A Hybrid Digital FIR Lattice Filter for PRML Magnetic Read Channel

by

Lily Wong Ok Yuen

Submitted to the Department of Electrical Engineering and Computer Science

in Partial Fulfillment of the Requirements for the Degrees of

Bachelor of Science in Electrical Science and Engineering

and Master of Engineering in Electrical Engineering and Computer Science

at the Massachusetts Institute of Technology

May 1998

[June 1998]

Author _____

Department of Electrical Engineering and Computer Science
May 21, 1998

Certified by _____

Anantha P. Chandrakasan
Associate Professor of Electrical Engineering
Thesis Supervisor

Accepted by _____

Arthur C. Smith
Chairman, Department Committee on Graduate Theses

A Hybrid Digital FIR Lattice Filter for PRML Magnetic Read Channel

by

Lily Wong Ok Yuen

Submitted to the
Department of Electrical Engineering and Computer Science

May 21, 1998

In Partial Fulfillment of the Requirements for the Degrees of
Bachelor of Science in Electrical Science and Engineering
and Master of Engineering in Electrical Engineering and Computer Science

ABSTRACT

The benefits and costs of using the lattice properties in the design of a finite-impulse response (FIR) filter are investigated. Using conventional design methods, the transversal structures are not very efficient in higher order digital FIR filter implementation due to the complexity of the multipliers. This design presented reduced hardware over the conventional transversal FIR filter by taking advantage of the numerical robustness of the lattice structure.

In this thesis, techniques were developed for efficiently synthesizing digital lattice filters using the stable transversal filter for comparison purposes. A preliminary system study was done to find the equalization tradeoffs between the continuous time filter (CTF) and the FIR. We performed MATLAB behavioral and structural simulations to evaluate the lattice structure and optimized the filter parameters. We used QuickSim VHDL and Synopsys compiler to implement the architecture for system specification, design refinement, and gate level simulation. The design used TSC4000 library supporting worst case 33C12 process for a 0.6μm metal pitch and 3.3V supply running at a speed 54 MHz with a power dissipation of 82mW. The chip area is 2289.25 equivalent NAND gates or an actual silicon area of 0.212mm^2. We used Aquarius auto place&route tool for layout.

Thesis Supervisor: Anantha P. Chandrakasan
Title: Associate Professor of Electrical Engineering

This thesis is dedicated to my parents,

my father, Joseph Ah-Sine Wong Ok Yuen

and

my mother, Marie Chantal Wong Ok Yuen

Acknowledgments

Goals are dreams with deadlines
~ Lily Wong ~

This thesis would not have been possible without the support and help from many people. My sincere gratitude to Sami Kiriaki for recommendation of the topic as well as his supervision and the valuable guidance he has provided me during the course of my stay at Texas Instruments Inc., both project-related and otherwise.

Special thanks to Bogdan Staszewski for playing a crucial role in the successful completion of this project. His mentorship on engineering design and guidance on practical engineering issues have been inspiring, to say the least. He has been a great source of moral support and a much-appreciated sounding board. In addition, I would like to thank all the members of the Read Channel design team for a great working environment, both technically and socially. Thank you all for making my internship an educational and enjoyable experience. I look forward to joining you after graduation.

I would like to thank Texas Instruments Inc. for funding this project, and Massachusetts Institute of Technology for the invaluable engineering internship opportunity.

I would also like to thank my thesis supervisor, Anantha Chandrakasan, for his guidance and input.

To all of my friends who have tolerated much of my stressful presence over the past few months and to those of you whom I have neglected, thank you for bearing with me.

Last but not the least, I send heartfelt thanks across the oceans to my parents, sisters and my grandmother. To my uncle Yves and my aunt Marie-Ange, who spoilt and pampered me over holidays and school breaks, thank you for being there when I needed advice and a reassuring word. Thank you all for believing in me. I could not have done it without your love, support and prayers.

Contents

List of Figures

List of Tables

List of Abbreviations

ADC	Analog-to-Digital Converter
AGC	Automatic Gain Control
AWGN	Additive White Gaussian Noise
BER	Bit Error Rate
cbd	Channel bit density
CTF	Continuous Time Filter
ECC	Error Correction Circuits
EPR4	Enhanced Partial Response Class 4
FIR	Finite Impulse Response
HDD	Hard Disk Drive
IIR	Infinite Impulse Response
ISI	Inter-Symbol Interference
LMS	Least-Mean-Square
LPF	Lowpass Filter
MAC	Multiplier-and-accumulator
ML	Maximum Likelihood
MSE	Mean Square Error
NRZI	Non-Return to-Zero and-Invert
PLL	Phase Locked Loop
PR	Partial Response
PRML	Partial Response Maximum Likelihood
PR4	Partial Response Class 4
RLL	Run-Length-Limited

RMSE	Root Mean Square Error
SGF	Signal Flow Graph
VA	Viterbi Algorithm
VGA	Variable Gain Amplifier
VSS	VHDL System Simulator
ZF	Zero-Forcing

Chapter 1

Introduction and Motivation

Increased use of disk drives throughout the computer industry is driving the industry's expansion, with sales boosts in all major computer applications, from desktop personal computers and notebook computers to network file servers and mainframe systems. According to the 1997 DISK/TREND Report on rigid disk drives, the 1997 sales revenues for the industry are expected to reach $34 billion. Rapid market expansion and new products will take the industry's sales in 2000 to over $75 billion [1]. Not only have revenues increased dramatically, but so have the performance figures. The increasing demand for higher storage capacity and faster information access across the magnetic media continues to be driven by the proliferation of storage-hungry multimedia applications and other feature-rich software. The disk drive industry continues to maintain a rapid growth rate that is pushing the design requirements for magnetic disk drives to unprecedented levels.

Over the past 10 years or so, the bit-capacity of mass-market hard-disk drives has increased 100-fold - the reward of improvements in read/write heads, magnetic materials, head-positioning systems, and much else (Figure 1-1). But as advances in those well-worn areas grow more costly and difficult, attention is turning to the read-channel electronics as a way to increase disk capacity and throughput rates in the future.

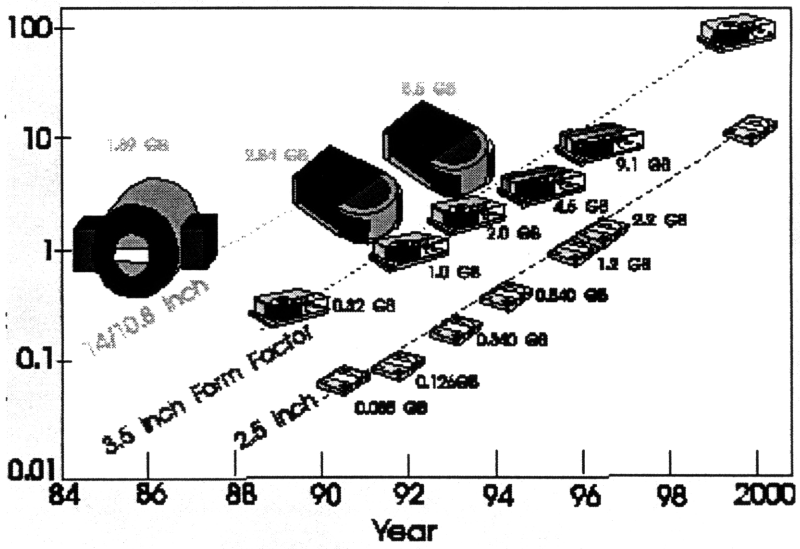


Figure 1-1: IBM - Disk drive form factor evolution [2]

The fastest growing segment of mass storage business has been the read channel electronics that features the Partial Response Maximum Likelihood (PRML) technology.

In 1996, 1 to 2 gigabyte drives led the industry in overall shipments, but by 2000 a typical drive will have a capacity of 10 to 20 GB, a transfer rate of over 400Mb/s, and will use more sophisticated signal processing and data recovery techniques as we move towards an integrated DSP chip. This imposes some very demanding requirements on the circuits in the disk drive, and meeting some of these specifications such as area and power is one of the goals of this research. A digital Finite Impulse Response transversal filter with Least Mean Square (LMS) adaptation is currently used in a high performance CMOS read channel IC that provides digital functions needed to implement Partial Response Class 4 (PR4) or Enhanced Partial Response Class 4 (EPR4) read channel for zoned recording hard disk drive systems with data rates from 80-250 Mbits/s.

The objective of this thesis is to investigate how well we can perform equalization of the incoming read signal to the PR4 and EPR4 waveform by a digital lattice filter, as well as take advantage of the lattice properties to reduce area and power. The lattice digital filter is widely used in various fields of signal processing. Its practical advantages,

such as the modular structure suitable for VLSI implementation, the superiority in the finite word length performance, the flexibility in choosing the order, and so forth are now well known [3-6]. Also, theoretical aspects of lattice filters have been extensively exploited in the literature in applications other than magnetic recording [7,8]. Only one single lattice structure, due to Itakura and Saito [9] is available for the implementation of all-zero filters. The lattice structure has been useful in speech analysis applications and promises to be useful in other applications as well.

Chapter 2

Magnetic Recording

Magnetic disk storage devices are an impressively intricate collection of mechanical and electronic subsystems, which operate at the limits of modern engineered technology. The design an manufacture of these Hard Disk Drives (HDD) requires expertise in electrical and magnetic fields, materials, aerodynamics, tribology, mechanics, electromechanics, digital communications, coding, signal processing, and circuit design. The design and fabrication of these mixed signal chips are an arduous task, involving teams of design and manufacturing engineers. As an application of integrated systems, magnetic recording presents some of the fastest applications of commercial baseband signal processing to date.

2.1 System Overview

The modern disk drive consists of three major circuit blocks namely the read, write and servo blocks as shown in Figure 2-1. The write block takes digital data from a computer bus, encodes this data into a (1,7) or (0,k) Run-Length-Limited (RLL) code, and writes this encoded data onto the disk media. The servo block keeps track of the location of the

data on the media, controls the position of the drive heads, and adjusts the motor speed. The servo mechanism must also be able to handle wide temperature fluctuations and severe vibrations especially in disk units installed in portable systems.

The read channel is the third significant block in the disk drive, which is the block that the Finite Impulse Response (FIR) filter is included in. The channel is responsible for taking a $1mV_{pp}$ analog signal and converting it into a digital signal with a raw bit error rate (BER) of about 10^{-6}. Error correction circuits (ECC) are then used to reduce the user data BER to about 10^{-12}. Typical blocks within the read channel are Automatic Gain Control (AGC), equalization, clock recovery, and data detection.

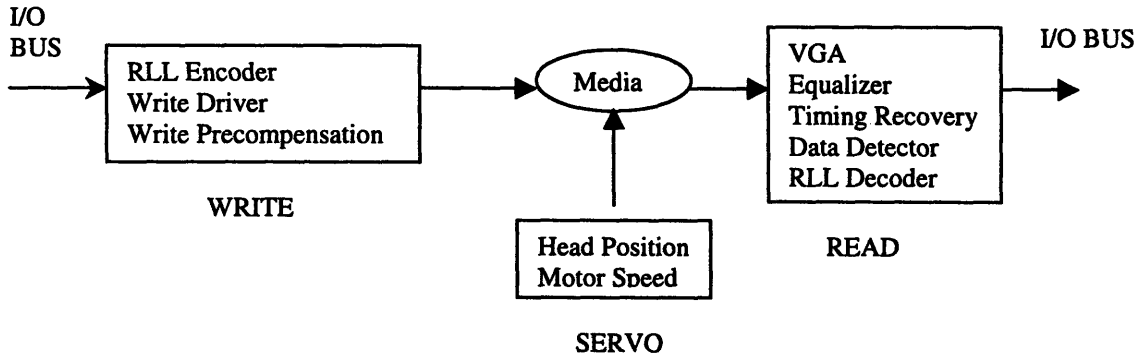


Figure 2-1: Disk drive system blocks

The storage and retrieval of information bits in magnetic recording channel resembles data transmission through a digital communication channel as shown in Figure 2-2. User 1 is a piece of software that writes digital data to the disk drive. The transmitter is essentially the write channel circuitry and the write head, the channel is the magnetic media, the receiver is the read head and the read channel circuitry. User 2 is another program or operating system. In both systems, the goal is to reliably pass as much information as possible, and the data is encoded at the input and decoded at the output.

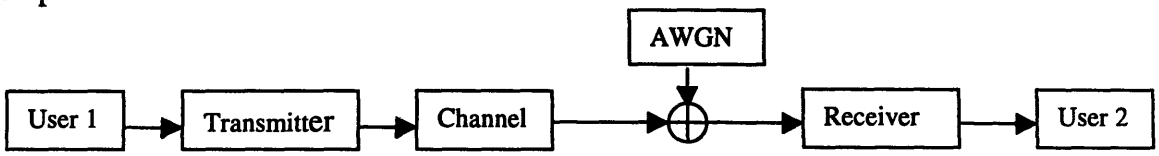


Figure 2-2: Model for a synchronous data communications system

2.2 The Disk Drive

Data on a disk is stored in circular tracks (Figure 2-3). The disks range in diameter from 35.6cm (14 inches) down to 3.3cm (1.3 inches), rotating speeds from 10000 to 300 rpm. The recorded information of the disk surface is divided into sectors. Servo information is recorded in the radially continuous narrow wedges between sectors. The servo sections give track number, sector number and tracking information.

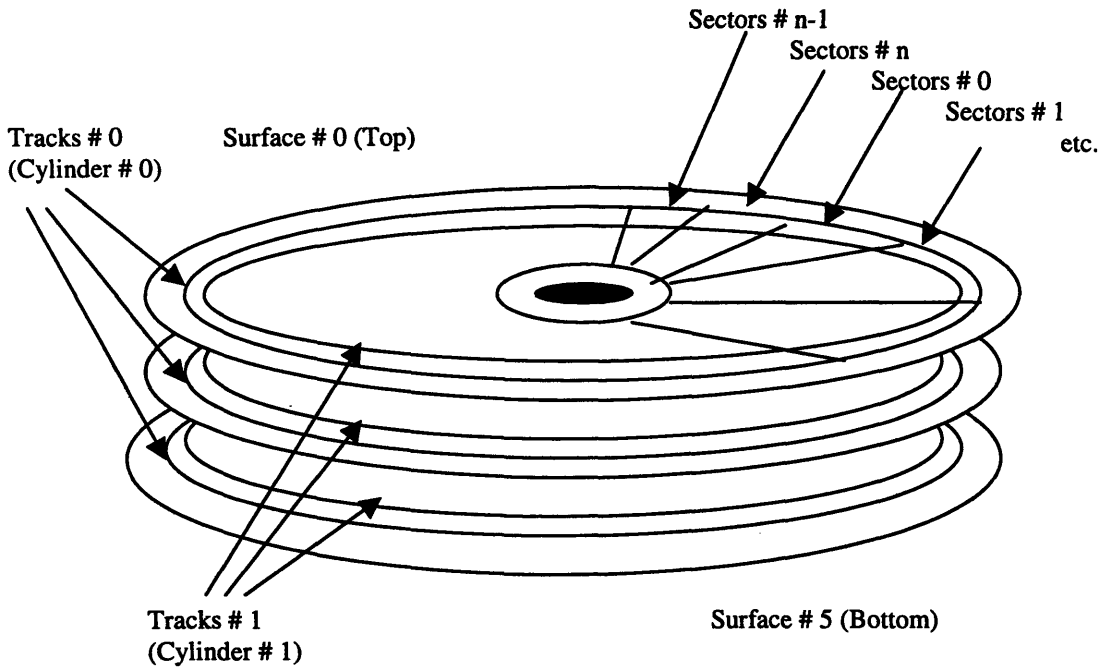


Figure 2-3: Data are recorded in sectors along concentric tracks.
All tracks with the same number make a "cylinder" [9]

Today's Read/Write Engineer views the channel as starting with a transition and adopts the Communications Theory viewpoint, which came from sending pulses down copper wire. This viewpoint has been applied to synchronous channels in drives. For Non-Return to-Zero and-Invert (NRZI) recording, whenever there is a change in direction, this generates a negative or positive pulse when the read head passes over the magnetic media and is considered a "1". An absence of flux is considered a "0". For

instance, a transition from all zeros to all ones is read back looking like a Lorentzian pulse (Figure 2-4).

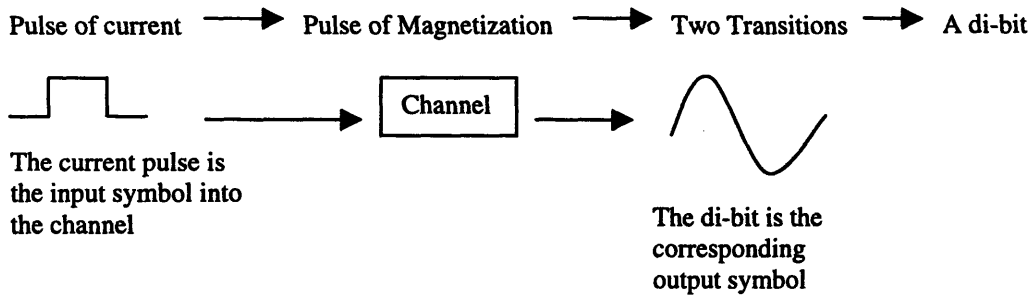


Figure 2-4: Response of the channel to a pulse of current input.

The pulse that is generated in the read head is described by:

$$L(t) = \frac{1}{1+\left(\frac{2t}{PW50}\right)^2} \tag{2.1}$$

The term PW_{50} refers to the width of a raw, unequalized pulse, in seconds or normalized to the clock period, T, at the 50% amplitude points. A typical range of PW_{50}/T for high density drives is 2 - 2.5. It is clear that if the pulse waveform is spread over a wide period of time and that if adjacent 1's or pulses are placed too close together, they will interfere with one another. This phenomenon is known as Inter-Symbol-Interference (ISI) and causes severe problems in both the clock recovery and data detection processes if not equalized out of the waveform. The unfamiliar reader is referred to [11-13] for more details on the fundamentals of magnetic recording.

2.3 PRML Detection

In general, a storage channel will have a high data density, while the transmission channel will have a high data rate. But the main difference between a storage channel

and a classic transmission channel is that, because of saturation recording, the magnetic channel can only process two levels (+1, -1), and the only way to increase the data rate is to decrease the spacing between transitions, whereas the transmission channel can use multilevel configurations to increase the transmitted data rate.

The response to two physically adjacent transitions is known as the pulse, or dibit response. The term "partial response" (PR) refers to the fact that the dibit response has more than one non-zero sample value. Only part of the response to a single bit falls within one time slot. One target shape is the extended partial-response class four, or EPR4, response. For such a response, a properly equalized isolated transition on the medium will produce a pulse with four non-zero samples. The sampled values of the equalized pulse will be 0, 1, 1, -1, -1, 0. Figure 2-5 shows the PR channel model. The (1-D) term in equation (2.2) results from the differentiating nature of the read process in that only changes in the direction of the magnetization result in a readback pulse. The frequency-domain plots of the Lorentzian pulses and targets are shown in Figure 2-6. The channel response has spectral nuls at d.c. and at one-half the sampling rate.

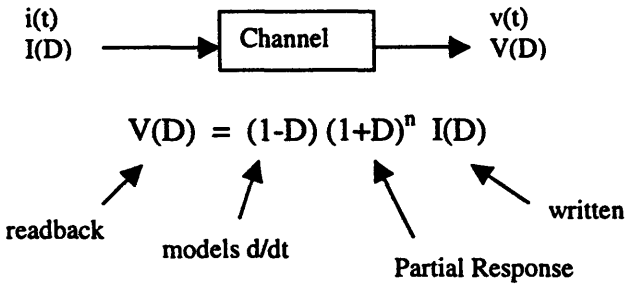


Figure 2-5: PR channel model

When the transitions are crowded close together, the neighboring pulses interfere with each other; a phenomenon commonly termed ISI. The dipulse response extends over several sample intervals. This interference coupled with the added complexity of channel noise, sets the upper limit in storage density as ISI and noise begin to cause number of errors in the detector. For most commercial drives, the tolerable detector error rate ranges from about 10^{-10} to 10^{-6} raw errors/bit, depending on the particular type of

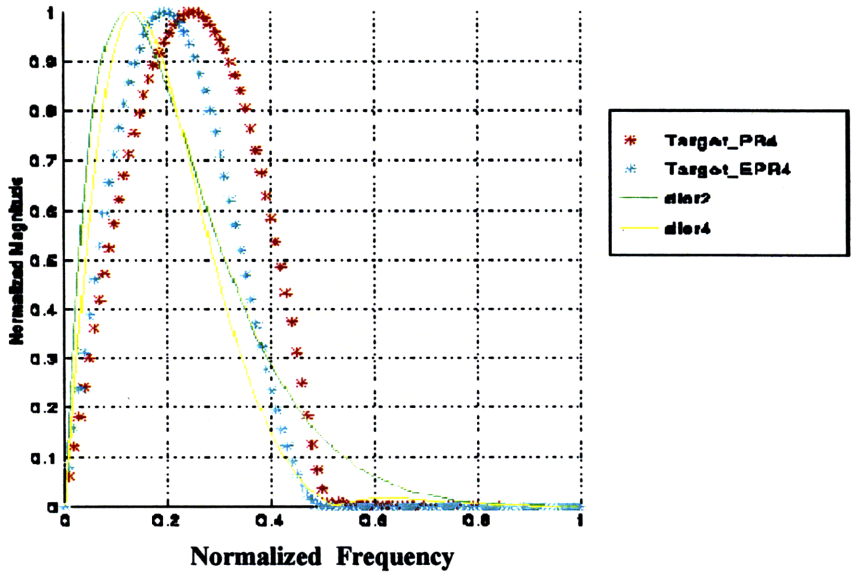


Figure 2-6: Frequency response of PR4, EPR4 targets and 2^{nd} ,4^{th} order Lorentzian

ECC used. Typical signals to noise ratios (SNR) for the signals off the read head are on the order of 15 to 25 dB.

