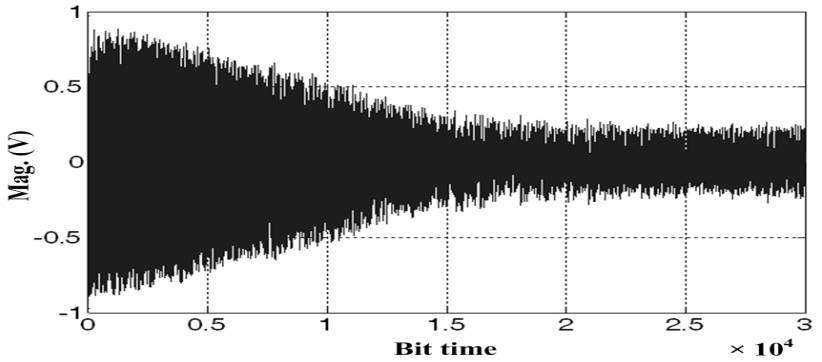


(b)

Figure 4.15 (a) DFE coefficients and (b) error of hopping update scheme with 1/8 data rate and 4-bit up/down counter.



(a)



(b)

Figure 4.16 (a) DFE coefficients and (b) error of hopping update scheme with 1/16 data rate and 4-bit up/down counter.

Table 4.3 MSE and convergence time for 4-bit Up/Down counter with hopping update scheme at four different frequencies.

adaptation frequency (data rate)	1	1/4	1/8	1/16
MSE	0.01115	0.01112	0.01097	0.01100
Convergence time (bit)	1300	5100	9720	18100

Chapter 5
A 10-Gb/s Adaptive Decision-Feedback Equalizer Implementation

In Chapter 4, we have introduced overall blocks in our equalizer. In this Chapter, we will focus on the design of all circuits. Section 5.1 will briefly discuss the overall blocks of our equalizer system. The details of our equalizer circuit design are given in Section 5.2. Equalizer simulation results and layout implementation are presented in Section 5.3. Finally, the plan for the measurement and verification of our chip environment setup is illustrated in Section 5.4.

We use a 65-nm standard process with regular threshold voltage CMOS technology for circuit design. The power supply is 1.2V. The minimum channel length of a MOS is 60 nm. In typical case, the threshold voltage of NMOS and PMOS is about 0.42V. The propagation delay of an inverter with a medium width is about 8 ps. These information will help us to design our circuits.

5.1 System Description

We have established the DT-ADFE system blocks in Chapter 4. Since we need to implement it in circuit level, a detailed system architecture should be built. Fig. 5.1 shows the circuit architecture of our equalizer system. The upper part is the main equalizer blocks, and the lower part is the coefficient adaptation circuits. We use

variable gain amplifier (VGA) as an automatic gain control (AGC) to adjust the swing of input signals. Node $\hat{y}_D$ is the output of DFE and it will determine the value of sign error. The sign of error signal for the sign-sign LMS algorithm is generated by slicing the equalizer output at the target ±1 levels. These slicers are set at zero level to detect the polarity of the received data. The sign-sign LMS algorithm engine runs at 1/8 times the data rate to save power. The transmission characteristics of the backplane change gradually with temperature and therefore, the adaptation algorithm can run slowly.

Fig. 5.2 is the detailed block diagram of sign-sign LMS engine. The thick lines are digital signals and the thin lines are analog signals. There are five signals needed to be sampled by 1.25 GHz: summation output ($\hat{y}_D$), D1, D2, and the two slicing outputs of target ±1 levels, $\hat{y}_D^{1}$ and $\hat{y}_D^{-1}$. These sampled five differential signals are fed into differential-to-single-end (DtoS) circuit to generate digital signals.

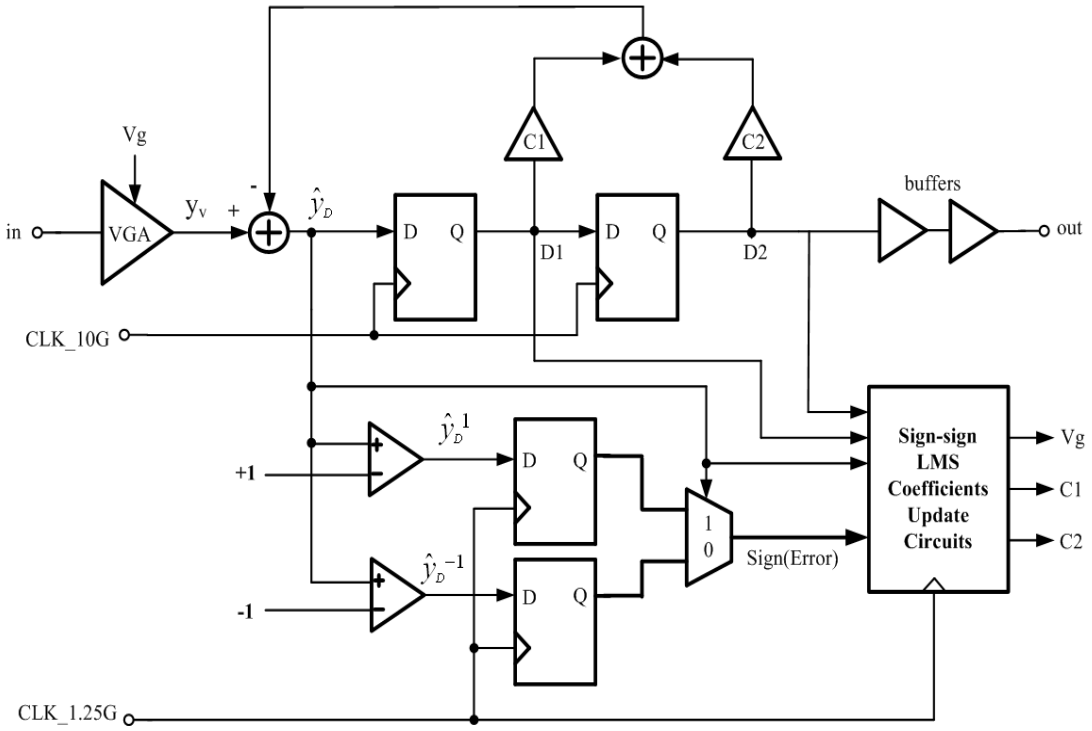


Figure 5.1 Circuit architecture of our equalizer system.

Multiplexer (MUX) and XOR logical circuits can do the operation of product of error sign and decision sign. From the results of Chapter 4, 4-bit Up/Down counter is used. Finally, the output voltages (Vg, C1, C2) of charge pumps are coefficients and are fed back to the DFE main block to do the ISI cancellation.

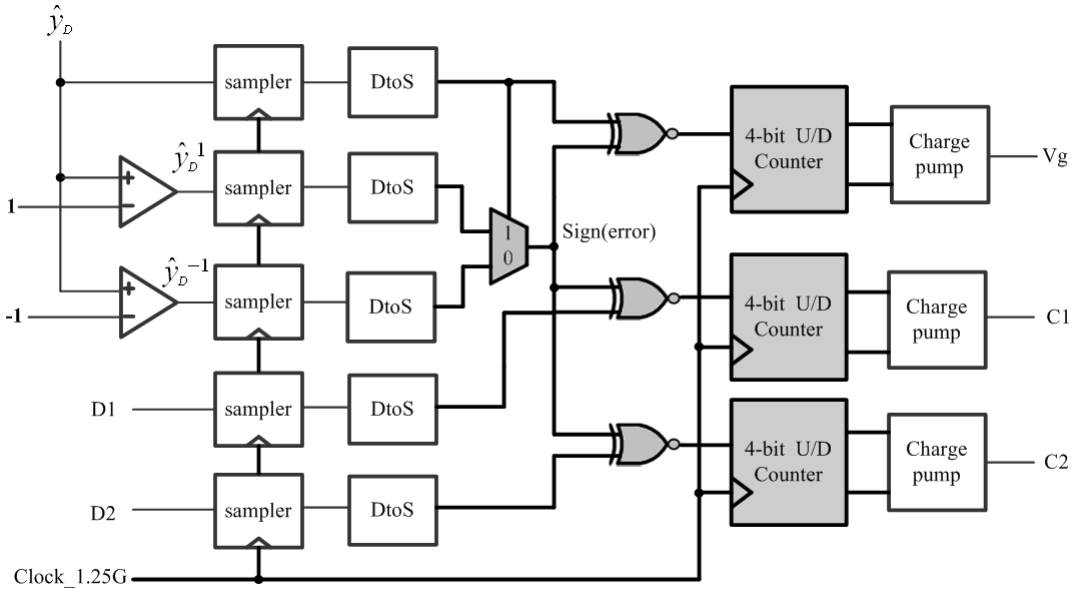


Figure 5.2 Sign-Sign LMS engine.

