2000

Mixed-signal equalizer for disk drive read channel and digital calibration for time-interleaved A/D converter

Huawen Jin
Iowa State University

INFORMATION TO USERS

This manuscript has been reproduced from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps.

Photographs included in the original manuscript have been reproduced xerographically in this copy. Higher quality 6" x 9" black and white photographic prints are available for any photographs or illustrations appearing in this copy for an additional charge. Contact UMI directly to order.

Bell & Howell Information and Learning
300 North Zeeb Road, Ann Arbor, MI 48106-1346 USA
800-521-0600

Mixed-signal equalizer for disk drive read channel and digital calibration for time-interleaved A/D converter

by

Huawen Jin

A dissertation submitted to the graduate faculty
in partial fulfillment of the requirements for the degree of
DOCTOR OF PHILOSOPHY

Major: Electrical Engineering (Microelectronics)
Major Professor: Edward KF Lee

Iowa State University
Ames, Iowa
2000

Graduate Collage
Iowa State University

This is to certify that the Doctoral dissertation of

Huawen Jin

has met the dissertation requirements of Iowa State University

Signature was redacted for privacy.

Committee Member

Signature was redacted for privacy.

Committee Member

Signature was redacted for privacy.

Committee Member

Signature was redacted for privacy.

Committee Member

Signature was redacted for privacy.

Major Professor

Signature was redacted for privacy.

For the Major Program

Signature was redacted for privacy.

For the Graduate Collage

To my wife Lin

and my parents

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

1 INTRODUCTION

1.1 Motivation

The advent of information age has brought enormous demands for storing, transmitting and processing digital data. An example is that the explosive growth in both personal computer and Internet has been fueled by high density hard disk storage unit. All these applications heavily depend on the ability in high speed signal processing. This trend has led to the rapid development in computation capability of modern Digital-Signal-Processing (DSP). This development has pushed the bottleneck of signal-processing path towards the analog front end. Therefore, the implementation of the interface between the analog world and the digital world to fully utilize the potential of DSP has become a major challenge in today's integrated circuit design. These integrated circuits involve mixed-signal designs that combine analog and digital parts. They aim to achieve high speed and high reliability at low cost.

In this dissertation, two topics that represent the trend in high speed circuit design have been chosen to demonstrate the design strategy of using mixed-signal design techniques to improve circuit performance. The first topic is the design and implementation of a mixed-signal equalizer for disk drive read channels. Experimental results have demonstrated the effectiveness of this mixed-signal architecture in achieving high speed and low power dissipation without using more advanced fabrication processes. The second part of the dissertation focuses on the calibration of a time-interleaved ADCs. Two calibration algorithms are proposed to calibrate two major non-ideal effects that limit the performance of such converters.

1.1.1 Mixed-Signal Equalizer for Disk Drive Read Channel

Magnetic disk storage system sets a good example of modern technology that combines precise mechanical and electronic systems. From digital storage devices in the 1960's with a few Kilobytes to present hard disk drive with ten's of Gigabytes capacity, magnetic disk storage has been one of the fastest growing baseband signal processing applications. The price per unit data has been dropping dramatically during the past decade due to improvements in signal processing technology.

The principle of digital magnetic storage is to use two directions of magnetic flux to store binary codes (either '1' or '0') on a magnetic film. The material used to store the data can remain magnetized after the applied magnetic field is removed. In order to record the data, the digital signal controls the polarity of the current in the coil of the write head. The writing current induces a magnetic flux in the spinning disk with a certain direction corresponding to the polarity of the current. During the read out procedure, the flux is sensed and converted back into digital signal. The spacing between the magnetic fluxes in the spinning disk determines the density of the storage device. During the entire write/read process, errors are inevitable. However the requirement of magnetic storage is to achieve high capacity and fast read out speed with low Bit-Error-Rate or BER.

The block diagram of the disk storage system is shown in Figure 1.1. The digital data will first be encoded to reduce the probability of error during the read out process. This encoding is usually done in digital domain. The encoded data is recorded onto the magnetic film using the read/write head. Compared to writing, the read out procedure encounters many obstacles. First of all, the output voltage that is sensed by the read/write head is usually a rather weak signal. Secondly, temperature change, environmental variation and disk film rotation errors during writing lead to the increase in the probability of read out errors. The third problem is cross talk between symbols when the storage density increases. This cross talk is usually referred to as Inter-Symbol-

Interference (ISI). All of them deteriorate the quality of the read out signal and increase the probability of error. Consequently the signal has to go through necessary signal processing before going to the detector. Figure 1.1 shows the analog-front-end circuit. An equalizer is added before the digital detector as well. The analog-front-end is usually composed of signal magnitude control unit (AGC) and low-pass pre-filters operating in the analog domain. The equalizer is used for compensating channel distortion and can be implemented with a pure analog, mixed-signal or pure digital approach depending on different design considerations. After the equalizer, all the blocks are usually implemented in the digital domain.

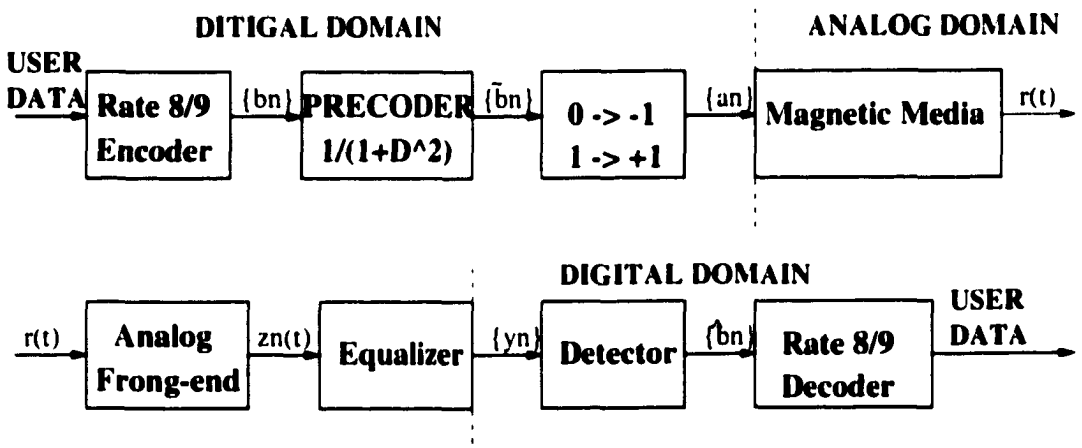

Figure 1.1 Model of Disk Storage System

From the above discussion, the mixed-signal characteristics of the disk storage system are obvious. Furthermore, the equalizer plays an important role in the mixed-signal processing presented in this system. The motivation of this work is to explore a new mixed signal equalizer that has the potential of achieving high speed and low power dissipation for a given fabrication process. Two prototype chips were designed and fabricated using HP 0.5μm CMOS process. Experimental results show that the proposed

mixed-signal programmable FIR type equalizer is capable of operating at a speed up to 360 MHz. This is mainly attributed to a new mixed-signal multiplier that simplifies the circuit design and thus enables high speed and low power consumption. Compared with the prior art, this work demonstrates that besides using better fabrication processes, a good circuit design has significant potential in improving the overall performance.

1.1.2 Calibration of Analog-to-Digital Converter

In many mixed-signal systems, an Analog-to-Digital Converter acts as a direct interface between the analog world and the digital world. It performs a conversion that provides the digital representation of an analog signal. Usually the resolution (number of bits used to represent the analog signal) and the conversion speed are the two major specifications of an ADC. For the application of disk drive read channels, very high speed ($> 500MSPS$) and low resolution ($\sim 6bits$) are required. In contrast, in audio or instrumentation system, low speed (tens of $KSPS$ range) and extremely high resolution ($\geq 20bits$) are necessary. The fast developing wireless communication requires both high speed ($\geq 50MSPS$) and high resolution ($\geq 12bits$) with large dynamic range. This kind of system poses a great challenge to ADC design. The second part of this dissertation focuses on ADC's that fall under the third category.

The most popular choice of implementing such high speed and high resolution ADC is to use pipelined structure, which can generate the digital output with resolution in the range of 10bits$\sim$14bits and a conversion rate of $> 10MSPS$. Another choice to achieve this high-speed requirement is to operate several ADC's with the same structure in parallel. This kind of structure is called Time-Interleaved ADC [1]. By sending the input signal into different sub-converters, which are called channels in this dissertation, the combined conversion speed can be higher than that of any single sub-converter.

Nonetheless, there is one problem inherent in time-interleaved ADC's: channel mismatch. Even though this structure has been proposed for many years, applications are limited because of the difficulty in solving this problem. Generally, there are two major

categories of channel mismatch: (1) Device mismatch. This includes geometric errors in size, shape, etc. in passive devices such as capacitors or active devices such as transistors. These errors may affect the overall channel characteristic. Mismatches between channels such as gain errors and offset errors may occur. (2) Timing mismatch. An ideal ADC is supposed to sample the input at evenly distributed sampling instants. However, in the multi-channel case, the clock phases for different channels will have non-zero skew, which will further induce distortion. This is an important source of errors that has not been effectively addressed.

The second part of this work concentrates on the calibration of these two errors in time-interleaved ADC's. In dealing with the first mismatch problem, a channel randomization method is proposed as a supplement to existing methods. For timing errors, a digital background calibration algorithm based on interpolation is proposed. A rigorous analysis on the unique issue of timing error effects is given. A theoretical analysis on the proposed method is also presented.

1.2 Dissertation Organization

Based on the above discussion, the whole dissertation is divided into two parts. Chapter 2 to Chapter 4 focus on the new mixed-signal equalizer. Chapter 5 to 7 are the second part focusing on the calibration of time-interleaved A/D Converter.

Chapter 2 introduces the basics of magnetic storage and the corresponding detection methods. It also introduces a commonly adopted adaptation algorithm for the equalizer.

Chapter 3 discusses the implementation details of Finite-Impulse-Response (FIR) based equalizer in disk drive read channel. Two equalizers are designed and fabricated using a 0.5μm CMOS technology. The focus of this chapter is a newly developed mixed signal multiplier. This chapter explains how this multiplier is essential to the whole system to achieve high speed and low power dissipation.

Chapter 4 provides the experimental results of the prototype chips. The tested per-

formance demonstrates the proposed implementation techniques.

Chapter 5 presents the background of *ADC*'s. Emphasis is put on the time-interleaved structure, especially its potential and its limitation.

Chapter 6 discusses the issues of timing errors in Time-interleaved structure. A calibration method that successfully reduces the timing error effects is proposed. Since it is a digital signal processing technique, no major changes in the analog circuits are required. Since it is also a background process, it does not disturb the main conversion process. Therefore calibration at real time can be achieved.

Chapter 7 discusses the issue of device mismatch effects, especially the gain and offset mismatches in time-interleaved *ADC*. A new method based on randomization is applied to enhance the dynamic performance of such converter.

Finally in Chapter 8, summary and conclusions of this PhD work are presented.

2 DISK DRIVE READ CHANNEL

2.1 Introduction

As introduced in Chapter 1, the steady improvement in the mechanical fabrication as well as the advances in signal processing technology improve the magnetic storage system to have high storage capacity while at a rather low cost. This chapter provides an introduction to the signal processing techniques in this system, especially the read out process.

In this dissertation, all the signal processing on the signal sensed by the read/write head from the magnetic media is referred to as the *disk drive read channel*, or in short *read channel*.

In the following sections, a quick review of disk storage technology is given first. Then the evolution of detection method is discussed. Finally the role of equalization in the read out process is introduced.

2.2 Overview of Magnetic Storage

The mechanism behind magnetic storage is that when a magnetic field is applied to the storage material, the material is able to retain the applied field even after the writing process is finished. It means that the information can be stored permanently unless a new writing process is performed.

On the magnetic film, binary data "1" or "0" can be distinguished as the transition of the orientation of the magnetic field. To explain this in more detail, if the data is "1", the magnetic flux is changed to the opposite orientation, and if the data is "0", there is

no change. The change in the magnetic flux from one orientation to the other is called a transition. "1" or "0" is thus stored by the presence or absence of a transition. This type of coding scheme is called *NRZI - Non-Return-to-Zero-Invert-on-"1"*. Since the disk is in motion, the change of magnetic flux from one direction to the other induces a voltage on the terminal of the read/write head during read out. As will be explained shortly, the detection of this transition involves a lot of signal processing. Figure 2.1 illustrates how the user data is written on to the media and the appearance of the signal during read back.

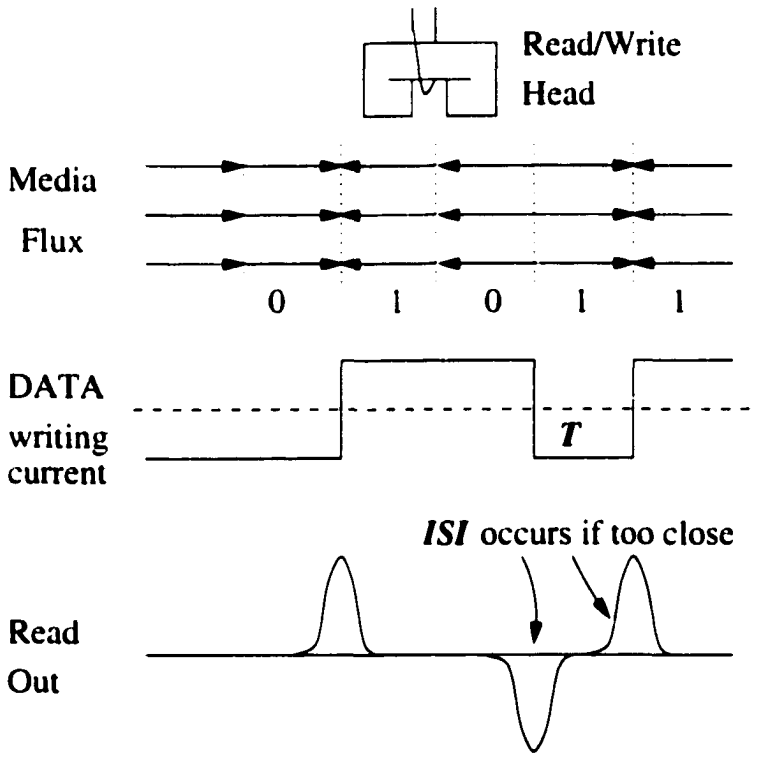

Figure 2.1 Read/Write Procedure

It can be observed that the disk drive read out signal from inductive head has a *DC* null since the induced voltage only responds to the magnetic flux changes. In addition, the frequency response of the channel deteriorates at high frequency. Therefore the disk drive channel has a band-pass response.

The data storage capacity can be increased by decreasing the spacing between transitions. This can be done in two ways. The first is to improve the read/write head by reducing the head gap size. However, this method is reaching its limit now[2]. The second method is to increase the disk rotation speed. Correspondingly, the system read/write clock needs to be increased. Modern hard disk rotation speed has increased from 3600 rpm to 7200 rpm and over 10,000 rpm while the read/write clock is reaching 300 MHz. As long as the size of one storage cell is larger than the minimum allowable magnetic fragment, the density can be improved along with the speed. The practical limitation lies on the achievable distance from the read/write head to the surface and the size of a magnetic particle.

Generally speaking, a disk storage system is composed of the magnetic media and the read/write head with pre-coding block on the write side, and the pre-amplifying, pre-filtering, equalization, decoding, etc. on the read side. Figure 2.2 shows the model of a disk drive read channel model illustrating the above blocks. For simplicity, only the major data path is listed. The functions of the above blocks are explained as follows:

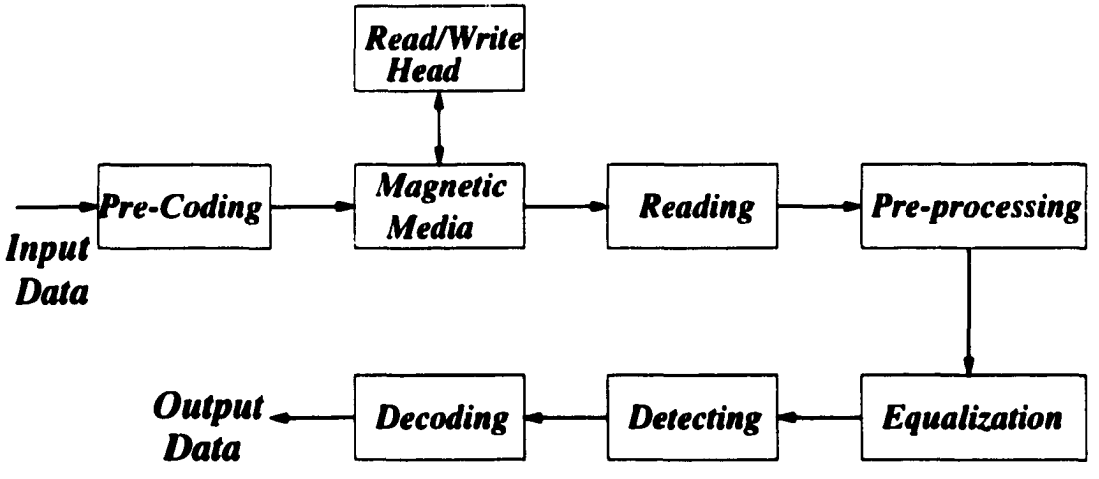

Figure 2.2 Data Path of Disk Storage System

Pre-coder:	Encodes the user data according to certain algorithm to ensure smooth write and read process. An example is that continuous *zeros* and *ones* should be avoided.
Magnetic media:	It includes multiple disks covered with magnetic materials. These disks rotate at a constant speed during normal operation.
Read/write head:	Interface between the magnetic media and the outside circuits. It uses non-contact method to interact with the magnetic media. It has small gap for leaking a controlled magnetic flux onto the media to magnetize the material. During writing, the direction of the current going through the head is controlled by the write data. During reading, the induced voltage will appear on its terminal.
Pre-amplifier:	Amplifies the read out signal to a certain range that is easier for the later circuit to process. This is usually composed of an Auto-Gain-Control amplifier. The general requirement is low noise.
Pre-filter:	Low-pass continuous-time filter acts as anti-alias filter to get rid of high-frequency noise.
Equalizer:	Shapes the read out signal to a certain form for better detection. It compensates the channel distortion. It is very important and directly affects probability of errors on the read out signal.
Detector:	Detects the symbol value according to the equalized signal. The output is still encoded.
Decoder:	Decodes the symbol to recover the user data. It reverses the process of data pre-coding when writing the data on the media.

2.3 Detection Methods

Detection method refers to the way to detect a symbol after the pre-amplifying and pre-filtering. A symbol refers to a pulse or voltage level that can be clearly identified by the detector. In this section, different detection methods are discussed.

When the storage density is relatively low, the distance between read out pulses is fairly large. Hence the detection method can be relatively simple. However, as the density keeps increasing, the distance between symbols becomes smaller and smaller. As seen clearly from figure 2.1, when two written pulses are close enough to have overlapped read out signals, *ISI* occurs. *ISI* not only changes the amplitude of consecutive read out pulses, but also shifts the positions of their peaks. More complex detection methods must be used in dealing with the *ISI* problem.

2.3.1 Peak detection

As introduced in the previous section, binary data are recorded as the presence and absence of magnetic flux transitions. The transition from one direction to the other is sensed as a single voltage pulse by the read/write head. When the disk drive density is not very high, a straightforward method is derived by 'searching' the peak of the read back signal to determine if a pulse has been read or not. Either positive or negative pulse is decoded as '1' and no pulse as '0'. This is called *Peak Detection*. This method provided satisfactory results in early hard disk drive systems.

To find out the position of the peak, a peak detector is used. It differentiates the read out signal with respect to time. The derivative equals to zero at the peak position. A simple zero-crossing test or a comparator can be used to detect this zero value. Figure 2.3 illustrates a single pulse and the derivative of the pulse. In order to avoid the influence of noise, certain kinds of threshold monitoring should be used to accurately locate the peak position. After a peak is detected, a signal of '1' is sent to the output. Since there is no timing information in the read out signal, everything is then associated

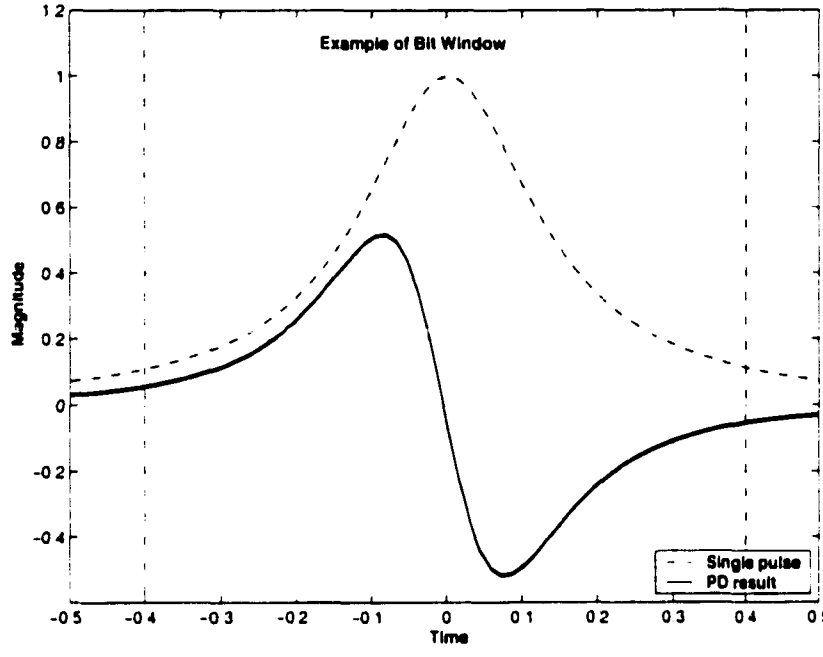

Figure 2.3 Principle of Peak Detection

with the disk rotation speed. A separate clock generator is used to synchronize the speed with the read out signal. Therefore, by accurately aligning the read out derivative and the pulse, correct data can be obtained. On the other hand, if the derivative can not be placed correctly across the pulse, an error is generated. This usually happens when there are continuous '0's, i.e. when no transition happens for a long periods of time. Since the detector has to recover the timing for synchronization, long period of time without any transition will make the clock generator 'lose lock'. This appears to be a common problem for other methods too. Therefore a special coding scheme used in the pre-coding block to prevent this [3].

2.3.2 Sampling detection

When the disk storage density increases, as can be seen from Figure 2.4, two adjacent pulses will have large interference. This has two effects. First it reduces the peak level, second it causes shifting of the peak. Under these circumstances, the peak detection cannot detect the two peaks correctly. When the storage density increases even further,

it is impossible for the peak detection method to find the data pulse. As an evolution of this, another method is introduced to overcome this problem.

Instead of looking for single transition position like in the peak detection, more information can be captured when a continuous time signal is used. In order to process this signal, a track-and-hold method called as *Sampling detection* is used. The idea behind this is that if the sampling rate is higher than the data recording rate, every transition pulse will have more than one sample. Even if the symbol shifts from its original position, sampling detection can still capture its shape. Even though the ISI may still

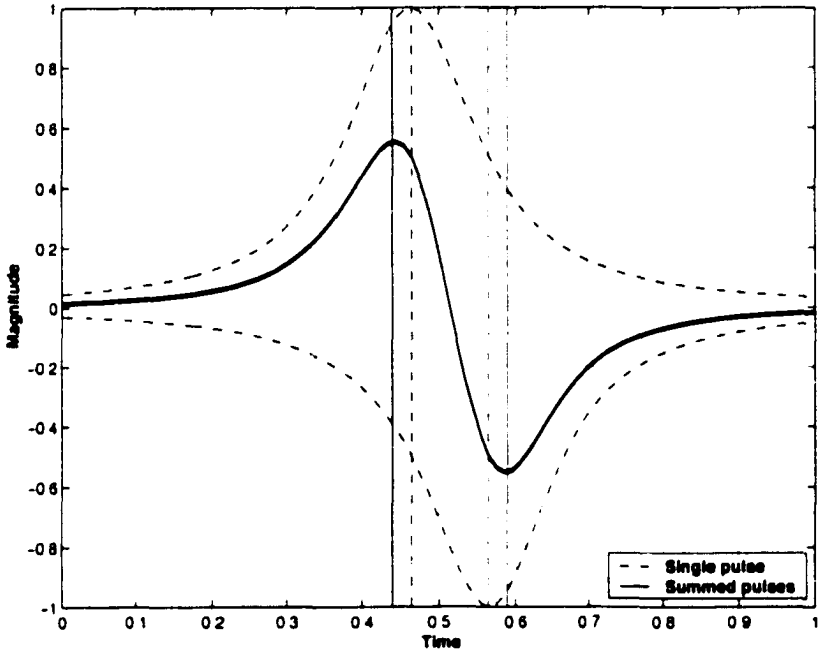

Figure 2.4 Inter-Symbol-Interference Generated by Two Adjacent Pulses

exist, the effects of ISI can be reduced or eliminated after using proper equalization methods with the help of known relationships between different pulses. To do the sampling, an analog approach does the processing in a discrete-time manner, while a digital domain method is to use a very high speed but moderate resolution analog-to-digital converter to sample the signal and convert the signal into digital form. The current level of hard disk drive requires a very high performance A/D converter with a speed of $\geq$

$500MSPS$. This is still a very challenging topic. There are two main approaches in the sampling detection category: the $PRML$ system with partial response signaling and the DFE system.

2.3.2.1 Partial response system

A popular way of doing the detection under the condition of highly packed symbols is Partial Response (PR) method [4]. Partial response method will improve the read back signal for an easier detection before the signal is sent to the detector. It is usually combined with Maximum-Likelihood detection ($PRML$) to further improve the detection effectiveness [5, 6].

Partial Response method is one of the 'sampling detection' method that the detector works on the sampled data from the read back signal. The idea of partial response is that by intentionally shaping the pulse with a known polynomial, the ISI will have a known form for a given data sequence. The Maximum-Likelihood-Sequence-Detector ($MLSD$) takes the sampled values to determine which data sequence best matches the sampled values. This data sequence becomes the detected data sequence. An efficient technique to implement the maximum likelihood detector is the *Viterbi* detector. Combining Partial Response with Maximum-Likelihood detection enables great improvement in disk drive read detection. Hence the method is usually referred to as $PRML$. Almost every modern disk drive adopts this method. Several most commonly used partial response functions are [7]:

$$PR4 : P(D) = 1 - D^2$$

$$EPR4 : P(D) = 1 + D - D^2 - D^3$$

$$E^2PR4 : P(D) = 1 + 2D - 2D^3 - D^4 \tag{2.1}$$

where D represents unit delay. Figure 2.5 gives the performances of these three functions in frequency domain. As can be observed from the frequency response, the partial response signaling tries to boost the middle frequency range. Recall that the disk drive

read channel has a band pass response, therefore partial response is very suitable for this type of system. The extra benefit from partial response function is that high frequency noise will not be amplified since differentiation, which amplifies high frequency noise, is not required in this technique.

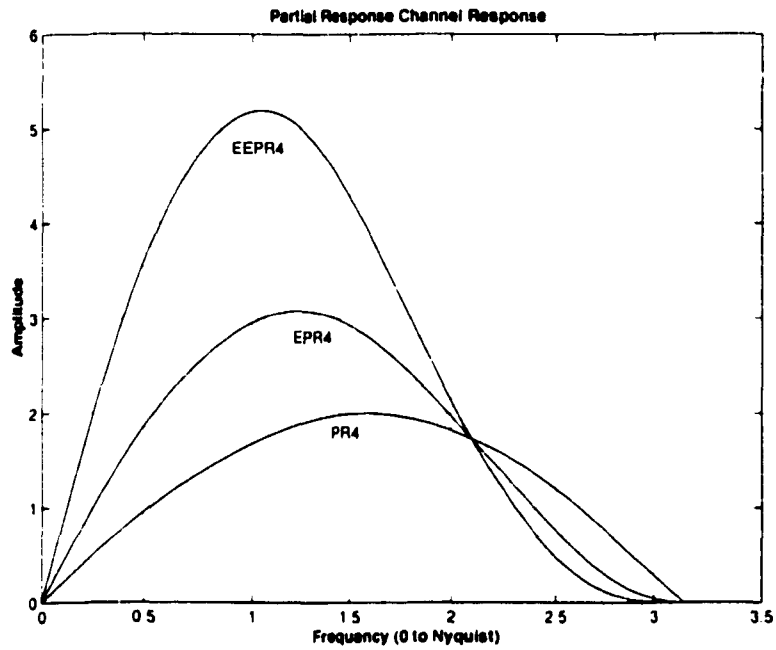

Figure 2.5 Partial Response Shaping

In order to allow the maximum likelihood detector to detect the most possible sequence, an *equalizer* is needed before the detector. It can be in the form of fixed coefficients or programmable FIR filter or an *Adaptive Equalizer* [5][8]. An adaptive equalizer is usually a linear system, usually in the form of FIR (Finite Impulse Response) filter with adaptable coefficients. With adaptive signal processing, correct equalization can be achieved despite channel variation and signal variation.

2.3.2.2 *DFE* system

Another sampling detection method is through Decision Feedback Equalization (DFE) [4, 9, 10, 11, 12]. In a partial response system, ISI still exists but with a controlled amount. The sampled values are detected by the maximum likelihood detector. In con-

trast, the *DFE* system consists of a symbol-by-symbol detector and two equalizers - *Forward* equalizer and *Feedback* equalizer. Both equalizers can be implemented in the form of *FIR* filters. The symbol-by-symbol detector makes the decision and sends to the Feedback equalizer that reverses the partial response made by the encoder. Since noise and other distortions may occur when the signal is read back from the magnetic media, adaptation on the equalizers is required [4].

2.3.3 Other methods

There are still other methods for disk drive read channel detection, such as a Fixed Delay Tree Search method which is explained in [13].

2.4 Signal Equalization

As introduced in the previous section, equalization must be performed to deal with *ISI*. Depending on the detection method, the target functions are different from each other. This section briefly explains two generally used equalizer forms.

2.4.1 Continuous time equalizer

Continuous time equalizer consists of a cascade of $g_m - C$ biquads [2][14]. The weighted sum from the first order derivatives of their output implements the poles and zeros of the equalizer. Usually the position of the poles/zeros can be designed independently. In doing so, the desired function can be implemented. Continuous-time filter can also be used for pre-filtering.

The advantage of these $g_m - C$ biquads is that they are very compact and the constructed equalizer is very small. The formation of the equalizer is also very easy to achieve by just cascading the biquads. However, it may not be easy to adjust after the circuit has been fabricated. Nonetheless, adaptive continuous-time filters are described in [15] and [16]. The use of a off-chip components limits their operation to very low frequency. In a more recent publication, a $g_m - C$ ladder structure can be implemented that

can have adjustable zeros by adjusting the weighted sum of the state variables. However, the frequency response of the transconductors must be much higher than the signal frequency. Therefore, an advanced fabrication process may be required to implement this structure [17].

2.4.2 Discrete time equalizer

Unlike continuous time equalizer, the transfer function of discrete time equalizer can be easily adjusted. The input analog signal is first converted into either discrete analog signal or digital signal, then passed through a FIR filter as an equalizer. The term equalizer can thus be applied to FIR filter as well.

The function of equalizer is formed by arranging the positions of 'zeros' of the FIR filter. The advantage of the discrete-time FIR equalizer is that a well known adaptation algorithm-LMS algorithm, can be used to adjust the filter coefficients. Due to the finite impulse response nature, more filter taps are usually needed to achieve a desired function than the continuous-time counterpart. Therefore the hardware of a discrete-time equalizer is more complicated than a continuous-time equalizer. However, as will be shown in this work, by appropriately combining the analog circuitry and digital circuitry, the final implementation can still be efficient. This dissertation will emphasize on a mixed-signal equalizer system design based on discrete time filter.

2.5 LMS Adaptation Algorithm

LMS (Least-Mean-Square) algorithm is a widely used adaptation algorithm for FIR filter. It provides robust convergence with simple hardware implementation. A FIR filter can be written as:

$$y(k) = \sum_{j=0}^{M-1} x(k-j)W[j] \tag{2.2}$$

where $W[j]$'s are the filter coefficients and $x(k-j)$ is the sampled input signal. The LMS updating equation for the filter coefficients can be expressed as:

$$W_{k+1}[j] = W_k[j] + \beta x_k err_k \tag{2.3}$$

while W_k is the coefficient weight at time k, x_k is input signal at time k, err_k is the error signal corresponding to the time k, and β is the step size that trades off between the accuracy of equalizer and tracking ability. To ensure stability of the adaptation algorithm, β can be set to less than the signal power times the number of taps and is usually far less than 1. The error signal can be written as:

$$err_k = y_{ideal}(k) - y(k) \tag{2.4}$$

When PR signaling is used, the ideal output values are finite discrete levels. The errors are calculated by subtracting the input signal to the detector from the output of the detector.