Optimal data recovery for an ISI channel can be accomplished through maximum likelihood (ML) detection, which is realized in practice by a Viterbi Algorithm (VA) [14]. VA operates by finding the "most-likely" path in the trellis with 2^{m-1} states, where m is the channel length. The decoded bit stream is the sequence that corresponds to the ML path.

2.4 Pulse Shaping

A direct implementation of the ML detection on a storage channel is impractical, since the long length of the channel dipulse response makes the number of states in the VA prohibitively large. For example, if the discrete-time channel dipulse response has 10 significant terms, 2^9 states are required. To avoid this problem, the equalizer is put in place so that the combined response of the channel and the filter has only few significant terms; thus making ML detection on the overall channel possible (Figure 2-7). This sampled-data technology in which the amplitude of the read waveform is sampled several times during each pulse after a preliminary analog equalization step is thus referred to as PRML. PR systems have been widely studied in literature [15-17].

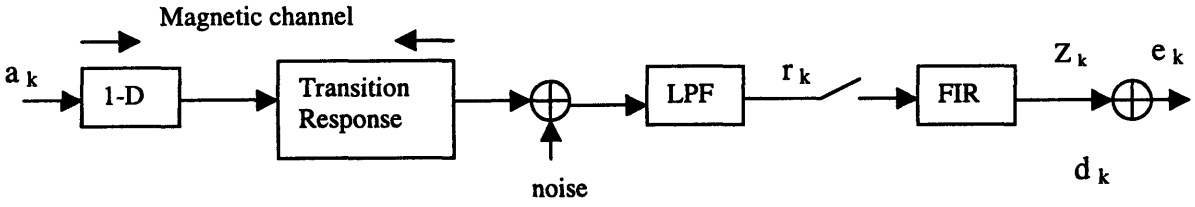


Figure 2-7: PRML Channel

The key advantage of PRML is that it works with sequences of received data, rather than single bits, comparing data sequences and determining which ones was most likely to have caused the observed signal. It is therefore more tolerant of ISI than pulse peak detection [18]. For our purpose, it is understood that the detector operates correctly on a sampled pulse waveform equalized to a particular PR signal target. PR signals provide a convenient way of specifying the shape of the sampled voltages.

Basically, the idea behind PR signaling is to increase the density so that ISI occurs, but control it so that the detector knows what signal shape to expect. These targets are specified as weighted values of unit delays z^{-1} and are usually conveniently chosen integer-coefficient polynomials of z^{-1}. Note that most of the PR polynomials in disk-drive literature use the symbol D instead of z^{-1}. PR targets can also be thought of as

the impulse-response coefficients of an FIR filter, thus modeling the entire write/read channel as an FIR system.

Table 2.1: General family of response targets with controlled ISI [19]

Name	N	Pulse Response	Step Response
PR4	1	0 1 0 -1 0	0 1 1 0
EPR4	2	0 1 1 -1 -1 0	0 1 2 1 0
EEPR4	3	0 1 2 0 -2 -1 0	0 1 3 3 1 0

Partial Response Polynomial, $P_n(D) = (1-D)(1+D)^n$

Equalization of the storage channel to many target functions could be considered. A suitable target whose responses are close to the actual channel response is chosen so that equalization is not performed at the expense of severe noise enhancement. At linear recording densities that are currently of interest, EPR4 is considered to which the channel can be equalized. For such a response, a properly equalized isolated transition on the medium will produce a pulse with four non-zero samples. The sampled values of the equalized pulse will be {0, 1, 2, 1, 0}. The discrete-time dipulse response of this channel is

$$G(D) = 1 + D - D^2 - D^3 \qquad (2.3)$$

where D is the delay element. The name PRML, therefore, stresses shaping the channel to an EPR4 target and performing ML detection using a VA.

Chapter 3

PRML Equalizer Design

A number of semiconductor vendors provide single-chip read-channel ICs for hard-disk drives. The implementation of the equalization and signal-processing functions range from all analog through mixed analog-digital to primarily digital approaches. The design approach depends on factors such as the manufacturer's experience, the process technology used, and the target data rate. The design objectives, such as data rate and detector complexity, are related to the desired cost and the status of the semiconductor process technology at the time of the design.

3.1 How PRML Works

PRML methods take advantage of the correlative nature of disk-drive channel [20]. Many excellent papers have been written describing the operation of these detection systems in a disk drive; the unfamiliar reader is referred to references [21-23].

PRML is a sampled-data technology in which the amplitude of the read waveform is sampled several times during each pulse, possibly after a preliminary analog equalization step (Figure 3-1). The analog samples will usually be digitally coded for further processing, although some manufacturers may prefer to do all the processing in the analog domain. Regardless of the type of processing, its goal is to yield a pulse shape

as close to an ideal target as possible. Equalization circuits shape the signal to conform to the desired PR4 response (Figure 3-2). When each pulse transition is sampled four times, the ideal PR4 (positive) transition response produces a 0, two 1s, and a 0. Viterbi sequence then identifies which sequence of data bits is the most likely to have been read from the disk, where 1 = transition, 0 = no transition.

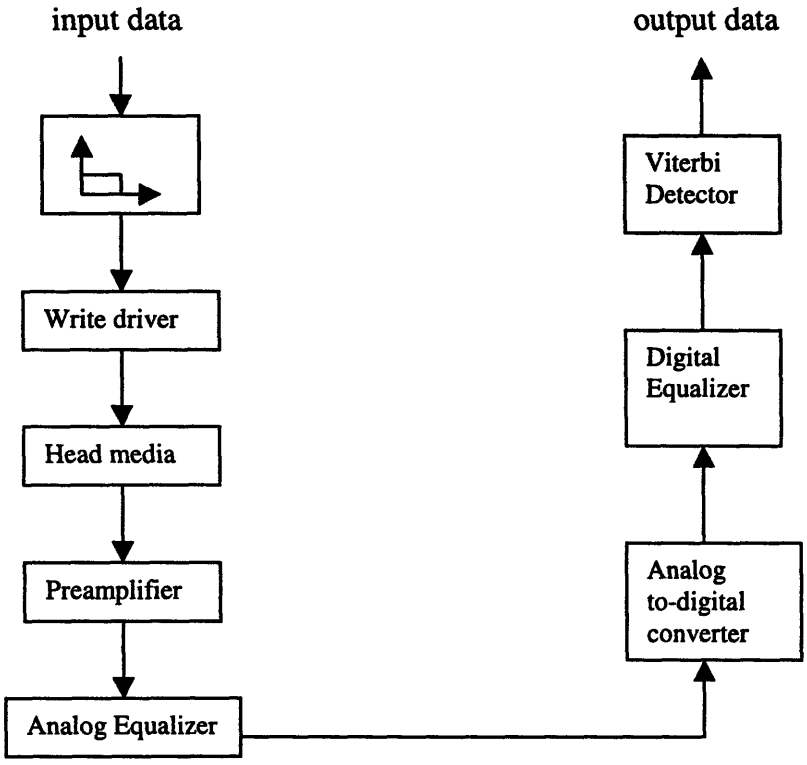


Figure 3-1: Equalization circuit for signal shaping

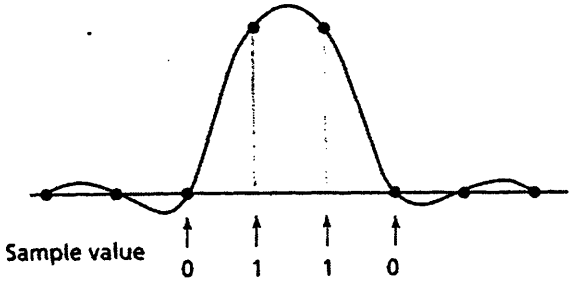


Figure 3-2: Ideal PR4 transition response

To properly equalize and detect the user-data bits, several data parameters must be adaptively controlled. These include channel gain, FIR filter coefficients, and sampling phase. Comparing the equalizer output samples with the expected three-level PR4 samples generates an error signal, which is used to produce adaptive control signals for each of the adaptive loops.

3.2 MSE Filter Design

The mean-square-error (MSE) approach is used in designing the FIR equalizer. As opposed to the other zero-forcing (ZF) approach [24] where the discrete-time response of the channel is forced to be identical to the EPR4 target, the principle of the MSE approach relaxes the constraint on the total removal of ISI and set of equalizer coefficients.

The MSE filter design is based on the minimization of the square of the error, $E(e_k^2)$, where the error is defined as the difference between the equalized samples, z_k, and their corresponding desired values, d_k, i.e.,

$$e_k = z_k - d_k . \tag{3.1}$$

For a PRML channel, characterized by the discrete impulse response, $1 + D - D^2 - D^3$, d_k is given by

$$d_k = a_k + a_{k-1} - a_{k-2} - a_{k-3}, \tag{3.2}$$

where a_k denotes the input sequence. z_k can be written as

$$z_k = f^T r_k, \tag{3.3}$$

where f and r_k are vectors containing filter coefficients and the FIR equalizer input samples respectively.

The solution for f is found iteratively using LMS algorithm. LMS updates filter coefficients in a direction opposite to and with an amount proportional to the gradient of $E(e_k^2)$ with respect to f. The equation for updating the FIR coefficients is as follows:

$$f_{k+1} = f_k - \beta r_k e_k . \tag{3.4}$$

3.3 System Architecture

Equalization of the read back signal to a partial response target in magnetic recording channel can be performed with a continuous time filter (CTF), an adaptive FIR filter or a combination of both [54]. In this section we will evaluate the combined CTF/FIR equalization configuration in the conventional PRML front-end architecture shown in Figure 3-3, and develop an estimate of the performance of the channel and the requirements in terms of tap order, boost and coefficient quantization effects. The mean-square-error function is evaluated within a simulated disk drive. To improve accuracy, the channel model is formed based upon measured waveforms from physical drives.

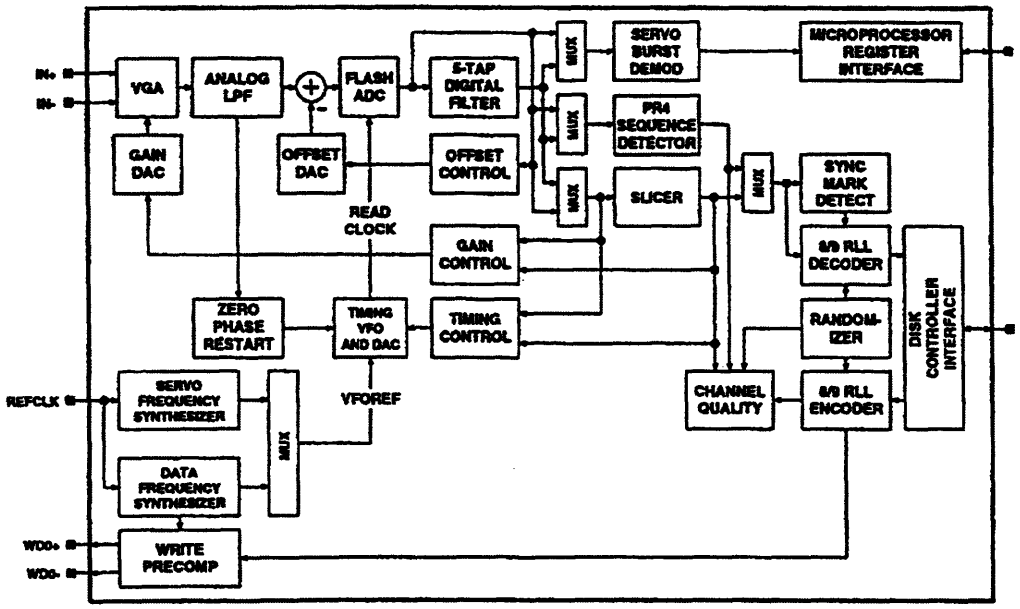


Figure 3-3: PRML block diagram [25]

In a conventional PRML front-end architecture, both a pre-amplifier and a variable-gain amplifier (VGA) amplify the sub-millivolt read-head signal to a desired level, typically $1V_{p\text{-}p}$. The gain of the VGA is regulated by an AGC. A programmable filter precedes the AGC, removing broadband noise and performing coarse equalization. The analog signal is then converted to digital by a sampler and 6-bit analog-to-digital converter (ADC). An adaptive equalizer conditions the signal to the proper PR target. Sampling phase is sensed at the output of the equalizer and corrections to the sampling instants are fed by to an oscillator driving the T/H of the ADC. After convergence of the equalizer and Phase-Locked-Loop (PLL), the properly conditioned signal is fed to a PRML detector, which decodes the bits encoded into the modulated waveform.

3.4 PRML Front-End

We will first investigate various trade-offs in the design of the Low-Pass Filter (LPF) and the digital FIR transversal filter in order to gain a better understanding of the FIR filter function in the PRML magnetic read channel.

A sequence of pseudo-random binary data is generated and run through an oversampled magnetic recording channel as shown in Figure 2-7. A MATLAB script-level optimization program was written which compares the sample values at the output of the equalizer against their corresponding desired values and a MSE value is computed. The program also keeps track of the optimum tap coefficient values of the FIR that minimize the distortion or mean-square error of the received pulse given by

$$RMSE \approx \frac{\sum_k e_k}{\sqrt{k}} \tag{3.5}$$

We will use the Root Mean Square Error (RMSE) value at the output of the FIR equalizer as the figure of merit in our analysis. If the error samples are uncorrelated, the MSE directly corresponds to the bit-error rate of the VA. However, in general, error

samples are correlated and their correlation as well as the MSE value will determine the bit-error rate.

The channel transition response is taken to be [26]

$$h(t)=\left[\frac{h_o}{1+\left(\frac{1.572(t+k)}{PW_{50}}\right)^2+\left(\frac{1.572(t+k)}{PW_{50}}\right)^4}\right]-\left[\frac{h_o}{1+\left(\frac{1.572(t-k)}{PW_{50}}\right)^2+\left(\frac{1.572(t-k)}{PW_{50}}\right)^4}\right] \quad (3.6)$$

where h_o is the zero-to-peak magnitude of the isolated transition response, which is normalized. This function is referred to as the 4th order Lorentzian. PW_{50} is the pulse width at half the peak magnitude. The read channel simulator is performed in the frequency domain analysis. The 4th order Lorentzian is chosen as a benchmark pulse. Simulations are performed for channel bit density (cbd) or PW_{50}/T of 2.5. The parameters for this read channel modeling include a partial-response target, normalized LPF bandwidth of 0.25 and no asymmetry, and unity read path gain. The EPR4 channel uses a PR4 front-end followed by a (1+D) transform. We found that the 5-tap FIR is the most optimal for PR4. Figure 3-4 shows the frequency-domain plot while Figure 3-5 shows the surface plot of the min RMSE (in dB) vs. tap values of a symmetrical 5-tap FIR filter.

The resolution for a combination of boost parameter and transversal FIR tap order is studied for an LPF boost range of 0 to12 dB. See Figure 3-6. Significant tolerance is observed for a range of the LPF boost setting. A 5- to 10-tap filter can compensate for the under- or over-boosting of the readback signal by the LPF. The plot shows that as the boost is increased from the optimal value of 6dB, the RMSE degrades by about 5dB. Higher tap order doesn't offer any significant improvement in RMSE in the range of 4 to 8 dB boost.

Figure 3-7 shows the coefficient quantization effect investigated with the same set of parameters as mentioned previously. The MATLAB simulation results show comparable performance for 3 to 9-tap transversal filters for up to 4-bit coefficient quantization. At higher resolution, the lower order tap filters indicate no significant

improvement in RMSE. The best case of equalization for 3- and 4-tap filters is with at least 4-bit quantization; a coarser resolution would reduce the equalization error to less than 28dB. These simulation results show the expected improvement in RMSE for a given tap order with higher bit resolution.

3.5 Previous Work

The equalizer architecture tradeoffs for magnetic recording channels have been extensively studied in literature. Aziz and Sonntag have performed the evaluation of the performance of several popular CTF configurations with FIR filters of different lengths using BER simulation [27]. Figure 3-3 shows a typical read channel design that uses a relatively low order CTF with a reasonable number of FIR taps. Research shows that, regardless of the number of taps in the FIR and CTF type, the performance of the CTF+FIR equalizers is about the same if the optimum CTF settings of corner frequency and boost are chosen. Therefore, the tradeoff can be made between the cost of the FIR filter versus the accuracy in the CTF setting and the sensitivity of the resulting system.

Of our particular interest is the case where a T spaced FIR filter, with a 7^{th} order linear phase Nyquist anti-aliasing CTF, is used to perform the equalization. It has been shown that this attempt at FIR only equalization cannot approach the performance of a system whose CTF provides boost and is optimized. There is some benefit to increase the number of taps from 4 to 6 to 10. But beyond ten taps, there is substantially more latency in the timing loop as the main tap position is pushed out. This results in increasing phase errors in the timing loop, hence overriding the improved equalization obtained with the larger number of taps. On the other hand, increasing the number of taps without changing the main tap position would lead to inadequate cancellation of the precursor ISI given that the CTF is not performing any equalization.

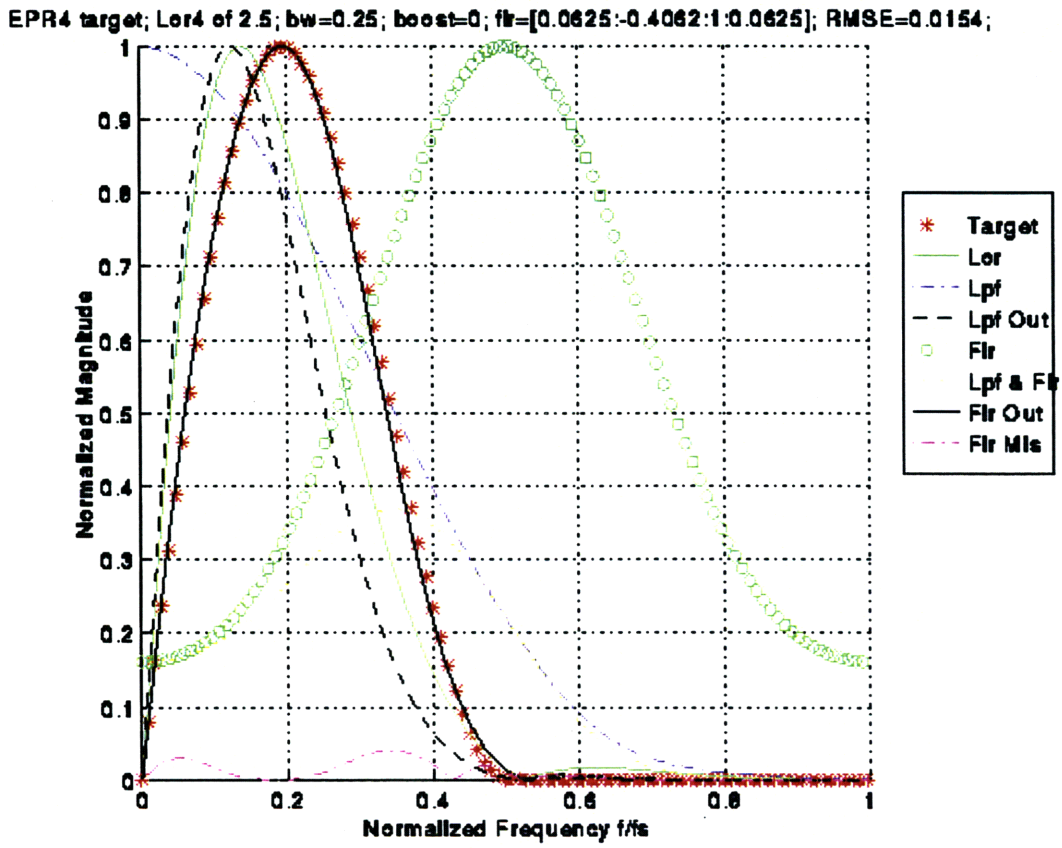


Figure 3-4: Frequency-domain plot of 5-tap transversal filter

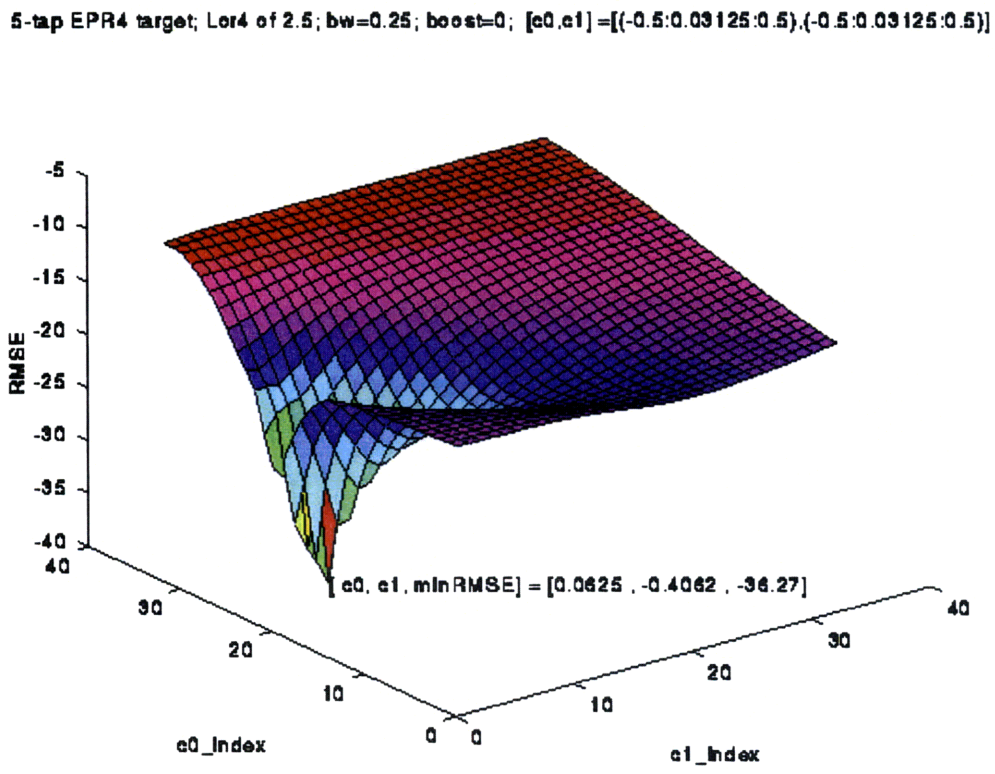


Figure 3-5: Surface plot of 5-tap transversal filter.

