

Copyright © 1993, by the author(s).  
All rights reserved.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission.

**CIRCUIT TECHNIQUES AND CONSIDERATIONS  
FOR IMPLEMENTATION OF HIGH SPEED CMOS  
ANALOG-TO-DIGITAL INTERFACES FOR DSP-BASED  
PRML MAGNETIC DISK READ CHANNELS**

by

Gregory Takeo Uehara

Memorandum No. UCB/ERL M93/96

15 August 1993

**CIRCUIT TECHNIQUES AND CONSIDERATIONS  
FOR IMPLEMENTATION OF HIGH SPEED CMOS  
ANALOG-TO-DIGITAL INTERFACES FOR DSP-BASED  
PRML MAGNETIC DISK READ CHANNELS**

Copyright © 1993

by

Gregory Takeo Uehara

Memorandum No. UCB/ERL M93/96

15 August 1993

**ELECTRONICS RESEARCH LABORATORY**

College of Engineering  
University of California, Berkeley  
94720

# **Circuit Techniques and Considerations for Implementation of High Speed CMOS Analog-to-Digital Interfaces for DSP-based PRML Magnetic Disk Read Channels**

by

**Gregory Takeo Uehara**

Doctor of Philosophy

Department of Electrical Engineering and Computer Sciences  
University of California at Berkeley

Professor Paul R. Gray, Chair

## ***Abstract***

In order to meet the demands of ever increasing storage capacity and transfer rate requirements in magnetic storage devices, sophisticated signal processing methods are being applied to the magnetic disk drive channel. One approach in particular, Class IV Partial Response with Maximum Likelihood detection (Class IV PRML) appears to be gaining widespread acceptance in the industry as the first of possibly many discrete-time detection approaches that will be used in commercial drives to meet this end. In systems employing Class IV PRML signalling, the read channel electronics resembles a baseband communication receiver requiring functions such as symbol-rate timing recovery, adaptive equalization, and sequence detection. These functions are often performed in the digital domain which facilitates implementation of robust digital equalizers with 6 or more taps. A key element required in these DSP-based channels is the analog-to-digital interface which performs lowpass filtering, possibly some signal pre-conditioning in the form of coarse equalization, sampling, and the analog-to-digital conversion. Due to the off-disk signal-to-noise ratio, the resolution requirements of the A/D converter are on the order of 5-6 bits.

Economical implementation of the analog-to-digital interface in terms of both power and cost is a key problem with implementation of Class IV PRML channels. In particular, implementation of the lowpass filter and pre-equalizer are key problems using conventional techniques requiring the use of either BiCMOS technology or external components.

This thesis describes circuit techniques that can be used to provide the analog-to-digital interface function in CMOS at speeds higher than otherwise possible using conventional approaches. A new switched-capacitor filter architecture is described which employs parallel structures allowing amplifiers multiple output periods for settling and thus breaking the speed bottleneck of conventional switched-capacitor filters. Both single- and multi-rate filters can be implemented allowing the implementation of high sampling rate decimation filters which may be used in the front-end lowpass filter when preceded by a non-critical continuous-time filter which provides attenuation near the sampling rate. A prototype integrated-circuit was designed and built in order to demonstrate the circuit techniques developed in this research. The prototype IC contains a lowpass filter, programmable equalizer, 6-bit analog-to-digital converter, and required clock generation and operates with a sampling rate of 100 MHz with 6 bits of resolution in a conservative 1.2  $\mu\text{m}$  CMOS process. The chip dissipates 900 mW from a 5 V power supply. The lowpass filter is comprised of a cascade of two second-order sections followed by a switched-capacitor 3:1 decimation filter with a sampling rate of 300 MHz. The programmable equalizer has a 3-tap raised cosine response and has three programmable settings.

There are two main parts to this thesis. The first part is a presentation of fundamentals of both digital communication theory and of how data is retrieved in the magnetic disk channel. This provides a background upon which the block level architecture of the prototype can be justified. In particular, we examine the motivation from a system level for equalization prior to the analog-to-digital converter in a system with a digital adaptive equalizer. The second part is a description of the new circuit techniques and architectures developed to achieve higher operating frequencies than possible using conventional approaches. This thesis demonstrates the feasibility of parallel structures in analog processing to achieve both high speed filtering and data conversion and the advantages of analog equalization in DSP-based communication receivers.

Approved by   
Committee Chairman

**To my parents**  
**and**  
**my brother David**

---

## TABLE OF CONTENTS

---

<b>CHAPTER 1</b>	<b>Introduction .....</b>	<b>1</b>
1.0	Introduction .....	1
1.1	Thesis Description.....	4
	References .....	7
<b>CHAPTER 2</b>	<b>Digital Communication Fundamentals .....</b>	<b>9</b>
2.0	Introduction .....	9
2.1	Basic Communication Channel.....	10
2.1.1	Magnetic Disk Read Channel .....	12
2.1.2	Additive White Gaussian Noise Channel .....	15
2.1.2.1	Linear Superposition .....	15
2.1.2.2	Additive White Gaussian Noise .....	18
2.1.2.3	Application to the Disk Drive Channel .....	18
2.1.2.4	Intersymbol Interference .....	20
2.1.3	Nyquist Pulses.....	20
2.2	Equalization.....	24
2.2.1	Approaches to Equalization .....	27
2.2.1.1	Continuous-Time Analog Equalizer .....	29
2.2.1.2	Discrete-Time Equalization .....	29
2.2.2	Stochastic Gradient Algorithm .....	30
2.2.3	Equalizer Architectures.....	37
2.2.3.1	Linear Equalizer .....	37
2.2.3.2	Decision Feedback Equalizer .....	38
2.2.3.3	RAM-Based DFE .....	40
2.2.3.4	Fractionally-Spaced Adaptive Equalizer .....	41
2.2.3.5	Other Equalizer Approaches .....	41
2.3	Timing Recovery .....	42
2.4	Coding .....	44
2.4.1	Partial Response Signalling .....	45
2.4.1.1	Transmitted Power Spectrum of a PAM Signal .....	46
2.4.1.2	Class IV Partial Response .....	47
2.4.1.3	Timing Recovery in PR-IV Channels .....	55
2.5	Sequence Detection (Viterbi Algorithm).....	58
2.6	Summary .....	71
	References .....	73

<b>CHAPTER 3</b>	<b>Detection Techniques for Magnetic Disk Drive Channels</b>	<b>75</b>
3.0	Introduction	75
3.1	Fundamentals of the Read/Write Process	76
3.1.1	Background Physics	76
3.1.2	Write Process	80
3.1.3	Read Process	81
3.1.3.1	Inductive Read Heads	81
3.1.3.2	Magnetoresistive Read Heads	82
3.2	Peak Detect Channels	83
3.3	Discrete-Time Detection	87
3.3.1	Partial Response Signalling	88
3.3.2	RAM-Based DFE	90
3.3.3	Fixed-Delay Tree Search with Decision Feedback Equalization	90
3.4	Comparison of Approaches	91
3.5	Summary	93
	References	95
<b>CHAPTER 4</b>	<b>Architectural Approaches</b>	<b>97</b>
4.0	Introduction	97
4.1	Pertinent Basics	99
4.1.1	Quantization Noise in Analog-to-Digital Conversion	99
4.1.2	Noise Shaping and Amplification	100
4.1.3	Latency in Feedback Loops	102
4.1.4	Analog Discrete-time vs. Digital Processing	103
4.2	Functions Required in a Baseband Communications Receiver	104
4.3	Filter Implementation Approaches	107
4.3.1	Analog Filtering Approaches	108
4.3.1.1	Sallen-Key Sections	109
4.3.1.2	Biquadratic Sections	110
4.3.1.3	Ladder Filter	112
4.3.2	Implementation of Monolithic Analog Filters	113
4.3.2.1	Continuous-Time Filters	114
4.3.2.2	Switched-Capacitor Filters	119
4.3.2.3	Comparison of Continuous-Time and Switched-Capacitor Filters	120
4.4	Architectural Approaches	120
4.4.1	Ideal Front-End Receiver	121
4.4.2	Practical Limitations	121
4.4.3	All Digital Receiver	123
4.4.4	Digital Receiver with Analog Pre-equalization	125
4.4.5	Digital Receiver with Adaptive Analog Pre-equalization	128
4.4.6	All Analog Receiver	129

4.5	Comparison of Approaches .....	130
4.6	Proposed Architecture for PR-IV Disk Drive Read Channel .....	131
4.7	Summary.....	140
	References .....	141
<b>CHAPTER 5</b>	<b>High Speed Switched-Capacitor Transversal Filters.....</b>	<b>143</b>
5.0	Introduction .....	143
5.1	Transversal Filters.....	147
5.1.1	Basic Transversal Filter Structure .....	147
5.1.1.1	Advantages of FIR Filters .....	149
5.1.1.2	Example: Raised Cosine Filter .....	150
5.1.2	Previous Work in SC Transversal Filters .....	151
5.1.2.1	Double Sampling Front-End .....	151
5.1.2.2	FIR-IIR Structures.....	152
5.1.2.3	Polyphase Filter Structures.....	153
5.1.2.4	Rotating Switch-Based FIR Structures.....	154
5.1.2.5	Analog Delay Line-Based FIR Structures .....	155
5.2	High Speed SC FIR Architectures .....	156
5.2.1	SC Decimation Filter Block Diagram .....	157
5.2.2	SC Decimation Filter Architecture.....	159
5.2.3	Single-Rate Filters.....	165
5.2.4	Long Impulse Response Decimation Filters .....	166
5.2.4.1	Example: Spread Spectrum Decorrelator .....	167
5.2.4.2	Partial Sum-Accumulator Bank Architecture.....	168
5.3	Sampling Clock Generation.....	171
5.4	Capacitor Design .....	173
5.5	Performance Limitations of High Speed SC FIR Filters.....	175
5.5.1	Model for Sources of Error .....	175
5.5.2	Input-Referred Offset Voltage .....	176
5.5.3	D.C. Gain.....	179
5.5.4	Variation in Response of Parallel Filters.....	181
5.5.5	Sampling Jitter and Fixed Phase Offsets.....	183
5.5.6	Summary of Non-Idealities .....	186
5.6	Adaptive SC FIR Filters .....	187
5.7	Application of the Architecture to the Implementation of Digital Filters .....	192
5.8	Summary.....	194
	References .....	196

<b>CHAPTER 6</b>	<b>High Speed Analog Multiphase Sampler and Summer .....</b>	<b>199</b>
6.0	Introduction .....	199
6.1	Summing Zero-Order Sample-and-Hold.....	201
6.1.1	Sampling and Summing Network.....	201
6.1.2	Zero-Order Hold Circuit .....	203
6.1.3	Switched-Capacitor Summing Circuit with On-the-Fly Reset .....	205
6.1.4	Non-idealities of SC Summer .....	207
6.2	Operational Transconductance Amplifier (OTA).....	209
6.2.1	Advantages Over Voltage Amplifiers.....	209
6.2.2	Feedback Factor Effect on Amplifier Performance.....	211
6.2.2.1	Settling Behavior .....	212
6.2.2.2	Open Loop Gain.....	213
6.2.2.3	Input-Referred Voltage Offsets.....	214
6.2.3	Folded-Cascode Operational Transconductance Amplifier (OTA).....	216
6.2.4	Telescopic Operational Transconductance Amplifier (OTA) .....	217
6.2.4.1	Speed and Voltage Swing .....	217
6.2.4.2	Common-mode Input Range.....	218
6.2.5	Telescopic OTA with Preamp.....	219
6.3	Sampling and Amplifier Clock Generation .....	225
6.3.1	Sampling Phase Generation .....	225
6.3.2	Non-Overlap Clock Generation .....	232
6.3.2.1	Local Non-Overlap Clock Generation .....	232
6.3.3	Global Non-Overlap Clock Generation .....	233
6.3.3.1	Circuit Approaches .....	234
6.4	Summary .....	241
	References .....	243
<b>CHAPTER 7</b>	<b>High Speed Analog-to-Digital Converters.....</b>	<b>245</b>
7.0	Introduction .....	245
7.1	Analog-to-Digital Converter System-Related Issues .....	247
7.1.1	Loading on the Previous Block.....	247
7.1.2	Quantization Noise and Distortion .....	249
7.1.3	Converter Latency.....	250
7.1.4	Comparator Metastability .....	250
7.2	A/D Converter Architectures.....	252
7.2.1	Flash Converters .....	252
7.2.2	Subranging Converters .....	254
7.2.3	Pipeline Converters.....	256
7.2.4	Two-Step Pipeline Converters without Interstage Gain .....	258

---

7.3	Highly Parallel Two-Step Pipeline Implementation .....	258
7.3.1	System Context .....	259
7.3.2	A/D Converter Sampling of the S/H Amplifier .....	261
7.3.3	Difference Amplifier -- Regenerative Latch Comparators .....	264
7.3.4	Key Implementation Considerations .....	268
7.3.5	Highly Parallel Implementation .....	269
7.4	Summary .....	274
	References .....	276
 <b>CHAPTER 8</b> Prototype Description and Design.....		<b>277</b>
8.0	Introduction .....	277
8.1	Continuous-Time Anti-Alias Filter.....	279
8.2	Operational Amplifier and Analog Bias Generation.....	283
8.3	Switched-Capacitor 3:1 Decimation Lowpass Filter.....	291
8.4	Programmable Switched-Capacitor Raised Cosine Equalizer.....	299
8.5	6-Bit 100 MHz Analog-to-Digital Converter .....	305
8.6	Clock Generation.....	324
8.7	Experimental Results.....	350
8.8	Prototype Layout .....	355
8.8.1	Layout.....	355
8.8.2	Power Consumption by Block.....	357
8.9	Summary.....	358
	References .....	360
 <b>CHAPTER 9</b> Summary and Conclusions .....		<b>361</b>
9.0	Introduction .....	361
9.1	Key Research Themes .....	362
9.2	Research Results.....	363
9.3	Future Work and Improvements .....	364
	References .....	366

## Acknowledgments

I am very grateful to my research advisor Professor Paul R. Gray whom I had the privilege to work with for 3 years in industry before coming to Berkeley and working with him here. He has been extremely encouraging and supportive and I have learned countless things through our association. His keen insight into engineering problems can only be matched by his integrity and commitment to quality. I was very fortunate to take 4 classes from Professor David G. Messerschmitt and have him as a joint advisor on my Masters project and on my qualifying exam committee. I learned a great deal about communications and signal processing through his classes and during discussions with him. Professor Jan Rabaey was on my qualifying committee and was always available for a discussion. Along with Professor Messerschmitt, two professors who are also outstanding teachers are Professors Robert G. Meyer and Ping K. Ko. I will always remember the clarity and intuition these professors shared during their lectures and openness to questions and respect for their students. The standards and care that I have seen each of these professors apply to their research and teaching are standards that I will strive for as a professor. I am grateful to Professor Joseph M. Kahn and Professor Charles Stone from the Department of Statistics who were readers of this thesis. Professor Stone also served on my qualifying exam committee.

There are many student colleagues who made this journey through graduate school so enjoyable and all have been important friends. When I first arrived at Berkeley, Professor Stephen Lewis (UC Davis) and Professor Vijay Madisetti (Georgia Tech) were both senior students here who helped me get acclimated and made me feel especially comfortable. Ken Nishimura helped me in countless ways with computing resources and procuring the necessary equipment for anything and everything I needed. Cormac Conroy shared many keen insights into parallel channel signal processing and many of his results are used in this thesis. He also coached me in preparation for many presentations. Weijie Yun helped me prepare for many talks and he and his wife Haifang have been wonderful friends. Robert Neff was a great guy to bounce circuit ideas off of. He helped get the PC working in the laboratory for testing of the prototype. Other colleagues who are now long time friends with whom I've had countless discussions with both technical and non- include Timothy Hu, Gani Jusuf, Nhat Nguyen, Monte Mar, and Ed Liu. Monte also shared his ADC testing software and set it up for me in the lab. Professor John Barry (Georgia Tech) helped me through many of the communication classes. Jon Duster was extremely helpful in teaching me about B-SIM and was always generous with his time. Jeffery Weldon read a portion of this thesis and gave me some excellent feedback. More importantly, he has rejuvenated my love of golf! There was never a question I had that Sam Sheng could not answer and he always took the time to do so. Anantha Chandrakasan always made time to share his insights into digital circuit design. I had a great time working together with two extremely hard working colleagues Caesar S. H. Wong and Jacques C. Rudell on a high speed DSP. Some of the other new guard whose thorough work I have benefitted from include Thomas Cho, David Cline, and David Helman (who did a great job on his Master's Design report). Cynthia Keys, Eric Boskin, and Zeina Daoud always had encouraging and affirming words that went further than they will ever know. It was always great to run into Sherry Lee or Henry Sheng in the hall because their pleasant and friendly demeanors would always cheer me up.

---

Andy Burstein shared his program *ext2spice* without which I would still be verifying my layout for first silicon! Having friends and professors like these makes it tough to leave Berkeley.

Two industrial friends were particularly supportive and encouraging of my research work in magnetic disk drive read channel electronics from the start. They are Roger Wood and Hemant Thapar of IBM Corporation. I am grateful for their insight into the disk drive channel and their kind encouragement. I am glad I got stuck in an airport with Rich Galbraith also of IBM. He initially suggested that there may be advantages to analog equalization in DSP-based read channels. I am grateful to my friends with whom I worked with in industry before coming to Berkeley. Those who were in the Telecommunications Group of the Intel Corporation back in 1983-1984 and those who were at Microlinear Corporation in 1985-1986. My friends at Microlinear really helped me grow as a circuit designer and engineer. Steve Dreyer, who was my boss and now is my friend is the one who really gave me a chance and taught me to engineer. Harlan Ohara taught me to break every problem down into bite-sized morsels and has always been there for me to call (and has loaned me a stereo for the last 7 years). Other colleagues at Microlinear from whom I learned a great deal about circuit design include Carlos Laber, Peter Kwok, Mike Armstrong, Chowdhury Rahim, Lee-Chung Yiu, Lawrence Tse, Brad Pickett, Hung Ngo and Uncle Joe Li.

Doreen and the Shew family were very special friends and I am grateful to them. It was great having friends like Lauren Cox-Pursley and Tabo Egawa to meet for lunch. From First Presbyterian Church of Berkeley there were Kathy and Andy Way and Youth Theatre whom I enjoyed playing music with and simply *being* with. Pastor Earl Palmer helped me see with new eyes what it means to be a Christian. The music of Stephen Curtis Chapman keeps reminding me that there is *More To This Life*. Junior High intern Eric Jacobsen was a very important friend and discussion partner.

The Hawaii Club guys (and gals) were great and there were many. Patrick Pai who is getting his Ph.D. at UCLA helped convince me that finishing graduate school at 30 would not be so bad (Not so bad at 33 either! Thanks Pat.). Keith Abe and Lance Takara with whom I had countless barbecues with were here from the start and flew up from Hawaii for my graduation helping to make it very special. My high school buddies and their wives and girlfriends were always there to call and they always managed to cheer me up when necessary. They, together with my family (Mom, Dad, Dave, the Shinkawas, aunties, uncles, cousins, and Ryan and Kristin) always helped to give me a perspective of where I am headed. My dear cousin Paul Shinkawa has always been my big brother. He always knows exactly what I need to hear and keeps me aiming high. My parents have been extremely supportive throughout the years and continued with financial and emotional support that helped make this journey through graduate school possible. I am very grateful to both of them. I am also grateful to my brother David for the trip to Maui and for being such a solid guy.

Finally, I am grateful to my Lord and Savior Jesus Christ for the guidance, peace, joy, and *excitement* that comes from following Him.

This research was supported by NSF MIP 9109525, the California MICRO Program, and National Semiconductor.

# CHAPTER 1

## Introduction

---

### 1.0 Introduction

As the need for higher storage capacities and transfer rates in magnetic disk drives continues to escalate, increased sophistication is required of the signal processing which occurs on the data signal read from the disk. *Peak detection* is the processing technique traditionally used for data recovery from the magnetic media, and is still used in the majority of drives shipped today. The signal received from the disk containing the information that has been stored is a sequence of alternating positive and negative pulse peaks which correspond to changes in the direction of magnetization of the media performed during the write process. When no transition occurs in the media, no peak occurs in the signal. In a peak detect read channel, the read signal is observed for the presence or absence of peaks corresponding to data bits of 1 and 0, respectively. Although this is an overly simplistic view of peak detection, it describes the simple and somewhat elegant nature of this approach which allows the electronics in the signal path to remain relatively simple. Integrated circuits containing all the circuitry required in both the write and the read channels, running at data rates up to 30 MHz and consuming 300 mW are commercially available [1].

However, in order to achieve higher recording densities and transfer rates, peak detection is being replaced in an increasing number of implementations by discrete-time processing approaches employing communication techniques such as adaptive equalization and sequence detection. One signalling approach in particular, *Partial-Response Maximum Likelihood* or *PRML* using *Class IV Partial-Response* signalling or *PR-IV* is becoming the first de facto discrete-time signalling standard. More sophisticated approaches are queued up and will be implemented as densities and transfer rates exceed that achievable by PR-IV.

The signal processing requirements in PR-IV channels include variable gain control, adaptive equalization, sequence detection, and timing recovery. Many of these functions are easily and robustly implemented in the digital domain making a high speed *analog-to-digital interface* an important requirement. The A/D interface is comprised of an A/D converter (ADC) and a filter implementing noise filtering, anti-aliasing, and possibly pre-equalization. Pre-equalization is attractive because it reduces low-frequency energy in the signals and in some cases may allow reduction of the required resolution of the ADC block, important at the high sampling rates used. Previous implementations of this block have used standard conversion techniques preceded by external L-C fixed filters or integrated  $G_m$ -C filters. Monolithic implementation of this filter/ADC combination is an important economic goal, but current goals for bit rate in the channel of 100Mb/sec and higher make it very difficult to implement the filtering function using standard continuous-time filtering techniques in currently available technologies.

Along with architectural choices such as the partitioning of the signal processing between the analog and digital domains, processing technology choices must be incorporated into the design process. Currently, the candidates are CMOS, bipolar, and BiCMOS. CMOS is attractive due to the high levels of integration possible. Furthermore, high speed digital circuits are possible which dissipate no static power. A major disadvantage is that due to the square law nature of the MOS transistor, a relatively low transconductance results which makes difficult the implementation of high speed analog functions as compared with its bipolar counterpart. On the other hand, digital circuits implemented in bipolar technology tend to dissipate large amounts of power. One solution is to use multi-chip approaches with bipolar technology used in the analog front-end followed by a high speed dedicated digital signal processor (DSP) implemented in CMOS. Another approach is

to use BiCMOS which has both technologies available on the same substrate. The disadvantage of this is that the cost associated with adding high performance bipolar devices to a CMOS process may add as much as 30 to 35 percent to the cost of CMOS processing. Microprocessors and electronic memory devices make up the majority of electronics revenue worldwide and most often are implemented in CMOS. As a result, there is a constant drive and competition toward reduction of the feature sizes of CMOS technology for increased performance in these mega-volume products. It is therefore very attractive to keep designs in CMOS, a technology in which high levels of integration are possible, improvements to the technology occur continually, and is available *before* similar BiCMOS technologies with the same CMOS feature size is brought up to speed.

This thesis examines approaches for CMOS implementation of the analog-to-digital interface required in magnetic disk read channels employing PRML signalling and performing adaptive equalization and sequence detection in the digital domain. A key theme running through much of this research is the implementation of high rate signal processors using interleaved lower speed processors employing a combination of pipelining and parallelism. This theme has been applied by many of my colleagues here at Berkeley under the guidance of Professor Paul R. Gray in different applications. Timothy Hu applied parallel processing in an implementation of a 450 MBit/sec fiber optic receiver in 1.2  $\mu\text{m}$  CMOS [2]. Cormac Conroy and David Cline implemented an 85 MSample/sec 8-bit analog-to-digital converter using parallel pipeline converters in 1.0  $\mu\text{m}$  CMOS [3]. Caesar Wong and Jacques Rudell are applying parallel processing and pipelining in an implementation of an 8-tap adaptive digital equalizer for 100 MBit/sec PRML disk drive channels. This last project will be described briefly in this thesis.

One of the key problems of implementation of the analog-to-digital interface is implementation of the lowpass anti-alias filter which precedes the analog-to-digital converter for sampling rates on the order of 100 MHz. A new switched-capacitor transversal filter architecture is introduced in this thesis which allows the implementation of filters with bandwidths far greater than achievable using conventional approaches. The architecture employs parallelism which allows amplifiers multiple output periods to settle. As a result, higher filter bandwidths are achievable than possible using conventional approaches. This filter architecture was used in an experimental CMOS prototype IC targeting the requirements of the PRML read channel in order to demonstrate

the potential of the architecture in high speed A/D interfaces. Both a lowpass decimation filter and a programmable symbol rate equalizer were implemented on the same chip preceding an on-chip 6-bit analog-to-digital converter. The sampling rate of the decimation filter is 300 MHz and the output sampling rate of the A/D converter is 100 MHz. This prototype was implemented in a 1.2  $\mu\text{m}$  CMOS technology.

## 1.1 Thesis Description

This thesis is broken primarily into 2 sections. In the first section, Chapters 2-4, the context for which the circuit techniques and approaches developed in this research is developed and presented. In Chapters 5-8, the circuit techniques are presented in detail. Chapter 9 contains a summary and conclusions.

In Chapter 2, we shall examine important communications concepts such as coding, equalization, timing recovery, and sequence detection. We will also develop the additive white gaussian channel model and apply it to the magnetic disk drive channel employing discrete-time detection techniques. The material in this chapter serves as background for the reader who may be unfamiliar with these basic and important concepts from communication theory.

In Chapter 3, the disk drive read channel will be examined more closely. The device which senses the recorded information on a disk is called the *read head*. There are two read head approaches in use today. One is the traditional inductive read head and the other is the newer magneto resistive head which may become quite prevalent in the future. After a description of both, we will follow this with a survey of detection techniques including peak detection and some of the more popular discrete-time techniques that have been proposed or are now in use.

In Chapter 4, we will examine different receiver architectures for baseband digital communication channels. In particular, we will look at different partitioning of the signal processing required in receivers between the analog and digital domains, and present some of the basic trade-offs between different partitioning approaches. Filtering, for noise limiting and equalization is one of the key signal processing functions required in a receiver before detection. Filtering can be performed in the analog domain, both in continuous and discrete-time, and in the digital domain.

Even in receivers that employ digital signal processing, analog filtering is required for anti-aliasing prior to sampling and the analog-to-digital conversion. We will examine the limitations associated with conventional approaches to continuous-time filter implementation. In channels like the magnetic disk drive channel, it is known beforehand that signal conditioning in the form of equalization will be required prior to detection. From an implementation and performance standpoint, it may be beneficial to perform some of this signal processing in the analog domain in receivers employing digital signal processing.

In this research, techniques were developed for implementation of high speed analog-to-digital converter front-ends for the magnetic disk read channel application employing discrete-time techniques such as Class IV Partial Response Signalling and Maximum Likelihood detection. We assume the use of a digital signal processor performing adaptive equalization and sequence detection. In Chapter 4, we propose an architecture for an experimental prototype demonstrating these techniques. This prototype contains two filter blocks employing a new switched-capacitor filter architecture employing parallel processing. This approach allows the implementation of filters in CMOS with bandwidths that are much greater than otherwise possible using conventional approaches. Both a lowpass anti-alias filter and a programmable equalizer are included along with a 6-bit analog-to-digital converter. From an overall system implementation viewpoint, it is not obvious what advantages, if any, are associated with providing some equalization in the analog domain before the analog-to-digital converter; particularly in a system in which digital signal processing is employed. This will be examined at the end of Chapter 4.

The material in these three chapters are mainly to provide a background and context for the read channel receiver architecture presented at the end of Chapter 4. A prototype integrated circuit demonstrating high speed parallel analog processing is the core design portion of this research and this prototype is the focus of Chapters 5-8.

In Chapter 5, new architectures for high speed switched-capacitor (SC) filters is presented. We will first present the basic filter architecture in the context of a decimation filter and then simplify it to a single-rate filter. We will also examine the implementation of filters with long impulse responses and propose an approach which will make the filter adaptive. Performance limitations associated with circuit implementation of these architectures are also presented.

Implementation of these filters require circuit building blocks such as a high speed amplifier and clock generation circuitry. Design approaches and considerations for the key blocks are presented in Chapter 6.

In Chapter 7, we will examine considerations for high speed analog-to-digital converters that are part of a signal processing IC. We will then examine candidate architectures to be used in the prototype before settling on the approach that was actually used. We will then develop a parallel implementation of this architecture in the context of the other circuitry on the prototype. In particular, the A/D converter was optimized to interface to the parallel filter structure which precedes it.

In Chapter 8, a detailed description of each of the blocks in the experimental prototype is presented. We will walk through the signal path and present the key ideas and design considerations associated with each of the blocks. Following this, results taken in the laboratory from the prototype are presented.

Finally in Chapter 9, we will summarize the key contributions of this research together with the key ideas contained in thesis.

## References

- [1] C. Peterson, P. Cheung, T.-S. Chung, W. Guo, K.-H. Loh, K. Miyasa, L. Pycior, M. R. Robinson, and K. Tomioka, "A 3-5.5V CMOS 32 Mb/s Fully-Integrated Read Channel for Disk-Drives", *Proceedings IEEE Custom Integrated Circuits Conference*, San Diego, CA, May 1993.
- [2] T. Hu and P. Gray, "A Monolithic 480 Mb/s Parallel AGC/Decision/Clock-Recovery Circuit in 1.2  $\mu\text{m}$  CMOS", *ISSCC 1993 Digest of Technical Papers*, vol. 36, pp. 98-99, Feb. 1993.
- [3] C. S. G. Conroy, D. W. Cline, and P. R. Gray, "A High Speed Parallel Pipelined ADC Technique in CMOS", *1992 Symposium on VLSI Circuits Digest of Technical Papers*, pp. 96-97, June 1992.

The first part of the document discusses the importance of maintaining accurate records of all transactions. It emphasizes that every entry should be supported by a valid receipt or invoice. This ensures transparency and allows for easy verification of the data.

In the second section, the author outlines the various methods used to collect and analyze the data. This includes both primary and secondary data collection techniques. The analysis focuses on identifying trends and patterns over time, which is crucial for making informed decisions.

The final part of the document provides a detailed breakdown of the results. It shows that there has been a significant increase in sales volume, particularly in the middle and lower income brackets. This suggests that the current marketing strategy is effective, but there may be room for improvement in targeting the higher income demographic.

Year	Q1	Q2	Q3	Q4	Total
2018	120	150	180	200	650
2019	150	180	220	250	800
2020	180	220	280	320	1000
2021	220	280	350	400	1250
2022	280	350	420	480	1530
2023	350	420	500	550	1820
2024	420	500	580	650	2150
2025	500	580	650	720	2450
2026	580	650	720	800	2750
2027	650	720	800	880	3050
2028	720	800	880	950	3350
2029	800	880	950	1020	3650
2030	880	950	1020	1100	3950
2031	950	1020	1100	1180	4250
2032	1020	1100	1180	1250	4550
2033	1100	1180	1250	1320	4850
2034	1180	1250	1320	1400	5150
2035	1250	1320	1400	1480	5450
2036	1320	1400	1480	1550	5750
2037	1400	1480	1550	1620	6050
2038	1480	1550	1620	1700	6350
2039	1550	1620	1700	1780	6650
2040	1620	1700	1780	1850	6950
2041	1700	1780	1850	1920	7250
2042	1780	1850	1920	2000	7550
2043	1850	1920	2000	2080	7850
2044	1920	2000	2080	2150	8150
2045	2000	2080	2150	2220	8450
2046	2080	2150	2220	2300	8750
2047	2150	2220	2300	2380	9050
2048	2220	2300	2380	2450	9350
2049	2300	2380	2450	2520	9650
2050	2380	2450	2520	2600	9950

## CHAPTER 2

# Digital Communication Fundamentals

---

### 2.0 Introduction

Reduced overall system cost and increased performance are key motivations for application specific integrated circuit (ASIC) approaches to communication system implementation. A firm understanding of concepts from communication theory allows the circuit designer to make decisions at the system level which may have important implications at the circuit level.

In this chapter, we will examine some of the key fundamentals of communication system design, applying them to the magnetic disk drive channel employing discrete-time detection methods. Rigorous and much more comprehensive treatments on all of the topics discussed in this chapter exist in the literature [1,2]. The purpose of this chapter is twofold. First, to provide a background of some important principals of communication system design for a later discussion of different receiver architectures and approaches. Second, present the basic principals of the magnetic disk read process using inductive read heads along with a discussion of the interpretation of the magnetic disk read channel employing sampled-data detection techniques as a discrete-time communications channel.

The majority of magnetic disk drives on the market today employ *peak detection* as opposed to discrete-time detection. Peak detection will be discussed in Chapter 3 along with a discussion of other approaches for data recovery in the disk drive channel. At higher recording densities, discrete-time detection has been shown to result in higher performance than peak detection [14,15,16] and much of this thesis is concerned with integrated circuit implementation of the analog-to-digital interface needed in discrete-time channels employing digital signal processing.

Most general treatments of communications channels assume passband channels where modulation is used to move the signal spectrum to a frequency much higher than that of the original signal spectrum. This is done in order to share the medium by multiple users as is done in the case of AM and FM radio where different stations transmit their signals over different frequency bands. Throughout this thesis, the focus will be on *baseband* channels where the bandwidth of the transmitted spectrum is on the order of the data rate as is the case in the magnetic disk drive channel.

## 2.1 Basic Communication Channel

A communication channel can be viewed as shown in Figure 2.1. Information from User 1 is passed to the *transmitter* which transforms the information into a signal which is compatible with the *channel*. Distortion and noise are picked up along the way through the channel. The *receiver* attempts to transform the signal received from the channel with added noise and distortion into the information originally sent by User 1 before passing it on to User 2. We shall refer to the end-to-end connection between users as the *communication channel* and the medium over which the transmitted signal is sent as simply the *channel*.



Figure 2.1 Block diagram of a communication channel.

Much of communication theory is involved with techniques for reliable and efficient transmission of information over different types of channels. Due to the impairments that occur to the signal along the way, there is always a non-zero probability of making an error in transforming the received signal back into the original transmitted information. Communication theory allows transmission across well behaved channels to occur with arbitrarily small probabilities of error at the expense of transmitter power, signal bandwidth, and implementation complexity.

Examples of channels include the airway in television, radio, and other wireless applications, copper cable in the case of a telephone connection, a glass or plastic fiber cable in a fiber optic channel, and even water in the case of undersea monitoring devices and navigation equipment. The magnetic disk drive channel used for information storage is another example of a channel. In this communication channel, data is sent from one user and retrieved by itself or another user at a later point in time. The time interval can be microseconds on up to many years.

We shall limit our discussion to the communication theory concepts and techniques important to the magnetic disk drive channel employing discrete-time signal processing. Returning to the simple communication channel model of Figure 2.1 for the disk drive application we have:

User 1: Application program needing binary data to be temporarily stored.

Transmitter: Transforms binary data to a form compatible with the channel and writes on to the magnetic disk.

Channel: Magnetic disk.

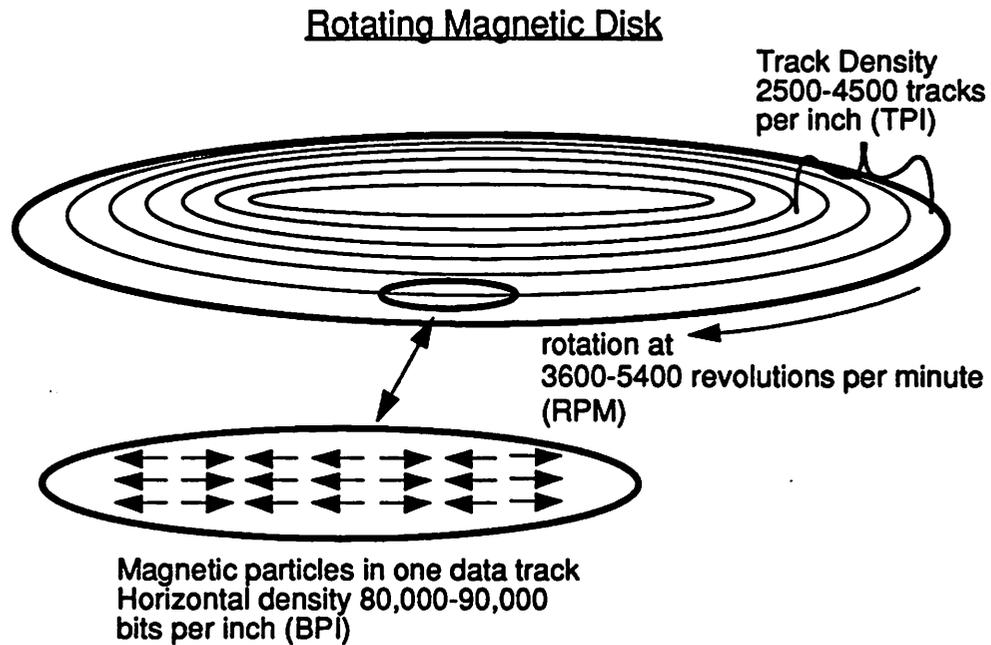
Receiver: Reads the signal from the channel and processes it to recover the original binary data sent by User 1. Passes receive data on to User 2 which in this case is User 1.

We shall now develop a mathematical model for the communication channel followed by an examination of some of the key techniques used in baseband receivers to recover the original binary data in the presence of noise and other impairments. Before proceeding with the development of this model, let us first examine the basics of the magnetic disk read channel which we will use as a vehicle to motivate and apply the communication techniques. After presenting an analytical expression for an isolated pulse from a magnetic disk, we will return to the development of the communication channel model.

### 2.1.1 Magnetic Disk Read Channel

Let us now examine the magnetic disk read process using an inductive read head. Much of this discussion is continued with more depth in Chapter 3. The intent is to present enough of the basics to apply discrete-time communication fundamentals.