5.2 Circuit Design

5.2.1 Variable Gain Amplifier

The first stage of our equalizer is the variable gain amplifier (VGA). Since the transmitted signals passed through the channel are attenuated, the main function of VGA is to amplify the signal at channel output to the values in the range of the following stages. Fig. 5.3 is our VGA cell. It is a differential pair with source resistive degeneration. The NMOS Mg is inserted in the source node of the differential pair. M3 and M4 are used as current sources and Vb is the bias voltage. The half circuit of VGA cell is shown in Fig. 5.4. The C_L represents the total capacitance at the VGA

output and the next stage. Rg is the equivalent resistance of Mg. From the half circuit, we can write the equivalent transconductance as

$$G_m = \frac{g_m}{1 + g_m(\frac{R_S}{2} \| \frac{R_g}{2})}, \tag{5.1}$$

and the low frequencies voltage gain can be expressed as

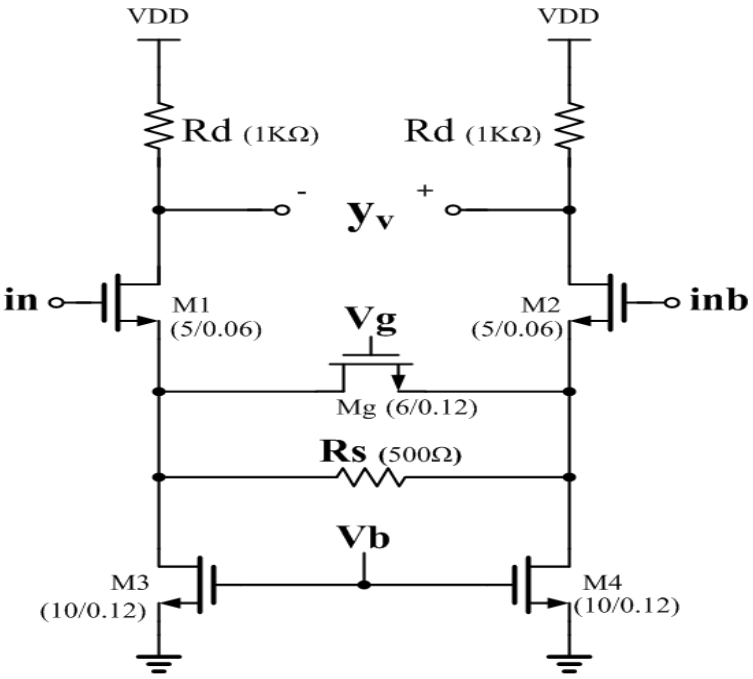


Figure 5.3 VGA cell.

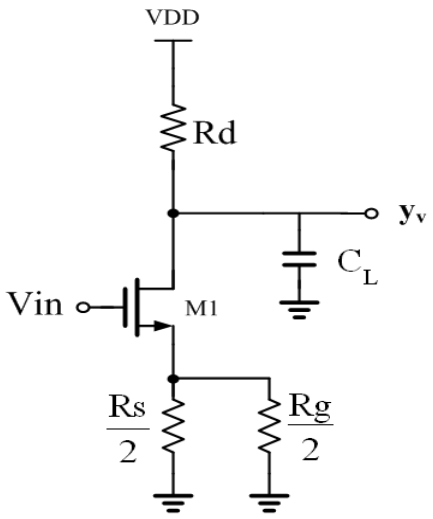


Figure 5.4 Half circuit of VGA cell.

$$Av = \frac{g_m R_d}{1 + g_m(\frac{R_S}{2} \| \frac{R_g}{2})} \approx \frac{2R_d}{R_S \| R_g}. \quad (5.2)$$

Obviously, we can vary the voltage gain by varying R_g. Once the transistor M_g conducts, it will operate in the linear region since the source and drain voltages are almost the same. The resistance of R_g can be approximately written as

$$R_g = \frac{L}{\mu_n C_{ox} W(V_{gs} - V_t)}. \quad (5.3)$$

When the gate voltage V_g changes, the equivalent resistance of M_g changes and the voltage gain changes.

Besides, the bandwidth of VGA is needed to take into consideration. Since the input data rate is 10 Gb/s, the maximum frequency of data is 5 GHz. Hence, we expect the bandwidth of VGA can accommodate 5 GHz at least. The dominant pole of the VGA cell is at the output, $1/(R_dC_L)$. Therefore, the loading resistor R_d determines the voltage gain and the bandwidth. We simulate the frequency response of cascaded 2 VGA cells with different V_g, ranging from 0.6V to 1.2V with 0.1V as step size, as shown in Fig. 5.5. When V_g is at 0.6V, the voltage gain at low frequencies is about

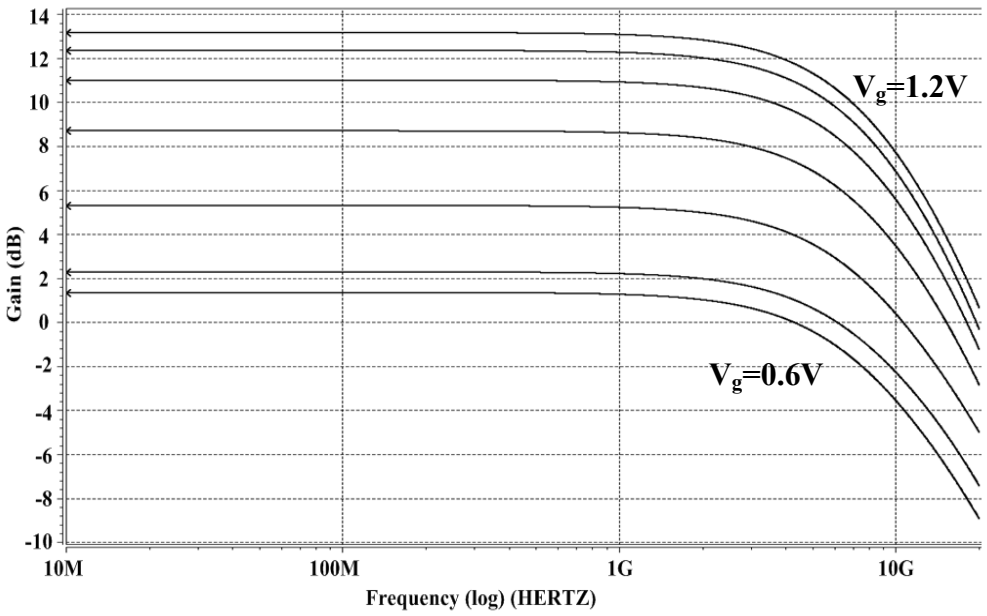


Figure 5.5 Frequency response of cascaded 2 VGA cells with different V_g.

1.36dB. When V_g is at 1.2V, the voltage gain at low frequencies is about 13.2dB. Therefore, there is 12dB gain range in total. Except the voltage gain, we can observe that the bandwidth of these situations is high enough for not corrupting the data. Consequently, the circuit is able to provide adequate gain adjustment and frequency bandwidth and is suitable for our equalizer system design.

5.2.2 Current-Mode Summer

The adder is an important element in DFE system. The task of adder is to add the input data and the feedback weighted taps for subtracting the post-cursor ISI. There are two requirements for a good adder: First, the operation of addition (or subtraction) should be accurate. Since the amplitude we dealt with is very small such as 300 mV or below 300mV, we hope the adder can do more precise addition work to not generate wrong decision. Second, the operation speed of addition (or subtraction) needs to be as rapid as possible since it is within the critical path of the DFE system. The two requirements of adder will determine the architecture and the overall performance of equalizer.