A simplified LMS algorithm can be written as:

$$W_{k+1}[j] = W_k[j] + \beta sgn(x_k) sgn(err_k) \tag{2.5}$$

which is also called 'sign-sign' algorithm because it uses signal's polarity to determine the updating directions of the filter coefficients. Each updating is either '+1' or '-1'. This provides a much easier implementation by eliminating complicated multiplication. Figure 2.6 gives an example with the ideal output level values equal to '+1', '0', '-1'. The principle is straightforward. First the output is compared with five different thresholds to determine which symbol it belongs to or closest to and whether it is higher or lower than one of the three ideal levels. While the output level is *below* an ideal level, the coefficient should *increase* to compensate for it, and vice versa. The adaptation algorithm used in this dissertation is based on the above *sign-sign LMS* algorithm. In the following chapters, the block to determine where the output signal lies relative to the multi-level thresholds is called a 'slicer'.

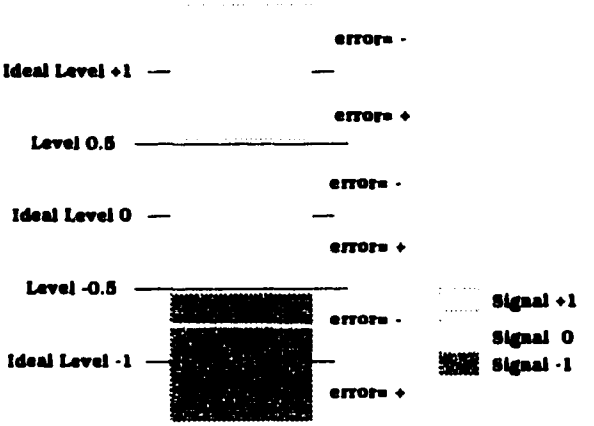

Figure 2.6 Demonstration of Slicer Used in LMS Algorithm

2.6 Summary

This chapter reviews the basics of disk storage systems. The goal is to explain the necessity of using equalizer in the disk read out channel signal processing. Partial response signaling is assumed to be used in this dissertation. When designing an adaptive system, $SS-LMS$ algorithm will be used to perform the updating of filter coefficients.

3 MIXED-SIGNAL EQUALIZER IMPLEMENTATION

3.1 Introduction

Based on the introduction of Disk Drive Read Channel in Chapter 2, discrete time mixed-signal implementation is chosen to realize the equalization block. There are two designs implemented in this work. The first one is a programmable FIR equalizer and the other one is an Adaptive Equalizer. The major difference between them is that the adaptive equalizer has detecting and updating circuits. The coefficients in adaptive equalizer are adjusted by $SS - LMS$ algorithm, which is introduced in Chapter 2.

Since both designs are programmable, they can be used to realize different types of transfer functions. Thus the designs are not limited to disk drive read channel equalization. The reason to choose this specific application is that the disk drive read channel requires relatively less number of filter taps. The main goal of these two designs is to demonstrate the effectiveness of the proposed mixed-signal multiplier, which is the essential part to achieve high speed and low power equalizer design.

This chapter describes the detail implementation of these two designs. It starts from the architecture selection followed by the top-level simulation and the circuit design.

3.2 Equalizer System

During read out of data from the magnetic media, there are many ways to do the signal processing after the analog front end. Shown in Figure 3.1 are several commonly adopted methods. The major concerns of choosing different structures are hardware cost, and feasibility.

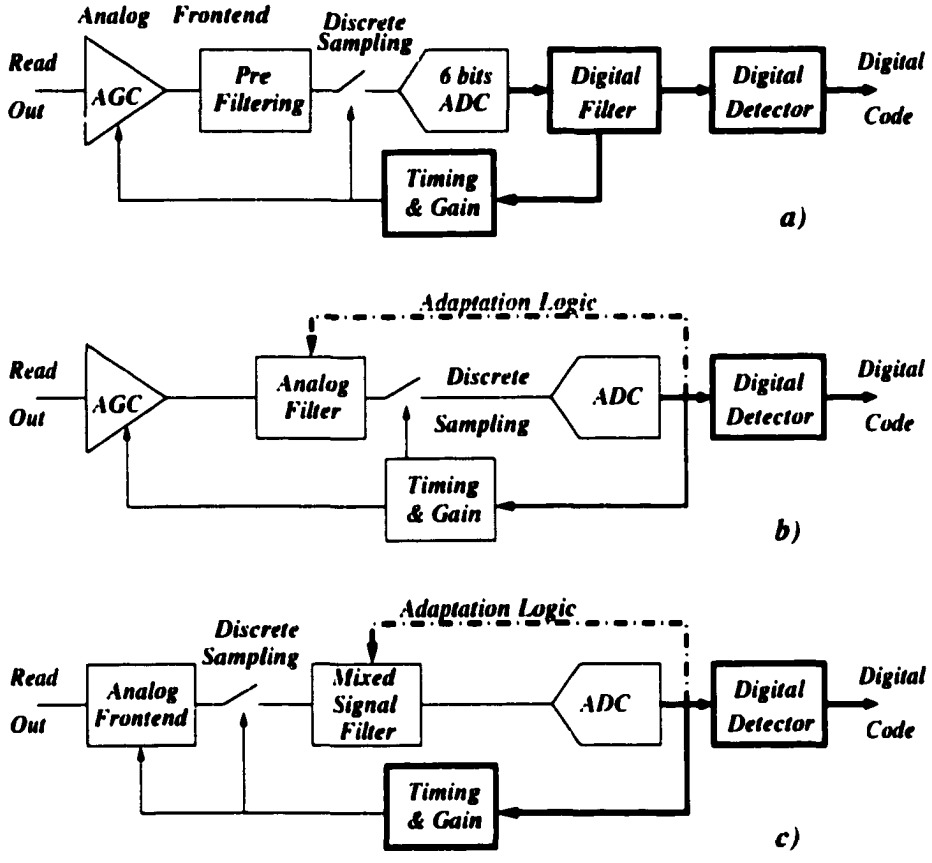

Figure 3.1 Block Diagram of Disk Read Out Channel. a) Digital conversion approach. b) Analog approach and c) Mixed-signal approach

Form a) uses a high speed Analog-to-Digital converter followed the analog front end to convert input signal into digital domain [19, 20]. High speed operation is required for the ADC and the digital filter to guarantee capturing of all the information from the read out signal. Usually a 6-bit ADC is enough [2]. Nonetheless the design of such ADC is challenging because its operating speed is required to be above $500MSPS$ and it is required to maintain the dynamic range of about 5 bits up to the $Nyquist$ rate. Right now, read channel ADC's with speed up to 800 $MSPS$ have been reported [21, 22, 23, 24, 25, 26]. After this ADC, the equalization is done in digital domain. In this approach, the equalizer occupies large area because of the digital multipliers used in the equalizer.

For the second approach, the equalizer is designed in continuous-time analog form which uses a single continuous-time equalizer after the automatic gain control (AGC) circuit [2, 17]. After the analog equalization, the signal is sampled into discrete form by passing through an analog-to-digital converter to do further detection. The analog-to-digital converter will have less requirement compared to the one used in the first method [2]. This design approach aims at high speed and low power dissipation. However, the design of the analog equalizer is difficult. Coefficients storage and updating may poses other challenges when adaptation is needed since the coefficients usually cannot be adjusted using simple adaption algorithm.

In the third approach, the analog front end is followed by a mixed-signal filter which equalizes the signal. The equalizer here is based on a digital filter structure, but the computation is done in mixed signal mode, i.e. the analog signal is multiplied directly with digital coefficients. Since it outputs an equalized analog signal, the requirement for the ADC is much relaxed. A simple 'slicer', or an ADC with low resolution, can be used instead of multiple bit ADC's used in the above two approaches. Furthermore, since this approach eliminates the digital multiplier that occupies very large area, the whole design can be compact and fast. Additional advantage comes from the digital filter structure. It is relatively easier to extend its application by loading new coefficients. This idea stimulates the development of the prototype circuit in this dissertation.

Among all these methods, the detector followed the digital filter in the first approach or the ADC's of the second and the third approach is always implemented in low cost digital domain for its Finite-State-Machine nature [5]. Nonetheless, analog detector is also possible [18].

3.2.1 FIR structure

Most equalizers have the form of a FIR filter. The first step to implement such equalization system is to design the FIR core. There are several ways to implement a FIR filter and the basic components are multiplier and adder as well as the delay unit.

By looking at the FIR expression shown in Equation (3.1),

$$y(k) = \sum_{j=0}^{M-1} x(k-j)W[j] \tag{3.1}$$

the latest sample $x(k)$ is always multiplied with the first coefficient $W[0]$, so on and so forth. Different ways to implement a FIR circuit is shown in Figure 3.2. The structure

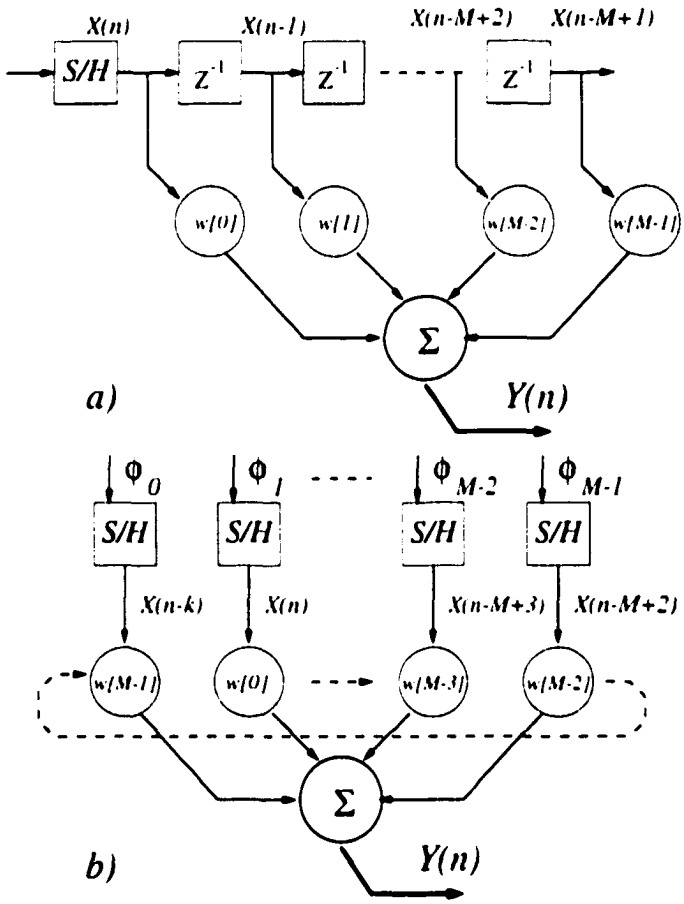

Figure 3.2 FIR Structure Diagram. a) Direct form b) Circular-buffer Form

in a) is a direct form implementation of the above equation [27]. Every new input sample is delayed through a delay line. z^{-1} represents the delay element. In the case of mixed-signal design, this delay element is an analog delay cell. M latest samples are multiplied with the corresponding filter coefficients. The final output is the summation of these products. Notice that only the samples are delayed, the coefficients are stationary. Case

b) is a circular buffer version of the FIR filter [29, 30, 31, 32]. Instead of using the delay line, it uses multiples of S/Hs to sample the input signal at different moments. The sampling phases are non-overlapping and appear once every M cycles. The sample is preserved for M cycles for computing the output until it is updated. Since the latest sample has to multiply with coefficient $W[0]$, two methods can be used to realize this. The first is to use an analog switching matrix which redirects the M latest samples to correct positions [30, 31, 32]. This method adds more components on the analog signal path. More distortion may be therefore introduced. The second method, which will be discussed in more detail later, is to rotate the filter coefficients to align with the correct samples. Figure 3.2b) shows the second solution. This approach minimizes the degradation of the analog signal. However, it may need more digital power to do the rotation.

Comparing these two architectures, when the filter is a mixed-signal design which computes analog signal with digital coefficients, form a) is not a good choice because the analog samples have to be delayed. The implementation of analog delay cell is difficult. In the case of form b), this analog delay cell is avoided. Therefore, the circular buffer type of FIR filter structure is chosen to be used in this dissertation.

3.3 Mixed-signal Multiplier Technique

From the FIR expression:

$$y(k) = \sum_{j=0}^{M-1} x(k-j)W[j] \tag{3.2}$$

the following blocks are needed:

Sample-and-Hold:	samples the analog signal and produces the corresponding discrete signal,
Multiplier:	executes the multiplication,
Summer:	adds up the products,

Rotation-loop: provides the required delays for the filter coefficients.

If adaptation is needed, two more blocks are needed:

Detector: generates error signal for adaptive processing.

Updating circuit: updates coefficients according to the error signal.

In a mixed-signal implementation, every term in equation (3.2) can be carried out in either analog, digital or mixed signal form. In [30], multiplication is done with adjustable transconductance circuit. Since an analog control voltage is used for adjusting the transconductance, the product is obtained by multiplying the analog input with an analog control voltage. Unfortunately, in the design described in [30], the analog control voltage is generated off-chip during testing, which is not a strict monolithic solution. In [14] and [31], the multiplication is done with an analog multiplier. In order to provide the analog coefficients, a set of separate D/AC's are needed to convert digital coefficients into analog domain. To save the number of D/AC's, some commercial products use only one D/AC repeatedly.

Unlike all the previous approaches, in this dissertation, mixed-signal technique is used throughout the structure. A mixed-signal multiplier that directly multiplies the analog signal with the digital coefficients is developed. The difficulty in implementing analog multiplier and the need of extra D/AC's are avoided. Since the coefficients need to be loaded and updated, the most convenient way is to keep them in digital form. The final summation is in discrete-time signal domain so that it can be combined with the mixed-signal multiplier. All of the above lead to a system diagram shown in Figure 3.3. The partition of analog, mixed-signal and digital in this system is clearly shown.

Each front-end S/H generates the signal $x(n-k)$, which is the sampled discrete representation of the analog input. Coefficient blocks represent $W[j]$'s. They are digital signals because the storage, updating as well as delaying can all be easily implemented and achieved a high speed operation in digital domain. Current mode design techniques are used to increase the operation speed of the multiplier.

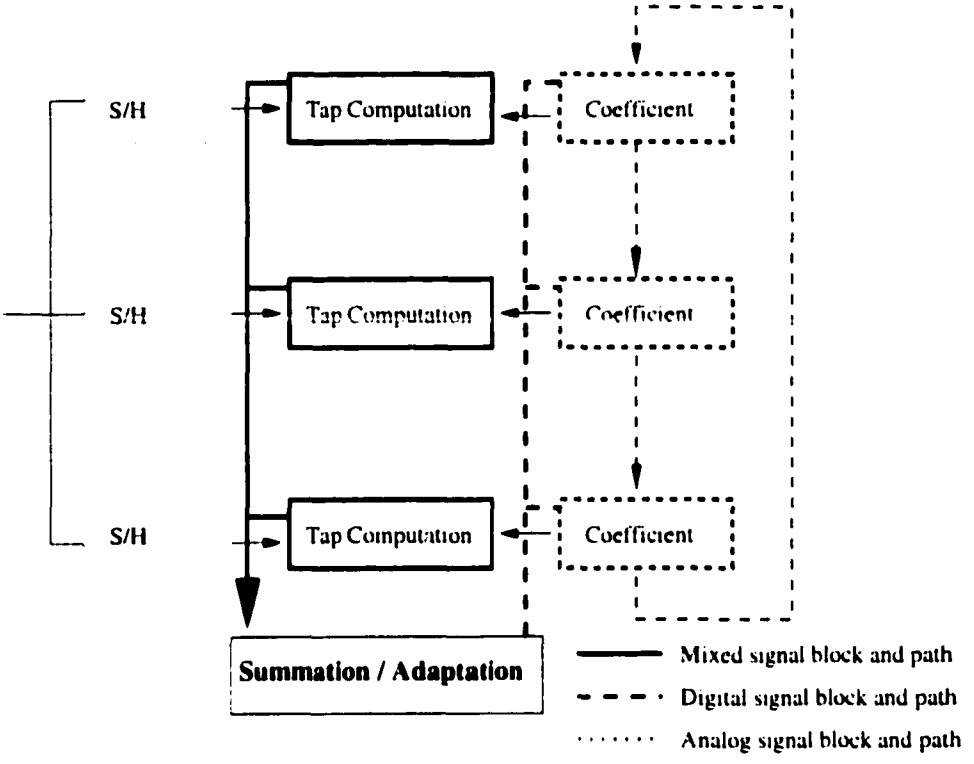

Figure 3.3 Block Diagram of Circuit Design

Among all the blocks, the multiplier is the core part. In this design, the basic cell in the multiplier is the four-quadrant operational switching block that can operate with signed digital number and signed analog signal. This block solves the problem existing in [30] where the coefficient's polarity has to control the analog signal through differential switch connection. This not only increases signal distortion along the data path, but also is hard to implement under high speed operation since it may take a longer time to settle. Moreover, the multiplier introduced in this dissertation not only realizes the multiplication but also provides the final summation.

From the above diagram, it can be observed that this architecture is a multi-channel system in that more than one component (S/H) are used to sample the input analog signal.

3.3.1 Design requirement

The specifications for an equalizer system are speed, number of FIR filter taps and the tap resolution, etc. For modern disk drive read channel, high speed operation is needed to meet the requirement of ever increasing storage density. This work is aiming at the operating speed of over 250 $MSPS$.

The number of taps used in a discrete FIR equalizer is between 4 to 12 taps depending upon different design goals and constraints [27, 30, 31, 32, 33, 34, 35]. Selection on the number of taps depends on the trade off in choosing pre-filtering structure. Usually a linear-phase filter such as an equi-ripple design is used to be the pre-filter so that the number of taps of the FIR filter can be less [36]. Therefore a 6 tap system is used for this design targeting for $PR4$ signaling. Both [33] and [34] use system level simulation methods to verify the validity of the number of taps needed. In this dissertation, a top level simulation environment is set up not only to verify the number of required taps but also to optimize and verify the design.

3.4 System Level Simulation

The importance of system level simulation has long been recognized in top-down hierarchical design flow. It can greatly enhance the efficiency of a designer to avoid possible mistakes in early design stage. Unlike circuit design that involves many details, system level simulation can easily give the designer insight into the performance upper bounds and lower bounds for different architectures. Usually a system level simulation can give results within very short simulation time, which can not be achieved using transistor level simulator, such as $SPICE$. The top level system simulation in this work is done in $MATLAB$.

3.4.1 Goals

The system level simulation performed both before and after circuit design is to reach the following goals:

1) To investigate the validity and the efficiency of the system architecture.

This includes the examination of the validation of number of bits and number of taps, as well as the adaptation algorithm.

2) Provide reasonable system relaxation according to different trade-offs.

Through system level simulation, relaxation and limitation caused by real circuit behavior can be obtained. One example is the inevitable delay of updating control signal caused by signal settling in real circuit. Through high level simulation, a design guideline is provided on how much the delay can be tolerated without affecting the system performance.

3) Provide information on the possible effects due to different error sources.

System-level simulation without error sources can't provide enough information to optimize the design. The important merit of behavioral simulation is the capability of adding one specific error and observing the corresponding system behavior solely due to that error. With this capability, it can provide a clear understanding of the most important error source so that efforts can be put on to solve it or reduce it in real circuit design.

4) Provide an effective tool that can work interactively later on during circuit design and testing.

3.4.2 Simulation setup

The system is setup according to the disk drive read channel shown in Figure 1.1. A random binary code generator is used to generate the user data. Then the data is passing through the precoding stage. After that the data sequence performs a convolution with the read channel impulse response. The output is a good mimic of the read out signal.

To simplify the simulation of the read out process, the front end pre-amp and certain filters are omitted since they do not change the characteristic of the signal significantly. The signal is directly sent into the equalizer for optimal partial response equalization. After the equalizer, the signal is plotted for reviewing different effects. The sign-sign LMS algorithm is used as the adaptation algorithm since it is used in the actual circuit implementation. To validate the correctness of the equalizer, the equalized data is further sent into a Viterbi decoder to recover the original binary data. A general way to validate the effectiveness of the equalization is by observing the eye diagram. A more accurate way is to exam the cross-correlation of the recovered data with the original binary data.

In order to simulate the white noise added during the read out of the data, an $AWGN$ (Additive Wideband Gaussian Noise) having amplitude of +/-5% of the signal amplitude is added to the read out signal. Furthermore, to simulate the possible comparator errors in circuit design, a 25% chances of random decision error is added.

To simulate the actual circuit implementation, each coefficients has only 6 bits of resolution in the simulation. Since there are S/Hs at the front end, the input data can not have infinite resolution due to sampling and settling error. Therefore, to imitate the S/H mechanism, infinite precision number is not appropriate to represent the sampled data. In the top level simulation, 8 bit resolution is used for the analog input signal.

Since in the actual circuit implementation, it needs time for processing, the updating of the coefficient has at least two cycles of delay. At cycle *Zero*, a new sample X_0 is brought for computation. A series of comparators have to sense the output near its settling at the end of the cycle *Zero*. During cycle *One*, comparator results are obtained and error signal is generated after certain delay. This error decision signal (up or down) is then latched and input into the updating circuitry. This latch clock is conveniently chosen to have the same phase as main clock. At cycle *Two*, the latched error decision signal is used to calculate new coefficient. From the above, the minimum delay is two cycles from the instant that X_0 is sampled to the instant that the corresponding coefficient is updated. So that there will be eight filter taps which include the six taps used

to do the filtering while two extra taps in resetting mode to achieve the two cycles delay.

Here is the summary of the system setup:

1. Totally eight filter taps in which six taps are used in the computation.
2. Each tap has a coefficient of 6-bit accuracy.
3. System contains $AWGN$ and comparator noise.
4. Adaptation is based on sign-sign LMS algorithm.
5. Updating procedure has two cycles' of delay.
6. Provides slicer for PR and EPR signaling (error decision generator). A PR slicer has 5 comparators targeting a three level output signaling, and a EPR slicer has 9 comparators targeting a five level output signaling.
7. Provides various control functions.

3.4.3 Simulation result

A series of results are shown from Figure 3.4 to Figure 3.6. Figure 3.4 is the simulated read out signal to equalizer. It is the result of the convolution of the user data sequence a_n (refer to Figure 1.1) and the simulated disk channel impulse response [4]. The channel response can be written as a "Modified Lorentzian" family where the response to a step function is:

$$s(t) = \frac{1}{1 + (2t/PW_{50})^{2x}} \tag{3.3}$$

where 'PW_{50}' indicates the pulse width measured at 50% of the pulse amplitude. It is usually referenced to the period of one writing clock cycle, T. The larger the number of PW_{50}, the higher the density and the more stringent the requirement of equalization would be. Modern disk system requires PW_{50} between 2.3T to 2.8T. This number is set to be 2.5T in this simulation. 'x' is a parameter usually chosen to be 1. This function is used in most cases when analyzing disk drive read channel signal response.

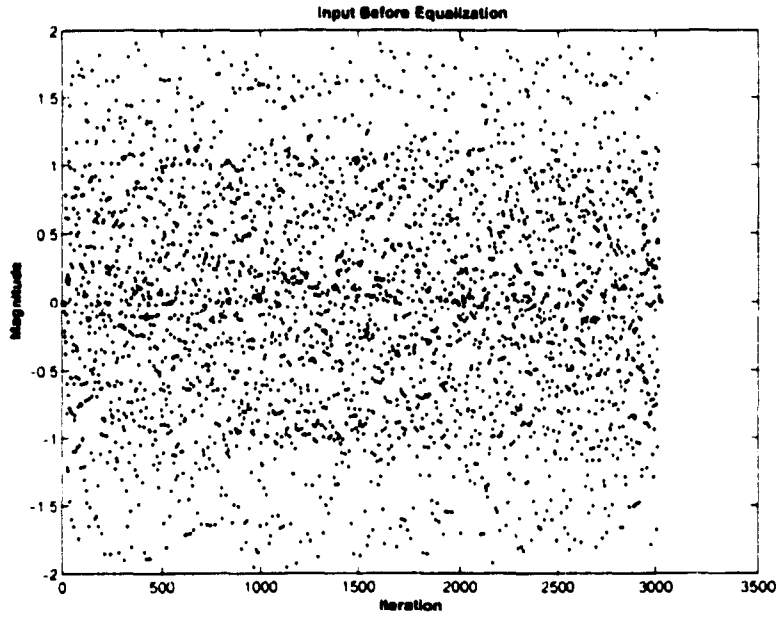

Figure 3.4 Sampled Signal Before Equalization

Figure 3.5 is the equalized result for $PR - IV$ signaling. It is clear that a seriously distorted input has been shaped into a ternary signal with three separate levels.

Figure 3.6 shows the corresponding equalized result for $EPR - IV$ signaling which has five levels.

By analyzing the simulation result, the adaptive algorithm is shown to be robust under the influence of channel noise and comparator decision errors. The results also provide guidance to circuit design. The simulations also show that coefficients with 6 bits of resolution are sufficient for a disk drive read channel. Also, six computation taps are enough for $PR - IV$ and $EPR - IV$ shaping.

Finally, to examine the validity of the algorithm, a Viterbi decoder is used to decode the equalized data to recover the original data. After doing a cross correlation as shown in Figure 3.7, it can be checked if the read back data is the same sequence as the original input user data. It is observed that when two sequences are exactly identical and overlapped, the summation of the correlation is a large value. When they are not exactly overlapped, the result turns out to be summation of random data and has a

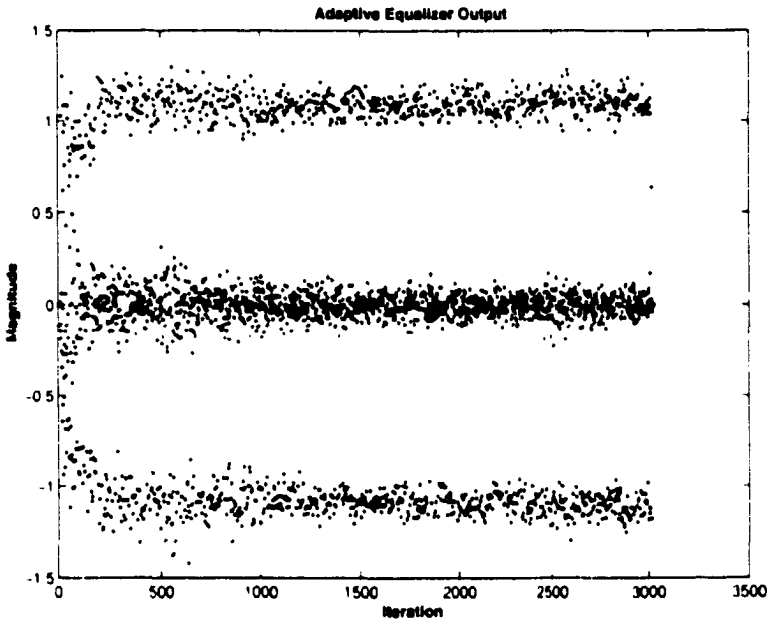

Figure 3.5 $PR-IV$ Channel Shaping Output

value of zero or near zero. If the two sequences are not the same at all, the correlation is near zero mean no matter how they are overlapped.

Several observations from system level simulation:

1) Ideal updating case: all coefficients are updating once every cycle. This setup has the best result compared to other setups, i.e. it has the fastest convergence speed and the largest-opening eyes (i.e. lowest Bit-Error-Rate).

2) When updating only one coefficient per clock cycle, similar result as that in 1) is achieved.

that of 3) When updating only one coefficient per clock cycle and with two cycles updating delays, the result is still acceptable in that convergence is reached after a relatively longer time when compared to that of the first case.

4) 6 taps of filter coefficients and 6 bits of resolutions meet the convergence requirement in situations 1) to 3).

5) By comparing the output from the Viterbi decoder and the original data, it shows that the shaped channel achieves correct result.

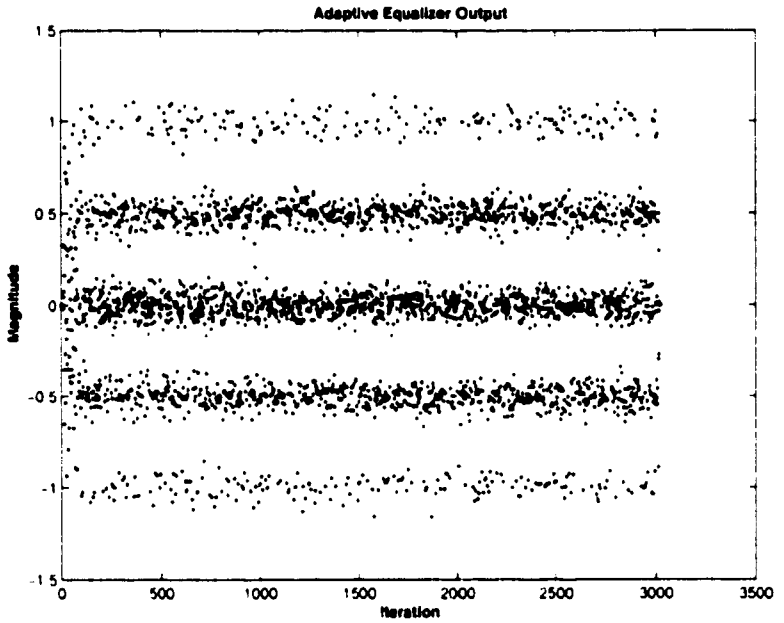

Figure 3.6 $EPR-IV$ Channel Shaping Output

6) The system architecture is robust enough to bear $AWGN$ and very large percentage of adaptation mistakes.

7) The convergence speed can be adjusted by changing the coefficient β in the LMS algorithm (equation (2.3)).

3.5 Circuit Design

After the system level simulation of the system architecture, the next step is to build each block according to the partition in Figure 3.3. The circuits were designed based on a 0.5μm CMOS technology. This technology provides three layers of metal, linear capacitor and resistor. The minimum drawn gate length is 0.6μm.

3.5.1 Current mode implementation

Circuit design can be carried out either in voltage or current mode. A voltage mode design will have difficulties for some operations computation such as subtraction and summation. Analog voltage is also difficult to be multiplied. A current mode design can

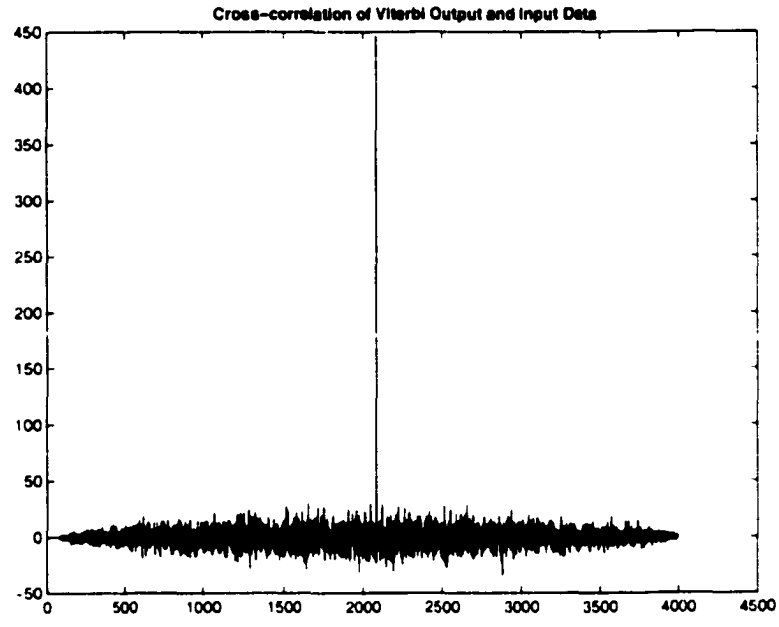

Figure 3.7 Cross Correlation of Viterbi Output with Input Data

avoid most of these problems. Multiplication, subtraction or summation in the current mode can be rather easy to realize. Therefore, current mode implementation is adopted throughout the design except at the output and coefficient updating stage.

3.5.2 System overview

The basic equation that describes a FIR filter with M taps of coefficients can be rewritten as:

$$y(k) = \sum_{j=0}^{M-1} x(k-j)W[j] \tag{3.4}$$

where $x(k)$ and $y(k)$ are the input and the output signals, respectively. $W[j]$'s are the filter coefficients. To compute the output value, delayed values of $x(k)$ have to be provided. Since $x(k)$ is a sampled analog signal, implementation of analog delay stages for the delayed values of $x(k)$ is often difficult, and the delay stages may degrade the analog signals. To address this problem, circular buffer structure has been proposed [29] and has been used in different designs [30, 31, 32]. In this architecture, an array of M numbers of sample-and-holds (S/Hs) is used for sampling the input signal sequentially

and periodically. The sampled inputs are then fed through a switching matrix to an array of multipliers, which multiply the delayed inputs with the corresponding filter coefficients. Since analog delay stages are not used, degradation of the analog signals is minimized. For the architecture proposed in this thesis, an alternative method to implement the circular buffer structure is used as shown in Figure 3.8.