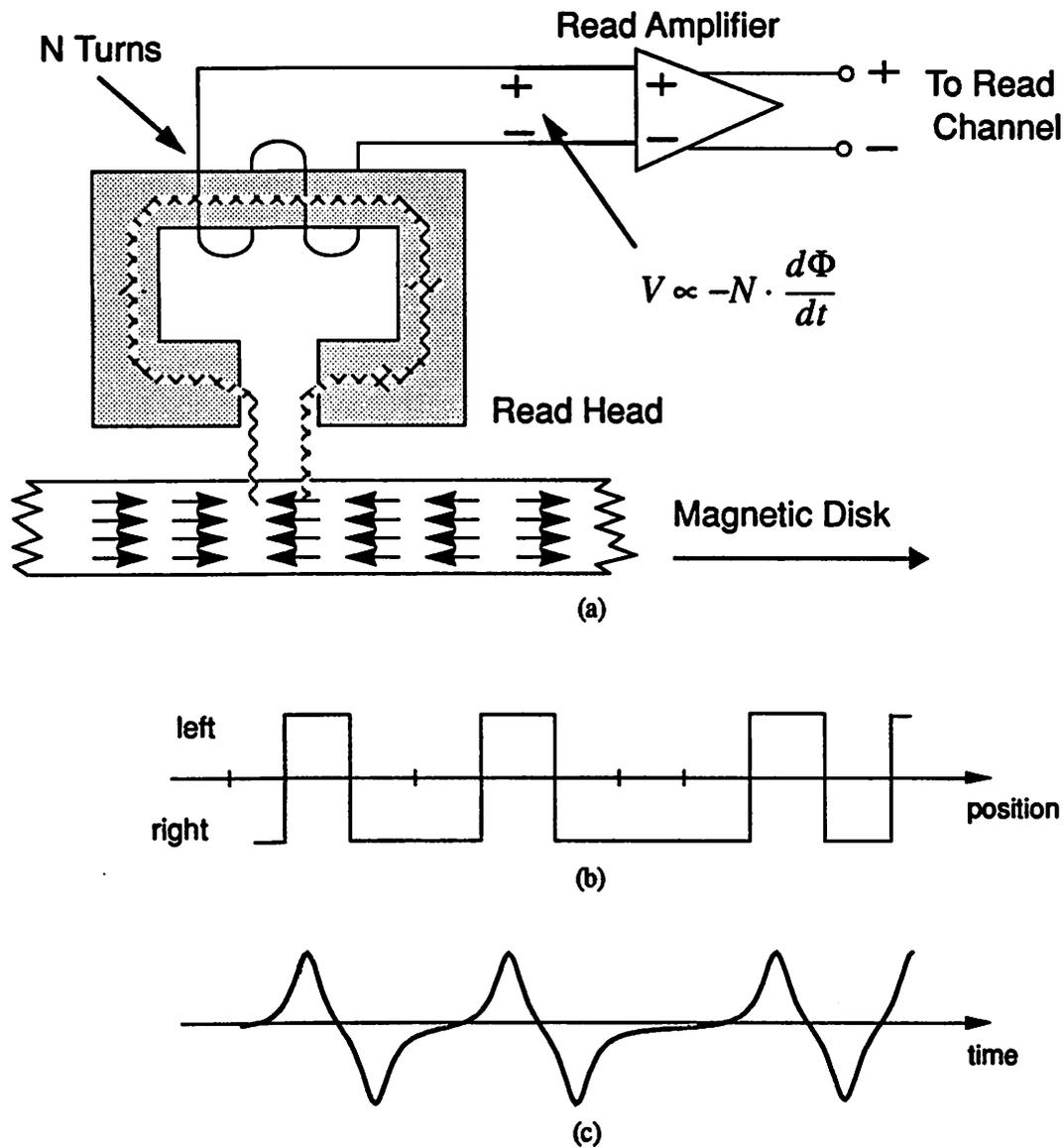
A magnetic disk can be thought of as a platter with many tiny little magnets evenly dispersed upon its surface as shown in Figure 2.2. These magnets are aligned in narrow rows called *tracks* in a circular fashion at different radii like the grooves on a phonograph. One measure of recording density is the number of tracks per inch or t.p.i. which today can be in the range of 2,500-4,500 tracks/inch [17]. Unlike the phonograph application where one continuous groove exists from the outside of the record to the inside, data is written onto the disk in concentric circles. Information is stored by magnetizing the magnets in one-of-two directions pointing left to right or right to left. The change of direction of the polarity of the magnetic particles results in the change of direction of the magnetic flux locally extending from the platter surface. Another measure of recording density is the number of flux changes per inch which are measured in the direction that data is written. This is referred to as linear density and currently is in the range of around 80,000-90,000 bits per inch. In conventional disk drives today, the platter spins at rates between 3600 and 5400 revolutions per minute (rpm).



**Figure 2.2** Conceptual diagram of a single magnetic disk platter.

Let us now consider the read process which converts the flux transitions into a voltage for the read channel circuitry as shown conceptually in Figure 2.3a. In the figure, one data track is shown passing by a stationary read head as the disk rotates. The read head is made of a high permeability alloy in order to efficiently sense the flux near the surface of the magnetic media. As the platter spins by the head, the time rate of change of the flux ( $d\Phi/dt$ ) is sensed by the  $N$ -turn coil. The result is a voltage at the output of the coil proportional to the number of turns and the time rate of change of the flux passing through the read head. Shown in Figure 2.3b is an example of data written onto the disk shown as magnetization in one of the two horizontal directions. The resultant read signal appears as a voltage at the terminals of the coil is shown in Figure 2.3c. Ideally, this would appear as impulses but due to the lowpass nature of the read process, the signal appears as a differentiated lowpass version of the magnetization pattern. The peak amplitude of the voltage at this point is on the order of hundreds of micro volts. This signal is then amplified by the *read amplifier* and passed to the read channel electronics which processes the signal in order to retrieve

the information stored on the disk. Note the difference between the x-axis labels of Figure 2.3b and Figure 2.3c, in particular, how *position* on the disk becomes *time* in the read signal.



**Figure 2.3** (a) Conceptual diagram of read process using inductive read heads. (b) An example of the magnetization pattern written onto the disk. (c) The corresponding read signal which is a differentiated low-pass version of the write pattern.

One of the analytical expressions used for an isolated transition in the read signal (which corresponds to a step change in the magnetization pattern) is the Lorentzian pulse shape shown in Figure 2.4 and given by

$$l(t) = \frac{1}{1 + \left(\frac{2t}{PW_{50}}\right)^2} \quad (2.1)$$

where  $PW_{50}$  (pulse width fifty) is the time duration distance between the points at which the amplitude is fifty percent of its peak value. This expression will be used to model the read signal with random data.

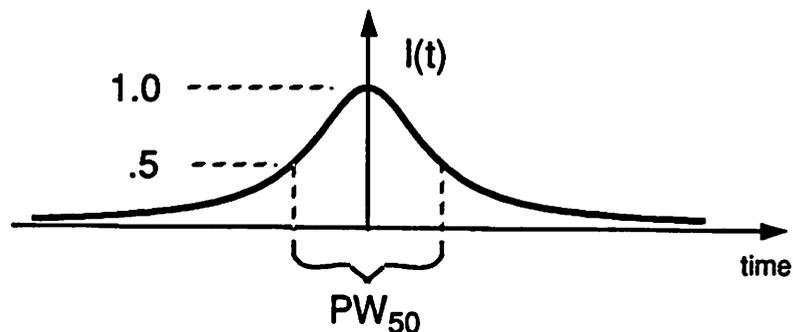


Figure 2.4 Lorentzian pulse which models an isolated transition.

## 2.1.2 Additive White Gaussian Noise Channel Model

Let us now develop the *additive white gaussian noise channel model* which facilitates the analysis and design of communication systems.

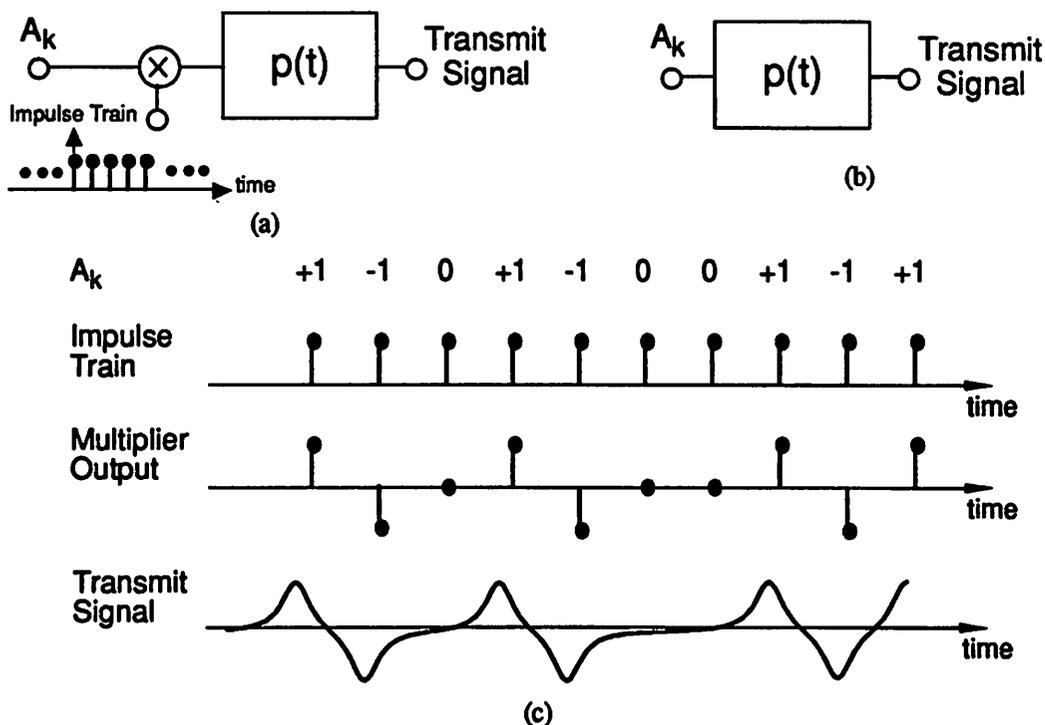
### 2.1.2.1 Linear Superposition

Linear superposition is an important concept in communication system theory. Under conditions that it is valid, the transmit or receive signal in a system with isolated pulse  $p(t)$  can be written as the linear superposition of all isolated pulses over all time given by

$$s(t) = \sum_{k=-\infty}^{\infty} A_k \cdot p(t - kT) \quad (2.2)$$

where  $p(t)$  is the isolated pulse shape (example: the Lorentzian pulse just described),  $A_k$  is from the *alphabet set*, and  $T$  is the period between pulses. The alphabet set depends on the particular system and defines the number of levels that will need to be differentiated by the receiver. Examples of the alphabet set are  $\{-1,+1\}$  in binary antipodal or 2-level signalling and  $\{-1,0,+1\}$  in ternary or 3-level signalling. This method of multiplying each isolated pulse with one of the elements of an alphabet is descriptively called *pulse amplitude modulation (PAM)*. Each isolated pulse multiplied by an  $A_k$  is a *symbol* which may represent one or more bits.

The PAM description allows a transmitter with impulse response  $p(t)$  to be modelled as shown in Figure 2.5a. A symbol stream made by multiplying an impulse train with a sequence of symbols  $A_k$  is passed to the transmit filter. The result is the transmit signal given by Eq. 2.2. The simplified version shown in Figure 2.5b assumes multiplication with the impulse train and will be used in the remainder of this thesis. Figure 2.5c illustrates signals at different points in Figure 2.5a. The symbol stream is first multiplied by the impulse train before convolution with the impulse response  $p(t)$ , resulting in the transmit signal.



**Figure 2.5** Block diagram of a transmitter employing PAM signalling. (a) Symbol stream multiplied by an impulse train before convolution with the transmit filter. (b) Simplified representation with impulse train implied. (c) Signals at different points in (a).

Assume the channel is linear and time-invariant (LTI) with impulse response  $c(t)$ . Further assume a receive filter in the front-end of the receiver with impulse response  $r(t)$ . The effects of both the channel and the receive filter can easily be included into the system model as shown in Figure 2.6. One of the objectives of the receiver is to decide which symbol was sent (at the rate of  $1/T$ ). This requires sampling the receive signal at a minimum of the symbol rate prior to making symbol decisions. The sampler and decision device (shown as a slicer) are shown in the figure with the symbol estimates indicated by  $\hat{A}_k$ . The *slicer* is a block which makes a decision on the sampled channel output. This decision must come from the expected alphabet set. For the case of binary antipodal signalling where the expected symbols at the receiver are +1 and -1, a simple slicer would have its threshold at 0. If the input to the slicer is above 0, it would output a +1. Otherwise, it would output -1. In most receivers, signal processing is performed on the sampler output prior to making symbol decisions in order to increase performance of the decision making process. We shall examine many of these processing techniques later in this chapter.

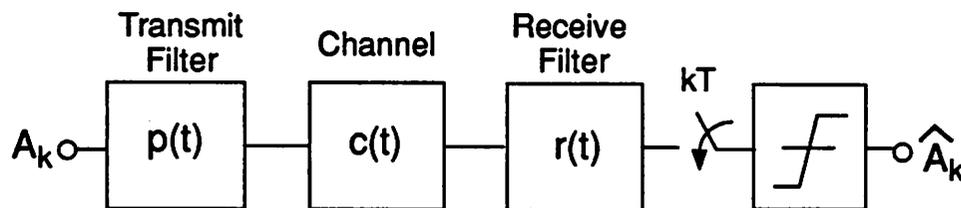


Figure 2.6 Block diagram of communication system including transmitter, channel, and receiver. The symbol  $kT$  over the sampler indicates sampling with a period of  $T$  seconds and  $k$  is a time index. (A two-level slicer is used generically in most of the figures.)

An important advantage of the linear channel model is that all three impulse responses can be combined into an equivalent impulse response  $h(t)$  where

$$h(t) = p(t) * c(t) * r(t). \tag{2.3}$$

Therefore, the block diagram of Figure 2.6 becomes that shown in Figure 2.7 with a single  $h(t)$  which is the convolution of all three impulse responses as given in Eq. 2.3.

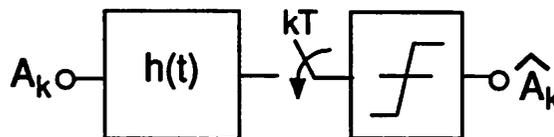


Figure 2.7 Simplified baseband PAM communications system with composite impulse response  $h(t)$ .

### 2.1.2.2 Additive White Gaussian Noise

The electronics in both the transmitter and receiver along with the channel have different noise sources which can result from thermal noise, cross-talk, shot noise, and flicker noise. For ease of analysis, these noise sources are usually assumed to be independent, gaussian, and white and modelled as a single noise source with variance equal to the sum of the variances of the individual noise sources referred to the input of the receive filter. A block diagram of such a communication system model shown in Figure 2.8. This approximation of a single noise source turns out to be quite reasonable for many practical communication systems.

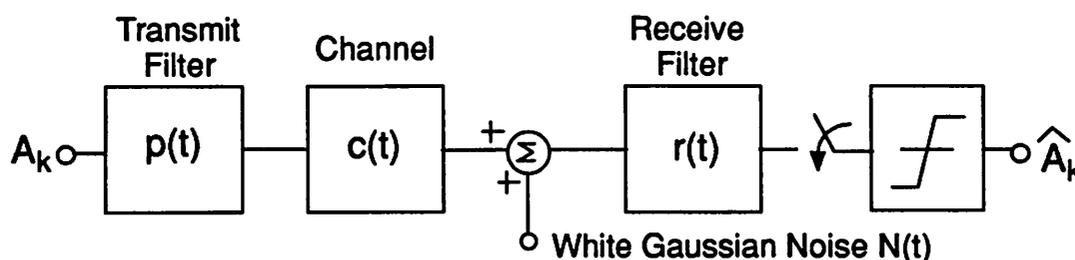


Figure 2.8 Additive white gaussian noise channel model for a baseband PAM communication system.

### 2.1.2.3 Application to the Disk Drive Channel

Let us return to the magnetic disk drive channel using an inductive read head. We will see in Chapter 3 that the magnetization of the media is analogous to write current which is sent to the head during the write process. Therefore, the input signal for saturation of the media in one of two directions is a saturating current waveform of  $\pm I$  where  $I$  is the magnitude of the current. On the other hand, the signal read from the medium is a differentiated and lowpass filtered version of this square wave type signal. Although the read process is inherently non-linear, under the proper conditions linear superposition can be applied to the modelling of this process [13] and the read pulses can be interpreted in the PAM form of Eq. 2.6

$$s(t) = \sum_{k=-\infty}^{\infty} B_k \cdot l(t - kT) \quad (2.4)$$

where  $l(t)$  is the isolated pulse shape (assumed to be a Lorentzian),  $B_k$  is from the alphabet set  $\{-1,0,+1\}$  and is a modified version of  $A_k$  from the alphabet set  $\{-1,+1\}$ , and  $T$  is the period between pulses. We will assume that data bits are simply mapped into  $A_k$  with a data bit 1 becoming a +1 and 0 becoming a -1.

A block diagram of a conceptual model of the write and read process is shown in Figure 2.9a and is simplified in the PAM form in Figure 2.9b where the differentiation associated with the read process is modelled by the 1-D filter in the *coder*. In the 1-D operation, the D represents a unit delay operator such that the output is equal to the current input minus the last input. Notice that due to the 1-D operation, symbols  $A_k$  which are two level become three level  $B_k$  symbols. Non-zero  $B_k$  correspond to a change in the direction of magnetization whereas  $B_k$  equal to zero means no change. The Lorentzian pulse shape turns out to model reasonably well the isolated pulses that result from the entire process. More complicated and accurate models for the isolated pulse shape have been proposed and can be found in the literature [18,19]. The multiplying factor of .5 after the coder simply keeps the peak amplitude of  $B_k$  normalized to unity.

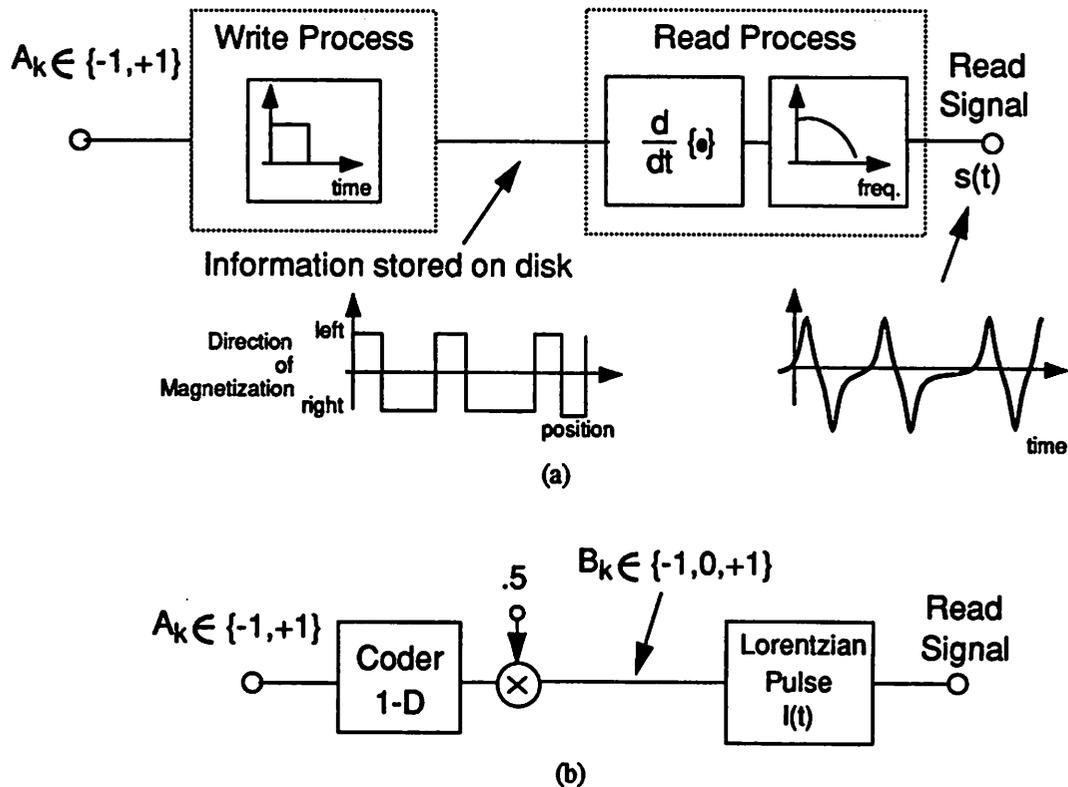


Figure 2.9 (a) Conceptual model of the write and read process. (b) Simplified PAM model.

### 2.1.2.4 Intersymbol Interference

An example of a read signal under the conditions described above are shown in Figure 2.10. As a result of linear superposition, isolated pulses tend to contribute destructively to its neighbors. The effect of a pulse on its neighbors is called *intersymbol interference* or ISI. In general, ISI can contribute both constructive and destructive interference. However, since in the disk drive channel each transition in the direction of magnetization must be followed by the opposite transition before occurring again, only destructive interference can occur. The ISI that a pulse contributes to the samples before and after the main lobe (pulse peak) are called *precursor* and *post cursor* ISI, respectively.

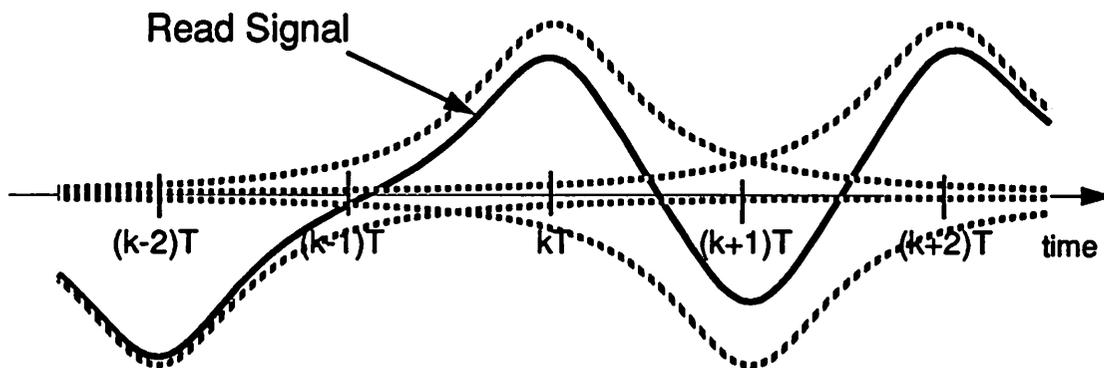


Figure 2.10 Idealized signal from a magnetic disk drive read amplifier.

As can be seen in the figure, ISI tends to reduce the amplitude of the signal which complicates the process of symbol detection. Particularly in the presence of noise, ISI degrades the ability of the detector to make correct decisions. Let us now examine a class of pulse shapes that result in no ISI called Nyquist pulses before proceeding on to the topic of equalization where filtering is used to reduce ISI in practical systems.

### 2.1.3 Nyquist Pulses

In this section, we shall examine Nyquist pulses. Prior to this, it is useful to first consider the discrete-time channel model of a communications system. As discussed earlier, one of the key objectives of a receiver is to determine the symbol sent by the transmitter. This can be a formidable

task in the presence of intersymbol-interference and noise that is accumulated throughout the signal path. In most baseband receivers, the receive signal is sampled at either the symbol rate or twice the symbol rate. It is desirable to sample at the lowest frequency possible because of the processing throughput requirements placed upon the receiver electronics. Let us assume symbol-rate sampling (often referred to as *baud-rate* sampling) in the receiver. There is usually some signal processing before and after sampling in the receiver. The processing that occurs prior to sampling is usually lowpass filtering which bandlimits the additive noise. The subsequent processing usually includes further filtering to reduce the effects of ISI. It is interesting to note that the concern about ISI is not a continuous-time consideration. Instead, the effects of ISI need to be reduced or eliminated only at the sampling instants which may simplify the solution by allowing the application of discrete-time techniques that can be performed in the digital domain. In the discrete-time channel model, we are concerned about the signal only at the sampling instants.

A block diagram of the simplified baseband communications system model is shown in Figure 2.11.  $Q(t)$  is the filtered noise process given by

$$Q(t) = N(t) * r(t) \tag{2.5}$$

where  $N(t)$  is the additive noise source and  $r(t)$  the impulse response of the receive filter. Once again,  $h(t)$  is the convolution of the impulse responses of the transmit and receive filters and the channel. The signal before sampler  $s(t)$  can be written

$$s(t) = \sum_{k=-\infty}^{\infty} A_k \cdot h(t - kT) + Q(t) . \tag{2.6}$$

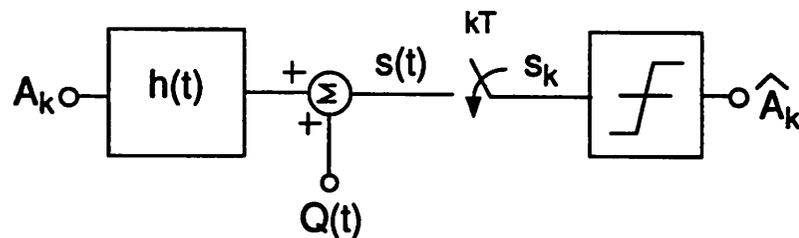


Figure 2.11 Simplified model of Figure 2.7 with the filtered noise process  $Q(t)$ .

After sampling at  $kT$  (where  $k$  is the time index), Eq. 2.6 becomes

$$s(t)|_{t=kT} = s_k = \sum_{m=-\infty}^{\infty} A_m \cdot h(kT - mT) + Q(kT). \quad (2.7)$$

Note in Eq. 2.7 that  $h(t)$  has been converted to its discrete-time counterpart which is simply  $h(t)$  sampled every  $T$  seconds. From this equation it is clear that the equivalent impulse response is of consequence to the sampled signal only at samples of  $h(t)$  taken at  $kT$ .

Receiver sample  $s_k$  can be explicitly written as the sum of the transmitted symbol at time index  $k$  plus the other terms in the summation of Eq. 2.7 plus a filtered noise sample

$$s_k = A_k \cdot h(0) + \sum_{m \neq k} A_m \cdot h(kT - mT) + Q(kT). \quad (2.8)$$

The middle term on the right side of Eq. 2.8 is the intersymbol interference. If the impulse response  $h(t)$  is zero at samples taken at  $kT$  except for  $k$  equal to zero, the intersymbol interference is also zero. Therefore, in order for there to be no ISI, the sampled impulse response must be

$$h(t)|_{t=kT} = \delta_k \quad (2.9)$$

which may include a scaling factor for non-unity  $h(0)$ . Under this condition Eq. 2.8 becomes

$$s_k = A_k + Q_k \quad (2.10)$$

where  $Q_k$  is equal to  $Q(kT)$ .

The sampled impulse response of  $h(t)$  can be written

$$h_k = h(t)|_{t=kT}. \quad (2.11)$$

For zero ISI, we want

$$h_k = \delta_k. \quad (2.12)$$

Taking the discrete-time Fourier transform (DTFT) of both sides [1], we get

$$\frac{1}{T} \sum_{m=-\infty}^{\infty} H(j\omega - jm\frac{2\pi}{T}) = 1. \quad (2.13)$$

This result is rather intuitive as shown in Figure 2.12. It states that for zero ISI, the spectrum of the sampled impulse response which is the convolution of the Fourier spectrum of the impulse response convolved with an impulse train with impulses spaced at the sampling frequency must sum to be constant. Due to the duality between the time and frequency domains, we know that the frequency spectrum of a time domain function that is an impulse must be a constant. This result is called the *Nyquist criterion* and the class of pulses meeting these conditions are referred to as *Nyquist pulses*.

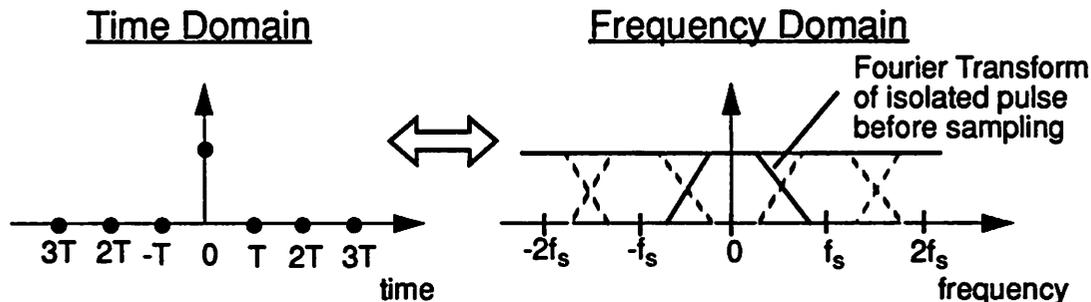
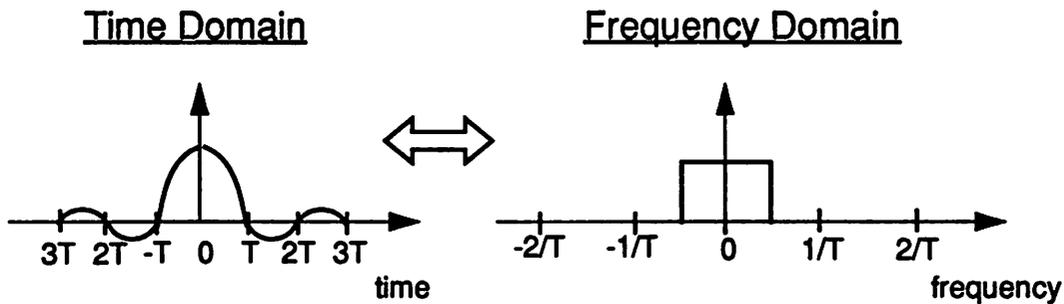


Figure 2.12 Time and frequency domain relationship for Nyquist pulses. The aliases of the Fourier Transform of the isolated pulse after sampling are shown as dashed lines. These aliases sum to be a constant.

The pulse with minimum bandwidth satisfying the Nyquist criterion is the sinc pulse shown below in Figure 2.13. The relationship between the bandwidth  $B$  and the time domain zero crossings of the sinc pulse which occur at intervals of  $T$  is that  $B = 1/2T$ . This means that the minimum bandwidth required to send Nyquist pulses is one-half the symbol rate. Unfortunately, it is not practical to send sinc pulses because of the signal processing required, i.e., to do so would require an ideal brick wall lowpass transmit filter which is not physically realizable. Instead, excess bandwidth up to twice the minimum bandwidth is often used.



**Figure 2.13** Minimum bandwidth pulse satisfying the Nyquist criterion is the sinc pulse shown with its Fourier Transform. The sinc pulse actually extends in either direction and is shown truncated in the figure.

## 2.2 Equalization

In most practical communication systems, the composite impulse response from the transmitter through the receive filter is not a Nyquist pulse and ISI is present. The role of the *equalizer* in the receiver is to perform further filtering on the received signal in order to minimize the effects of ISI [10,27]. The receive filter is usually a lowpass filter with a bandwidth that is roughly that of the receive signal. Since the noise bandwidth is usually much larger than the signal bandwidth, the lowpass filter bandlimits the noise prior to sampling. A simplified block diagram showing the composite channel of Figure 2.11 is shown with the addition of another filter which is the equalizer with impulse response  $e(t)$  before the sampler in Figure 2.14. The equalizer adds its response to the composite channel response  $h(t)$  which can be programmable or adaptive as indicated by the arrow. If it is *adaptive*, the filter can be designed to recursively program itself toward the optimum response based on samples of its own output. We shall examine some adaptive equalizer structures later in this section. The equalizer allows the receiver to correct for variations of the channel that

may arise due to reasons such as changing temperature or distance. It can also correct for changes in the transmit and receive filters.

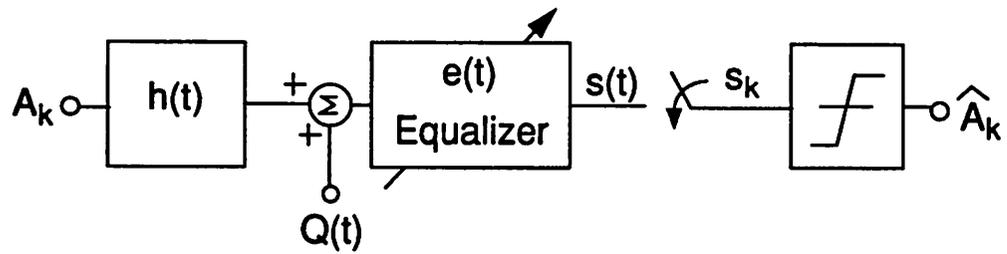


Figure 2.14 Composite model with the addition of the equalizer.

We see in the figure that the noise also passes through the equalizer. Suppose the composite frequency domain response  $H(\omega)$  requires boosting at high frequencies to remove ISI, the noise at these frequencies will also be boosted at the equalizer output. Boosting of the noise is referred to as *noise enhancement*. The equalizer must perform a trade-off between removing ISI and noise enhancement.

Let us now consider an example demonstrating the effect of equalization. Shown in Figure 2.15 is an example of a waveform with alphabet set  $(-1,0,+1)$  before equalization corrupted by ISI. Also shown in the figure are the isolated pulses which make up the composite signal. The dotted lines labelled .5 and -.5 are the thresholds that a simple slicer expecting the three level code would use. Note that samples of the signal at the symbol rate would result in decision errors by the slicer.

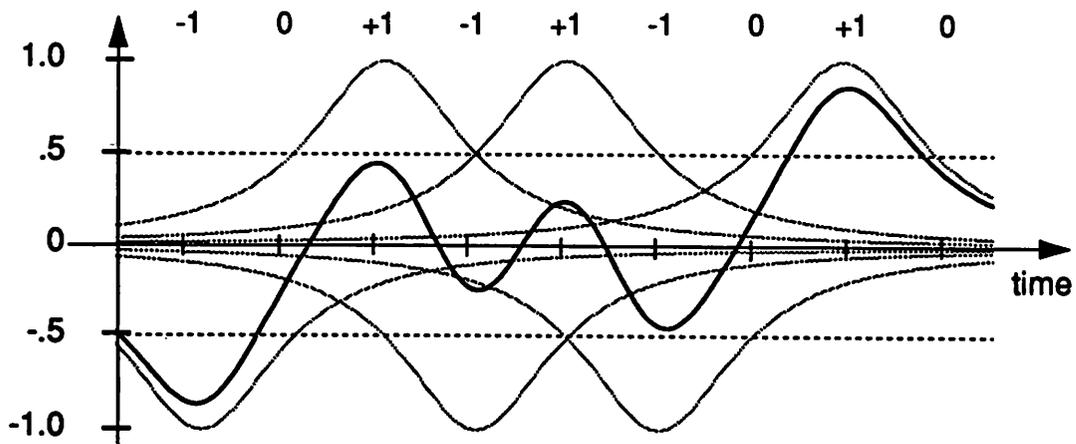
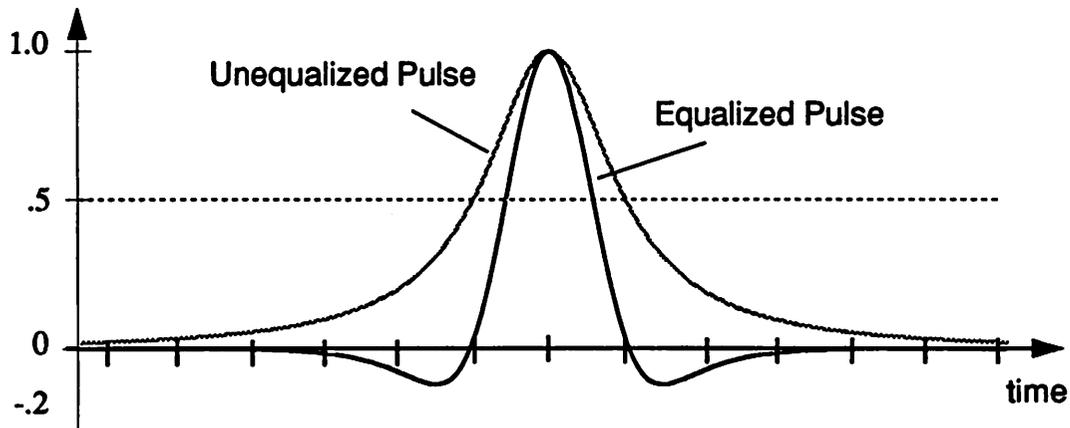


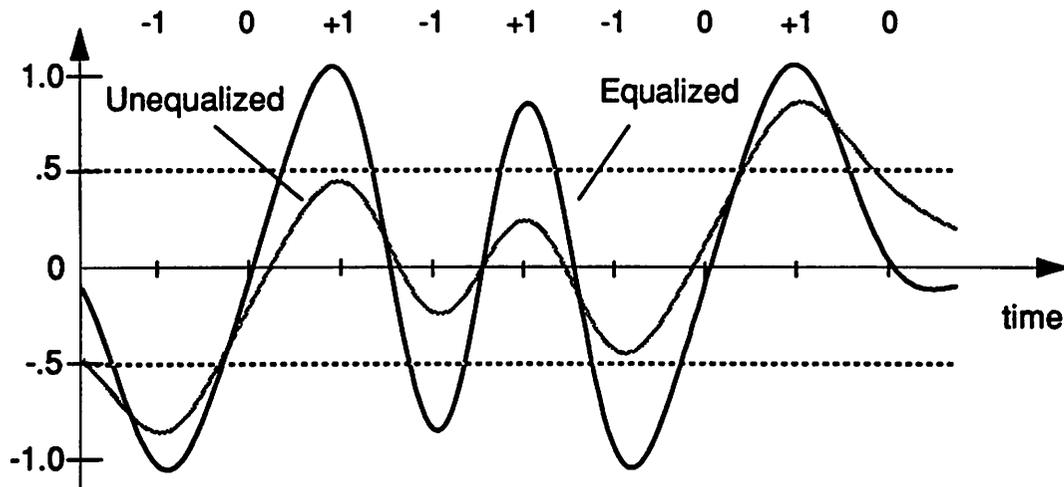
Figure 2.15 Signal before equalization suffering from the effects of ISI. Vertical lines on the time axis indicates sampling instants in the detector.

The signal above was then convolved with a 3-tap equalizer. The result on an isolated pulse before and after equalization is shown below in Figure 2.16. The equalization is imperfect and intersymbol interference is still present, but notice how it has been reduced at intervals of the symbol rate.



**Figure 2.16** Isolated pulse before and after equalization. The equalization used in this example is not perfect but significantly reduces ISI at the sampling instants. Tick marks on time axis indicate the sampling instants.

The signal of Figure 2.15 is shown after equalization below in Figure 2.17. Notice the effect of the reduction in ISI and how it has now made detection with a simple slicer possible. This example demonstrates equalization and how it aids in the detection process.