The signals at the VGA output are differential and small voltage range (about 0.7V to 1.1V single-end). Hence, it is straightforward to use differential type analog adder to do the addition. If the adder can do multiplication at the same time, it will have many advantages such as smaller area and lower power. Fig. 5.6 is this kind of adder and we call it "current-mode summer." M1 and M2 is the main differential pair and the VGA output is fed into $y_v / \overline{y_v}$. The first tap input is $D1 / \overline{D1}$, and the second tap is $D2 / \overline{D2}$. M7 is set as a current source and Vb is a fixed bias voltage. M8 and M9 are adaptive current sources which are controlled by the tap coefficients, V_{C1} and

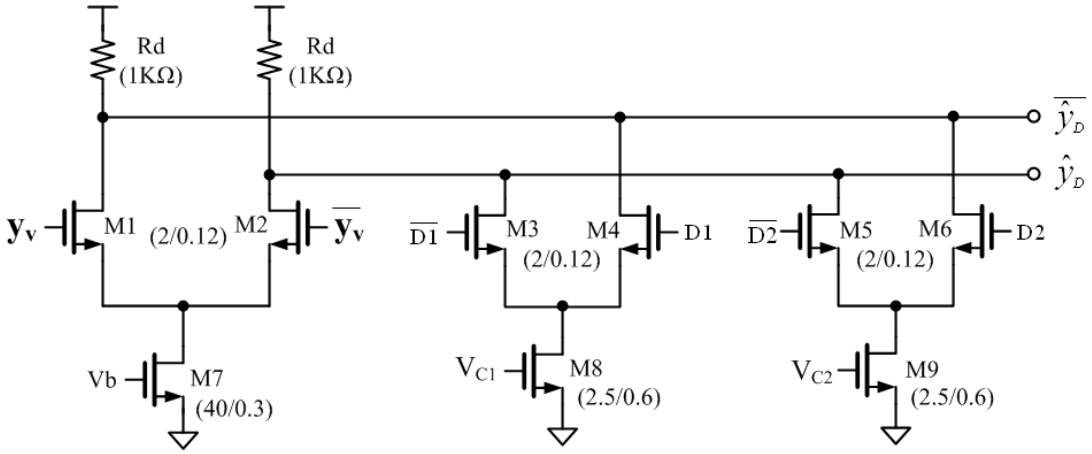


Figure 5.6 Current-mode summer.

V_{C2}. The voltages of V_{C1} and V_{C2} will determine the relative currents in their own differential pair. Note that the drain of M3 and M5 are connected to $\hat{y}_D$ and the drain of M4 and M6 are connected to $\overline{\hat{y}_D}$. This setup is opposite to the main differential pair which the positive input y_v connects $\overline{\hat{y}_D}$ while negative input $\overline{y_v}$ connects $\hat{y}_D$. This is because we know the post-cursor ISI of channel is positive and then we need to create the negative feedback taps to cancel the ISI. Therefore, the adder is in essential a subtractor. Since D1 and D2 are the outputs of D flip-flops (DFF), they are binary signals (±300V_{pp}, we set +300V_{pp} as ONE and -300V_{pp} as ZERO). Thus, we can write the output equation as follows:

$$\hat{y}_D = y_v - C1 \cdot D1 - C2 \cdot D2 \tag{5.4}$$

where $\hat{y}_D$ is the output, y_v is the main input, C1 and C2 are the coefficients of the first tap and second tap, respectively. The tail current magnitudes in the current switches set the desired tap weight. We can say the DFE correction is added to the signal by pulling weighted currents from either the +/— leg of a differential amplifier output using current switches. Since the operation is done in current domain, we call it "current-mode summer."

The operation speed of the current-mode summer must be taken into

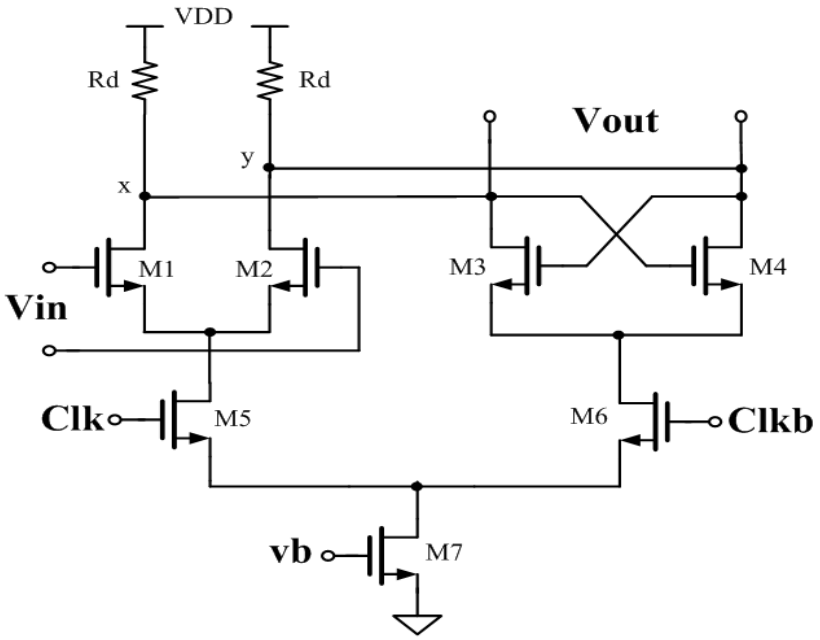
consideration. The delay of the summer needs to be much smaller than the overall timing budget of the feedback loop. The output time constant R_dC_L where C_L is the total capacitance at the output node limits the slope of transition at the output waveform. The summer is loaded by the input capacitance of the next stage DFF, the parasitics of the summer devices themselves, and the wires used for these connections. In order to achieve a high SNR, the summer output needs to get close to its final value before it can be safely used by the next stage. Since our target data rate is 10 Gb/s, the time constant of the summer needs to be much smaller than 100ps. From simulation, the total capacitance at the summer output is less then 20fF. We set the output resistance of summer as 1 KΩ. Therefore, the time constant of summer output is about 20ps which is only one-fifth of 1UI, and it allows enough timing budget for circuits in the critical feedback loop.

5.2.3 Current-Steering Latch

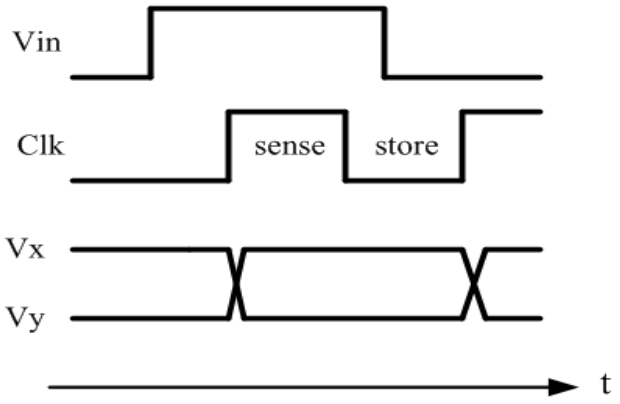
In our equalizer system design, we need two D flip-flops (DFF) as delay elements and five samplers in the sign-sign LMS engine. For high-speed consideration, we implement these DFFs and samplers with current-steering latch, also known as "current-mode logic (CML) latch", as shown in Fig. 5.7. DFFs are back-to-back latches with inverted clocks. Illustrated in Fig. 5.7, a current-steering latch consists of an input pair, M1-M2, for sensing the input and a regenerative pair, M3-M4, for storing the state. The sense and store modes are established by the clock pair, M5-M6. When *Clk* is high, the tail current is steered to M1-M2, allowing V_{out} to track V_{in} while M3 and M4 are off. When *Clk* goes low (*Clkb* goes high), the latch is in store mode, the input pair is disabled, and the cross-coupled pair is enabled, storing

the logical levels at output.