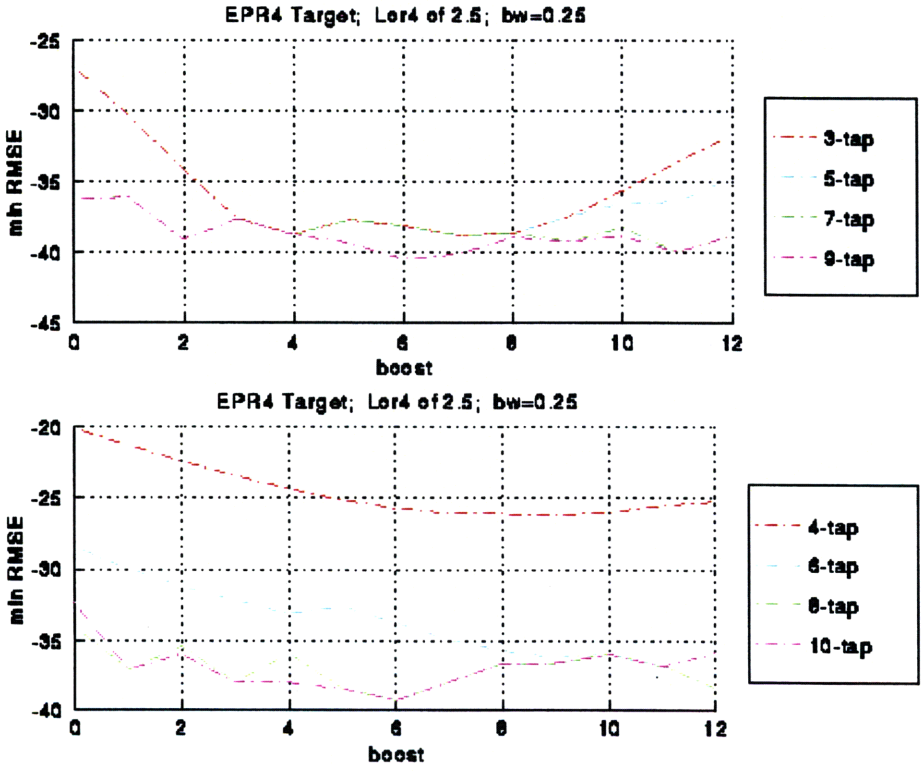


Figure 3-6: Minimum RMSE vs. boost for odd and even tap FIR

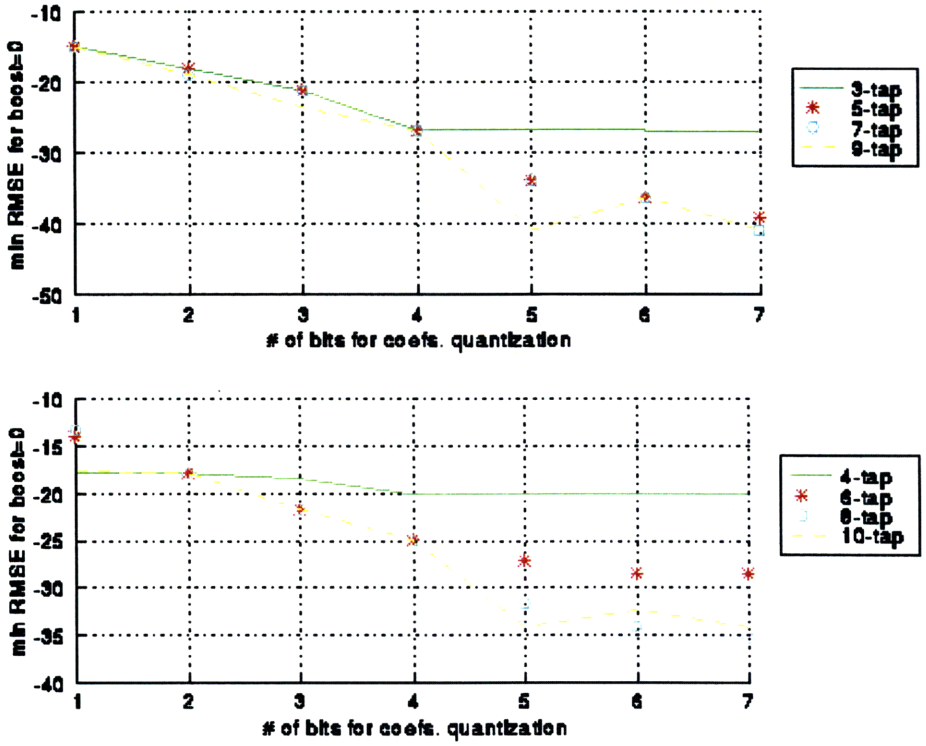


Figure 3-7: Coefficient quantization effect on 3- to 10-tap FIR

Chapter 4

FIR Architecture

Digital FIR filters are key component of state-of-the-art PRML detection read/channel IC's but consume significant circuit area and power. High-speed high-order programmable filters are difficult to be implemented efficiently because of the high implementation cost and the programmability requirements. In the literature, various kinds of filter architectures have been proposed for different applications [28-30].

There exist filter structures with superior performance where the filter coefficient values are not directly obtainable from the coefficient values of the z-transform transfer function by inspection. Instead, the z-transform transfer function is expressed as a system of nested difference equations and realized in one complete block. Among these well-known structures are the ladder [31-34] and lattice networks [6,9,35-38]. The lattice structures, in particular, have also found wide applications in system modeling, in digital speech processing and in the implementation of adaptive filters. [39-41].

The purpose of this research is to develop a novel lattice-based digital FIR filter architecture with programmable coefficients. We consider mainly the architecture implementation level of FIR filters to obtain a filter architecture that would take advantage of the lattice properties. This chapter will introduce the basics of lattice structure and the performance study of FIR equalizer candidates for PR4/EPR4 target equalization. By exploiting the lattice properties for a longer and more robust FIR equalizer, it would be much easier to design a simple LPF with no boost specifications, and still meet the timing requirements in the front-end channel.

4.1 Current Design Trend

Traditional direct or transpose architectures requiring explicit multiplication and addition of the samples and tap weights are well known in the disk drive read channel application [42]. The custom-designed Distributed Arithmetic digital filters offer improved performance and area relative to earlier standard cell implementations of this filter by logical splitting of the input shift register into odd and even interleaves using a signed-digit, offset binary numbering system [43,44].

Multiplier-and-accumulator (MAC) based architectures [45,46] are frequently adopted for programmable filters. However, the cost of multipliers is high and they are not suitable for high-order filters. Published work suggests efficient high-speed digital filter architectures in DSP applications where the FIR filter is also one important building block. Design approaches such as power-of-two coefficients, minimum-adder multiplier blocks, block realization using matrix representation of convolution and coefficient-dependent multiplication scheme have resulted in hardware-efficient architectures for VLSI digital filters and Field Programmable Gate Array (FPGA) implementations of direct-form FIR digital filters [47-52]. The increasing demand for video signal processing and transmission has seen the increasing use of high-speed and high-order programmable FIR filters for performing adaptive pulse shaping and signal equalization on the received data in real time [53]. However, long latency and high implementation cost make the above mentioned techniques unsuitable for PRML adaptive or high filtering order applications in the magnetic read channel. Increasing the filter tap order results in substantial latency and introduces increased phase errors in the timing loop that dominates any improved equalization [54]. The direct form is suitable only for low-order filters.

The basic structures of FIR systems such as direct (cascade and parallel form structures) have been developed either directly from the system function or from the input/output difference equation. Other structures with special properties have also been developed. One approach is based on state-variable representations and linear transformation [4]. Still another approach is to develop digital filter structures with certain desirable properties such as wave digital filters [55] and lattice structures [56].

Our approach is to understand the motivation and the properties of the FIR lattice structures by defining the structure as a flow graph and then use flow graph manipulations and z-transform analysis to illuminate the properties of the structure. The lattice filter is an alternative means of realizing a digital filter transfer function. Although the lattice filter structure (also known as the ladder structure) does not have the minimum number of multipliers and adders for a transfer function realization, it does have several advantageous properties. These include cascading of identical sections, coedficients with magnitude less than 1, good numerical round-off characteristics, orthogonalization of the input signal on a stage-by-stage basis, flexibility in choosing the order.

Several lattice structures have been proposed for the implementation of all-pole and pole-zero digital filters [57-61]. For instance, one way to reduce the delay-free computation path has been proposed for an infinite impulse response (IIR) filter where retiming is used by moving every other delay element along the bottom path of the lattice to the top path. Merging the arithmetic blocks and deterring the carry propagating addition until after the final stage filter stage can also reduce hardware and achieve significant speed when implementing the multiplier with a Wallace tree.

Although much research has been done on lattice digital filters, much of this work has focused on autoregressive/moving average modeling techniques for speech and signal processing, high-speed VLSI and wideband filters implementation. Only a single lattice structure, due to Itakura and Saito [9], is available for the implementation of all-zero filters for speech analysis and synthesis. There has yet been no published research on the use of lattice structure to equalizer architectures in the read channel application. But the lattice filter structure has been useful in many of the above mentioned applications and promises to be useful in our application as well, where transversal finite impulse response filters are used in an adaptive manner.

4.2 Transversal FIR Filter

We are required to approximate the desired response *D(z)*, with a realizable FIR filter transfer function, *H(z)*. The transfer function of a FIR filter with real impulse response is defined by the *z*-transform of its impulse response

$$H(z) = \sum_{k=0}^{n-1} h_k z^{-k} \tag{4.1}$$

where the frequency response, *H(f)*, is obtained by evaluating *H(z)* on the unit circle,

$$z = e^{j2f\pi} \tag{4.2}$$

Although FIR filters can have recursive computation methods, in practice they are always implemented as transversal filters. The Signal Flow Graph (SFG) of the transversal filter shown in Figure 4-1 is a simple tapped delay line, without any feedback, in which the outputs of each stage, multiplied by a fixed constant, are added together. The most important advantage of these transversal filters is their inherent stability property.

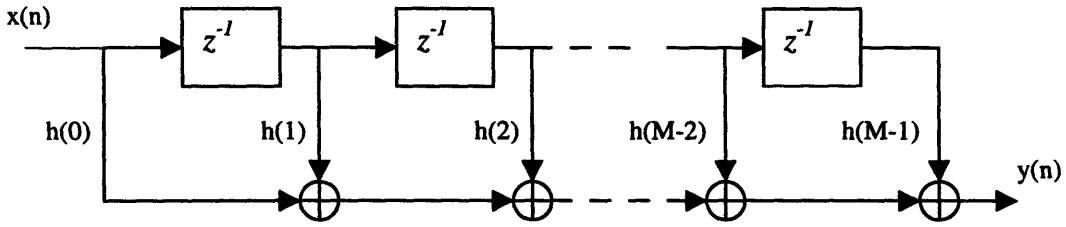


Figure 4-1: Transversal FIR filter.

The *y(n)* output sequence of a FIR filter is a linear convolution of the input data sequence *x(n)*, and the sequence of the filter coefficients a_k, *k=0,..., M-1*:

$$y(n) = \sum_{k=0}^{M-1} a_k x(n-k) \tag{4.3}$$

4.3 Lattice Structure Basics

The digital lattice filter realization consists of a repeated structure of two multipliers, two adders, and a delay element z^{-1}. Figure 4-2 illustrates the signal flow of an (M-1)-stage lattice system along with a typical stage that shows the following set of difference equations:

$$f_o(n) = g_o(n) = x(n) \tag{4.4a}$$

$$f_m(n) = f_{m-1}(n) + K_m g_{m-1}(n-1) \qquad m = 1, 2, \ldots, M-1 \tag{4.4b}$$

$$g_m(n) = K_m f_{m-1}(n) + g_{m-1}(n-1) \qquad m = 1, 2, \ldots, M-1 \tag{4.4c}$$

The output $f_m(n)$ of the m-stage or (M-1)-stage lattice filter can be expressed as

$$f_m(n) = \sum_{k=0}^{m} a_m(k) x(n-k) \tag{4.5}$$

Since it is a convolution sum, it follows that the z-transform relationship is

$$F_m(z) = H_m(z) X(z) \tag{4.6}$$

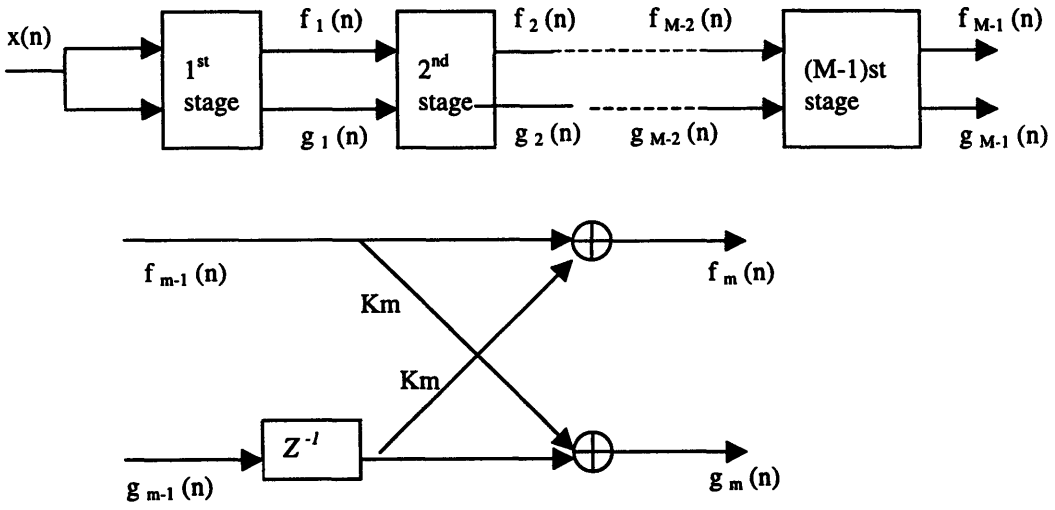


Figure 4-2: (M-1) -stage lattice filter

In order to see that that the FIR lattice system is a useful and interesting structure its properties must be understood. First, it is recognized as an FIR system where all of the signal flow is from left to right. Since there are no loops, the impulse response has finite length. If the input is $x[n] = \delta[n]$, this impulse propagates immediately to the output along the top row of horizontal branches, with unity gain. All other paths from the input to the output pass through at least one delay element. Thus, $h[0]=1$. An impulse at the input also propagates to the output through the bottom row of horizontal branches, encountering N delay elements and a final gain of k_M. All other paths from the input to the output zigzag between the top row of horizontal branches and the bottom row. Therefore, each of these paths passes through at least one unit delay element and at most (N-1) delay elements. Therefore, the impulse response values for 0<n<N are determined by sums of products of the k_i's. Thus, we conclude that the system function relating $F_m(z)$ to $X(z)$ is of the form

$$H(z) = \frac{F_m(z)}{X(z)} = A_m(z) = \left[1 + \sum a_m z^{-m}\right] \quad \text{where } 1<m<N \tag{4.7}$$

The other output component from the lattice, namely, $g_m(n)$, can also be expressed in the form of a convolution sum as in (4.5) by using another set of coefficients, say $\{b_m(k)\}$. The output $g_m(n)$ from an m-stage lattice filter is by the convolution sum of the form

$$g_m(n) = \sum_{k=0}^{m} b_m(k)x(n-k) \tag{4.8}$$

where the filter coefficients $\{b_m(k)\}$ are associated with a filter that produces $f_m(n)=y(n)$ but operates in reverse order. Consequently,

$$b_m(k) = a_m(m-k) \qquad k = 0, 1, \ldots, m \tag{4.9}$$

In the z-transform domain (4.9) becomes

$$G_m(z) = B_m(z)\, X(z) \tag{4.10}$$

where $B_m(z)$ represents the system function of the FIR filter with coefficients $\{b_m(k)\}$, that is,

$$B_m(z) = \sum_{k=0}^{m} b_m(k) z^{-k} \tag{4.11}$$

Since $b_m(k) = a_m(m-k)$, (4.11) may be expressed as

$$B_m(z) = \sum_{k=0}^{m} a_m(m-k) z^{-k} = \sum_{l=0}^{m} a_m(l) z^{l-m} = z^{-m} \sum_{l=0}^{m} a_m(l) z^{l} = z^{-m} A_m(z^{-1}) \tag{4.12}$$

The relationship in equation (4.12) implies that the zeros of the FIR filter with system function $B_m(z)$ are simply the reciprocals of the zeros of $Am(z)$. Hence $B_m(z)$ is called the reciprocal or reverse polynomial of $A_m(z)$.

Now that we have established the equivalence between an mth-order direct-form FIR filter and an m-order or m-stage lattice filter, we can next transfer the recursive lattice equations in (4.4a) through (4.4c) to the z-domain. Thus we have

$$F_o(z) = G_o(z) = X(z) \tag{4.13a}$$

$$F_m(z) = F_{m-1}(z) + K_m z^{-1} G_{m-1}(z) \qquad m = 1,2, \ldots, M-1 \tag{4.13b}$$

$$G_m(z) = K_m F_{m-1}(z) + z^{-1} G_{m-1}(z) \qquad m = 1,2, \ldots, M-1 \tag{4.13c}$$

If we divide each equation by $X(z)$, we obtain the desired results in the form

$$A_o(z) = B_o(z) = 1 \tag{4.14a}$$

$$A_m(z) = A_{m-1}(z) + K_m z^{-1} B_{m-1}(z) \qquad m = 1,2, \ldots, M-1 \tag{4.14b}$$

$$B_m(z) = K_m A_{m-1}(z) + z^{-1} B_{m-1}(z) \qquad m = 1,2, \ldots, M-1 \tag{4.14c}$$

Thus the lattice stage is described in the z-domain by the matrix equation

$$\begin{bmatrix} A_m(z) \\ B_m(z) \end{bmatrix} = \begin{bmatrix} 1 & K_m \\ K_m & 1 \end{bmatrix} \begin{bmatrix} A_{m-1}(z) \\ z^{-1} B_{m-1}(z) \end{bmatrix} \tag{4.15}.$$

We will next develop the relationships for converting the lattice parameters $\{K_i\}$, that is, the reflection coefficients, to the direct-form filter coefficients $\{a_m(k)\}$, and vice versa.

4.4 Transversal to Reflection Coefficients Transformation

If this is to be a useful structure, the relationship between the lattice parameters $\{K_i\}$, that is, the reflection coefficients, and the direct-form filter coefficients must be studied. Equivalently, if the system functions are defined, so is its impulse response as well.

Suppose that we are given the FIR coefficients for the direct-form realization or, equivalently, the polynomial $A_m(z)$, and we wish to determine the corresponding lattice filter parameters $\{K_i\}$. For the m-stage lattice, we immediately obtain the parameter $K_m = a_m(m)$, where $a's$ are the transversal filter coefficients. To obtain K_{m-1} we need the polynomials $A_{m-1}(z)$ since, in general, K_m is obtained from the polynomial $A_m(z)$ for m=M-1, M-2, ...,1. Consequently, we need to compute the polynomials $A_m(z)$ starting from m=M-1 and stepping down successively to m=1.

The recursive relation for the polynomials is easily determined from (4.14b) and (4.14c).

$$\begin{aligned} A_m(z) &= A_{m-1}(z) + K_m z^{-1} B_{m-1}(z) \\ &= A_{m-1}(z) + K_m [\, B_m(z) - K_m A_{m-1}(z) \,] \end{aligned} \tag{4.16}$$

where $B_m(z)$ is the reverse polynomial of $A_m(z)$.

Solving for $A_{m-1}(z)$, we obtain

$$A_{m-1}(z) = \frac{A_m(z) - K_m B_m(z)}{1 - K_m^2}, m = M-1, M-2, \ldots, 1 \tag{4.17}$$

Thus we compute all lower-degree polynomials $A_m(z)$ beginning with $A_{m-1}(z)$ and obtain the desired lattice coefficients from the relation $K_m = a_m(m)$. The step-down recursion is similar to the Schur-Cohn stability test described in Proakis [62]. The procedure works as long as the $|K_m| \neq 1$ for m = 1, 2, ..., M-1.

It is relatively easy to obtain a formula for recursively computing K_m from the step-down recursive equation in (4.17). We begin with m=M-1 and step down to m=1.

$$K_m = a_m(m) \qquad a_{m-1}(0) = 1 \tag{4.18}$$

$$a_{m-1}(k) = \frac{a_m(k) - K_m b_m(k)}{1 - K_m^2} = \frac{a_m(k) - a_m(m)a_m(m-k)}{1 - a_m^2(m)}, \qquad (4.19)$$

$$\text{for } 1 \le k \le (m-1)$$

The above equation indicates that the recursive equation breaks down if any lattice parameters $|K_m| = 1$. If this occurs, it is indicative of the fact that the polynomial $A_{m-1}(z)$ has a root on the unit circle.