**Figure 2.17** Signal before and after equalization. Tick marks on time axis indicated sampling instants.

### 2.2.1 Approaches to Equalization

Three of many possible equalizer configurations are shown in Figure 2.18. Equalization can be performed in the analog domain as shown in Figure 2.18a, where it might be folded into the receive filter though it is shown separately in the figure. The equalization can also be performed in the discrete-time domain which may be either analog or digital as shown in Figure 2.18b. It can also be partitioned into both continuous and discrete-time. Another architecture is a feedback equalizer architecture where a part or all of the equalizer is in the feedback path of the slicer as shown in Figure 2.18c.

There are important trade-offs associated with the partitioning of the signal processing in a receiver. Some of the more important trade-offs are discussed in detail in Chapter 4. The purpose of this section is to provide examples of the many ways in which equalization may be performed and to introduce the concept of adaptive equalization.

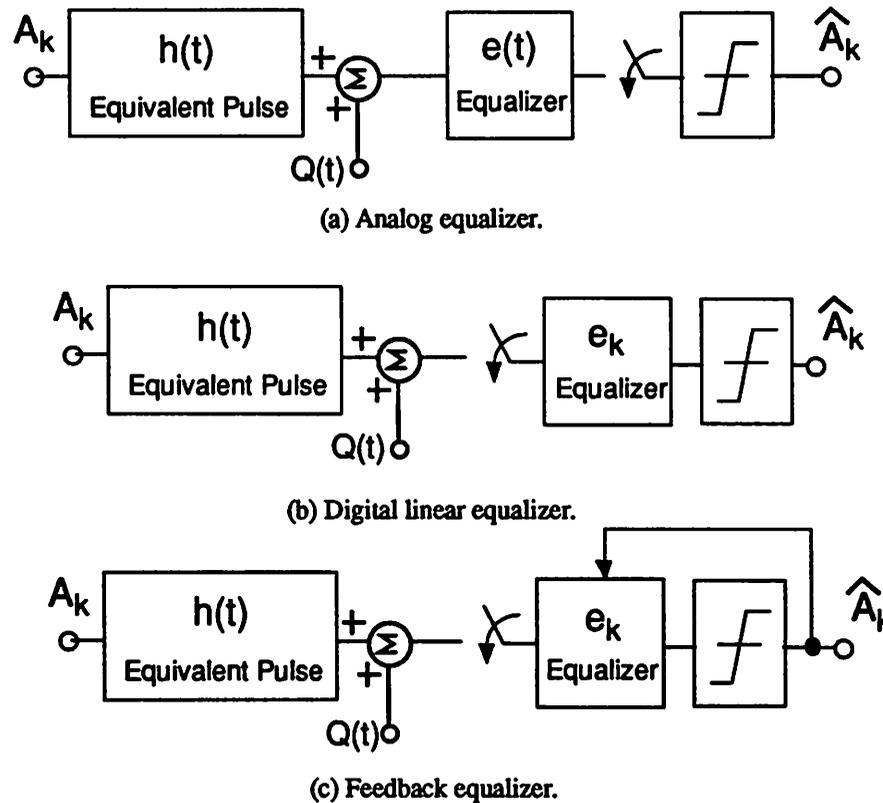


Figure 2.18 Block diagrams of three of the many possible equalizer approaches.

Before we continue, let us first define two of the key measures of equalizer performance. The output of the equalizer is the input to the decision device. Assuming there are no errors in the symbol decisions, we define the error signal  $E_k$  to be

$$E_k = A_k - S_k \quad (2.14)$$

where  $A_k$  is the transmitted symbol and  $S_k$  is the equalizer output. Under the assumption that no decision errors are made, the output of the slicer  $\hat{A}_k$  is equal to  $A_k$  so that the error  $E_k$  may be generated as shown in Figure 2.19 in the receiver using the input and output of the slicer. The mean-square error is defined by

$$\text{mean-square error} = E[|E_k|^2]. \quad (2.15)$$

The mean-square error after the adaptive equalizer converges is a key performance measure of the equalizer.

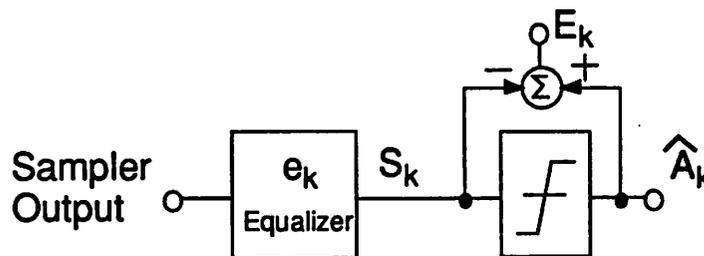


Figure 2.19 Practical method for generation of  $E_k$ .

Another key measure of the performance of an adaptive equalizer is the number of symbol inputs required for the filter to converge; in effect, the convergence time. Let us loosely define convergence as the point at which a time-averaged  $E_k^2$ , which is an estimate of the mean-square error, settles to a steady state value. The choice of equalizer architecture is greatly influenced by the convergence speed required in the application. Wireless mobile communications have quickly changing channel characteristics which require very fast converging adaptive equalizers with wide tracking bandwidths. Whereas phone lines which are the communication channels for voice band

data modems are slowly changing and there is much time at start of the communication session for equalizers in this application to converge. Furthermore, they only need change very slowly over the course of the call.

### **2.2.1.1 Continuous-Time Analog Equalizer**

The analog equalizer before the sampler must be performed by a continuous-time filter. In general, these filters tend to be programmable rather than adaptive. There has not been a very robust practical approach proposed for continuous-time adaptive equalizers. This is primarily because continuous-time filters in general are made up of a cascade of blocks which introduce poles and zeroes into the transfer function. Although there are techniques to move these poles and zeroes to higher or lower frequencies, there is no approach based on a single parameter to drive the many degrees of freedom into a condition to reach a desired state. Furthermore, the poles and zeroes often interact which further complicates matters.

One application where adaptive continuous-time filters have been used is in data transmission over wire pair or coaxial cable channels in the absence of bridged taps (which can greatly change characteristics of the medium) [1]. It turns out that in this controlled case, the length of the cable is the only degree of freedom affecting its response. A fixed equalizer set to equalize for the worst case length is followed by an adaptive equalizer which approximates the response of the difference between the actual and worst case line lengths. By monitoring the peak signal at the output of the adaptive equalizer, the poles in the adaptive equalizer could be controlled together in a negative feedback loop to drive the peak signal to a desired value. Since most practical applications have channels that have many degrees of freedom (i.e., are affected by temperature, length, processing variations, etc...), adaptive equalizers are usually performed using other filtering approaches.

### **2.2.1.2 Discrete-Time Equalization**

There are many ways for implementing equalizers in discrete-time and these can be implemented in both the analog and digital domains. We shall assume digital implementation in this section although a proposal for implementation of an analog adaptive equalizer is presented in Chapter 5. Practical discrete-time equalizers can be programmable or adaptive. If programmable,

control from an external block chooses one of the possible settings as in the continuous-time equalizer case. However, there are many ways to implement adaptive equalizers in discrete-time.

Most practical adaptive equalizers are based on transversal filter structures. This is due to the ease with which these filters can be made adaptive and the flexibility of the adaptation algorithms. In this next section, 4 key adaptive filter structures including the *linear equalizer*, *decision-feedback equalizer (DFE)*, *RAM-based DFE*, and the *fractionally spaced linear equalizer* will be examined. It is useful to have an intuitive feeling for some of the key advantages and disadvantages of each approach.

## 2.2.2 Stochastic Gradient Algorithm

Let us start out this section by examining the *Stochastic Gradient* algorithm also called the *Least Mean-Square (LMS)* algorithm which is one algorithm used to control the coefficients in adaptive transversal filters. The output of a transversal filter can be written

$$y_k = x_k \cdot C_0 + x_{k-1} \cdot C_1 + \dots + x_{k-N+1} \cdot C_N \quad (2.16)$$

where  $x_n$  denotes the input at time  $n$  and the  $C_m$  represents coefficient  $m$  of the  $N$ -tap transversal filter. In the context of an adaptive equalizer for baseband communications channels, the block diagram of Figure 2.20 results.

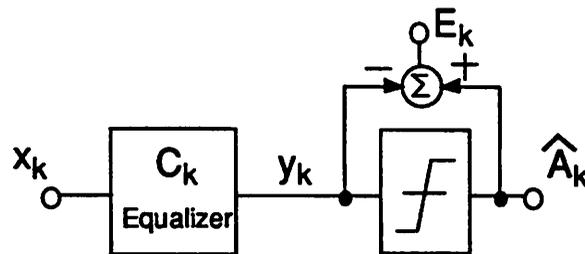
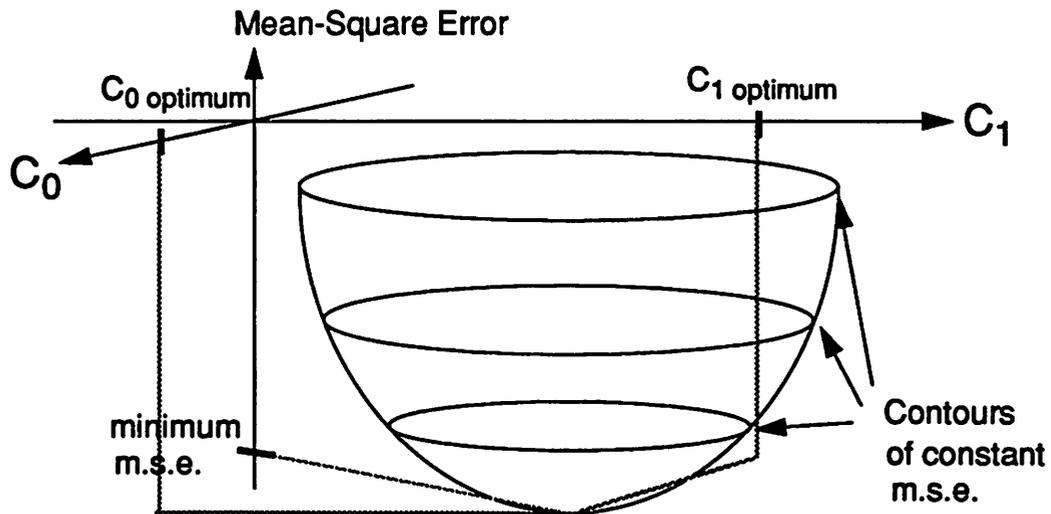


Figure 2.20 Equalizer given by Eq. 2.16 followed by a slicer.

The mean-square error (defined earlier in this section) is a function of the filter taps and it can be shown that under normal conditions found in practical systems, there is a unique set of coefficients  $C_k$  that result in a minimum mean-square error. Let us consider the simple case of a two-coefficient filter shown conceptually in Figure 2.21 where the mean-square error is plotted in the z-direction as a function of different values of the two coefficients  $C_0$  and  $C_1$  plotted in the x and y-directions. The result is a bowl shaped diagram with the bottom point of the bowl corresponding to the optimum value of the coefficients and the minimum mean-square error as labelled in the figure. Also labelled in the figure are concentric circles corresponding to contours of mean-square error. These contours in practice tend to be ellipses [21] but are drawn as circles for simplicity.

Further examination of this structure will provide useful insights into the operation of the SG algorithm for convergence of the coefficients.



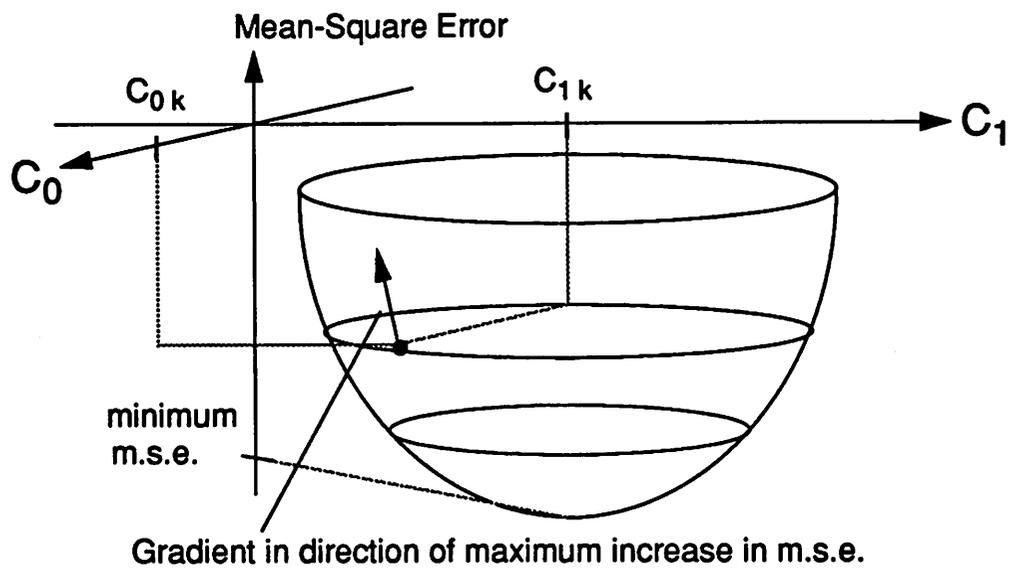
**Figure 2.21** Example of a plot of mean-square error (m.s.e.) as a function of filter tap weights for a two-tap filter.

The *gradient* of the mean-square error  $\nabla_c E[|E_k|^2]$  is a vector in the direction of maximum increase in the mean-square error. Let us refer to Figure 2.22 a where the coefficient values at time

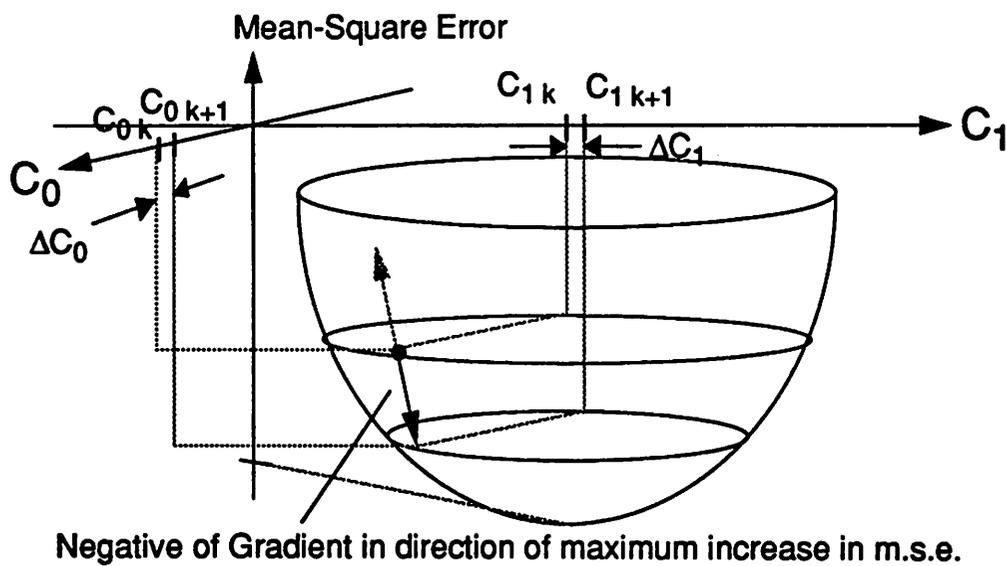
$k$  are shown with the corresponding mean-square error value. Also shown is the gradient of the mean square error. This vector has components in both coefficient axes directions. The SG algorithm operates as follows: a vector pointing in the direction *opposite* to the gradient is added to the current operating point as shown in Figure 2.22b. The result is that the new operating point is closer to the bottom of the bowl. Each of the coefficients are thus calculated by the recursive equation

$$C_{n_{k+1}} = C_{n_k} - \beta \nabla_{C_n} [|E_k|^2] \quad (2.17)$$

where  $C_{n_k}$  is coefficient  $n$  at time  $k$ ,  $\beta$  is a scaling factor which controls the step-size, and  $\nabla_{C_n} [|E_k|^2]$  is the gradient of the mean-square error in the direction of  $C_n$  at time  $k$ . Shown in the figure are the two new values of the coefficients after the update. The magnitudes of the coefficient updates labeled  $\Delta C_m$  are the scaled gradient terms of Eq. 2.17.



a. Gradient of the mean-square error.



b. Negative of gradient used to update coefficients.

**Figure 2.22** Conceptual diagram illustrating operation of the Stochastic Gradient Algorithm for convergence of a two-tap adaptive equalizer.

In practice, it is not possible on the fly to directly calculate the gradient of the mean-square error. Instead, a “noisy” or “stochastic” gradient is substituted for the gradient of Eq. 2.17 which on-average, works to drive the coefficients to their optimum values. The update equation for coefficient  $n$  is

$$c_{k+1} = c_k + \beta E_k x_{k-n} \quad (2.18)$$

where  $E_k$  is the error output at time  $k$  and  $x_{k-n}$  is the input into filter tap  $n$ . This is the Stochastic Gradient Algorithm. One important property of this algorithm is that after convergence, the mean-square error is minimized. This means that the algorithm naturally considers the trade-off between noise enhancement and minimization of intersymbol interference and converges to the condition in which the minimum mean-square error solution is found.

This algorithm like many used in communications system design is *decision-directed*, meaning decisions must be made on the filter output in order to compute the recursive updates. The output of the decision device may or may not be the final decisions output to the receiver. In many cases, further processing may be performed on the equalized samples such as in the case of a sequence detector which we shall examine later in this chapter. However, in order to keep latency in the update process to reasonable levels, a simple threshold decision device is usually used to provide quick decisions for the update algorithms. Errors in these decisions do not tend to have much impact due to the fact that the update process is already a noisy one. It is important however, that particularly during start-up or initial convergence that correct decisions are made or else the decision-directed algorithm may drive the system to an undesirable yet stable operating state. In an adaptive equalizer where zero is a valid output (as in the disk drive channel), a stable yet undesirable operating state might be where all the filter taps converge to zero providing zero at the output. Care must be taken to not allow this to occur either through a known preamble sequence and forcing correct decisions during initial start-up, or constraining the entire channel to be within some limits such that proper operation may be ensured.

There is an important principle called the *orthogonality principle* which states that when the filter coefficients are optimum,

$$E[E_k x_{k-n}] = 0 \quad (2.19)$$

which means that the error is *orthogonal* to the input signal. In words, this means that there is no more information that can be extracted from this cross-correlation. Therefore, the average value of the second term on the right side of Eq. 2.18 is zero after convergence. It can be shown that this term is proportional to an unbiased estimate of the m.s.e. gradient in the direction of coefficient  $n$ . Figure 2.23 shows a block diagram of an adaptive equalizer where each of the  $N$  taps are recursively updated using Eq. 2.18.

By Eq. 2.18, each of the coefficients are adapted independently so each filter tap is in fact independent of all the others. Let us consider by way of a simple example how the recursive equation drives the coefficients to the optimum value. Suppose at time  $k$  that tap  $C_1$  is positive (as it should be), the input to that particular tap  $x_{k-1}$  is positive, and the error  $E_k$  is positive meaning that the equalizer output is smaller than it should be. The algorithm assumes that all coefficients are independent and therefore each partial product contributes independently to the equalizer output. Under the current conditions, the product of  $E_k$  and  $x_{k-1}$  will be positive and result in an increase in  $C_1$ . Intuitively, the algorithm assumes that since the equalizer output is smaller than it should be and the input to the tap and the tap weight are both positive, the tap weight is probably too small. In any time period, the algorithm may drive the coefficients in the wrong direction, but on average it will drive it in the correct direction.

There are many practical considerations regarding the implementation of adaptive equalizers including quantization effects, methods for decreasing the convergence time, etc... which may be found in the literature [21,27]. The purpose of this treatment is merely to provide an introduction to the operation of this extremely important block often used in communications receivers.

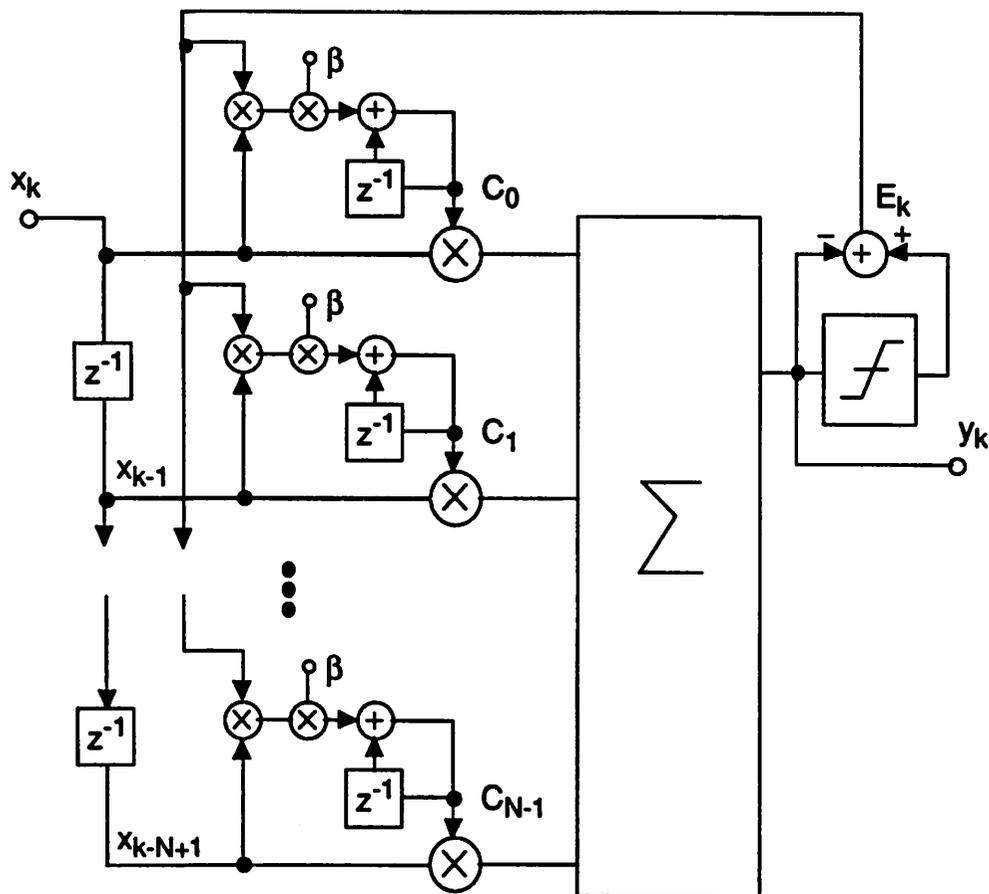


Figure 2.23 Block diagram of an adaptive transversal filter employing the Stochastic Gradient or LMS algorithm.

### Step Size Control

The step size  $\beta$  controls the rate of convergence. The larger the  $\beta$ , the more quickly coefficients change and the quicker the algorithm takes the filter to the region of the minimum mse. However, since convergence depends on the time averaging of a noisy random process, the coefficients continue to change and will wander near their optimum values. For this reason, it is desirable to slow down the movement of the coefficients after convergence to operate near the bottom of the mse bowl of Figure 2.22. Normally, a large step size is used at the beginning in order to decrease the convergence time to get near the optimum coefficient values and decreased after con-

vergence to provide a smaller mean-square error. This technique of changing the step size is called *gear shifting* and is used in many recursive feedback algorithms. Different algorithms exist for optimum control of the step size to improve performance and can be found in throughout the literature.

Three important permutations of the adaptation algorithm that have been shown to converge are shown below. Instead of using the input to the tap weight and the actual error, the sign of one of them or both may be used where the signum (sgn) function is +1 if its argument is 0 or positive and -1 otherwise. These algorithms can be useful in practical implementations because of the reduction in the required amount of computation. Although they can be shown to converge, they may take more time to converge and this is the major disadvantage. The permutations of Eq. 2.17 are

$$c_{k+1} = c_k + \beta \operatorname{sgn}(E_k) \operatorname{sgn}(x_{k-n}) \quad (2.20)$$

$$c_{k+1} = c_k + \beta E_k \operatorname{sgn}(x_{k-n}) \quad (2.21)$$

$$c_{k+1} = c_k + \beta \operatorname{sgn}(E_k) x_{k-n} \quad (2.22)$$

### 2.2.3 Equalizer Architectures

Let us now examine how this algorithm can be applied to some important adaptive filter structures. We shall henceforth refer to the Stochastic Gradient algorithm as the LMS algorithm and the three permutations above as Sign-LMS algorithms.

#### 2.2.3.1 Linear Equalizer

The linear equalizer is simply an adaptive equalizer that precedes a slicer or other decision device. It uses the LMS algorithm or one of its permutations and its structure is that of Figure 2.23. It is the simplest of the four equalizer structures we will describe. The linear equalizer attempts to reduce ISI while keeping in check the noise enhancement associated with any boost at any of the frequencies within the filter bandwidth by converging to the mean square error solution.

### 2.2.3.2 Decision Feedback Equalizer

The decision feedback equalizer has two filter sections as shown in Figure 2.24. One is a forward equalizer which is a linear equalizer section and the other is a feedback equalizer. The input of the feedback filter are decisions from the slicer. These decisions are said to be *noiseless* because in the absence of decision errors, they are the actual symbols that were sent absent of the additive noise from the channel and electronics. These decisions enter the feedback equalizer and post cursor ISI is then removed. The objective of the forward filter is then to remove only precursor ISI. The performance of the DFE is better than that of the linear equalizer because there is less noise enhancement as a result of the noiseless input into the feedback filter.

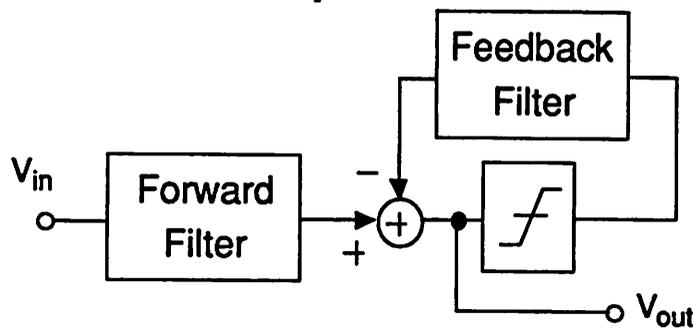


Figure 2.24 Block diagram of a decision-feedback equalizer.

Refer to Figure 2.25 where the signal waveform of Figure 2.10 is shown to illustrate the operation of a feedback filter. The feedback filter after convergence takes the +1 decision at time  $kT$  and the -1 decision at  $(k+1)T$  and multiplies them by an estimate of the isolated pulse response at times  $t=2T$  and  $t=T$ , respectively and subtracts the results from the current input to the slicer. The estimate of the pulse response at these sample times are what the feedback filter should converge to.

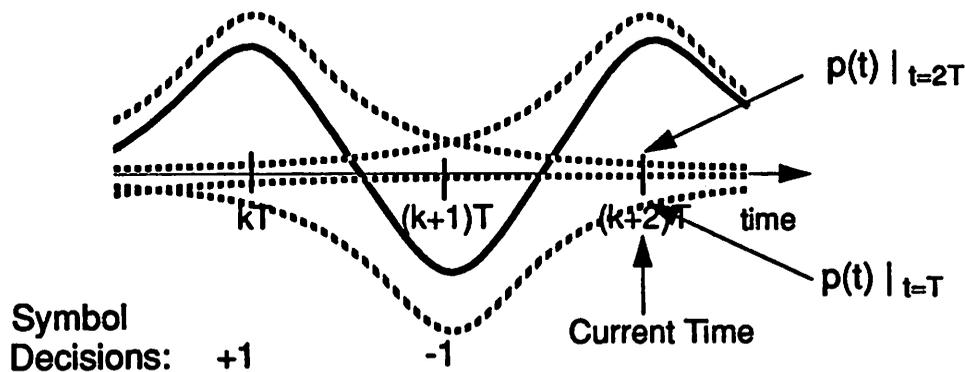


Figure 2.25 Diagram illustrating post cursor ISI of two previous symbol periods on the signal at time  $(k+2)T$ .

A more complete block diagram of a DFE is shown in Figure 2.26. The LMS algorithm is often used to control the filter tap weights. The feedback and forward filters may have a different number of filter taps, depending on the symmetry of the isolated pulse. Another advantage of the DFE structure is that it can be designed to whiten the noise spectrum at the input of the slicer because of the presence of the feedback filter. The noise can be whitened by the forward filter, introducing ISI that is then removed by the feedback filter. Further discussion is beyond the scope of this treatment and can be found in [1,2,21].

One major disadvantage of the DFE structure is that of error propagation. In the case where the slicer makes an error, the erroneous decision passing through the feedback filter back to the slicer input can result in a propagation of the single error event to one or more subsequent symbols. In most instances in which DFEs are used, the length of these errors is bounded and the advantages of the DFE outweigh the disadvantage of error propagation.

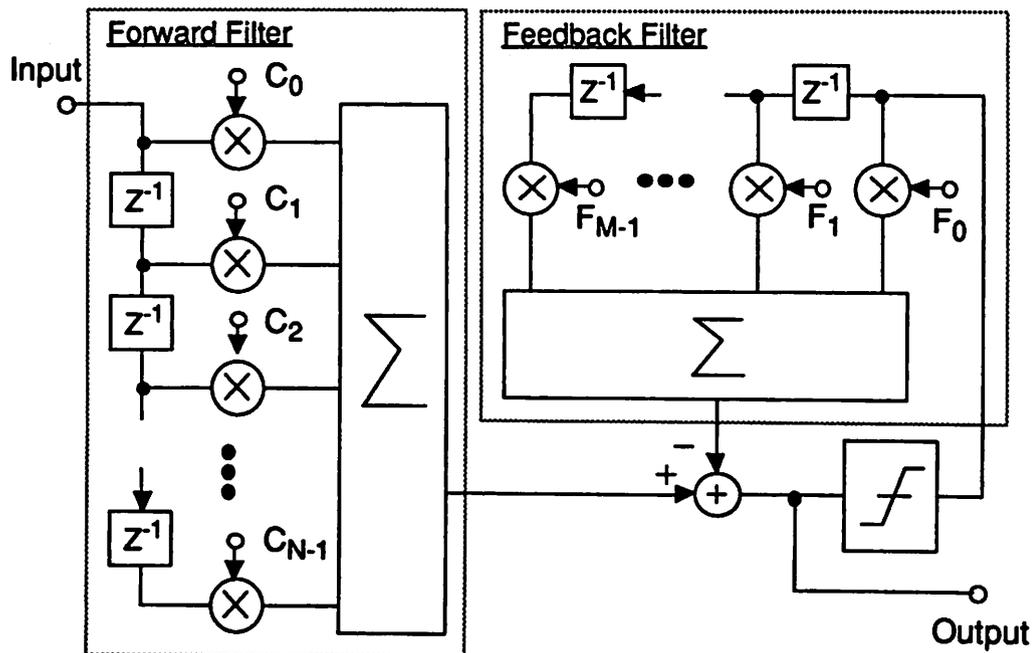


Figure 2.26 Detailed block diagram of a DFE. (Coefficient update circuitry not shown.)

### 2.2.3.3 RAM-Based DFE

The RAM-based DFE was first proposed for non-linear echo cancelers in data modems [22]. It has been investigated as an architecture for the removal of non-linear ISI in magnetic disk read channels [23,24]. In some important communication channels, the model of linear superposition is not entirely accurate and the ISI is a function of the particular symbol sequence. In these situations, the equalizers previously discussed are not able to remove the effects of non-linear ISI. In the RAM-based DFE, the non-linear post cursor ISI is removed using the architecture shown in Figure 2.27. Rather than multiplying the symbol decisions by tap weights as done in the conventional DFE, the decisions make up a RAM address of a memory location which contains an estimate of the post cursor ISI for that particular symbol sequence. This estimate is subtracted off of the output of the forward filter to create the equalizer output. There are different ways to converge such a filter. One way is to begin with the conventional DFE structure and store the linear approximation of the ISI for each possible symbol sequence in the RAM. Then in normal operation, these values can be updated recursively depending on the slicer error. Convergence of the RAM-based DFE is much more complicated than the linear equalizer and the conventional DFE, requiring a complicated architecture and control. This is a major disadvantage of this approach.

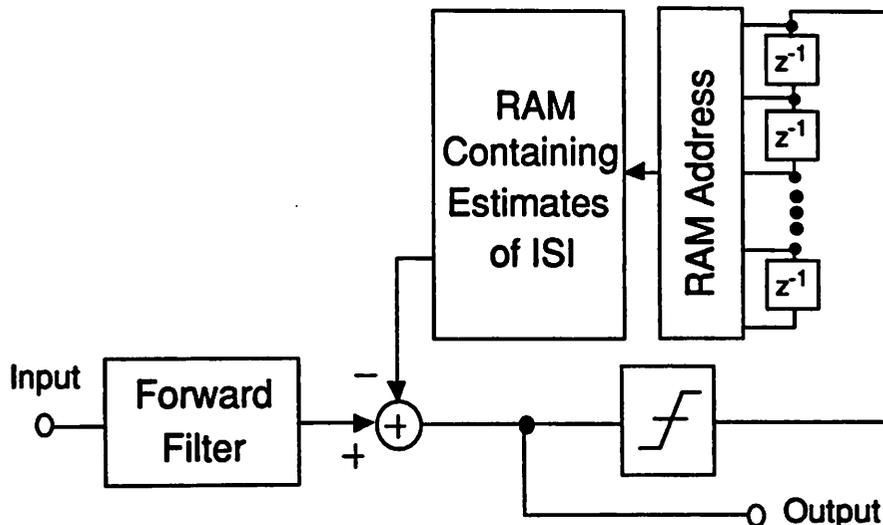


Figure 2.27 Block diagram of a RAM-Based DFE.

#### 2.2.3.4 Fractionally-Spaced Adaptive Equalizer

By the Nyquist sampling theorem, if a signal is sampled at greater than twice the highest frequency present, the original signal can be recreated. A *fractionally spaced equalizer* is a linear equalizer that samples at a *fraction* of the symbol period, therefore at a frequency higher than the symbol frequency. Most practical fractionally-spaced equalizers sample at twice the output rate. There are two key advantages to this architecture.

First, assuming a pulse shape with 100% excess bandwidth or less, sampling at twice the symbol rate is effectively Nyquist sampling. As a result, there is an insensitivity to the actual sampling phase. Normally, we assume that we need to sample at the peaks of the signal. By Nyquist sampling at *any* phase and then equalizing, the equalizer is able to add the time delay which corresponds to a phase equalizing filter to result in outputs normally associated with sampling at the signal peaks. So the first advantage is the insensitivity to sampling phase. This architecture is still sensitive to sampling jitter, but not a constant phase offset.

Second, when sub-Nyquist sampling as is often done in high speed receivers i.e., sampling at the symbol rate, phase distortion in the passband and particularly at frequencies near one-half the symbol rate can result in nulls in the frequency spectrum as a result of aliasing. It is very difficult for symbol rate equalizers to perform equalization to equalize such frequency nulls. Sampling at the Nyquist rate in a fractionally spaced equalizer greatly improves the ability of the equalizer to equalize these channel conditions.

The key disadvantage of fractionally spaced equalizers are the requirements of operating at twice the symbol rate. A common compromise is to use a fractionally spaced forward equalizer followed by a symbol rate feedback equalizer. Although a fractionally spaced equalizer sampling at twice the symbol rate has an effective impulse response that is half as long as a symbol rate counterpart, it has been shown that the FSE will perform at least as well or better [11]. The application of fractionally-spaced equalizers to the magnetic disk drive channel is examined in [12].

#### 2.2.3.5 Other Equalizer Approaches

Another important filter structure is the lattice filter which is particularly important in time varying channels that change very quickly. The particular advantage of adaptive lattice filters are

their ability to adapt quickly. However, their implementation complexity usually limits their application to situations where they are absolutely necessary [21].

## 2.3 Timing Recovery

Up to this point, we have assumed that the sampling in the receiver was occurring in the ideal frequency and phase. The *timing recovery* function consists of generation of a timing function which provides an output signal that is proportional to the phase offset of the sampling phase [25]. This timing function is used in a negative feedback loop (typically a phase-lock loop (PLL)) to drive the sampling phase to be at or near the desired phase. The majority of high speed communications receivers do not employ fractionally spaced equalizers and therefore, performance of these receivers are greatly impacted by the performance of the timing recovery function.

The motivation for the timing recovery function is shown in Figure 2.28. Suppose there was a difference between the transmit clock frequency  $f_{\text{transmitter}}$  and the receive clock frequency  $f_{\text{receiver}}$  of only 1ppm, Then for every million symbols transmitted, a minimum of 1 bit would get lost, immediately placing a high end on the bit error rate of  $10^{-6}$ .

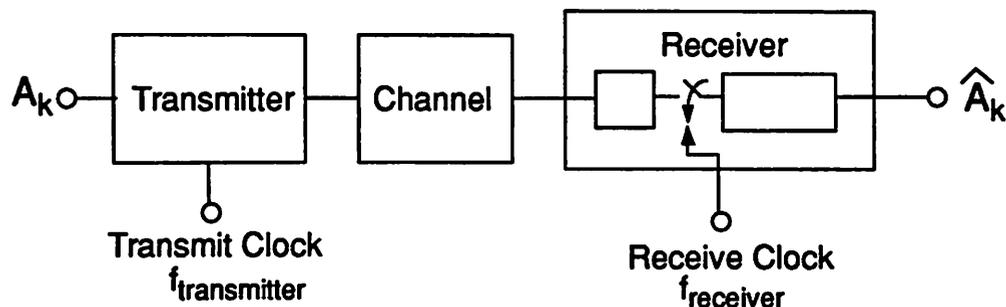


Figure 2.28 Open loop timing recovery where the transmit and receive clocks may not be synchronized.

The approach shown in Figure 2.29 uses a timing recovery loop to drive a voltage controlled oscillator (VCO) so that the phase of the sampler tracks the incoming data signal.

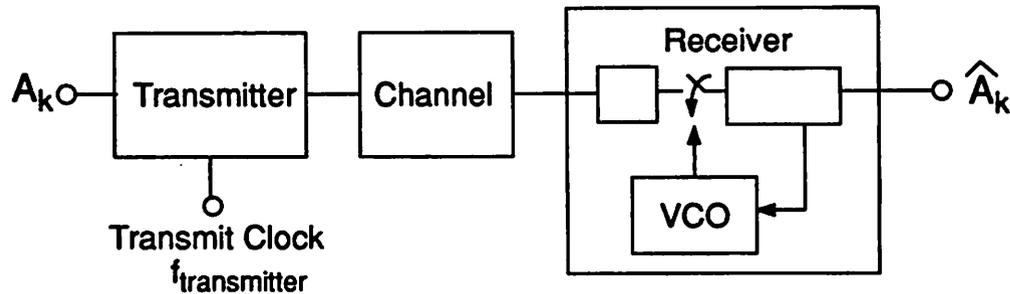


Figure 2.29 Block diagram showing local generation of the VCO clock using samples of the received signal.