Unlike rail-to-rail latch, the CML latch of Fig. 5.7 operates with relatively small swings at V_{in} and V_{out}, typically less then 1 V_{pp}. We set the output signals of DFFs as ±300 mV_{pp} to be logical levels. Note that the loop gain provided by the cross-coupled pair must exceed 1 so as to ensure the state is stored indefinitely. This means that $g_{m3,4}R_d$ is larger than 1. Since we set output swing as 300 mV (single-ended) and R_d is 1 KΩ, the current of M7 would be 300 μA and $g_{m3,4}$ should be larger than 1 mA/V.



(a)



(b)

Figure 5.7 Current-steering latch, (a) schematic and (b) timing waveform.

Hence, M3 and M4 must be sufficiently wide to provide reasonable loop gain and consume a moderate voltage headroom.

In our equalizer system, the critical path includes the propagation delay of the first DFF. Since a DFF is composed of two CML latches, we need to investigate the total time constant at the two outputs. From simulation, we obtain the total time constant is about 25ps. From previous subsection, we know the summer output has a time constant less than 20ps. Thus, the sum of the two time constant is about 45ps, which is about half of 100ps. Because there is 55ps left in the path, it is enough for the setup time of DFF. For this reason, full-rate clock scheme of DFE architecture can be employed in our design.

The clock frequency of the delay element is 10 GHz while the clock frequency of sampler in the sign-sign LMS engine is 1.25 GHz. The difference of operation frequency makes us use two kinds of latches. We know the CML latch acquires the advantage of high speed at the cost of high power consumption. Since the samplers operate at 1.25 GHz, only one-eighth of 10 GHz, we can design the latches with smaller width as compared to those in delay elements to save power.

5.2.4 Up-Down Counter

As discussed in Chapter 4, the up/down counter and charge pump form a mixed-signal integrator to work as one part of sign-sign LMS algorithm. The up/down counter has the benefit of reducing the MSE, thereby improving the SNR and lowering the BER. From the results of Section 4.3, the 4-bit up/down counter has a better performance in MSE. Hence, we choose 4-bit up/down counter for our design. In fact, the performance is limited by the number of DFE coefficients when a large

number of counter bits is used. In other words, we will gain little performance from using more counter bits when DFE coefficients are the same. The block diagram of a 4-bit up/down counter is shown in Fig. 5.8 [30]. The up/down counter is composed of 4 DFFs and some logic gates. The *Up_b/Down* is the input signal. When *Up_b/Down* is high, the counter will down count. When *Up_b/Down* is low, the counter will behave as an up-counter. Since the charge pump needs two inputs, an *up* signal and a

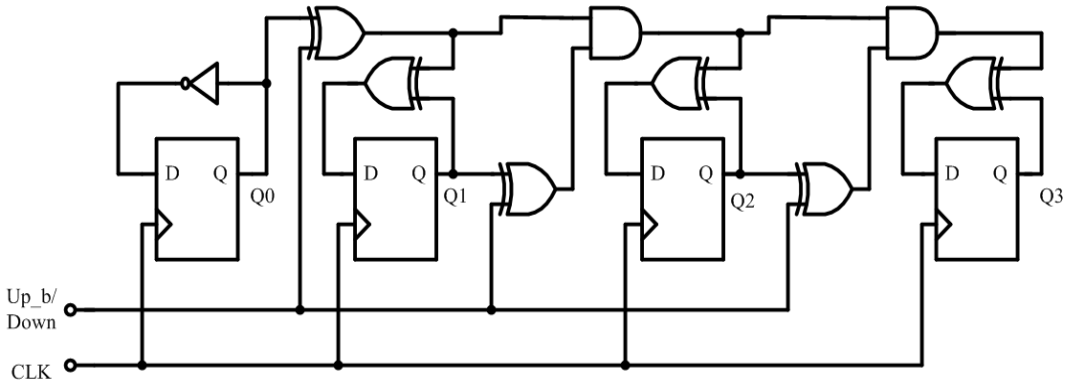


Figure 5.8 Block diagram of 4-bit up/down counter.

down signal, we must create two output signals from the 4-bit up/down counter. Therefore, based on Fig. 5.8 we will construct our 4-bit up/down counter.

Because the counting numbers of 4-bit up/down counter in Fig. 5.8 will range from 0 to 15, we can set +7 (0111) and -7 (1001) in the 2's complement representation as our thresholds of overflow bit and underflow bit, respectively. In other words, when the counter reaches its maximum value (+7), the output signal *Carry* will be high and then the counter is reset to zero. While the counter reaches its minimum value (-7), the output signal *Borrow* will be high and the counter is also reset to zero. On other conditions, *Carry* and *Borrow* will remain low. Fig. 5.9 is the simulated result of the 4-bit up/down counter. The clock (CLK) frequency is 1.25 GHz. When *Up_b/Down* is high, the counter counts down, or it will count up. Q3 is the most significant bit while Q0 is the least significant bit. Clearly, when the output is 1001

(-7), *Borrow* will be high for a clock period, and when the output is 0111 (+7), *Carry* will be high for a clock period. Here, we use asynchronous reset to reset the four DFFs. The simulation manifests correct functionality of the 4-bit up/down counter.

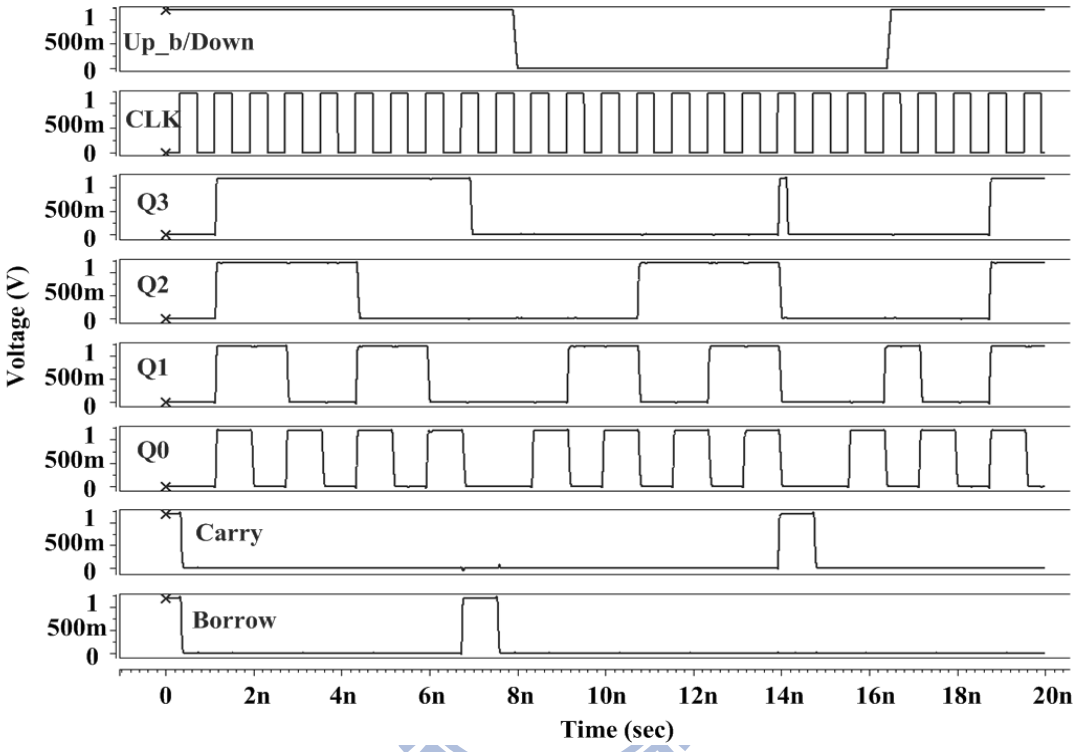


Figure 5.9 Simulated result of 4-bit up/down counter.

5.2.5 Charge Pump

In the sign-sign LMS engine, the charge pump plays an important role in the coefficient update operation. The conventional CMOS charge pump circuits have some current mismatch characteristics due to channel length modulation effect, especially in deep sub-micron process. Fig. 5.10 is our charge pump circuit. As illustrated in Fig. 5.10 , M5~8 is the replica of the charge-pump path, including M1~4. An operational amplifier is added to let the ref voltage follow CP_OUT voltage by a negative feedback loop. Charge current and discharge current can be matched by the

feedback loop due to the same drain voltage in charge and discharge current sources. As long as the high gain of the operational amplifier is provided, ref and CP_OUT are almost the same. Since I2=I3 is always true and the feedback loop equalizes the ref and CP_OUT voltage value, the signal path of M5~8 is the perfect replica of one of M1~4. As a result, I4=I3 and I1=I2, and they balance the current in I1 and I4. Therefore, a current matching technique is completed.