A similar analysis can be performed for the conversion of lattice coefficients to direct-form filter coefficients using (4.12) and (4.14b). The solution is obtained recursively, beginning with m=1. Thus we obtain a sequence of (M-1) FIR filters, one for each value of m. The procedure is best illustrated by means of an example.

Given a 3-stage lattice filter with coefficients $K_1 = \frac{1}{4}, K_2 = \frac{1}{4}, K_3 = \frac{1}{3}$, we determine the FIR filter coefficients by recursively solving (4.14b), beginning with m=1. Thus we have

$$A_1(z) = A_0(z) + K_1 z^{-1} B_0(z) = 1 + K_1 z^{-1} = 1 + \frac{1}{4} z^{-1}$$

Hence the coefficients of an FIR filter corresponding to the single-stage lattice are

$a_1(0) = 1, a_1(1) = K_1 = \frac{1}{4}$. Since $B_m(z)$ is the reverse polynomial of $A_m(z)$, we have

$$B_1(z) = \frac{1}{4} + z^{-1}$$

We next add the second stage to the lattice. For m=2, (4.14b) yields

$$A_2(z) = A_1(z) + K_2 z^{-1} B_1(z) = 1 + \frac{5}{16} z^{-1} + \frac{1}{4} z^{-2}$$

Hence the FIR filter parameters corresponding to the two-stage lattice are

$a_2(0) = 1, a_2(1) = \frac{5}{16}, a_2(2) = \frac{1}{4}$. Also

$$B_2(z) = \frac{1}{4} + \frac{5}{16} z^{-1} + z^{-2}$$

Finally, including the third stage to the lattice results in the polynomial

$$A_3(z) = A_2(z) + K_3 z^{-1} B_2(z) = 1 + \frac{19}{48} z^{-1} + \frac{17}{48} z^{-2} + \frac{1}{3} z^{-3}$$

Consequently, the desired direct-form FIR filter is characterized by the coefficients

$$a_3(0) = 1 \qquad a_3(1) = \frac{19}{48} \qquad a_3(2) = \frac{17}{48} \qquad a_3(3) = \frac{1}{3}$$

This examples illustrates a lattice structure with parameters K_1, K_2 and K_3 that corresponds to a class of 3 direct-form FIR filters with system functions $A_1(z)$, $A_2(z)$ and $A_3(z)$. The characterization of a class of m FIR filters in direct form requires m(m+1)/2 filter coefficients. In contrast, the lattice-form characterization requires only m reflection coefficients K_i's. This is because the lattice provides a more compact representation for the class of m FIR filters due to the fact that the addition of stages to the lattice does not alter the parameters of the previous stages. On the other hand, the addition of the mth stage to the lattice with (m-1) stages results in a FIR filter with system function $A_m(z)$ that has coefficients totally different from the coefficients of the lower-order FIR filter with system function $A_{m-1}(z)$.

4.5 Lattice FIR Candidates

With the above understanding of transversal filter and lattice filter transformations we will design symmetrically programmable range of tap coefficients for our target equalization. For comparison purposes, we will perform a consistent evaluation and design of an FIR lattice filter equivalent to a 5-tap FIR transversal filter with transversal coefficients as follows:

$$c_4(0) = \frac{4}{32} \qquad c_4(1) = -\frac{17}{32} \qquad c_4(2) = \frac{63}{32} \qquad c_4(3) = -\frac{17}{32} \qquad c_4(4) = \frac{4}{32} \tag{4.20}$$

Suppose we have a filter of order m=1. The output of such a filter is

$$y(n) = x(n) + a_1(1)x(n-1) \tag{4.21}$$

This output can be obtained from a first-order or single-stage lattice filter, illustrated in Figure 4-3, by exciting both inputs by $x(n)$ and selecting the output from the top branch. The output is exactly (4.21), if we select $K_1 = a_1(1)$.

The output of a direct-form FIR filter for which m=2 is

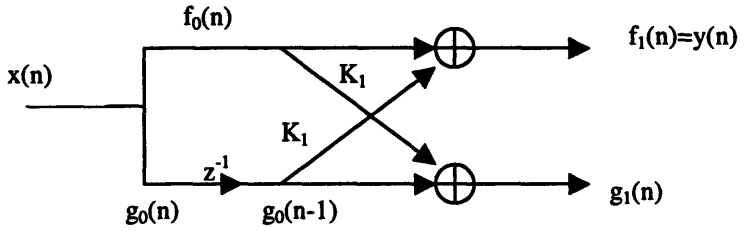
$$y(n) = x(n) + a_2(1)x(n-1) + a_2(2)x(n-2) \tag{4.22}$$

By cascading two lattice stages as shown in Figure 4-4, it is possible to obtain the same output as (4.22). The output from the first stage is

$$\begin{aligned} f_1(n) &= x(n) + K_1 x(n-1) \\ g_1(n) &= K_1 x(n) + x(n-1) \end{aligned} \tag{4.23}$$

The output from the second stage is

$$\begin{aligned} f_2(n) &= f_1(n) + K_2 g_1(n-1) \\ g_2(n) &= K_2 f_1(n) + g_1(n-1) \end{aligned} \tag{4.24}$$



$$f_0(n) = g_0(n) = x(n)$$
$$f_1(n) = f_0(n) + K_1 g_0(n-1) = x(n) + K_1 x(n-1)$$
$$g_1(n) = K_1 f_0(n) + g_0(n-1) = K_1 x(n) + x(n-1)$$

Figure 4-3: Single-stage lattice filter

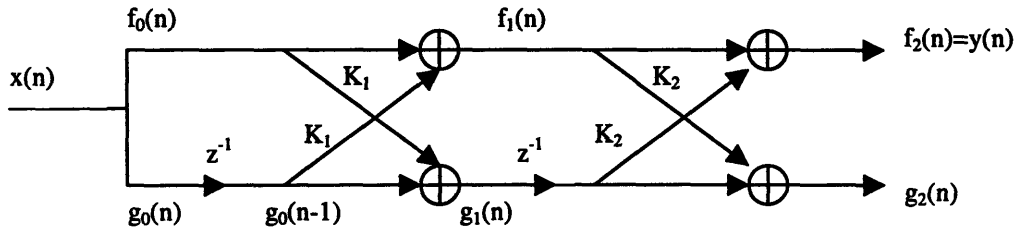


Figure 4-4: Two-stage lattice filter

It is apparent from our previous analysis of the lattice structure that the filter coefficients for the lattice filter that produces $f_l(n)$ are $\{1, K_l\} = \{1, a_l(l)\}$ while the coefficients for the filter with output $gl(n)$ are $\{K_l, 1\} = \{a_l(l), 1\}$. These two sets of coefficients are in reverse order. For the two-stage lattice filter, with the output given by (4.24), $g_2(n)$ can be expressed in the form

$$\begin{aligned} g_2(n) &= K_2 f_1(n) + g_1(n-1) \\ &= K_2[x(n) + K_1 x(n-1)] + K_1 x(n-1) + x(n-2) \\ &= K_2 x(n) + K_1(1+K_2)x(n-1) + x(n-2) \\ &= a_2(2)x(n) + a_2(1)x(n-1) + x(n-2) \end{aligned} \tag{4.25}$$

Similarly,

$$\begin{aligned} f_2(n) &= f_1(n) + K_2 g_1(n-1) \\ &= x(n) + K_1 x(n-1) + K_2[K_1 x(n-1) + x(n-2)] \\ &= x(n) + K_1(1+K_2)x(n-1) + K_2 x(n-2) \\ &= x(n) + a_2(1)x(n-1) + a_2(2)x(n-2) \end{aligned} \tag{4.26}$$

Two direct-form structures of the FIR filter are illustrated in Figure 4-5.

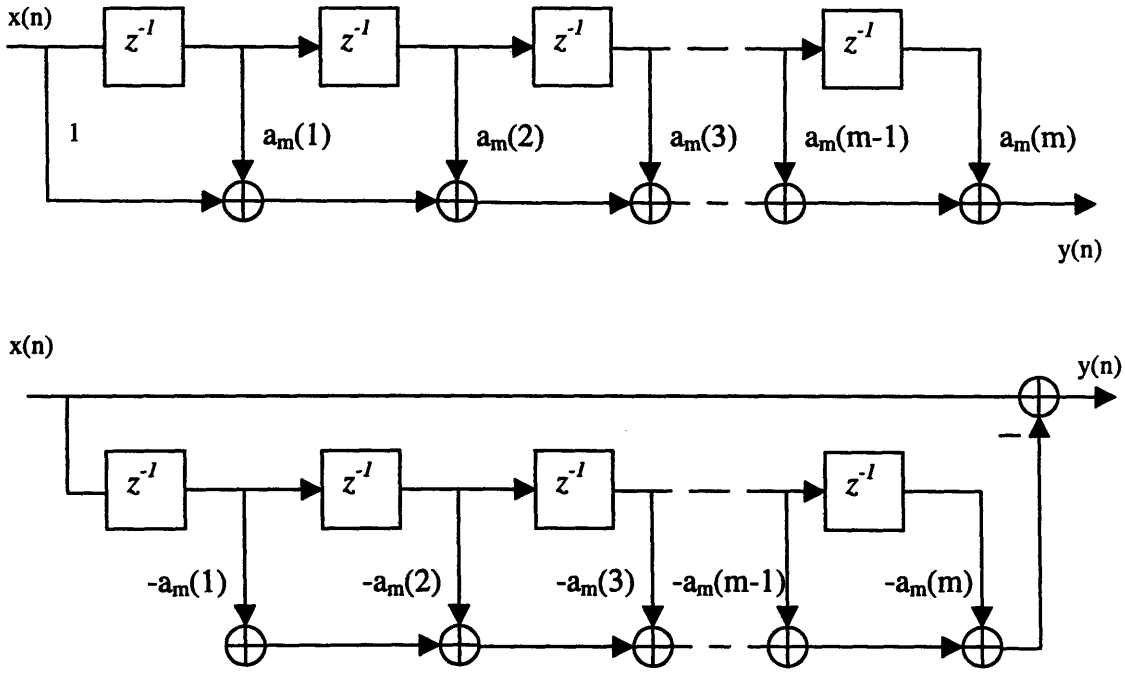


Figure 4-5: Direct-form realization of the FIR prediction filter

We first considered a scaled direct lattice implementation and a prediction error filter. The first option would implement an equivalent 5-tap FIR transversal filter with a 4-stage lattice filter as shown in Figure 4-6. But the top row output of the basic lattice filter is always 1 due to the unity gain along the no delay path. Hence, the value of the equivalent transversal tap 0 is always 1. We can consider a constant scaling of the coefficients at the lattice filter output. However, that would lead to an increase in the computational complexity and hardware implementation, thus defeating any improvement in speed and area obtained by using the lattice structure.

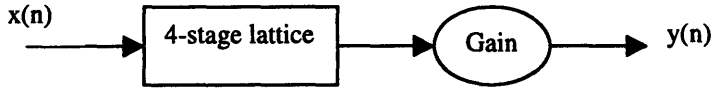


Figure 4-6: Scaled direct lattice implementation

We next consider the one-step forward linear predictor, which forms the prediction of the value $x(n)$ by a weighted linear combination of the past values $x(n-1)$, $x(n-2),\ldots,x(n-p)$. Hence the linearly predicted value of $x(n)$ is

$$\hat{x}(n) = -\sum_{k=1}^{p} a_p(k)x(n-k) \tag{4.27}$$

where the $\{-a_p(k)\}$ represent the weights in the linear combination. These weights are called the prediction coefficients of the one-step forward linear predictor of order p. The negative sign in the definition of $x(n)$ is for mathematical convenience and conforms with current practice in the technical literature.

Linear prediction is equivalent to linear filtering where the predictor is embedded in the linear filter as shown in Figure 4-7. It is called a predictor error filter with input sequence $\{x(n)\}$ and output sequence $\{f_p(n)\}$.

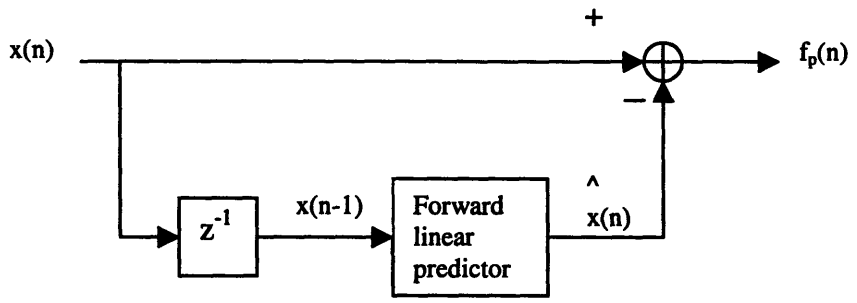


Figure 4-7: Forward linear prediction

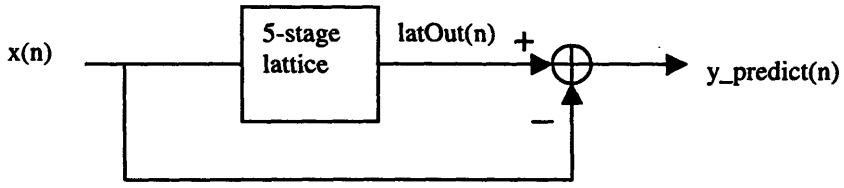
The difference between the value *x(n)* and the predicted value *x(n)* is called the forward prediction error, denoted as $f_n(n)$;

$$f_p(n) = x(n) - \hat{x}(n) = x(n) + \sum_{k=1}^{p} a_p(k)x(n-k) \tag{4.28}$$

By cascading 5 lattice stages, we obtain the forward prediction error $f_5(n)$ at the output *y(n)* where the equivalent transversal coefficient values are:

$$f_5(n) = x(n) + a_5(1)x(n-1) + a_5(2)x(n-2) + a_5(3)x(n-3) + a_5(4)x(n-4) + a_5(5)x(n-5) \tag{4.29}$$

Subtracting *x(n)* from the output of the forward signal gives the desired sample value but delayed. See Figure 4-8.



$$y\_predict(n) = latOut(n) - x(n)$$

Figure 4-8: Lattice Predictor Block Diagram

$$y\_predict(n) = a_5(1)x(n-1) + a_5(2)x(n-2)$$
$$+ a_5(3)x(n-3) + a_5(4)x(n-4) + a_5(5)x(n-5)$$

(4.30)

Equivalently,

$$y(n-1) = y\_predict(n)$$
$$= c_4(0)x(n-1) + c_4(1)x(n-2) + c_4(2)x(n-3) + c_4(3)x(n-4) + c_4(4)x(n-5)$$

Both the scaled and the lattice predictor filter blocks were evaluated in MATLAB. A sequence of pseudo-random binary data is generated to simulate the Analog-to-Digital Converter which is usually implemented as a fully parallel 6 bit flash converter. The ADC output is run through the FIR block. Figure 4-9 shows the equalized output of the scaled version of the direct implementation. The FIR lattice predictor filter also equalizes to PR4 target as shown in Figure 4-10 after performing the subtraction.

However, we observed a major setback with these two lattice architectures when we look at the range of the programmable lattice coefficients. Let's first refer back to specified user programmable coefficients range for a five tap digital transversal filter used to provide additional equalization for fine shaping of the incoming read signal to PR4 waveform. The two outer coefficients are generally set to a range from –1/2 to +1/2. The second and fourth coefficients are set to a range from –1 to +1. The center coefficient is set to a range of +1/2 to +3/2. Using the transversal to lattice filter coefficients transformations described in section 4.4, we ran Monte-Carlo simulations in MATLAB for transversal to lattice coefficient conversions. Figure 4-11 shows the range of equivalent transversal coefficients with the outer taps scaled to 1. MATLAB programs randomly pick values within the typical customer coefficient specifications, then scale and convert to reflection coefficients, *k's*. The range of *k's* we obtained was not bounded. Similarly, we ran MATLAB simulations for the predictor filter to determine the equivalent range of reflection coefficients. The results shown in Figure 4-12 again indicate unbounded *k's*. It is obvious that the combination of *k's* required to implement the equivalent transversal coefficients would be inefficient in hardware implementation.