There are many approaches to timing recovery. A useful classification of the different approaches can be found in [1]. One approach called the *spectral line method* consist of first passing the signal through a non-linearity and then through a high-Q band pass filter to extract the energy out of the non-linearity near the symbol rate. Most often, this approach requires the use of surface acoustic wave (SAW) filters due to their high-Q and excellent frequency selectivity. Unfortunately, the processing required to manufacture these filters is not yet compatible with CMOS technologies, limiting the applicability of this approach to special high speed applications. Another approach requires the use of excess signal bandwidth in order to simplify the timing recovery function. This method is used in Ethernet where a transition is present in the center of every symbol. Approaches such as these are not applicable to high speed situations, particularly where channel bandwidth is at a premium.

In 1966, Muller and Mueller [25] presented a technique referred to as a *baud-rate sampling* technique which enables samples taken once per symbol period to be used to create what they called a *timing function*. The timing function is a signal whose amplitude is a function of the phase error as shown in Figure 2.30. Discrete-time baud-rate sampled approaches using methods described in [7,25,26] lend themselves best to CMOS implementation since narrow-band filters which are difficult to implement are not required.

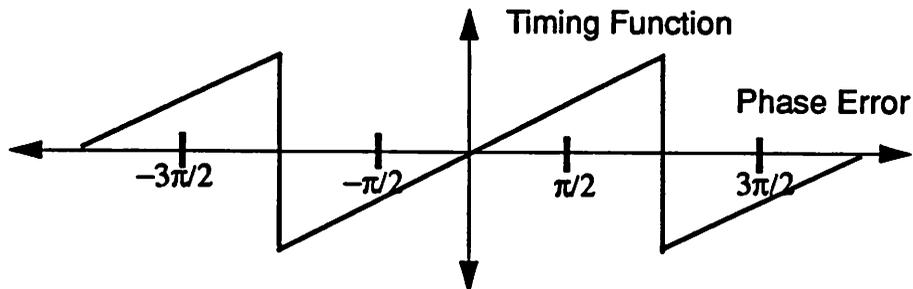


Figure 2.30 Ideal timing function plotted as a function of phase error.

## 2.4 Coding

Since the time we introduced the concept of symbols  $A_k$  in PAM signalling, we have assumed symbols into and out of the communication system. However, *bits* are really what we desire to send and receive. *Coding* consists of conversion of the data bits into the symbols that are sent into the channel. Upon detection in the receiver, the symbols are then converted back into data bits by the *decoder*. This process is shown in Figure 2.31.

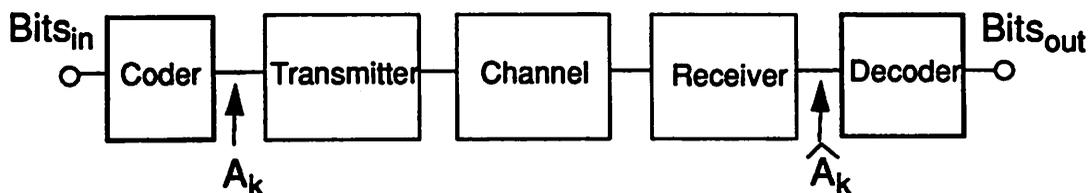


Figure 2.31 Block diagram including the Coding/Decoding functions in one direction of transmission.

In most instances, some form of *redundancy* is introduced into the coded sequence. This means that the number of bits potentially represented by the symbols is greater than the number of bits of information they actually carry. From the block level view as shown in Figure 2.31, coding provides important functions at two levels.

First, it plays an important role in the process of the transmission and detection. Coding can be used to ensure the presence of sufficient information for timing recovery, gain control, and adaptive equalization. Furthermore, it can be used to control the spectrum of the transmitted sequence for optimal transmission through the channel. In other words, it can be used to shape the spectrum of the transmitted signal to be compatible with the spectrum of the channel.

Second, coding can be used to provide reliability from end-to-end ( $\text{bits}_{\text{in}}$ -to- $\text{bits}_{\text{out}}$ ). This comes in the form of error detection and correction which may be used independently or together. As a result of coding, information can be sent with arbitrary accuracy at the expense of bandwidth.

Coding is an extremely important function with a tremendous impact on the overall reliability and performance of a communication channel [8,28]. It is vital to the operation of most of the key receiver functions which depend on coding to provide sufficient information for operation of adaptive algorithms. Coding allows for error detection and correction. Furthermore, it is coding that allows the spectrum of the transmitted waveform to be optimized for a specific channel.

In this next section, we will present the basic principals behind Partial Response signalling which is one class of codes used for spectrum control. Partial response is particularly important to the magnetic disk channel and some of the reasons for this should become clearer in the next section. The treatment presented is a very cursory treatment of an extremely deep topic. Our objective is merely to provide by way of example some of the key advantages of coding and how it can be used to control the spectrum of the signal that passes through a communication channel.

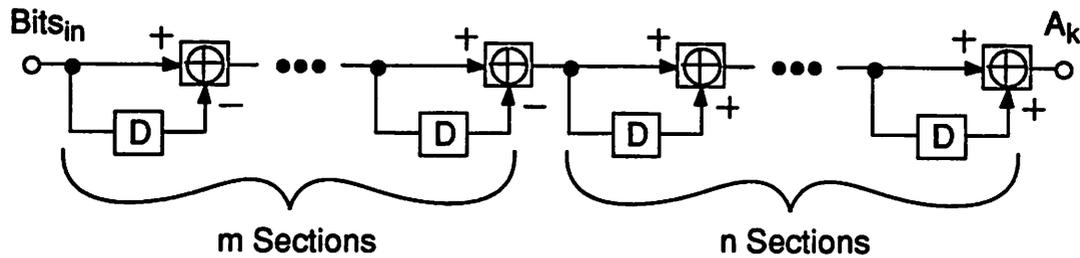
### 2.4.1 Partial Response Signalling

Earlier in the chapter we discussed Nyquist pulses which have no ISI. In Partial Response Signalling, we introduce what is called *controlled ISI* into the transmitted signal. This is accomplished

as shown in Figure 2.32 by passing the bit sequence through a partial response polynomial  $P(D)$  where  $D$  signifies the delay operator. The general partial response polynomial is

$$P(D) = (1 - D)^m \cdot (1 + D)^n \quad (2.23)$$

where  $m$  and  $n$  are integers greater than or equal to 0. Except for the case where  $m$  and  $n$  both equal 0, the output sequence will have a minimum of 3 levels. All arithmetic is performed modulus 2.



**Figure 2.32** Block diagram of a General Partial Response coder. All additions and subtractions are mod 2.

From Eq. 2.23 and Figure 2.32 we can see that for an isolated 1 bit preceded and followed by zeros which corresponds to an input impulse, the corresponding impulse response lasts for more than one symbol period resulting in a known amount of ISI. This ISI is referred to as *controlled ISI* which occurs as a result of the partial response polynomial spanning more than one symbol period.

#### 2.4.1.1 Transmitted Power Spectrum of a PAM Signal

Let us now examine the transmitted power spectrum of a PAM signal. In Eq. 2.2, the transmit signal using PAM signalling was given to be

$$s(t) = \sum_{k=-\infty}^{\infty} A_k \cdot p(t - kT) \quad (2.24)$$

In [1], the power spectrum of this PAM signal is derived and shown to be

$$S_s(j\omega) = \frac{1}{T} |P(j\omega)|^2 S_A(e^{j\omega T}) \quad (2.25)$$

The pulse shape affects the spectrum through the  $|P(j\omega)|^2$  term and the  $S_A(e^{j\omega T})$  term is determined by the input data statistics and shaping of the associated spectrum through coding.

Earlier in this chapter, we modelled the differentiation inherent in the read process of magnetic disk channels employing inductive sensing by the 1-D operation. This in fact is a partial response coder. For an input bit sequence  $b_k$  and output symbol sequence  $A_k$ , the effect of a coder  $P(D)$  of 1-D on the output spectrum can be given as

$$\begin{aligned} S_A(e^{j\omega T}) &= S_B(e^{j\omega T}) |P(e^{-j\omega T})|^2 \\ &= S_B(e^{j\omega T}) |1 - e^{-j\omega T}|^2 \\ &= 4S_B(e^{j\omega T}) \sin^2\left(\frac{\omega T}{2}\right). \end{aligned} \quad (2.26)$$

From Eq. 2.26 we can see how the spectrum of the bit sequence  $S_B(e^{j\omega T})$  is shaped by the 1-D coder. The expression  $\sin^2(\omega T/2)$  has a zero at d.c. and goes to unity at one-half the symbol rate and as a result, a high-pass type transfer function is realized. This result is intuitive because the 1-D operation is analogous to a discrete-time differentiation which in the frequency domain has a zero at d.c. As a result of this particular coder, the symbol sequence has no d.c. component and is thus compatible with many of the communications channels that do not pass d.c.

#### 2.4.1.2 Class IV Partial Response

We shall now examine Class IV Partial Response (PR-IV) signalling, primarily because it is important to the magnetic disk drive read channel. It also illustrates how coding can be used to match the signal spectrum to the particular attributes of a communication channel. The application of PR-IV to the magnetic disk drive channel was first proposed in 1969 by researchers Kobayashi and Tang at IBM [3,4]. We shall examine the application of PR IV to the magnetic disk drive channel in this section and examine prospects of its future in Chapter 3.

In PR IV signalling, the partial response polynomial has  $m$  and  $n$  both equal to 1, resulting in

$$P(D) = (1 - D)(1 + D) = (1 - D)^2. \quad (2.27)$$

Thus, nulls are introduced into the spectrum at d.c. by the  $1-D$  term and at one-half the symbol rate by the  $1+D$  term. Let us now examine the motivations for PR-IV signalling in the magnetic disk channel.

We shall now examine the motivation for the  $1+D$  portion of the PR-IV polynomial as applied to the magnetic disk read channel. For no ISI to occur in samples of the read signal in the absence of equalization, transitions on the disk need to be separated sufficiently far so that the transitions do not interact. The magnetization pattern associated with the no-ISI case is shown in Figure 2.33.

A  $1+D$  coder has the output response to an impulse of two consecutive 1s surrounded by 0s. The same isolated pulse is shown with sampling phases for no ISI in Figure 2.34a and PR-IV signalling in Figure 2.34b. For the no ISI case, the samples are taken in the conventional phases with the main lobe of the pulse response sampled with 0s on either side. Transitions on the disk must be spaced a distance corresponding to a time difference of  $T_{\text{no\_ISI}}$  in order to achieve no intersymbol interference. If we redefine the sampling phases and not sample at the peak of the lobe as shown in Figure 2.34b, two equal valued samples can be taken. With the appropriate gain factor, these two samples can be normalized to unity.

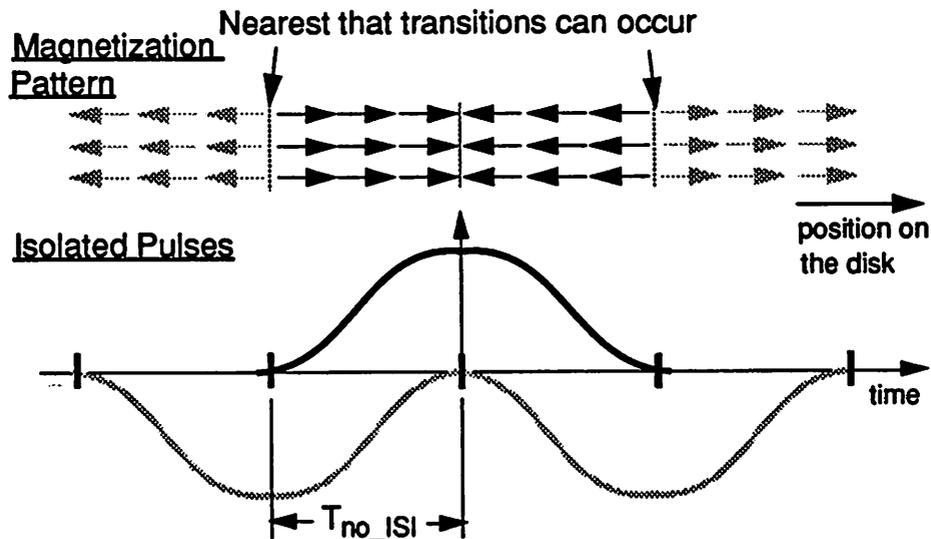


Figure 2.33 Magnetization pattern and corresponding isolated pulses for no ISI. The center pulse and the associated transition in the magnetization pattern is shown with solid lines.

The key advantage of this approach is that

$$T_{PRIV} = \frac{2}{3} T_{noISI} \quad (2.28)$$

which corresponds to the transitions being written on the disk in the case of PR-IV at two-thirds the spacing of the no ISI case. Since for every 2 transitions in the no ISI case, the PR-IV case allows 3, this suggests a 50% increase in density is possible simply by using this different sampling phases of the pulse as is done in PR-IV signalling.

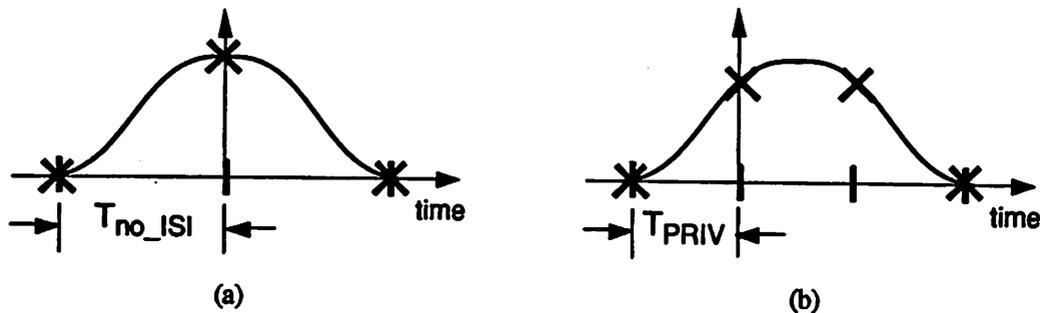


Figure 2.34 Sampling of the *same* isolated pulse for (a) no ISI and (b) PR-IV signalling.

The relationship between the magnetization transition spacing on the disk and the resulting position of the isolated pulses in time as shown in Figure 2.34 for the no ISI case is shown for PR-IV in Figure 2.36. The resulting composite signal assuming all the transitions shown would exhibit much more amplitude degradation than the no ISI case due to the fact that transitions can occur more closely. But since the degradations occur in a known manner, the resulting signal can be properly interpreted by an appropriate detector. In practice, the effect of amplitude degradation and added redundancy in the coding to assure proper operation of the adaptive algorithms results in about a 30% increase in recording density.

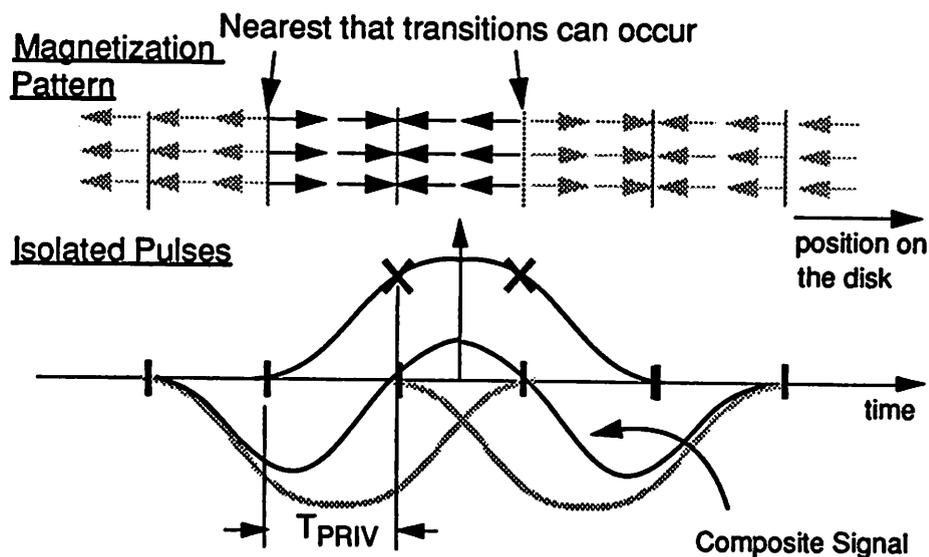


Figure 2.36 Magnetization pattern and corresponding isolated pulses for PR-IV signalling. Center pulse and associated transitions are shown with solid lines.

Class IV Partial Response as stated earlier has a partial response polynomial of  $1-D^2$  which can be written as the product of  $(1-D)$  and  $(1+D)$ . When applied to the magnetic disk channel, the  $(1-D)$  term comes from the inherent differentiation associated with the read process. The  $(1+D)$  term is introduced by use of the sampling phases shown in Figure 2.36. Thus, the  $1-D^2$  comes very naturally to the disk drive channel.

The coding effect performed by the channel must be undone either during precoding of the write process, decoding upon the read, or through a combination of both in order get the output bits the same as the input bits.

Let us now walk through an example of the operation of the coding in a PR-IV channel. Let us begin with a the channel block diagram shown in Figure 2.37. The input bit sequence is first passed through a precoder with the inverse response of the  $1-D^2$  PR-IV response before writing to the disk. The transfer function of the precoder is

$$\frac{B(D)}{A(D)} = \left[ \frac{1}{1-D^2} \right]_{\text{mod } 2} \quad (2.29)$$

Bringing both denominators to the top, we get

$$B(D) \cdot (1-D^2) = A(D) \quad (2.30)$$

Solving for  $B(D)$ , we end up with

$$B(D) = A(D) + B(D)D^2 \quad (2.31)$$

which transforms into the time domain as

$$b_k = a_k + b_{k-2} \quad (2.32)$$

where the addition is performed mod 2.

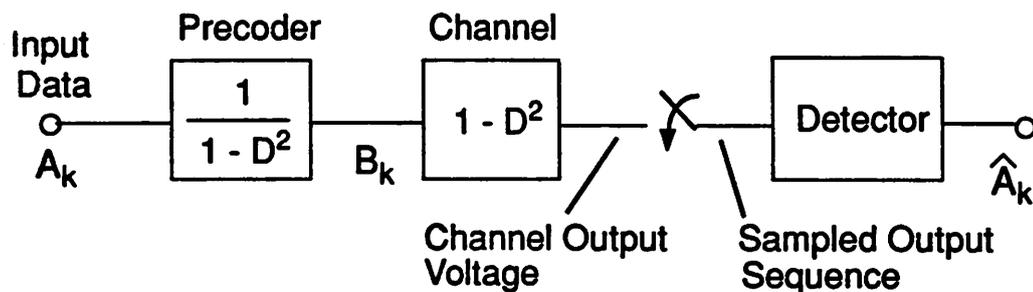


Figure 2.37 Block diagram of PR-IV Channel with precoder.

Let us now run through an example with an bit sequence and observe the signals at progressive stages through the channel. This example is taken from the original paper proposing PR-IV to the disk drive application [3] and is shown in Figure 2.38.

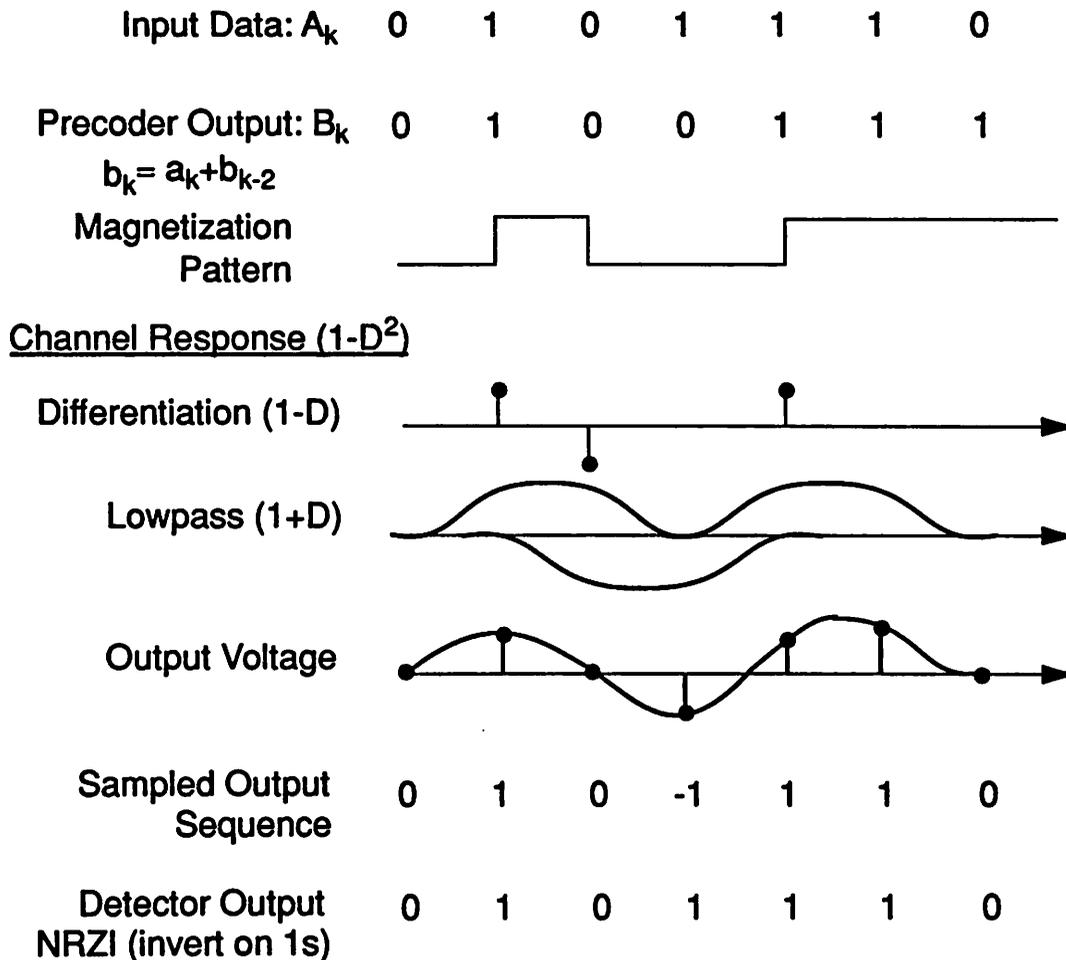


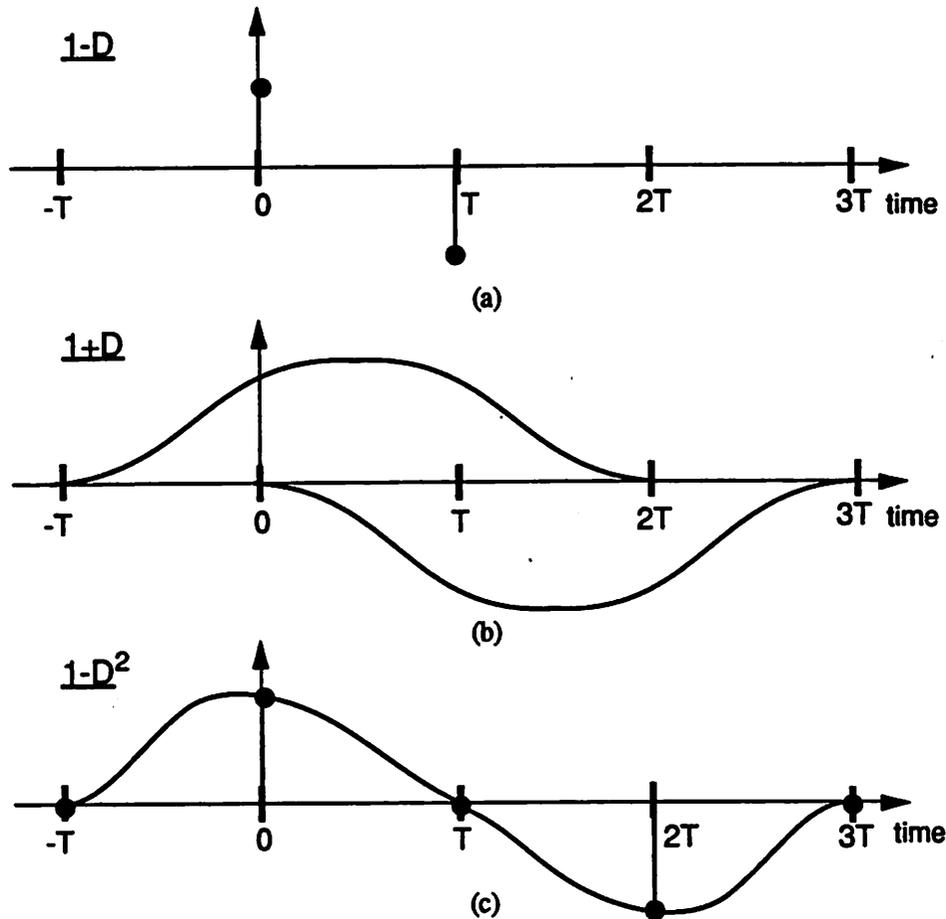
Figure 2.38 Signals at different points in the PR-IV channel of Figure 2.37.

The input data sequence  $A_k$  is first passed through the precoder, becoming the  $B_k$  sequence used to magnetize the media. The magnetization pattern changes polarity with the precoded data, polarized in one direction for a logic 0 and the other for a logic 1. During the read process, processing associated with the PR-IV transfer function occurs on the magnetization pattern with the signal labelled *Output Voltage* in the figure. The PR-IV processing is shown coming from two

steps: first, the differentiation (1-D) and second, the lowpass filtering and redefinition of the sampling phase (1+D). The output sequence is sampled at the symbol rate and the detector maps the output 0s to data 0s and the output +/-1s to data 1s. This mapping of the symbols back into bits occurs due to the NRZI (non-return to zero, invert on 1s) method of signalling with which the application of PR-IV to the digital magnetic channel was developed in [3]. In NRZI recording, changes in the direction of magnetization correspond to data 1s and the absence of transitions, to data 0s.

### ***Bandwidth Requirements and Equalization***

Let us now take a closer look at the bandwidth and equalization requirements for a PR-IV channel. Once again, the impulse response of a PR-IV channel is  $1-D^2$  which is the result of the combination of a 1-D and a 1+D transfer function. In Figure 2.39a, the impulse response of the 1-D portion is shown. There are many pulse shapes which satisfy the 1+D requirement of equal valued non-zero sampled response at  $t=0$  and  $T$ , similar to the way in which there are many practical pulse shapes satisfying the Nyquist criterion which have a non-zero sampled response only at  $t=0$ .



**Figure 2.39** Response of the  $1-D^2$  polynomial to an impulse. (a)  $1-D$  portion resulting in two impulses. (b) Pulse shapes satisfying  $1+D$  response shown after convolution with both impulses in (a). (c) Resulting channel impulse response.

One of these pulse shapes is shown in Figure 2.39b convolved with both impulses from Figure 2.39a. These are then combined in Figure 2.39c. There are a few things worthy of note. First, as in the case of Nyquist pulses, several pulse shapes are possible. The only requirement is that the  $1+D$  response be met at the sampling instants. Second, Nyquist pulses will not necessarily satisfy the requirements of this channel. For zero ISI, a constraint different from the Nyquist criterion must be met, i.e., instead of equalizing to  $\dots,0,0,1,0,0,\dots$ , we equalize to  $\dots,0,0,1,0,-1,0,0,\dots$  for an input impulse or to  $\dots,0,0,1,1,0,\dots$  for an isolated transition. And finally, the  $1-D^2$  transfer function introduces two zeros, one at d.c. and the other at one-half the sampling rate. As a result, in a low-pass channel like the disk drive channel, the equalizer need not provide as much boost near one-

half the symbol rate as it would in the case of Nyquist pulses. This reduces the high frequency boost requirements and thus the amount of noise enhancement at these frequencies.

### 2.4.1.3 Timing Recovery In PR-IV Channels

Now that we have described PR-IV signalling as it applies to the disk drive read channel, let us use this as a vehicle to describe as an example to describe the operation of timing recovery in these channels.

There are two modes of operation for many of the adaptive algorithms in a communication receiver, *acquisition* and *tracking*. During the acquisition mode, the starting state of a block is quickly changed to conform to the input signal. During tracking, small updates are added to the parameters of the block in order to track changes in the input signal. In general, during acquisition, training symbol sequences are normally sent which have known patterns so that the adaptive algorithms can correct the system parameters they control until the desired output is achieved. During this time, step-size parameters and loop bandwidths are normally increased in order to provide fast acquisition at compromised performance. Once acquisition is complete, the receiver moves on to the tracking mode during which time the data symbols are sent. Only small updates are required in the adaptive algorithms during tracking in order to track random perturbations and drift that may occur with time. Therefore, step-size parameters and loop bandwidths are normally decreased. An important trade-off is that between tracking bandwidth and performance.

The timing recovery function normally requires these two modes. We will first present an algorithm used for tracking in PR-IV systems and then one for acquisition. We will see why one algorithm cannot be used during both modes of operation. Both algorithms use equalized samples taken from the output of the adaptive equalizer  $y_k$  and the decisions made by the slicer  $\hat{x}_k$ . These decisions are only tentative and are used only by the adaptive algorithms when a Viterbi detector is employed. In the disk drive application, these decisions are -1, 0, or +1.

#### Algorithm for Tracking

The algorithm used to generate the timing function is

$$\Delta\tau_k = -y_k\hat{x}_{k-1} + y_{k-1}\hat{x}_k. \quad (2.33)$$

The signal  $\Delta\tau_n$  is used to update a second-order phase-lock loop. A block diagram of a system using Eq. 2.33 for timing recovery is shown in Figure 2.40.

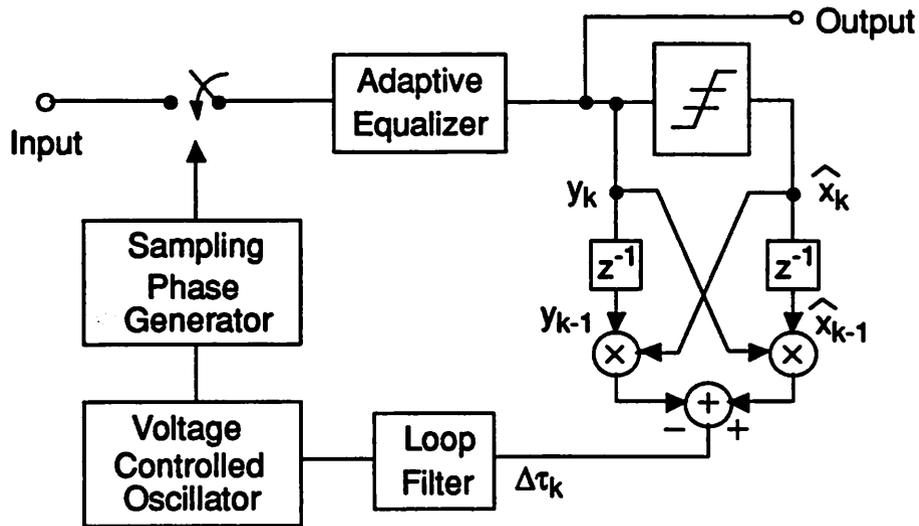


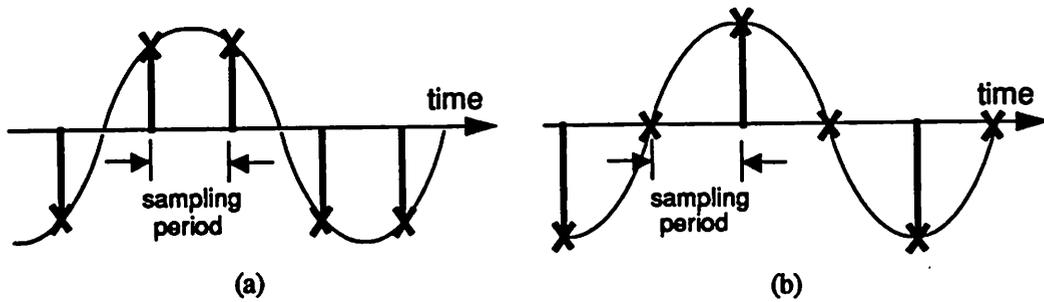
Figure 2.40 Block diagram of timing recovery loop during tracking mode.

#### Algorithm for Acquisition

During acquisition, it is common to “send” a tone at  $1/4T$  which is one quarter of the symbol rate. This provides a large amount of information for the timing recovery block to converge. As can be seen in Figure 2.41, there are two stable operating phases in which Eq. 2.33 can converge. However, the phase shown in Figure 2.41a is the desirable phase which results in the pattern  $1,1,-1,-1,1,1,\dots$ . The phase resulting in no zeros is desired. Therefore, another algorithm is needed which disallows the phase shown in Figure 2.41b which is equivalent to modifying Eq. 2.33 to allow only the symbols  $-1$  and  $+1$  as given by

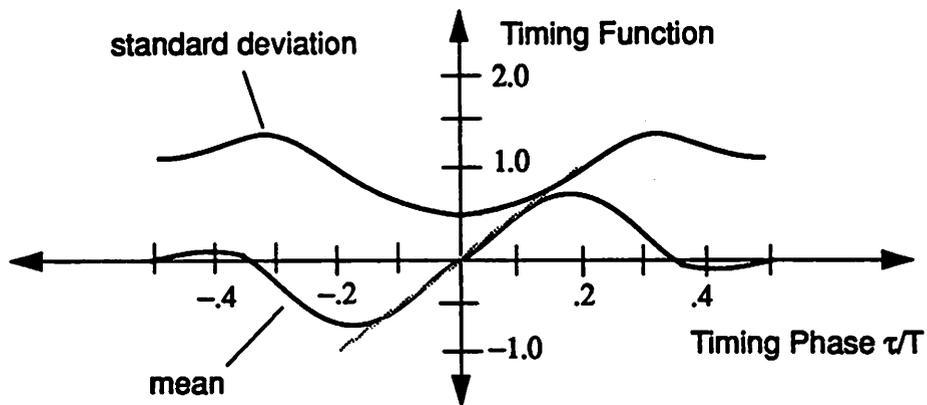
$$\Delta\tau_k = -y_k \text{sgn}(\hat{x}_{k-1}) + y_{k-1} \text{sgn}(\hat{x}_k). \quad (2.34)$$

The block diagram of Figure 2.40 can be easily modified to use the sign of the output and the delayed output.



**Figure 2.41** Possible sampling phases using tracking algorithm during acquisition. Note that sampling periods are the same. (a) Desired sampling phase for PR-IV. (b) Undesired sampling phase.

As described earlier, the output of the timing function generator is a random quantity whose time averaged value drives the output in the correct direction. Figure 2.42 which has been taken from [5] is a plot of both the mean and standard deviation of the timing function as a function of the normalized timing phase  $\tau/T$ . Note the nearly linear range of the mean at around  $\pm 20\%$  phase error and the reduction of the standard deviation in this vicinity. This linear range is what makes operation of the approach possible.



**Figure 2.42** The mean and standard deviation of the timing function plotted as a function of timing phase. (Taken from [5]).

## 2.5 Sequence Detection (Viterbi Algorithm)

In situations in which there is symbol-to-symbol correlation (i.e., symbols are not independent), it is advantageous to look at a sequence of symbols before making a decision rather than making symbol-by-symbol decisions. *Sequence detection* is a detection method in which symbol decisions are made based on a sequence of symbols rather than on a symbol-by-symbol basis. Performance advantages can be achieved in cases where there is a correlation between symbols so that the probability of receiving a particular symbol depends on its neighboring symbols. Correlation can be written into any communications signal via coding techniques resulting in a performance advantage of making symbol decisions based on a sequence of symbols as opposed to symbol-by-symbol decisions. In the magnetic disk drive channel, this correlation in the samples of the read signal comes naturally due to the 1-D associated with the differentiation in the read process.

The Viterbi algorithm is an efficient approach for implementation of practical sequence detectors and has been shown to be the optimum sampling detector in the presence of additive uncorrelated gaussian noise [1,2]. In most practical implementations of the Viterbi Algorithm, an adaptive equalizer is required because the detector works best with equalized samples that have unwanted intersymbol interference removed. *Unwanted* is emphasized because signal correlation due to coding introduces ISI into the signal which is actually desired.

There have been many implementation methods proposed for the Viterbi Algorithm applied to the PRML Magnetic Disk Channel [4,6,7,30]. In this section, we shall examine the Viterbi algorithm as it applies to the disk drive channel taking a far less mathematical treatment than found in most of the literature. We will begin with establishing the concept of Euclidean distance in symbol detection. Following this, we will introduce the state diagram and its relationship to the disk drive channel. We will then observe how a random sequence of bits with 2 levels, becomes a 3 level code (off the disk), and how this relates to the state transition diagram. Following this we will introduce the concept of the trellis diagram which is one way to map a channel output sequence over time which is useful in understanding the implementation of the algorithm. At this point, most of the mechanics of the algorithm will have been presented and we will then walk through an example of the algorithm operating on a state sequence. Finally, we will present a working form of

the algorithm and interpret its connection to what will have been described. Most of the mathematics will be left out of the discussion but can be found in the references.

### Euclidean Distance

One metric used in symbol detection is that of Euclidean distance. Using this metric, the Euclidean distance between a sequence of equalized samples from the channel and all possible sequences are calculated. The sequence with the smallest euclidean distance to the received sequence is said to be *closest* to the received sequence and is output as the most likely sequence that was transmitted. Mathematically speaking, for  $N$  transmitted symbols  $a_k$  and received equalized symbols  $y_k$ , the  $a_k$  sequence which minimizes

$$|y - a|^2 = |y_1 - a_1|^2 + |y_2 - a_2|^2 + \dots + |y_n - a_n|^2 \quad (2.35)$$

is the sequence *closest* in Euclidean distance. The complexity of a receiver evaluating Eq. 2.35 would become extremely complicated for practical values of  $N$ .

In the disk drive channel application, use of the Viterbi algorithm allows the determination during each symbol period of the two most likely sequences called *survivor sequences* up to that particular time. All other sequences can be disregarded<sup>1</sup>. The Viterbi algorithm is based on Euclidean distance and will make decisions based upon this metric.

### State Transition Diagram

The state transition diagram is a useful way to illustrate the relationship between the input bit sequence and the resultant output symbol sequence obtained by sampling the output of the channel which in this case is the read signal. However, before proceeding with a presentation of the state transition diagram it is useful to examine the relationship between the number of *states* and the number of *symbols*. These numbers are not generally the same.