We have known that the charge pump current, charging time or discharging time, and the output capacitive load of charge pump have directly impacts on the step size. The step size is proportional to Ich, the charge pump current, Δt, the switch close time, and is inverse proportional to C_{load}, the output capacitance. Since Δt=800ps, we design that Ich is about 50μA and C_{load} is about 5pF to obtain ΔV about 8mV, the voltage charged or discharged one time, to keep the step size small.

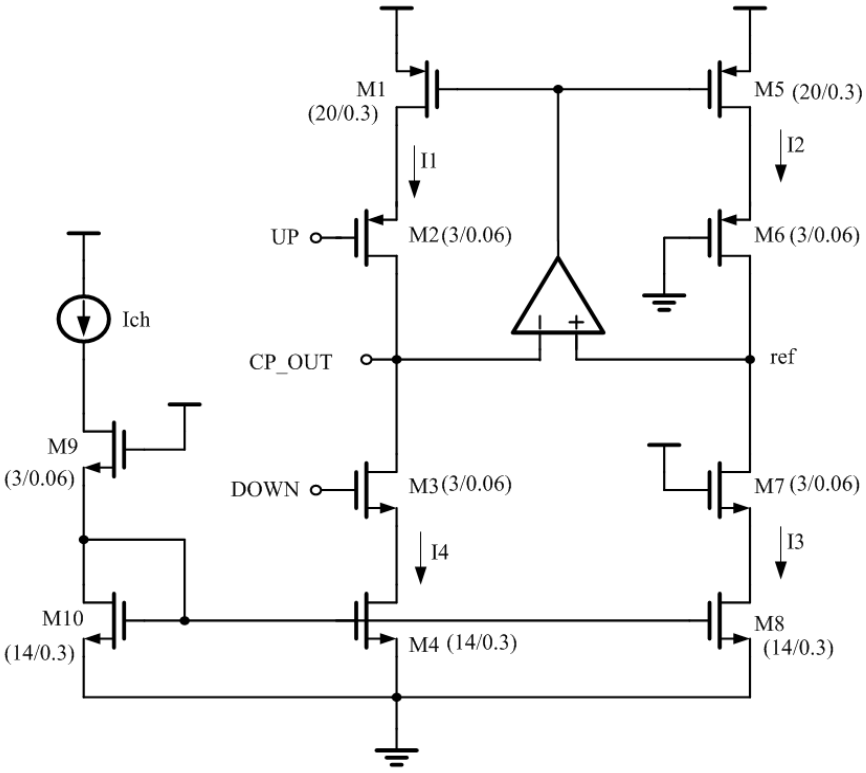


Figure 5.10 Schematic of the charge pump.

Fig. 5.11 is the layout view of charge pump. The use of charge pump instead of counter and DAC is its small area. The area of the charge pump is 14 × 14 (μm^2).

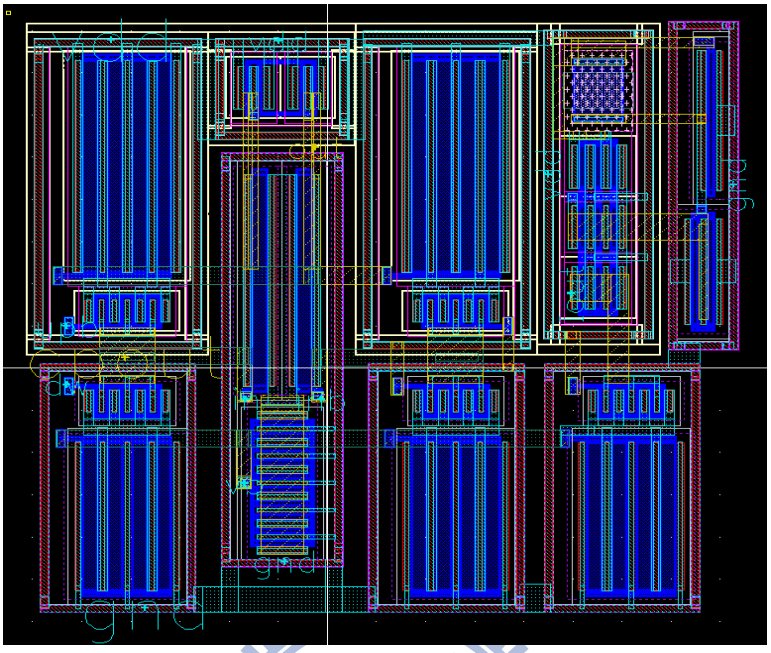


Figure 5.11 Layout view of the charge pump.

5.3 Simulation Results and Layout Implementation

We construct the overall system in circuit level and simulate it with HSPICE. We use the RLGC element in HSPICE to fit the response of a backplane channel as described in section 2.3. It has 15-dB attenuation at 5-GHz. Based on this channel model, we apply PRBS pattern to the channel input and observe the eye diagram of three points: channel output, summation output of equalizer, and buffer output. The PRBS pattern is set to have ±300 mV swing and its data rate is 10-Gb/s. We set the logical levels of ONE and Zero as +300 mV and -300 mV of differential voltages, respectively. The 10-GHz clock signal is used with a voltage source in simulation and the 1.25-GHz clock source for adaptation circuits is generated using a divide-by-8 circuit.

Fig. 5.12 exhibits the controlling voltage of VGA and two coefficient voltages. We can observe that these three voltages converge at last. The voltages of V_g, V_{C1}, and V_{C2} at steady state are about 1.05 V, 0.91 V, and 0.68 V, respectively.

Fig. 5.13(a) shows the eye diagram of channel output. Due to the ISI effect, the signal is attenuated and hard to be distinguished. Fig. 5.13(b) is the eye diagram of equalizer summation output. The vertical eye opening is over ±200 mV_{pp}. We measure that the peak-to-peak jitter of the summation output eye is 31 ps while the peak-to-peak jitter at buffer output is 5ps. The eye diagram of buffer output is shown in Fig. 5.13(c). The vertical eye opening can almost reach ±300 mV_{pp}. The eye diagrams and coefficient voltages reveal that our equalizer system can operate correctly and cancel the post-cursor ISI.

The chip layout view is shown in Fig. 5.14. The total chip area is 510 × 510 (μm^2) Fig. 5.14(b) is the core equalizer system. Table 5.1 shows the summary of our proposed equalizer. The comparison with other works is listed in Table 5.2.