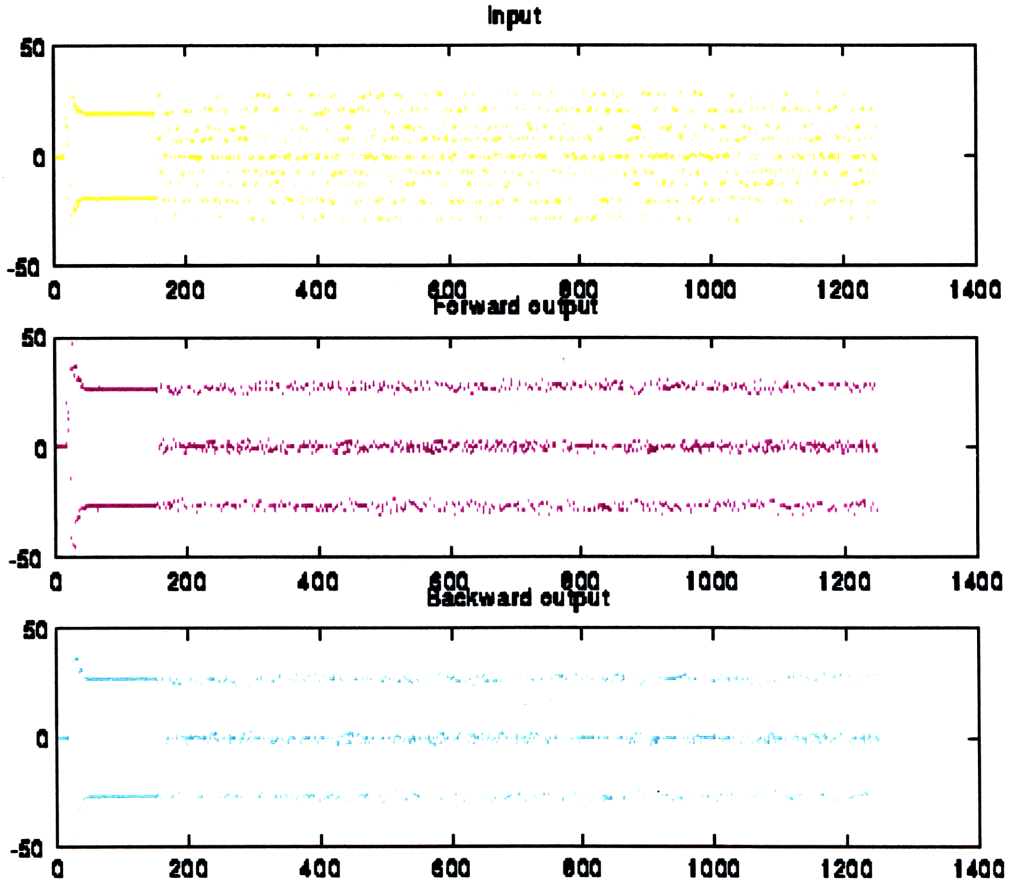


Figure 4-9: Equalization by scaled direct lattice filter

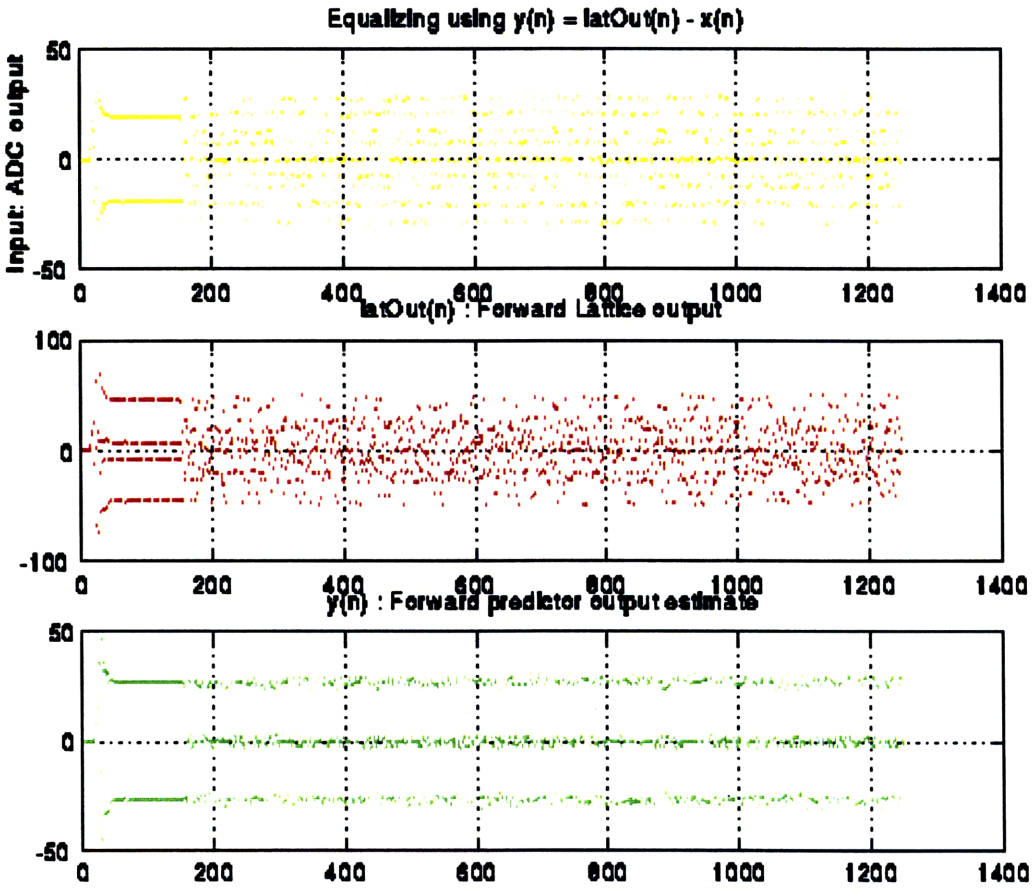


Figure 4-10: Equalization by predictor lattice filter

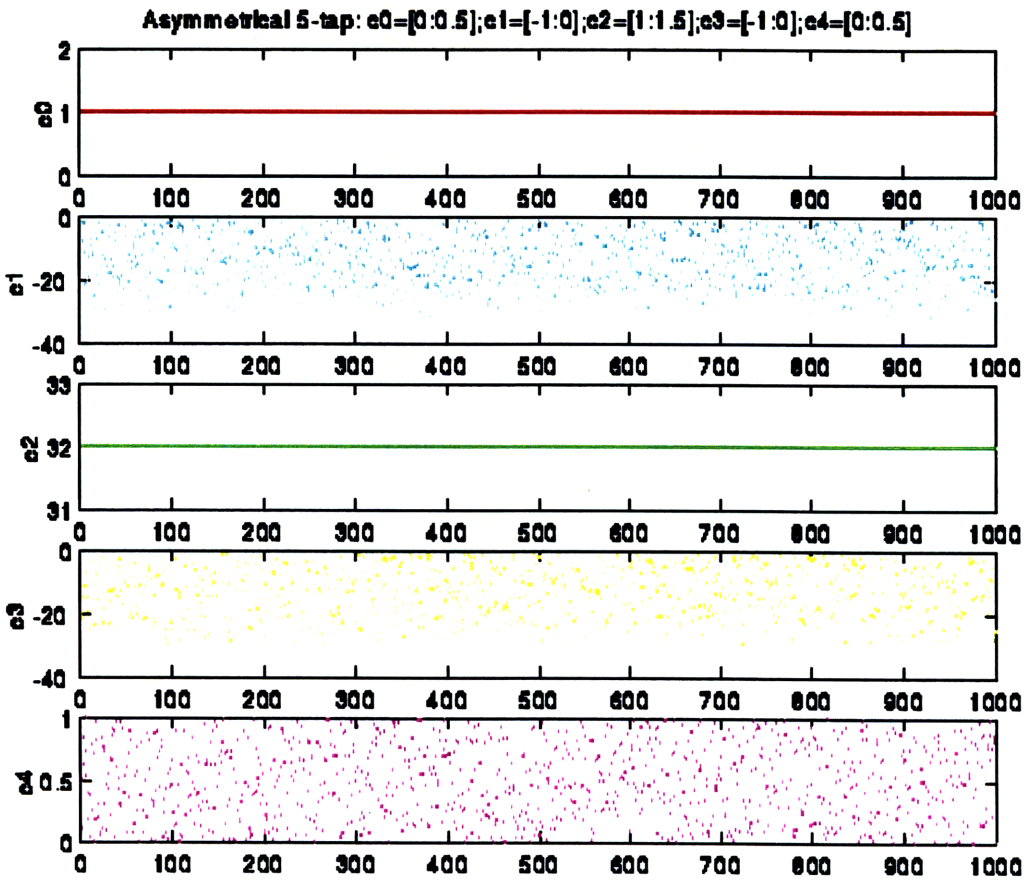


Figure 4-11: Equivalent range of transversal coefficients for scaled direct lattice filter

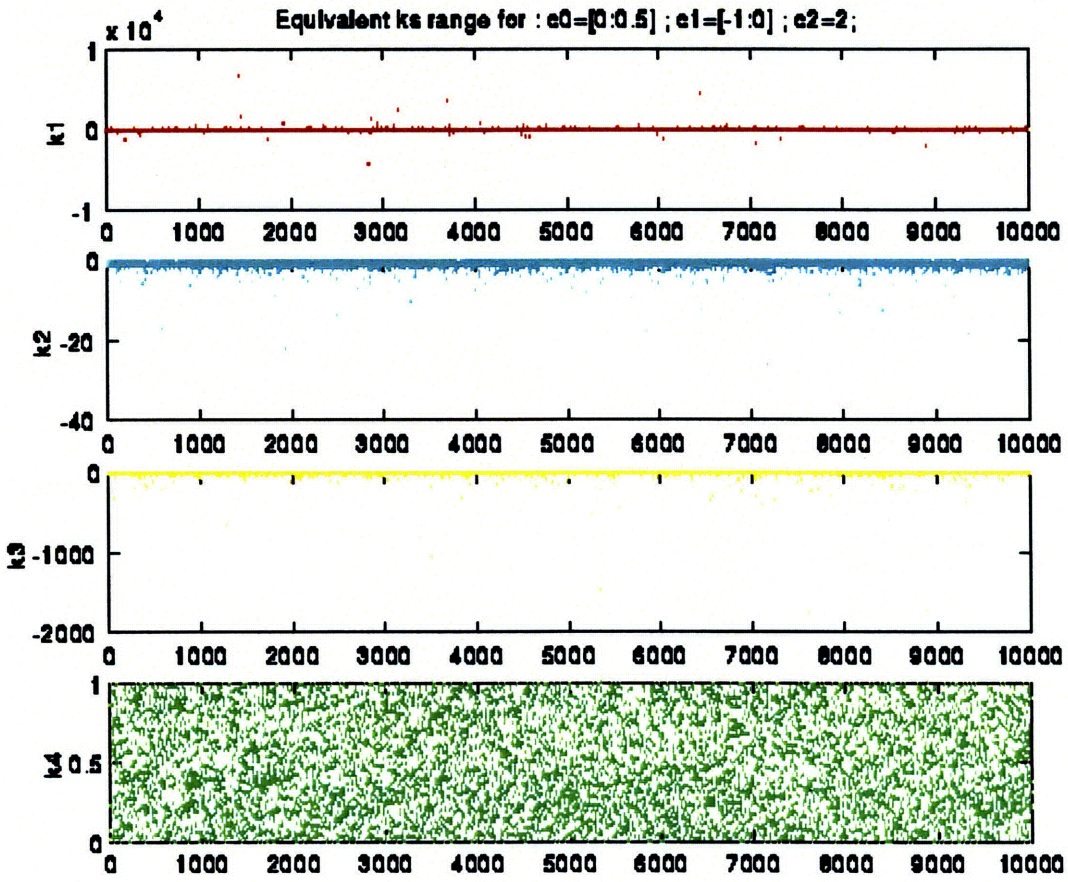


Figure 4-12: Range of reflection coefficients for predictor lattice filter.

4.6 Hybrid FIR Lattice Filter

We have previously found in our lattice structure analysis that the filter coefficients for the lattice filter that produces the forward output $f_m(n)$ are $\{1, K_1, \ldots, K_m\}=\{1, a_m(1), \ldots, a_m(m)\}$ while the coefficients for the backward output $g_m(n)$ are $\{K_m, \ldots, K_1, 1\} = \{a_m(m), \ldots, a_m(1), 1\}$.

Keeping in mind that these two sets of coefficients are in reverse order, we chose the lattice configuration shown in Figure 4-13. By introducing a delay in the top lattice block and summing the delayed forward signal and the backward signal, we obtain symmetrically programmable range of coefficients whose values are determined by two independent sets of reflection coefficients: the top lattice block coefficients, *TK's* and the bottom lattice block coefficients, *BK's*.

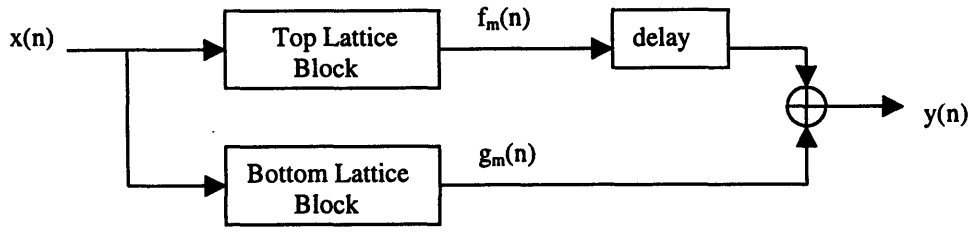


Figure 4-13: Hybrid lattice filter block diagram

We will focus on the design of a five-tap filter to compare the area and speed performance of the hybrid digital FIR lattice filter with the traditional digital FIR transversal filter. Higher order filters can easily be implemented based on the following general conditions:

$$y(n) = f_m(n-D) + g_m(n)$$
$$= \begin{Bmatrix} even\ \#\ stages & , & even\ delay: odd\ \#tap & 2\ center\ taps \\ & & odd\ delay\ : even\ \#tap & 1\ center\ tap \\ odd\ \#\ stages & , & even\ delay: even\ \#tap & 1\ center\ tap \\ & & odd\ delay\ : odd\ \#tap & 2\ center\ taps \end{Bmatrix} \quad (4.31)$$

where M is the number of lattice stages for the top and bottom blocks, such that the delay D = M or (M+1).

For the 5-tap hybrid lattice filter, the output is given by (4.31) for two 2-stage lattice block, that is, M=2. The center tap is fixed to 2. Thus, $f_2(n)$ and $g_2(n)$ can be expressed in the form

$$f_2(n) = x(n) + K_{T1}(1 + K_{T2})x(n-1) + K_{T2}x(n-1)$$
$$g_2(n) = K_{B2}x(n) + K_{B1}(1 + K_{B2})x(n-1) + x(n-2)$$

Therefore,

$$\begin{aligned} y(n) &= f_2(n-2) + g_2(n) \\ &= K_{B2}x(n) + K_{B1}(1 + K_{B2})x(n-1) + 2x(n-2) + K_{T1}(1 + K_{T2})x(n-3) + K_{T2}x(n-4) \\ &= c_4(0)x(n) + c_4(1)x(n-1) + c_4(2)x(n-2) + c_4(3)x(n-3) + c_4(4)x(n-4) \end{aligned} \tag{4.32}$$

It can be seen from (4.32) that

$$\begin{aligned} c_4(0) &= K_{B2} \\ c_4(4) &= K_{T2} \\ c_4(1) &= K_{B1}(1 + K_{B2}) \\ c_4(3) &= K_{T1}(1 + K_{T2}) \end{aligned} \tag{4.33}$$

Considering the set of transversal coefficients from (4.20) and applying the relationship from (4.18) and (4.19), we obtain the following equivalent reflection coefficients for the equivalent five-tap hybrid FIR lattice filter:

$$\begin{aligned} K_{T1} &= K_{B1} = -0.465 \\ K_{B2} &= K_{B2} = 0.125 \end{aligned} \tag{4.34}$$

Using the K values in MATLAB simulations to perform PR4 equalization on the sequence of pseudo-random binary data generated to simulate the 6-bit ADC output, we find that the hybrid lattice architecture provides equivalent performance to that of the regular transversal filter. See Figure 4-14.

Further simulations to observe the mapping between transversal and reflection coefficients showed nicely bounded ranges of the equivalent reflection coefficients. The MATLAB program randomly picked $c_4(0)$ in the range of 0 to 0.5 and $c_4(1)$ in the range of −1 to 0, and transforms to lattice parameters K_1 and K_2. The plot is shown in Figure

4-15. Higher order filter implementations for typical transversal coefficient range are also verified for the equivalent lattice reflection parameters. Figures 4-16 and 4-17 show MATLAB simulation plots for 3-stage and 4-stage lattice blocks corresponding to 7-tap and 9-tap hybrid FIR lattice filter designs.

This chapter has covered the approach to the new lattice-based FIR filter architecture based on the requirements for PRML target matching.

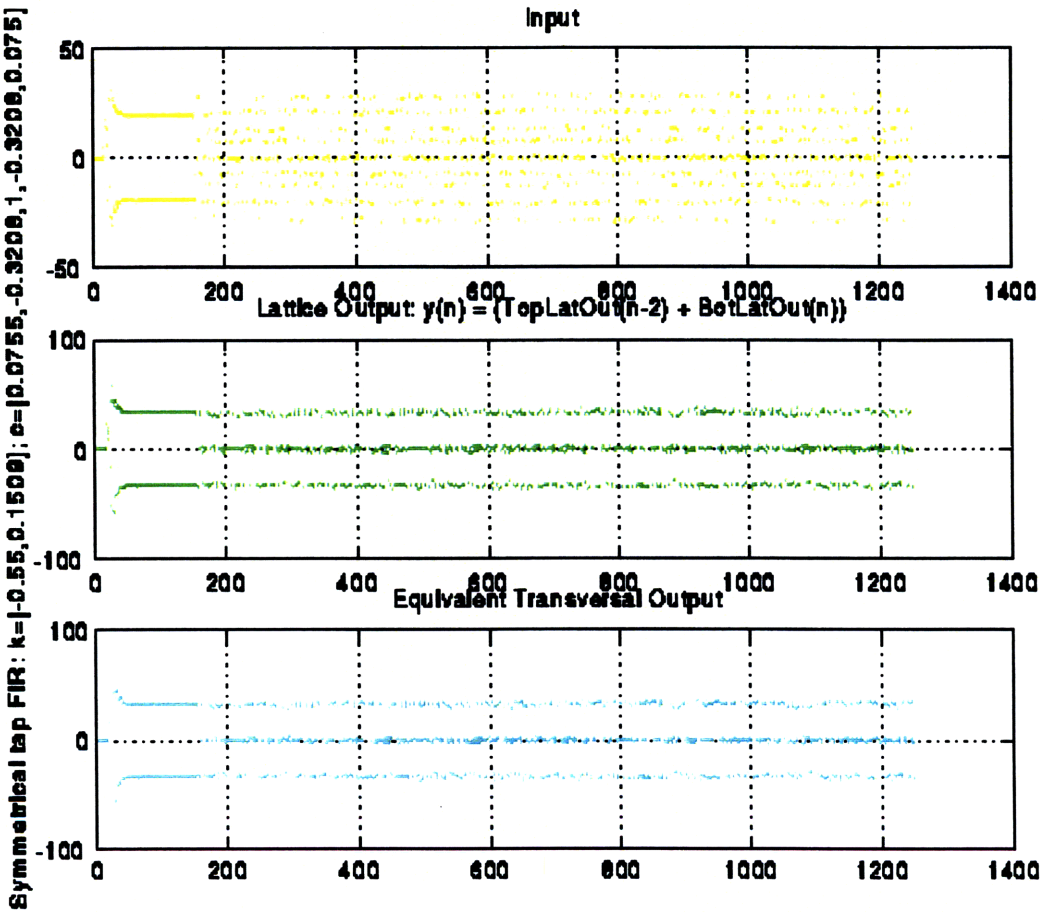


Figure 4-14: Equalized output of hybrid FIR lattice filter

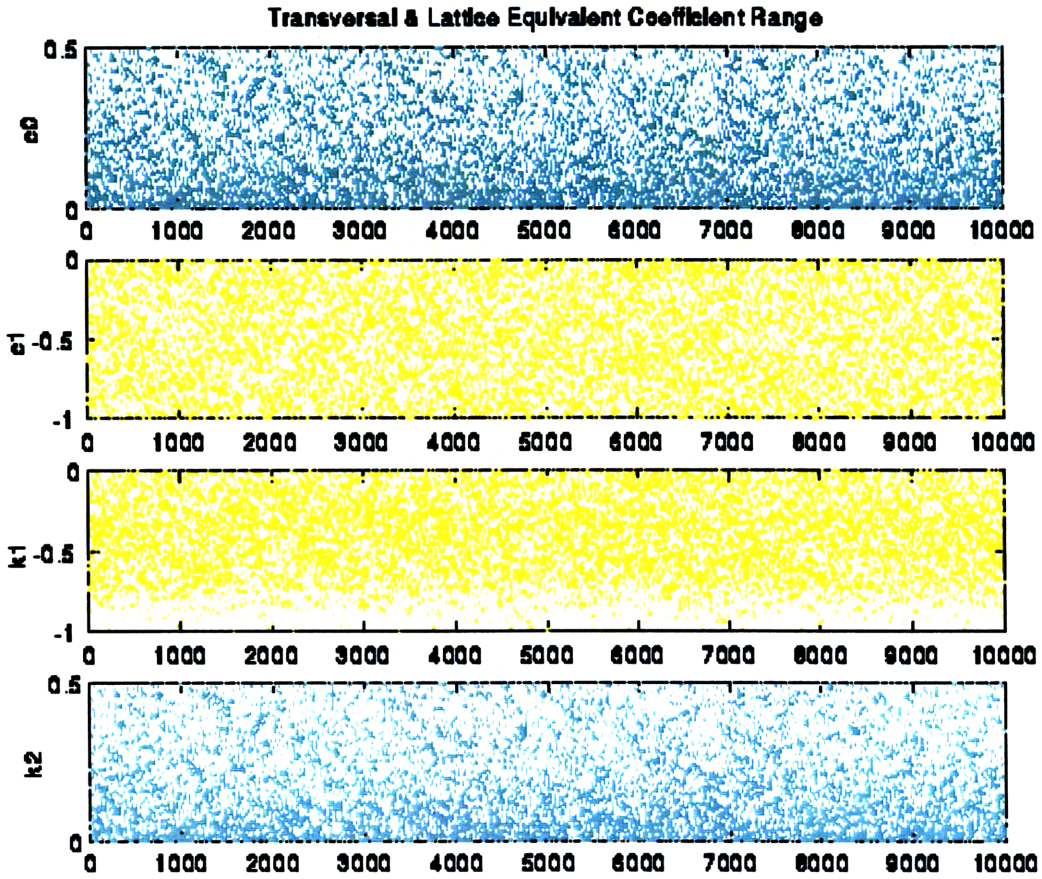


Figure 4-15: Transversal and reflection coefficients range

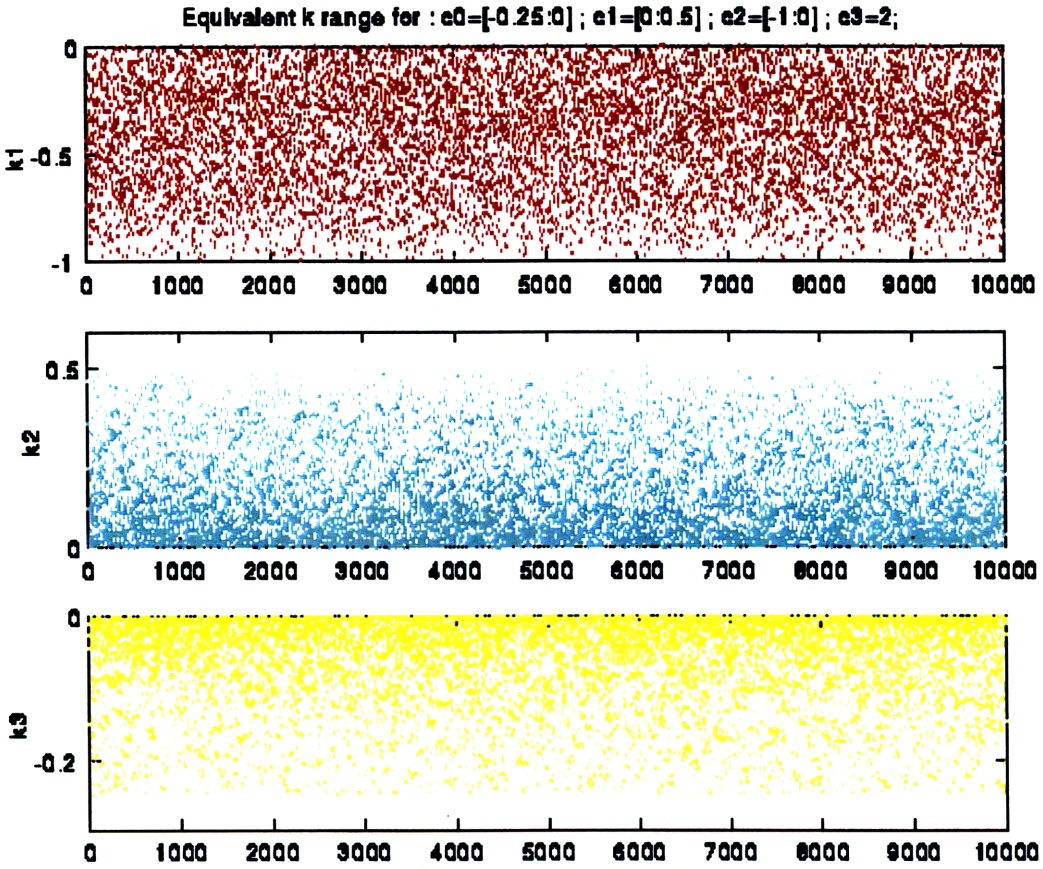


Figure 4-16: Reflection coefficients range for 3-stage lattice block

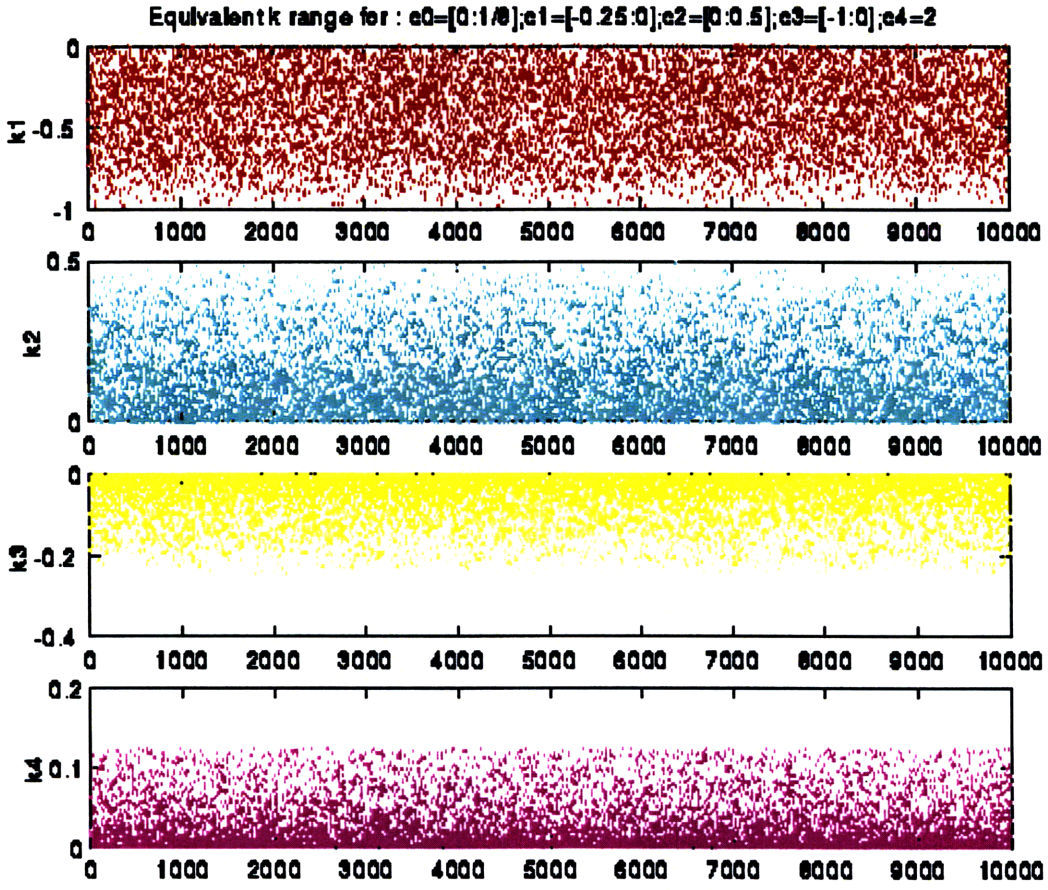


Figure 4-17: Reflection coefficients range for 4-stage lattice block

Chapter 5

Design and Implementation

When a digital filter transfer function is implemented using a digital system, it invariably involves quantization of signals and coefficients. As a result, the overall input-output behavior is not ideal. In this chapter, the effect of the quantization of the FIR coefficients on the channel equalization will be considered. As expected, quantization will alter the FIR filter response, which in turn degrades the output MSE due to mis-equalization. In general, the effect of quantization noise on the FIR filter response will be more significant when the zeros are more closely clustered together.