Let us define two states corresponding to each of the possible binary inputs. During each symbol period, the current state (whatever it may be) can either stay the same or change to the other. Suppose the input bit sequence is as shown in Figure 2.43 with the corresponding *write signal* as

---

1. This justification can be found in the literature and will not be presented here.

shown. Upon readback, the signal labelled *read signal* would result, which the receiver would sample at the locations indicated by the dots. Notice that three levels occur in the read signal samples while there are only two possible input bits or states. The three output levels occur in a deterministic way due to the differentiation present in the read process which we model as 1-D. For a 0-1 transition in the input data, a positive peak results while for a 1-0 transition a negative peak results. If there is no transition in the input data, there is no peak. Therefore, since we defined two states corresponding to each of the input bits, there is a peak (positive or negative) every time we change state. The polarity of the peak depends on the polarity of the state change. Through this example we see that a system with only two states can have 3 output levels. The key point is that the three output levels are *not* equally likely but depend on the previous symbol and therefore each symbol contains only one bit of information corresponding to one of the two states.

Since the two level input sequence (input bits) results in a deterministic three level output sequence, given the output sequence, it should be possible to determine the input sequence. This, in essence this is what the Viterbi detector does and does so in a very efficient manner.

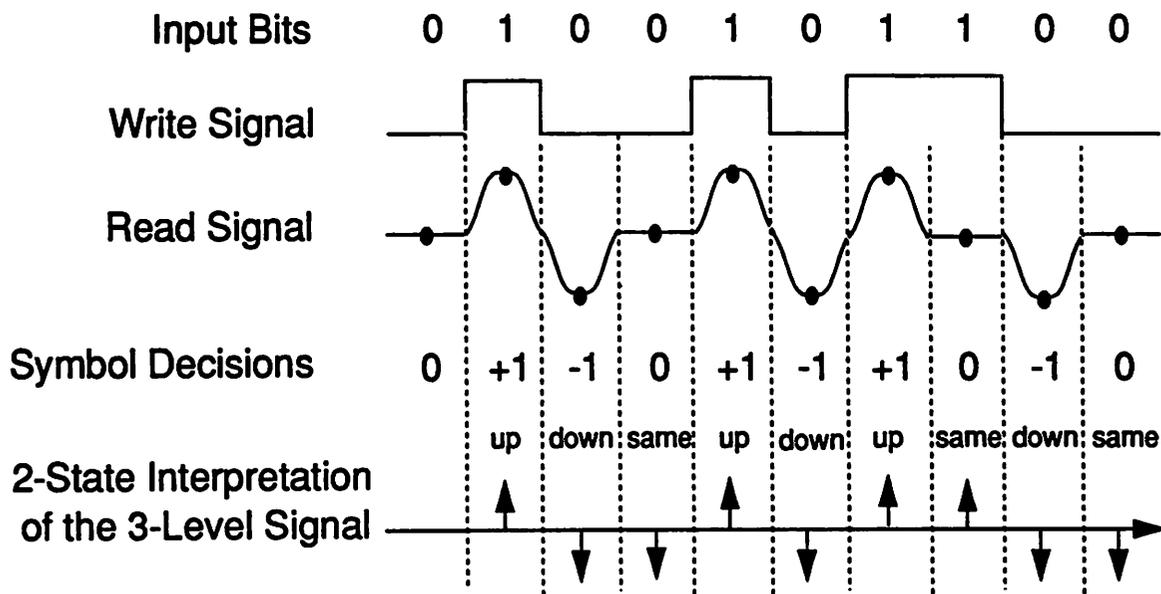
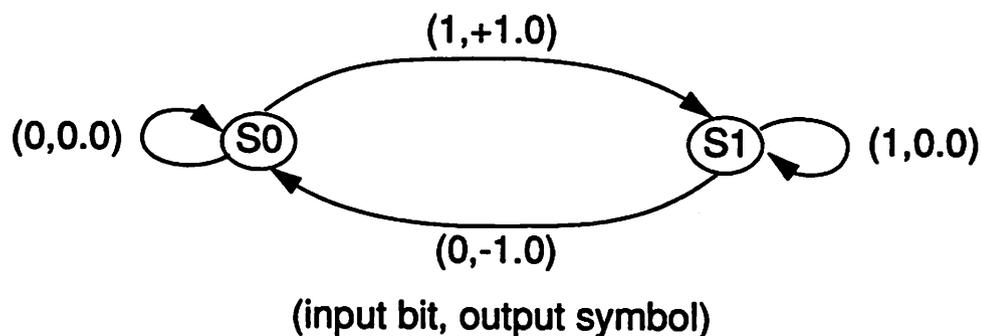


Figure 2.43 NRZ example of input bits resulting a 3 level read signal and a 2-state interpretation that returns it to a two level signal. (In NRZ, a peak results every time the input bits change.)

Let us now refer to Figure 2.44 and examine the state diagram which illustrates the relationship between *states* which we shall define, the input bits, and the corresponding channel output (in the absence of noise and intersymbol interference). Let us define two states  $S_0$  and  $S_1$  which corresponds to the most recent input bit. If the input bit was a 0 (1), the state is  $S_0$  ( $S_1$ ). Now emanating from each state are arcs labelled  $(a_k, y_k)$  where  $a_k$  is the current input bit and  $y_k$  is the corresponding output symbol. The arcs terminate on the state that would result from input bit  $a_k$ . Notice that two of the arcs terminate on the same state whereas two terminate on the other state.

Suppose we start off in  $S_0$  and the input bit is a 0. According to the state diagram, the output symbol will be 0.0 and the next state as well will be  $S_0$ . Now suppose the input bit is a 1. The resulting output symbol will be +1.0 and the state will change from  $S_0$  to  $S_1$ . Let us turn back to Figure 2.43 where we see that when a 0 bit follows another 0 bit, the resulting read signal sample (or channel output) is 0.0. And when a 1 bit follows a zero bit, the read signal sample is +1.0. The state diagram is simply another way to look at the relationship between the input bits and the output symbols. It allows us to easily see the allowed progressions of output symbols and their relationship to the states. For example, if a -1.0 is received at the channel output, the state is certainly  $S_0$  and the input bit must have been a 0. However, if the channel output is 0.0, we would have to wait until a +1.0 or a -1.0 was output before we knew which state we were in. This latter example is important because it illustrates how attempting to guess the state (and corresponding bit) by observing channel output samples requires knowledge of the current state or else an ambiguity can exist. In this last example, if we were in  $S_1$  and received 0.0, we would know that the next symbol received would either be 0.0 again or -1.0 depending on the next input bit.



**Figure 2.44** State Transition Diagram for a disk drive channel employing NRZ coding.

### Trellis Diagram

The trellis diagram allows observation of a channel output sequence traversing the State Transition Diagram over time. Let us refer to Figure 2.45a where the trellis for a two state system is shown. There are two rows of circles corresponding to states  $S_0$  and  $S_1$  as labelled in the figure. The horizontal axis is time. The example shown in the figure assumes  $S_0$  to be the starting state and shows that at time  $k+1$ , the state changes to  $S_1$ , at  $k+2$ , the state returns to  $S_0$ , and so forth. In Figure 2.45b, we see that if we start out in either state  $S_0$  or  $S_1$  at time  $k$ , at time  $k+1$  the state can either change to the other or stay the same.

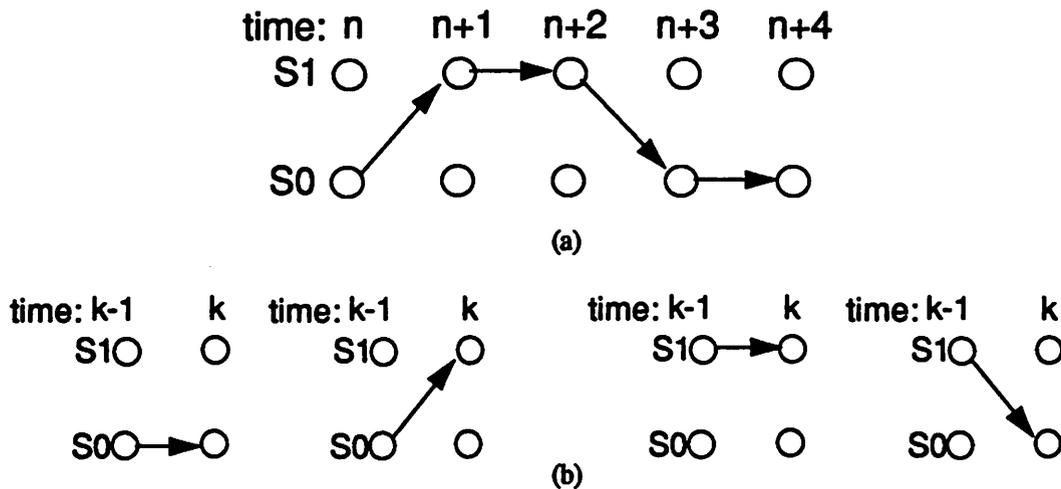


Figure 2.45 Trellis diagram. (a) Traversing a trellis starting in state  $S_0$  at time  $n$  and changing state at each time index. (b) Two paths possible out of each state.

The Viterbi algorithm calculates at each time, the most probable source state into each of the two possible states. For instance, for state  $S_1$ , it determines whether the last state was most probably  $S_0$  or  $S_1$  based on the equalized channel output sample. It does so by calculating what is called a *branch metric* for each of the paths into a node. The branches are defined in such a way that the larger the branch metric, the more probable that particular path. Branch metrics for states  $S_0$  and  $S_1$  are shown in Figure 2.46. Let us illustrate with an example. Returning to the state transition diagram, we see that if the input bit changes from 1 to 0 (state change from  $S_1$  to  $S_0$ ), the ideal channel output will be  $-1.0$ . Suppose the channel output at time  $k$  is in fact  $-1.0$ . Calculating both

branch metrics into S0, the results are 0.0 from S0 and +.5 from S1. The conclusion is that the path from S1 is more probable.

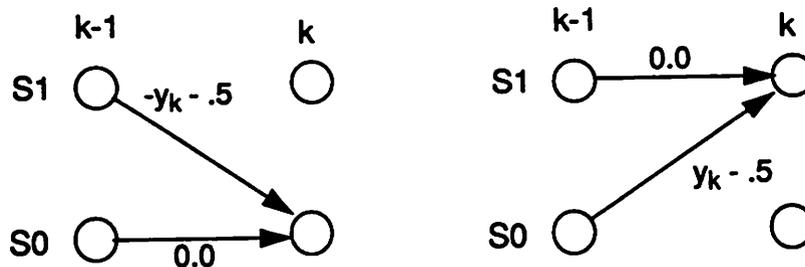


Figure 2.46 Paths into each state labelled with branch metrics.

Branch metrics are calculated for both possible destination states because the Viterbi algorithm at all times keeps the two *most probable paths* (for a 2-state decoder) which are the most probable paths into S0 and S1. Every so often, these paths merge, i.e., both destination states come from the same source state. Shortly we will see what this means.

If we combine the possible pairs of outcomes from Figure 2.46, we find that there are only 3 possible combinations into states S0 and S1 as shown in Figure 2.47. The criss-cross combination of S0 into S1 and S1 into S0 is not possible because  $y_k$  is either positive or negative so one of the branches with branch metric 0.0 will always be chosen.

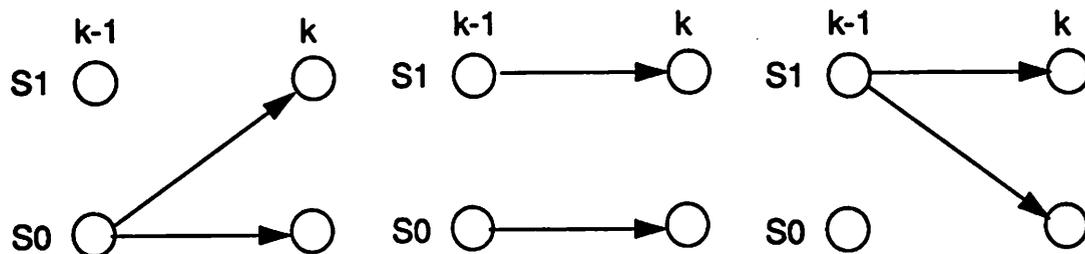


Figure 2.47 Three possible combinations of state transitions for metrics shown in Figure 2.46.

We see in the two outside combinations of paths that both paths come from the same state. When this occurs, both of the *most probable paths* pass through the same node which means that this node is most likely in the correct path and all states leading to that path must therefore be in

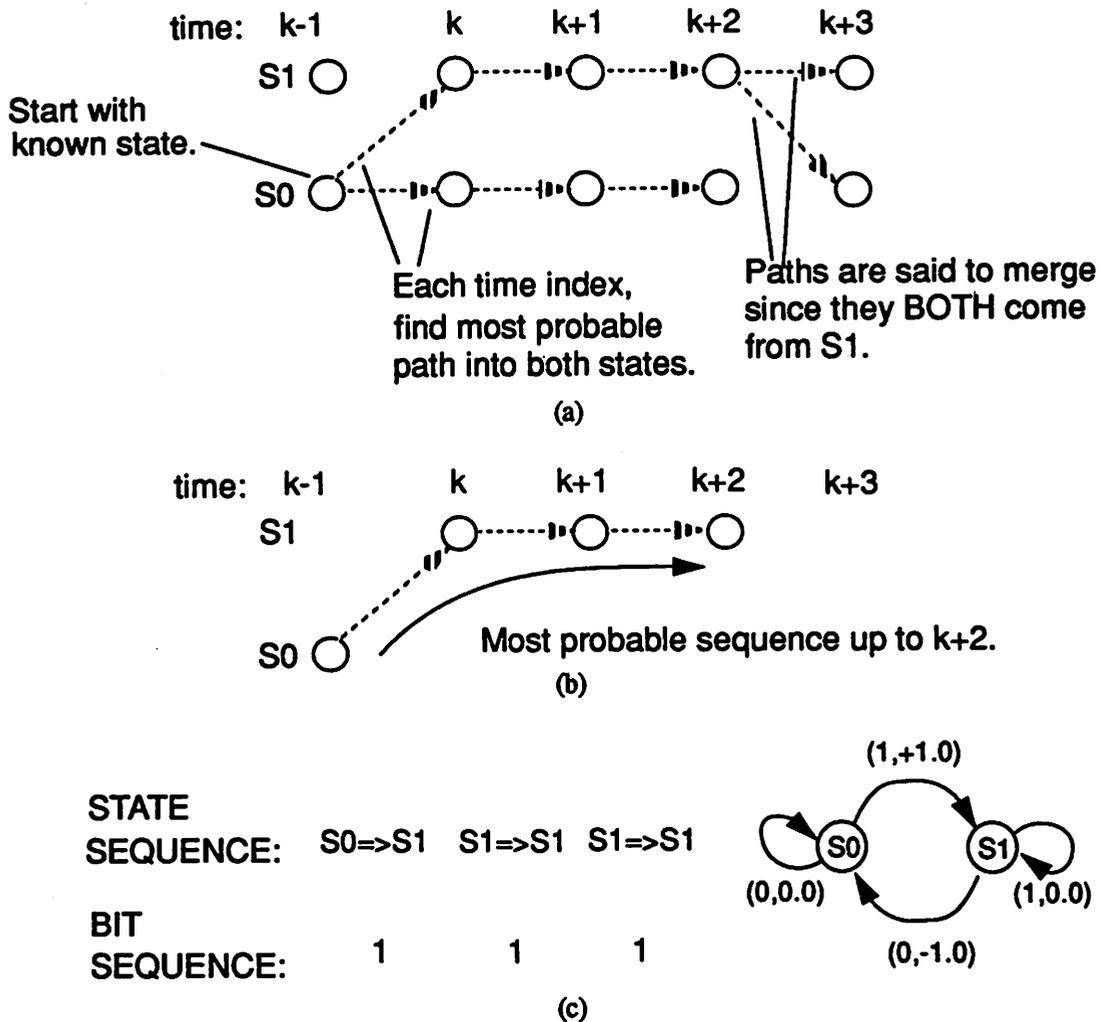
the correct sequence of states. These states are then mapped into bits and output from the sequence detector.

Let us now summarize the key points of this discussion of the mechanics of the Viterbi algorithm. Following this, we will look at an example before getting into the implementation aspects of the algorithm. The mechanics of the Viterbi algorithm for the two-state decoder requires understanding of the following:

- There is a relationship between the bit sequence, the channel output samples (equalized and noiseless for our discussion), and the state sequence.
- A bit sequence translates into a symbol sequence which results in a unique path through a trellis.
- On the receiving end, we do not know the bits, so instead using each new channel output we find the most probable path into the two possible states. The starting state must be known. Two separate paths may develop. When they finally merge, i.e., both emanate from the same state, the sequence of states leading up to that state is the most likely path since both paths agree at that point. The path traversing the sequence of states up to that point is translated back into bits.

Let us walk through an example to illustrate this more clearly. Refer to Figure 2.48 where an example showing the selection of a most probable sequence is shown together with the translation of that sequence back into a bit sequence. First, we start in state  $S_0$  as shown in Figure 2.48a. At each time index, a new channel output arrives and the branch metrics are calculated. The two largest metrics into both states are retained. These are called *survivor metrics*. This proceeds with each new channel output until the paths merge as shown in this case at time  $k+2$  (requires input at time  $k+3$ ). At this point, we move to Figure 2.48b where the surviving sequence is shown. This state sequence is then converted back into bits using the state transition diagram as shown in Figure 2.48c. The state sequence started at  $S_0$  and changed to  $S_1$  at time  $k$ . In the next two periods, the state did not change. In the state diagram we see that the  $S_0$  to  $S_1$  transition corresponds to an input bit of 1. Similarly, staying in  $S_1$  for two periods means the corresponding input bits were also both 1s. Therefore, the output bit sequence is 1,1,1 as shown in the figure.

Now that we understand the mechanics of the algorithm, we will present the algorithm itself without the mathematical justification which can be found in the literature.



**Figure 2.48** Mechanics of the Viterbi algorithm based on the state sequence. (a) Two most probable paths at each time index starting from state  $S_0$ . (b) Most probable sequence after the merge. (c) Mapping state sequence into bit sequence using state transition diagram.

Before presenting the algorithm and some implementation aspects, it is useful to describe the parallel-shift and load circuit which is an important building block in implementation of the algorithm. This block is essentially a D Flip-Flop with a second input. Upon a trigger from the input clock, it either shifts its D input to the output Q (parallel shift) or passes its N input to the output. The multiplexing occurs via control external to the block.

In an actual implementation of the Viterbi algorithm, the detector does not wait until a merge occurs and then output all the bits up to the point of the merge. Instead, there are two rows of parallel-shift and load blocks (which we will refer to as *two rows of flip-flops*), the depth of which exceeds the number of symbols for which a merge is guaranteed to occur (controlled through coding [28]). During operation, a portion of each row of flip-flops contains the two state sequences up to the point of the last merge. The rest of both rows contain the *same* state sequence. The reason for this is that upon a merge, only one string will have the most probable sequence. This correct sequence is parallel loaded to the other row of flip-flops (over-writing the sequence that was there) resulting in both rows containing the same information. The output can therefore be taken from either row of flip-flops because both should contain the same data by the end due to merges. Thus, one bit is output during each period.

Let us refer to Figure 2.49 where an example state sequence corresponding to a channel output is shown on the trellis over time. Right below the trellis are the two rows of flip-flops, the bottom of which corresponds to  $S_0$  and the top to  $S_1$ . At each increase in the time index, a 0 and a 1 are input into the bottom and top rows of flip-flops, respectively. This is analogous to the two paths entering both states  $S_0$  and  $S_1$  during each increase in the time index. This is shown in Figure 2.49a for a sequence starting in state  $S_0$ . At time  $k+1$ , another 0-1 pair are input at the left end, and the 0-1 state pair from time  $k$  is shifted right. Note that the time index corresponding to the flip-flop outputs are shown in all the figures. Finally, at time  $k+1$  there is a merge at  $S_0$  as shown in Figure 2.49c. This means that the correct path most likely passed through  $S_0$  at times  $k$  and  $k+1$ . As a result, the two flip-flop outputs in the bottom row corresponding to state  $S_0$  at time  $k$  and  $k+1$  are shifted upwards into the top row of flip-flops so that both rows contain 0s at time  $k$  and  $k+1$ . The state of the system at time  $k+5$  is shown in Figure 2.49d. There were also merges at time  $k+2$  and  $k+4$  which are reflected in the state information present in the two rows of blocks. Only one merge needs to occur between the time a 0-1 pair is input from the left and reaches the end. The number of flip-flops in a row is often referred to as the *memory depth* and must be designed so that the probability of no merge occurring is arbitrarily small. This is related to the coding that is employed on top of the channel coding (i.e., for timing recovery, error correction, etc...) [7,28].

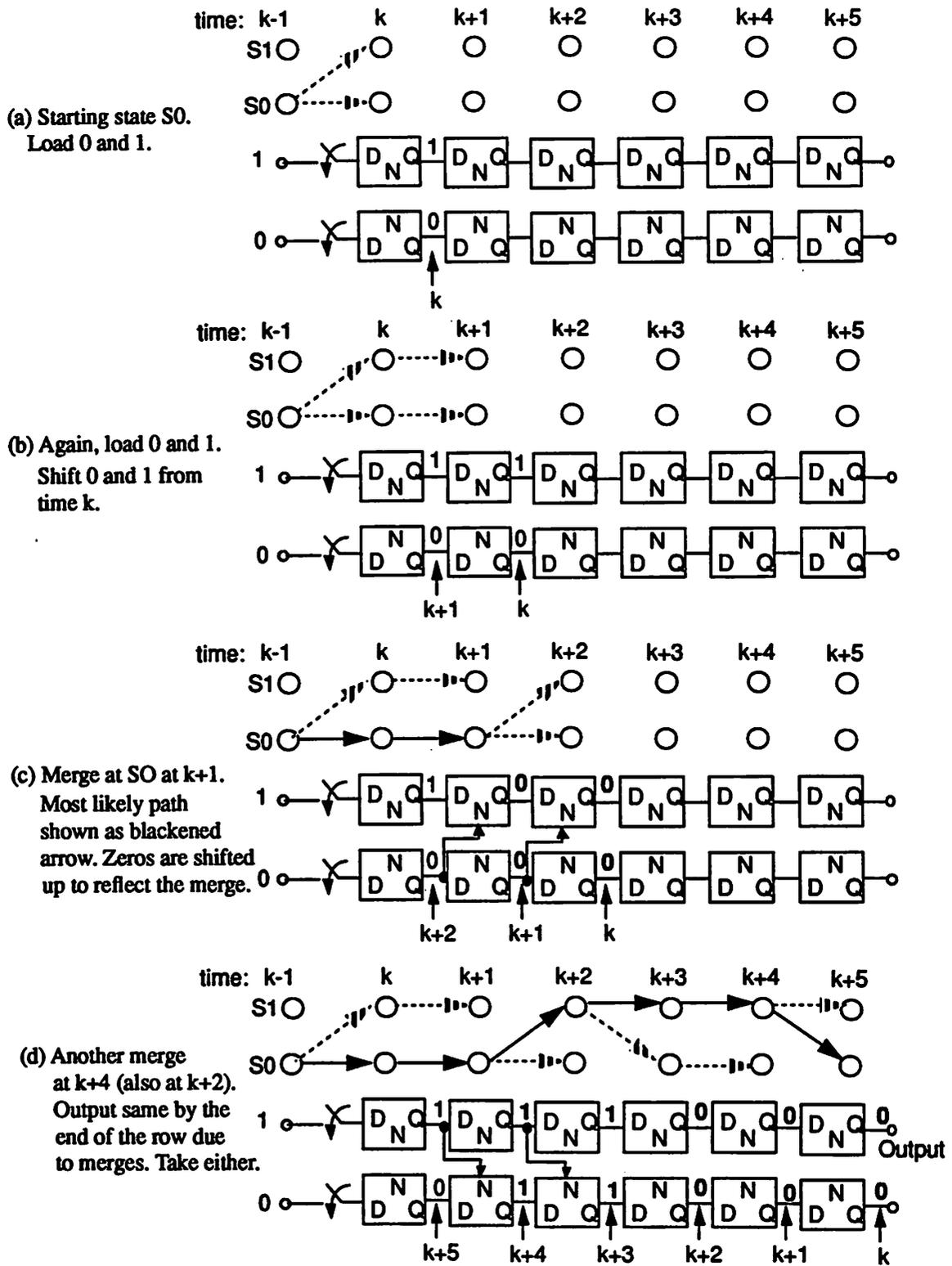


Figure 2.49 Example of operation of parallel-shift and load. Upper row of flip-flops corresponds to state S1 and the lower to state S0.

From an implementation standpoint, the Viterbi algorithm is simply a control algorithm for the two rows of flip-flops. Assuming the starting state is known, the algorithm needs only to determine which of the three combinations of state transitions as shown in Figure 2.47 was most likely. The algorithm must keep track of the current state because calculation of the path metrics depends on what state the system is (thought to be) in. For example, if the channel output is  $-1.0$ , it is more likely that the next channel output will be a  $0.0$  or  $+1.0$ . Therefore, knowledge of the current state is important. Since in NRZ coding, the state only changes when a  $+1.0$  or  $-1.0$  is received (in an idealized noiseless model), the detector essentially *looks* for these transitions. Therefore, if the last merge occurred in state  $S_0$  upon a  $+1.0$  channel output, the detector begins to “look” for a  $-1.0$ . If a  $0.0$  should come along, it does not change the state so in some sense is inconsequential (although important from the standpoint of the data!).

A block diagram of the detector is shown in Fig. 2.50. There are essentially two blocks. One is the *Control Block* which operates on the channel output, keeping track of the current state and determining which of the 3 combinations of transitions was most likely. The other is the two strings of flip-flops controlled by the Control Block. The contents of the two strings of blocks contains the two most likely paths and become equal after a merge.

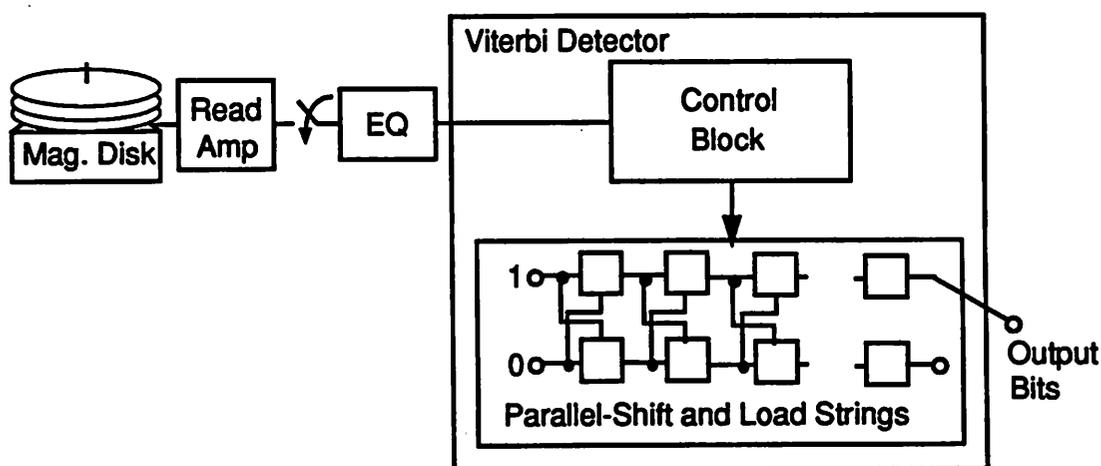
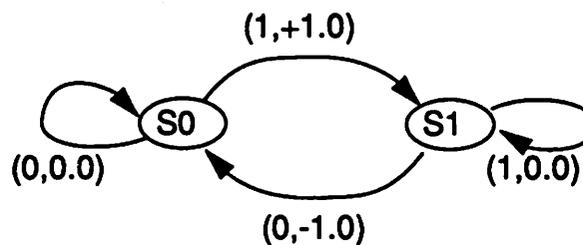


Figure 2.50 Block diagram illustrating the two key blocks in a Viterbi detector.

Let us now examine a version of the Viterbi algorithm<sup>1</sup> used by the Control Block which operates on the sampled equalized output of the channel. Since the current state is not known precisely, the algorithm uses information from the *most recent merge*. Instead of comparing the current input to a set of references as done in a slicer, the control block compares the quantity  $y_k - y_p$  with a set of conditions where  $y_k$  is the current channel output and  $y_p$  was the channel output at the time of the most recent merge. Depending on the most recent merge, a different set of conditions for comparing  $y_k - y_p$  are used. These conditions along with the corresponding outputs are shown in Figure 2.51.



Most Recent Merge in State S1			Most Recent Merge in State S0		
	Condition	Output		Condition	Output
1	$y_k - y_p > 1.0$	s1 $\Rightarrow$ $y_k \Rightarrow y_p$ s0 $\Rightarrow$ Merge in S0	4	$y_k - y_p < -1.0$	s1 $\Rightarrow$ $y_k \Rightarrow y_p$ s0 $\Rightarrow$ Merge in S1
2	$1.0 \geq y_k - y_p > 0.0$	s1 $\Rightarrow$ $y_p$ same s0 $\Rightarrow$ No Merge	5	$-1.0 \leq y_k - y_p < 0.0$	s1 $\Rightarrow$ $y_p$ same s0 $\Rightarrow$ No Merge
3	$0.0 \geq y_k - y_p > -1.0$	s1 $\Rightarrow$ $y_k \Rightarrow y_p$ s0 $\Rightarrow$ Merge in S1	6	$0.0 \leq y_k - y_p < 1.0$	s1 $\Rightarrow$ $y_k \Rightarrow y_p$ s0 $\Rightarrow$ Merge in S0

Figure 2.51 Diagram illustrating operation of the Control Block of the Viterbi detector.

The algorithm is applied to an example in Figure 2.52. The operation is as follows. The information of the most recent merge and current  $y_p$  take us to a particular condition in Figure 2.51 (numbered from 1-6). The channel output  $y_k$  is used in determining  $y_k - y_p$  and the resulting output transition. If a merge occurs,  $y_p$  is updated to the current  $y_k$  and becomes the current  $y_p$  during the next period. Output bits correspond to merges in the states.

1. This is one of many possible implementations of the Viterbi algorithm for a two state-trellis. It is attractive due to its simplicity and ease with which it facilitates implementation [31].

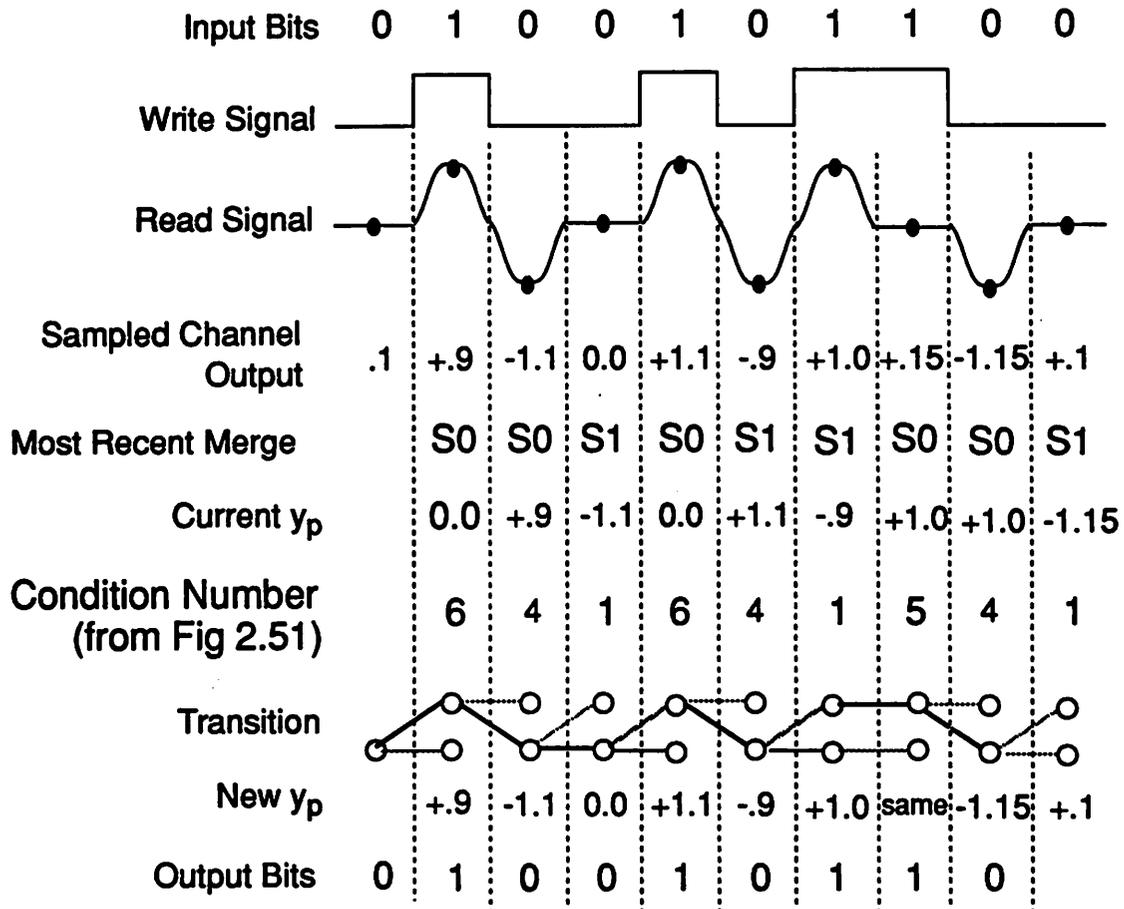


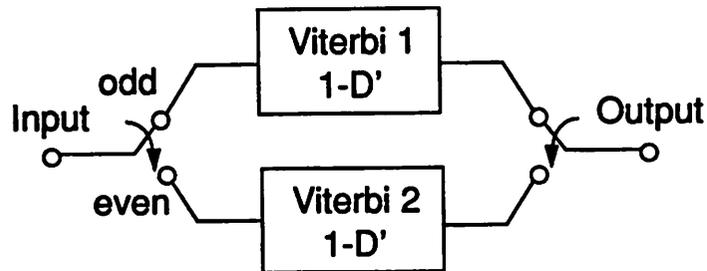
Figure 2.52 Example illustrating operation of the Viterbi detector.

The literature is filled with extremely rigorous treatments of the Viterbi algorithm [1,2,31]. Circuit implementations are described in [4,6,7,30]. Our purpose here was to generate an intuitive understanding of the mechanics of the algorithm for the case of the two-state trellis as applied to the magnetic disk drive channel and present its operation in the context of an example.

#### Application to PR-IV

Due to the  $1-D^2$  nature of the PR-IV channel, every other symbol is related through coding and thus the even and odd symbols can be decoded independently. The Viterbi detector can then be implemented with two interleaved  $1-D'$  detectors each operating at one-half the symbol rate as opposed to one full-rate  $1-D^2$  detector.  $D'$  in the interleaved detectors would actually be equal to

2D. An implementation of this approach is shown in Figure 2.53 with the odd samples going to one and the even going to the other. A commutator at the output re-samples the data to get it in the correct order. The detector described in this section was in fact for a 1-D channel so would work in this application directly.



**Figure 2.53** Implementation of  $1-D^2$  Viterbi detector with two half-rate  $1-D'$  detectors where  $D = 2D'$ .

## 2.6 Summary

In this chapter, we introduced some key concepts in communication theory important to baseband receiver design and applicable to the magnetic disk read channel employing discrete-time processing. The concept of an AWGN channel was introduced and applied to the magnetic recording channel. Key functions required in baseband receivers were discussed including equalization, timing recovery, coding, and sequence detection. We examined some of the underlying principals behind the operation of these blocks, using the disk drive read channel application as a vehicle demonstrating their operation. Many rigorous treatments of each of the topics covered in this chapter may be found in the literature. Since much of the communication theory used in baseband channels was developed in the 60s and 70s, pointers in the references of recent text books by Proakis [2] and Lee and Messerschmitt [1] can be very useful. Descriptions of magnetic disk drive systems employing PRML signaling can be found in [7,9,29].

In Chapter 3, we shall take a closer look at the read process in the magnetic disk channel and survey some of the more important detection techniques that have been proposed. In Chapter 4, architectures for baseband receivers are presented, focusing primarily trade-offs of the partitioning

of the signal processing required in these channels between the analog and digital domains. The principals presented in this chapter form an important backdrop for the discussion in Chapter 4.

## References

- [1] E. A. Lee and D. G. Messerschmitt, *Digital Communication*, Kluwar Academic Publishers, 1988.
- [2] J. Proakis, *Digital Communications, Second Edition*, McGraw-Hill, 1989.
- [3] H. Kobayashi and D. T. Tang, "Application of Partial-Response Channel Coding to Magnetic Recording Systems," *IBM Journal of Research and Development*, pp. 368-375, July 1970.
- [4] H. Kobayashi, "Application of Probabilistic Decoding to Digital Magnetic Recording," *IBM Journal of Research and Development*, pp. 64-74, January 1971.
- [5] F. Dolivo, "Signal Processing for High-Density Digital Magnetic Recording," *Proceedings of COMEURO 89*, Hamburg, W. Germany, May 1989.
- [6] R. W. Wood and D. A. Peterson, "Viterbi Detection of Class IV Partial Response on a Magnetic Recording Channel," *IEEE Trans. on Communications*, vol. com-34, no. 5, pp. 454-461, May 1986.
- [7] R. D. Cideciyan, F. Dolivo, R. Hermann, W. Hirt, and W. Schott, "A PRML System for Digital Magnetic Recording," *IEEE Journal on Selected Areas in Communications*, vol 10, no. 1, pp. 38-56, January 1992.
- [8] P. Siegal, "Recording Codes for Digital Magnetic Storage," *IEEE Transactions on Magnetics*, vol. MAG-21, no. 5. pp. 1344-1349, Nov. 1986.
- [9] J. Hong, R. W. Wood, D. Chen, "An Experimental 180 Mb/sec PRML Channel for Magnetic Recording," *IEEE Transactions on Magnetics*, Vol. 27, No. 6, pp. 4532-4537, November 1991.
- [10] J. M. Cioffi, W. L. Abbott, H. K. Thapar, C. M. Melas, and K. D. Fisher, "Adaptive Equalization in Magnetic-Disk Storage Channels," *IEEE Communications Magazine*, pp. 14-29, Feb. 1990.
- [11] G. Ungerboeck, "Fractional Tap-Spacing Equalizer and Consequences for Clock Recovery in Data Modems," *IEEE Transactions on Communications*, Vol. COM-24, No. 8, pp. 856-864, August 1976.
- [12] H. K. Thapar, P. A. Ziperovich, and R. W. Wood, "On the Performance of Symbol- and Fractionally-Spaced Equalization in Digital Magnetic Recording," *Eighth Conference on Video, Audio, and Data Recording*, Birmingham, April 1990.
- [13] J. C. Mallinson and C. W. Steele, "Theory of Linear Superposition in Tape Recording," *IEEE Transactions on Magnetics*, Vol. MAG-5, p 886 (1969).
- [14] F. Dolivo, "Signal Processing for High-Density Digital Magnetic Recording," *Proceedings of COMEURO 89*, Hamburg, W. Germany, May 8-12 1989.
- [15] J. J. Moon and L. R. Carley, "Performance Comparison of Detection Methods in Magnetic Recording," *IEEE Transactions on Magnetics*, vol. MAG-26, no. 6, pp. 3155-3172, November 1990.