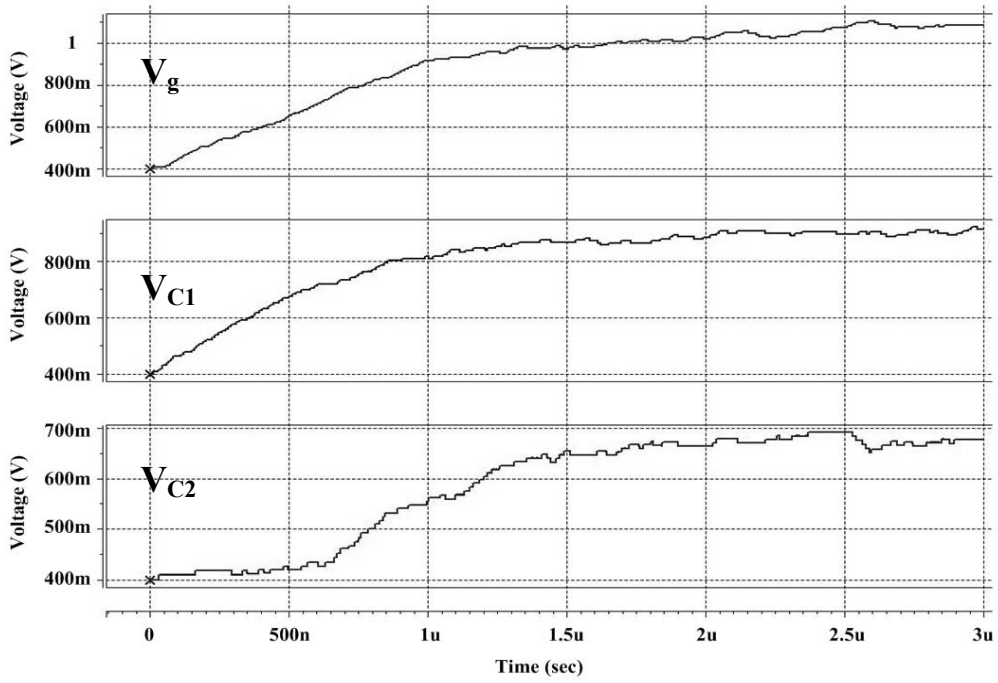
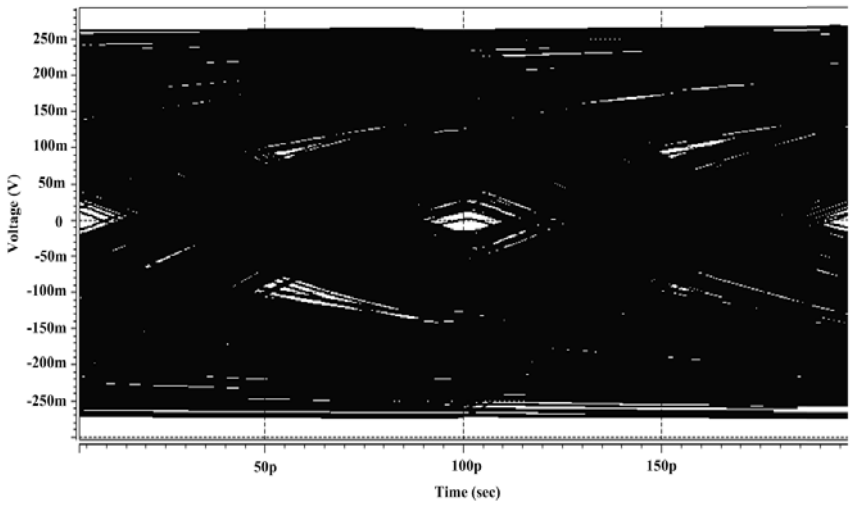
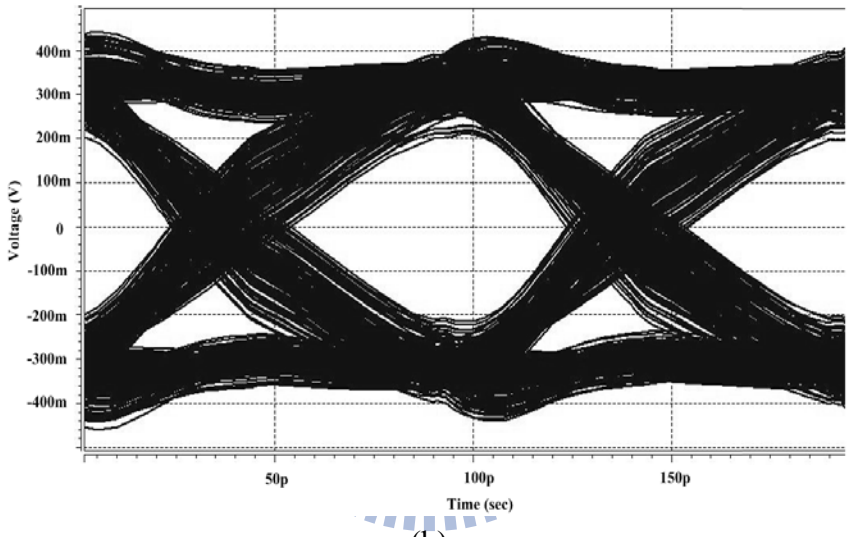


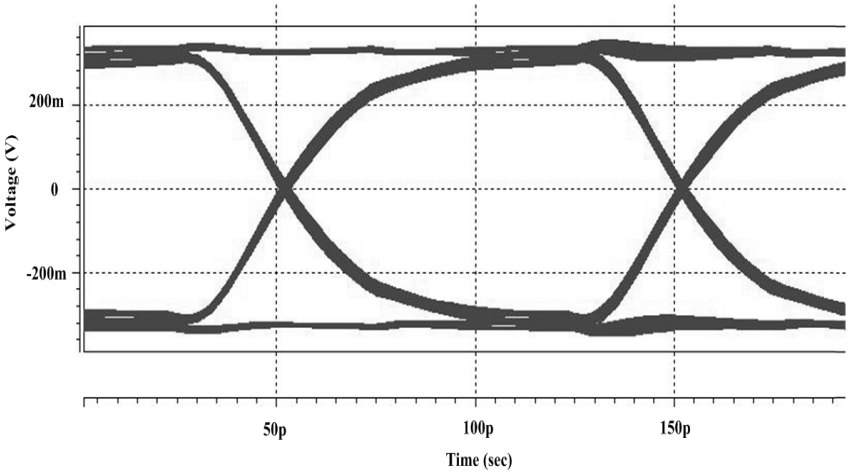
Figure 5.12 Simulated coefficient voltages.



(a)

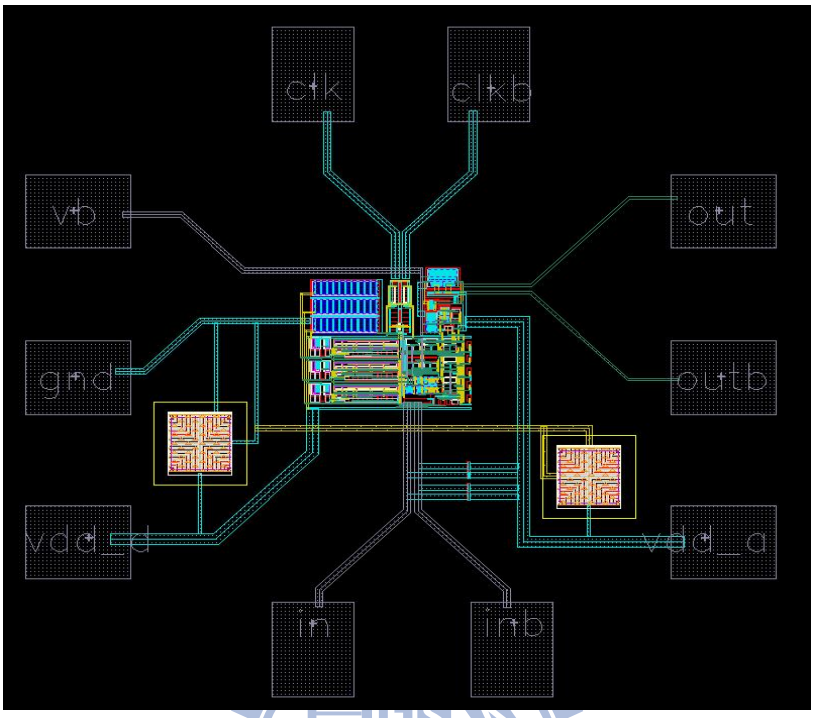


(b)

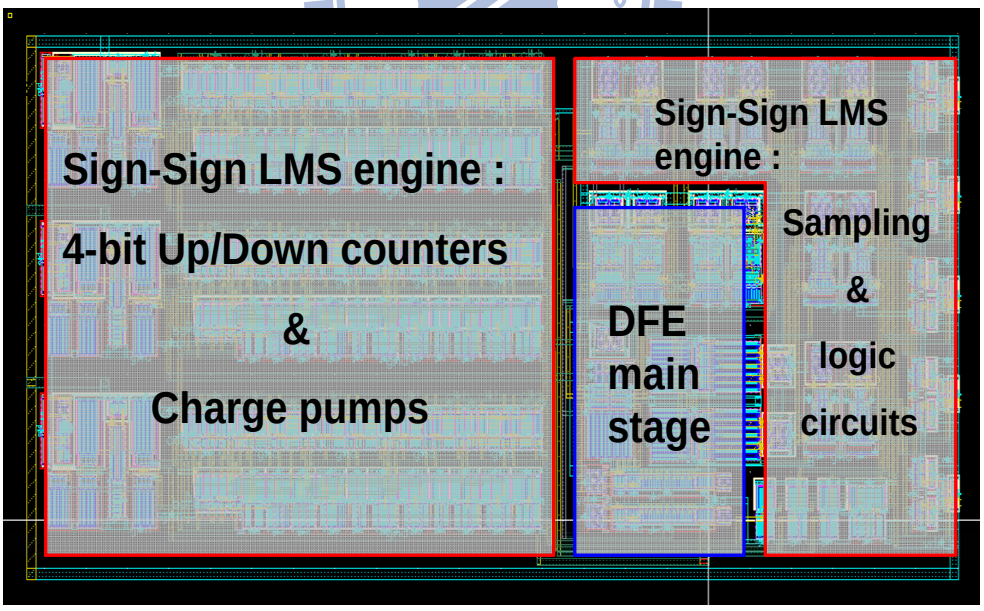


(c)

Figure 5.13 Simulated eye diagram. (a) channel output. (b) equalizer summation output. (c) buffer output.