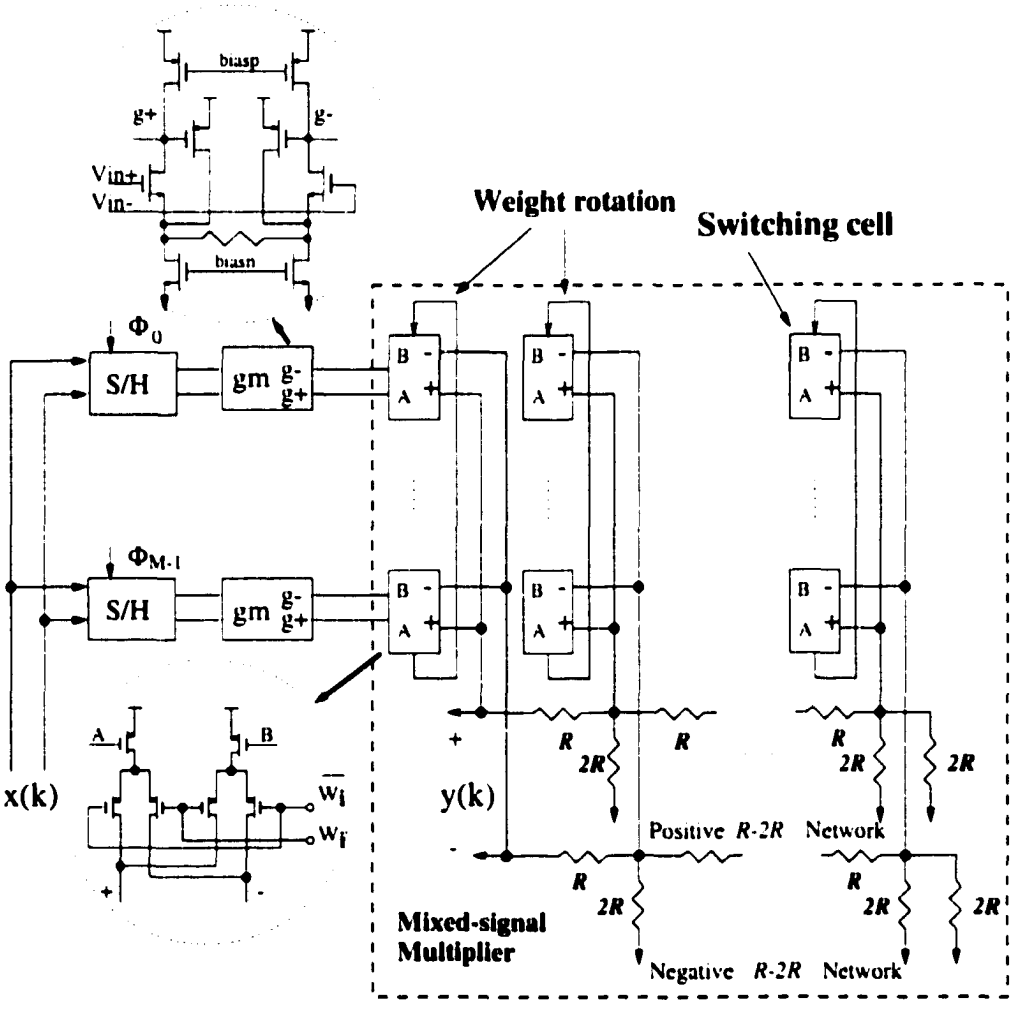

Figure 3.8 Proposed Analog FIR Filter Architecture

Each filter coefficient that is stored in the digital registers inside each row of switch cells is rotated circularly from one row to the other. The analog signals stored in the S/Hs are connected directly, without going through any buffer stages, to the mixed-signal multipliers (ie. transconductor stages, g_m's). The advantage of this structure is

that it minimizes the degradation of the analog signals by not having the signals going through the buffers and the switching matrix, which may also affect the settling time of the signals at the multiplier inputs. The disadvantage of this structure is that it increases the digital power consumption because of the number of bits involved in the rotation of the digital filter coefficients. However, the digital power becomes less significant in more advanced CMOS processes.

To relax the settling time requirement of the S/Hs, one can increase the number of taps from M to $M + L$ with L being the number of digital coefficients set to zero. These zero coefficients are rotated to the taps, which have their S/Hs sampling the input signals. Since the digital coefficients are zero, the output of these S/Hs will not affect the output $y(k)$. Hence, it allows these L sampling S/Hs to have longer acquisition and hold settling times than one sampling period. As a result, the S/Hs will not limit the sampling rate. In the real circuit design, L is set to two and a special signal named *Skip* is used to make these two coefficients out of computation by forcing them to zero.

Similar to other circular buffer structures, the proposed architecture requires matching between transconductors as well as matching between S/Hs. These mismatches and the offsets of the S/Hs and transconductors will result in spurs on the output spectrum. Furthermore, timing errors on the sampling clock phases of the S/Hs also produce unwanted modulations with the input signal. All these effects can be minimized by careful layout.

3.5.3 Multiplier

To achieve direct multiplication between the analog inputs and the digital filter coefficients, a mixed-signal multiplier is proposed. This technique can also be applied to continuous equalizers. Assume that a digital coefficient $W[j]$ can be expressed as:

$$W[j] = \sum_{i=0}^{N-1} [W_i[j] - \overline{W_i[j]}]2^i \tag{3.5}$$

where $W_i[j] \in [0,1]$ and is the ith bit of $W[j]$. Using Equations (3.4) and (3.5), the output signal y(k) can be rewritten as:

$$y(k) = \sum_{i=0}^{N-1} 2^i \sum_{j=0}^{M-1} [W_i[j] - \overline{W_i[j]}]x(k-j) \tag{3.6}$$

The above equation suggests that $y(k)$ can be obtained by first computing the sum of the products of $x(k-j)$'s and $W_i[j]$'s, and then computing the binary weighed sum of the results obtained from the previous step. Each transconductor shown in Figure 3.8 is assumed to have multiple outputs that are distributed inside the switch cells of each row. The multiplication of $x(k-j)$ and $W_i[j]$ can be simply realized by representing $x(k-j)$ as a differential current using the transconductor stage. This current is switched to the positive and negative outputs of the switch cells according to $W_i[j]$. The output currents of all the cells in the same column are then added together yielding the sum of the products of $x(k-j)$'s and $W_i[j]$'s at column i. The resultant differential current is fed to the ith input of the $R-2R$ network, which realizes the required binary weighted sum in Equation (3.6). The $R-2R$ network also acts as a current-to-voltage converter such that the output value is an analog output voltage.

The operation of a single bit multiplication is obtained in the switching cell, which is shown in Figure 3.9. It can be described using the following equation:

$$[(W_i[j]x_+(k) + \overline{W_i[j]}x_-(k)) - (\overline{W_i[j]}x_+(k) + W_i[j]x_-(k))] \tag{3.7}$$

where $x(k) = x_+(k) - x_-(k)$ is the differential input current.

3.5.4 Design of Transconductor

The operational transconductance amplifier (OTA) is one of key parts of the FIR core. The requirement is to be able to operate at high speed. The basic part of the transconductor is shown in Figure 3.10 [28]. The transconductance for this structure is equals to $g_m = 1/R$. The current mirror formed by M_3 and M_4 provides constant currents through the input transistors pair M_1 and M_2 so that the V_{gs} drop on the

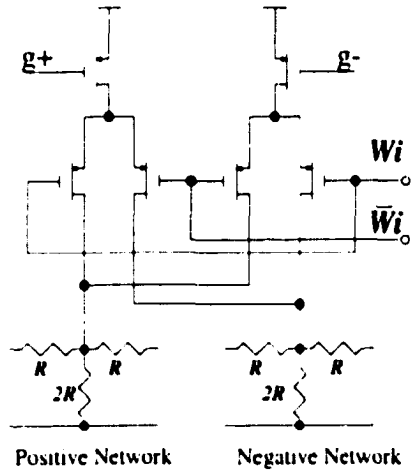

Figure 3.9 Switching Block.

M_1 and M_2 are constant. The two $PMOS$s provide a feedback loop that makes the nodes connecting to the resistor to have low impedance. The input differential voltage is reproduced across the resistor. The resistor acts as a Voltage-to-Current converter to have a transconductance of $1/R$. As a result, resistor matching determines the matching between transconductors. In addition, if poly resistors are used, the transconductors will have good linearity. An advantage of this structure is that it can be modified easily to have multiple outputs by simply including more output transistors. Instead of having current outputs, the gate voltages of M_5 and M_6 which sense the current converted by the resistor are used as output. The gate voltages of $g+$ and $g-$ are connected to the gates of the $PMOS$ transistors in the switching cell in Figure 3.9.

One factor that limits the speed of the operation of OTA is the settling time requirement on the gate voltages of M_5 and M_6. Since large output current level is required in the switching cell, large $PMOS$s in the switching cells are preferred. However this will slow the settling of the OTA. Thus a trade-off is needed to make between speed and output current levels.

To minimize the propagation signal delay through the R-2R network, a segmentation approach is used. The transconductor output currents that are connected to the two

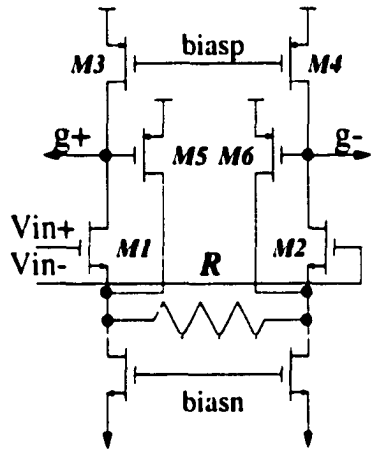

Figure 3.10 OTA Structure.

MSBs of the filter coefficients W(j)'s are scaled so that the two most significant binary weighed sum coefficients in equation (3.6) are partially realized by increasing the sizes of the PMOSs in the two MSB switch cell columns. As a result, the number of sections in the R-2R network is reduced from 6 to 4, and the propagation delay from the last section to the output is minimized. On the other hand, if the R-2R network is eliminated completely, the PMOSs in the switching cells have to be scaled in a binary fashion. As a result, the nodes g+'s and g-'s of the transconductors will take a long time to settle due to the increase in loading capacitance. Thus, the number of R-2R sections has to be selected appropriately. Since the output of the R-2R network cannot drive large off-chip loads directly, on-chip source followers as output buffers are used for buffering the output signal.

3.5.4.1 Design of Up-Down Counter

As the last issue in the *MDAC* design, the digital coefficient used by each tap is stored in a ten bit *DFF* register array. Generally, a 6-bit computation is enough for disk drive read channel application. However, since this design has to be compatible with the adaptive equalizer, the actual bits stored need to be more. Considering the adaptation algorithm is sign-sign *LMS*, each time either '1' or '-1' is needed to be added to the

coefficient. It is easy to do the updating with a up-down counter which will increase or decrease by one LSB at the next clock cycle [11]. The $SS-LMS$ algorithm equation can be written as:

$$W_i[n+1] = W_i[n] + \beta sgn(x(n))sgn(err_n) \tag{3.8}$$

where the β is the control factor usually less than 1. The value of β determines the convergence speed and also the accuracy of the coefficients. The smaller the β, the slower the convergence speed and smaller variation of the coefficients, and vice versa. Implementing the scaling factor β separately is difficult and complex, so it is better to combine β with the Up-Down counter. Thus the counter has a total number of 10 bits with only the first 6 MSB being used for multiplication. Since the adaptation of the coefficient is by adding or subtracting 1 in the 10 bit counter, β is equals to $2^{-4} = 0.0625$. In the real circuit, a separate switch is used to choose between two different β values, i.e. switching between $\beta = 0.25$ and $\beta = 0.0625$. The reason to choose between two different values is to have faster convergence during tracking while have small variation during normal operation. This mechanism is also called $gear-shifting$, and has been verified in the top level simulation.

The implementation of the Up-Down counter is straightforward. A 10 bit DFF array is used and works synchronously. The counting action is performed by flipping corresponding bit/bits according to the present output of DFFs and the up/down signal. A logic block will determine what is the input of every DFF in the next clock cycle. When certain bit is changing, the input of that DFF is the flipped version of the present output, otherwise the input will remain the same as the present output.

For increasing sequence, ie.

$$0000000000 \rightarrow 0000000001 \rightarrow 0000000010 \rightarrow 0000000011 \ldots$$
$$\rightarrow 0101010111 \rightarrow 0101011000 \ldots.$$

It can be observed that the bit i is changed only when all the lower bits are '1'. It can

be written as :

$$Q_i^{n+1} = Q_i^n \div f_1(Q_{i-1}^n, Q_{i-2}^n, ...Q_1^n Q_0^n) \tag{3.9}$$

where $f_1()$ is a logic operation that produces '1' only when all the lower bits are equal to '1'. Symbol '÷' is the XOR operation. A similar step happens to the count down sequence, ie.

$$1111111111 \rightarrow 1111111110 \rightarrow 1111111101 \rightarrow 1111111100 \ ...$$
$$\rightarrow 0101010000 \rightarrow 0101001111 \ ....$$

It can observed that the bit i is changed only when all the lower bits are '0'. To write this in the form of equation:

$$Q_i^{n+1} = Q_i^n \div f_2(Q_{i-1}^n, Q_{i-2}^n, ...Q_1^n Q_0^n) \tag{3.10}$$

where $f_2()$ is a logic operation that produces '1' only when all the lower bits are '0'. These two functions $f_1()$ and $f_2()$ are identical given that $\overline{Q}$ is used instead of Q in equation (3.10). Thus the same logic block can be used to count up or count down with another function block providing Q or $\overline{Q}$, which is:

$$F = Q * Up + \overline{Q} * Down \tag{3.11}$$

The whole schematic of up/down counter is shown in Figure 3.11.

3.5.4.2 Coefficients loading and output

In order to be able to load coefficients in to the filter, several 6-bit multiplexers ($MUXes$) are implemented. Since the coefficients will be rotated in the filter, they can be loaded in serial mode. Since dynamic DFF is used for saving the area and power, asynchronous loading is not possible because there is a minimum speed requirement on the dynamic DFF [38]. Thus the coefficients rotation loop must work synchronously with loading. All these operations require a precise timing.

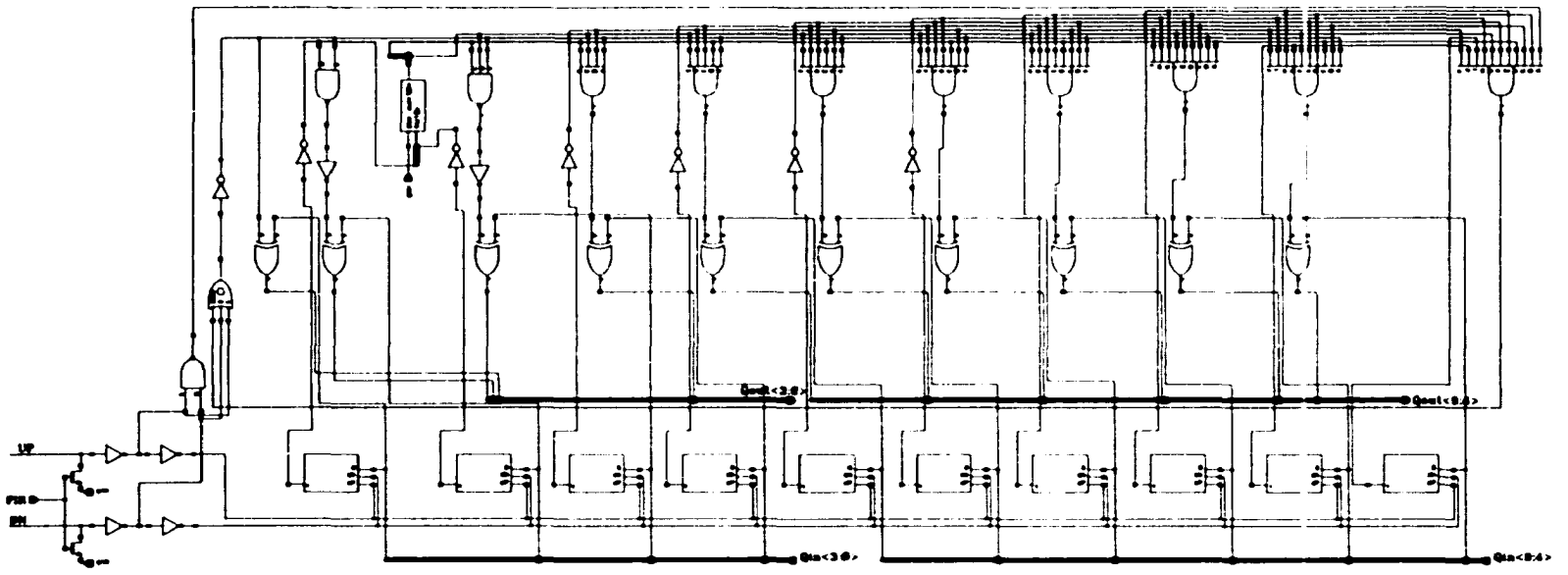

Figure 3.11 Up-Down Counter

In the test chip, two sets of pre-calculated coefficients are stored in two on-chip *ROM*s. Another block is designed to be able to accept coefficients off-chip. In this block, static *DFF* cells are used.

When the coefficients after adaptation are read out from the chip, it has the same serial manner as loading. To drive off chip loads such as the probes of a Logic Analyzer or an Oscilloscope, a driver chain is provided for each bit. These drivers are embedded within the bonding pad in the layout.

3.5.5 Design of Sample and Hold

In this application, the main requirement on the *S/H* is its settling time. The *S/H* is relatively simple and is shown in Figure 3.12. It uses a NMOS switch and a sampling capacitor of $250fF$ [29]. A dummy NMOS is added to reduce the clock feed through effects.

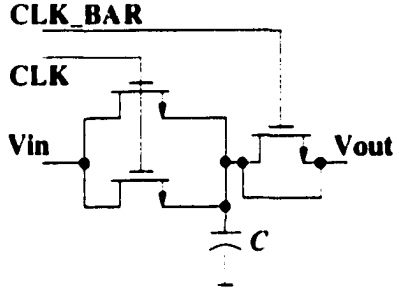

Figure 3.12 Structure of Sample and Hold.

3.5.6 Clock Generator

The requirement for clock generator is to have eight non-overlapped output phases to control the sampling instants of the *S/H*s. These phases are generated using a shift register chain shown in Figure 3.13. Shown in the graph has eight bit output. The principle of this clock generator is that the input of the shift register is '1' only when the first 7 output bits are '0's. This function will make sure that only one '1' is existed in

this chain. The existence of the logic '1' output will last for only one clock cycle so that no timing error will be caused. By going through the following example, the operation of this clock generator can be clarified. Assuming the initial state as '01010101', this initial state will go through the shift register as:

$$01010101 \rightarrow 00101010 \rightarrow 00010101 \rightarrow 00001010 \rightarrow$$
$$00000101 \rightarrow 00000010 \rightarrow 00000001 \rightarrow 10000000 \rightarrow 01000000 \rightarrow$$

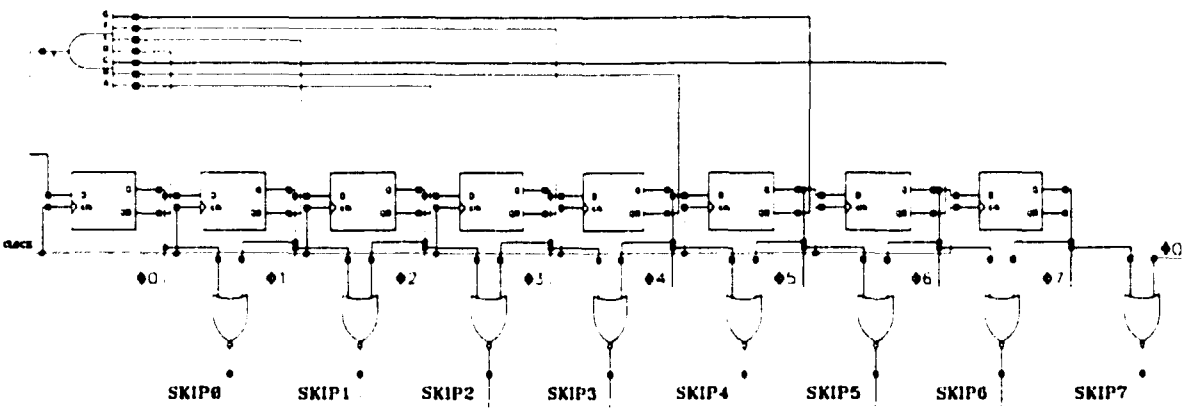

Figure 3.13 Shift Clock Generator.

This shiftclock generator is also used to generate the '*Skip*' signal. Since the present sampling signal will not be used in both present clock cycle and the next sampling cycle, the 'Skip' logic function can be derived as:

$$Skip_i = \overline{Q_i + Q_{i+1}} \tag{3.12}$$

where Q_i and Q_{i+1} represent sampling control signals of the current tap and the next tap respectively.

3.5.7 Comparator

To generate an error signal used in the *SS-LMS* algorithm, comparators are needed. In top level simulation, it is already been shown that the design requirement on comparator is rather loose. The adaptation algorithm can tolerate 25% comparator decision error. The designed comparator is shown in Figure 3.14. It consists a preamp stage and a

latch stage. The preamp provides voltage gain and minimizes kick back effects [28]. When the control signal goes low, the comparator output is regenerated to either logic '1' or logic '0'.

3.6 Programmable FIR Filter

Building blocks discussed so far are used to construct the *FIR* filter system. In order to verify its performance, coefficients for both *PR* and *EPR* are programmed in the on-chip *ROM*s. The circuit simulation is done with *HSpice*. Capacitor load for simulating the actual test condition is added. The *HSpice* simulation shows the *FIR* filter is capable of operating over 330 MHz. One simulation result at 250MHz is shown in Figure 3.15.

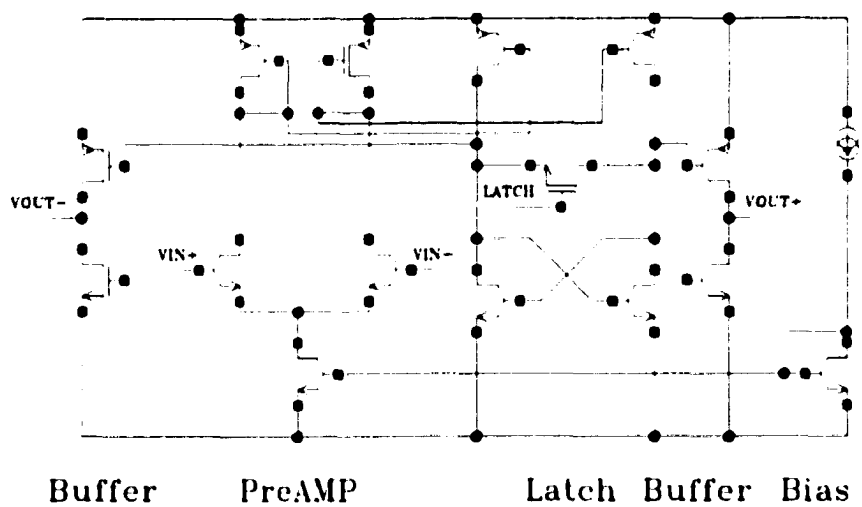

Figure 3.14 Comparator Cell.

3.7 Adaptive Equalizer

Adaptive equalizer is different from *FIR* design. It has more circuitry to obtained the 'slicing' and updating function. Because of area constraint, updating coefficient one at a time is used. It has been validated in the system level simulation.

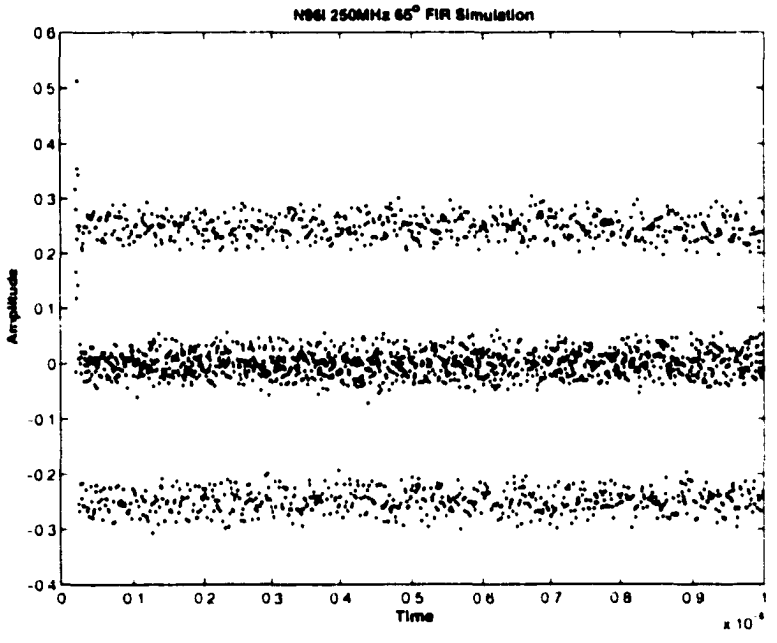

Figure 3.15 FIR Simulation Result with PR Shaping

Three major extra blocks are added into the FIR circuit. The first component is a slicer that consists of five comparators, latches and logic units to provide the error signal. The second component is an up/down counter. The third extra component is a single comparator that provides the polarity of the input signal. Figure 3.16 shows the addition of the two main blocks.

Updating procedure can be processed in parallel, indicating that all coefficients can be updated at the same time. Since the up/down counter occupies large silicon area, this method is replaced by doing the updating in sequential mode, indicating that only one coefficient is updated for each clock cycle. According to Figure 3.16, while the coefficients are in rotation, they are sequentially moved in and out of an up/down counter which adjusts the output value according to the error signal. The error decision block has also two cycles' delay from the instant that the sampled input is valid to the instant that the consequent error decision is used for updating. This delay has been considered and verified in the system level simulation. One drawback of adding more circuitry is that the synchronization among different circuits becomes harder. The direct result is that

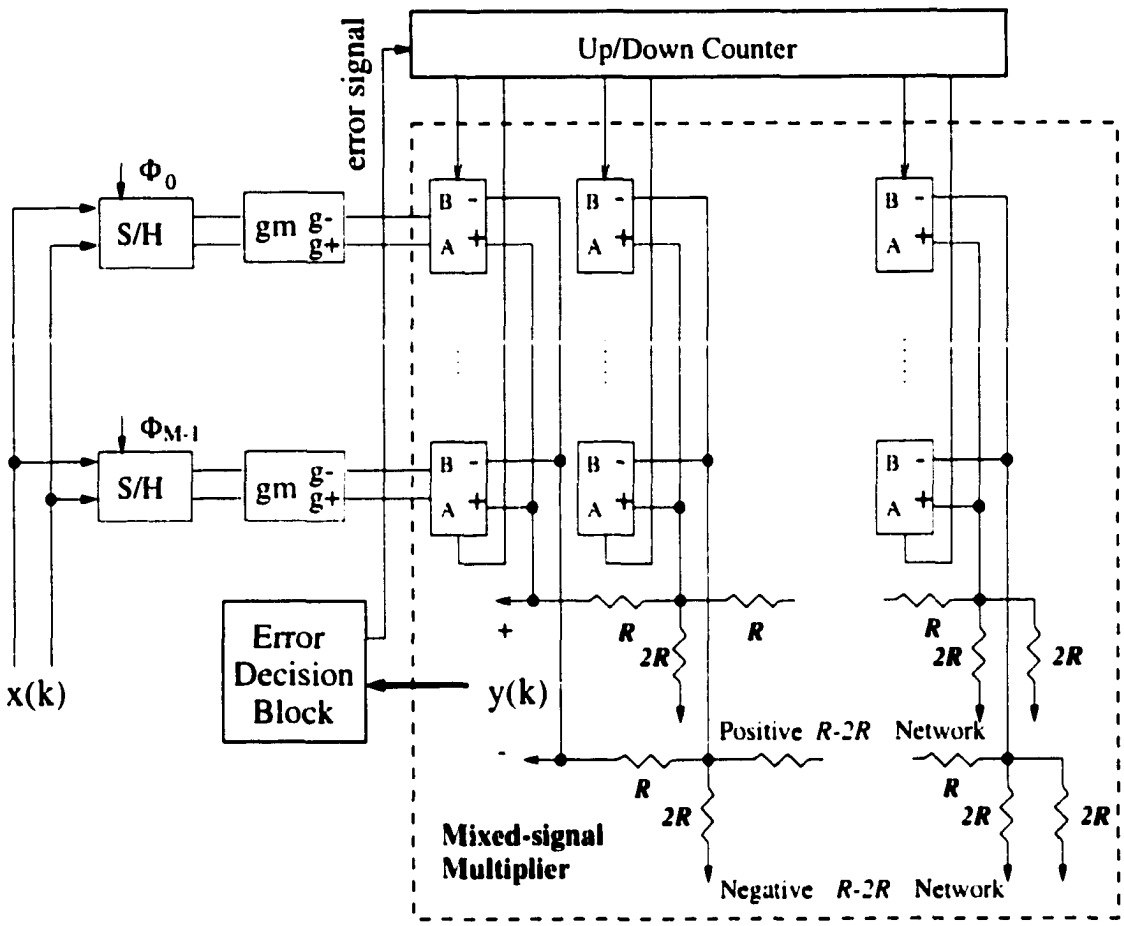

Figure 3.16 System Diagram of the Adaptive Equalizer

the operating frequency is lower than that of the FIR filter.

In order to verify the circuit simulation result, C and $MATLAB$ programs are written to extract $HSpice$ simulation result to provide the output that can be compared with that comes from system level simulation. The extracted result from $HSpice$ is cross-correlated with the result from the top level simulation with the same input sequence. This validates the correctness of the circuit design.

Figure 3.17 is the simulation results of the adaptive equalizer from $SPICE$ simulations with $MATLAB$ doing data processing. It shows that the output samples are converged to the desired three levels correctly.

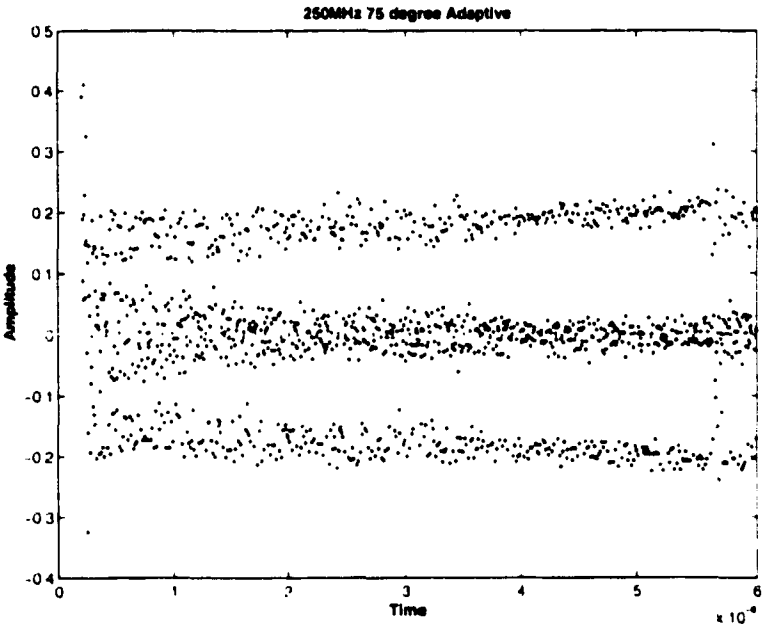

Figure 3.17 Circuit Simulation Result of Adaptive Equalizer

3.8 Summary

Two high speed mixed-signal circuits, programmable FIR filter and adaptive equalizer, were designed using a $0.5\mu m$ CMOS technology. Current mode technique is applied in the circuit design. These two circuits can be used to equalize the disk drive read channel signal, or used as stand alone filters. A special mixed-signal multiplier is proposed and is used as the core of the FIR structure. *HSpice* simulation verifies the functionality of these two designs at a sampling rate of up to 300 MHz operation.

4 EXPERIMENTAL RESULT

4.1 Overview

Based on the proposed structure of the mixed-signal multiplier, two prototype chips were designed and fabricated using a 0.5μm CMOS technology. In order to demonstrate the high speed operation of the proposed mixed-signal approach, the targeting speed was pushed beyond 250 MHz. The first chip is a programmable FIR filter with coefficient loading. The second chip is an adaptive equalizer. Both chips use the same FIR filter core. The adaptive equalizer is constructed by adding an updating functional block into the core. The die photos of both chips are shown in Figure 4.8 and Figure 4.9 at the end of this chapter.

In this chapter, the test setup for testing the prototype chips is presented first. A brief discussion of PCB design issues is also included. The majority of this chapter focuses on the detailed testing methods and testing results of the prototype chips.

4.1.1 Test setup

The test setup is shown in Fig. 4.1. Similar to many IC testing, it consists of the DUT (Device-Under-Test), signal sources and measuring equipment. The DUT is the 40-pin DIP packaged prototype chip. There are 3 types of sources: (1) power supply, (2) input signal generator, and (3) clock signal generator. Unlike typical ADC testing that only requires pure sinusoidal signals as the input signals, testing of the equalizer not only requires pure sinusoidal signals but also requires a special data sequence generator that mimics the corrupted hard disk drive read out signal at a rate over $300MS/s$. This

later requirement can be provided from a high speed arbitrary waveform generator at rather high cost. To overcome this difficulty, an additional chip is designed in order to generate the required data source at a speed up to $360MS/s$. It consists of a 10-bit 256-word ROM with high driving capability. By connecting a commercial high speed D/A converter at its output, the required signal source can be obtained.