- [16] W. L. Abbott, J. M. Cioffi, and H. K. Thapar, "Performance of Digital Magnetic Recording with Equalization and Offtrack Interference," *IEEE Transactions on Magnetics*, Vol. 27, No. 1, pp. 705-716.
- [17] H. M. Sierra, *An Introduction to Direct Access Storage Devices*, Academic Press, Inc., 1990.
- [18] C. D. Mee and E. D. Daniel, *Magnetic Recording Handbook: Technology & Applications*, McGraw Hill, 1990.
- [19] M. Camras, *Magnetic Recording Handbook*, Van Nostrand Reinhold, 1988.
- [20] A. S. Hoagland, *Digital Magnetic Recording*, Robert E. Krieger Publishing, Co., 1963.
- [21] M. L. Honig and D. G. Messerschmitt, *Adaptive Filters: Structures, Algorithms, and Applications*, Kluwer Academic Publishers, Boston, 1984.
- [22] O. Agazzi, D. G. Messerschmitt, and D. A. Hodges, "Non-linear Echo Cancellation of Data Signals," *IEEE Transactions on Communications*, COM-30(11), pp. 2421-2433, November 1982.
- [23] K. Fisher, J. Cioffi, and C. M. Melas, "An Adaptive DFE for Storage Channels Suffering from Non-linear ISI and Transition Dependent Noise," ICC 89, Boston, June 1989.
- [24] J. M. Cioffi, W. L. Abbott, H. K. Thapar, C. M. Melas, and K. D. Fisher, "Adaptive Equalization in Magnetic Disk Storage Channels," *IEEE Communications Magazine*, pp. 14-29, February, 1990.
- [25] K. H. Mueller and M. Muller, "Timing Recovery in Digital Synchronous Data Receivers," *IEEE Transactions on Communications*, COM-24, pp. 516-531, May 1976.
- [26] S. U. H. Qureshi, "Timing Recovery for Equalized Partial-Response Systems," *IEEE Transactions on Communications*, pp. 1326-1331, December 1976.
- [27] S. U. H. Qureshi, "Adaptive Equalization," *Proceedings of the IEEE*, Vol. 73, No. 9, p. 1349-1387, September 1985.
- [28] H. K. Thapar and A. Patel, "A Class of Partial Response Systems for Increasing Storage Density in Magnetic Recording," *IEEE Transactions on Magnetics*, Vol. 23, No. 5, pp. 3666-3668, September 1987.
- [29] J. D. Coker, R. L. Galbraith, G. J. Kerwin, J. W. Rae, and P. A. Ziperovich, "Implementation of PRML in a Rigid Disk Drive," *IEEE Transactions on Magnetics*, Vol. 27, No. 6, pp. 4538-4543, November 1991.
- [30] R. R. Spencer, "Simulated Performance of Analog Viterbi Detectors," *IEEE Journal on Selected Areas in Communications*, Vol. 10, No. 1, pp. 277-288, January 1992.
- [31] M. J. Ferguson, "Optimal Reception for Binary Partial Response Channels", *The Bell System Technical Journal*, Vol. 51, No. 2, pp. 493-505, February 1972.

## CHAPTER 3

# Detection Techniques for Magnetic Disk Drive Channels

---

### 3.0 Introduction

Two key performance measures for magnetic disk drives are transfer rate and recording density. These objectives are met by rotating the disk at higher rates and writing transitions onto the disk at closer intervals. Two important elements in magnetic disk recording systems which make this possible are the head and disk technology and the signalling techniques used to write and retrieve information to and from the disk. The head and disk technology affect the pulse shape of the signal that is finally read from the disk as well as add noise and distortion to this signal. The signalling techniques control what is written to the disk so that the signal that is read from the disk has desirable characteristics to aid in and increase the reliability of the detection process. There are other very important elements such as the servo system [2] which aligns the head over the particular track being written to or read but will not be discussed in this thesis.

In this chapter, we will present the basic concepts behind the write and read processes used in conventional disk drives today. Then, we will examine some of the detection methods used on the read signal that allow retrieval of the information stored there. We shall begin with a brief exami-

nation of the physics behind writing and reading digital data to and from the magnetic channel. This will be followed by a presentation of two of the more important implementation methods of the read head which are the inductive read (IR) and magnetoresistive (MR) heads. The two primary types of detectors for magnetic disk channels are peak detection and discrete-time detection. Both will be discussed. There appear to be performance advantages particularly at higher recording densities with the application of discrete-time techniques. We shall present the more popular techniques and highlight some of their key advantages. One important disadvantage of discrete-time approaches is the sophistication required of the signal processing in the read electronics. Techniques proposed for integrated circuit implementation of some of the key functions required in discrete-time channels are presented later in this thesis.

The treatment presented here is mainly to provide a background of the fundamental principals upon which the magnetic disk drive channel operates. The physics associated with the processes of the write and read process are well understood and several in-depth descriptions can be found in the literature on all of the topics discussed in this chapter [1,2,4,5].

## **3.1 Fundamentals of the Read/Write Process**

The physics involved in the read and write process are extremely complex. We shall examine only the basic principles behind these processes.

### **3.1.1 Background Physics**

A discussion of the write process requires a review of magnetism. Although it may not be obvious, it turns out that all substances are attracted to or repelled by magnets. However, since the effect of magnets on most substances is so small, we tend to neglect the magnetic susceptibility of most of them. Those that show a strong attraction towards magnets are called *ferromagnetic* materials. For the purposes of magnetic recording, *electromagnetism* and *permanent magnets* are important. Electromagnetism occurs when ferromagnetic materials are used together with electrical windings which are used to magnetize permanent magnets. A permanent magnet is a substance which retains its magnetism even after the magnetizing force is removed.

Let us now examine the basic concepts behind electromagnetism. Following this, we will examine qualities of permanent magnets and then see how electromagnetism and permanent magnets are used together in digital magnetic storage.

The basic principals behind electromagnetism are shown in Figure 3.1 where a current source is shown driving a coil of wire through a switch. This current flow results in the presence of magnetic flux shown as the dashed lines in the figure. This magnetic flux results in the coil of wire becoming a magnet when current is present with N and S poles as indicated in figure. When this current is removed, the magnetic properties are also removed. If a core of ferromagnetic material is placed inside the coil of wire, electromagnetism is greatly enhanced and the resulting magnetic flux can be increased by orders of magnitude. Another important phenomenon is that if the switch is open and a *changing* flux is present in the coil, a voltage appears across the terminals of the coil proportional to the time rate of change of the flux.

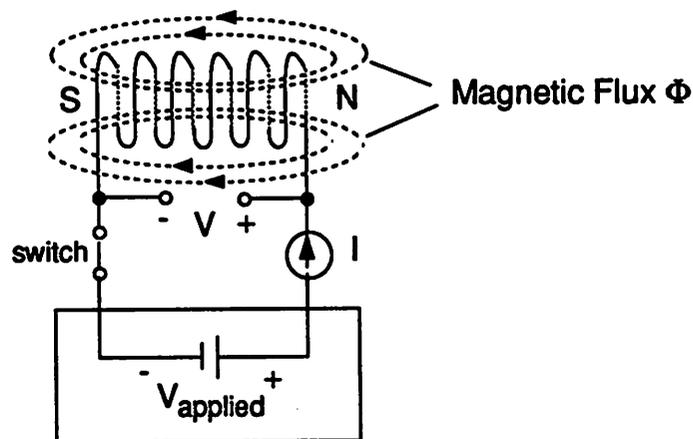


Figure 3.1 Conceptual diagram illustrating principals of electromagnetism.

It is useful to consider the idealized magnetic circuit shown in Figure 3.2 in order to define some of the fundamental relationship between the applied current and resulting magnetic flux. There is an Ohm's Law for magnetic circuits which is

$$\Phi = \frac{F}{R} \quad (3.1)$$

where  $\Phi$  is the magnetic flux,  $F$  is the magnetomotive force, and  $R$  is the reluctance. Therefore  $\Phi$  is analogous to current,  $F$  to voltage, and  $R$  to resistance in an electrical circuit. The magnetomotive force is given by

$$F = NI \quad (3.2)$$

where  $N$  is the number of turns in the coil and  $I$  is the applied current. The reluctance is given by

$$R = \frac{l}{\mu_0 \mu_r A_c} \quad (3.3)$$

where  $l$  is the length or average circumference of the ring,  $\mu_0$  is the permeability of free space,  $\mu_r$  is the relative permeability of the core material, and  $A_c$  is the cross sectional area of the ring. Equations 3.1 - 3.3 can be combined to get

$$\Phi = \frac{NI\mu_0\mu_r A_c}{l} \quad (3.4)$$

This means that in order to increase the magnetic flux, one can increase the number of turns in the coil, increase the input current, use a core material with a higher permeability, increase the cross sectional area of the ring, or decrease the length of the ring.

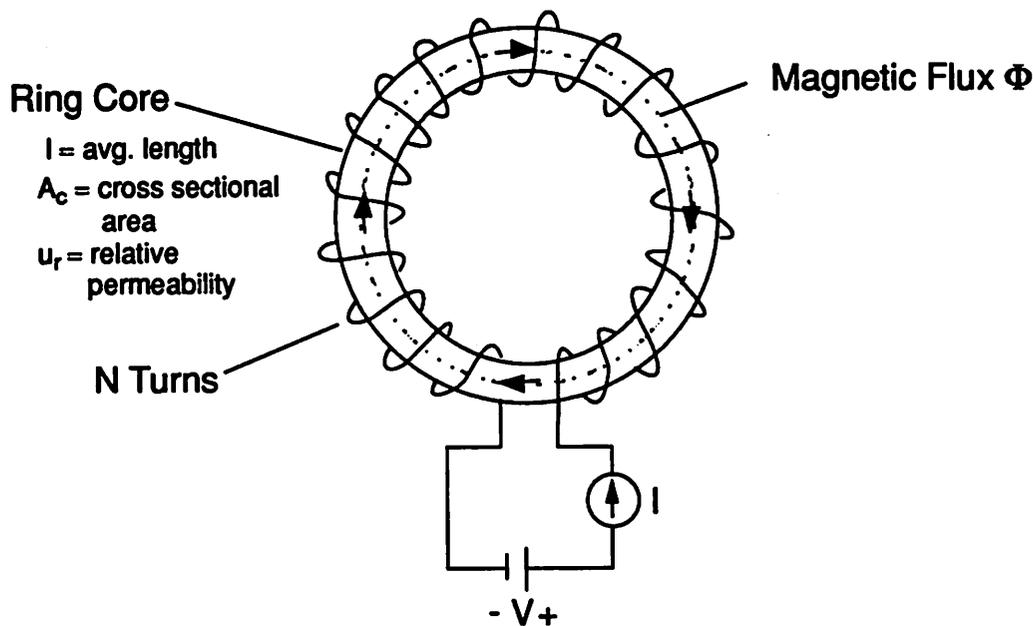


Figure 3.2 Illustration for definition of Ohm's Law for a Magnetic Circuit.

Let us now examine the principals behind permanent magnets. Shown in Figure 3.3 is a plot of the relationship between the applied magnetomotive force  $F$  and the resulting magnetic flux  $\Phi$ . As  $F$  is increased positive from zero,  $\Phi$  increases almost linearly at first and then saturates as shown in the figure. However, as  $F$  is reduced and taken negative,  $\Phi$  traverses a different path. Once again, as  $F$  goes negative, there is an almost linear relationship to  $\Phi$  and before it eventually it saturates. As  $F$  is taken positive, a new path arises. This path eventually intersects with the path from the initial positive excursion of  $\Phi$  and for all subsequent tracing of  $F$  and  $\Phi$ , the outer most tracings will result. In other words,  $\Phi$  starts from zero and passes through zero only the first time. The hysteresis effect we see of the outer tracings is a characteristic of the particular magnetic material.

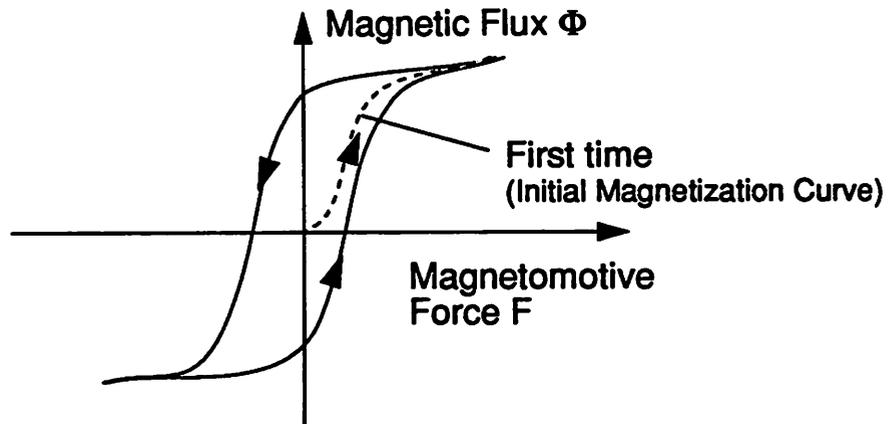


Figure 3.3 Hysteresis loop that results from plotting  $\Phi$  versus  $F$ .

Normally,  $F$  and  $\Phi$  are normalized to remove the physical dimensions and instead  $H$  and  $B$  are plotted as shown in Figure 3.4 where  $H$ , the field intensity is given by

$$H = \frac{F}{l} \quad (3.5)$$

and  $B$ , the flux density is given by

$$B = \frac{\Phi}{A_c} \quad (3.6)$$

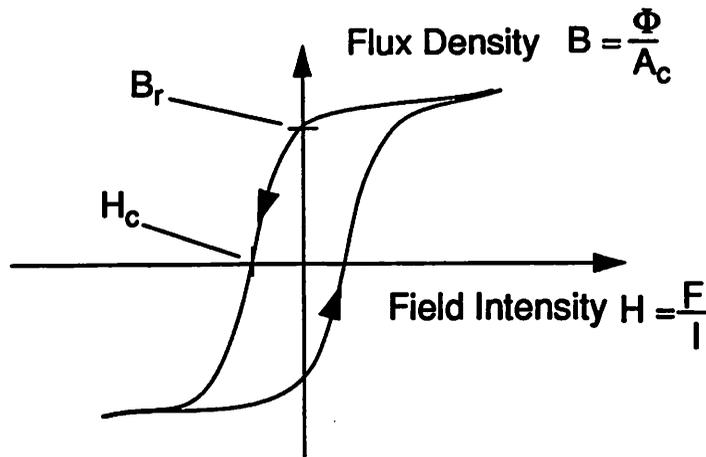


Figure 3.4 Hysteresis loop (B vs. H) with  $B_r$  and  $H_c$  indicated as shown.

Two important quantities are marked in Figure 3.4 and they are  $B_r$  and  $H_c$ . Both are important measures of permanent magnetization.  $B_r$  is called the *remnant magnetization* and is the magnetization which remains after the field  $H$  is removed. The other quantity  $H_c$  is called the *coercivity*. It is the reverse field required to reduce  $B_r$  to zero.

For substances used in the media for magnetic recording, it is desirable to have large  $B_r$  so that the material will retain its magnetism and a large  $H_c$  so that the magnetism can be removed or changed only by a strong magnetic field. Such materials are referred to as *hard* or *permanent* ferromagnetic materials. Ferromagnets that can be readily demagnetized are also important in magnetic recording for use in the heads that perform writing, reading, and erasing.

### 3.1.2 Write Process

The magnetic head is the key interface between the electronics and the magnetic media. The same head can be used to write, read, and also to erase. However, separate heads are often used so that each may be optimized for a particular function. Magnetic heads are made of a high permeability core material with a low coercivity which makes them easy to demagnetize. Many approaches and head shapes are used for the magnetic head, one of which is shown in Figure 3.5 in the shape of a ring. A key feature of the magnetic head is a gap at the place in which the head interfaces to the media. As current is applied to the coil (from the write electronics), some of the

field from the resulting magnetomotive force fringes outward at the gap enabling magnetization of the medium. The time variation of the input current results in spatial variations in the remanent magnetization along the medium. In digital recording, saturation recording is used where the field intensity applied to the medium is enough to magnetize the media in one of the two directions. Therefore, with a binary input current of  $\pm I$ , the spatial magnetization of the media for a fixed rate of rotation of the disk is a recording of the input current.

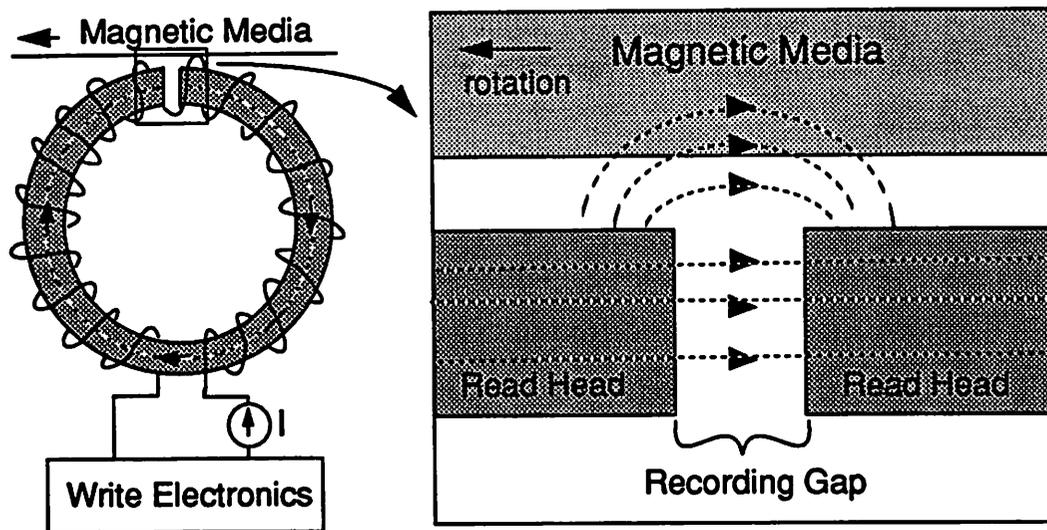


Figure 3.5 Magnetic head illustrating the presence of the recording gap.

### 3.1.3 Read Process

There are two primary approaches for reading the recorded digital information from a magnetic disk. They are the inductive read head which is the approach used in the majority of disk drives today and magneto-resistive (MR) heads which are a key technology currently in development which promises to make a big impact on recording densities and transfer rates.

#### 3.1.3.1 Inductive Read Heads

The read process using inductive read heads were described briefly in Chapter 2. The same head can be used as in the write process but the terminals of the coil are connected to the input nodes of a wideband amplifier called the *read* amplifier. As the magnetized media passes by the

head, fringing fields emanating from the magnetized media couple into the core which is wrapped by the  $N$ -turn coil. The  $d\Phi/dt$  sensed by the coil produces a voltage at the terminals of the coil which is amplified by the read amplifier. The fringing flux prefers the high permeability path through the core rather than air which improves the efficiency of the process

One problem of inductive read heads at high recording densities is the phenomenon of *undershoots*. Undershoots are little peaks that precede and follow an isolated pulse by anywhere between 15-25 periods with a polarity opposite to that of the main pulse. Since they are present so far away from the actual pulse, they are difficult to equalize. Most often, they are simply treated as noise. However, undershoots may especially be a problem for discrete-time techniques such as PRML detection.

In general, linear superposition can be applied to the disk drive channel [12,22]. There is however, a phenomenon referred to as *non-linear peak shift* (different from ISI per se) which results in the deterministic shifting of pulses depending on the particular data sequence that was written to the channel. Since this phenomenon is well understood, a technique called *write pre-compensation* is often used to compensate for these shifts during the write process. In write pre-compensation, the spacing of pulses written to the disk is modulated depending on the particular data pattern in order to pre-compensate for the non-linearities incurred during the read process. When this approach is taken, the resulting channel appears to be absent of non-linearities and can be treated as linear.

### 3.1.3.2 Magnetoresistive Read Heads

Magnetoresistive (MR) heads use a completely different approach to sensing the signal from the media. They are based on the magnetoresistive effect present in films of certain materials. In a simplistic form, the MR head is made of a strip of one of these materials mounted in a plane perpendicular to the media. Instead of sensing the time rate change of flux, the impedance of a magnetoresistive head changes with the applied magnetization. A block diagram of the operation of a MR head is shown in Figure 3.6. A bias current is forced through the MR head. As the disk rotates and the applied field changes, the impedance of the head correspondingly changes resulting in a

voltage change across the head. This voltage is amplified and the d.c. bias is removed using circuit techniques [11].

There are a number of important advantages of MR heads as compared with inductive heads. First, signal levels are larger coming out of an MR head since it responds directly to the flux intensity present as opposed to the time-rate change of flux in the inductive read head case. Therefore, it can be used with a broad range of disk velocities. Second is the absence of the undershoots present in the inductive read heads. And third is that the linearity associated with MR heads is superior to that of inductive read heads, facilitating the use of discrete-time techniques. Because of these advantages MR head technology promises to make a big impact as densities increase. Currently, only a few manufacturers have MR head technology. It is quite expensive to develop the expertise required to produce them in a production environment. They should become much more prevalent as more disk drive manufacturers gain access to this technology.

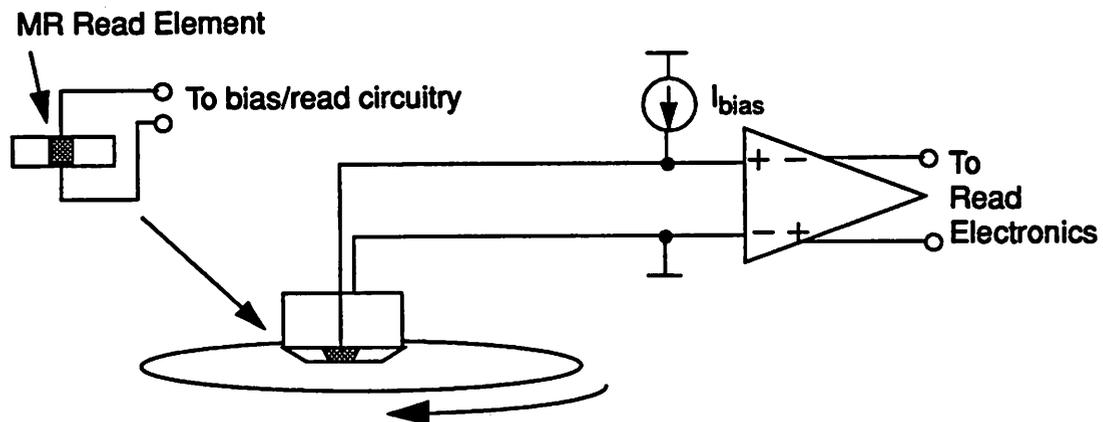


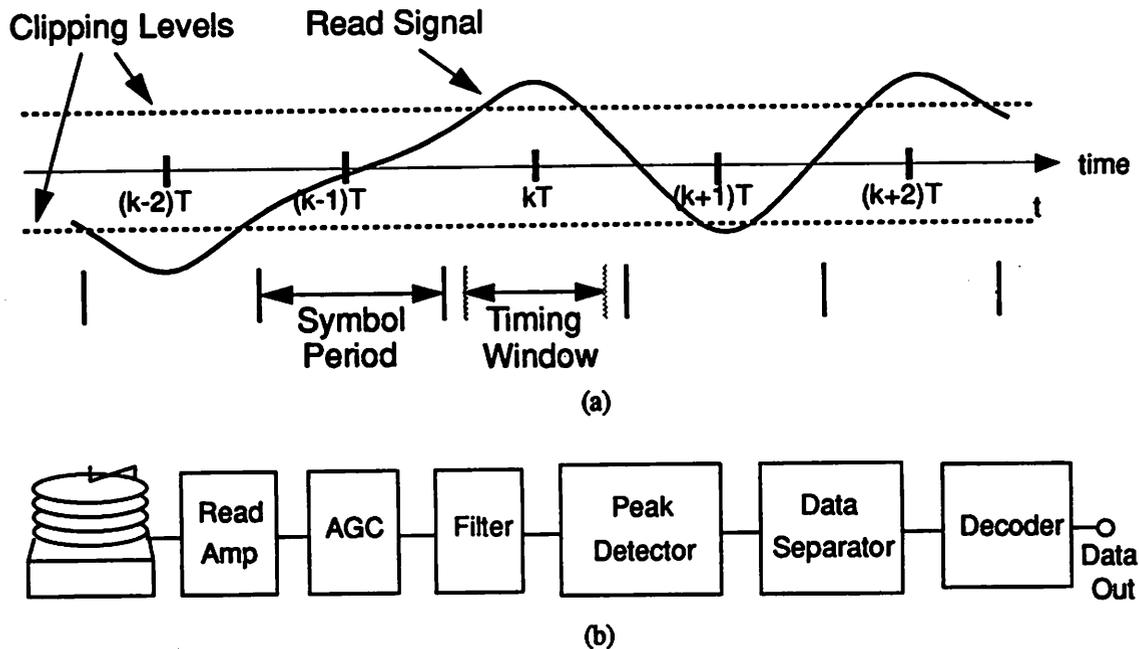
Figure 3.6 Simple schematic diagram of the use of an MR read head.

## 3.2 Peak Detect Channels

The majority of disk drives on the market today use the method of *peak detection* for data recovery shown conceptually in Figure 3.7. Peak detection is a non-linear detection technique in which the read signal is observed for positive and negative peaks which correspond to transitions in the write signal. The absence of a peak means no transition. The signal is observed once per symbol period for a duration called a *timing window* which is slightly shorter than a symbol

period. If a peak is detected, the amplitude of the signal must also be larger than reference levels called *clipping* levels. This qualification procedure improves the performance of this detection method in the presence of noise and other perturbations which may result in erroneous peaks.

A simplified block diagram of a peak detect-based read channel is shown in Figure 3.7b. The signal off the disk is on the order of a few millivolts and is amplified by the Read Amp up to the tens of millivolts range. The read amp is often situated together with the read head. The signal is then passed to the automatic gain control (AGC) block which normalizes the signal amplitude for further processing. The amplitude out of the AGC is normally a few hundred millivolts. Next, the signal is passed to the Filter which performs bandlimiting on the noise as well as signal pre-conditioning prior to peak detection. This signal pre-conditioning usually involves some high frequency boost to slim pulses and reduce ISI and helps in the qualification procedure since peaks are enhanced [3,6,13]. Thus, these filters are often called *pulse-slimming* filters or equalizers. After the filter is the peak detection block which observes the conditioned signal for qualified peaks. When a peak is detected, a "1" is output and a "0" otherwise. The peak detect block operates continuously on the signal and thus the location of the output "1" depends upon where within the timing window the peak is detected. As a result, the width and location of the logic 1 pulse out of the peak detector varies. The Data Separator re-samples the output of the peak detector block and aligns the data with a local clock in order to provide a synchronous data flow for further processing. Finally, the Decoder undoes the effects of any precoding that may have been performed prior to the write process for error detection/correction. The output of the decoder is the user data.



**Figure 3.7** Simplified view of the peak detection process. (a) Read signal shown with clipping levels and timing window definitions. (b) Block diagram of a peak detect read channel.

A conceptual block diagram of the functions in one implementation of the peak detector block and the associated signals at different points within the block are shown in Figure 3.8 [2]. The filtered signal is first processed by the *base clip* block which has the transfer function shown in the figure. The output signal follows the input signal only when the input signal is larger than the clipping levels. The signal is then rectified and passed to a *differentiator* which performs an analog differentiation. Note in the figure that the differentiator output passes through zero at the signal peaks where the derivative goes through a change in slope. The output of the differentiator is passed to an *inverting limiter* which has a small offset as shown in the figure. The limiter results in a logic "1" when the differentiator output is negative. The offset is required in the limiter so that noise on the differentiator output when it is near zero does not cause toggling of the limiter output. Note that each logic "1" in the limiter output corresponds to a peak in the read signal. The differing widths between the logic "1" output of the limiter and the variations in its exact location require the data separator for synchronization.

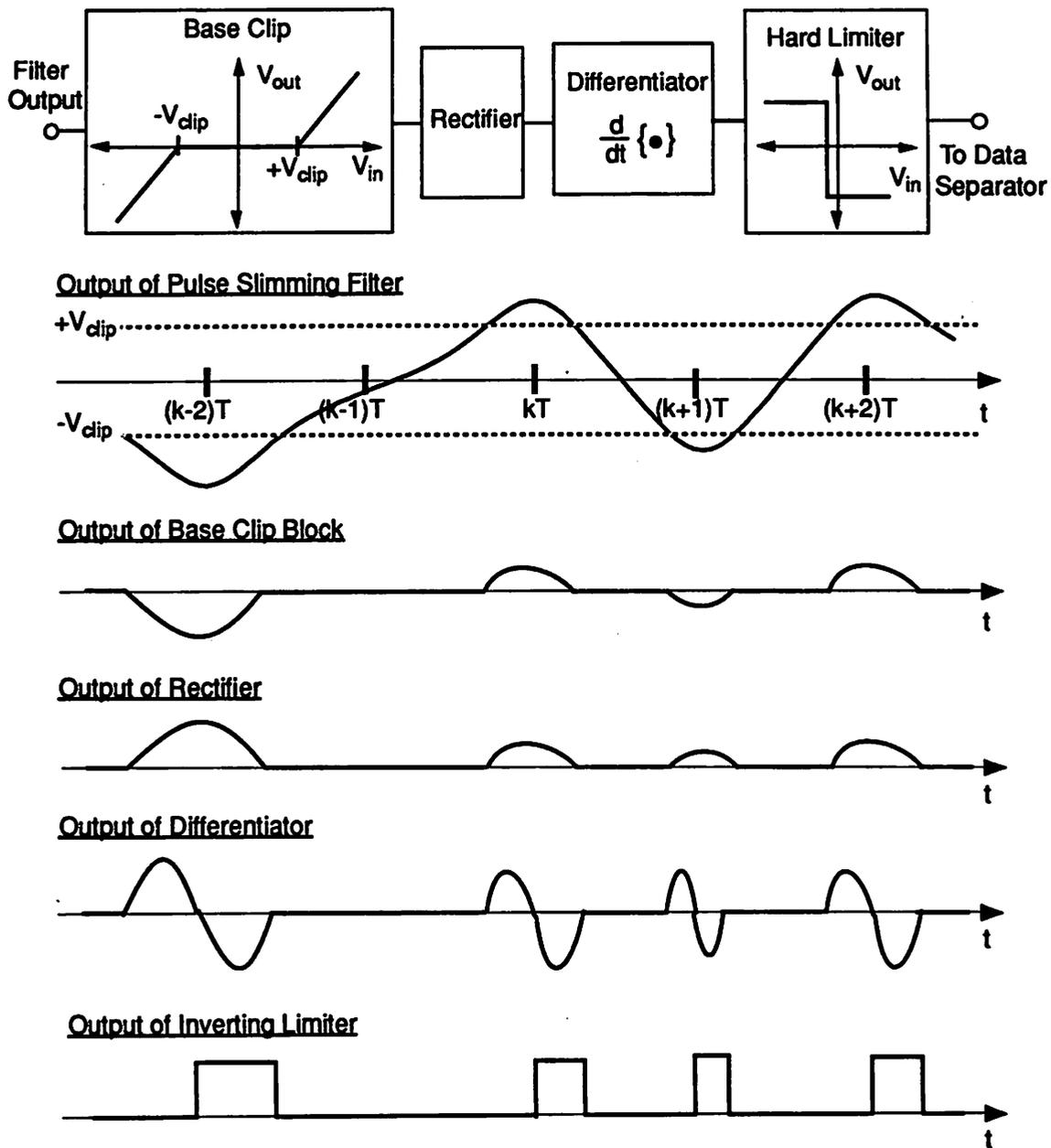


Figure 3.8 Conceptual block diagram of operation of a peak detector block.

At higher transition densities, the deleterious effects of intersymbol interference including the reduction of peak amplitude and peak shift degrade the performance of peak detect channels. Amplitude reduction requires that clipping levels be reduced making the system more susceptible to noise. The problem with peak shift is that in the presence of noise and jitter in the signal phase, peaks may shift outside of the timing window and be decoded in the wrong position.

Peak detection works well in the absence of intersymbol interference and is an attractive solution under this condition because of the relative simplicity with which the detection electronics can be implemented. Unfortunately, at higher recording densities and transfer rates ISI is unavoidable. It has been shown that at higher recording densities and transfer rates sampled-data detection methods are superior to peak detection [8,9]. We shall next examine sampled-data detection methods before comparing the various approaches.

### 3.3 Discrete-Time Detection

Under the conditions that linear superposition holds, sampled-data methods employing equalization and sequence detection may be applied to the magnetic disk drive channel as discussed in Chapter 2. Discrete-time methods allow the application of communications techniques such as adaptive equalization and sequence detection. A block diagram of a discrete-time read channel is shown in Figure 3.9. After the AGC block, the rest of the receiver is very different from the peak detect channel. In the receiver shown, it is assumed that the equalization and sequence detection is performed in the digital domain<sup>1</sup>. This receiver is comprised of two main parts. The first is the CODER which contains a filter which provides anti-alias and noise filtering, a sampler driven by voltage controlled oscillator (VCO) in a timing recovery loop, and an analog-to-digital converter (ADC) which converts the analog samples into the digital domain. The second is a dedicated digital signal processor which contains an adaptive equalizer, symbol detector, and timing function generator which operates on the equalized samples and drives the VCO.

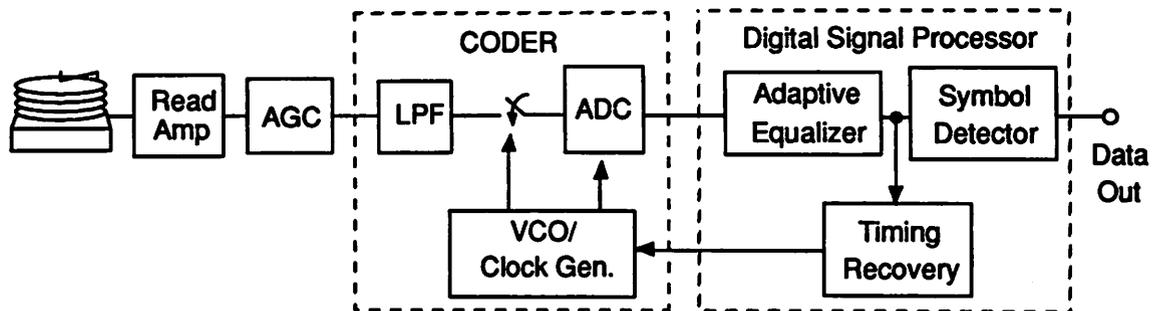


Figure 3.9 Block diagram of a read channel employing discrete-time detection methods.

1. In Chapter 4 we shall examine various ways of partitioning the signal processing between the analog and digital domains.

The different discrete-time detection methods that have been proposed use either different equalization methods and/or different techniques for symbol detection. The simple model comprised of a sampler, equalizer, and symbol detection as shown in Figure 3.10 will be used in the remainder of this section. We will assume the presence of a timing recovery loop. The processing can either be in the analog or digital domain, or a combination of both. The discrete-time methods

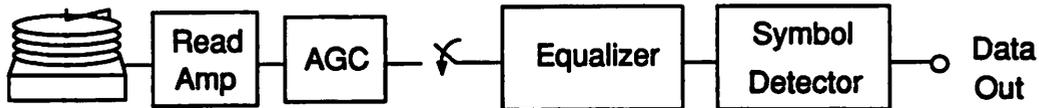


Figure 3.10 Simplified discrete-time channel block diagram.

we shall examine are Partial Response (PR) Signalling, the RAM-Based DFE, and Fixed-Delay Tree Search.

### 3.3.1 Partial Response Signalling

In the time period between 1969-1970, researchers at IBM proposed the application of Partial Response (PR) signalling [14] with maximum-likelihood (ML) detection [15] to the magnetic disk drive channel. As discussed in Chapter 2, these techniques are together referred to as Partial Response Maximum-Likelihood or PRML signalling. Partial response signalling by definition introduces controlled ISI and therefore allows a higher net recording density. The use of maximum likelihood detection in the form of Viterbi sequence detectors results in coding gain. *Coding gain* is a measure of the robustness to reduction in signal-to-noise ratio that a coding approach allows. It is defined as the difference between the signal-to-noise ratios for the same probability of error with and without the coding approach applied.

As discussed in Chapter 2, the general partial response (PR) coding polynomial is

$$P(D) = (1 - D)^m (1 + D)^n. \quad (3.7)$$

Many PR systems have been proposed for application in the disk drive channel with different combinations and values of both  $m$  and  $n$ . The term  $(1-D)^m$  introduces  $m$  zeros into the spectrum at d.c.

and  $(1+D)^n$  introduces  $n$  zeros at one-half the symbol rate. The two systems that are most often seen in the literature are Class IV PR or PR-IV using  $m=n=1$  for a  $P(D)$  of  $1-D^2$  and Extended Class IV PR or EPR-IV that uses  $m=1$  and  $n=2$  for a  $P(D)$  of  $1+D-D^2-D^3$ . PR-IV introduces a zero at d.c. and zero at one-half the symbol rate. Whereas EPR-IV introduces a zero at d.c. and two zeros at one-half the symbol rate. The effect of this is to require less energy near one-half the symbol rate to be equalized. The zeros at d.c. and one-half the symbol rate in the PR-IV case match the disk drive channel very well but still requires high frequency boost for equalization at higher recording densities. The extra zero at one-half the symbol rate results in EPR-IV requiring even less boost at high frequencies and thus suffers less noise enhancement. However, there is a significant difference in the implementation of the Viterbi detector when maximum likelihood detection is employed. The detector goes from a 2-state detector with 3 different output levels in PR-IV to an 8-state in EPR-IV with 5 different levels. As a result, implementation advantages such as implementing the PR-IV detector as two interleaved half-rate detectors is not possible in EPR-IV making implementation much more difficult.