(a)



(b)

Figure 5.14 Layout view of the proposed equalizer. (a) Total view. (b) zoom-in on the equalizer system.

Table 5.1 Summary of the proposed equalizer

Supply voltage	1.2 V
Data Rate	10 Gb/s
Transmitted signal amplitude	±300 mV_{pp}
Equalized signal at summer output	about ±200 mV_{pp}
Total power consumption @ 10 GHz	40.63 mW
- Equalizer system	11.18 mW
- VGA	1.48 mW
- DFE main stage	2.25 mW
- sign-sign LMS engine	
- analog circuits	6.17 mW
- digital circuits	1.28 mW
- Clocking circuits	8.07 mW
- Buffer stage	21.38 mW
Total chip area	510 × 510 (μm^2)
- Core area	115 × 95 (μm^2)
- VGA	18 × 7 (μm^2)
- DFE main stage	36 × 45 (μm^2)
- sign-sign LMS engine	95 × 72 (μm^2)
- Clocking circuits	15 × 40 (μm^2)
- Buffer stage	33 × 52 (μm^2)
Peak-to-Peak jitter at summer output	31 ps
Peak-to-Peak jitter at buffer output	5 ps
Convergence time	about 20000 bits time

Table 5.2 Comparison with other works

	Our Work	[24]	[31]	[32]
Technology	**65 nm**	90 nm	0.13 μm	0.25 μm SiGe BiCMOS
Supply voltage	**1.2 V**	1 V	N/A	2.5 V
Data Rate	**10 Gb/s**	6 Gb/s	10 Gb/s	10 Gb/s
Architecture	**Direct-Feedback**	DFE with soft decision	Direct-Feedback	Look-ahead
Clock Rate	**Full-rate**	Quarter-rate	Full-rate	Half-rate
# of Post Taps	**2**	2	2	2
Adaptive	**Yes**	No	Yes	No
Channel Loss	**15 dB @ 5GHz**	6.2 dB @ 3GHz	N/A	14 dB @ 5GHz
Power (Core) -DFE main stage	**11.18 mW** **2.25 mW (simulated)**	5 mW	250 mW	7 mW
P-P jitter	**31 ps (simulated)**	N/A	N/A	33 ps
Core area -DFE main stage	**115 × 95 (μm^2)** **36 × 45 (μm^2)**	45 × 98 (μm^2)	> 2000 × 2000 (μm^2)	N/A

5.4 Measurement Environment Setup

To test our proposed equalizer, we use the testing scheme exhibited in Fig. 5.15. All DC power supply sources are provided by Keithley 2400 Source Meter. Agilent N4901B Serial BERT generates the 10 Gb/s transmitted data to our channel and 10 GHz clock source to our equalizer. The output signal from our test chip is fed back to it for calculating the bit-error rate (BER). Tektronics TDS6124C Digital Storage Oscilloscope observes the waveform and the eye diagram of the chip output. Table 5.3 lists the inputs and outputs of our proposed equalizer.

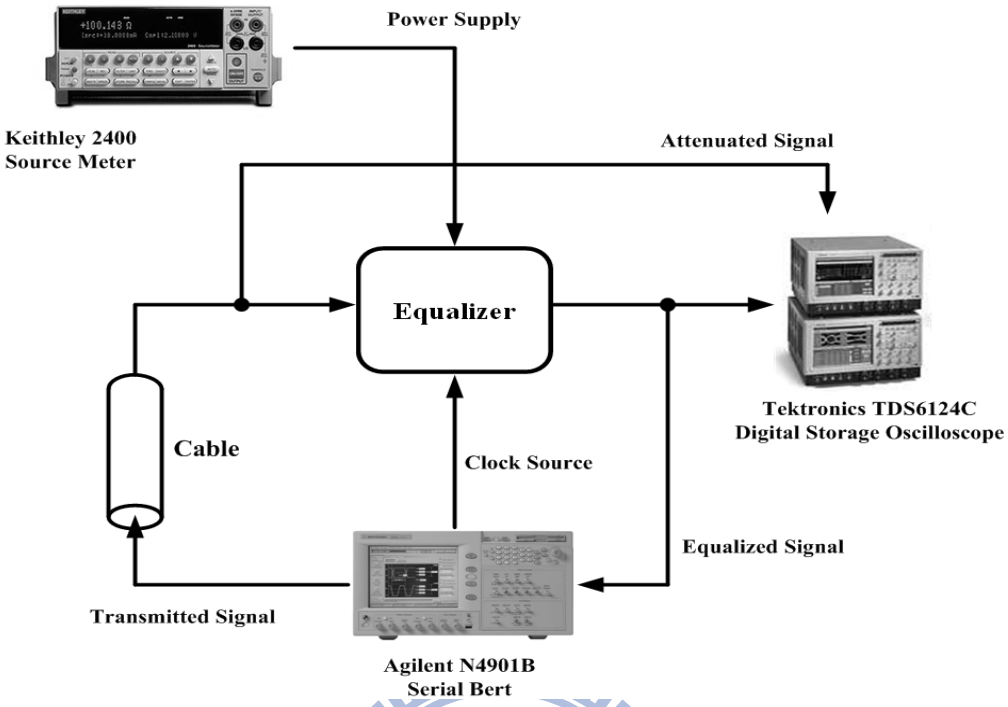


Figure 5.15 Test environment setup.

Table 5.3 Inputs and outputs of our proposed equalizer.

Input	Clock source × 2 Input data × 2 Bias voltage VDD_a for analog circuits VDD_d for digital circuits Ground
Output	Output data × 2

Chapter 6

Conclusions and Future Work

In this thesis, an implementation of 2-tap adaptive decision-feedback equalizer operating at 10 Gb/s of data rate is presented. The data eye at equalizer output can reach ±200 mV_{pp} and has a 31ps peak-to-peak jitter. The total chip area including pads is 510 × 510 μm^2 and the total power consumption is 40.63 mW. The proposed circuit is designed in a 65nm CMOS technology.

From the concern of reducing power consumption, we explore a coefficient-update scheme called hopping update scheme to slow down the clock rate in the coefficient adaptation block. Since the channel may vary slowly with temperature and time, it is not necessary to update the coefficients very fast. Moreover, we utilize a 4-bit up/down counter and a charge pump to form a mixed-signal integrator for controlling the coefficient. The 4-bit up/down counter has the advantage of reducing the input offset of the charge pump and better MSE performance. Instead of using DAC, the easy and small charge pump can do the same work and save area.

Since the data rate is over several multi-Gbps, if the impact of ISI induced by channel is very severe, the number of DFE taps needs to increase, or an FFE is required in the transmitter or in the receiver side to alleviate the design of DFE. Most important of all, if there exists a CDR, the clock could sample the data automatically and the equalizer and CDR could cooperate regularly to become a simple receiver.

Reference

[1] W.J. Dally and J. Poulton, "Transmitter Equalization for 4-Gbps signaling," *IEEE Mirco*, vol. 17, pp. 48-56, Jan./Feb. 1997.