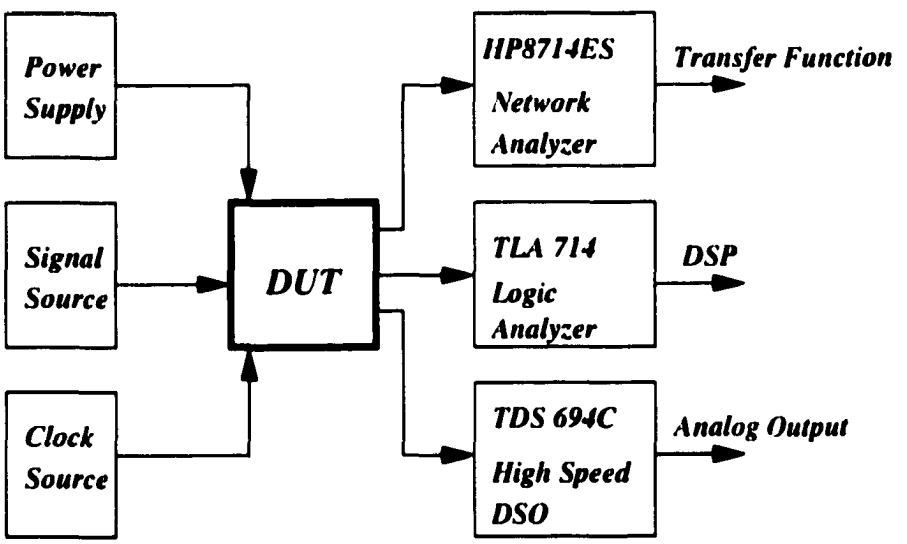

Figure 4.1 Block Diagram of Testing Setup

On the measuring equipment side, Network Analyzer can be used to measure the transfer function of the FIR filter. Logic Analyzer and high speed Digital-Sampling-Oscilloscope (DSO) are used to analyze the system dynamic performance. These equipments can be used separately or jointly.

4.1.2 Printed circuit board

PCB (Print Circuit Board) plays an important role in IC testing, especially in high speed IC testing. In this dissertation, all PCBs are made from two side copper clad board using a milling machine. This machine provides modest accuracy with rather low fabrication cost. The significant advantage of using milling machine is the flexibility of making changes. It is possible to make modification even after the board has been

soldered. One of the testing $PC'Bs$ is shown in Figure 4.2. When designing $PC'B$ for high speed testing, noise distortion is the biggest issue. A lot of effort is put on reducing the coupling among clock paths, signal paths and power supplies. Micro-stripline impedance controlling is used to reduce the signal reflection and to improve the signal quality [39].

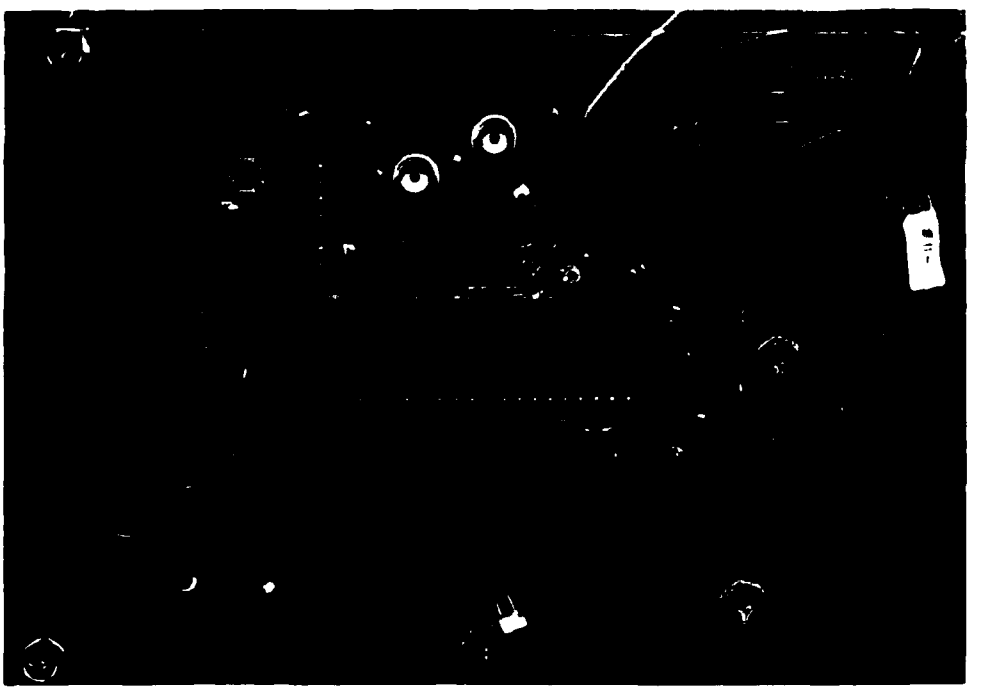

Figure 4.2 Test $PC'B$ Top Plate

4.2 FIR Filter Test

The goal of testing the FIR filter is to demonstrate the functionality of the proposed mixed-signal multiplier. The basic requirement for a FIR filter in many application is its linearity. The FIR filter should remain linear for different coefficient values that are loaded to the filter, the frequency of the clock signal and the input signal with adequate magnitude. The test is organized as follows: (1) *linearity test*: It shows the static transfer characteristic of the FIR filter. (2) *single tone test*: The response of the FIR filter to a single sinusoidal wave input. It shows the linearity from dynamic point

of view. (3) *dual tone test*: Using two equally sized sinusoidal signals as input to observe the inter modulation products. The result can be used to determine what signal level should be chosen to achieve a desired intermodulation ratio [28]. (4) Finally, the FIR filter programmability test. This test measures the transfer functions of different sets of coefficients. This final test demonstrates the overall functionality of the FIR filter as a programmable filter, which can be used as the equalizer in disk drive read channel.

4.2.1 Linearity

In this test, a sinusoidal signal with various amplitudes is injected into the filter. The output is a sinusoidal signal. It is plotted versus the input signal to examine the linearity of the filter. Since the input signal is a single frequency sinusoidal wave, no other frequency components should generated if the system is linear. Since different sets of coefficients will not affect the linearity of the FIR filter but only influence the magnitude of output signal, any kind of coefficients that give a reasonable output magnitude can be used in this test. Therefore a simple set of coefficients with all coefficients set to zero except one coefficient is used in the test to have adequate output magnitude.

In this test, the input common mode voltage limits the maximum input range of the single ended input signal. On the other hand, large input common mode voltage will generate large output common mode voltage on the $R-2R$ network such that the switching cell can not work properly. Thus an appropriate choice should allow large input range while maintain the switching cell working. For this reason, a common mode voltage of 1.1V is used in all the following tests.

The final output gain of the FIR filter is:

$$A_{output} = A_{S/H} \cdot A_{DAC} \tag{4.1}$$

where $A_{S/H}$ is the gain of the Sample-and-Hold which should be very close to one, A_{DAC} is the combination of transconductance and the gain of the multiplier. The gain of multiplier is the combined result of the loaded coefficients and the resistance used in the

$R-2R$ network. Figure 4.3 shows the linearity test result. It is clear that when the input signal has amplitude less than the common mode voltage, which is 1.1V, the filter can maintain a good linear relationship between the input and the output. By using least-square linear fit method, the best fit line is shown together with the test results in Figure 4.3. The largest measured error is less than 6 percent of the best fit value, which is a very small number showing the good linearity.

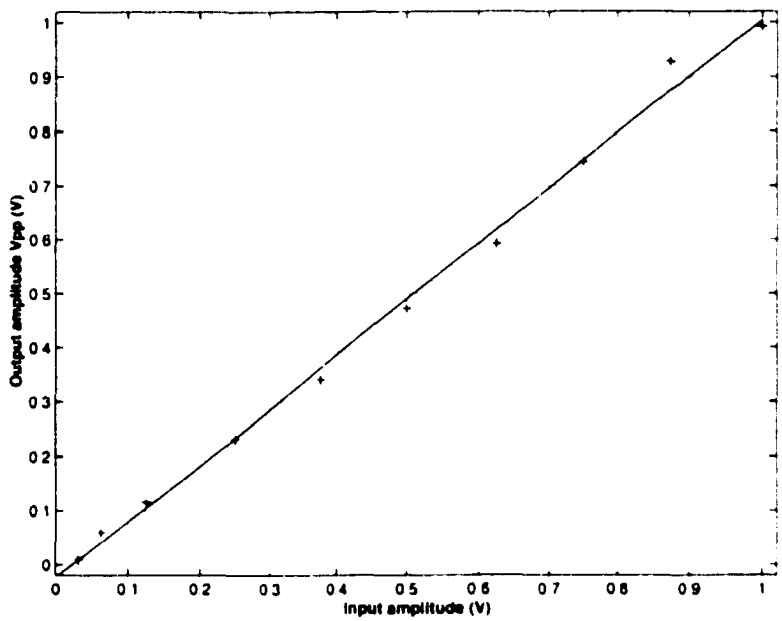

Figure 4.3 Output Amplitude Voltage vs. Input Amplitude Voltage

4.2.2 Single tone result

Single-tone test result is obtained from an analog real-time oscilloscope, which should provide very high sampling rate and very wide input bandwidth that guarantees the accuracy of the captured signal. Since capturing the data for the spectrum analyzer requires certain setup that is not available, high speed oscilloscope with build-in FFT function is used to observe the spectrum instead. The result is similar to that of using spectrum analyzer.

As stated before, a linear system will not generate frequency components other than the input sine wave. However, a nonlinear system will have tones on the multiples of input signal frequency which are called harmonics. Figure 4.4 shows the spectrum of the output signal when the input signal is a single frequency sine wave.

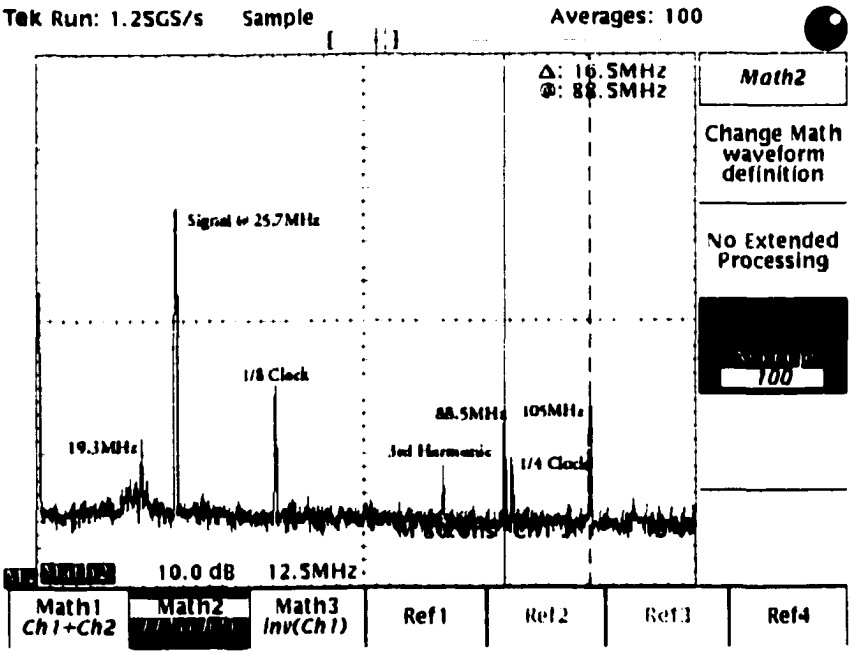

Figure 4.4 Output Spectrum of 25.7 MHz Input at 360 MHz Clock Frequency

The input signal has a frequency of 25.7 MHz and an amplitude of 625 mV. The system clock frequency is set to 360 MHz. Five categories of frequency components can be identified on the output spectrum: (1) input signal, which is the highest tone on the spectrum. (2) harmonics of the input signal. Since fully differential technique is used, even order harmonics are not noticeable on the spectrum. Among odd order harmonics, the 3rd harmonic is the dominant one with magnitude about -38 dB below the input signal. The 5th and higher odd order harmonics are not visible on the graph. (3) sub multiples of the clock frequency: this may be caused by the mismatch of offsets existing in the S/Hs and transconductors. (4) other interference: the tones at 88.5 MHz and 105.1 MHz come from the nearby FM radio stations. (5) inter-modulation between all

the above components. In Figure 4.4, the 1/8 clock frequency and the signal frequency generate the component at about 19.3 MHz.

4.2.3 Dual tone result

In a nonlinear system, dual-tone test can be used to examine the intermodulation distortions. Two sinusoidal signals at 25.7 MHz and 28 MHz are injected. The input signal level is 625 mV. The clock frequency is 360 MHz. The inter-modulation products are located at 23.4 MHz, 25.7 MHz, 28 MHz and 30.3 MHz around the input two signals. Other inter-modulation components are usually too small to be observed.

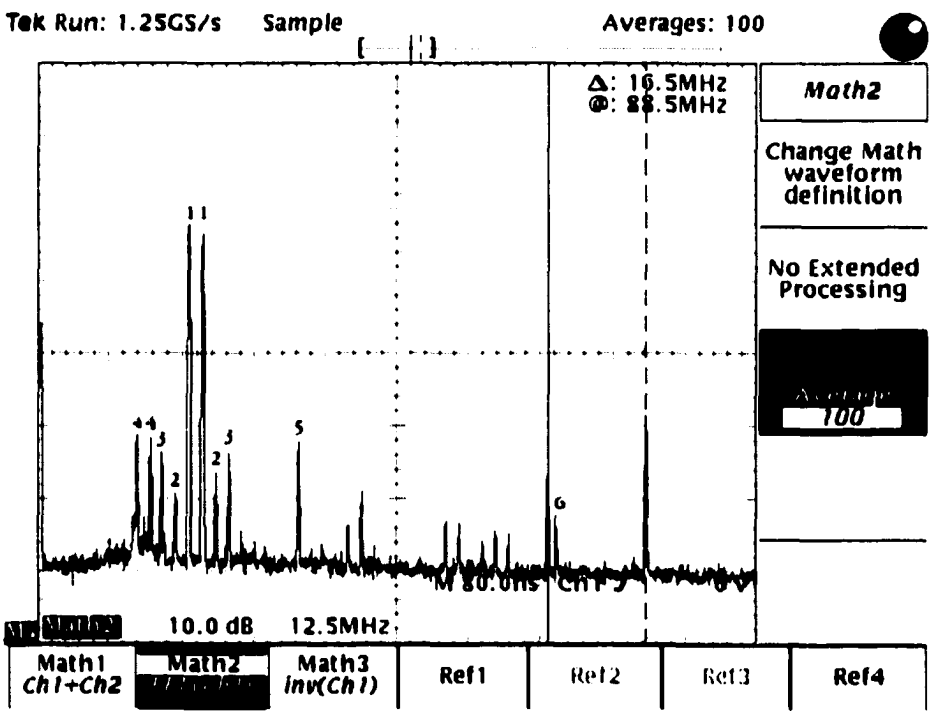

Figure 4.5 Output Spectrum of Dual Tone Input

Figure 4.5 shows the output spectrum of the two tone test. The above components can be clearly observed. However, there are several tones other than the four discussed above. On this spectrum, the 1's represent signal tones at 25.7 MHz and 28 MHz; the 2's represent the first pair of inter-modulation products at 23.4 MHz and 30.3 MHz; the

3's represent the second pair of inter-modulation products at 21.1 MHz and 32.6 MHz; the 4's indicate the inter-modulations between 1/8 clock and the input signals at 19.3 MHz and 17 MHz; the tone labeled 5 is the 1/8 sub-harmonic of the clock signal (45 MHz); and the tone labeled 6 is the 1/4 sub-harmonic of the clock signal (90 MHz). The dynamic range ($SFDR$) achieved here is above 30 dB with 625 mV_{pp} input signal magnitude.

However, the inter-modulation products will increase in a cubic fashion as the input signal magnitude increases [28]. Thus the test result can be used to determine the input signal size when given an inter-modulation distortion requirement.

4.2.4 FIR filter programmability test

In order to test the transfer function and the programmability of the FIR filter, different sets of coefficients are loaded into the filter to establish different low-pass responses. The clock frequency is set to be 250 MHz even though higher frequency can be achieved. S_{21} is measured by a network analyzer. One of the major obstacles of testing under higher frequency lies in that the loading logic becomes difficult to be synchronized with faster clock. The test results are under the conditions listed in Table 4.1.

Table 4.1 LPF Testing Condition

Clock Frequency	250	MHz
Curve 1 Cut-off Frequency	0.2	Nyquist Rate
Curve 2 Cut-off Frequency	0.3	Nyquist Rate
Curve 3 Cut-off Frequency	0.4	Nyquist Rate
Curve 4 Cut-off Frequency	0.5	Nyquist Rate

In Figure 4.6, the gray curves are the ideal low pass responses with different cut-off frequencies as shown in Table 4.1. The measured responses are shown in black curves. Comparison between the ideal and the tested response shows the consistency between these two. This demonstrates the programmable functionality of the FIR filter. It also shows that this FIR filter can be used to obtain different kinds of transfer functions.

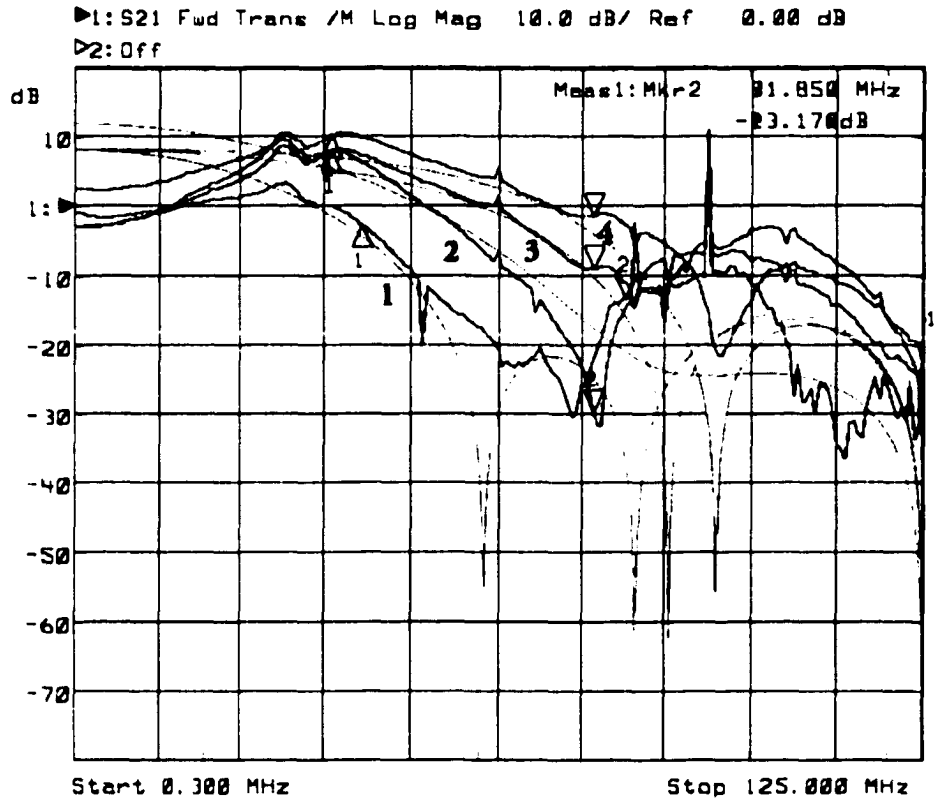

Figure 4.6 Transfer Function of LPF with Different Cut-off Frequency

The difference between the ideal response and the measured response at low frequency range is due to the effect of the input transformer. Since the transformer is inserted before the signal input pins to generate differential signal from a single ended signal source, large attenuation will appear at the frequency range that is lower than the -3 dB bandwidth of the transformer. Even with this attenuation, the correspondence between ideal and tested curve are maintained.

4.2.5 Summary of test results

Many tests had been performed to examine the functionality of the FIR filter. Results collected in Table 4.2 and Table 4.3 show the mixed-signal multiplier operates correctly for different clock frequencies or input frequencies.

Despite the non-ideal testing environment, test results obtained are acceptable. It

Table 4.2 Single Tone Test Summary

Input	Magnitude	Clock	HD_3
25.7 MHz	0.375 V	360 MHz	-42 dBc
25.7 MHz	0.625 V	360 MHz	-38 dBc
25.7 MHz	0.875 V	360 MHz	-42 dBc
25.7 MHz	0.375 V	300 MHz	-40 dBc
25.7 MHz	0.625 V	300 MHz	-38 dBc
25.7 MHz	0.875 V	300 MHz	-39 dBc
24.8 MHz	0.375 V	250 MHz	-50 dBc
24.8 MHz	0.625 V	250 MHz	-45 dBc
24.8 MHz	0.875 V	250 MHz	-40 dBc

Table 4.3 Dual Tone Test Summary

Input	Magnitude	Clock	Inter-Modulations
25.7/28 MHz	0.375 V	360 MHz	-30 dBc
25.7/28 MHz	0.625 V	360 MHz	-28 dBc
25.7/28 MHz	0.875 V	360 MHz	-22 dBc
25.7/28 MHz	0.375 V	300 MHz	-29 dBc
25.7/28 MHz	0.625 V	300 MHz	-26 dBc
25.7/28 MHz	0.875 V	300 MHz	-21 dBc

demonstrates two important facts: first, this design has acceptable dynamic performance. Second, the FIR filter is programmable. From the test results above, the largest distortion comes from the sub-multiples of the clock frequency. In many cases, they are larger than the inter-modulation distortion and the harmonic distortion of input signals. Although tones due to sub-multiples of the clock signal will degrade the performance of the FIR filter, the overall system SNR is sufficient for disk drive read channel application. The transfer function tests demonstrate the programmability of the FIR filter. Performance similarity between the ideal and tested transfer functions confirms that FIR filter using circular buffer structure is fully functional.

Table 4.4 is a brief summary of the performance of the programmable FIR filter.

Table 4.4 Chip Specifications

Active circuit Area	1.1 by 1.1	mm
Power Supply	3.3	Volt
Clock Frequency	up to 360	MHz
Total Power Dissipation	462	mW
Buffer Power Dissipation	307	mW
Analog Power Dissipation	79.2	mW
Digital Power Dissipation	75.9	mW
Input Signal Voltage Range	$0 \sim 1.0$	Volt
Dynamic Range (0.5V input amplitude)	> 35	dB

4.3 Adaptive Equalizer Test Overview

After testing of the programmable FIR filter, the next step is to test the adaptive equalizer. However, the ROM chip that provides the simulated read channel data does not operate appropriately, so that the adaptation test can't be performed.

Nonetheless, the functionality of adaptation can be examined from the following observations: when the adaptation function is turned on, the monitored coefficients change and then stop to non-zero and non-saturated values. This observation shows that the detection and the updating blocks are working correctly. When the filter core is tested with the adaptation function turned off, it behaves exactly the same as the FIR filter, which is expected. Because of the additional updating block, the maximum operating frequency was less than that of the programmable FIR filter. However, an operating speed up to 320 MHz was observed.

4.4 Performance Comparison With Prior Art

In order to show the merit of this design, comparison was made between this programmable FIR filter and the prior art with similar structure and technology. All of them are FIR filters with analog or mixed-signal designs for the disk drive read channel applications.

The result of this comparison is shown in Table 4.5. It is clear that a great im-

provement is achieved in this design due to the new mixed-signal multiplier. Further comparison can be made by defining Figure-of-Merit as:

$$FOM = \frac{Power}{Speed} \frac{3.3}{V_{supply}} \tag{4.2}$$

Equation (4.2) considers the differences caused by power supply and makes proper scaling adjustment. In this comparison, the effect of fabrication process is not included. If it is considered, the performance of this design will be even better than the prior art. Figure 4.7 demonstrates the calculated FOM using Equation 4.2).

Table 4.5 Comparison of different designs

Design	Technology	Speed	Analog Power	Performance
1 This work	3.3V 0.6μm CMOS	360 MHz	79 mW	38 dBc HD_3
2 [32] 1998	3.3V 1.2μm CMOS	170 MHz	66 mW	40 dBc HD_3
3 [2] 1996	5V 2μm CMOS	55 MHz	40 mW	N/A
4 [31] 1997	5V 0.8μm BiCMOS	160 MHz	200 mW	N/A
5 [30] 1996	5V 0.6μm CMOS	200MHz	410 mW	N/A
6 [35] 1995	5V 2μm CMOS	20 MHz	45 mW	47.4 dBc HD_3
7 [40] 1997	5V 0.8μm CMOS	20 MHz	70 mW	46 dB THD
8 [41] 1995	3.3V 0.5μm CMOS	250 MHz	780 mW	N/A

Several conclusions can be drawn from the above comparison: (1) the design in this dissertation has the highest speed. This is solely due to the proposed circuit design techniques without the use of advanced processes. (2) the design in this dissertation is the most power efficient design for the achievable clock rate. (3) mixed-signal or analog FIR has much lower power dissipation. Among the above eight design examples, design number 8 uses pure digital structure that consumes the largest power.

4.5 Summary

Two high speed mixed-signal circuits for disk drive read channel (a programmable FIR filter and an adaptive equalizer), were designed based on a 0.5μm CMOS technology. The design goal is to demonstrate that mixed-signal implementation of equalizer is

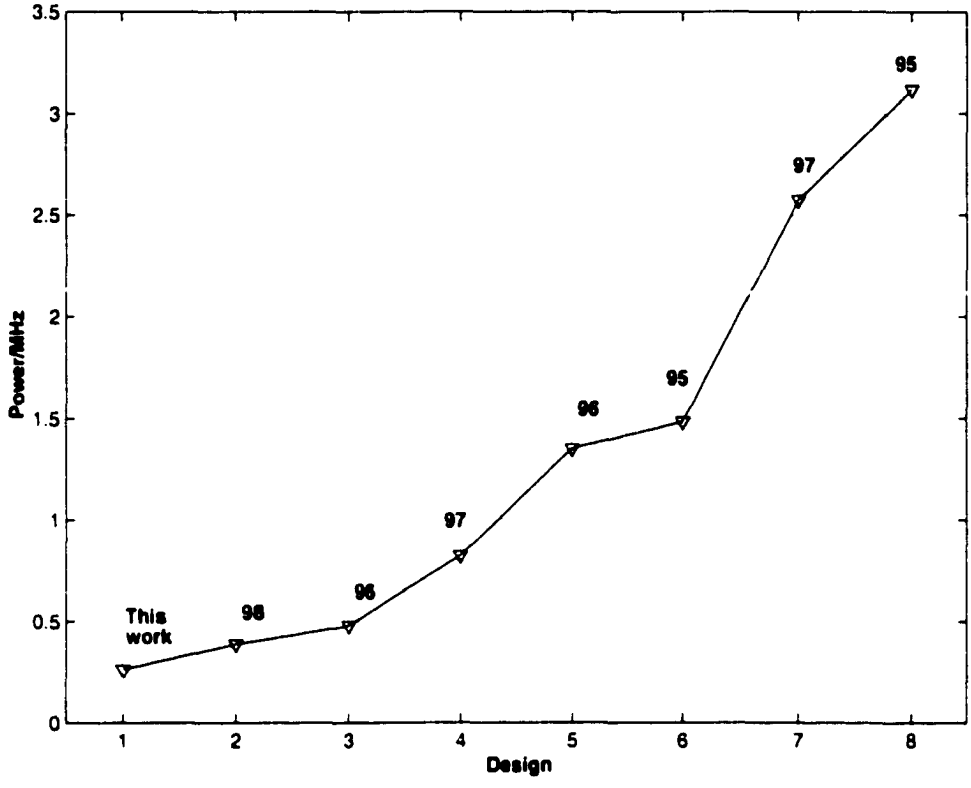

Figure 4.7 Comparison of Power-per-MHz

capable of achieving very high speed at low power consumption. The core of the design is a new mixed-signal multiplier.

Test results show that the programmable FIR filter is fully functional. The highest operating speed is beyond the design goal. The success of this FIR filter sets a good example to show the potential of achieving good performance through good design techniques.

Due to the lack of appropriate signal sources, the adaptive equalizer did not undergo full functional test. Since the adaptive equalizer and the FIR filter share the same computation blocks, the adaptive equalizer was tested as a programmable FIR filter. The result is similar to that of the FIR filter chip. The highest operating frequency is slightly lower than that of the FIR filter since more circuitry is added within the data path.

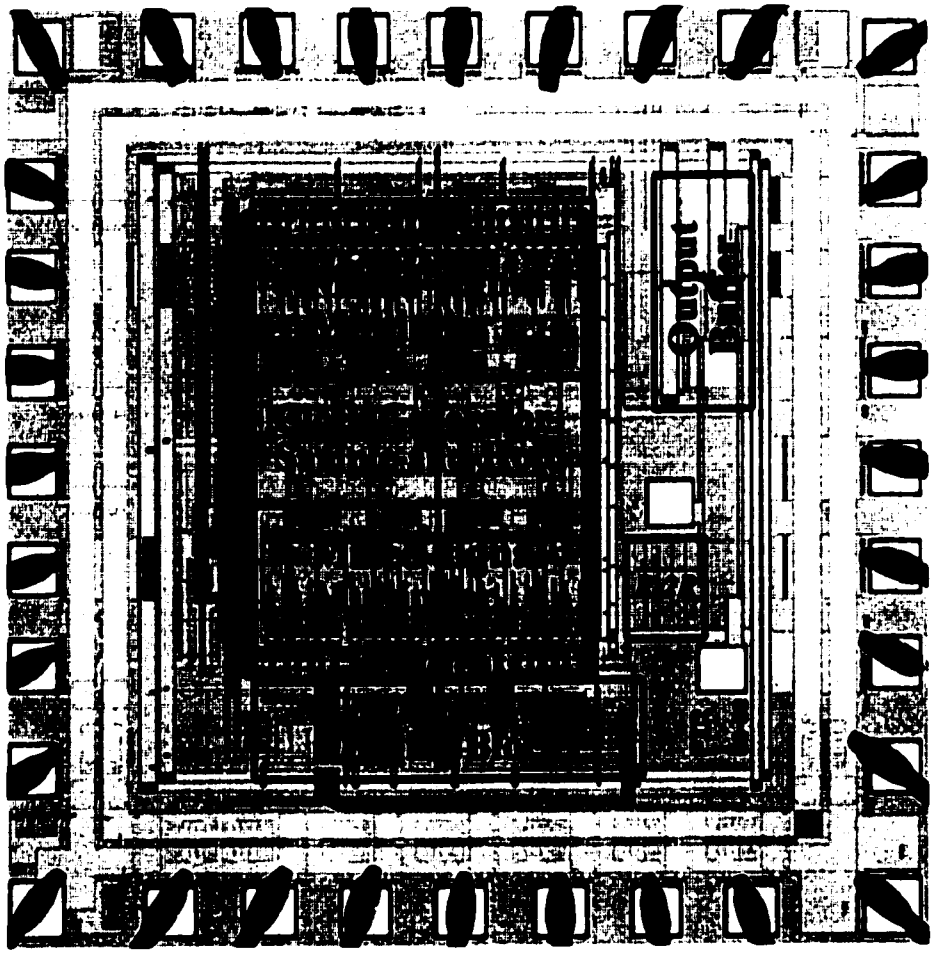

Figure 4.8 Die micrograph of FIR Filter

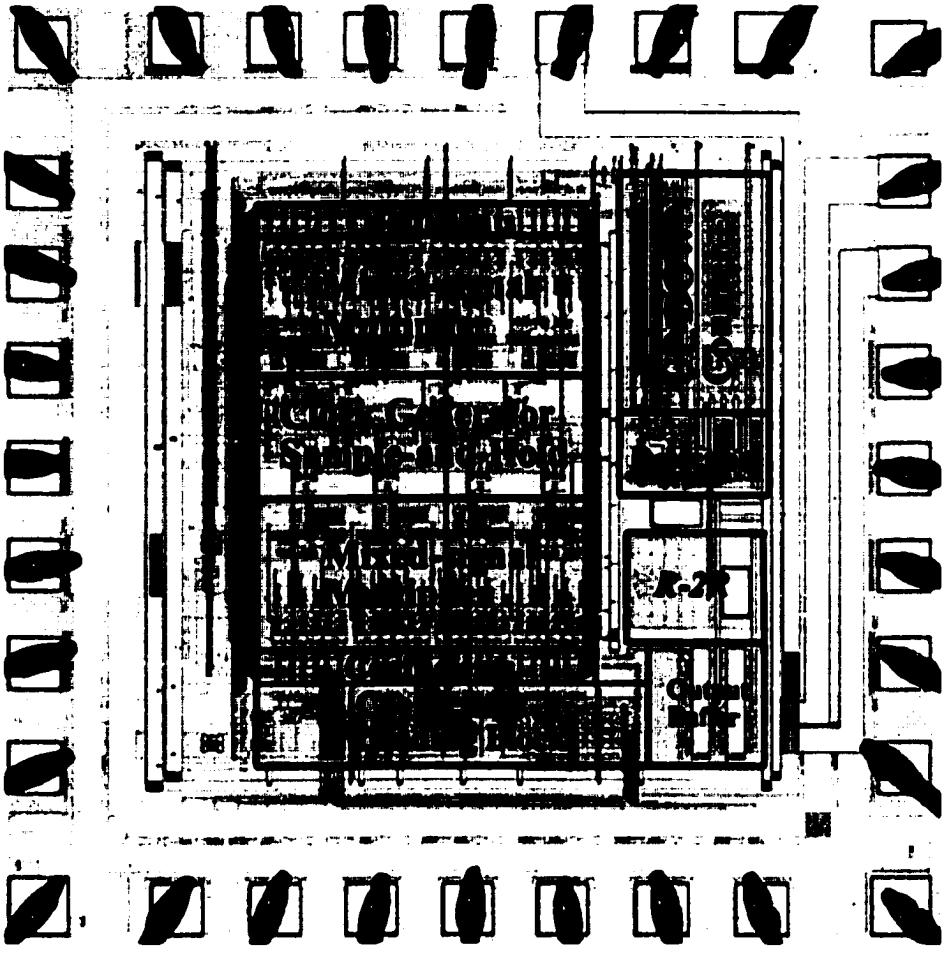

Figure 4.9 Die micrograph of Adaptive Equalizer

5 ANALOG-TO-DIGITAL CONVERTER

5.1 Introduction

The second part of this dissertation is about the calibration of the non-ideal effects existing in analog-to-digital converter (ADC). With the rapidly growing power of Digital Signal Processing (DSP), the demand of high speed and high resolution ADC keeps increasing. In order to further enhance the performance of ADC, calibration of non-ideal effects due to IC fabrication has become particularly important.