A comparison of different PR systems is presented in [10]. SNR after equalization was the metric for comparison. It is shown that at low densities ( $PW50=1$ ), PR-IV performs better than EPR-IV for both perfect sampling phase and a 10% sampling phase error. However, at high densities ( $PW50=3$ ), EPR-IV performs better but only with small errors in the sampling phase. Intuitively, it performs better because it experiences less noise enhancement after equalization than PR-IV. But it also is a lot more sensitive to sampling phase.

The key trade-offs between PR approaches include:

- **Number of levels.** More levels results in increased difficulty in differentiating between levels in the presence of noise. The number of levels greatly affects implementation of the Viterbi detector due to the number of required states.
- **Bandwidth required.** Higher order polynomials result in a lower required signal bandwidth. Since the magnetic disk channel is lowpass in nature, the use of higher order polynomials results in less high frequency boost required by the equalization resulting in simpler equalizers and less noise enhancement.
- **Sensitivity to sampling phase.** The use of higher order polynomials tend to be more sensitive to errors in the sampling phase.

At some point, the signalling approach chosen may be a strong function of the implementation of the detection circuitry. This will be particularly true if the power and cost of implementation of the electronics required by these channels cannot be brought sufficiently low. The key trade-offs from an implementation standpoint will be between implementation of the equalization, Viterbi detector, sensitivity to sampling phase and the number of bits required in the analog-to-digital converter.

### 3.3.2 RAM-Based DFE

The linear channel model for the disk drive read channel which assumes the validity of linear superposition breaks down in the absence of write pre-compensation or with certain types of read heads. Therefore, the RAM-based DFE was proposed for the read channel application in order to reduce the effects of non-linear ISI [16]. The RAM-based DFE was described in Chapter 2.

The acceptance of the RAM-based DFE in the disk drive community has not been very widespread. One of the key reasons is the complexity associated with implementation of this architecture is rather high. MR head technology has become a commercially viable head technology and shows promise for channels that remain quite linear up to very high densities and transfer rates. It remains to be seen whether or not the RAM-based DFE will have much of an impact in future read channel implementations.

### 3.3.3 Fixed-Delay Tree Search with Decision Feedback Equalization

Another discrete-time detection method proposed for use in the disk drive applications is *Fixed-Delay Tree Search with Decision Feedback Equalization (FDTS/DF)* [9]. Fixed delay tree search is a depth-limited exhaustive tree-search algorithm based on the binary tree representation of the channel shown in Figure 3.11. Starting from the first node, two branches extend corresponding to states 0 and 1. Then from each of these nodes, two more branches extend on growing exponentially with the depth of the tree. Each branch has an associated branch metric which is a measure of the likelihood of that particular branch given a channel output (similar to Viterbi). The basic idea is starting from the first node, calculate the metrics for each of the possible paths and find the path through the depth of the tree that is most likely in the mean-square sense. The branch emanating from the first node containing this path is kept as the correct path and the other half of

the tree is discarded. Then in the next symbol period, a new channel output is received and the nodes at the ends of the surviving branches are each extended one branch pair and the process is

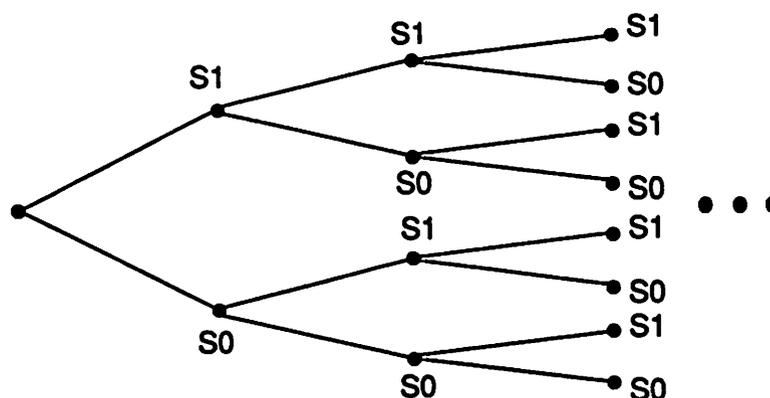


Figure 3.11 Binary Tree representation of the channel.

repeated. The outputs can be input into a feedback filter which subtracts the effects of post cursor ISI, resulting in the FDTS/DF structure shown in Figure 3.12.

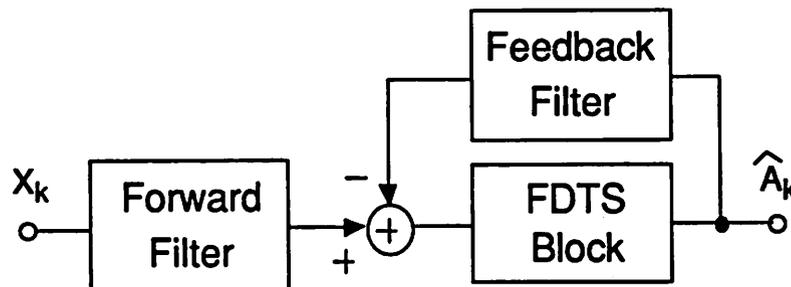


Figure 3.12 Block diagram of a FDTS/DF detector.

### 3.4 Comparison of Approaches

“I don’t believe anybody’s simulations but my own.”

Conference Attendee  
Magnetic Recording Conference  
1991

There are many comparisons of approaches in the literature [7,9] that compare the relative merits of certain approaches to others. Some comparisons are from a purely theoretical standpoint assuming ideal detection circuitry and considering only SNR. While others include one or two aspects about the detection circuitry for instance considering the number of taps in an adaptive equalizer, the sensitivity to timing phase, or the number of states required by a maximum likelihood detector and the impact on the implementation. It would not be fair to quote any one person's results because the assumptions that are made in the different papers tend all to be different. Instead we shall conclude this chapter with some concluding remarks, thoughts, and facts.

Peak detect channels have been employed since the advent of the magnetic recording industry and in an extremely high volume business as magnetic disk drives there is much inertia which brings about slow change. At lower densities, the inherent robustness of peak detection coupled with the ease of integrated circuit implementation makes it clearly the approach of choice. However, at higher densities, things become different. Although much of the literature suggests the outright advantages of discrete-time techniques, it must be recognized that the year IBM announced the 1 Gbit in a square inch prototype which employed discrete-time signaling, a prototype achieving 500 Mbits in a square inch employing peak detection was reported that same year [21].

PR-IV and higher order partial response channels seem to show the most promise at higher densities and transfer rates. PR-IV was the signalling method used in IBM's demonstration of 1 gigabit in a square inch of recording area [17]. This method was also used in an experimental channel with a transfer rate of 180 Mbits/sec [18]. The key stumbling block of the implementation of PRML channels is the economical implementation of the read channel electronics in terms of both power and cost. Power is an issue because of the heat it generates, particularly in drives with small form factors. Entire peak detect channels have been implemented in CMOS up to data rates of 30 Mbits/sec on a single silicon substrate dissipating 300 mW [19]. Whereas, the highest data rate read channel IC reported for a PRML system has been for a data rate of 65 MSymbols/sec at a power consumption of 1.75 W in a 1.0  $\mu\text{m}$  BiCMOS technology [20].

Important to PRML-based channels will be further refinement of MR head technology which allows the application of linear superposition to the read channel. To date, only IBM has a refined

MR head technology although many companies have spent many years and dollars attempting to develop this important technology.

Approaches such as the RAM-based DFE and FDTS/DF have been shown to have performance advantages under certain conditions. There has been much development in the way peripheral work such as coding for the PR-IV channels and the implementation of these techniques into products. It is unclear whether these other approaches will shown enough of an advantage to slow the inertia developing in the area of PR channels. After all, it took about 20 years from the first proposal of PR signalling to the digital magnetic recording channel before it made its way into a product. However, due to the competition in the industry to push for greater and greater performance, it is possible that these techniques will eventually find themselves in products providing whatever incremental performance advantages they are capable of.

It does appear that discrete-time channels employing partial response signalling will be one of the approaches used to achieve higher densities and transfer rates. Continued development in the area of MR head technology will indeed be a catalyst to further application of PR signalling. The challenge to circuit designers is for economical silicon implementation of these high speed channels in terms of both power and cost.

### **3.5 Summary**

We began this chapter with an examination of the fundamental physical principals behind the write and read process in digital magnetic recording. We examined the use of both the traditional inductive read head and the magnetoresistive (MR) heads which should become more important as densities increase and more sophisticated signal processing techniques are applied to the read channel. We then examined both peak detection and a few of the ways in which discrete-time detection has been proposed for the read channel. Currently, it appears that the industry is heading toward the increased application of PR signalling techniques.

In the next chapter, we will examine different block level architectures for baseband digital communication channels. Since the magnetic disk drive read channel employing discrete-time pro-

cessing can be modelled as a baseband communications receiver, the techniques described in the next chapter will be applicable to the disk drive channel.

## References

- [1] M. Camras, *Magnetic Recording Handbook*, Van Nostrand Reinhold Company, Inc., 1988.
- [2] C. D. Mee and E. D. Daniel, *Magnetic Recording Handbook: Technology & Applications*, McGraw-Hill Publishing Company, 1990.
- [3] Y. Tahara, H. Takagi, and Y. Ikeda, "Optimum Design of Channel Filters for Digital Magnetic Recording," *IEEE Transactions on Magnetics*, vol. MAG-12, No. 6, pp. 749-750, November 1976.
- [4] J. C. Mallinson, "Tutorial Review of Magnetic Recording," *Proceedings of the IEEE*, Vol. 64, No. 2, pp. 196-208, February 1976.
- [5] R. W. Wood, "Magnetic Recording Systems," *Proceedings of the IEEE*, Vol. 74, no. 11, pp. 1557-1569, November 1986.
- [6] G. V. Jacoby, "Signal Equalization in Digital Magnetic Recording," *IEEE Transactions on Magnetics*, Vol. Mag-4, No. 3, pp. 302-305, September 1968.
- [7] F. Dolivo, R. Hermann, and S. Olcer, "Performance and Sensitivity Analysis of Maximum-Likelihood Sequence Detection on Magnetic Recording Channels," *IEEE Transactions on Magnetics*, VOL. 25, No. 5, pp. 4072-4074, September 1989.
- [8] F. Dolivo, "Signal Processing for High-Density Digital Magnetic Recording," *Proceedings of COMEURO 89*, Hamburg, W. Germany, May 8-12 1989.
- [9] J. J. Moon and L. R. Carley, "Performance Comparison of Detection Methods in Magnetic Recording," *IEEE Transactions on Magnetics*, vol. MAG-26, no. 6, pp. 3155-3172, November 1990.
- [10] J. J. Moon and L. R. Carley, "Partial Response Signaling in a Magnetic Recording Channel," *IEEE Transactions on Magnetics*, Vol. 24, No. 6. pp. 2973-2972, November 1988.
- [11] D. P. Swart and T. J. Schmerbeck, "An 8-Channel, Head Preamplifier for Combination Magnetoresistive Read Elements and Inductive Write Elements," *ISSCC 1993 Digest of Technical Papers*, San Francisco, Vol. 36, pp. 218-219, February 1993.
- [12] J. C. Mallinson and C. W. Steele, "Theory of Linear Superposition in Tape Recording," *IEEE Transactions on Magnetics*, Vol. MAG-5, p 886 (1969).
- [13] P. Pai, A. A. Abidi, and R. A. Gomez, "A Simple Continuous-Time Equalizer for Use in Magnetic Storage Read Channels," *IEEE Journal on Selected Areas in Communications*, Vol. 10, No. 1, pp. 289-299, January 1992.
- [14] H. Kobayashi and D. T. Tang, "Application of Partial-Response Channel Coding to Magnetic Recording Systems," *IBM Journal of Research and Development*, pp. 368-375, July 1970.
- [15] H. Kobayashi, "Application of Probabilistic Decoding to Digital Magnetic Recording," *IBM Journal of Research and Development*, pp. 64-74, January 1971.
- [16] K. Fisher, J. Cioffi, C. Mike Melas, "An Adaptive DFE for Storage Channels Suffering from Non-linear ISI and Transition Dependent Noise," *ICC Boston*, 1989.

- [17] T. D. Howell, D. P. McCown, T. A. Diola, Y.-S. Tang, K. R. Hense, and R. L. Gee, "Error Rate Performance of Experimental Gigabit Per Square Inch Recording Components," *IEEE Transactions on Magnetics*, Vol. 26, No. 5, pp. 2298-2302, September 1990.
- [18] J. Hong, R. W. Wood, D. Chen, "An Experimental 180 Mb/sec PRML Channel for Magnetic Recording," *IEEE Transactions on Magnetics*, Vol. 27, No. 6, pp. 4532-4537, November 1991.
- [19] C. Peterson, P. Cheung, T.-S. Chung, W. Guo, K.-H. Loh, K. Miyasa, L. Pycior, M. R. Robinson, and K. Tomioka, "A 3-5.5V CMOS 32 Mb/s Fully-Integrated Read Channel for Disk-Drives," *Proceedings IEEE Custom Integrated Circuits Conference*, San Diego, CA, May 1993.
- [20] R. Philpott, R. Kertis, R. Richetta, T. Schmerbeck, and D. Schulte, "A 7 MB/sec (65 MHz) Mixed-Signal Magnetic Recording Channel DSP Using Partial Response Signaling with Maximum Likelihood Detection," *Custom Integrated Circuits Conference 1993*, pp. 10.4.1-10.4.4, May 1993.
- [21] R. A. Jenson, J. Mortelmans, and R. Hauswiter, "Demonstration of 500 Megabits Per Square Inch with Digital Magnetic Recording," *IEEE Transactions on Magnetics*, Vol. 26, No. 5, pp. 2169-2171, September 1990.
- [22] J. C. Mallinson, *The Foundations of Magnetic Recording: Second Edition*, Academic Press, 1993.

## CHAPTER 4

# Architectural Approaches

---

### 4.0 Introduction

Discrete-time techniques for the magnetic disk drive channel become necessary at higher recording densities and transfer rates where intersymbol interference reduces the reliability of peak detection techniques. PRML techniques are currently being applied in a commercial disk drive with a transfer rate of 64Mbits/sec using monolithic read channels implemented in BiCMOS [1]. Research prototype channels employing PRML signalling and operating at rates up to 180Mbits/sec have been reported [2]. In order for monolithic implementation to occur at rates of this order and at acceptable power levels, new circuit approaches and architectures need to be developed. At some point, the cost and power of the electronics and the energy required to spin the disk sufficiently fast to get these high transfer rates may make other approaches such as simultaneous reads and writes to and from parallel arrays of lower speed disks more attractive [3].

When operating with signals having bandwidths in the tens of mega-hertz and above, circuit implementation has a very strong impact on the achievable performance. Parasitic effects including resistance, capacitance, and inductance become very important at these frequencies in both

analog and digital circuits. Also important at these frequencies are second order effects of the active devices (transistors) themselves. Until very recently, CMOS technology was limited in analog circuit applications to frequencies in the 10MHz range. That has changed with sub-micron CMOS processes becoming industry standard and the increased availability of BiCMOS processes in which high speed NPN devices are available along with sub-micron CMOS.

The ultimate goal of the data communications circuit/system designer is to get data in the form of bits from one place to another (in space or time, the latter being the case for the magnetic disk drive channel) while achieving some desirable trade-off of speed, power, and cost. It is becoming more important for circuit designers to be *system* designers in order to make the best choices between the various technologies, architectures, and implementation approaches and to know when new approaches need to be developed. This high level view is fundamental to finding the optimum solution.

In this chapter, we shall examine different architectural approaches to receiver design. There are different ways in which the signal processing can be partitioned between the analog and digital domains with trade-offs associated with each. We will examine some of these trade-offs. A priority is made for implementation in CMOS-only technology since the cost of such a process is roughly 30% lower than BiCMOS. A new approach for implementation of high speed filters is presented later in this thesis with a prototype demonstrating these techniques described in Chapter 8. The block level architecture used in this prototype will be presented near the end of the chapter. Some of the performance advantages which result from the block level architecture will be presented.

In the first section, we shall present some pertinent background information such as how noise gets shaped by the signal processing through the channel. The information in this section is background and will be used throughout the rest of the chapter. Next, we will examine different architectural approaches viewing the entire receiver as a system and examine trade-offs of the partitioning of signal processing between the analog and digital domains. And finally, we will present and discuss a block level diagram of the prototype.

## 4.1 Pertinent Basics

In this section, some pertinent background basics will be presented. These concepts are important to the sections that follow. We will examine quantization noise associated with analog-to-digital conversion, noise shaping and amplification, latency in feedback systems, and the important differences between analog discrete-time and digital signal processing.

### 4.1.1 Quantization Noise in Analog-to-Digital Conversion

In an analog-to-digital (A/D) converter, samples of an analog or continuous-time signal are compared to a number of reference voltages. In an  $N$ -bit A/D converter, there are  $2^N$  reference voltages. There are many architectural approaches for A/D converters but ultimately they each end up giving an output code corresponding to the reference voltage that the sampled voltage is either greater than and closest to or simply closest to. Another way to look at this is suppose each of the reference voltages were subtracted from the input voltage and let us call each of the differences *residues*. The code corresponding to the smallest positive residue or the residue with the smallest magnitude would be chosen. We will assume the former metric in the rest of this section.

This is illustrated in Figure 4.1 where the continuous-time signal is shown sampled and references for a 3-bit converter (8 references) are shown in the center. In this example, the reference which the signal is closest to without being greater than is 110. This is the output code for this sample. The corresponding residue of the comparison is shown on the right of Figure 4.1. The res-

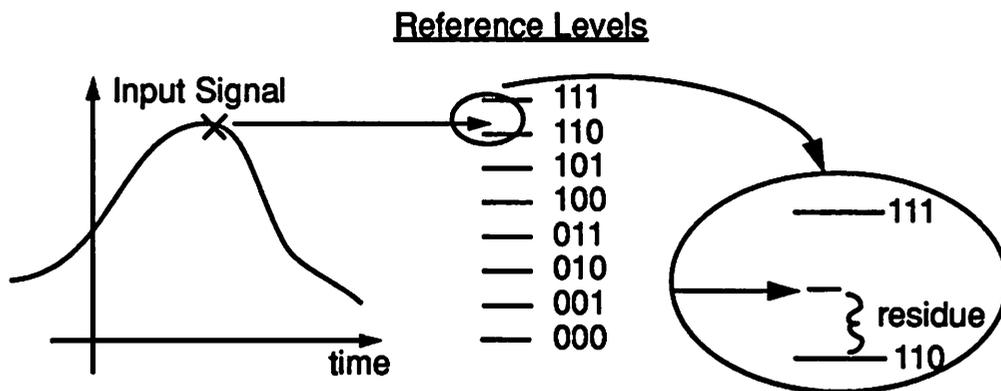


Figure 4.1 Sampled analog signal and reference comparison for a 3-bit conversion.

idue is the error in this idealized conversion. Since the input has a very small probability of being exactly the same amplitude of one of the reference voltages, the residue will usually be non-zero

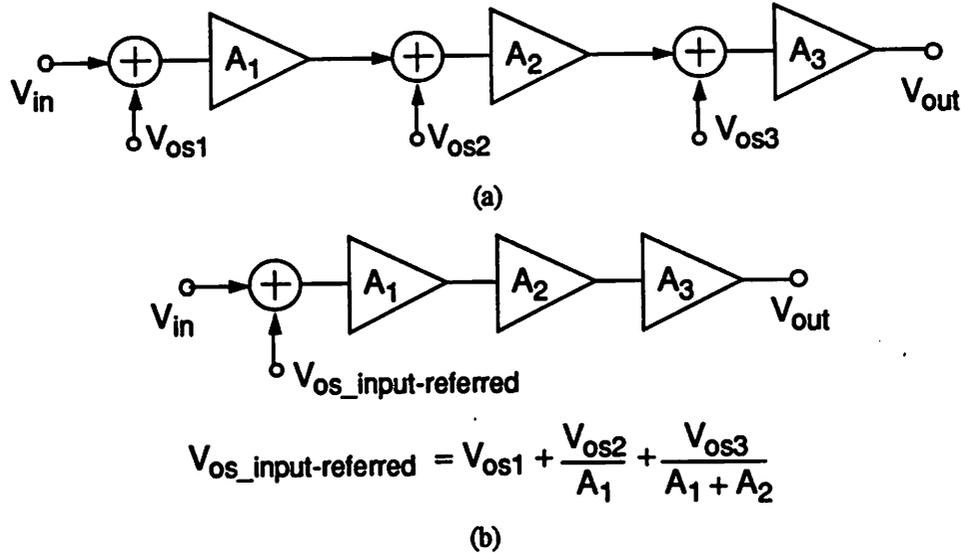
although its average value will be zero. This residue is called *quantization noise* and is a result of the finite spacing of levels due to the finite number of bits in the A/D converter. Quantization noise is often modelled as an additive noise source at the input of an ideal A/D converter. The statistics of the residue really depend on the statistics of the input signal and the number of bits in the A/D conversion. Adding another bit to the converter doubles the number of levels, lowering the maximum possible amplitude of a residue by a factor of two. Therefore, quantization noise is reduced by 6 dB when an additional bit is resolved by the A/D converter. For ease of analysis, quantization noise is usually modelled as additive white gaussian noise. This approximation may not be very accurate at low converter resolutions but greatly simplifies analyses [4].

### 4.1.2 Noise Shaping and Amplification

Any source of voltage offset or noise introduced anywhere in the signal path passes through all of the subsequent processing that the signal does from that point on through detection. Let us first consider the effect of channel processing on d.c. inputs (voltage offsets) and then of noise with spectral characteristics.

Figure 4.2 illustrates a simple channel of three amplifiers with gains  $A_1$ ,  $A_2$ , and  $A_3$ . Each of the amplifiers have offsets  $V_{os1}$ ,  $V_{os2}$ , and  $V_{os3}$ , respectively, which we shall assume are uncorrelated random variables. We can model this channel with an ideal set of offset-free amplifiers and an equivalent input-referred offset voltage source [5]. This source is found by treating each of the offsets independently and dividing each by the channel gain up to the point it is introduced and summing the resulting quantities together as shown in Figure 4.2b.

The advantage of this approach is that we can now treat the amplifiers as ideal gain elements (no offset) and the equivalent input-referred offset source as a second input to the channel. We notice in the figure that when referred back to the input, the value of an offset source is divided by the channel gain up to the point it is introduced. Thus to minimize the input-referred offset, it is desirable to have as much gain in the first stage amplifier (or block) as possible without introducing distortion.



**Figure 4.2** Input-referred offset. (a) Three gain blocks with a different input-offset voltage. (b) All three offset voltages combined into a single input-referred offset voltage.

Let us now consider the more general case where instead of a dc offset, a noise source is introduced by each block where the noise is additive and Gaussian and not necessarily white. We know from system theory that the output power spectrum  $S_y(\omega)$  due to an input power spectrum  $S_x(\omega)$  of a linear time-invariant system with transfer function  $H(\omega)$  is given by

$$S_{out}(\omega) = S_{in}(\omega) \cdot |H(\omega)|^2. \tag{4.1}$$

The spectrum of the input noise is shaped by the linear time-invariant filter. The noise spectrum can be referred to the input using the same approach as with the voltage offsets but using the square of the transfer function as the multiplying factor. However, in the case of noise, it may be both easier and perhaps more useful to consider the noise spectrum at the output. This is because given the noise spectrum, the noise variance can be calculated by integrating the spectrum over the bandwidth of interest.

The key point of this section is that noise introduced at various points in the channel undergoes the same signal processing as the signal after the point at which it is introduced. Thus, if high frequency boost is required by the signal, noise at these frequencies will be enhanced as well. Placing a lowpass filter in the front-end of a receiver, bandlimiting the noise introduced by the channel is

often desirable because it removes noise at frequencies which are of no consequence to the detection process.

### 4.1.3 Latency in Feedback Loops

Latency or processing delay through the signal path in a negative feedback system affects the stability of the loop. This affects first- and second-order loops, as well as those based on the stochastic gradient algorithm. Intuitively speaking, since a time delay corresponds in the frequency domain to a linear phase shift with no effect on the magnitude response, the phase margin of any feedback system is reduced with the introduction of any delay in the feedback path. As this delay approaches the closed loop time constant of the loop, increasing amounts of ringing in the step response and eventually instability will result.

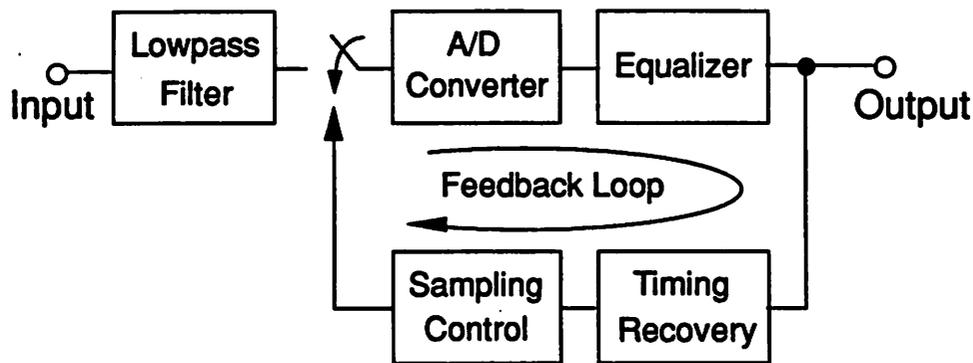


Figure 4.3 Block diagram of key receiver blocks in timing recovery path.

Let us illustrate this with an example. Suppose there is a 100 Mbit/s system with a 100 MHz local clock locked onto the incoming data through a second-order phase lock loop. Let us further assume that the phase detector is a timing function generator operating on equalized samples in the digital domain. Suppose the loop filter in the Timing Recovery block of Figure 4.3 is designed for a closed-loop bandwidth of 1 MHz with a phase margin of 45 degrees. The relationship between a time domain delay of  $\tau$  to frequency domain phase shift is given by

$$\delta(t - \tau) \Leftrightarrow e^{-j\omega\tau}. \quad (4.2)$$

For an example, suppose it took 10 periods before the timing recovery loop was updated due to latency. The phase shift that results from this delay directly reduces the phase margin. In this example where the closed-loop bandwidth is around 1 MHz, we desire to know the amount of phase shift at this frequency. Using Eq. 4.2 we find that at 1 MHz, the phase shift is 18 degrees. Therefore, a system with 45 degrees of phase margin without latency in the feedback path has only 27 degrees which would result in excessive ringing in the step response and is not a safe design margin in practice.

#### 4.1.4 Analog Discrete-time vs. Digital Processing

Both analog sampled-data and digital signal processing occurs in discrete-time so that analysis techniques for discrete-time signal processing is applicable to both. However, there are some important differences between the two. In digital signal processing, there is quantization of the input, intermediate, and output signals to one of  $2^N$  steps where  $N$  is the number of bits ( $N$  need not be the same at each point in the system). Whereas in analog discrete-time processing, these signals may take on one of a continuum of values and the only discretization is in time.

In the case of multipliers, both inputs into a digital multiplier are quantized by definition. However, in the analog domain assuming one input is continuous in amplitude, the second can either be continuous or discrete valued. With the use of the Gilbert Cell which multiplies two analog signals [5], there is no inherent quantization. The other method uses switched-capacitor circuit techniques (as those used in the FIR filter architectures presented in Chapter 5) where multipliers are realized by capacitor scaling. The input voltage is sampled on capacitors. Since the charge sampled on a capacitor is equal to the product of the input voltage and the capacitor value, by scaling the capacitor, a different multiplicand can be realized. Normally, ratios of capacitors are used in order realize *relative* multiplicands. This is because the voltage to charge conversion that results from sampling onto a capacitor is usually reversed during a subsequent charge to voltage conversion involving an operational amplifier and feedback capacitor (See Chapter 5). In cases where a multiplicand needs to be programmable, binary arrays or similar are usually used which effectively quantizes the possible values. Although there is no fundamental limitation to the granularity with which ratios of capacitors can be realized, processing variations limit the accuracy of capacitor ratios somewhere between 5 and 10 bits depending on the capacitor size and geometry. In sum-

mary, the limitation to the granularity of a digital multiplier depends on the number of bits used. Whereas, processing variations limit the accuracy which switched-capacitor multipliers can achieve.

## 4.2 Functions Required in a Baseband Communications Receiver

Let us now examine the key functional blocks required in a baseband communications receiver. Shown in Figure 4.4 is a block diagram of a conventional receiver employing adaptive equalization and sequence detection in the digital domain. It is assumed that the input signal from the channel has already been converted into a voltage by some appropriate method. Depending on the application, the amplitude of this voltage may vary from a few millivolts up to a few volts. Each of the blocks perform their respective function with the goal of recovering the information that was transmitted which is then passed on to the receiving USER.

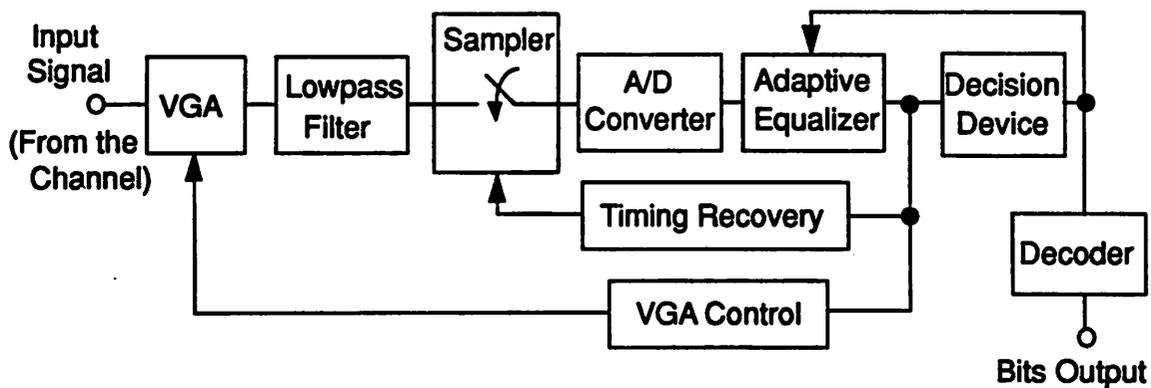


Figure 4.4 Block diagram of conventional baseband communications receiver.

In this section, we shall briefly describe the purpose of each block.

### Variable Gain Amplifier (VGA)

The peak amplitude of the signal input to the receiver can have extremely large variations depending on a number of factors such as distance from the transmitter, temperature, etc... since

they may affect the amount of amplitude loss from the transmitter. In the disk drive application, factors such as variations in flying height (height at which the read head passes over the disk) affect the signal amplitude and changes with time.

It is desirable in any signal processing application to operate with the maximum signal amplitude possible (with consideration to linearity in analog signal processing) in order to minimize additive noise effects. The *variable gain amplifier* or *VGA* as its name suggests is an amplifier with a variable gain that is set in a negative feedback loop.

The peak signal is sensed somewhere in the signal path after the output of the amplifier and depending if the signal is too large or too small, the gain is increased or decreased accordingly. Key requirements of the VGA are that it have acceptable noise and distortion performance and bandwidth over all gain settings.

### Noise Limiting Filter

Once the signal amplitude is normalized by a VGA or some other appropriate method, a noise limiting filter is usually the next processing block. In applications where discrete-time signal processing such as equalization is performed further on in the signal path, this noise limiting filter acts as an anti-alias filter. Since most practical applications require the use of adaptive equalization, we shall focus on the noise limiting filter in the context of an anti-aliasing filter.

Some key requirements of this filter are:

- Provide sufficient stop-band attenuation to reduce aliasing effects
- Minimum amplitude reduction in the passband
- Near-linear phase in the passband and through the transition region

Too much droop in the passband may require later boost by the equalizer which will also boost any noise that is introduced beyond this block and before the equalizer. The need for linear phase in the passband results from the need to limit delay distortion through the filter which may be fixed by subsequent equalization. This is important for both signal detection and timing recovery. The need for linear phase through the transition region has to do with the effect on the aliased spectrum that results after sampling. The aliased spectrum is often called the *folded-spectrum* [6,7] and is the discrete-time Fourier transform of samples of the isolated pulse through the channel taken at

the symbol rate. This pulse is the convolution of the transmit filter, channel impulse response, and receive filter. Since this pulse generally will not be ideally bandlimited, some aliasing will occur in the folded-spectrum. When the phase is non-linear through this region of overlap, the overlap areas of the spectrum may add destructively, resulting in nulls in the spectrum. These nulls are often referred to as *spectral nulls* and require large amounts of boost for equalization which results in noise enhancement.

As a result, Bessel filter responses of between 5-7 orders are often used in these applications. These responses trade transition region steepness for linear phase through much of this region.

### Equalizer

The equalizer block is perhaps the most important block in a receiver since it enables reliable transmission to occur in the presence of ISI by minimizing its effects. Since most practical channels introduce attenuation at frequencies within the signal passband, some of the job of the equalizer is to provide some boost at these frequencies in order to minimize ISI effects. This naturally results in the enhancement of noise present at these frequencies as well. Thus, the equalizer response must be a compromise of ISI reduction and noise enhancement.

In communications channels with data rates above 40 MHz, equalizers most often are adaptive in order to deal with time varying changes that occur in the channel characteristics. At lower frequencies, equalizers with varying degrees of programmability will often suffice.

Equalization can be performed in the analog or digital domains, or in a combination of both. We shall examine this further later in this chapter. In applications where it is performed digitally, a sample-and-hold and an analog-to-digital converter is required.

### Sample-and-Hold and Analog-to-Digital Converter

The sample-and-hold and analog-to-digital converter is not required in all communications receivers, only those performing discrete-time signal processing. The sample-and-hold samples the signal and holds it for conversion by the analog-to-digital converter into the digital domain.

The most common function requiring digital processing is adaptive equalization. In practical communications systems employing adaptive equalization, sampling is usually performed at the symbol rate or at twice the symbol rate (for fractionally spaced equalizers or timing recovery)

The sampling phase and frequency must be locked to the incoming signal for sampling at the desired instant. This is controlled by a timing recovery control circuit.

### Timing Recovery

The timing recovery circuit generates a signal which controls a local oscillator. This oscillator generates a timing reference which triggers the sampling control signal to the sample-and-hold block. There are many ways to implement the timing recovery function and some of these shall be explored later in this chapter.

### VGA Control

This block monitors the signal somewhere after the VGA and feeds back the signal which controls the VGA gain.

### Decision Device

The decision device operates on the equalized samples and makes a determination of what the received symbol are. This may be a symbol-by-symbol decision as in the case of a slicer or it may consider a sequence of received symbols to make its decisions for increased noise immunity as done in a sequence detector.

### Decoder

This block undoes any coding that has been performed on the signal for spectrum or error control and outputs the data to the receiving user.

## **4.3 Filter Implementation Approaches**

From a block level perspective, the two required filtering functions required in the receiver is that of lowpass noise limiting and equalization. Implementation of these filtering functions are

perhaps the most important decisions required at the implementation level in baseband receiver design<sup>1</sup>. Most of the control and detection algorithms can be implemented in either the analog or digital domains. The algorithms for functions such as timing recovery and Viterbi sequence detection most often operate on a discrete-time signal. Switched-capacitor circuit techniques are particularly well suited for analog implementation of these algorithms [9]<sup>2</sup>, although they are even more readily performed in the digital domain. The choice of how, in particular, the equalization is to be performed and in what domain, analog or digital, is what determines whether or not an A/D converter will be needed and where these algorithms (particularly Viterbi) will be performed. If an A/D converter is present for digital equalization, then a digital implementation of the Viterbi detector and possibly the timing recovery algorithm would tend to follow because of the ease of implementation. However, if analog equalization is sufficient for both these algorithms, then it may be cheaper from the perspective of power to perform the processing in the analog domain because an A/D converter would not be required. We will return to this discussion at the end of Section 4.4 after presenting a few of the possible receiver architectures in that section.

In the next few sections, we will examine analog filter techniques used in high speed receivers.

### 4.3.1 Analog Filtering Approaches

Most analog filters used in practice are based on either Sallen-Key sections, biquadratic sections, or ladder filters. Sallen-Key sections tend to be used in cases requiring only modest amounts of anti-alias filtering and have the advantage that they may be realized without an amplifier. Biquadratic and ladder filter implementations on the other hand require opamp or high gain transconductor. We shall examine first the Sallen-Key section then the biquad filter, and then the ladder filter. Following this, we shall look at implementation of biquads and ladder filters using switched-capacitor techniques and continuous-time filter techniques. And finally we shall compare implementation approaches for high speed filtering, focusing on advantages and limitations of each approach.

---

1. We are restricting our discussion to simple channels with receiver structures as shown in Figure 4.4. Much more complicated architectures are required in simultaneous bi-directional communications over a single wire pair as required in high speed data communications applications [8].

2. This assumes PR-IV signalling for Viterbi detection which is relatively simple to implement with only 2 states. Higher order PR approaches such as EPR-IV with 8-states may require implementation in the digital domain.

### 4.3.1.1 Sallen-Key Sections

In many applications such as anti-aliasing, particularly where a sharp transition band is not required, it may still be desirable to have a second-order response because it allows the realization of a linear phase response. Recall that a linear phase response is not possible with a cascade of two simple poles since the composite phase response is the superposition of the individual phase responses.

The  $Q$  of a complex pole pair is the ratio of the imaginary part to the real part and is a rough measure of the sensitivity of the filter to component variations. A  $Q$  greater than one means that in the  $s$ -domain, the pole pair is closer to the imaginary axis than the real axis. The higher the pole- $Q$  of a complex pole pair, the more sensitive it is to variations in its actual location. One way to view this is to consider that the frequency response can be evaluated by traversing up the  $j\omega$ -axis and dividing the product of the distances from each point on the  $j\omega$ -axis to the zeros by the product of the distances to the poles. Therefore, poles and zeros near the  $j\omega$ -axis will have more of an impact on the response and thus the filter is more sensitive to their locations. With a second-order response, linear phase results in a filter with a  $Q$  of about unity which is a filter with low sensitivity. So such a filter has both the advantage of linear phase and low sensitivity.

Sallen-Key sections are well suited to these applications and are particularly attractive because only one active gain element is necessary and in some applications, it need not have very much voltage gain. In fact, a voltage follower can be used in many applications. Figure 4.5 shows a block diagram of a Sallen-Key section. The resulting transfer function and design equations can be found in [10,11]. In order to illustrate the attractiveness for simple anti-aliasing applications, let us consider the specific case where  $K$  is unity, the resistors and capacitors are equal, and a Butterworth response is desired for a maximally flat passband and linear phase. Under these conditions, the transfer function reduces to

$$H(s) = \frac{\omega_o^2}{s^2 + \left(\frac{\omega_o}{Q}\right)s + \omega_o^2} \quad (4.3)$$

where the complex poles reside at frequency  $\omega_o = \frac{1}{\sqrt{2}RC}$  and the Q of the complex pole pair is  $\frac{1}{\sqrt{2}}$ , independent of the value of R and C [10].