5.1 Fixed-Point Representation of Numbers

The representation of numbers in fixed-point format is a generalization of the familiar decimal representation of a number as a string of digits with a decimal point. In this notation the digits to the left of the decimal point represent the integer part of the number, and the digits to the right of the decimal point represent the fractional part of the number. Thus a real number X can be represented as

$$\begin{aligned} X &= (b_{-A}, ..., b_{-1}, b_0, b_1, ..., b_B) \\ &= \sum_{i=-A}^{i=B} b_i r^{-i}, \qquad 0 \le b_i \le (r-1) \end{aligned} \tag{5.1}$$

where b_i represents the digit, r is the radix or base, A is the number of integer digits, and B is the number of fractional digits. We adopt the two's-complement format for the representation of signed binary fractions. In this format a negative number is represented by forming the two's complement of the corresponding positive number. In other words, the negative number is obtained by subtracting the positive number from 2.0. Basically, the two's complement is formed by complementing the positive number and adding one LSB. Thus

$$X_{2C} = 1.b_1 b_2 \ldots \quad b_B + 00\ldots01 \qquad (X < 0) \tag{5.2}$$

where + represents modulo-2 addition that ignores any carry generated from the sign bit. For example, the number -3/8 is simply obtained by complementing 0011 (3/8) to obtain 1100 and then adding 0001. This yields 1101, which represents -3/8 in two's complement.

5.2 Digital Implementation

In a digital implementation of an FIR filter, there are essentially two sources of quantization error to be considered: ADC conversion and finite word-length arithmetic.

The process of converting a continuous-time (analog) signal to a digital sequence that can be processed by a digital system requires that we quantize the sampled values to a finite number of levels and represent each level by a number of bits. The ADC in the read channel is a fully parallel 6 bit flash converter using 62 identical comparators. The A/D output format is 2's complement with the maximum positive value of +31 and the maximum negative value –31. For our present discussion, we assume a quantization process with a uniform step size $\frac{1}{32}$ and a set of 64 quantizing levels positioned at $0, \pm\frac{1}{32}, \pm\frac{2}{32}, \ldots$.

In a digital machine, a finite word length is used to store the result of internal arithmetic calculations. Assuming that no overflow takes place during the course of computation, additions do not introduce error if fixed-point arithmetic is used, whereas

each multiplication introduces an error after the product is quantized. The statistical characterization of finite word-length arithmetic errors may be quite different from that of analog-to-digital conversion errors. Finite word-length arithmetic errors may have a nonzero mean, which results from either rounding off or truncating the output of a multiplier so as to match the prescribed word length.

The presence of finite word-length arithmetic raises concern in the implementation of the filter, particularly when the tap weights of the filter are updated on a continuous basis. The digital version of the filter exhibits a specific response or propagation to such errors, causing its performance to deviate from the ideal, that is infinite-precision, form of the filter. A filter of practical value is said to be numerically stable if the use of finite-precision arithmetic results in deviations from the infinite-precision form of the filter that are bounded. It is important to note that if a filter is numerically unstable, then increasing the number of bits used in digital implementation f the filter will not change the stability condition of that implementation.

Another issue that requires our attention in a digital FIR filter implementation is that of numerical accuracy. Unlike numerical stability, however, the number of bits used to implement the internal calculations of the filter determines the numerical accuracy of an adaptive filter. The larger the number of bits used, the smaller the deviation from ideal performance, and the more accurate would be the digital implementation of the filter.

5.3 Bit Analysis

The main characteristic of digital arithmetic is the limited (usually fixed) number of digits used to represent numbers. This constraint leads to finite numerical precision in computations, which leads to round-off errors and nonlinear effects in the performance of filters.

In general, the performance of the lattice filter should be comparable to transversal filter. However, the hybrid lattice filter structure is more robust in fixed-point implementations. The idea is to simplify the multiplier (by making it smaller) by

exploiting the numerical robustness. For instance, for a 2-stage top and bottom hybrid lattice structure, the output is

$$\begin{aligned} y(n) &= K_{B2}x(n) + K_{B1}(1+K_{B2})x(n-1) + 2x(n-2) + K_{T1}(1+K_{T2})x(n-3) + K_{T2}x(n-4) \\ &= c_4(0)x(n) + c_4(1)x(n-1) + c_4(2)x(n-2) + c_4(3)x(n-3) + c_4(4)x(n-4) \end{aligned} \tag{5.3}$$

The combination of k_i's product gives the equivalent transversal coefficient c_i's. Smaller number of bits can therefore be tolerated in lattice implementation because

$$\begin{aligned} & 0.1\,(binary) \times 0.1(binary) = 0.01(binary) \\ or \quad & (decimal) \quad 0.5 \times 0.5 = 0.25 \end{aligned} \tag{5.4}$$

Even if we restrict the size of two to be single bit, their product has two bits. But the product is not "realized" in the lattice implementation. It simply results from the use of the lattice structure whereby the equivalent transversal coefficients have a lot more bits than what the transversal filter implementation would have had, had we used the same size multiplier in it.

We investigate the performance trade-offs in maximum error, root mean-square error and offset for different combinations of number of bits of precision of the reflection coefficients. For the purpose of our study, we have picked the symmetrical transversal filter taps as per equation (4.20). Hence, the two sets of reflection coefficients for the top and bottom 2-stage lattice blocks are simply $\{K_1, K_2\}$. From (4.34), we get $K_1 = K_{T1} = K_{B1} = -0.465$ and $K_2 = K_{B2} = K_{B2} = 0.125$. Hence, each of the five equivalent filter coefficients obtained after the sum operation in Figure 4-14 is user programmable. Asymmetrical transfer functions would simply need different sets of top and bottom reflection coefficients.

A sequence of pseudo-random binary data is generated and run in MATLAB according to the block diagram shown in Figure 5-1. The additional equalization for fine shaping of the incoming read signal to PR4 is provided by the hybrid of 2-stage lattice. K_2 can be set to a range from $-\frac{1}{2}$ to $\frac{1}{2}$. The other lattice reflection coefficient K_1 can be set to a range from –1 to 1. The bit analysis is carried out in MATLAB for the equivalent 5-tap programmable hybrid digital lattice filter for PR4 equalization, with (1+D) conversion to EPR4.

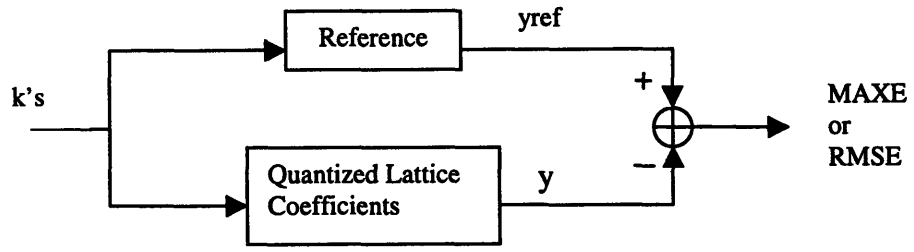


Figure 5-1: Block diagram for bit analysis in MATLAB

$$MAXE = norm(y - yref)$$
$$RMSE = \frac{norm(y - yref)}{\sqrt{length(y)}} \tag{5.5}$$

The results are shown in Figure 5-2 for PR4 target. Note that only integer values of the number of bit representation of K_2 are valid. As we discussed earlier, we observe that as we increase the number of bits used for the reflection coefficients from 1 to 5 (excluding sign bit), the smaller is the deviation from ideal performance.

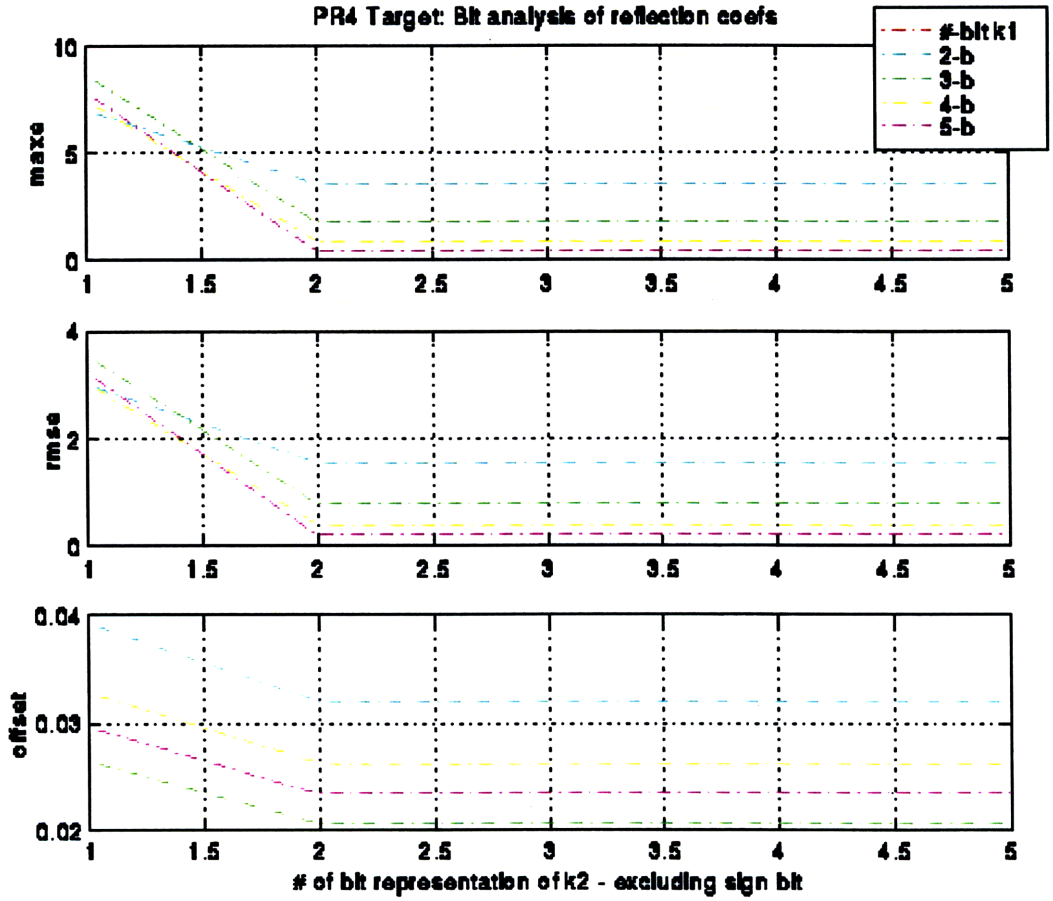


Figure 5-2: Bit analysis of reflection coefficients K_1, K_2 for PR4 target.

5.4 FIR Bit Accumulation

In the realization of FIR filter, the word length of the register provided to store the coefficients limits the accuracy with which the reflection coefficients can be specified. Since the coefficients used in implementing a given filter are not exact, the zeros of the system function will be different from the desired zeros. Consequently, we obtain a filter having a frequency response that is different from the frequency response of the filter with unquantized coefficients. In addition, when performing computations such as multiplication with fixed-point arithmetic, we are faced with the problem of quantizing a number from a given level of precision to a level of lower precision.

We first need to study the bit accumulation for a 2-stage lattice blocks to prevent any signal overflow. We picked 5-bit K_2 in order to start with comparable lsb of the outer tap coefficients for both lattice and transversal implementations. From our bit analysis in the previous section, we observed that a bit representation ranging from 2 to 5 bits (excluding sign bit) for K_1 could be considered. We also know we can obtain products not "realized" in the lattice implementation. Since the resulting equivalent transversal coefficients end up having a lot more bits than what the transversal filter would have had with the same size multiplier, we can chose 3 bits representation for K_1.

The lattice coefficients for first and second stages are independent and each has a resolution of $\frac{1}{4}$ and $\frac{1}{32}$ respectively. Table 5.1 gives a summary of the reflection coefficients specifications for our design. Each stage forward and backward node sample is multiplied by the corresponding coefficient.

Table 5.1: Reflection coefficients specification

Coefficient Name	No. of bits per coefficient	Exact coefficient range
TK_1, BK_1	3	-3/4 3/4
TK_2, BK_2	5	-15/32 15/32

Table 5.2 shows in details the bit accumulation for the hybrid lattice filter structure given in Figure 5-3. The input data is represented as 6-bit buses DTI. Since there are two identical 2-stage lattice blocks to implement an equivalent symmetrical five taps FIR transversal filter, it suffices to go through one signal path to figure out the maximum values at each node and allocate wide enough buses that will prevent any overflow.

Table 5.2: Lattice FIR bit accumulation

Signal	Resolution	Max. Values	exclude sign bit	No. of bits
DTI or *X*	1	31		6-b
sla(*TF0*,2)	1/4	(31*4)=124	$<2^7$	8-b
$TG0 * TK_1$	1/4	(31*3)=93	$<2^7$	8-b
TF1	1/4	(124+93)=217	$<2^8$	9-b
sla(*TF1*,5)	1/128	(217*32)=6944	$<2^{13}$	14-b
$TG1 * TK_2$	1/128	(217*15)=3255	$<2^{12}$	13-b
TF2	1/128	(6944+3255)=10199	$<2^{14}$	15-b
DTO or *Y*	1/128	(10199*2)=20398	$<2^{15}$	16-b

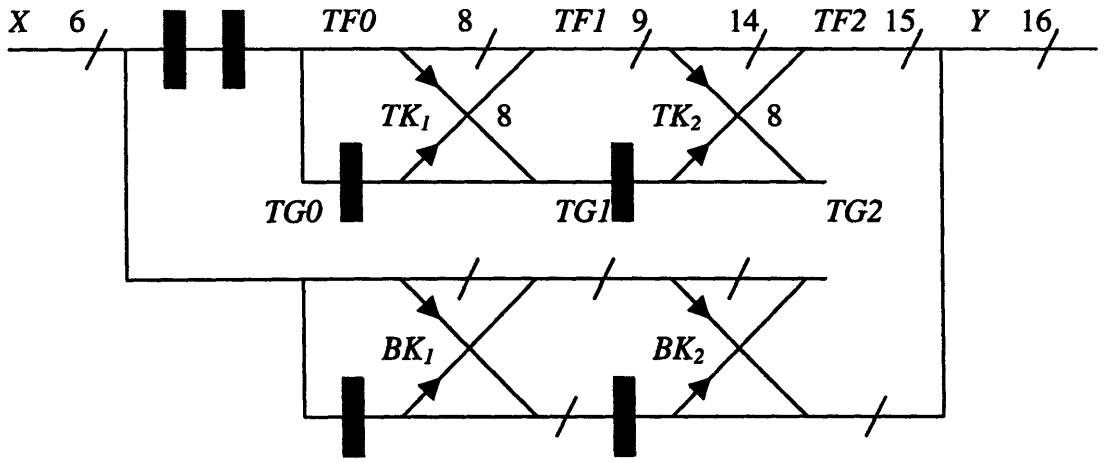


Figure 5-3: Hybrid FIR lattice filter structure

Without any hardware or performance optimization, this first pass filter design yields a maximum error of 1.544 and a root mean square error of 3.582 for 3-bit and 5-bit reflection coefficients representation for stage 1 and stage 2 respectively. Table 5.2 indicates higher coefficient resolution as we proceed along the signal flow, which is consistent with the numerical robustness of the lattice filter structure as we analyzed in section 5.3. Figure 5-4 shows the equalized FIR output of the sample data for K_1 and K_2 set to –0.465 and 0.125 respectively. The histogram of the FIR output and RMSE plot both indicate good equalization performance. Under the given simulation environment and randomly generated input data, we observe the equalized lattice filter output values are well centered at zero threshold for signal slicing.

Furthermore, we ran MATLAB simulations to perform FIR equalization with the optimum reflection coefficients without any boosting from the CPF. Figure 5-5 gives the surface plot of the RMSE versus reflection coefficients, K's values. The optimum RMSE value is 0.01 or –39.63 dB with K_1 and K_2 set to –0.75 and 0.1562 respectively.

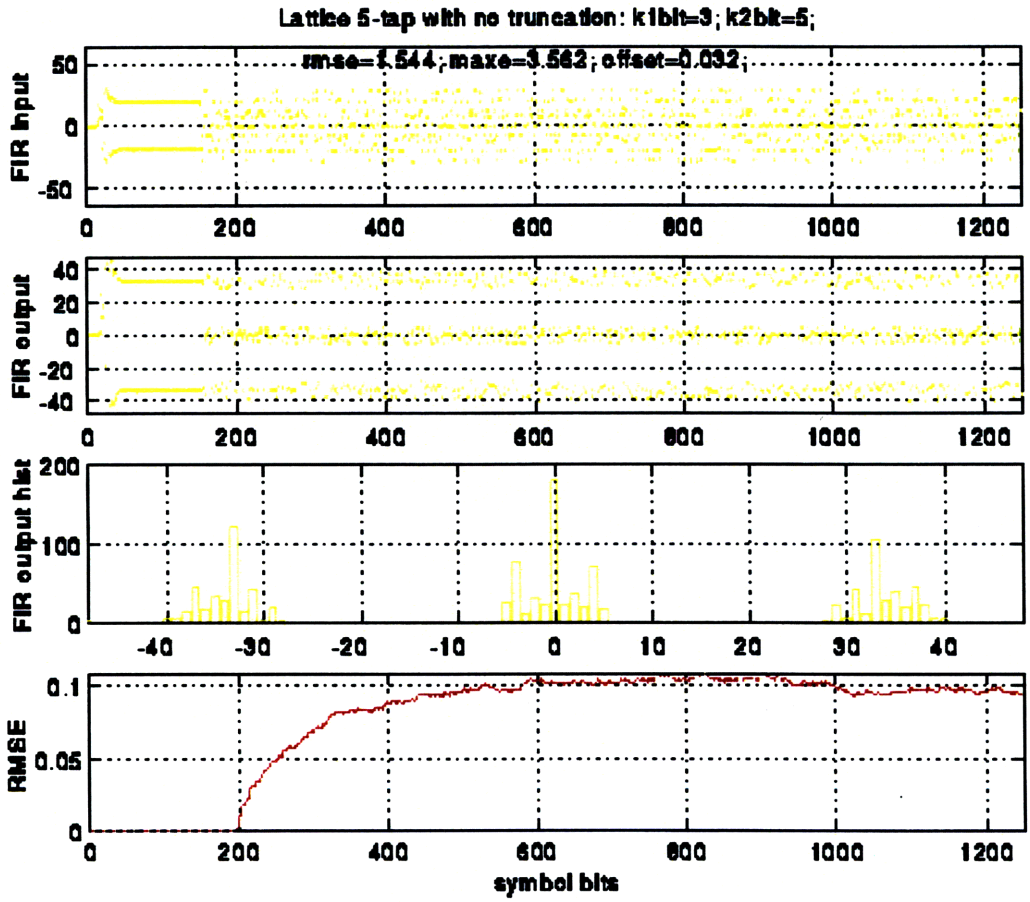


Figure 5-4: Equalized lattice FIR output with 3-bit K_1 and 5-bit K_2.