In this chapter, basic ADC aspects are reviewed first, including brief discussion of performance specifications as well as basic structures. Different ways to categorize ADC will be presented. A new ADC classification method based on *data path* will then be introduced. From the above classification, the time-interleaved ADC is then discussed. Since this architecture is the focus of this research, its advantage as well as its inherent problems will be emphasized.

5.2 Analog-to-Digital Converter Fundamentals

The basic function of ADC is to translate data from the analog world into digital world. Generally, an ADC performs three operations: 1, **Sampling**- ADC samples the continuous analog input signal to a discrete time signal. The corresponding digital code will be valid only for that sampling instant. 2, **Quantizing**- ADC divides a full signal range into many levels or discrete signal levels and maps the sampled signal onto one of the levels. This mapping should have a linear relationship. 3, **Encoding**- ADC then assigns a digital code to the mapped level. The digital code can be of any kind but

the most commonly used one in the digital computer is the binary code. Usually, the whole procedure from sampling to the final output of the digital code is referred to as one conversion. Several basic attributes of a ADC are:

Resolution n:	number of bits that the ADC provides.
Speed $MSPS$:	number of samples that the ADC can process in one second.
Latency:	number of sampling intervals needed to finish a conversion.
Signal Range:	the full input range that the ADC can handle.
Power:	power needed to finish continuous conversion under certain speed.

In the above attributes, speed and resolution are the most important ones. Different applications have different requirements on these two attributes.

5.3 ADC Performance Specification

There are two major classes of performance specifications: (1) *Static Performance*: uses DC signal to measure the linearity of an ADC. (2) *Dynamic Performance*: measures the performance under certain speed using spectrum analysis method and sine wave as input.

5.3.1 Static performance

Without considering the speed, one basic equation is used to express the ADC function by defining its transfer function as:

$$V_{output} = aV_{input} + b \tag{5.1}$$

where V_{output} is the reconstructed analog value from digital output of the ADC. Equation (5.1) provides the definition of the gain of a converter as a, and the offset as b. An ideal ADC should have $a = 1$ and $b = 0$. In reality, after best-fitting the ADC transfer function, there is always some deviation of a and b from their ideal values.

Besides gain and offset variations that reflect the overall static performance, the detailed linear property on each code should be characterized. When drawing the input voltage vs. the output digital code, a transfer function of the ADC is obtained. Ideal input/output transfer function will be a straight line with 45 degree slope if the input and output have same scale. Since the converter performs the quantization on the signal, the magnitude level is no longer continuous but discrete. The real conversion curve will then appear as steps with each step representing a discrete level. Shown in Figure 5.1 are these two curves when $n = 4$ with the solid line representing the actual curve with quantization and the dashed line representing the ideal curve. When the actual curve has an offset error and a gain other than 1, the best fit line can be used as the ideal curve.

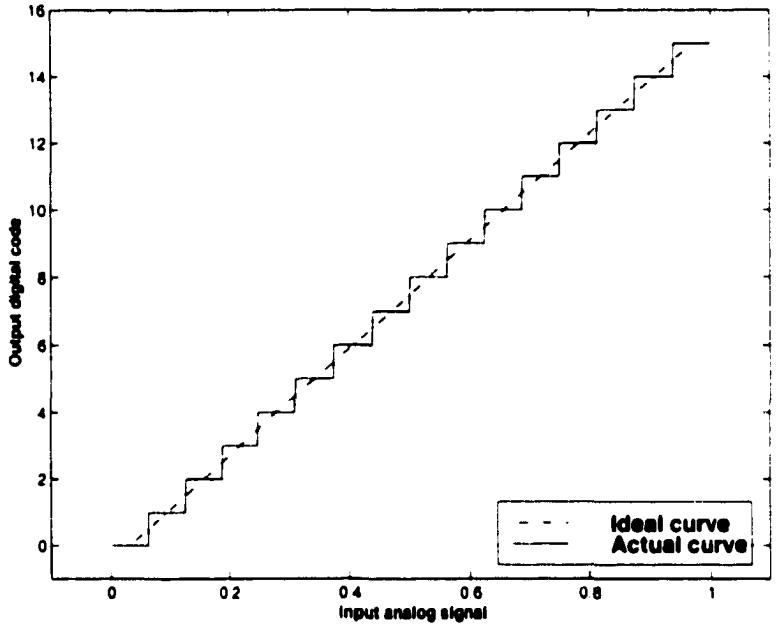

Figure 5.1 Ideal and Real Transfer Function Curves of ADC

The difference between the ideal curve and the actual curve at certain input value is the *quantization error*. The value of Least-Significant-Bit (LSB) is defined as the smallest ideal step in the quantization curve. Or it can be written as:

$$1LSB = \frac{V_{fullscale}}{2^n} \tag{5.2}$$

Anything less than $\pm\frac{1}{2}LSB$ will not be have any difference at the converter output. Thus the value of $\pm\frac{1}{2}LSB$ is generally used to measure whether a distortion is large enough to generate error on the final digital output.

In the above figure, the actual curve is still the one without any other error sources except quantization error. In the real case, this curve will have non-uniformly distributed steps. The Differential-Non-Linearity (DNL) can be defined as the analog difference between two adjacent quantization levels less 1 LSB. The Integrated-Non-Linearity (INL) is defined as the sum of the DNL. The ideal value for DNL and INL are zero. In reality, DNL should be less than $\pm 0.5LSB$ to maintain a n bit resolution. INL should be less than $\pm 1LSB$. The INL and DNL will give very good insights of the static linearity performance of each code of a converter [37].

5.3.2 Dynamic performance

Another important standard to evaluate the performance of an ADC is the dynamic performance testing, which reflects the ADC performance under certain speed. A sine wave with its amplitude slightly smaller than the full input range will be used as an input. The output digital code will be combined to form a series of discrete digital code. A Fast-Fourier-Transform is then performed to obtain the spectrum where several important parameters are defined. Take Figure 5.2 as an example.

(1)**SFDR**: Spurious-Free-Dynamic-Range. In many cases, it is simply referred to as dynamic range. It is defined as the distance between the signal level and the highest tone other than the signal and the DC component. This indicates the largest input signal range that the converter can convert without any distortion.

(2)**SNR**: Signal-to-Noise-Ratio. It can be defined in many ways. A broad definition is :

$$SNR = \frac{Power_{signal}}{\sum Power_{other}} \tag{5.3}$$

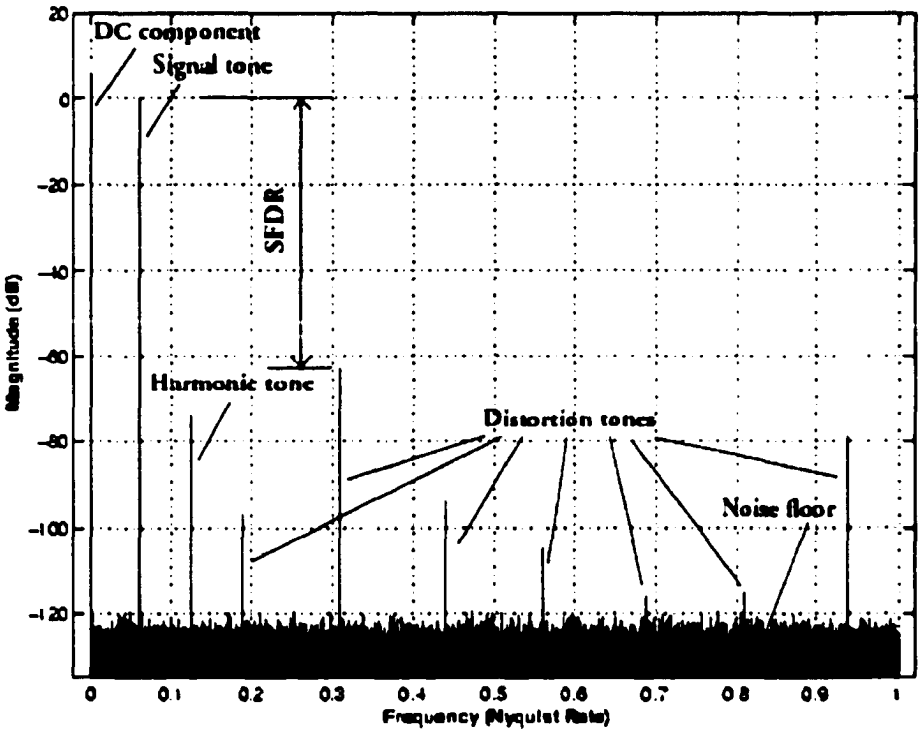

Figure 5.2 Dynamic Performance Definition of ADC

A confined definition is :

$$SNR = \frac{Power_{signal}}{\sum Power_{other} - \sum Power_{harmonic}} \tag{5.4}$$

In this case, another parameter is to define Total-Harmonic-Distortion as :

$$THD = \frac{Power_{signal}}{\sum Power_{harmonic}} \tag{5.5}$$

Both definitions of SNR are used in the literature. However, only the first one is used in this thesis to take into consideration of the influence from harmonics. Sometimes, the first definition is also referred to as $SNDR$ (Signal-to-Noise-plus-Distortion-Ratio) or $SINAD$ (SIgnal-to-Noise-And-Distortion-ratio). In this thesis, SNR and $SNDR$ are consider as the same concept.

(3)$ENOB$: Based on SNR value, the Effective-Number-of-Bit($ENOB$) is defined as:

$$ENOB = (SNR - 1.76)/6.02 \tag{5.6}$$

for a sine wave input with its peak-to-peak amplitude the same as the maximum input range of the ADC. $ENOB$ points out the actual achievable output bits from an ADC under certain conversion speed and input signal frequency. From this point of view, it is a very important parameter and is widely used in literature.

(4)*Noise floor*: The noise floor is due to the quantization errors and all other noise sources. Under ideal cases, the noise floor is only due to the quantization error which is calculated in [37]. However, when there are other noise sources, such as those coupled from the power supply, the clock jitter or device noise, the noise floor will be raised. For different applications and different designs, the amount from other noise sources is quite different. Therefore, it is impossible to have an universal equation to describe where the noise floor would be. In generally, if an error is random in nature, it results in the rising of noise floor.

(5)*Distortion tone*: It is caused by deterministic errors. The error source may be from an asymmetric circuit layout, the clock generator and devices along the data path. In the frequency spectrum, spurious tones are generated because these errors tend to repeat periodically. These tones are called distortion tones. Observing the spectrum in Figure 5.2, there are many distortion tones. The direct influence of these tones are reduced in dynamic range or $SFDR$. In many cases, the tones are very high compared to the noise floor, they become the dominant factors when computing SNR and $ENOB$.

5.4 Basic Structure of ADC

ADCs are generally divided into two main types: (1) *Nyquist* converter and (2) Oversampling $\Sigma - \Delta$ converter. In *Nyquist* converter, the input and the output has one-to-one linear relationship. Theoretically the input signal frequency can be up to the *Nyquist* rate (though in reality it is difficult) with close to ideal $ENOB$. On the contrary, $\Sigma - \Delta$ converter is based on oversampling and noise shaping techniques. The output is modulated and there is no one-to-one linear relationship between the input

sample and the output code. This type of *ADC* is mainly used in low speed and very high resolution application. The oversampling $\Sigma - \Delta$ converter will not be discussed in this dissertation. The structures that will be discussed are those used in *Nyquist ADC*'s.

(1) Flash structure: It contains 2^n comparators which can directly determine the quantization level and generate the digital code by decoding. It provides by far the fastest conversion speed as well as the shortest latency with moderate resolution (approximate 6 bits).

(2) Pipelined structure: It contains several stages, each stage is a flash type sub-converter which generates one or more bits and later on combined to output n bits. Each stage can have same or different bits of resolution as long as the total number of bit is n. The total comparators used in pipelined structure is much less than 2^n. Therefore pipelined converter is more cost efficient and is the most commonly used structure for high speed and high resolution applications. The pipelined structure can also achieve the same throughput as the Flash converter in that it can provide one new conversion output for each clock cycle except that the period of the clock cycle is usually longer than that of a flash converter. The drawback of the pipelined structure is that its latency is much longer than that of the flash structure.

(3)Two-step structure: It utilizes two stages with each stage as a flash converter to do the conversion. It dramatically reduces the number of comparators but still has a rather low latency. It can be regarded as a special type of pipelined converter. However, since many circuit design techniques can be used to further boost the performance of this structure, it is usually listed separately.

(4)Another alternative of pipelined structure is by using one stage over and over again. This is called algorithmic converter. The hardware cost is much lower than both pipelined and flash structures at the cost of longer conversion time. It has the same latency of a pipelined structure but it can not provide one conversion for each clock cycle. Therefore the throughput is very low. It is a good choice for low speed

application.

5.5 Classification of *ADC*

The detailed implementation of a *ADC* can have many different flavors. Therefore it is hard to compare two *ADC*'s impartially. First of all, the technology used in designing *ADC* can be any of the many kinds that are available today, from low cost CMOS, BiCMOS to Bipolar or rather expensive GaAs. Secondly, with the same technology, the architecture can be quite different. Thirdly, even with the same technology and the same architecture, there are still a lot of other differences to be considered. Therefore, it is hard to categorize *ADC* according to only one standard.

Besides the two main types of *ADC* discussed before, they can also be categorized according to the different structures above. Or they can be categorized according to their application, their speed and resolution. In this work, the *ADC*'s are categorized by considering whether channel parallelism is applied in the top architecture. According to this point of view, all *ADC*'s belongs to either *single data path* converter or *multiple data path* converter. A single data path converter will require the minimum hardware while multiple data path one will increase the hardware cost by a factor of the number of paths that are used. The concept of single/multiple data path referred in this dissertation doesn't apply to certain kind of noise shaping structures for the reason that the noise shaping needs to have feedback and thus establishes multiple data loops.

5.5.1 Single data path converter

As stated above, the single data path structure applies to those converters where no signal processing is done in parallel style. It can have any kind of the structures discussed above. There are many good reasons to use single path data converter. It is simple, small and thus relatively cheap. Almost all the reported *ADC*'s are single data path converters. However, the performance, especially the conversion speed, is always limited

by the technology that is used. Usually the upper limit of speed is set by the speed of the amplifier that can be reached in that technology. When considering to achieve higher throughput without using any advanced technology, converter having more than one data path is becoming attractive.

5.5.2 Multiple data path converter

A multiple data path converter is defined as converter that has similar or identical converters working together. Each single converter is called a conversion channel. The detailed structure of each channel is determined by the application requirements. Sometimes this kind of converter is also called 'parallel' structure since different converters (channels) are working in parallel. In this thesis, a formal name of **Time-Interleaved ADC** will be used to represent this type of converter.

5.5.2.1 Time-Interleaved *ADC*

The architecture of a time-interleaved *ADC* with M channels operating at an overall sampling rate of F_s is shown in Figure 5.3. Each channel is a complete n-bit *ADC*. The *S/H* block at the front of each channel samples the analog input at alternate time intervals and the sampled value is converted into digital code at a speed of $\frac{F_s}{M}$ within each channel. Digital outputs from all channels are recombined by the digital multiplexer that provides the full data rate of F_s. The biggest advantage of this structure is the speed relaxation by a factor of M on each channel so that the complexity of analog circuit design of each channel is loosened. With the same technology, if every channel can achieve a speed of f, the combined speed can be Mf. Therefore, it is an attractive way of enhancing the speed of *ADC*. The idea is introduced in [1] in 1980. Ever since its first introduction, its application has been limited to ultra high speed sampling system with advanced process due to its hardware cost and its unique non-ideal error effects as discussed later [42, 43, 44, 45].

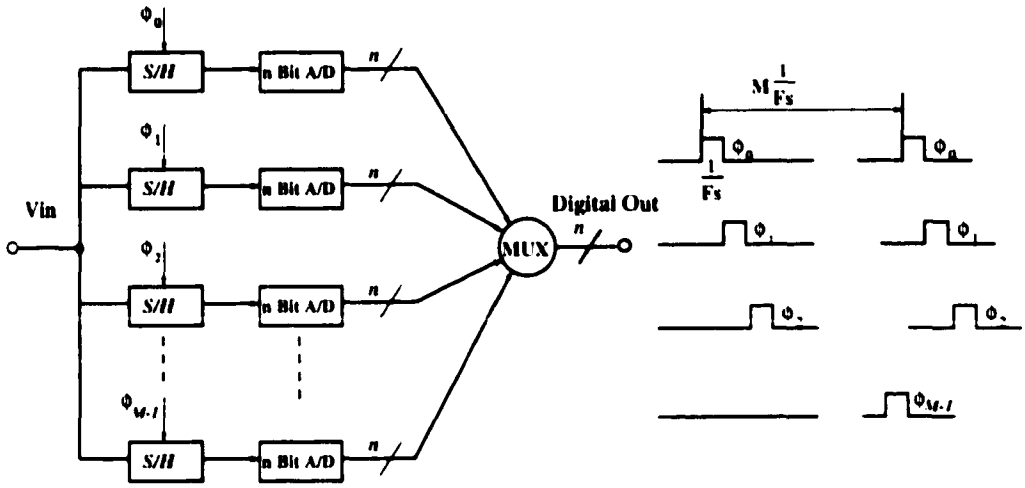

Figure 5.3 Time-Interleaved ADC

5.6 Performance Limiting Factors

There are many factors affecting the final ADC's performance. It can be categorized as the influence from device parameters, such as device sizes, values, shapes, etc., the influence from clock generator and noise coupling from other circuitry through power supply or substrate. Since the effect of clock generator cannot be easily measured statically, its effect is usually analyzed through ADC's dynamic performance (or in frequency spectrum).

5.6.1 Device parameters

Device parameters may include various things. It may be the absolute physical parameters such as capacitor size, switch size or transistor size, or the relative parameters such as matching.

The design of device parameters always involves lot of trade-offs. For example, the choice of capacitor size must take into account of the resolution and the speed requirement simultaneously. In order to achieve certain resolution, the $\frac{kT}{C}$ noise must be much smaller than 1 LSB. Therefore the bigger the capacitance the better. At the same time, it must be small enough to meet the speed requirement. The final choice is a trade-off

between the two.

Since almost all modern circuit designs utilize symmetric differential circuit design technique, the absolute values are not always as important as the matching characteristic. Usually it is the matching characteristic that limits the final performance of ADC. Due to this fact, how to correct the mismatch has become a very valuable topic. The correction can be done either dynamically on-chip or one time during factory testing. All these are referred to as ADC calibration [46, 47, 48, 49, 50].

In the case of multiple data path ADC, the mismatches of gains and offsets between channels will generate additional distortion apart from the device mismatches itself. One method to reduce this channel mismatch effect is introduced in Chapter 7.

5.6.2 Timing errors

Timing error comes from non-ideal clocks used in an ADC. The accuracy of time intervals is easily deteriorated by all kinds of noise and disruptions. Since the converter first needs to sample the analog signal, the exact moment of sampling is very important. If the sampling is not done at equal time intervals, differences between ideal and actual sampled values will arise. If this is deterministic, distortion tones will appear, and if it is random in nature, the noise floor will be risen.

5.6.2.1 S/H timing error

In the single data path converter structure, the only timing error will be the S/H timing uncertainty error. Since generating clock without timing errors is not possible, it is important to determine how serious the induced error could be.

Assume the input to be a sine wave with the frequency of F_{in} and an amplitude of A. The largest error that the timing error can introduce is at the zero-crossing point where the sine wave has the largest slope of $A2\pi F_{in}$. Suppose the timing uncertainty is δt, the biggest error must be less than:

$$A2\pi F_{in}\delta t < 1LSB$$

$$\delta t < \frac{T_{in}}{2^n \pi} \tag{5.7}$$

Equation (5.7) gives the upper limit of the clock jitter when the input signal has the maximum allowable input frequency. For example, a 50 MHz sampling rate ADC at 10 bit will require less than 6 ps timing error.

5.6.2.2 Timing mismatch in multiple data path ADC

Besides the timing error mentioned above, there is additional timing consideration in multiple data path ADC, namely the timing mismatch between channels. Unlike in single data path converter, clock mismatches may be a very important error source that limits the ADC performance in multiple data path ADC [42][51]. In Chapter 6, a calibration method is proposed to deal with such timing error effects.

5.7 Summary

ADC is very important in modern signal processing. It suffers from many non-ideal effects such as mismatches, noise, etc. In order to improve the performance by satisfying the critical requirements, calibration is necessary.

As an attractive approach of achieving higher throughput, time-interleaved ADC has unique advantages over single data path converters. However, it has additional problem that needs to be taken care of : channel mismatch. This includes mismatches from the channel gain, offset, non linearity and timing. In the following two chapters, two new methods to calibrate the above mismatches will be discussed in detail.

6 CALIBRATION OF TIMING ERROR EFFECTS IN TIME-INTERLEAVED A/D CONVERTER

6.1 Introduction

In recent years, a general trend in wireless communication systems is to push the digital signal processing circuitry toward the front end as much as possible. This trend imposes stringent requirements on the design of analog-to-digital converters. ADC's for such application not only need to have high speed ($> 50MHz$) and high resolution ($\geq 12bits$), but also require large dynamic range ($SFDR \geq 80dB$) [52] [53]. All of these make the design of traditional pipelined ADC for this application more difficult.

As an alternative approach for such application, time-interleaved A/D converter relaxes the speed of each channel and makes the design easier. However, as mentioned in the previous chapter, care needs to be taken on additional issues inherent in this type of converter. Although different techniques have been proposed for reducing the effects of channel parameter mismatches [48][54][55], not much work has been proposed to reduce or calibrate the effects of timing errors. However, as will be shown shortly, timing errors in time-interleaved ADC can greatly degrade the performance, and in many cases it can become the dominant factor that limits the application of this architecture [56]. In this dissertation, a digital background calibration technique based on interpolation is proposed to reduce the channel timing error effects, especially the skew effects. Since most of the calibration process is carried out on the digital outputs, very little change is needed on the analog part of the ADC.

6.2 Timing Errors in Time-interleaved *ADC*

The architecture of a time-interleaved *ADC* with M channels operating at an overall sampling rate of F_s is shown again in Figure 6.1. The detailed structure of each channel does not affect the results of the proposed technique. The only requirement is that each individual channel should be a complete *ADC*.

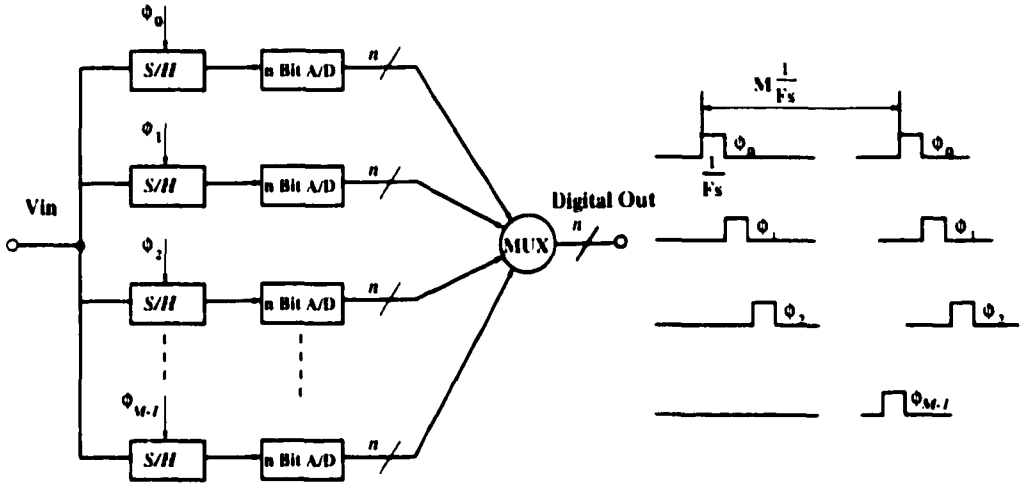

Figure 6.1 Time-Interleaved *ADC*

Different channels convert samples according to the sequential clock phases ϕ_0, ϕ_1, ... ϕ_{M-1} at the rate of F_s/M. Digital outputs from all channels are recombined by the digital multiplexer that provides a full data rate of F_s sample/sec. Since the sampling clocks for individual channels are generated by a multi-phase clock generator, timing errors on the sampling instant of each channel is unavoidable. There are two types of timing errors to be identified. (1) **Clock skew**: Δt_i, *deterministic* difference between ideal and real sampling edge. Clock skew is mainly due to device mismatch and asymmetric layout of the clock generator. It can also be caused by the disturbance from the power supply which may contain a deterministic component.(2) **Random jitter**: δt_i, *random* changes on the clock edges (also known as *phase noise*). Random jitter is mainly due to device noise and random noise coupled from the power supply and substrate. These two effects are illustrated in Figure 6.2. If channel 0 is used as a reference and channel

i has a clock skew of Δt_i, this will cause the sampled value x_i to have an error of Δx_i. The deterministic clock skew error will produce an unwanted phase modulation on the output data and create unwanted tones. As a result, the $SFDR$ will be degraded[1]. Due to the random jitter, additional error δx_i on the sampled data will result. Since this effect is random in nature, it tends to spread out the effect on the whole output spectrum by raising the noise floor and hence, results in decreased SNR.

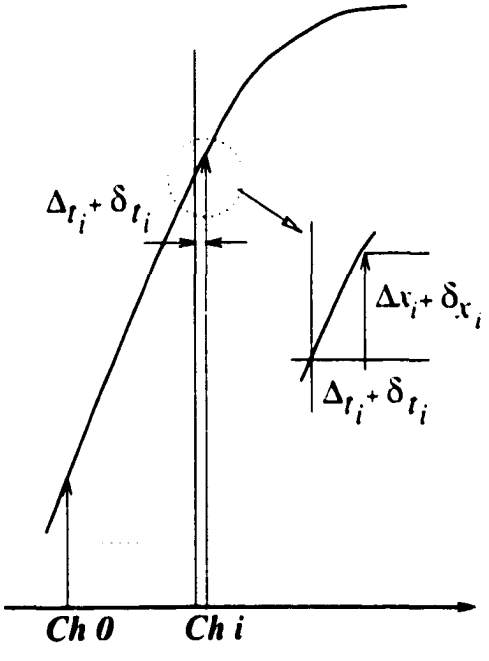

Figure 6.2 Illustration of the Effect of Clock Skew and Random Jitter

As an illustration, a simulated output spectrum of a time-interleaved ADC with only clock skew and random jitter effects is shown in Figure 6.3. The ADC has 8 ideal channels, and each channel has 14 bits of resolution. The random jitter is assumed to have a uniform distribution within a range of ±0.5% of the sampling period, ie. $\delta t_i \in [-0.005T_s, +0.005T_s]$. The clock skews are assumed to be within the same range with a different fixed value for each channel. The input is a pure sine wave with amplitude slightly less than the full range. Shown in Figure 6.3a is the ideal case without timing error. As shown in Figure 6.3b, the ADC that suffers from clock skews and random

jitter has unwanted tones and higher noise floor. Both $SFDR$ and SNR are reduced to be less than 70dB while the ideal 14-bit ADC has a $SFDR$ of 120dB and a SNR of 85dB.

For wireless communication applications, the $SFDR$ of the ADC is of major concern, hence all analysis and therefore the proposed technique will concentrate only on skew effects. A detailed analysis of clock skew effect is carried out in the Appendix. Similar analysis can also be found in [51][57]. For an input signal with a Fourier transform of $X(\omega)$, the ADC output spectrum $Y_s(\omega)$ can be written as:

$$Y_s(\omega) = \frac{1}{MT_s}\sum_{l=0}^{M-1}\sum_{k=-\infty}^{\infty} X(\omega - k\frac{\omega_s}{M})e^{-j(\omega - k\frac{\omega_s}{M})\Delta t_l}e^{-jkl2\pi/M} + noise \qquad (6.1)$$

where ω_s is the sampling frequency, and Δt_l is the clock skew of the lth channel. The term $noise$ in Equation (6.1) contains both quantization noise and the noise due to random jitter. When the input signal $x(t)$ is a complex sinusoidal signal, the output spectrum due to clock skew alone becomes

$$Y_s(\omega) = \frac{1}{MT_s}\sum_{l=0}^{M-1}\sum_{k=-\infty}^{\infty} \delta(\omega - \omega_{in} - k\frac{\omega_s}{M})e^{-j(\omega - k\frac{\omega_s}{M})\Delta t_l}e^{-jkl2\pi/M} \qquad (6.2)$$

For $k = 0$, it represents the desired signal, which has a magnitude of $\frac{1}{MT_s}\sum_{l=0}^{M-1} e^{-j\omega_{in}\Delta t_l}$. The unwanted tones appear near multiples of ω_s/M with $k \neq 0$. The $SFDR$ can then be calculated as the difference between the magnitude of the desired signal and the magnitude of the highest unwanted tone. Notice that the number of channels determines the number of unwanted tones. The magnitudes of the clock skews determine the magnitudes of the unwanted tones.

Traditionally, there are two approaches that help to minimize the timing error effects. The first is to use a front-end Sample-and-Hold (S/H) [54]. Since this S/H uses only one single clock to sample the input signal, it does not have the clock skew problem. Notice that random jitter still exists. The main challenge of this method is the need to design a high-accuracy S/H operating at the full sampling rate of F_s that can drive all the following S/Hs and settle to the desired accuracy within half clock cycle. Another

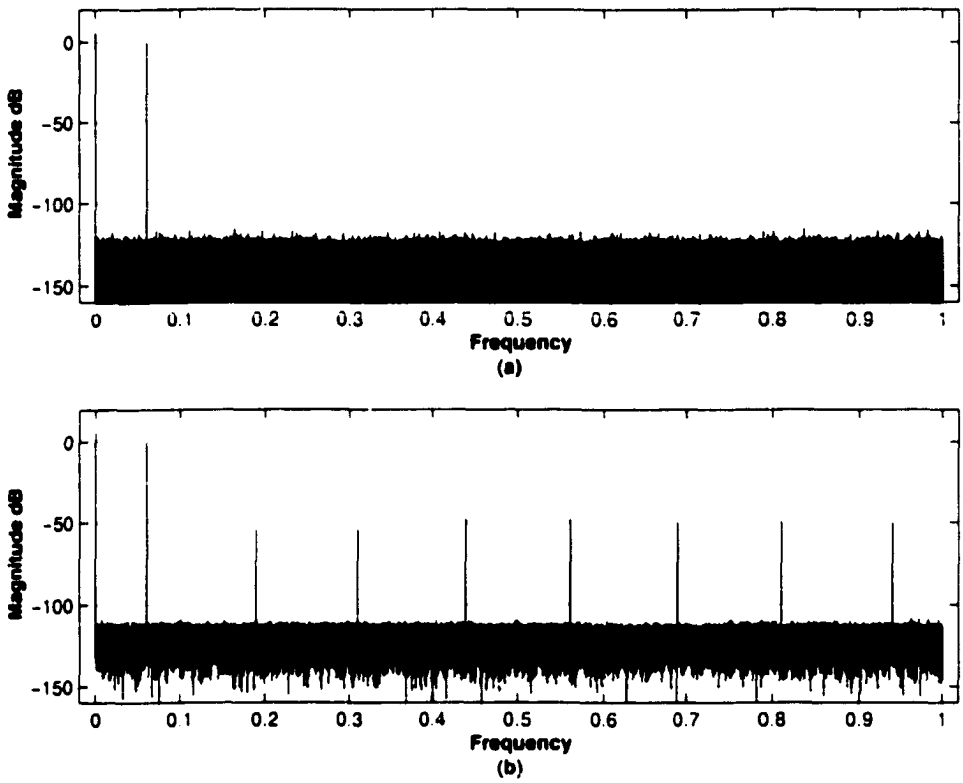

Figure 6.3 Simulated Output Spectrum (a) without clock skew and random jitter (b) with clock skew and random jitter

approach is to design a very low clock skew and low jitter multi-phase clock generator (for example, using delay locked loop [55]). Nevertheless, it remains a challenge to achieve low clock skews between different clock phases.