[2] R. Farjad-Rad, C. K. Ken Yang, M. A. Horowitz, and T. H. Lee, "A 0.4-μm CMOS 10-Gb/s 4-PAM Pre-Emphasis Serial Link Transmitter," *IEEE Journal of Solid-State Circuits*, vol. 34, pp. 580-585, May 1999.

[3] C. H. Lin, C. H. Tsai, C. N. Chen, and S. J. Jou, "Multi-Gigabit Pre-Emphasis Design and Analysis for Serial Link," *IEICE Transactions on Electronics*, vol. E88-C, pp. 2009-2019, 2005.

[4] C. M. Chu, C. H. Chuang, C. H. Lin, and S. J. Jou, "A 6Gbps Serial Link Transmitter with Pre-emphasis," *VLSI Design, Automation and Test 2007, International Symposium.*

[5] J. E. C. Brown, P. J. Hurst, and L. Der, "A 35 Mb/s Mixed-Signal Decision-Feedback Equalizer for Disk Drives in 2-μm CMOS," *IEEE Journal of Solid-State Circuits*, vol. 31, pp. 1258-1266, Sept. 1996.

[6] J.-S. Choi, M.-S. Hwang, and D.-K. Jeong, "A 0.18-μm CMOS 3.5-Gb/s Continuous-Time Adaptive Cable Equalizer Using Enhanced Low-Frequency Gain Control Method," *IEEE Journal of Solid-State Circuits,* vol. 39, pp. 419-425, Mar. 2004.

[7] Y. Tomita, M. Kibune, J. Ogawa, W. W. Walker, H. Tamura, and T. Kuroda, "A 10-Gb/s Receiver With Series Equalizer and On-Chip ISI monitor in 0.11-μm CMOS," *IEEE Journal of Solid-State Circuits,* vol.40, pp. 986-993, Apr. 2005.

[8] J. Kim, J. Yang, S. Byun, H. Jun, J. Park, C. S. G. Conroy, and B. Kim, "A Four-Channel 3.125-Gb/s/ch CMOS Serial-Link Transceiver With a

Mixed-Mode Adaptive Equalizer," *IEEE Journal of Solid-State Circuits,* vol. 40, pp. 462-471, Feb. 2005.

[9] J. F. Bulzacchelli, M. Meghelli, S. V. Rylov, W. Rhee, A. V. Rylyakov, H. A. Ainspan, B. D. Parker, M. P. Beakes, A. Chung, T. J. Beukema, P. K. Pepeljugoski, L. Shan, Y. H. Kwark, S. Gowda, and D. J. Friedman, "A 10-Gb/s 5-Tap DFE/4-Tap FFE Transceiver in 90-nm CMOS Technology," *IEEE Journal of Solid-State Circuits,* vol. 41, pp. 2885-2900, Dec. 2006.

[10] T. Beukema, M. Sorna, K. Selander, S. Zier, B. L. Ji, P. Murfet, J. Mason, W. Rhee, H. Ainspan, B. Parker, and M. Beakes, "A 6.4-Gb/s CMOS SerDes Core With Feed-Forward and Decision-Feedback Equalization," *IEEE Journal of Solid-State Circuits,* vol. 40, pp. 2633-2645, Dec. 2005.

[11] K. Khrishna, D. A. Yokoyama-Martin, A. Caffee, C. Jones, M. Loikkanen, J. Parker, R. Segelekn, J. L. Ssonntag, J. Stonick, S. Titus, D. Weinlader, and S. Wolfer, "A Multigigabit Backplane Transceiver Core in 0.13-μm CMOS With a Power-Efficient Equalization Architecture," *IEEE Journal of Solid-State Circuits,* vol. 40, pp. 2658-2666, Dec. 2005.

[12] V. Balan, J. Caroselli, J. G. Chern, C. Chow, R. Dadi, C. Desai, L. Fang, D. Hsu, P. Joshi, H. Kimura, C. Y. Liu, T. W. Pan, R. Park, C. You, Y. Zeng, E. Zhang, and F. Zhong, "A 4.8－6.4-Gb/s Serial Link for Backplane Applications Using Decision Feedback Equalization," *IEEE Journal of Solid-State Circuits,* vol. 40, pp. 1957-1967, Sept. 2005.

[13] A. X. Widmer and P. A. Franaszek, "A DC-Balanced, Partitioned-Block, 8B/10B Transmission Code," *IBM J. Res. and Develop.*, vol. 27, pp. 440-451, Sept. 1983.

[14] B. Razavi, *Design of Integrated Circuits for Optical Communications*, McGraw-Hill, 2003.

[15] C. K. Ken Yang, "Design of High-Speed Serial Links in CMOS," Ph.D.

Dissertation, Stanford University, 1998.

[16] S. Ramo, J. R. Whinnery, and T. van Duzer, *Fields and Waves in Communication Electronics*, New York: Wiley, 1984.

[17] D. K. Cheng, *Field and Wave Electromagnetics*, Addison Wesley, 1989.

[18] H. B. Bakoglu, *Circuits, Interconnections, and Packaging for VLSI*, Addison Wesley, 1990.

[19] *Star-HSPICE User's Manual*, Avant! Corporation, 2001.

[20] S. Gondi, and B. Razavi, "Equalization and Clock and Data Recovery Techniques for 10-Gb/s CMOS Serial-Link Receivers," *IEEE Journal of Solid-State Circuits,* vol. 42, pp. 1999-2011, Sept. 2007.

[21] J. Lee, "A 20-Gb/s Adaptive Equalizer in 0.13-μm CMOS Technology," *IEEE Journal of Solid-State Circuits,* vol. 41, pp. 2058-2066, Sept. 2006.

[22] C. A. Belfiore and J. John H. Park, "Decision Feedback Equalization," *Proceedings of the IEEE,* vol. 67, pp. 1143-1156, Aug. 1979.

[23] S. Kasturia and J. H. Winters, "Techniques for High-Speed Implementation of Nonlinear Cancellation," *IEEE Journal On Selected Areas In Communications*, vol. 9, pp. 711-717, June 1991.

[24] K. L. Wong, A. Rylyakov, and C. K. Ken Yang, "A 5-mW 6-Gb/s Quarter-Rate Sampling Receiver With a 2-Tap DFE Using Soft Decisions," *IEEE Journal of Solid-State Circuits,* vol. 42, pp. 881-888, Apr. 2007.

[25] A. Emami-Neyestanak, A. Varzaghani, J. F. Bulzacchelli, A. Rylyakov, C. K. Ken Yang, and D. J. Friedman, "A 6-mW 10-Gb/s Receiver With Switched-Capacitor Summation DFE," *IEEE Journal of Solid-State Circuits,* vol. 42, pp. 889-896, Apr. 2007.

[26] J. Proakis, *Digital Communication*, McGraw Hill, 2001.

[27] S. Haykin, *Adaptive Filter Theory*, Prentice-Hall, 2004.

[28] Y. S. Huang, “Multi-Gbps Equalizer Technology for Serial Link System,” Master Thesis, National Chiao Tung University, 2008.

[29] M. Q. Le, P. J. Hurst, K. C. Dyer, “An Analog DFE for Disk Drives Using a Mixed-Signal Integrator,” *IEEE Journal of Solid-State Circuits,* vol. 34, pp. 592-598, May 1999.

[30] M. R. Stan, A. F. Tenca, and M. D. Ercegovac, “Long and Fast Up/Down Counters,” *IEEE Transactions On Computers*, vol. 47, pp. 722-735, July 1998.

[31] A. Momtaz, D. Chung, N. Kocaman, J. Cao, M. Caresosa, B. Zhang, and I. Fujimori, “A Fully Integrated 10-Gb/s Receiver With Adaptive Optical Dispersion Equalizer in 0.13-μm CMOS,” *IEEE Journal of Solid-State Circuits,* vol. 42, pp. 872-880, Apr. 2007.

[32] A. Kiaei, B. Matinpour, A. Bahai, and T. H. Lee, “A 10-Gb/s Equalizer with Decision Feedback for High Speed Serial Links,” *IEEE Custom Integrated Circuits Conference*, pp. 285-288, 2007.