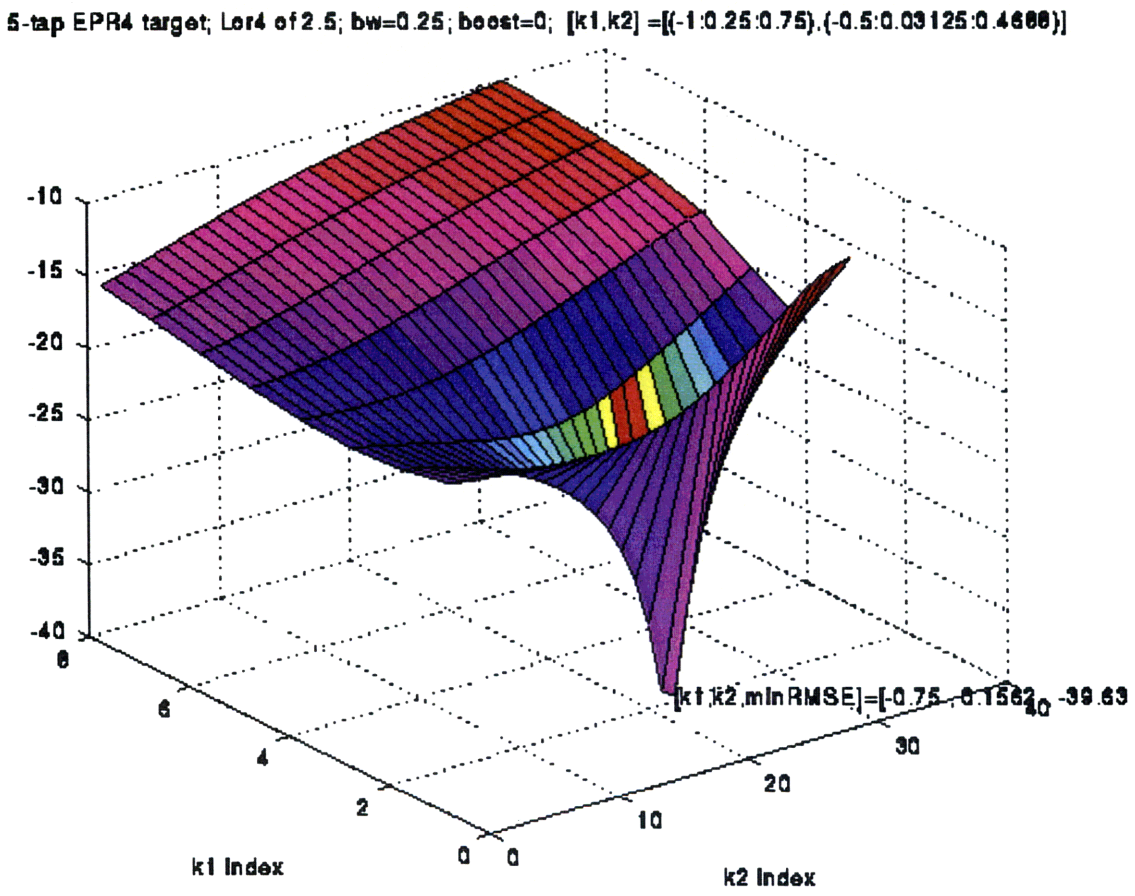


Figure 5-5: Surface plot of 5-tap hybrid lattice filter with 3-bit K_1 and 5-bit K_2.

Note: K_1 range is –1 to 0.75 with a resolution of 0.25
and K_2 range is –0.5 to 0.4688 with a resolution of 0.03125
Minimum RMSE of –39.6dB is obtained for K_1=-0.75 and K_2=0.1562

5.5 Design Optimization

In the hybrid lattice FIR the signal flow is from left to right as shown in Figure 5-3. Let us first establish some signal naming conventions. We will refer to the top lattice block as the T-block and the bottom one as the B-block. The letter T or B also precedes the signal names and nodes. Basically, the T-block and B-block are identical except for the extra two delays at the input of the T-block.

We introduced certain techniques that drastically improved the lattice filter performance both in terms of hardware cost as well as performance. Let's consider the path from the input to the output of the T-block. After passing through the two 6-bit register elements, the 6-bit incoming data propagates immediately along the top row of horizontal branches, with unity gain. All the other paths from the input to the output pass through at least one delay element. For the very first stage, the 6-bit data is delayed by the 6-bit register or one-delay element, and multiplied by the 3-bit coefficient TK_1. The result of this individual multiplication can be an 8-bit number. The signal *TF0* is arithmetically shifted left by 2 in order to keep a common denominator. The sum for the *TF1* signal is done providing 9 bits of range. It is important that the sum does not overflow the range. Next, the second stage requires a 9-bit register for the lattice delay at the bottom horizontal row. The 9-bit signal is multiplied by the 5-bit coefficient TK_2. This resulting product is truncated from 13-bit down to 9-bit. Signal *TF1* is arithmetically left-shifted by 1. The sum at the output of the second stage *TF2* is done providing only 10 bits of range. The final sum of the T-block and B-block is truncated to the 9 most significant bits. The samples out of the FIR are expected to be equalized to a PR4 waveform with the output values clustered around +32, 0, and –32, corresponding to the –1, 0, and +1 of the PR4 sample set.

In this chapter, we evaluated the full implementation of the hybrid FIR lattice filter for read channel application in MATLAB. The performance of the optimized lattice design is indeed comparable to the transversal implementation (Table 5.3).

Table 5.3: Optimization results

Filter Structure	MAXE	RMSE
Optimized Transversal	0.9375	0.4103
Non-optimized Lattice	0.9766	0.5383
Optimized Lattice	0.7500	0.3062

Figure 5-6 shows PR4 equalized output of optimized FIR lattice filter design.

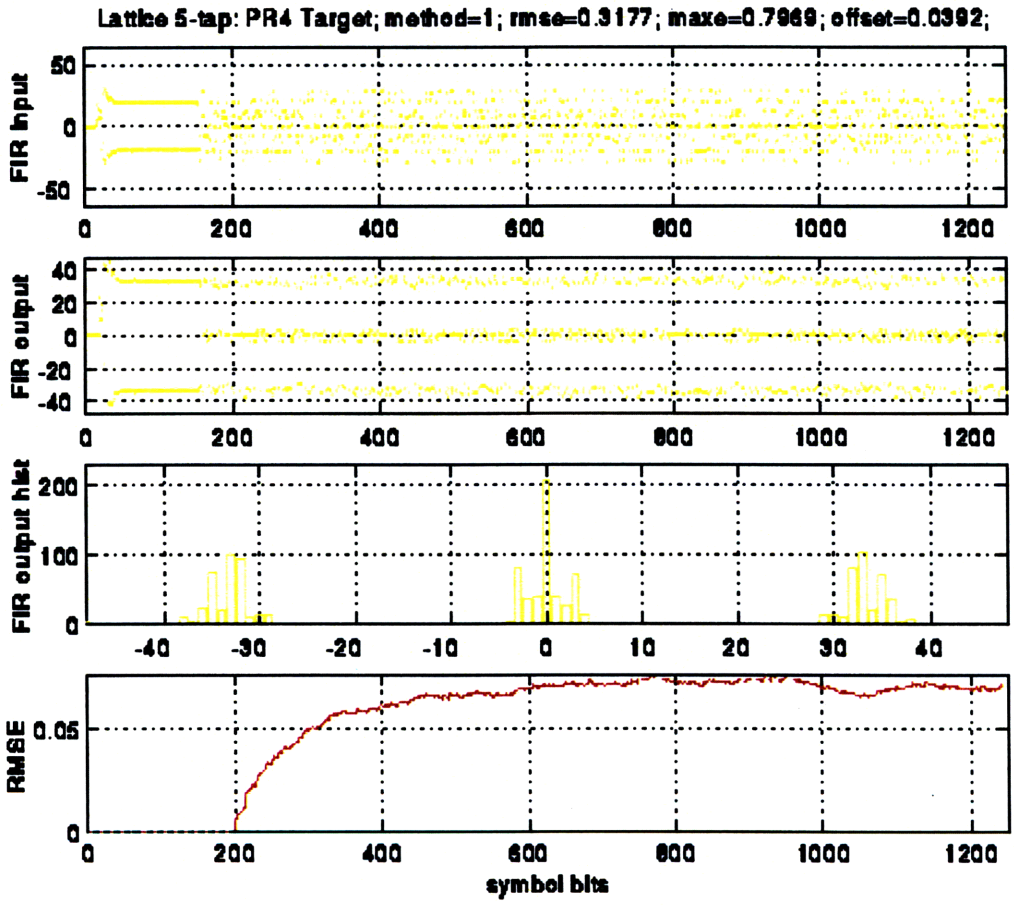


Figure 5-6: PR4 equalized output of optimized lattice filter

Chapter 6

Chip Design and Synthesis

The key issue in the design of the hybrid lattice filter was to obtain a robust FIR filter that not only performs the desired partial response equalization without LPF boost but also provides improved performance based on the mean square error approach. Starting with the well-known transversal filter implementation, we studied the architecture and system modeling of the digital FIR filter in the magnetic read channel. We then performed extensive MATLAB behavioral and structural simulations to fully evaluate the lattice structure. Having optimized the filter parameters for equalization, we are now ready to implement the hybrid lattice architecture using Synopsys' VHDL System Simulator ™ (VSS) for system specification, design refinement, and gate level simulation.

We initially specified the system at a high level of abstraction using VHDL to describe the entire design. We used VSS to analyze the system design and explore alternative design ideas to arrive at the most efficient architecture for the lattice FIR filter. During this stage, we validated the system concept by running a number of simulation cycles in both MATLAB and HDL. The final architecture proposed for the new hybrid lattice filter is shown in Figure 6-1.

The input signal *"/dti"* is defined as a standard logic vector of 6 bits coming out of the ADC block. The signals use a clock period of 5ns. The reflection coefficients are programmed to "110" for K_1 and "00100" for K_2. We used the VHDL interface to create a template for a VHDL testbench to test the design. Figure 6-2 shows the PR4 equalized

output. The timing diagram of the equalized 9-bit output signal *"/dto"* as well as corresponding integer values *"/dto_0"* is shown in Figure 6-3. In addition, we checked the equalized output sample data obtained in VHDL with our previous MATLAB simulations, and confirmed matching results. Figure 6-4 shows the top-level architecture for the lattice design.

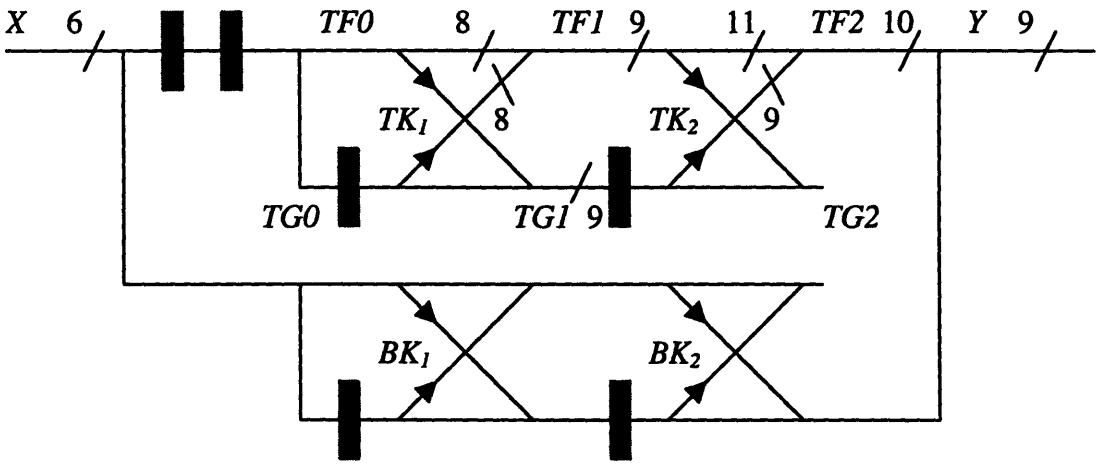


Figure 6-1: Hybrid FIR lattice filter structure for bit analysis

Once we have validated the system concept, we refined the components of the system for synthesis. We partitioned the system into VHDL components and sub-systems, and used VSS to help analyze the design as we develop it into synthesizable models. It is crucial to verify the synthesizable VHDL components in the context of the entire system. This serves as an important checkpoint to confirm the system functionality before going through the synthesis process.

After we were satisfied that the design is correct, we synthesized it to an efficient gate-level implementation. The simulator is tightly integrated with high quality synthesis. At the gate-level, we simulate the design with full timing information, including post-layout timing for all nets. For silicon sign-off, the simulator used the TI TSC4000 library. The operational environment is for a 3.0-3.6V supply at a junction temperature of 0-125°C. We used the 33C12 process of 0.6μm metal pitch and 0.504μm length of device at 3.3V.

The FIR lattice filter runs at a speed of 54 MHz using the standard library cells with a power dissipation of around 82mW worst case. This first attempt to implement the FIR filter using the lattice structure clearly does not meet the higher speed requirement our particular read channel application. But we must again emphasize that the main goal here is to achieve a more compact filter implementation by exploiting the numerical robustness of the lattice structure. Indeed, the total cell area of the hybrid lattice is 2289.25 equivalent NAND gates, almost half the size of the digital transversal FIR filter implementation. The equivalent NAND gate area is $110\mu m^2$. Hence, the drawn area for the lattice FIR design is around $0.25mm^2$. With the 33C12 process we obtain an actual chip size of $0.212mm^2$. Finally, the layout is done using Aquarius for auto place&route (Figure 6-5).

Note that the design can still be further optimized for minimum area and power. The speed performance can be improved by defining the critical path and determine the bottleneck cells to see what benefits can be obtained by using customized cells. Furthermore, the broad modeling support in Synopsis enables full system simulation from concept validation to silicon sign-off in a consistent simulation environment throughout the design cycle. Using Synopsys, there is great flexibility to achieve higher performance and accuracy by remapping the library for reoptimization.