6.3 Post Conversion Background Calibration Through Interpolation

If each channel in the *ADC* is ideal, the digital output of each channel will represent the correct magnitude at the actual sampling instant. The unwanted tones are generated due to the fact that no timing information is given to these digital outputs and these outputs are then interpreted as the outputs at the ideal sampling instants. However, it is possible to reconstruct the original signal based on the digital outputs at the actual sampling instants if the timing information is known. Jenq [58] proposed a method to reconstruct the original signal by processing the digital outputs in frequency domain.

This approach is suitable for instrumentation applications. However, it may not be suitable for real time applications such as wireless communication applications.

6.3.1 Interpolation algorithm

To alleviate the effect due to timing errors, a digital technique based on interpolation is proposed in this dissertation. The basic idea is as follows: Assume that there is only one channel that has timing error, the output data are given as $\cdots, y_{i-3}, y_{i-2}, y_{i-1}, y_i + \Delta y_i + \delta y_i, y_{i+1}, y_{i+2} \cdots$ at the corresponding sampling instants of $\cdots, t_{i-3}, t_{i-2}, t_{i-1}, t_i + \Delta t_i + \delta t_i, t_{i+1}, t_{i+2}, \cdots$, where $\Delta y_i + \delta y_i$ represents the difference between the actual output and the ideal output. To correct the error added to y_i, one can estimate the actual value of y_i using the output data $\cdots, y_{i-3}, y_{i-2}, y_{i-1}, y_i + \Delta y_i + \delta y_i, y_{i+1}, y_{i+2} \cdots$ by interpolation if the timing error $\Delta t_i + \delta t_i$ is known. Although the clock skew Δt_i can be measured as discussed later in this chapter, the random jitter δt_i at a particular time instant is difficult to measure. However, the random jitter has zero mean and it does not affect the $SFDR$. Therefore, the error due to random jitter is assumed to be negligible (i.e. $\delta y_i = 0$ and $\delta t_i = 0$) to simplify the problem. It is also true when the effect of skew is the main concentration as in the wireless communication systems.

Using this basic algorithm, the entire ADC with calibration has an architecture as shown in Figure 6.4. Notice that the architecture has two major characteristics: First, the process is done *after* the conversion of each channel. Second, it is done in the *digital* domain, which means that there are no major changes in the analog circuit design. These characteristics are very important since any additional high-speed analog circuit will be difficult to design.

Although different interpolation algorithms can be used to estimate y_i, an iteration algorithm based on *Neville*'s method [59] is chosen. Since this method is considered to be a linear interpolation method, no extra unwanted tones such as inter-modulation products will be generated. Assume that there is only one channel that suffers from clock skew, the interpolation problem can be formulated as follows:

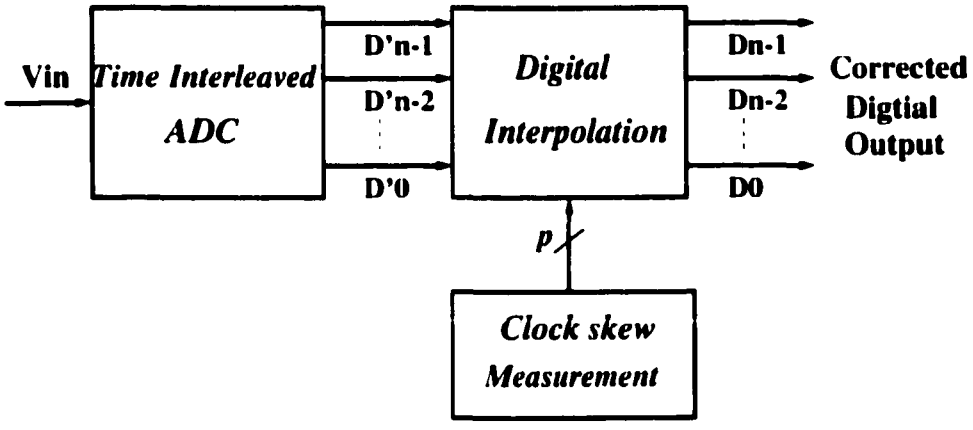

Figure 6.4 Block Diagram for the Proposed Timing Error Calibration Technique

Given an input signal $y(t)$ and a number of ADC' output samples

$$\cdots, (s_{i-2}, y_{i-2}), (s_{i-1}, y_{i-1}), (s_i, y_i + \Delta y_i), (s_{i+1}, y_{i+1}), (s_{i+2}, y_{i+2}) \cdots,$$

find the value of $y(s)$ for a given instant s where $\cdots, s_{i-2} = t_{i-2}, s_{i-1} = t_{i-1}, s_i = t_i + \Delta t_i, s_{i+1} = t_{i+1}, s_{i+2} = t_{i+2}, \cdots$ and the notation (s_l, y_l) represents the ADC' output sample y_l at instant s_l.

To make it convenient for the iteration operation in the interpolation algorithm, a two dimensional array $y_{i,l}$ is defined with l corresponding to the iteration step. The initial values $y_{k,0}$'s are equal to y_k's ($k \neq i$) and $y_{i,0}$ is equal to the output $y_i + \Delta y_i$ that suffers from clock skew. Assume that s is any time instant between $s_{i-L/2+0.5}$ and $s_{i+L/2-0.5}$ where L is the number of points (assume to be odd number) to be used in the interpolation. For $l < L$, the interpolation is based on the following iterative equation

$$y_{k,l} = \frac{(s - s_k)y_{k-1,l-1} - (s - s_{k-l})y_{k,l-1}}{s_{k-l} - s_k} \tag{6.3}$$

Using the above equation, a final result $y_{i+n/2-0.5,n-1}$ can be obtained. It represents the estimated value of the ADC output $y(s)$ at the time instant s. Therefore, based on the digital output values at the sampling instant of $\cdots, t_{i-2}, t_{i-1}, t_i + \Delta t_i, t_{i+1}, t_{i+2}, \cdots$, the correct digital output y_i at $s = t_i$ can be estimated if the timing error due to clock slew Δt_i is known. When more than one channel have skews, equation (6.3) still holds except

that $s_k = t_k + \Delta t_k$ for $k = -L/2 + 0.5 \sim L/2 - 0.5$, and the same algorithm can be used for correcting the output error for each channel.

6.3.2 Simulation results

To demonstrate the proposed technique, a time-interleaved ADC is simulated with the same setup as discussed in Figure 6.3. An example is given for the case when the ratio between the sampling rate and the input signal, R_s, is equals to 33. Figure 6.5 shows the ADC output spectrum before and after interpolation. In this case, a

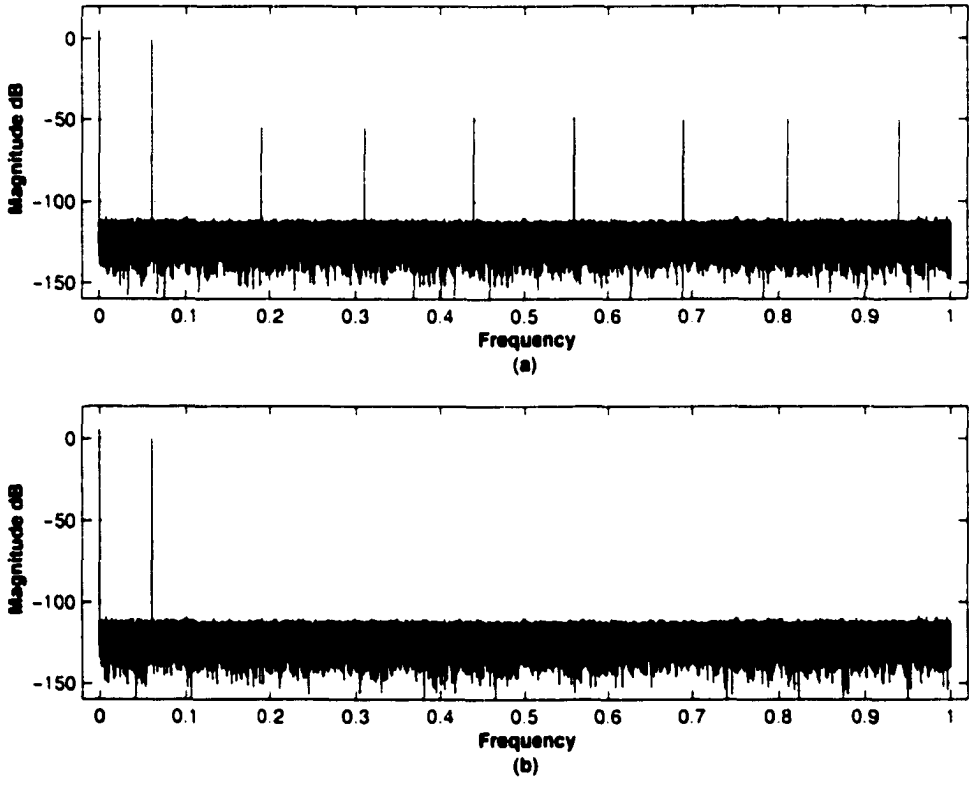

Figure 6.5 ADC Output Spectrum (a) before and (b)after 6-point interpolation

6-point interpolation (using 6 outputs to estimate y_i) is used. The output spectrum after interpolation shows that all the unwanted tones due to clock skew are greatly reduced and the $SFDR$ improves significantly. However the noise floor due to random jitter remains the same. In Figure 6.6 gives the simulation result of having three different

input signals. This result demonstrates that the interpolation algorithm is a linear algorithm and can be used to correct any input waveforms. Since $SFDR$ is the most important specification in wireless communication applications, simulation results are compared in terms of $SFDR$ vs. R_s in the following. The result is shown in Figure 6.7. Curves corresponding to the $SFDR$ before interpolation and after 2-point, 4-point, 8-point, 16-point and 32 point interpolations are shown.

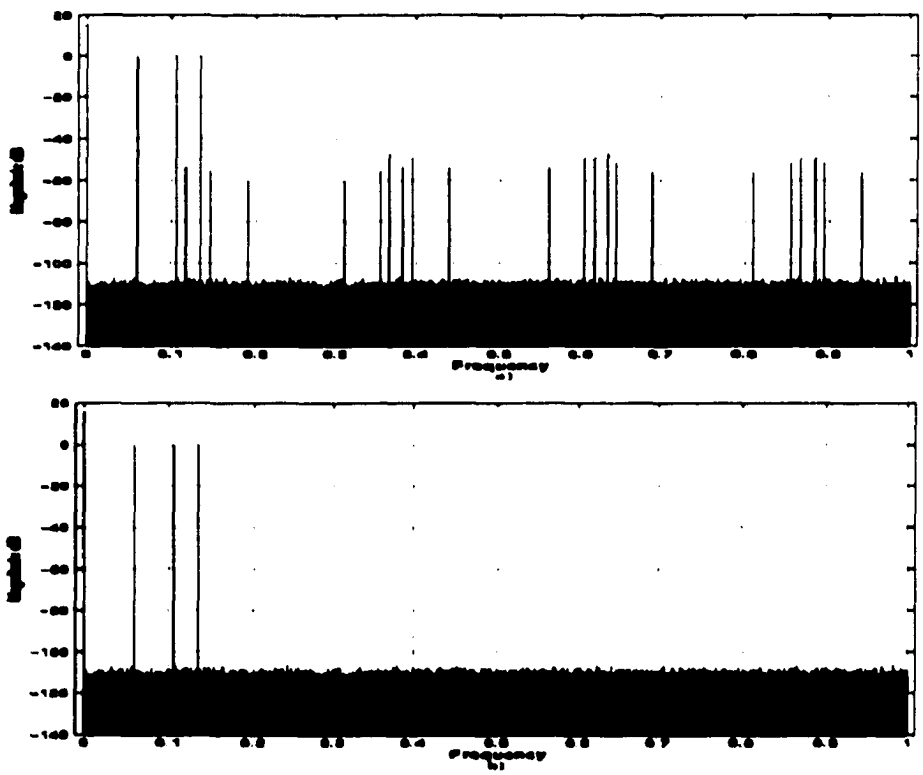

Figure 6.6 ADC Output Spectrum with Three Input Signals (a) before and (b)after 6-point interpolation

It can be observed that there is $20 \sim 60$dB improvement in the $SFDR$ after applying the proposed technique depending on the number of points used in the interpolation and R_s. In many cases, the unwanted tones virtually disappear under the noise floor so that the improvement is limited to the noise floor level. It can be further observed that as the input frequency approaches the $Nyquist$ rate ($R_s = 2$), the improvement of the proposed method becomes less effective. This is due to the fact that fewer sampling

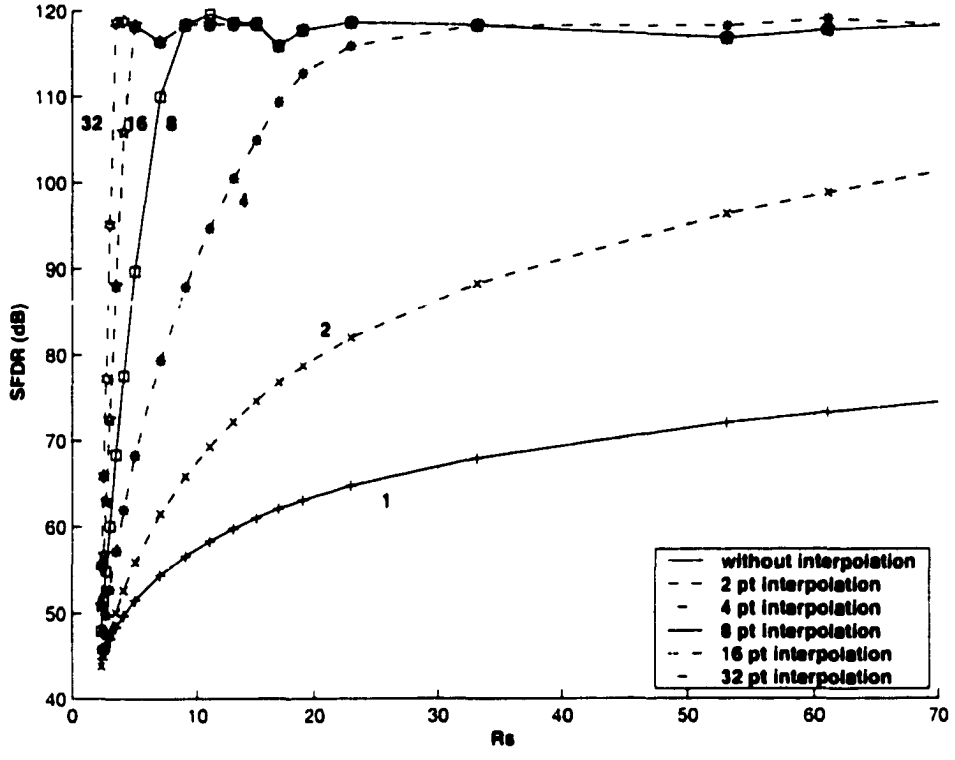

Figure 6.7 $SFDR$ vs. R_s for Interpolations with Different Number of Points

points are used in one cycle of sine wave in the interpolation to estimate the correct output. Notice that when more points are used in the interpolation, the performance will be improved, especially for input frequencies close to the *Nyquist* rate. For most applications, a maximum number of 32 points used for interpolation is sufficient since a $SFDR$ of higher than 100 dB can be achieved for an input signal frequency up to 1.5 times less than the *Nyquist* frequency (i.e. $R_s = 3$). Simulations are also performed for multi-tone input signals. The results are similar to the single-tone case, and no inter-modulation products are generated in the output spectrum. This indicates that the proposed technique is inherently linear.

6.3.3 Theoretical analysis

To further understand the effectiveness of the proposed technique, theoretical expressions for the $SFDR$ after interpolation are presented in this section. The first step is to analyze the case with only one channel suffering from clock skew. Errors due to clock skew will be analyzed for several cases namely : without interpolation, with 2-point interpolation and 3-point interpolation.

With $T_{in} = 1/f_{in}$ and $T_s = 1/F_s$, several terminologies are defined as follow: $\frac{T_s}{T_{in}} = \frac{1}{R_s}$, $\frac{\Delta t}{T_s} = c$ and $\frac{\Delta t}{T_{in}} = \frac{c}{R_s}$, where R_s is the *sampling ratio*, and c is defined as the *percentage* of clock skew for a given sampling period T_s. Assume that the channel that suffers from clock skew samples the input at the instant $t + \Delta t$ where t is the ideal sampling instant for this channel. For a complex sinusoid input without digital interpolation, the ADC output values for the instants before and after t can be written as $...., (t-2T_s, e^{j2\pi f_{in}(t-2T_s)}), (t-T_s, e^{j2\pi f_{in}(t-T_s)}), (t, e^{j2\pi f_{in}(t+\Delta t)}), (t+T_s, e^{j2\pi f_{in}(t+T_s)}), (t+2T_s, e^{j2\pi f_{in}(t+2T_s)}), ...$ and the error due to clock skew, err_1, can be expressed as

$$err_1 = e^{j2\pi f_{in}(t+\Delta t)} - e^{j2\pi f_{in}t} \tag{6.4}$$

where $e^{j2\pi f_{in}t}$ is the ideal output value at t. If a direct Fourier transform is applied to this error, it can be expressed as

$$ERR_1(\omega) = \pi(\delta(\omega - \omega_{in})e^{j\omega\Delta t} - \delta(\omega - \omega_{in})) \tag{6.5}$$

Since the channel that suffers from clock skew samples the input every MT_s seconds, this sampling mechanism causes unwanted tones to appear across the spectrum and the result can be expressed as $\frac{1}{MT_s}\sum_{k=\infty}^{\infty} ERR_1(\omega - k\frac{2\pi}{MT_s})$. Thus the magnitude of the unwanted tone at $k\frac{2\pi}{MT_s}$ is equal to $\frac{\pi}{MT_s}(|\ e^{j\omega_{in}\Delta t} - 1\ |)$. Since $SFDR$ is defined as the ratio between the input signal and the highest unwanted tone, it can be expressed as $SFDR = 20log_{10}\ |\ M/(e^{j\omega_{in}\Delta t} - 1)\ |$. Using the terminologies defined earlier, the $SFDR$ without interpolation, $SFDR_1$, can be written as

$$SFDR_1 = 20log_{10}\ |\ M/(e^{j2\pi c/R_s} - 1)\ | \tag{6.6}$$

To simplify the expressions for the derivation of the $SFDR$ of the 2-point interpolation ($SFDR_2$), define x_1, y_1, x_2 and y_2 as follow:

$$\begin{cases} (x_1, y_1) & \leftarrow & (t - T_s, e^{j2\pi f_{in}(t-T_s)}) \\ (x_2, y_2) & \leftarrow & (t + \Delta t, e^{j2\pi f_{in}(t+\Delta t)}) \end{cases}$$

The result after the 2-point interpolation, inp_2, for the time instant t can be written as:

$$inp_2 = y_1 + (t - x_1)\frac{y_2 - y_1}{x_2 - x_1}$$

and the error between the interpolated value and the ideal value, err_2, becomes

$$err_2 = inp_2 - e^{j2\pi f_{in}t}$$

With the terminologies defined above, the above equation can be rewritten as

$$err_2 = \frac{c}{1+c}e^{j2\pi f_{in}(t-T_s)} + \frac{1}{1+c}e^{j2\pi f_{in}(t+\Delta t)} - e^{j2\pi f_{in}t} \tag{6.7}$$

Similar to the case of without interpolation, $SFDR_2$ can be derived as

$$SFDR_2 = 20log_{10} \mid M/(\frac{1}{1+c}e^{j2\pi c/R_s} + \frac{c}{1+c}e^{-j2\pi/R_s} - 1) \mid \tag{6.8}$$

The $SFDR$ for the 3-point interpolation, $SFDR_3$, can also be derived using the above steps. x_0, x_1, x_2, y_0, y_1, y_2 can be defined as follow:

$$\begin{cases} (x_0, y_0) & \leftarrow & (t - T_s, e^{j2\pi f_{in}(t-T_s)}) \\ (x_1, y_1) & \leftarrow & (t + \Delta t, e^{j2\pi f_{in}(t+\Delta t)}) \\ (x_2, y_2) & \leftarrow & (t + T_s, e^{j2\pi f_{in}(t+T_s)}) \end{cases}$$

The result after the 3-point interpolation, inp_3, for the time instant t, and the error between the interpolated value and the ideal value, err_3, are given as:

$$\begin{aligned} inp_3 &= ((t - x_0) * y_{12} - (t - x_2) * y_{01})/(x_2 - x_0) \\ err_3 &= inp_3 - e^{j2\pi f_{in}t} \end{aligned}$$

where,

$$\begin{aligned} y_{01} &= ((t - x_0) * y_1 - (t - x_1) * y_0)/(x_1 - x_0) \\ y_{12} &= ((t - x_1) * y_2 - (t - x_2) * y_1)/(x_2 - x_1) \end{aligned}$$

Using the terminologies defined above, err_3 can be expressed as

$$err_3 = \frac{1}{2}\left(\frac{2}{1-c^2}e^{j2\pi f_{in}(t+\Delta t)} + \frac{c}{1+c}e^{j2\pi f_{in}(t-T_s)} - \frac{c}{1-c}e^{j2\pi f_{in}(t+T_s)}\right) - e^{j2\pi f_{in}t} \tag{6.9}$$

The corresponding $SFDR$ can be written as

$$SFDR_3 = 20log_{10}\left|2M/\left(\frac{2}{1-c^2}e^{j2\pi/R_s} + \frac{c}{1+c}e^{-j2\pi/R_s} - \frac{c}{1-c}e^{j2\pi/R_s} - 2\right)\right| \tag{6.10}$$

From the above analysis, it can be observed that all the errors after interpolation are only related to the input samples and no inter-modulation products are produced. Thus, as stated before, the proposed technique is linear, which is a very important requirement for wireless communication applications. Furthermore, although the above analysis is obtained by assuming that there is only one channel that suffering from clock skew, the same steps can be used to analyze the cases of having multi-channels suffering from clock skews, except that the expressions become more complicated. Nevertheless, the analysis for the case of having only one channel that suffers from clock skew can be used to predict the results for having multi-channels (and all the channels) that suffer form clock skews. The comparison is shown in Figure 6.8. The curves with markers represent the results from simulation. The above theoretical analysis matches well with the simulation results before the magnitudes of the unwanted tones are lower than the noise floor caused by random jitter and quantization error. Therefore, the above analysis steps can be used for predicting the $SFDR$s for the case where more than one channel suffer from clock skews.

For high order interpolations, analysis similar to the ones shown above can also be carried out. However, the expressions become complicated as the order increases. An empirical formula is proposed to estimate the $SFDR$s so that, during the design stage, it can be used as a guide line for selecting the required order for a given $SFDR$ requirement and a given maximum input frequency. This empirical formula can be written as

$$SFDR_L \approx 20log_{10}(R_s^{dd} \cdot C) \tag{6.11}$$

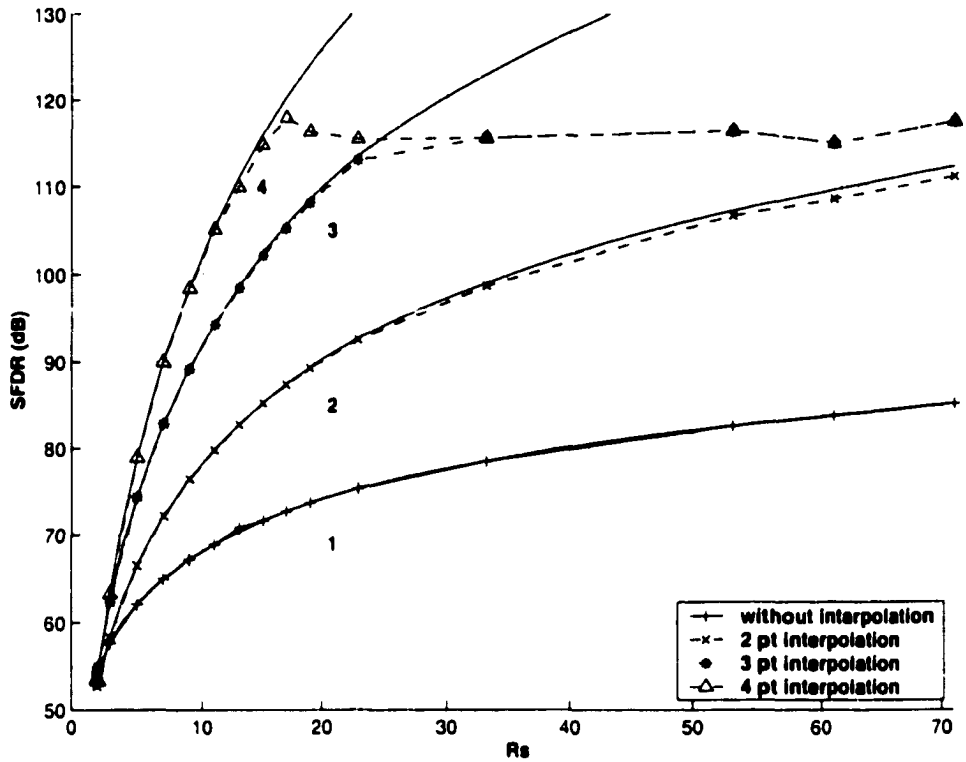

Figure 6.8 Comparison of the Analytical Results and the Simulated Results (curves with markers)

where the parameter *dd* is related to the number of points (L) used in the interpolation and the parameter C is related to the magnitudes of the clock skews. These two parameters are found by fitting equation 6.11 in to the simulated results. Table 1 shows the values of these two parameters after fitting. Based on these results, the parameter *dd* is found to be approximately equal to $7^{log_{10}L}$, and the parameter C is roughly equal to $10^{C_L/20}/2^{dd}$ where C_L is equal to $SFDR_1$ at $R_s = 2$. With these approximations, the empirical formula gives very good results for L less than 32 as shown in Figure 6.9. As before, the curves with markers represent the simulated results. For L greater than 32, the $SFDR$ increases rapidly as R_s increases and the results predicted using equation 6.11 start to deviate from the simulation results. However, the same formula can be used for $L > 32$ if the parameters *dd* and C are increased by 25% and decreased by one-tenth for every 50% increase in L, respectively.

Table 6.1 Parameters dd and C for the Empirical Formula

Points L	Order dd	Offset C
1	1	83.97
2	1.87	36.01
4	3.62	8.86
8	6.40	1.333
16	10.1	6.67e-4
32	17	6.67e-5

6.4 Digital Background Clock Skew Measurement

As mentioned above, the interpolation can only be carried out if Δt_i (clock skew for a particular channel) is known. This can be obtained through calibration cycles either before or during normal operation. Calibration before normal operation can be carried out using the following procedure. Initially, each channel is assumed to be an ideal ADC. During the calibration cycle, a pre-determined linear ramp signal is fed to the ADC. Ideally, the digital output will be proportional to the sampling instant and the corresponding digital output values can be obtained. If clock skew exists, the output values will be different from the ideal values. Hence, clock skews can be measured by comparing the difference between the actual output values and their corresponding ideal values. The measured clock skew for channel i can be determined as

$$\Delta t_i = \frac{\Delta D_i}{\text{slope of the ramp}} \tag{6.12}$$

where ΔD_i is the difference between the actual output value and the ideal output value at the ideal sampling instant t_i.

For the above measurement, the ramp signal should have a slew rate fast enough to obtain an accurate clock skew measurement. It is assumed that the minimum clock skew need to be measured is $skew_{min}$ seconds and the ADC has n bits of resolution. To measure the clock skew, ΔD_i has to be greater than 1 LSB. Then, for a given ADC full

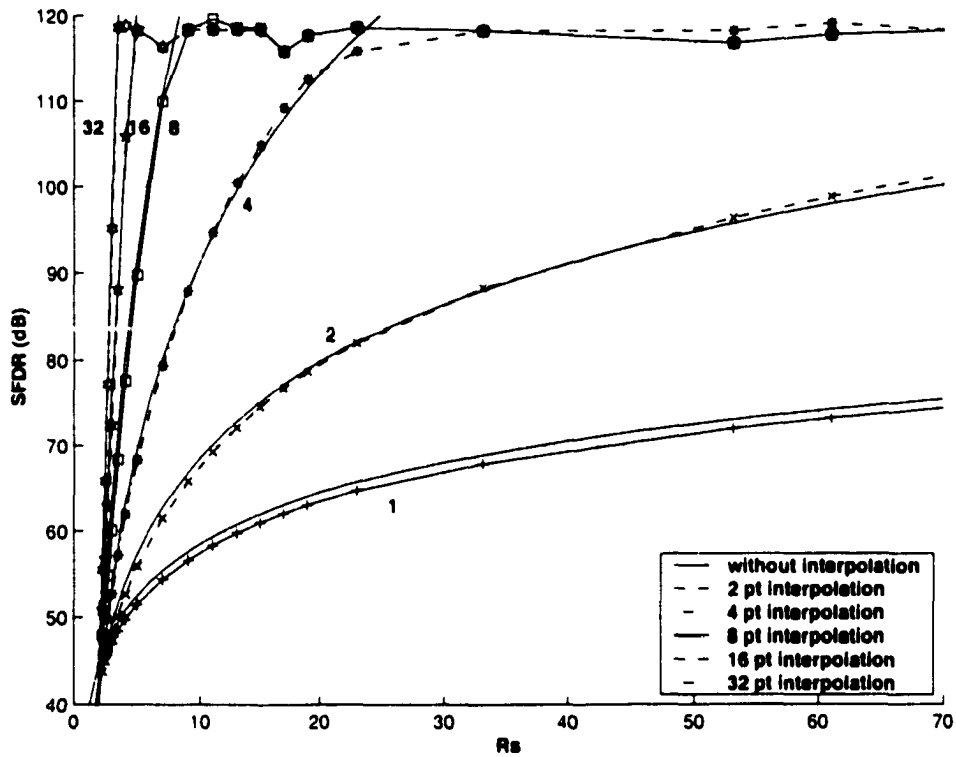

Figure 6.9 Comparison of the Simulated Results and the Results from Empirical Formula

input range, $V_{fullscale}$, the minimum slope of the ramp can be determined as

$$min_{slope} = \frac{V_{fullscale}}{2^n \times skew_{min}} \tag{6.13}$$

Although the accuracy of the above clock skew measurement is always affected by random jitter, the effect of random jitter is usually smaller than the effect of clock skew. In addition, the random jitter will have zero mean. Therefore, by repeating the above clock skew measurement procedure a number of times, it will average out the effect of random jitter and an accurate measurement of clock skew for each channel can be obtained. By performing this method on every channel, the clock skews for different channels can be obtained.

In the literature, there are also other methods used in instrumentation equipments to accurately measure the clock skews, for example in [60]. However, clock skews may vary due to aging and temperature variation. Furthermore, the actual clock skews may vary

during normal operation after the calibration cycle is finished. This is due to the fact that different circuits or systems are applied to ADC' during normal operation. As a result, the deterministic component of the power supply and substrate noise that affects the clock skew will be different. Therefore, a background measurement may be required.

A method of background measurement that requires no major changes on the analog part of the ADC' is proposed in this dissertation. Figure 6.10 shows the required modification on each channel. A ramp signal V_r with period of T_S is added to the input signal before each S/H samples the signal. If the input signal is assumed to be zero, the output of each ADC' will be a DC' value proportional to the clock skew and the slope of the ramp as discussed above. If the input is *any* signal with zero mean, then the clock skew can be estimated by feeding the output of each ADC' to a digital low pass filter (LPF) as shown in (Figure 6.10) such that a DC' value proportional to the clock skew can be obtained. (Notice that the LPF also serves the purpose of filtering out any random jitter, which may affect the measurement of clock skew.) Since this DC' value appears at the output of the channel, the estimated clock skew value for channel i, ED_i, has to be subtracted from the output channel i before interpolation can be performed. Notice that it is quite possible that none of the ED_i's are equal to zero since the sampling instants of any channels may not happen at the instant when V_r crosses zero. To have meaningful results from the interpolation process, one of the channels is assumed to have no clock skew and is used as reference. Then the relative clock skew for channel i can be calculated by subtracting the estimated clock skew value of the reference channel from ED_i.

The digital LPF is required to have a very low cutoff frequency and high order so that low input frequency signal will not be attenuated when ED_i is subtracted from the output of each channel. To minimize the hardware requirement, the LPF can be realized based on averaging N number of outputs using a simple accumulator as shown in Figure 6.10. The output of the accumulator is then used to update the estimated clock skew value (stored in register R_i) at a rate of F_s/MN. To obtain a good clock

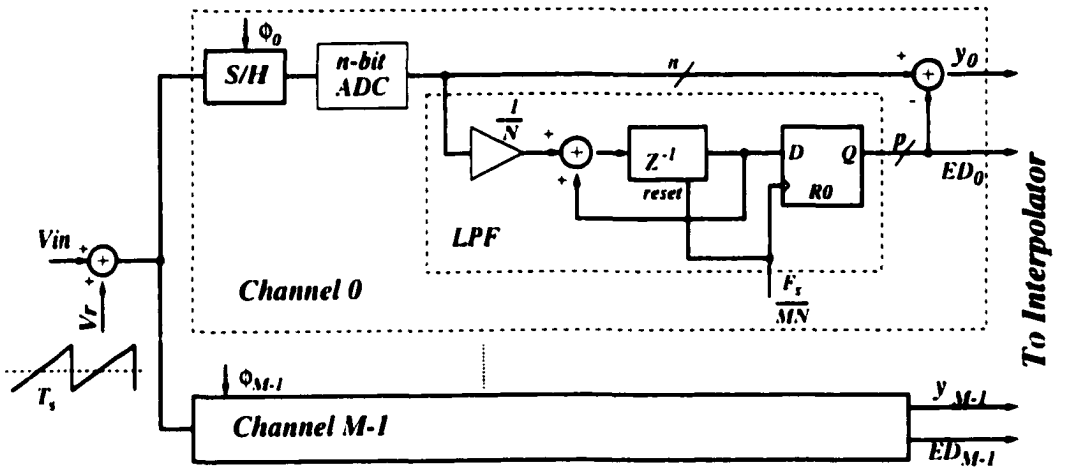

Figure 6.10 Proposed Background Clock Skew Measurement

skew estimation and to allow very low input frequency signal, a large N should be used.

Since a DC value proportional to the clock slew of each channel will be added to the input signal during the background skew measurement, the dynamic range of the ADC will be reduced. To estimate this degradation, first assume that the input is zero. Then a clock skew of $\pm 50\%$ of T_s can be measured by each channel when the slope of the ramp input is equal to $V_{fullscale}/T_s$. Since $V_{fullscale}$ is represented as 2^n at the ADC output where n is the resolution of the ADC, a given clock skew range of $\pm x\%$ ($< \pm 50\%$) of T_s will have a value within a range of 2^p where p can be determined to be $\lceil log_2(2^n x/50) \rceil$ bits and is used for representing the value of ED_i. Therefore, the reduction in dynamic range of the ADC is given as $2^n - 2^p$. As an illustration, if there is $\pm 5\%$ of clock skew and the ADC is 14 bits, p will be equal to 11 bits and the dynamic range of the ADC will drop from 84.3 dB to 83.1 dB. From this example, it can be observed that even though the clock skew is quite large, the reduction in dynamic range due to background clock skew measurement can still be neglected. For a more practical clock skew of $\pm 0.5\%$ of T_s, p is equal to 8 bits and the reduction in dynamic range is 0.15 dB.

For practical realization, addition of V_{in} and V_r in Figure 6.10 can be implemented by a slight modification of the S/H in each channel as shown in Figure 6.11. For conventional S/H, nodes A and B are usually connected to the common-mode voltage.

Instead, they are connected to the ramp input such that the addition of V_{in} and V_r can be obtained. As a result, there are no major changes on the analog parts of the ADC. If the clock skew is within $\pm 0.25T_s$, the ramp signal can also be replaced with a triangular wave or a sine wave with a frequency of $1/T_s$ (Figure 6.12), which can be generated at high frequency through the use of the main ADC clock that has the same frequency. When sine wave is used, the ED_i should be corrected by the inverse sine function before applied to the interpolation process. Notice that the overall noise floor of the ADC will increase slightly due to the random jitter associated with the ramp (triangular or sine) input. However, the $SFDR$ will not be degraded as long as the connections between the ramp input and the individual S/Hs are short and identical by laying out all the switches S_1s (Figure 6.11) of different S/Hs at one place. Notice that mismatches between S_1s will not affect the overall $SFDR$ since any clock skews due to this mismatch will be measured by the background clock skew measurement and the output values will be corrected by the interpolation algorithm.

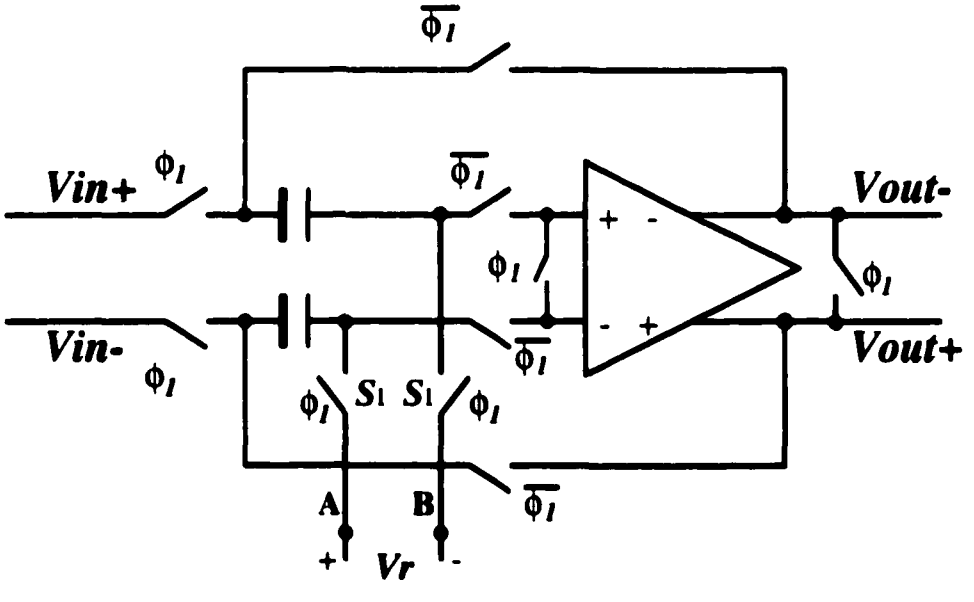

Figure 6.11 Sample-and-Hold/Adder in Each Channel used in the Proposed Background Clock Skew Measurement

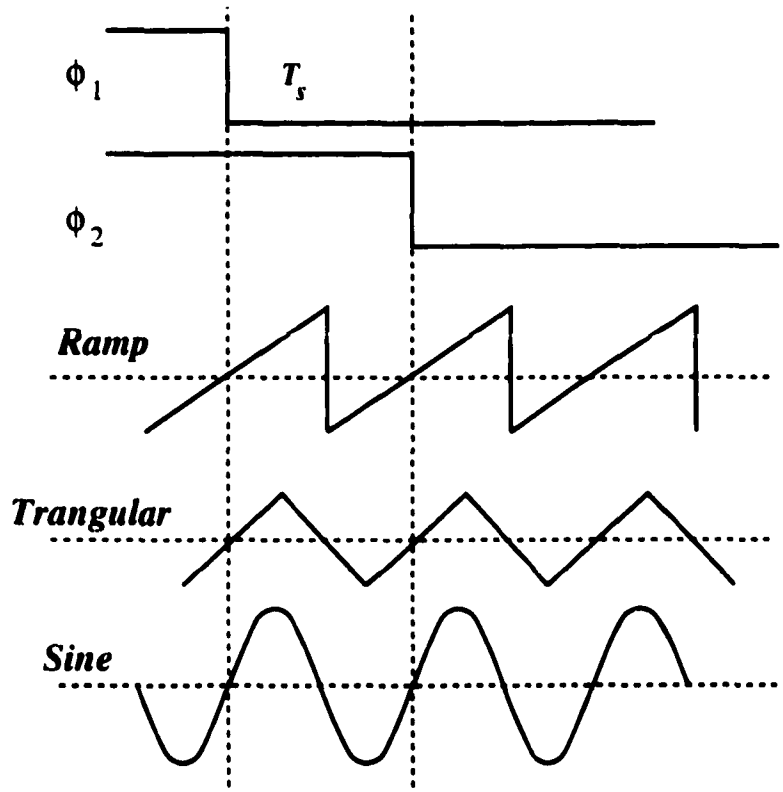

Figure 6.12 Different Kinds of Vr that can be Used for the Background Clock Skew Measurement

6.5 Practical Consideration for Digital Interpolation

Although the interpolation algorithm can be realized as an iterative process described by Equation (6.3), it may not be fast enough to produce outputs at a data rate of F_s unless a pipeline architecture is used. Fortunately, it can be shown that the interpolation algorithm can be realized as a FIR filter. Therefore, the hardware complexity can be greatly reduced and high-speed operations can be obtained. As an example, assume that there is only *one* channel that suffers from clock skew, and a 3-point interpolation is used. Then, equation (6.9) can be rewritten as:

$$inp_3 = \frac{1}{2}\left(\frac{-c}{1-c}y_n + \frac{2}{1-c^2}y_{n-1} + \frac{c}{1+c}y_{n-2}\right) \tag{6.14}$$

Equation (6.14) is similar to the expression of a 3rd order FIR filter. Interpolation with more than 3 points can also be derived using the same principle. For a L-point

interpolation, it can be observed that a Lth order FIR filter is required, and all the FIR filter coefficients will be related to the clock skew value as illustrated in equation (6.14). Notice that these coefficients will be fixed until a new clock skew value is obtained from the background clock skew measurement. Since the clock slew value does not change rapidly over time, a slow but simple hardware can be used to compute all the filter coefficients using an iteration operation similar to Equation (6.3). Thus, the computation complexity can be relatively simple even for high order interpolation. In the literature, a 24-point digital interpolation has also been applied to calibrate a pipelined ADC [47]. Therefore, high order interpolation as high as 32 points is still quite feasible in practical implementation.

In the above discussion, only one channel is assumed to be suffered from clock skew and, hence, the interpolation process is only required every MT_s seconds. In general, all the channels may have clock skews, and every output has to be computed using the same interpolation process. In this case, the same FIR filter can be used to generate every output, except that the filter coefficients have to be time varying and rotated with a period of MT_s according to the corresponding channels. In other word, a new set of pre-computed filter coefficients has to be loaded to the filter when the filter is used to compute a new output. Since there are M channels, the number of filter coefficient sets is equal to M.

Similar to most digital signal processing techniques, finite word length will affect the results of interpolation as illustrated in Figure 6.13. Curve (1) is the result for no clock skew correction. Curve (2) is the result for a 6-point interpolation with floating point computation. Curve (3) is the result for the same interpolation with only 18 bits of accuracy in the computation. It can be observed that there is some degradation in $SFDR$ due to finite word length effects since extra truncation errors increase the noise floor. Nevertheless, there is still $20 \sim 40$dB improvement over the one without using the proposed technique. Since the interpolation algorithm can be realized as a FIR filter, the word length requirement is similar to the requirement for implementing a FIR filter.

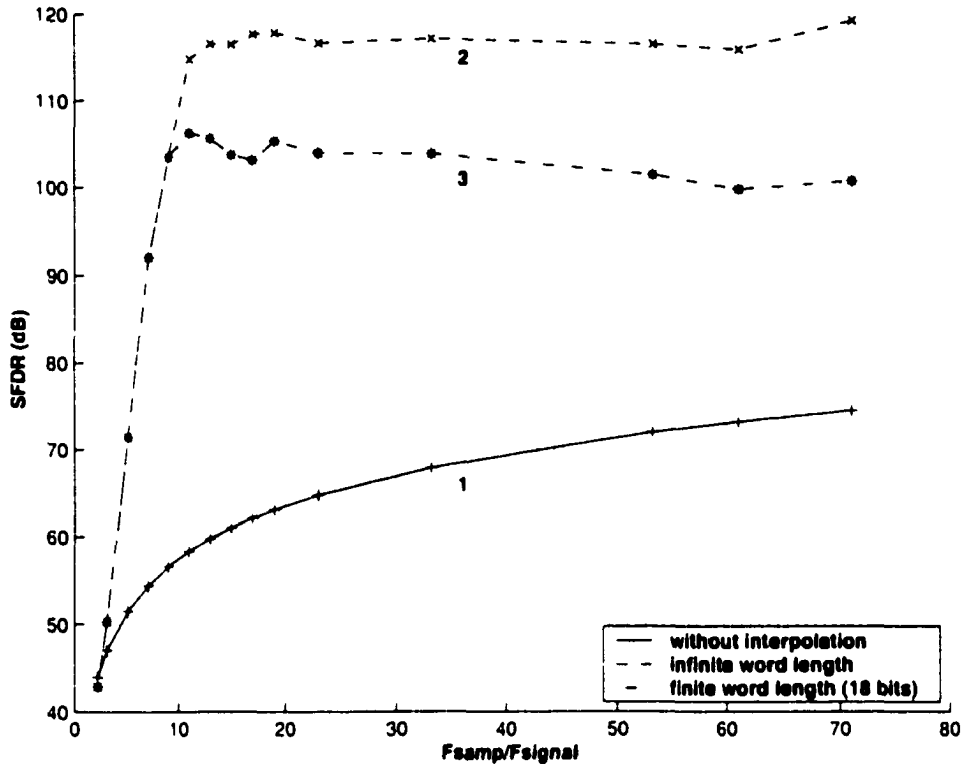

Figure 6.13 Effect of Finite Bit Length

6.6 Conclusion

A digital background calibration technique for reducing timing error in time-interleaved *ADC*' is proposed in this dissertation. The proposed technique is based on digital interpolation, which requires timing information of each channel. To obtain this information, a digital background clock skew measurement method is also proposed. Besides the practical issues in implementing the proposed technique, simulation results as well as theoretical analysis for addressing the limitation of this technique are presented. An improvement in $SFDR$ of 20 $\sim$ 60 dB can be obtained depending on the input frequency and the number of points used in the interpolation. The major advantage of this technique is that no major changes in the analog circuits are required for the *ADC*', and most of the calibration and processing steps are carried out in the digital domain. Given the steady improvement on the digital signal processing hardware due to advances in

CMOS technologies, the proposed technique can be used to realize high-speed and high accuracy ADC's for communication as well as instrumentation applications.

7 CALIBRATION OF CHANNEL MISMATCHES IN TIME-INTERLEAVED A/D CONVERTER

7.1 Introduction

As discussed in the Chapter 5, time-interleaved structure takes the advantage of parallelism and has the potential of enhancing the throughput of ADC significantly. However, additional errors need to be taken care of in this particular structure. Besides channel timing error mismatch discussed in the previous chapter, its accuracy is also seriously affected by amplifier gain errors, the offsets of comparators and reference sources. These errors and offsets will result in nonlinearity and distortions in the output digital values.

To alleviate this problem, some methods have been proposed, including the self-calibration techniques for the two-channel pipeline structure [48][61]. However, most of the self-calibration techniques usually require complex digital circuits, and hence increase the design difficulties and the required die area.

In this chapter, a method for reducing the gain error and offset effects by randomly rearranging the conversion channels in the time-interleaved ADC structure is proposed. The major advantage of this technique is that it requires low digital hardware complexity and therefore, small die area.

7.2 Device Mismatches in Time-Interleaved ADC

To explain the proposed method, the time-interleaved ADC is shown again in Figure 7.1.

Before introducing the proposed method, several assumptions are made. Since dif-

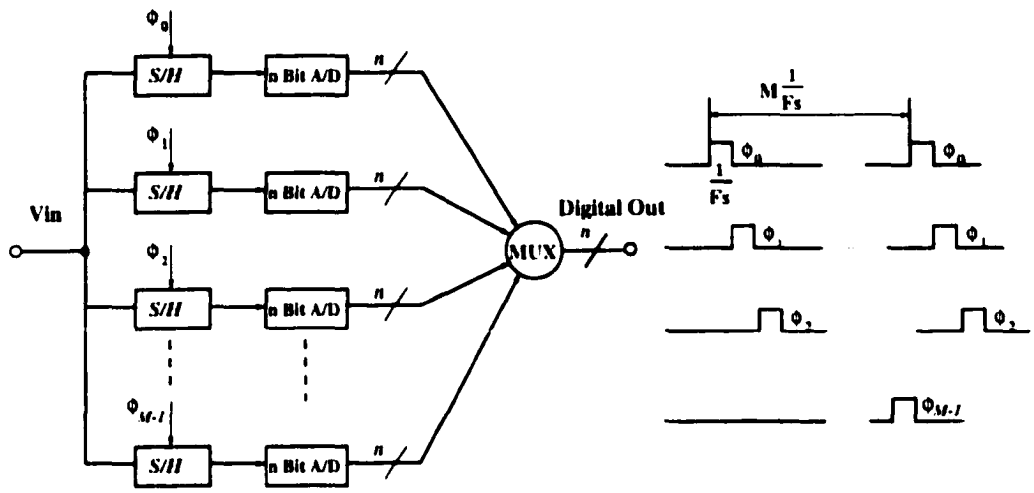

Figure 7.1 Time-interleaved ADC

ferent structures can be used to implement each n-bit ADC, it is assumed that each converter is an algorithmic converter. Inside each converter, a comparator is used to compare the input analog signal to a voltage reference and make a 1-bit output decision at each clock cycle. During the same clock cycle, the analog signal is subtracted from the reference voltage if the output of comparator is a logic high. The subtracted value is then multiplied by 2 to generate the "residue" value. If the output of the comparator is a logic zero, the "residue" value is twice of the analog input. In the next clock cycle, the residue is sent back to the input for computing the next bit. After n clock cycles, n-bit digital output will be generated.

The residue value V_i for each converter at the time interval i can be written as follow:

$$V_i = A_i \times (V_{i-1} - D_i \times V_{ref}) \quad i = 1, \cdots n \tag{7.1}$$

$$\begin{aligned} D_i &= 1 \quad V_{i-1} < V_{ref} \\ D_i &= 0 \quad V_{i-1} \leq V_{ref} \end{aligned} \tag{7.2}$$

where A_i is the amplifier gain and its nominal value is equal to 2. V_i is the residue value at the ith time interval and the input for the $i+1$th interval. D_i is the ith digital output bit. When $i = 1$, V_{i-1} is equal to V_0 which represents the input analog sample V_{in} stored

in the S/H inside each channel. When $i = n$, V_i is the final residue analog value. The input analog signal to each algorithmic converter at the ith interval can also be written as:

$$V_{i-1} = D_i \times V_{ref} + \frac{V_i}{A_i} = V_i^{DAC} + \frac{V_i}{A_i} \tag{7.3}$$

where V_i^{DAC} is equal to $D_i \times V_{ref}$.

To better simulate the real distribution of gain errors and offsets, the mean value of component mismatches is assumed to be equal to zero. For example, the average amplifier gain and offset for different algorithmic ADC's are equal to two and zero, respectively. This is an acceptable assumption considering that the component variations are caused by process variation in both positive and negative directions.

In references [51] and [62], the gain and offset mismatches in the Time-interleaved ADC had been analyzed. The detailed expressions are given in the above two references and will not repeat here. Figure 7.2 illustrates the effects from these two mismatches. It is simulated with an 8-channel, 8-bit time-interleaved ADC converter. $MATLAB$ is used to compute the output spectrum. The simulation considers both gain errors and offsets. For offsets, both comparator offsets and reference voltage offsets are considered in the simulation. The gain errors and the offsets between channels are assumed to be ±2.5% and ±5% of their nominal values (gain=2 and reference voltage=0.5), respectively. The mean values are set to be the nominal values. The simulation result suggests that the error effects can be dominant factor that hinders the ADC operation.

7.3 Channel Randomization

To reduce the gain error and offset effects, a method that is based on randomly choosing the ADC channels is proposed. Since randomly choosing the channels will reduce the correlation between different converted digital outputs, and also the gain errors and offsets from different channels will appear in both positive and negative directions, the effects caused by the component mismatches tend to be averaged out.

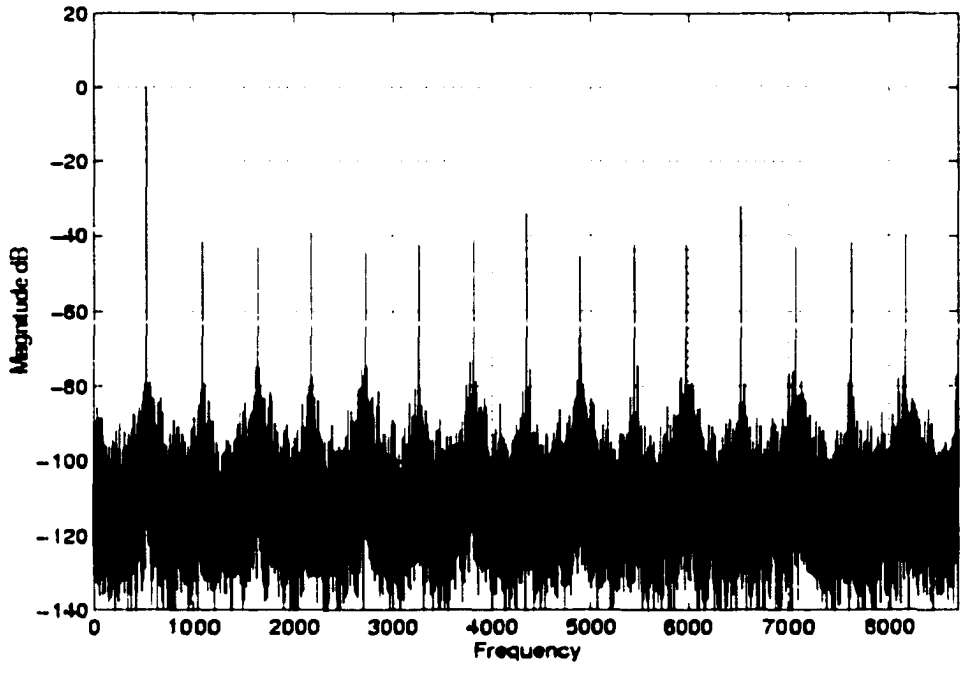

Figure 7.2 The Simulation Result without Randomization

Consequently, reduction in distortion level can be expected.

To randomly choose ADC' channels, many methods can be used depending on the structure of the individual ADC's. In this dissertation, randomization is obtained by randomly sending the residue signal of each algorithmic ADC' converter back to one of the ADC' channels. Therefore, conversion of one input sample is not based on only one channel as the conventional time-interleaved structure does, but rather based on several channels that are randomly chosen.

The proposed structure is shown in Figure 7.3. The $RMUX$ indicates the array used for randomly sending the residue signals back to the ADC' channels. The inputs to the $RMUX$ array are the residue signals from the outputs of the ADC' channels and the input signal from the S/H. The outputs of the $RMUX$ are connected to the input of the M ADC' channels. The array works in a manner that it can randomly connect its input signals to any output ports without joints between different output ports. For example, the residue signal from the ith converter can be randomly assigned to one

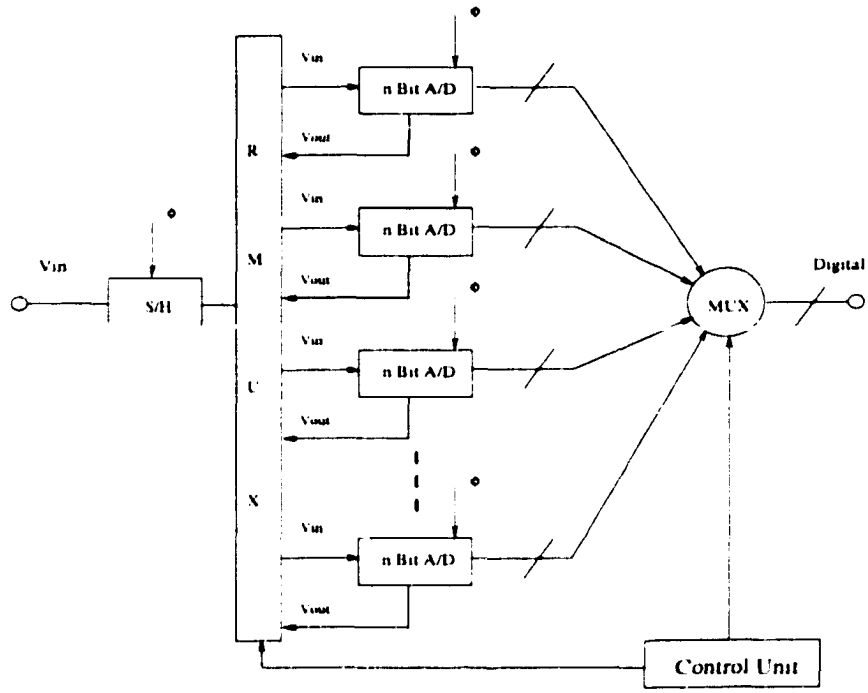

Figure 7.3 Time-interleaved ADC with Random Rearrangement

of the M channels during each clock period. The MUX at the output of the ADC converter has to be modified so that it will not select the M ADC converter outputs in a sequential way but rather the way that corresponds to the actual output sequence under the control of the Control Unit.

To show the effect of the channel randomization, the noise power due to gain error can be analyzed as follows. By using equation 7.3, the output of the algorithmic ADC converter in each channel can be written as [61] :

$$V_{in} = V_1^{DAC} + \frac{V_2^{DAC}}{A_1} + \cdots + \frac{V_n^{DAC}}{A_1 \cdots A_{n-1}} + \frac{V_n}{A_1 \cdots A_n} \tag{7.4}$$

where A_i is the amplifier gain at the ith time interval. A_i can be written as $A_i = A + \Delta_i$, where A is the nominal gain and Δ_i is the gain error at the ith time interval. The final item of $\frac{V_n}{A_1 \cdots A_n}$ is the inherent quantized error. By neglecting the last term, the above

equation can be rewritten as:

$$\begin{aligned} V_{in} &= V_1^{DAC} + \frac{V_2^{DAC}}{A(1+A^{-1}\Delta_1)} + \cdots \\ &\quad \frac{V_n^{DAC}}{A^{n-1}(1+A^{-1}\sum_{i=1}^{n-1}\Delta i)} \\ &\simeq V_1^{DAC} + \frac{V_2^{DAC}}{A}(1-A^{-1}\Delta_1) + \cdots \\ &\quad \frac{V_n^{DAC}}{A^{n-1}}(1-A^{-1}\sum_{i=1}^{n-1}\Delta i) \\ &= \sum_{i=0}^{n-1}\frac{V_i^{DAC}}{A^i} - \sum_{i=2}^{n}\frac{V_i^{DAC}}{A^i}\sum_{j=1}^{i-1}\Delta_j \end{aligned} \tag{7.5}$$

The first item in equation (7.5) can be recognized as the ideal digital representation of V_{in}. The second term is raised due to gain errors. The noise power V_n^2 due to gain errors can be expressed as:

$$\begin{aligned} V_n^2 &= \frac{V_2^2}{A^4}\Delta_1^2 + \frac{V_3^2}{A^6}(\Delta_1+\Delta_2)^2 + \\ &\quad \frac{V_4^2}{A^8}(\Delta_1+\Delta_2+\Delta_3)^2 + \cdots \\ &= \frac{V_2^2}{A^4}\Delta_1^2 + \frac{V_3^2}{A^6}(2\Delta_1\Delta_2+\Delta_1^2+\Delta_2^2) + \\ &\quad \frac{V_4^2}{A^8}(\Delta_1^2+\Delta_2^2+\Delta_3^2+ \\ &\quad 2\Delta_1\Delta_2+2\Delta_1\Delta_3+2\Delta_2\Delta_3) + \cdots \end{aligned}$$

If channel randomization is used, Δ_i can be assumed to be independent to each other. Therefore, the expected value for $\Delta_i\Delta_j$ is equals to zero for $i \neq j$. However, this is not the case when channel randomization is not used. Therefore, using channel randomization, a reduction in the noise power is expected. Furthermore, the amount of reduction is influenced by different distributions in Δ_i.

7.4 Hardware Implementation

When *RMUX* consists of M input ports and M output ports, there are $M!$ ways to select the inputs to the output ports. The hardware required to implement the

RMUX will be very complicated unless an efficient architecture is used. Furthermore, the allowable time to randomly arranging the channel is also limited by the sampling rate. Therefore, the random arrangement of channels has to be finished within one clock period. In the following, a *matching* means an arrangement to randomly connect M input signals to M output ports.

To implement this matching, a *Benĕ* network [63] can be used. *Benĕ* network is a kind of matching network. It can do a one-to-one mapping with disjoint paths so that a path for each input signal to be randomly connected to any one of the output ports will be provided. The structure of a *Benĕ* network is shown in Figure 7.4.

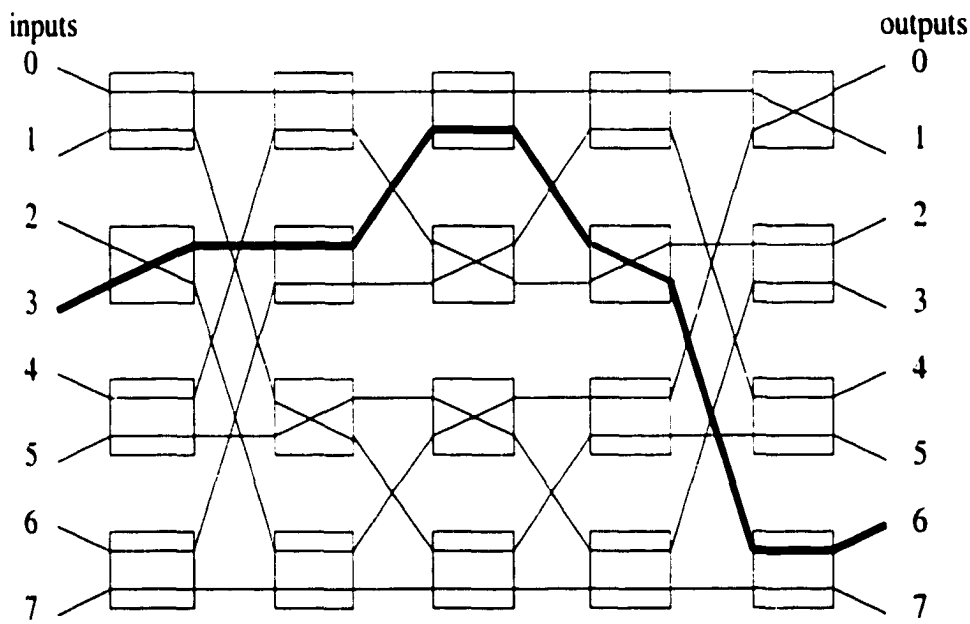

Figure 7.4 One Matching Path with *Benĕ* Network

The building blocks of a *Benĕ* network are 2×2 switches which are represented by rectangles in the figure. Assuming ports A and B are the input ports of a switch and port C and D are the output ports of the switch, the 2×2 switch can only connect port A to port C and port B to port D *or* port A to port D and port B to port C. The connection of the switch is controlled by a 1-bit random data which is assumed to be

generated using a linear feedback shift register. A network having 8 input and 8 output ports requires 20 switches. In general, the number of required switches is

$$N \log_2 N - N/2 \tag{7.6}$$

for N input and N output ports.

To illustrate the random arrangement of the inputs to the output ports, a particular arrangement made by an 8 input and 8 output ports network is shown in Figure 7.4. By controlling each 2×2 switch, the input signals can be rearranged to the output ports without joint path as shown in the figure. A path from input port 3 to output port 6 is highlighted. It can be checked that this network can provide all the possible 8! ways of connections.

It is obvious that there are different ways to implement the switch. A possible realization of the 2×2 switch is to use transmission gates. It requires one digital latch for storing the connection information and 8 transistors for implementing the 4 transmission gates. If the residue values from the ADC channels are the input signals to the *Benĕ* network, each signal have to propagate through $2\log_2 N - 1$ switches. Therefore, the propagation delay may not be short enough for high speed operation.

To achieve short propagation delay, one possibility is to use a *cross-bar switching network* in Figure 7.5, where each cross point will be a switch transferring the analog signal from column i to row j. A control code C_{ij} is assumed to be used to turn on only one switch in each row. If the control codes can be randomly generated for each row such that there are no joints between different ports, channel randomization can be achieved.