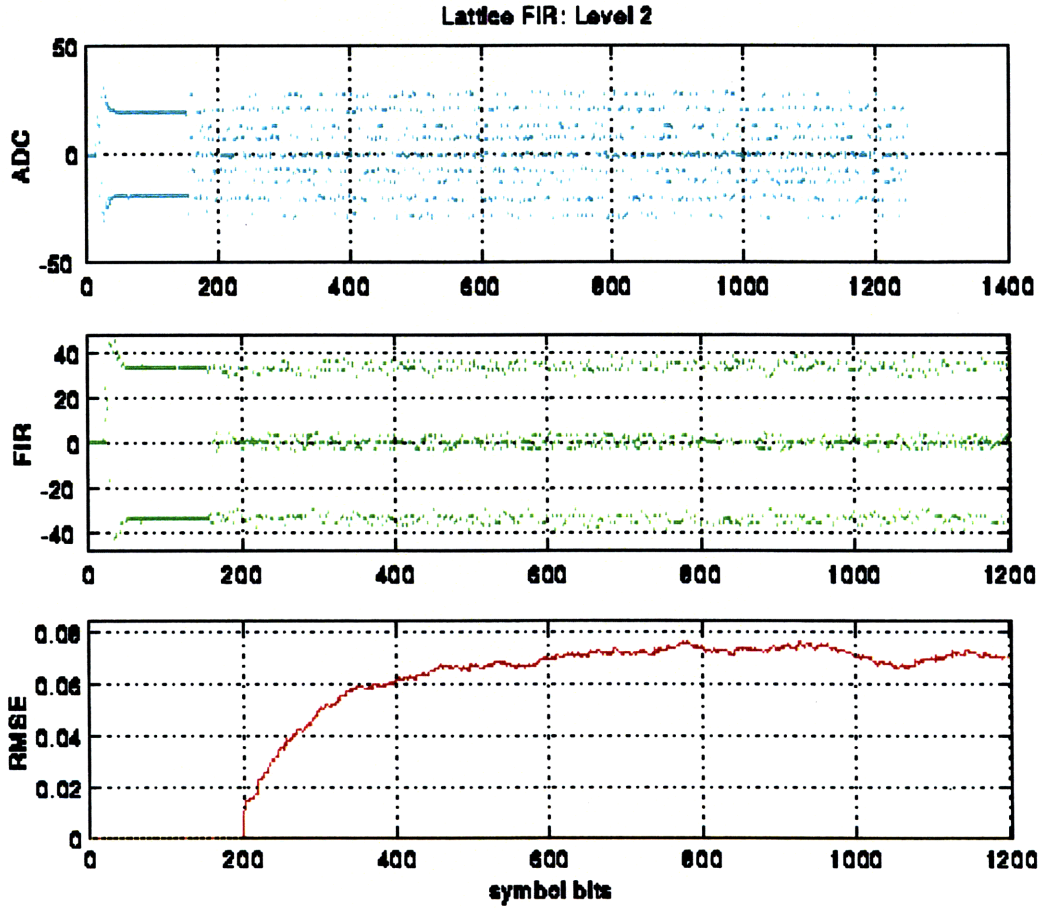


Figure 6-2: Equalized output

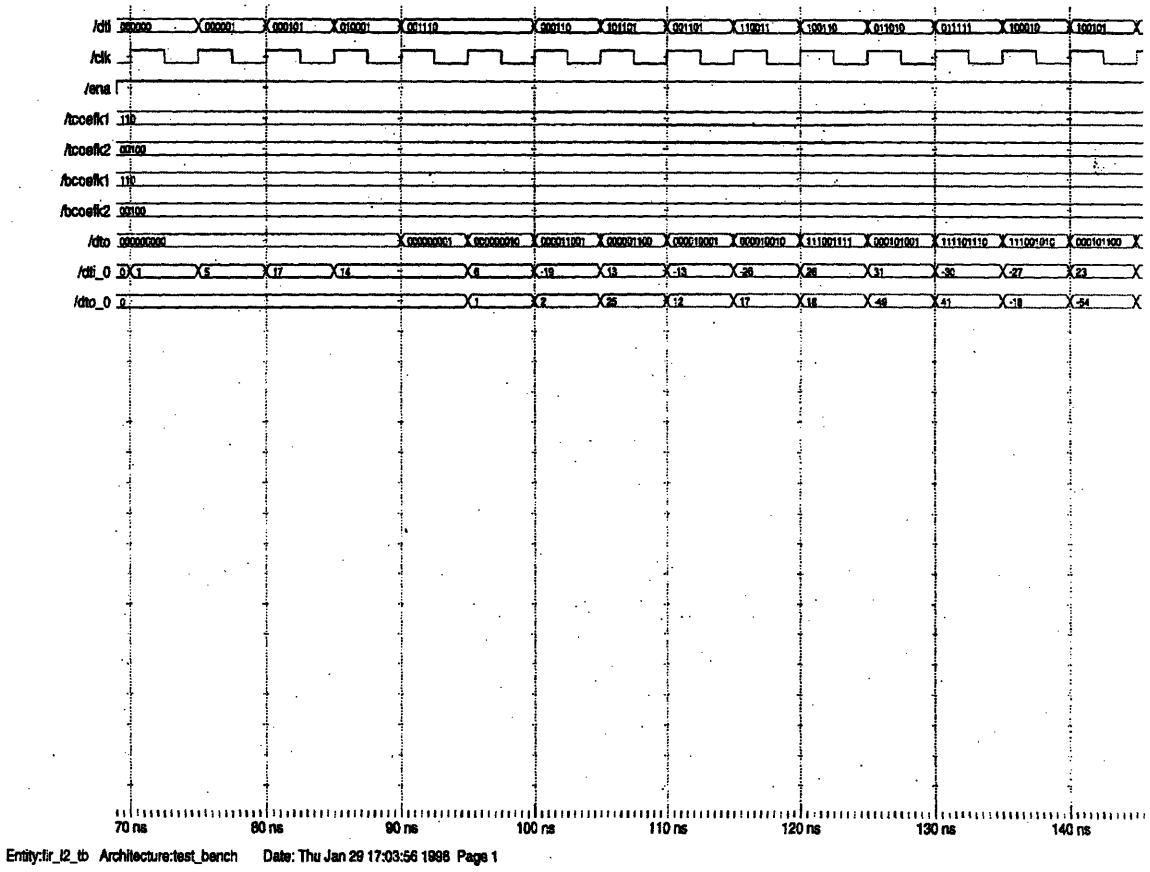


Figure 6-3: Timing diagram

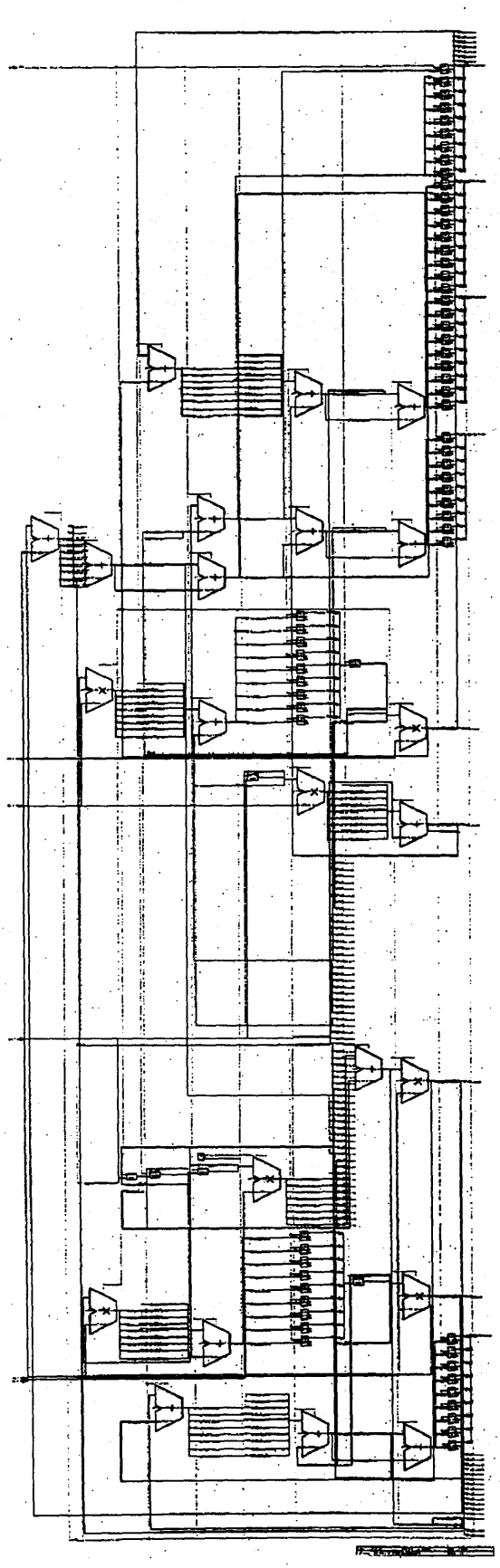


Figure 6-4: Top level block diagram of lattice FIR filter

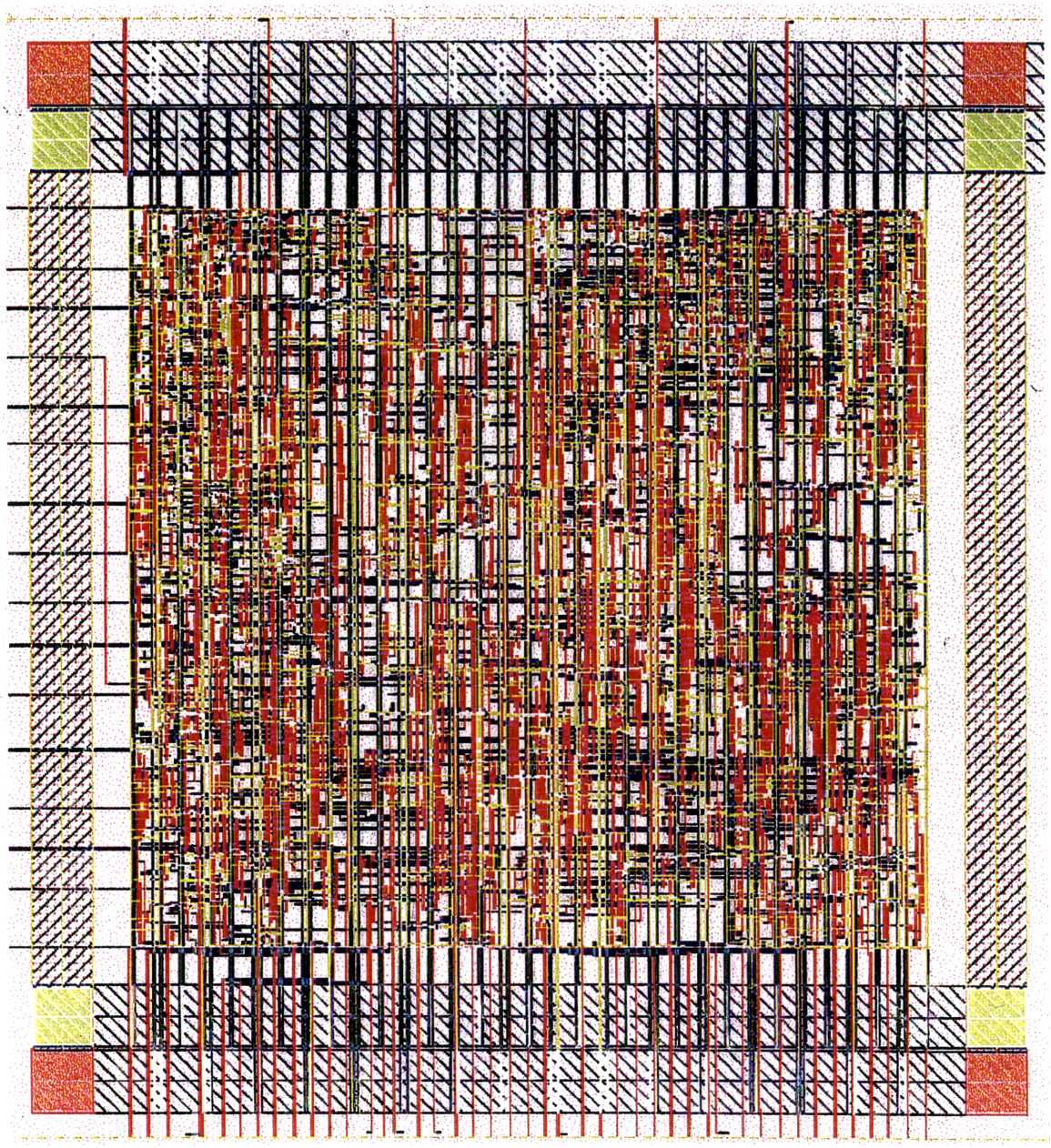


Figure 6-5: Chip layout using Aquarius auto place&route

Chapter 7

Conclusions and Future Work

7.1 Summary and Conclusions

In this thesis, we explored the use of the lattice structure for a finite-impulse-response FIR filter design in the partial-response maximum-likelihood (PRML) read/write channel for magnetic storage applications. The task of a novel lattice-based filter architecture was chosen because other filter architectures employing distributed arithmetic and the more traditional transpose structure have already been extensively studied in literature. The 5-tap FIR is most optimal for Partial Response Class 4 (PR4). For comparison purposes, we opted for a 5-tap order filter and used the same simulation environment and process design flow for developing the new lattice architecture.

Chapter 2 gave an overview of the hard disk drive (HDD) system and described the digital PRML read channel requirements as opposed to the classic transmission channel. Increased data rate and higher area density are causing significant degradation in the signal-to-noise ration (SNR). Hence, the motivations to introduce PRML techniques to compensate for this loss.

The mean-square-error (MSE) design approach is introduced in Chapter 3. For comparison purposes, we started with the discussion of the more conventional combined CTF/FIR configuration used for equalization. The various trade-offs of boost parameter and tap order were investigated to gain a better understanding of the FIR filter role for

equalization purposes. This chapter described an Enhanced Partial Response Class 4 (EPR4) channel that uses a Partial Response Class 4 (PR4) front-end followed by a (1+D) transform. MATLAB simulations showed that 5-tap FIR is the most optimal for PR4.

Given the already well-known traditional direct or transpose filter architectures that require explicit multiplication and addition in the disk drive read channel application, we attempted to propose a novel FIR filter design that would exploit the properties of the lattice structure. It is important to note that whilst the lattice structures have been studied for different pole-zero digital filters in speech, signal processing and VLSI applications, only one single all-zero lattice filter structure has been proposed by Itakura and Saito for speech analysis and synthesis. However, the lattice structure does promise to be useful wherever transversal, predictive, or finite impulse response filters are used in an adaptive manner. We introduced the basics of the lattice structure as well as the transversal to reflection and vice versa coefficients transformation in Chapter 4. We studied lattice FIR candidates in MATLAB before deciding on a hybrid model that meets the requirements for PRML target matching.

After choosing the lattice architecture, the next step was to fully evaluate the proposed filter model in MATLAB and optimize for PR4 equalization in Chapter 5. With an understanding of the lattice properties, the challenge was to optimize for partial-response equalization by applying certain techniques to improve the MSE function. Once the filter parameters were defined, the next step in Chapter 6 was to implement the hybrid lattice architecture using the Synopsys' VHDL System Simulator™ (VSS) for the next stages of design including system specification, design refinement, and gate-level verification. We finally used Aquarius for auto place&route.

This research demonstrated the implementation of the hybrid digital FIR lattice filter using standard multipliers and adders. By exploiting the binary representation of the filter coefficients, essential complexity reduction is achieved. A lattice filter requiring fewer overall gates was built by reducing the number of bit representation of the coefficients and using smaller data buses. Thus, we reduced the cell area of the hybrid lattice filter by a factor of two compared to the transversal filter implementation. Although the speed performance is still to be desired, there is room for improvement by

considering pipelined structures, some improved architecture and cell optimization especially in the critical path.

7.2 Future Work

So far, the transversal FIR filters have been dominant due to such factors as relative simplicity of implementation and well-known principles of operation. But as increasing efforts are being directed towards more and more complex detection schemes, there arises the need for more robust architectures that would meet the driving market for low-cost, high-performance, high-capacity mass storage.

This thesis introduced the lattice structure within the read channel application and focused on obtaining a more robust digital FIR filter. This is nothing but the tip of the iceberg. There is a lot more work to be done related to the timing control blocks as well as adaptation algorithm schemes. Currently, the essential task of the READ timing loop to allow the channel to acquire the data correctly is done by passing the sampled output signal of the FIR block on to the ERROR block. We can exploit the lattice structure and use the orthogonal backward prediction errors to perform an earlier and continuous adjustment to the VCO's frequency and phase during the pulse shaping by the lattice FIR block. Thus, achieving earlier error estimation in the PLL and AGC loop can lead to reduced latency.

It will be interesting to perform experiments using the one-multiplier lattice structure proposed by Makhoul [36]. Where multiplication is expensive computationally or for hardware implementation, the one-multiplier form can result in substantial savings because of the reduced number of multiplies. The present two applications of all-zero lattice digital filters proposed in literature were for adaptive linear prediction and adaptive Wiener filtering in the form of an efficient fast start-up equalizer. But this different lattice form with one-multiplier promises to be useful especially combined with the orthogonalization and decoupling properties of the successive stages in the lattice filtering of signals.

Another possible extension of this thesis is to apply the suggested lattice structure for higher tap order filters. Just as the 5-tap order provides best performance for the transversal implementation, the optimum equalization will be determined for the lattice structure design. By incorporating new techniques of error correction coding and timing calculation together with the inherent robustness of the lattice filter in fixed-point implementations, we may consider increasing beyond ten taps without incurring substantial latency in the timing loop. Hence, an attempt at lattice FIR only equalization can approach the performance of a system whose CTF provides boost and optimization.

Bibliography

[1] Rigid disk drive news at "http://www.disktrend.com/newsrig.htm"

[2] IBM website at "http://www.storage.ibm.com/storage/technolo/grochows/grocho01.htm"

[3] A. G. Gray and J. d. Markel, "A normalized digital filter structure," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-23, pp. 268-277, June 1975.

[4] R. E. Crochiere and A. Y. Oppenheim, "Analysis of linear digital networks," *Proc. IEEE*, vol. 63, pp. 581-595, Apr. 1975.

[5] B. Friedlander, "Lattice method for spectrum estimation," *Proc. IEEE*, vol. 70, pp. 990- 1017, Sept. 1982.

[6] J. D. Markel and A. H. Gray, "Roundoff noise characteristics of a class of orthogonal polynomial structures," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-23, pp. 473-486, Oct. 1975.

[7] T. Kailath, "A view of three decades of linear filtering theory," *IEEE Trans. Inform. Theory*, vol. IT-20, pp. 146-181, Mar. 1974.

[8] M. Morf and D. T. Lee, "State-space structure of ladder canonical forms," *Proc. 18th Conf. Decision*, Dec. 1980, pp. 1221-1224.

[9] F. Itakura and S. Saito, "Digital filtering techniques for speech analysis and synthesis," *Proc. 7th Int. cont. Acoust., Budapest*, 1971, Paper 25-C-1, pp. 261-264.

[10] Finn Jorgensen, *The Complete Handbook of Magnetic Recording*.

[11] J. B. Gan, "Pushing the limits of magnetic recording," *IBM Res. Mag.*, vol. 28, no. 2, pp. 7-11, Summer 1990.

[12] H. N. Bertram, "Fundamentals of the magnetic recording process," *Proc. IEEE*, vol. 74, no. 11, pp. 1494-1512, Nov. 1986.

[13] R. D. Cideriyan, F. Dolivo, R. Hermann, W. Hirt, W. Schott, "A PRML System for digital magnetic recording," *IEEE Journal on selected areas in Commun.*, vol. 10, no. 1, Jan. 1992.

[14] G. D. Forney, "The Viterbi algorithm," *Proc. IEEE*, vol. 61, no. 3, pp. 268-278, Mar. 1993.

[15] P. Kabal and S. Pasupathy, "Partial-response signaling," *IEEE Trans. Commun.*, vol. COM-23, no. 9, pp. 921-934, Sept. 1975.

[16] H. K. Thapar and A. Patel, "A class of partial response systems for increasing storage density in magnetic recording," *IEEE Trans. Magnet.*, vol. 23, no. 5, pp. 3666-3668, Sept. 1987.

[17] W. Schott, "Equalization for partial-response systems," *Proc. 23rd Asilomar Conf. Signals, Syst., Comput.*, Pacific Grove, CA, Oct. 1989, vol. 1, pp. 461-465.

[18] K. D. Fisher, W. L. Abbott, J. L. Sonntag, and R. Nesin, "PRML detection boosts hard-disk drive capacity," *IEEE Spectrum, The Practical Engineer*, Nov. 1996.

[19] L. Richard Carley, "Advanced signal detection strategies for digital magnetic recording," *Carnigie Mellon University Data Storage Systems Center*, 1995.

[20] H. Kobayashi and D. T. Tang, "Application of partial-response channel coding to magnetic recording systems," *IBM Journal of Research and Development*, vol. 15, pp. 368-375, jan. 1971.

[21] H. Kobayashi and D. T. Tang, "Application of probabilistic decoding to digital magnetic recording systems," *IBM Journal of Research and Development*, vol. 15, pp. 368-375, Jan. 1971.

[22] J. G. David Forney, "Maximum-likelihood sequence estimation of digital sequences in the presence of intersymbol interference," *IEEE Transactions on Information Theory*, vol. IT-18, pp. 363-378, May 1972.

[23] R. W. Wood and D. A. Peterson, "Viterbi detection of class IV partial response on the magnetic recording channel," *IEEE Transactions on Communications*, vol. COM-34, pp. 454-461, May 1986.

[24] S. Haykin, "Adaptive Equalization," *Adaptive Filter Theory, 3rd Edition*, pp. 71-73.

[25] G. T. Tuttle & Co., "A 130 Mb/s PRML read/write channel with digital-servo detection," *ISSCC Dig. of Technical Papers*, Feb. 1996, pp. 64-65.

[26] Mike Machado, "Synchronous recording channels: PRML and beyond," *KnowledgeTek*, 1996.

[27] G. Uehara and P. Gray, "A 100MHz output rate analog-to-digital interface for PRML magnetic-disk read channels in 1.2μm CMOS," *ISSCC Digest of Technical Papers*, Feb. 1994.

[28] S. Samueli, "An improved search algorithm for the design of multiplierless FIR filters with powers-of-two coefficients," *IEEE Trans. Circuits Syst.*, vol. 36, pp. 1044-1047, July 1989.

[29] S. A. White, "Applications of distributed arithmetic to digital signal processing: A tutorial review," *IEEE Asia-Pacific Conf. Circuits Syst., July 1989*, pp. 4-18.

[30] H. R. Lee, C. W. Jen, and C. M. Liu, "On the design automation of the memory-based VLSI architectures for FIR filters," *IEEE Trans. Consumer electron.*, pp. 619-630, Aug. 1993.

[31] A. Fettweis, "Digital filter structures related to classical filter networks," *Arch, Elek, ubertragung.*, vol. 25, pp. 79-89, 1971.

[32] A. Sedlmeyer and A. Fettweis, "Digital filters with true ladder configuration," *Int. J. Circuit Theory Appl.*, pp. 5-10, Mar. 1973.

[33] S. S. Lawson, "Digital filter structures from classical analogue networks," *Ph.D. dissertation, Univ. London*, Oct. 1975.

[34] A. M. Ali, "Design of low-sensitivity digital filters by linear transformations," *IEEE Trans. Circuits Syst.*, vol. CAS-27, p. 435-444, June 1980.

[35] A. H. Gray, Jr. and J. D. Markel, "Digital lattice and ladder filter synthesis," *IEEE Trans. Audio electroacoust.*, vol. AU-21, pp. 491-500, Dec. 1973.

[36] J. Makhoul, "A class of all-zero lattice digital filters: Properties and applications," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-28, pp. 304-314, Aug. 1978.

[37] S. K. Mitra, P. S. Kamat, and D. C. Huey, "Cascaded lattice realization of digital filters," *Circuit Theory and Applications*, vol. 5, pp. 3-11, 1977.

[38] Y. C. Lim and S. R. Parker, "Digital lattice filter design using a frequency domain modeling approach," *Proc. 1982 IEEE Int. Conf. Acoust., Speech, Signal Processing, Paris*, pp. 282-285.

[39] D. T. L. Lee, M. Morf, and B. Friedlander, "Recursive least squares ladder estimation algorithms," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-29, pp. 627-641, June 1981.

[40] J. Çhung, H. Kim, and K. K. Parhi, "Pipelined lattice WDF design for wideband filters," *IEEE Transactions on Circuits and Systems – II: Analog and Digital Signal Processing*, vol. 42, no. 9, pp. 616-618, Sept. 1995.

[41] B. Kumar, S. C. Dutta Roy, S. Sabharwal, "Interrelations between the coefficients of FIR digital differentiators and other FIR filters and a versatile multifunction configuration," *Signal Processing 39* (1994) 247-262.

[42] L. Thon, P. Sutardja, F. Lai, and G. Coleman, "A 240 MHz 8-tap programmable FIR filter for disk drive read channels," *ISSCC Dig. Tech Papers*, Feb. 1995, pp. 82-83.

[43] D. J. Pearson, S. K. Reynolds, A. C. Megdanis, S. Gowda, K. R. Wrenner, M. Immediato, R. L. Galbraith, and H. J. Shin, "Digital FIR filters for high speed PRML disk read channels," *IEEE Journal of Solid-Stage Circuits*, vol. 30, no. 12, Dec. 1995.

[44] D. J. Pearson, S. K. Reynolds, A. C. Megdanis, S. Gowda, K. R. Wrenner, M. Immediato, R. L. Galbraith, H. J. Shin, "250 MHz Digital FIR filters for PRML disk read channels," *1995 IEEE International Solid-State Circuits Conference* ISSCC95/Session 5/Disk and Arithmetic Signal Processors/Paper WP5.2

[45] M. Sid-Ahmed, "A systolic realization for 2-D digital filters," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 37, pp. 560-565, Apr. 1989.

[46] B. Edwards, A. Corry, N. Weste, and C. Greenberg, "A single chip ghost canceller," *IEEE J. Solid-State Circuits*, vol. 28, pp. 379-383, Mar. 1993.

[47] M. Z. Komodromos, S. F. Russell, and P. T. P. Tang, "Design of FIR filters with complex desired frequency response using a generalized Remez algorithm," *IEEE Transactions on Circuits and Systems – II: Analog and Digital Signal Processing*, vol. 42, no. 4, April 1995.

[48] P. Gentili, F. Piazza, A. Uncini, "Evolutionary design of FIR digital filters with power-of-two coefficients," *Dip. Di Elettrotica ed Automatica, University of Ancona Via Brecce Bianche, 60131 Ancona, Italy.*

[49] A. G. Dempster, and M. D. Macleod, "Use of minimum-adder multiplier blocks in FIR digital filters," *IEEE Transactions on Circuits and Systems – II: Analog and Digital Signal Processing*, vol. 42, no. 9, Sept. 1995.

[50] C. S. Burros, "Block realization of digital filters," *IEEE Transactions on Audio and Electroacoustics*, vol. AU-20, no. 4, Oct. 1972.

[51] Bela Feher, "Coefficient-dependent logic synthesis of FIR digital filters," *Microelectronics Journal*, 25(1994) 228-235.

[52] H-R Lee, C-W Jen, and C-M Liu, "A new hardware-efficient architecture for programmable FIR filter," *IEEE Transactions on Circuits and Systems – II: Analog and Digital Signal Processing*, vol. 43, no. 9, Sept. 1996.

[53] S. Herman, The Complete Ghost Canceller. *ICCE '93 Educational Session*, June 1993.

[54] P. M. Aziz and J. L. Sonntag, "Equalizer architecture tradeoffs for magnetic recording channels," *Lucent Technologies-Bell Laboratories, Allentown, PA.*

[55] A. Fettweis, "Wave digital filters: Theory and Practice," *Proc. IEEE*, vol. 74, no. 2, pp. 270-327, Feb. 1986.

[56] J. D. Markel, and A. H. Jr. Gray, *Linear Prediction Speech*, Springer-Verlag, New York, 1976.

[57] K. A. Feiste, and E. E. Swartzlander, "High-speed VLSI implementation of FIR lattice filters," *Proceedings of ASILOMAR-29, 1996 IEEE*, pp. 127-131.

[58] Y. C. Lim, "Parallel and pipelined implementations of injected numerator lattice digital filters," *IEEE Transactions on Circuits and Systems – II: Analog and Digital Signal Processing*, vol. 42, no. 7, July 1995, 480-486.

[59] L. M. Smith, "Weighted least-squares design technique for two-dimensional finite impulse response digital filters," *IEEE Transactions on Circuits and Systems – II: Analog and Digital Signal Processing*, vol. 42, no. 4, April 1995, pp. 282-287.

[60] P. P. Vaidyanathan, "Passive cascaded-lattice structures for low-sensitivity FIR filter design, with applications to filter banks," *IEEE Transactions on Circuits and Systems*, vol. CAS-33, no. 11, Nov. 1986.

[61] J. Makhoul, "Stable and efficient lattice methods for linear prediction," *IEEE Trans. Acoustics, Speech, Signal Process.*, vol. ASSP-25, pp. 423-428, Oct. 1977.

[62] J. G. Proakis, and G. M. Dimitris, *Digital Signal Processing*, Third Edition.

4550-93