The input ports of the cross-bar network are connected to the outputs of conversion step $k-1$, its output ports are connected to the inputs of conversion step k. To generate the control code, the inputs to the *Benĕ* network are assumed to be the input port addresses, ie. 1 to N. After randomizing the connections made by the 2×2 switches, the addresses will be transferred to different output ports. For example, the address of

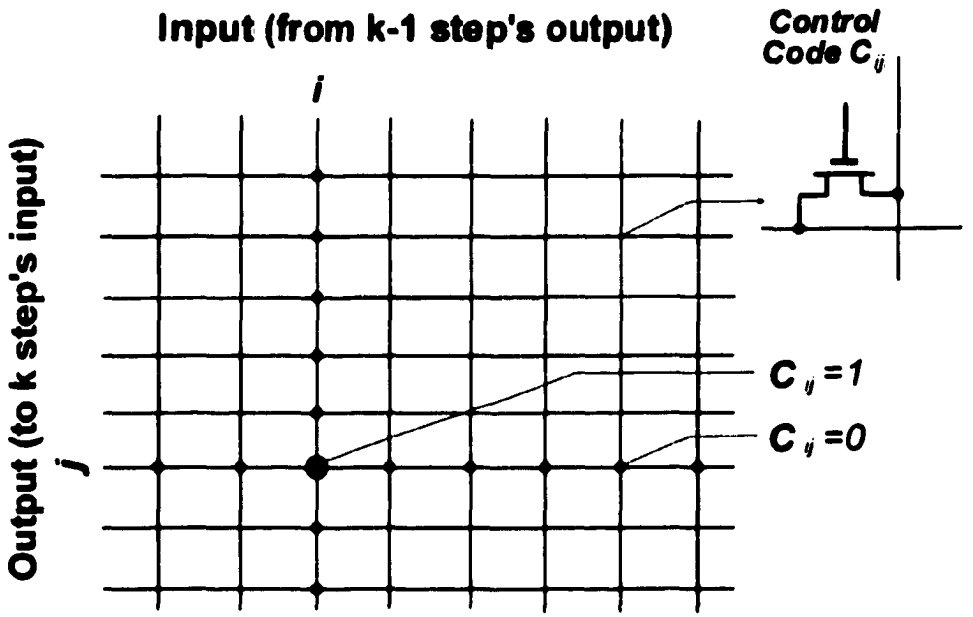

Figure 7.5 Cross-bar Switching Network

the input port i is transferred to the output port j. Then the control code is assigned as $C_{ij} = 1$. Therefore, the ith switch on row j in the cross-bar network will be turned on which indicates that the ith output from step $k-1$ is transfered to the jth input for step k. Because the control code C_{ij} is randomly generated, channel randomization is obtained. Since the *Benĕ* network can now be implemented digitally using a pipelined structure, generation of code for each row can be obtained at very high speed. Therefore, channel randomization can be completed within one clock period.

7.5 Simulation Results

To demonstrate the proposed method, the same 8-channel, 8-bit time-interleaved *ADC* converter is simulated. In Figure 7.6, the noise is almost evened at the same level and the total harmonic distortion is reduced by more than 17 dB.

Other distributions of gain errors and offsets are also simulated. In all cases, the proposed method can provide improvement in the *THD* by more than ten *dB*s. It

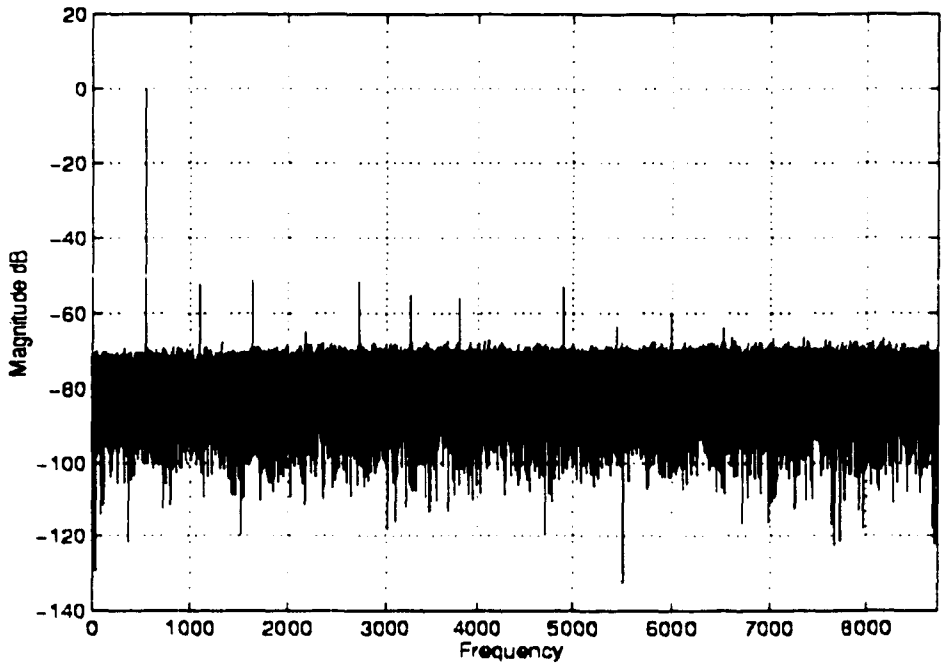

Figure 7.6 The Simulation Result with Randomization

should be pointed out that no matter what kind of component mismatch distribution is introduced, the harmonic distortion will be reduced. However, different distributions of gain errors and offsets will result in different improvement as discussed before.

Comparing the two simulation results, it can be observed that the noise floor in Figure 7.6 is higher than that of Figure 7.2. The reason comes from the fact that randomization is used. Since this method doesn't eliminate errors but rather redistributes them, it converts deterministic effects into random effects. From the introduction in Chapter 5, any errors with random characteristics will result in the increase of noise floor. However since many applications have a tight distortion tone (dynamic range) requirement but relatively relaxed requirement on noise floor, this method can be adopted. Compared with methods introduced in the literature with high hardware complexity, this method will dramatically lower the cost.

7.6 Summary

A new method for reducing gain errors and offsets in time-interleaved *ADC*' is introduced. The method is based on randomly choosing the *ADC*' channels inside the time-interleaved structure. Hardware implementation for channel randomization is also proposed. This method is simulated based on an 8-channel, 8-bit time-interleaved *ADC*' structure. From the simulation, it shows that the harmonic distortion caused by the amplifiers gain errors and the offsets of comparators and references is reduced by more than 10 *dB*s. This improvement can be achieved using a relatively low hardware cost.

8 SUMMARY AND CONCLUSIONS

8.1 Summary

The first part of this dissertation is the development of an equalization system for disk drive channel. The research on this topic is concentrated on the implementation of a high speed equalizer based on a new mixed-signal multiplier structure. By choosing this mixed-signal solution, this design avoids the need of high speed ADC in disk drive read channel signal processing as well as large digital blocks. Compared with pure analog solution, this design has the flexibility of adjusting coefficients without compromising the performance. Results of the FIR filter test chip show the functionality of the proposed mixed-signal multiplier at very high speed. In particular, the prototype chip is a low power and high speed FIR filter that has achieved good THD performance. The chip area is also compact thanks to the mixed-signal current mode implementation. By comparing with other published discrete-time FIR designs for disk drive read channel application, this design provides by far the best power-per-speed performance.

The second part of this dissertation presents solutions to reduce the non-ideal parameter effects in Time-interleaved ADC's. Despite many outstanding features of the time-interleaved structure, parameter mismatches between different conversion channels have hindered its applications. A randomization algorithm is proposed to reduce the effects come from device parameter mismatches. For the more important and harder to solve error come from the timing mismatch, this dissertation proposed a digital interpolation method that can automatically measure the timing error and continuously do the calibration. In most cases, this algorithm can eliminate the fixed distortion tones ap-

pearing on the output signal spectrum due to timing mismatch. Furthermore, since this method is proposed to be a background operation without affecting the continuous operation of the analog-to-digital conversion as well as a calibration done in digital domain, this algorithm provides an effective way to solve the timing-error effects. Though no test chip is implemented for the above two methods, the effectiveness of these methods has been demonstrated through theoretical analysis and simulation results. Especially in the timing error calibration, the performance improvement is quite significant.

8.2 Conclusions

The following list is the conclusions or observations used in this dissertation. It is arranged according to the sequence of chapters.

Chapter 2:

- When disk storage becomes increasingly compact, the read out signal needs to be equalized before decoding so that written symbols can be detected correctly.
- $PR-IV$ signaling is used in this dissertation since it uses signal response with controlled ISI to boost the read out signal for easy detection.
- Sign-Sign LMS algorithm targeting $PR-IV$ signaling with three level target value is used in designing adaptive equalizer.

Chapter 3:

- A high level simulation environment is developed for disk drive read channel so that it can be used for performance optimization and providing design guidance.
- Effectively use of mixed-signal design can result in compact design and high speed of operation.
- A discrete time FIR system is used for equalization. It can provide fast operation and small silicon area when compares to a pure digital implementation.

- Mixed-signal multiplier is proposed as the core circuit in the equalizer design.
- Based on the proposed mixed signal multiplier technique, a programmable FIR filter and an adaptive equalizer were designed and implemented.
- Circuit simulation result complies with high level system simulation.

Chapter 4:

- Test setup for high speed circuits needs precaution. This includes well designed PCB and correctly chosen equipment.
- The FIR filter under test can reach 360 MHz operation speed while having 3rd harmonic less than 35 dBc.
- The FIR filter is fully programmable to form any kind of transfer functions. Good correspondence is achieved between theoretical and tested results in the LPF case.
- The adaptation blocks are functional.

Chapter 5:

- ADC can be classified according to the data path it used.
- Time-interleaved ADC is an important research topic.
- Mismatches in time-interleaved ADC channels must be compensated.

Chapter 6:

- Timing error effect is one of the dominant factors influencing the performance of time-interleaved ADC, especially the clock skews between different channels.
- Methods directly aiming at reducing timing error effects are seldom addressed.
- A digital background timing error measurement technique can be used to estimate clock skews for different channels. All processes can be built in digital domain and within the main ADC circuit. No major change in analog circuit design is required.

- Interpolation method can improve ADC performance by correcting the clock skew effect with the help from clock skew measurement. This method is also shown to be linear without generating inter modulation components on the spectrum.
- The effectiveness of interpolation method has been shown analytically. $MATLAB$ simulations agree with the analysis results.
- The improvement on $SFDR$ is significant which is in the range of 20 to 40 dB.
- The more points used in the interpolation algorithm, the better improvement on $SFDR$.
- The improvement of $SFDR$ is influenced by the sampling ratio. However when the points used in interpolation is large enough, such as 32 points, input signals that are very close to $Nyquist$ rate can be corrected to have nearly ideal performance.
- The improvement of $SFDR$ is influenced by finite digital word length.
- The proposed interpolation algorithm can be implemented as in FIR structure.

Chapter 7:

- Channel Randomization can reduce device mismatch effects on $SFDR$ so that the dynamic range is increased.
- Using of $Beně$ network reduces hardware required in doing channel Randomization.

8.3 Contributions

As a final summary, the contributions of this dissertation work are as follow:

1. Exploring a new architecture in mixed-signal design.
2. Searching a solution for a well-known structure from a new point of view.

8.4 Future Work

Future research will be extended from the conclusions in the two major parts in this dissertation. Since the mixed-signal multiplier is proven to be functional, structures based on it can be proposed to implement more complicated filter systems. On the other hand, the FIR based equalizer can be developed for the application of higher order partial response system. Still another angle is put on the improvement of adaptation method, which will utilize the multiplier for faster operation and quicker convergence.

Research on improving time-interleaved ADC' will be concentrated on improving the efficiency of calibration method. Silicon implementation is also one of the concentrations with further analysis of the ADC'.

APPENDIX A ANALYSIS OF TIMING ERROR EFFECTS

In this appendix, an analysis is carried out to describe the timing error effects, which are illustrated in Figure A.1. For a given input signal $x(t)$, case (a) represents the ideal sampled values of the input signal $x(t)$ given as

... $x(0)$, $x(T_s)$, $x(2T_s)$ ... $x(MT_s)$, $x(MT_s + T_s)$, ...

Case (b) represents the non-uniform sampled values given as

... $x(0 + \Delta t_0)$, $x(T_s + \Delta t_1)$, $x(2T_s + \Delta t_2)$ ... $x(MT_s)$, $x(MT_s + T_s + \Delta t_1)$, ...

When the digital outputs are combined, all the sampled values are considered as evenly sampled. This corresponds to case (c) where the samples happen at the evenly spaced instants but with errored results. In above Δt_l represents the timing error of channel l due to both clock skew and random jitter. However, for the following derivation, Δt_l will only be used for representing the clock skew of channel l for simplicity.

There are two things that can be noticed from Figure A.1. First, it can be considered that the input signal is sampled *evenly* by each channel with a sampling period of MT_s although there is timing error on each channel when compared between channels. Second, channel 0 is used as a reference and the timing errors of other channels are compared to the sampling instant of channel 0. Hence, channel l can be viewed as having an input equal to $x(t + \Delta t_l)$ and sampling this input using a sampling clock with period of MT_s. The expression for the sampling clock of each channel, $s_l(t)$, can be expressed as

$$s_l(t) = \sum_{k=-\infty}^{\infty} \delta(t - kMT_s - lT_s) \qquad \text{(A.1)}$$

Based on the above expression, the sampled output of channel l, x_{sl}, is given as:

$$x_{sl} = x_l(t) \cdot s_l(t) \tag{A.2}$$

The final ADC output $y(t)$, can be written as:

$$y(t) = \sum_{l=0}^{M-1} x_{sl}(t) + q \tag{A.3}$$

where q is the quantization error. $y(t)$ can be further expressed as

$$y(t) = x(t) \cdot \sum_{k=-\infty}^{\infty} \delta(t - kT_s) + \sum_{l=0}^{M-1} \sum_{k=-\infty}^{\infty} \delta(t - kMT_s - lT_s) \cdot (x_l(t) - x(t)) + q \tag{A.4}$$

From the above equation, it can be observed that the first term is the same as the output from an ideal ADC, and the second term is the error term due to the timing error in each channel. Assume that the input signal $x(t)$ has a frequency spectrum of $X(\omega)$. Then the frequency spectrum of the sampled input for channel l, $X_{sl}(\omega)$, can be determined using the fact that

$$X_{sl}(\omega) = \frac{1}{2\pi} X_l(\omega) * S_l(\omega) \tag{A.5}$$

where $X_l(\omega)$ is the Fourier transform of $x_l(t)$ and is equal to $X(\omega)e^{-j\omega\Delta t_l}$. Furthermore, with

$$S_l(\omega) = \frac{2\pi}{MT_s} \sum_{k=-\infty}^{\infty} \delta(\omega - k\frac{\omega_s}{M}) e^{-jkl2\pi/M} \tag{A.6}$$

the frequency spectrum of the ADC output, $Y_s(\omega)$, can be derived as

$$Y_s(\omega) = \frac{1}{MT_s} \sum_{l=0}^{M-1} \sum_{k=-\infty}^{\infty} X(\omega - k\frac{\omega_s}{M}) e^{-j(\omega - k\frac{\omega_s}{M})\Delta t_l} e^{-jkl2\pi/M} + noise \tag{A.7}$$

where the term *noise* represents the frequency spectrum of the quantization noise and other errors due to random jitter.

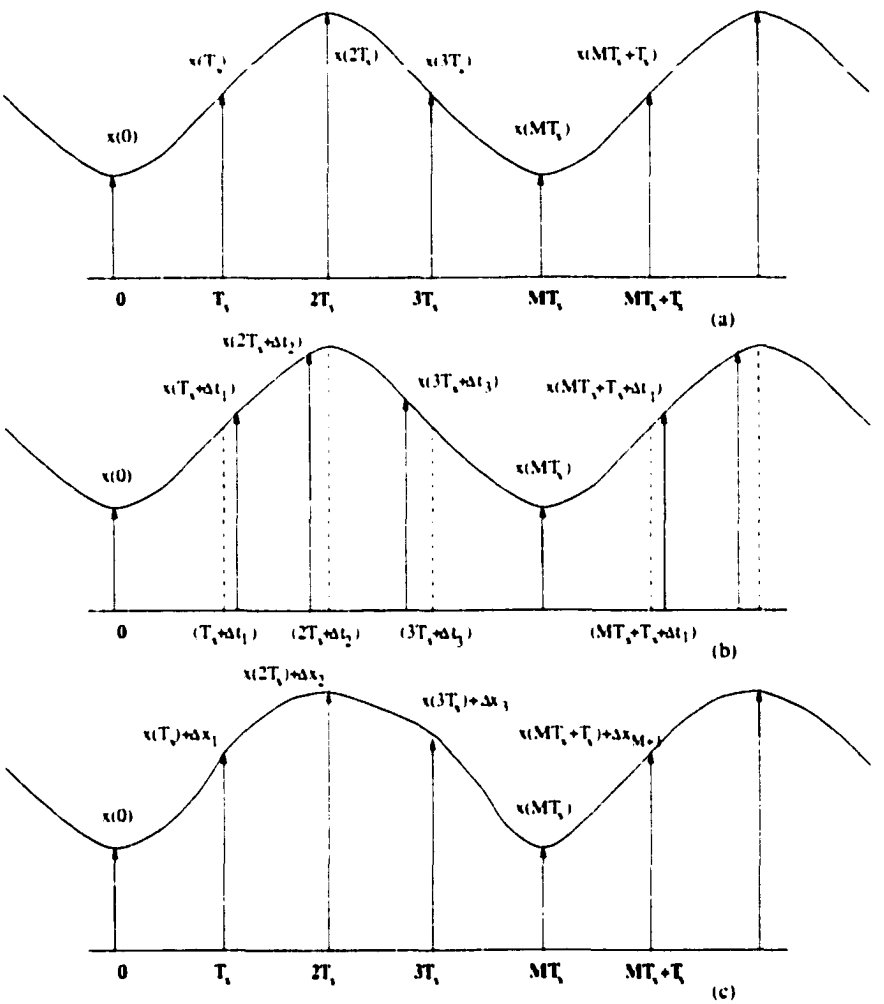

Figure A.1 Effects of Timing Error (a) Ideal Sampling (b) Non-uniform Sampling (c) Equivalent to uniform Sampling

BIBLIOGRAPHY

[1] W. C. Black, and D. A. Hodges, "Time Interleaved Converter Array," *IEEE Journal of Solid-State Circuits*, Vol. SC-15, pp. 1022-1029, 1980.

[2] Patrick Pai, "Equalization and Clock Recovery in Magnetic Storage Read Channels," *Final Report*, UCLA, 1996.

[3] P. Siegel, "Recording Codes for Digital Magnetic Storage," *IEEE Trans. Magnet.*, vol. MAG-21, no. 5, pp. 1344-1349, Sept, 1985.

[4] John Cioffi, William Abbott, Hemant Thapar, C. Michael Melas and Kevin Fisher, "Adaptive Equalization in Magnetic-Disk Storage Channels," *IEEE Communications Magazine*, pp. 14-29, Feb, 1990.

[5] Roy Cideciyan, Francois Dolivo, Reto Hermann, Walter Hirt and Wolfgan Schott, "A PRML System for Digital Magnetic Recording," *IEEE Journal on Selected Areas in Communications*, Vol. 10 No. 1, pp 38-56, January 1992.

[6] Rick A. Philpott, Robert A. Kertis, Ray A. Richetta etc. "A 7Mbyte/s (65 MHz), Mixed-Signal, Magnetic Recording Channel DSP Using Partial Response Signaling with Maximum Likelihood Detection," *IEEE Journal of Solid State Circuits*, vol. 29, no. 3, pp. 177-183, March 1994.

[7] Peter Kabal and Subbarayan Pasupathy, "Partial-Resopose Signaling," *IEEE Transactions on Communications*, Vol. Com-23, No. 9, Sept, 1975.

[8] Sapthotharan K. Nair, Hamid Shafiee and Jackyun Moon, "Equalization and Detection in Storage Channels," *IEEE Transactions on Magnetics*, Vol. 32, No. 5, pp. 5206-5216, Sept 1996.

[9] James E. C. Brown, Paul J. Hurst and Lawrence Der, "A 35 Mb/s Mixed-Signal Decision-Feedback Equalizer for Disk Drive in 2-μm CMOS," *IEEE Journal of Solid-State Circuits*, vol. 31, no. 9, pp. 1258-1266, Sept, 1996.

[10] Bret C. Rothenberg, James E. C. Brown, Paul J. Hurst and Stephen H. Lewis, "A Mixed-Signal RAM Decision-Feedback Equalizer for Disk Drives," *IEEE Journal of Solid-State Circuits*, Vol. SC-32, pp. 713-720, May, 1997.

[11] Ravinder S. Kajley, Paul J. Hurst, and James E. C. Brown, "A Mixed-Signal Decision-Feedback Equalizer That Uses a Look-Ahead Architecture.," *IEEE Journal of Solid-State Circuits*, Vol. SC-32, pp. 450-459, March, 1997.

[12] Nicholas P. Sands, Max W. Hauser, Guojin Liang, Gerrit Groenewold, Steven Lam, Chao-Ho Lin, John Kuklewicz, Luke Lang, and Raman Dakshinamurthy, "A 200Mb/s Analog DFE Read Channel," *IEEE ISSCC, 96*, TP 4.6, pp. 72-73, 1996.

[13] J. Moon and L. R. Carley, "Performance comparison of detection methods in magnetic recording," *IEEE Trans. Magnetics*, vol. 26, no. 6, pp. 3155-3172, Nov. 1990.

[14] James E. C. Brown, Paul J. Hurst, Bret C. Rothenberg and Stephen H. Lewis, "A CMOS Adaptive Continuous-Time Forward Equalizer, LPF, and RAM-DFE for Magnetic Recording," *IEEE Journal of Solid-State Circuits*, Vol. 34, No. 2, pp. 162-169, Feb, 1999.

[15] K. A. Kozma, D. A. Johns, and A. S. Sedra, "Automatic tuning of continuous-time integrated filters using an adaptive filter technique," *IEEE Transactions on Circuits and Systems*, Vol. 38, pp. 1241-1248, 1991.

[16] D. A. Johns, W. M. Snelgrove, and A. S. Sedra, "Continuous-time LMS adaptive recursive filters," *IEEE transactions on Circuits and Systems*, vol 38, pp. 769-778, 1991.

[17] A. Bishop, I. Chan, S. Aronson, etc., "A 300 Mb/s BiCMOS Disk Drive Channel with Adaptive Analog Equalizer," *IEEE ISSCC 99*, MP 2.7, pp. 46-47, 1999.

[18] T. W. Matthews and R. R. Spencer, "An Integrated Analog CMOS Viterbi Detector for Digital Magnetic Recording," *IEEE transactions on Circuits and Systems*, vol 28, pp. 1294-1302, Dec. 1993.

[19] Gregory T. Uehara and Paul R. Gray, " A 100 MHz A/D Interface for PRML Magnetic Disk Read Channels," *IEEE Journal of Solid State Circuits*, vol. 29, no. 12, pp. 1606-1613, December 1994.

[20] Sanroku Tsukamoto, Lan Dedic, Toshiaki Endo, Kazuyoshi Kikuta, Kunihiko Goto, and Osamu Kobayashi, "A CMOS 6b 200M Sample/s 3V-Supply A/D Converter for a PRML Read Channel LSI," *IEEE ISSCC 96*, TP 4.5, pp. 70-71, 1996.

[21] Sanroku Tsukamoto, Toshiaki Endo and William G. Schofield, "A CMOS 6b 400MSamples/s ADC with Error Correction," *IEEE ISSCC*, FA 9.8, pp.152-153, February 1998.

[22] Yuko Tamba and Kazuo Yamakido, " A CMOS 6b 500MSamples/s ADC for a Hard Disk Drive Read Channel," *IEEE ISSCC*, WA 18.5, pp. 324-325, February, 1998.

[23] Kwangho Yoon, Sungkyung Park and Wonchan Kim, "A 6b 500MSamples/s CMOS Flash ADC with a Background Interpolated Auto-Zeroing Technique," *IEEE ISSCC*, WA 18.6, pp. 326-327, February, 1998.

[24] Iuri Mehr and Declan Dalton, " A 500-MSample/s 6-Bit Nyquist-Rate ADC for Disk-Drive Read-Channel Applications," *IEEE Journal of Solid State Circuits*, vol. 34, No. 7, pp. 912-920, July 1999.

[25] Kouji Sushihara, Hiroshi Kimura, Youichi Okamoto, Kazuko Nishimura and Akira Matsuzawa "A 6b 800 MSamples/s CMOS A/D Converter," *IEEE ISSCC*, WP 26.2, pp. 428-429, February 2000.

[26] k. Nagaraj, D. A. Martin, M. Wolfe, R. Chattopadhyay, S. Pavan, J. Cancio, and T. R. Viswanathan, "A 700 MSamples/s 6b Read Channel A/D Converter with 7b Servo Mode," *IEEE ISSCC*, WP 26.2, pp. 426-427, February 2000.

[27] Kaveh Parsi, Robert P. Burns, Alan Chaiken, etc., "A PRML Read/Write Channel IC Using Analog Signal Processing for 200 Mb/s HDD," *IEEE Journal of Solid-State Circuits*, Vol. 31, No. 11, pp. 1817-1830, November 1996.

[28] David A. Johns and Ken Martin, "Analog Integrated Circuit Design," John Wiley and Sons, Inc. New York, 1997.

[29] Stephen Lyle, Glen Worstell and Richard Spencer, "An Analog Discrete-Time Transversal Filter in 2.0 μm CMOS," *Proc. 26th Annual Asilomar Conf. Signals, Systems, and Computers*, pp. 970-973, 1992.

[30] Danfeng Xu, Yonghua Song and Gregory T. Uehara, "A 200MHz 9-Tap Analog Equalizer for Magnetic Disk Read Channels in 0.6μm CMOS," *IEEE ISSCC, 96*, TP 4.7, pp 74-75, 1996.

[31] Sami Kiriaki, T. Lakshmi Viswanathan, Gennady Feygin, Bogdan Staszewski, Rick Pierson, Bill Krenik, Mickey De Wit, and Krishnaswamy Nagaraj, "A 160MHz Analog Equalizer for Magnetic Disk Read Channels," *ISSCC 97*, SA 19.5, pp. 322-323, 1997.

[32] Xiaodong Wang and Richard R Spencer, "A Low-Power 170-MHz Discrete-Time Analog FIR Filter," *IEEE Journal of Solid-State Circuits*, Vol. 33, No. 3, March, 1998.

[33] Ban-Sup Song, and David C. Soo, "NRZ Timing Recovery Technique for Band-Limted Channels," *IEEE Journal of Solid-State Circuits*, Vol. 32, No. 4, pp. 514-520, 1997.

[34] Ramon Gomez, Maryam Rofougaran and A. Abidi, "A Discrete-Time Analog Signal Processor for Disk Read Channels," *IEEE ISSCC*, FA 13.1, pp. 212-214, Feb., 1993.

[35] Bret C. Rothenberg, Stephen H. Lewis and Paul J. Hurst, "A 20-Msample/s Switched -Capacitor Finite-Impulse Response Filter Using a Transposed Structure," *IEEE Journal of Solid State Circuits*, vol. 30, no. 12, pp. 1350-1356, Dec. 1995.

[36] Kevin D. Fisher and William L. Abbott, "PRML Detection Boosts Hard-Disk Drive Capacity," *IEEE Spectrum*, pp. 70-76, Nov., 1996.

[37] Rudy van de Plassche, *Integrated Aanlog-to-Digital and Digital-to-Analog Converters*, Kluwer Academic Publishers, Dordrecht, Netherlands, 1994

[38] Yuan Jiren, Ingemar Karlsson and Christer Svensson, "A True Single-Phase-Clock Dynamic CMOS Circuit Technique," *IEEE Juournal of Solid State Circuits*, Vol. SC-22, No. 5, October, pp 899- 901, 1987.

[39] *PCB* Design Consideration, *Analog Device Inc.* Boston, MA, 1999.

[40] Yee Ling Cheung and Aaron Buchwald, "A Sampled-Data Switched-Current Analog 16-Tap FIR Filter with Digitally Programmable Coefficients in 0.8μm CMOS," *IEEE ISSCC*, TP 3.3, pp.54-55, February 1997.

[41] Dale J. Pearson etc. "250 MHz Digital FIR Filters for PRML Disk Read Channels," *IEEE ISSCC*, WP 5.2, pp.80-81, February 1995.

[42] K. Poulton, J. J. Corcoran and T. Hornak, "A 1-GHz 6-bit *ADC* System," *IEEE Journal of Solid-State Circuits*, Vol. 22, No. 6, pp 962-970, Dec. 1987.

[43] A. Matsuzawa et al., "A 6b 1GHz Dual-parallel A/D Converter," *ISSCC Digest Technical Paper*, pp. 174-175, 311, Feb. 1991.

[44] K. Rush and P. Byrne, "A 4GHz Data Acquisition Syatem," *ISSCC Digest Technical Paper*, pp. 176-177, 312, Feb. 1991.

[45] H. Kimura, A. Matsuzawa, T. Nakamura and S.Sawada, "A 10-b 300-MHz Interpolated-parallel A/D Converter," *IEEE Journal of Solid-State Circuits*, vol. 28, no. 4, pp. 438-446, April 1993.

[46] Y. M. Lin, B. Kim and P. R. Gray, " A 13-b 2.5 MHz Self-calibrated Pipelined A/D Converter in 3-μm CMOS," *IEEE Journal of Solid-State Circuits*, Vol. 26, no. 4, pp. 628-636, April, 1991.

[47] S U Kwak, B S Song and K. Bacrania, "A 15-b, 5-Msample/s Low-Spurious CMOS *ADC*" *IEEE Journal of Solid-State Circuits*, Vol. 32, No. 12, pp. 1866-1875, December, 1997.

[48] Daihong Fu, K. C. Dyer, S. H. Lewis and Paul J. Hurst "A Digital Background Calibration Technique for Time-Interleaved Analog-to-Digital Converters," *IEEE Journal of Solid-State Circuits*, Vol. SC-33, pp. 1904-1911, Dec, 1998.

[49] S. H. Lewis and P. R. Gray, "A Pipelined 5-Msamples/s 9-bit Analog-to- digital converter," *IEEE Journal of Solid-State Circuits*, vol. SC-22, no. 6, pp. 954-961, December 1987.

[50] S. H. Lewis, "Optimizing the Stage Resolution in Pipelined, Multistage, Analog-to-digital Converters for Video-rate Applications," *IEEE Trans. Circuits Syst. II, Analog Digital Sig. Proc.*, Vol. 39, no. 8, pp. 516-523, August, 1992.

[51] Cormac S. G. Conroy, "A High-Speed Parallel Pipeline A/D Converter Technique in CMOS," PhD Dissertation, University of California at Berkeley, 1994.

[52] AD9042 Data sheet, *Analog Device Inc.*, Boston, MA, 1998.

[53] ADS807 Data sheet, *Burr-Brown Co.*, Tuson, AZ, 1998.

[54] Kwang Young Kim, Naoya Kusayanagi and A. A. Abidi, "A 10-b, 100-MS/s CMOS A/D Converter," *IEEE Journal of Solid-State Circuits*, Vol. 32, No. 3, pp. 302-311, March, 1997.

[55] C. S. G. Conroy, D. W. Cline and P. R. Gray, "An 8-b 85-MS/s Parallel Pipeline A/D Converter in 1-μm CMOS," *IEEE Journal of Solid-State Circuits*, Vol. 28, No. 4, pp. 447-454, April, 1993.

[56] Bengt Jonsson and Hannu Tehunen, "A Dual 3-V 32-MS/s CMOS Switched-Current *ADC* for Telecommunication Applications," *IEEE International Symposium on Circuits and Systems*, June, 1999.

[57] Yih-Chyun Jenq, "Digtial Spectra of Nonuniformly Sampled Signals: Fundamentals and High-Speed Waveform Digitizers," *IEEE Trans on Instrument and Measurement*, Vol. 37, No. 2, pp. 245-251, June 1988.

[58] Yih-Chyun Jenq, "Perfect Reconstruction of Digital Spectrum from Nonuniformly Sampled Signals," *IEEE Trans on Instrument and Measurement*, Vol. 46, No. 3, pp. 649-651, June, 1997.

[59] Richard L. Burden, etc., *Numerical Analysis*, Prindle, Weber & Schmit, pp. 88-92, 1981.

[60] Yih-chyun Jenq, "Digital Spectra of Nonuniformly Sampled Signals: A Robust Sampling Time Offset Estimation Algorithm for Ultra High- Speed Waveform Digitizers Using Interleaving," *IEEE Trans on Instrument and Measurement*, Vol. 39, No. 1, pp. 71-75, Feb. 1990.

[61] E. G. Soenen, and R. L. Geiger, "An Architecture and An Algorithm for Fully Digital Correction of Monolithic Pipelined ADCs," *IEEE Trans. on Circuits and Systems-II*, Vol. 42, pp. 143-153, 1995.

[62] W. C. Black, PhD Dissertation, University of California at Berkeley, 1980.

[63] F. Thomson Leighton. *An Introduction to Parallel Algorithms and Architectures.* San Mateo, CA: Morgan Kaufamann, 1992.

ACKNOWLEDGMENTS

I would like to express my sincere appreciation to my research advisor Professor Edward KF Lee. I have been honored to have worked under his supervision. For the past several years of my stay at Iowa State University, Prof. Lee provided many profound thoughts and provoked invaluable guidance for my research. I am also deeply indebted to him for providing me a good research environment.

It would not have been for me to achieve the final research goal without the help from my wife, Lin Wu. Thanks for the discussion on my projects and all the cooking during the busy nights when sending out the test chip.

I would like to express my appreciation to Dr. Edward Lee, Dr. Randall Geiger, Dr. Bill Black, Dr. Marwan Hassoun and Dr. Johnny Wong for kindly agreeing to be my committee member.

I also would like to thank Ms Baiying Yu for helping on thesis submission matters, Ms Maria Blanco for all the various kinds of help from ordering components to organization of birthday lunch and Mr. Scott Service for providing computer supporting, and the Roy J. Carver Charitable Trust, Grant #98-229 for providing Carver Lab for testing.