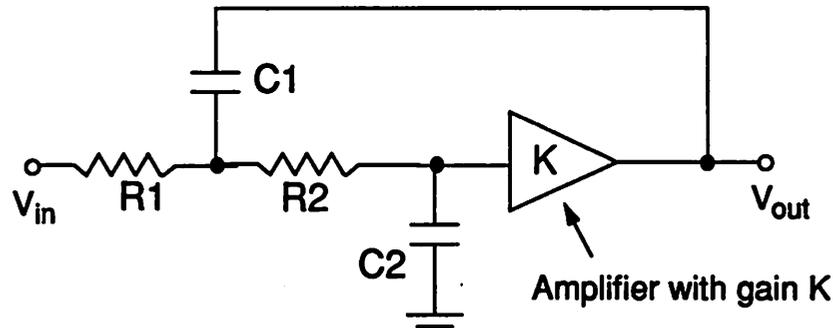


Figure 4.5 Sallen-Key Filter.

#### 4.3.1.2 Biquadratic Sections

A very important class of filters for realization of general second-order transfer functions are *biquadratic sections* or *biquads*. These are filters realize the biquadratic transfer function

$$H(z) = -\frac{a_2 z^2 + a_1 z + a_0}{b_2 z^2 + b_1 z + 1}, \quad (4.4)$$

which, assuming the frequencies of interest are much lower than the sampling rate can be approximated in the s-domain by

$$H(s) = -\frac{K_2 s^2 + K_1 s + K_0}{z^2 + \left(\frac{\omega_o}{Q}\right)s + \omega_o^2} \quad (4.5)$$

where the complex poles reside at  $\omega_o$  with pole-Q equal to Q. As is evident in Eq. 4.5, this general form of the biquadratic transfer function has an independent complex pole pair in both the numerator and denominator. Thus a filter based on this transfer function can have either a lowpass, bandpass, notch, or high-pass frequency response.

A block diagram of a circuit implementation of Eq. 4.5 is shown in Figure 4.6. All elements have been normalized by equal valued capacitors  $C_A$  and  $C_B$ . Therefore,  $K_0$  and  $K_2$  are unit-less

and  $K_1$  has units of 1/ohms. Referring to Eq. 4.5, four different classes of filter responses are pos-

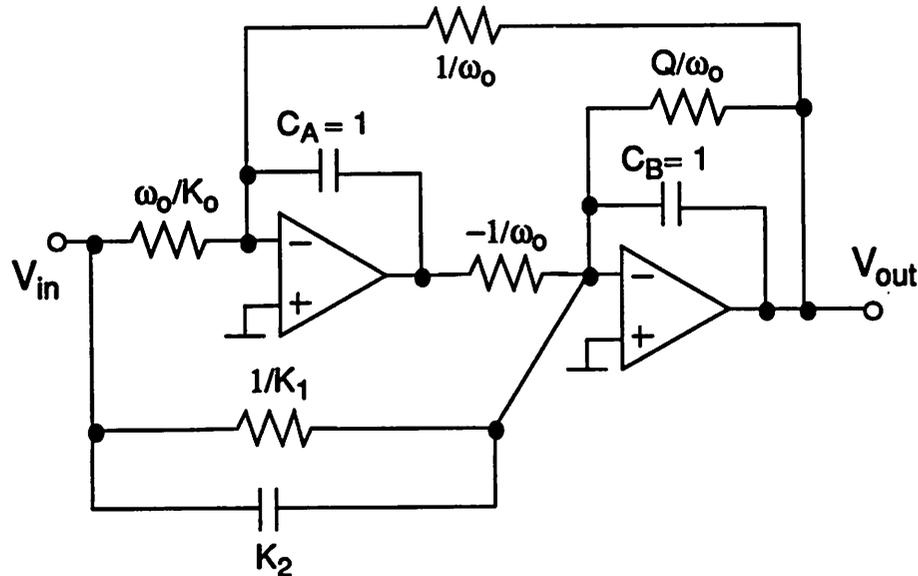


Figure 4.6 Biquad schematic. All elements normalized by units of picofarads. (Taken from [10])

sible simply by removing resistors and/or capacitors as follows:

**Lowpass Filter Response:**  $K_1 = K_2 = 0$ ,

**Bandpass Response:**  $K_0 = K_2 = 0$ ,

**Notch Response:**  $K_1 = 0$ ,

**High-pass Response:**  $K_0 = K_1 = 0$ .

The key problem implementation of the active-RC biquad shown in Figure 4.6 is realization of the active-RC integrator. Most monolithic CMOS implementations of analog filters avoid the use of resistors whenever possible for a two reasons. First, MOS devices have relatively low drive capability and it is difficult to design on-chip amplifiers with high gain and bandwidth capable of driving resistive loads. Second, since the integrator time constant is the RC product, large variations may occur in this time constant as the values of R and C vary independently with temperature and process variations. Furthermore, the capacitors used are normally parallel plate capacitors and thus no tuning of the time constant is possible since both the resistors and capacitors are realized

with passive components. The continuous-time and switched-capacitor filter techniques to be described in later sections solve these problems.

### 4.3.1.3 Ladder Filter

Biquadratic filters are by nature second-order sections. A first-order section can easily be implemented with an active integrator or simple R-C network. In order to build higher order filters, any number of biquads and first-order sections can be cascaded. However, higher order filters usually require higher pole-Qs which are very sensitive to component variations. Filters based on ladder networks allow the implementation of higher order filters with low sensitivity to component variations. It has been shown that doubly terminated RLC ladder networks can be designed with near zero sensitivities in their passbands [14]. Since this sensitivity is related to the resultant transfer function  $H(s)$ , it follows that active filters realizing ladder network-based transfer functions should have a corresponding robustness in their passbands to component variations.

There are many references containing RLC ladder networks for varieties of filter orders and response types which make it very simple to design filters based on these structures [11,12]. These references contain tables of pole and zero locations along with values for resistors, capacitors, and inductors for one-Hertz filters. The idea behind having all values scaled to a one-hertz bandwidth filter is that components can be scaled in order to scale the filter response. All responses can be found including everything from Butterworth to Bessel (wraps around the alphabet) and for most practical filter orders. Unfortunately monolithic implementation of these filters directly is not possible due to the absence of high quality inductors in monolithic form and variations in resistor and capacitor values due to processing and temperature variations. Therefore, these networks need to be transformed into a form suitable for monolithic implementation.

An example of an RLC network is shown in Figure 4.7a. This network corresponds to a doubly-terminated LC fifth-order all-pole low-pass filter where  $R_1$  and  $R_2$  are the termination resistors. One technique for transforming this network into an active-ladder network using active elements and no inductors is to use flow-graph techniques [13]. Using the techniques described in [13], the circuit is labelled with node and loop equations which allow the network to be transformed into a flow-graph as shown in Figure 4.7b. The objective is to obtain a flow-graph requir-

ing the operations readily implemented in monolithic form which are addition, subtraction, scaling (multiplication by a constant), and integration as shown in the figure.

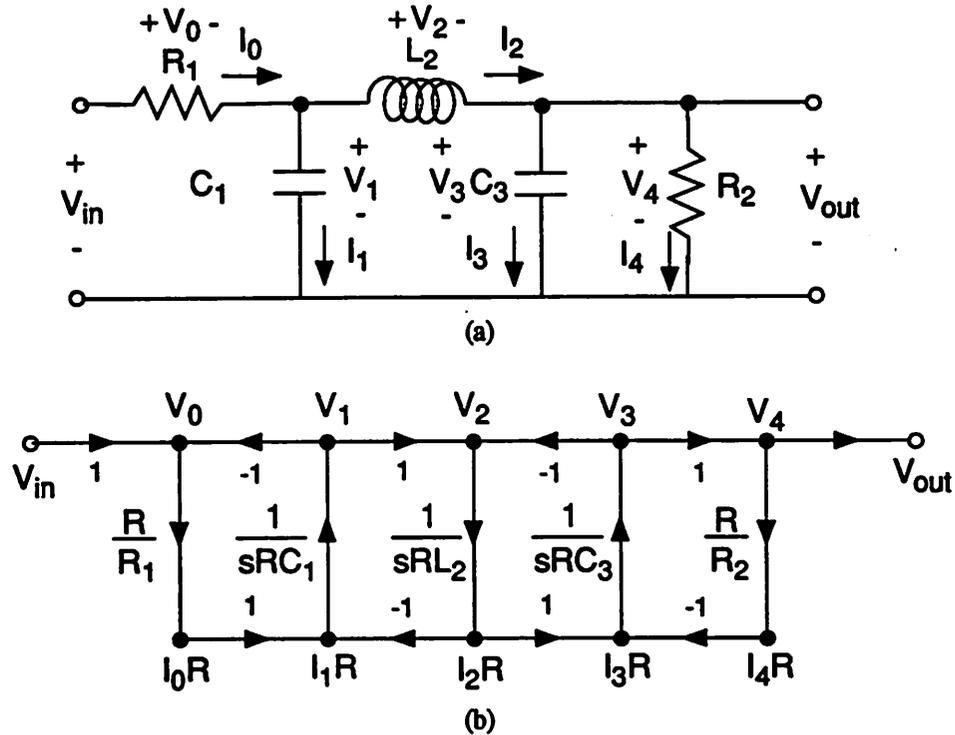


Figure 4.7 Ladder network transformation into flow-graph form. (a) Ladder network with node and current definitions. (b) One of many equivalent flow-graph diagrams.

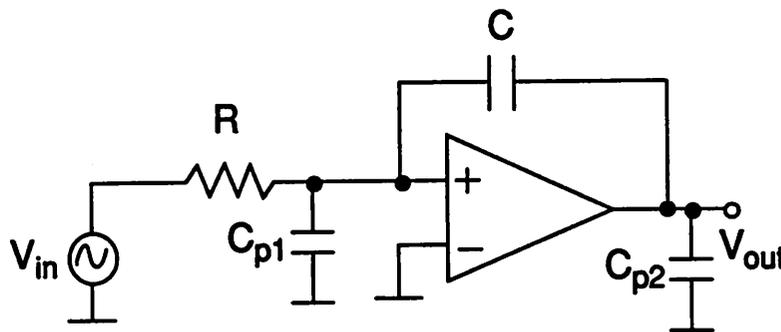
The most difficult of the four operations (addition, subtraction, scaling, and integration) needing to be implemented is that of integration. After that is solved, the other three follow quite simply. In the next section we will consider different ways to implement the integration operation using monolithic circuit approaches in both continuous- and discrete-time.

### 4.3.2 Implementation of Monolithic Analog Filters

As seen in the previous two sections, both biquads and ladder filters can be transformed into networks comprised entirely of the operations of addition, subtraction, scaling, and integration. These operations can robustly be implemented using a variety of techniques in both continuous- and discrete-time. In the section, we will examine some of the different ways in which the integration operation can be implemented.

**Active Integrators**

Before proceeding, it is useful to consider the active integrator using a resistor, capacitor, and operational amplifier. The effect of the opamps is to reduce the sensitivity of this circuit to parasitic capacitance. There are two parasitic capacitors shown in the figure,  $C_{p1}$  is connected to the summing node of the opamp and  $C_{p2}$  is connected to the output of the opamp. The effect of  $C_{p1}$  is divided by the open loop gain of the opamp and  $C_{p2}$  appears as a load capacitance that is driven by the opamp output so does not affect the integrator time constant. The advantages of the opamp diminish as the amplifier loop gain begins to drop off due to its finite bandwidth. Therefore, the advantages of the opamp can only be realized in the frequency range that the opamp appears nearly ideal, limiting the frequency range of opamp-based approaches.



**Figure 4.8** Active integrator using an operational amplifier.

The key point is that opamp-based approaches have an operating window where the opamp reduces parasitic sensitivities. In order to be used in high speed applications, they must consume large amounts of power in order to achieve high-bandwidths. And even then, they tend to limit the achievable frequency range possible in a given technology.

### 4.3.2.1 Continuous-Time Filters

In most high speed CMOS continuous-time filter implementations, due to the variations present in CMOS processing, approaches tend to use fixed capacitor and variable resistance techniques in order to control the integrator time constants [15]. This is because it is much easier to change the effective value of a resistor as compared with changing the value of a capacitor in CMOS technology. Two ways that have been proposed for this are MOSFET-C and transconduc-

tance-C filters. These approaches are shown conceptually in Figure 4.9. In the integrator used in

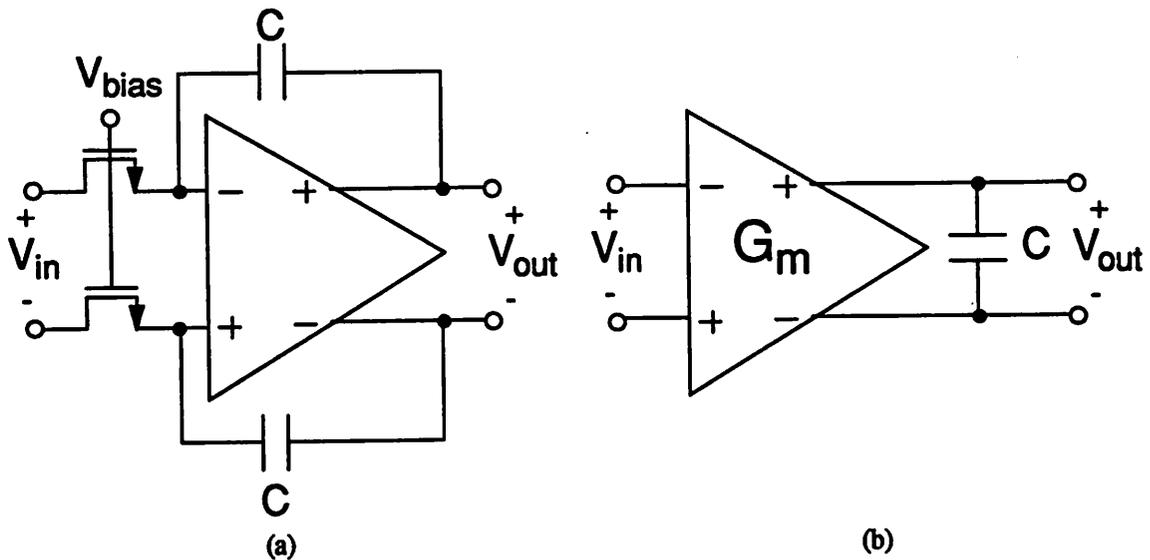


Figure 4.9 (a) MOSFET-C integrator. (b) Transconductance-C integrator.

MOSFET-C filters [16] shown in Figure 4.9a, the MOS devices connected to the input operate in the linear region and by controlling the gate bias of these devices, their on-resistance controls the integrator time constant. However, driving resistors is difficult to do in CMOS, particularly at high frequencies. Therefore, in the remainder of this section, we will focus on implementation of the transconductor in transconductance-C filters shown in Figure 4.9b where a transconductor drives a load capacitor. Two of the ways in which to implement integrators and primarily the transconductor are shown in Figure 4.10.

In the approach shown in Figure 4.10a [22], MOS devices M3 and M4 act as degeneration resistors for the input devices. As the product of the transconductance of M1 and the on-resistance of M3 get much larger than unity, the transconductance of the input device pair approaches  $1/R_{on\_M3,M4}$ . By placing another transconductor stage in parallel, depending on the polarity, addition and summing can easily be performed in the current domain. A number of filter implementations which have been reported use the structure of Figure 4.10a [17,18]. One of the key drawbacks of this approach is the relatively small signal levels required in order to maintain linearity of the input stage which is on the order of hundreds of millivolts.

In the approach shown in Figure 4.10b [19], The input is applied differentially to the gates of M1 and M2 which are biased in the linear region. Cascode devices M3 and M4 fix the  $V_{ds}$  of the input devices. As  $V_{gs}$  of the input devices are modulated, their on-resistance is modulated. Since the  $V_{ds}$  is fixed, the resistance modulation results in a modulation of the current which is reflected to the load capacitor C. Due to the finite transconductance of M3 and M4, current modulation changes their  $V_{gs}$  which affects the resultant  $V_{ds}$  across devices M1 and M2. This results in distortion. One implementation presented in [20] uses BiCMOS technology with M3 and M4 implemented using bipolar NPN devices. This greatly reduces the modulation of the  $V_{ds}$  of devices M1 and M2 and resulted in linearity performance of .24% THD with an input voltage of 2 Vp-p.

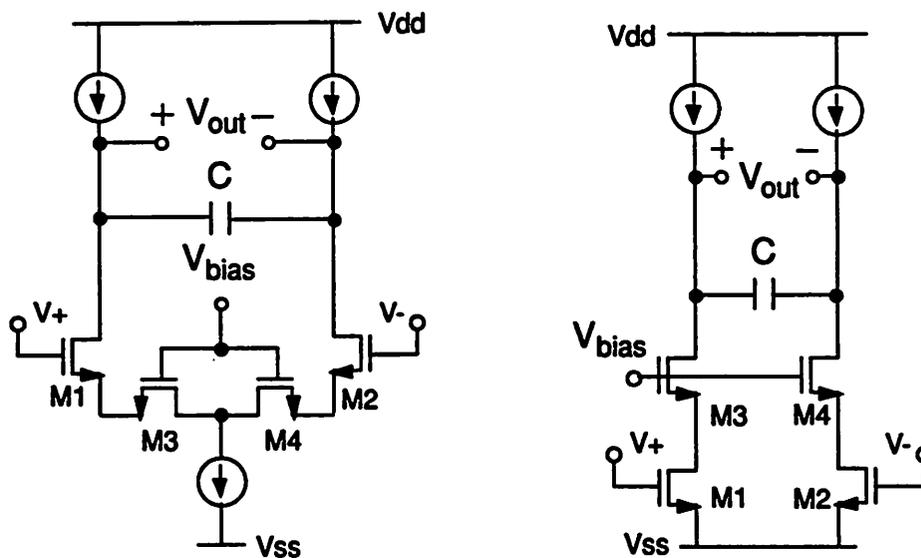


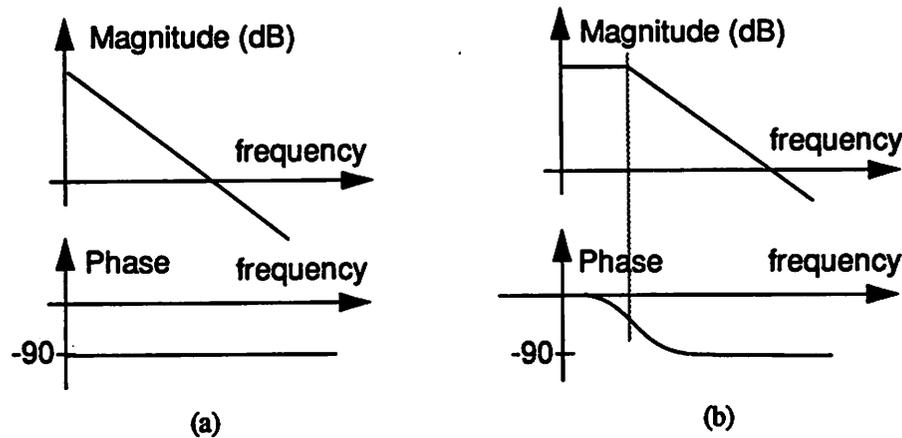
Figure 4.10 Transconductance-C integrators. (a) Degeneration control. (b) Input modulates on-resistance

Both of the integrator approaches shown in Figure 4.10 suffer from parasitic capacitance on the output node which adds directly to the integrating capacitor. The approach used in [20] had the transconductor of Figure 4.10b feed an active integrator. As a result, a robust parasitic insensitive design was implemented but at the cost of filter bandwidth.

#### Tuning Requirements of Continuous-Time Filters

An ideal integrator has the property that its gain at d.c. is infinite and there is a phase lag through the unity gain frequency of 90 degrees. The practical integrator has finite gain at d.c. and

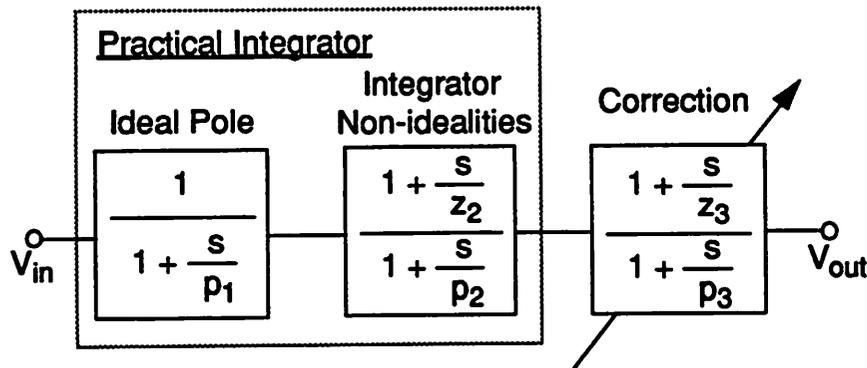
the phase at unity gain may be larger or smaller than 90 degrees. A phase larger than 90 degrees is referred to as *excess phase* or a *phase-lag* and is the result of a non-dominant pole contributing extra phase shift at the unity gain frequency. The other situation of a *phase-lead* occurs when due to finite gain, the phase does not reach 90 degrees at the unity gain frequency. The gain and phase plots for the ideal integrator and the an ideal practical integrator with 90 degrees of phase shift at unity gain are shown in Figure 4.11 [21].



**Figure 4.11** Integrator magnitude and phase responses. (a) Ideal integrator. (b) Practical integrator.

The effects of phase-lag and lead can be modelled as shown in Figure 4.12. Along with the ideal transfer function is another pole and zero, where the pole models the effect of a non-dominant pole and the zero, the effect of finite d.c. gain or a feed forward path in the circuit. Tuning methods are used to add an additional pole and zero in order to cancel the effects of these non-idealities and get 90 degrees of phase shift at the unity gain frequency. Many techniques have been proposed to perform this tuning [20,22,23]. A key problem of tuning the integrator  $Q$  is to have the pole and zero that are introduced track their respective non-idealities over temperature and supply variations. The approach described in [20] is a particularly robust approach to achieving this end.

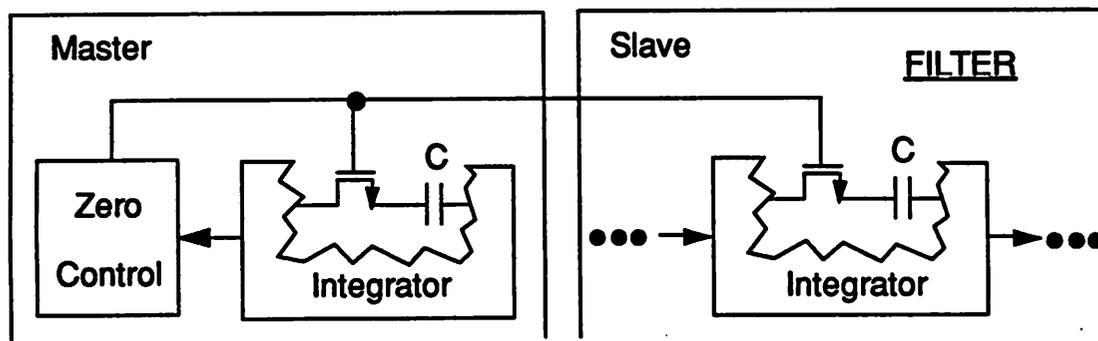
In general, two tuning loops are required. One is for tuning the effective  $R_s$  in the integrator time constants which set the pole locations. The other is for tuning the effective  $Q$  of the integrators in order to achieve -90 degrees of phase shift at the unity gain frequency of the integrators.



**Figure 4.12** Block diagram of a practical integrator with the ideal pole  $p_1$  and a pole-zero pair modelling the non-ideal effects. The correction pole-zero pair is introduced to cancel the effects of both a phase lead and lag.

### Practical Limitations

The high frequency performance of continuous-time filters is limited by the accuracy with which the integrator Qs can be tuned. Most techniques use a master-slave approach shown in the conceptual diagram of Figure 4.13. The figure illustrates an example where in the master circuit, the integrator Q is indirectly sensed in a negative feedback loop and the location of an extra zero is placed to cancel the effects of the non-dominant pole. This control voltage is then used in the slave integrators which are the integrators which make up the filter. One way the integrator Q can be sensed is to connect two integrators back-to-back creating a resonator. Only when the phase shift through each of the integrators is -90 degrees will the poles of this system lie on the  $j\omega$ -axis and a steady-state oscillation condition result [23]. The high frequency limitation is set by the accuracy with which the cancellation zero can be placed in the actual filter.

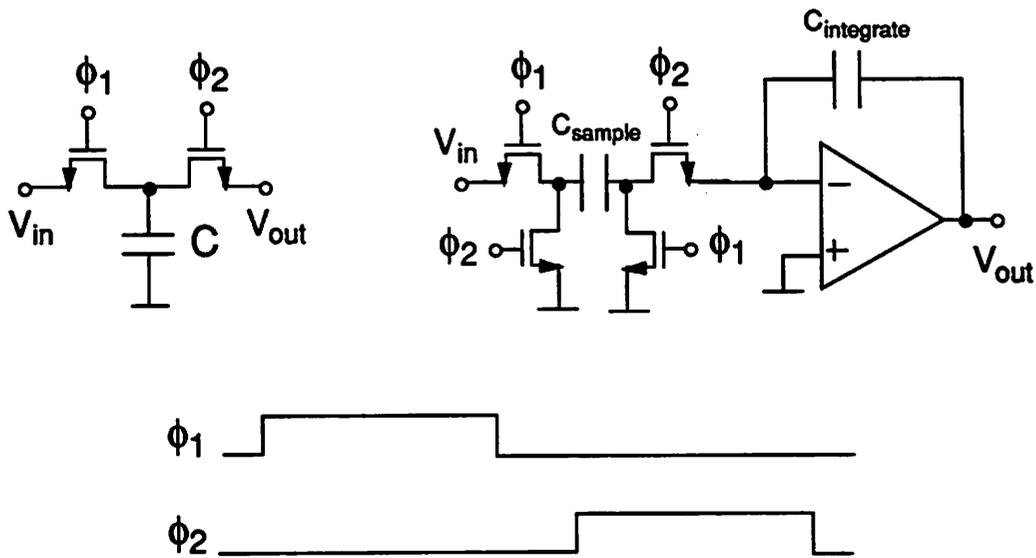


**Figure 4.13** Master-Slave tuning for control of the zero location to cancel a non-dominant pole.

As a result of finite matching between the components in the master and the slave, there is a finite accuracy with which the phase can be controlled, setting limits on the attainable filter  $Q$ s. Thus, one of the key trade-offs in continuous-time filter design is that between filter- $Q$  and bandwidth.

### 4.3.2.2 Switched-Capacitor Filters

Another way integrators can be implemented in CMOS is the use of switched-capacitor (SC) filters where resistors are replaced with switched-capacitors as shown in Figure 4.14a [24]. During  $\phi_1$ , the input capacitor is charged up to the input voltage and during  $\phi_2$ , this charge is passed to the right allowing the equivalent current flow of a resistor of value  $1/fC$  where  $f$  is the clock frequency. As a result, an integrator can be implemented without resistors. One implementation of an SC integrator using an improved implementation of the switched capacitor to reduce parasitic sensitivity [25] is shown in Figure 4.14b. The resulting integrator time constant is at  $C_{integrate}/fC_{sample}$  where the capacitors are as shown in the figure. In this result, we find that the time constant is dependent only on ratios of capacitors and the clock frequency  $f$ . These are two of the primary advantages of switched-capacitor filters.



**Figure 4.14** Switched-Capacitor (SC) Integrator. (a) SC resistor. (b) Parasitic insensitive SC resistor in integrator circuit. (c) Periodic non-overlapping clock waveforms with frequency  $f$ .

### 4.3.2.3 Comparison of Continuous-Time and Switched-Capacitor Filters

In this section, we have seen two important ways that higher-order monolithic filters are implemented: a cascade of biquadratic filter sections or active ladder filter implementations. In either case, implementation was reduced to realization of the operations of addition, subtraction, scaling, and integration. Assuming the first three were simple once the integration operation could be performed, we focused on ways in which integrators could be implemented in both continuous- and discrete-time. Instead of performing a side-by-side comparison, we will list some of the key advantages and disadvantages of both approaches.

The majority of high speed continuous-time filters reported in CMOS and BiCMOS have been transconductance-C filters. Higher speed operation is possible compared with SC approaches. Tuning is required for both control of the integrator  $Q$ s and control of the filter bandwidth and performance is limited by the finite ability to do so. Parasitic capacitance on transconductor outputs adds directly into time constants, affecting the filter response characteristics. For linearity reasons, signal levels need to be kept relatively small.

The speed limitations of switched-capacitor filters come from amplifier settling times. Due to this constraint, the bandwidths of SC filters reported to date are dramatically lower than those of continuous-time filters. The frequency response is controlled by the clock frequency and capacitor ratios. The limits to performance are due to capacitor matching, finite amplifier gain, and bandwidth.

## 4.4 Architectural Approaches

As the feature sizes of CMOS technologies continues to be reduced, the drive will be to do more and more signal processing in the digital domain due to its robustness and the scalability of digital circuits. However, this increases the requirements of the analog-to-digital converter which can result in higher overall system cost and complexity as compared to implementations with analog processing. With the goal of maximizing the cost/performance ratio, the circuit/system designer must consider various partitions of signal processing in receivers. In the next few sections we shall examine a few important approaches and consider the advantages and disadvantages of

each. We will begin with the Ideal Front-End Receiver and discuss practical limitations to its implementation before presenting some practical approaches.

#### 4.4.1 Ideal Front-End Receiver

In Figure 4.15 is a block diagram of an ideal front end receiver for a baseband communications channel. It is said to be ideal because the signal to noise plus ISI ratio is maximized. The first block is a matched filter whose impulse response is matched to the receive pulse shape. Following this block is the sampler which may be at the symbol rate or higher if a fractionally-spaced equalizer is used. The matched filter combined with the sampler sampling at the proper phase becomes a *correlation receiver* [6,26]. The next block is an equalizer which removes intersymbol interference. The equalized samples are passed on to two other blocks. The decision device will ideally be a sequence detector (such as Viterbi) which may allow coding gain. If it is a simple slicer, the decisions will at least be made on the best possible information. The other block is the timing recovery block which generates a timing function to drive the sampling function.

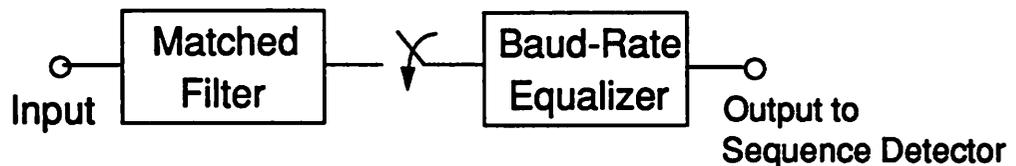


Figure 4.15 Block diagram of the ideal front-end receiver.

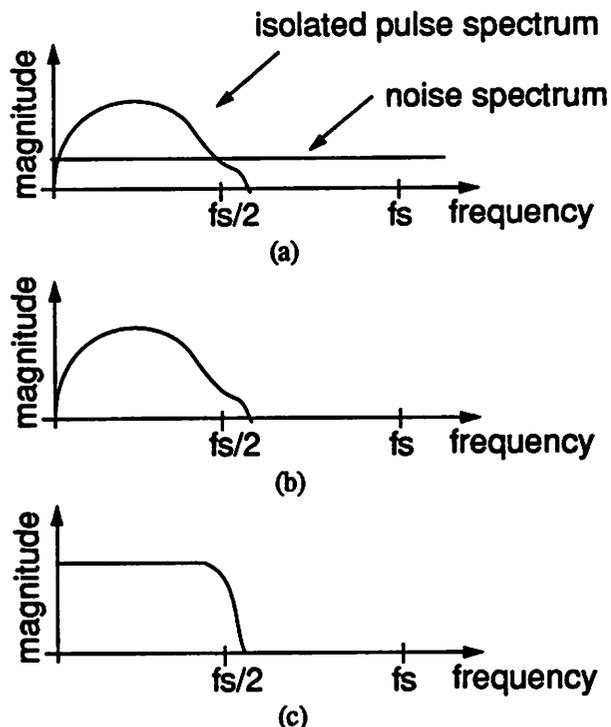
#### 4.4.2 Practical Limitations

There are a number of limitations in the implementation of practical communication receivers which limit the realization of the ideal receiver. This is because the received pulse shape is rarely known precisely enough to specify the proper response in order to achieve a matched filter. Therefore, the first block in practical receivers is seldom a matched filter. This is due mainly to the dispersive nature of the channel and variations in its characteristics. (The fractionally spaced equalizer has been shown to be equivalent to a matched filter followed by a baud rate equalizer. However, this is difficult to achieve in high data rate channels due to the cost of sampling at twice or more than the symbol rate.) Instead, a compromise is struck with the use of a lowpass filter.

Since we assume a baseband channel, the signal frequencies of interest lie from somewhere around dc up to somewhere near the symbol rate, bandlimiting the signal helps to reduce the signal-to-noise ratio in the samples.

In Figure 4.16, the spectrum of an isolated pulse is shown together with a noise spectrum. The noise is assumed to be additive white gaussian noise. The frequency response of both a matched-filter and a lowpass filter are also shown. The matched-filter emphasizes parts of the signal spectrum where there is more energy in the signal and attenuates where there is less. In this way, the signal-to-noise ratio of samples taken in the proper phase is maximum. The lowpass filter is a compromise in cases where the matched-filter response is not known or easily realizable. In the example shown, the compromise lowpass filter will pass noise at low frequencies and near  $f_s/2$  that the matched filter would have removed or reduced.

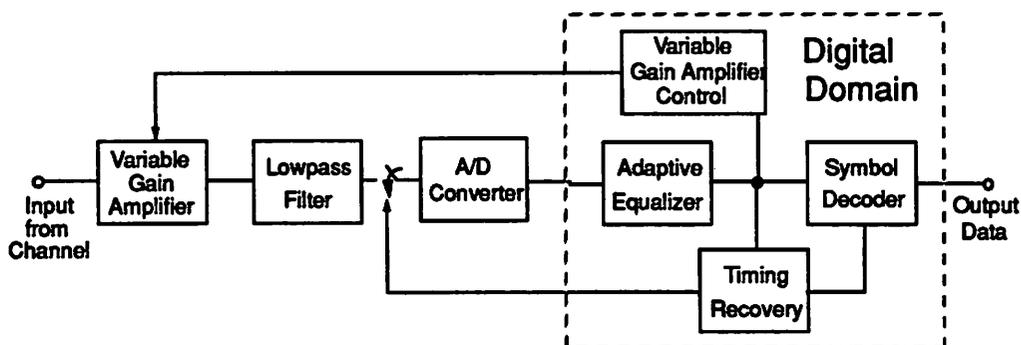
The key requirements of the lowpass filter is that it provides sufficient attenuation of out-of-band noise and has a linear phase response through the passband and through as much of the transition band as possible.



**Figure 4.16** (a) Spectrum of isolated pulse and noise. (b) Matched-filter frequency response. (c) Lowpass filter frequency response.

### 4.4.3 All Digital Receiver

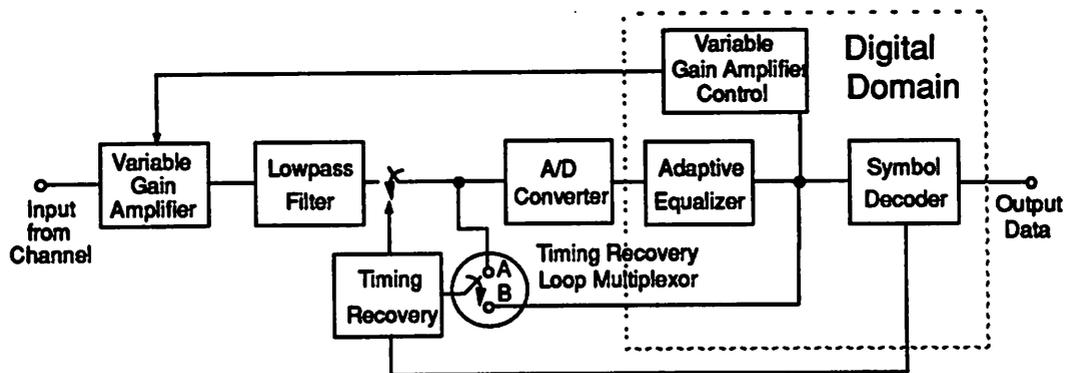
A block diagram of the *all digital receiver* is shown in Figure 4.17. The description *All Digital Receiver* is somewhat of a misnomer because in the front-end of this receiver are two very important high performance analog blocks. The first block is the lowpass anti-alias filter that takes the place of the matched-filter. The second is the A/D converter which converts samples of the analog waveform into the digital domain. Between these blocks is the sample-and-hold which may be a stand alone block or folded into the A/D converter which follows.



**Figure 4.17** Block diagram of an all digital receiver.

The timing recovery loop begins with the sampler and is followed by the A/D converter, adaptive equalizer, decision device, timing recovery algorithm, and is converted back into the analog domain through a digital-to-analog converter (DAC). This DAC function is implicit in the Timing Recovery block and is not shown separately. The DAC output is filtered and is what controls the local oscillator which generates the sampling phase. There is quite a bit of latency in the timing recovery feedback loop which subsequently requires a small loop bandwidth for stability. This slows the initial acquisition of the timing phase because the time required for initial acquisition tends to be inversely proportional to the loop bandwidth [27]. Since a large loop bandwidth is desirable during initial acquisition for quick lock onto the frequency and phase of the data signal, another approach is sometimes needed.

One solution is the use of two loops for timing recovery, one for initial acquisition and the other for tracking as shown in Figure 4.18. During initial acquisition, a preamble such as a sine wave of known frequency is received in order to maximize the information for the timing recovery loop. During this period, the switch is in position A and the portion of the loop through the ADC and equalizer is by-passed, reducing the latency and allowing a high bandwidth loop. This is possible because a sine wave can be generated with no apparent ISI and thus equalization is not required for lock onto the preamble. The switch is flipped to position B for tracking of random data requiring equalization.



**Figure 4.18** All digital receiver with two timing recovery loops. Position A used during initial acquisition and position B used during data reception.

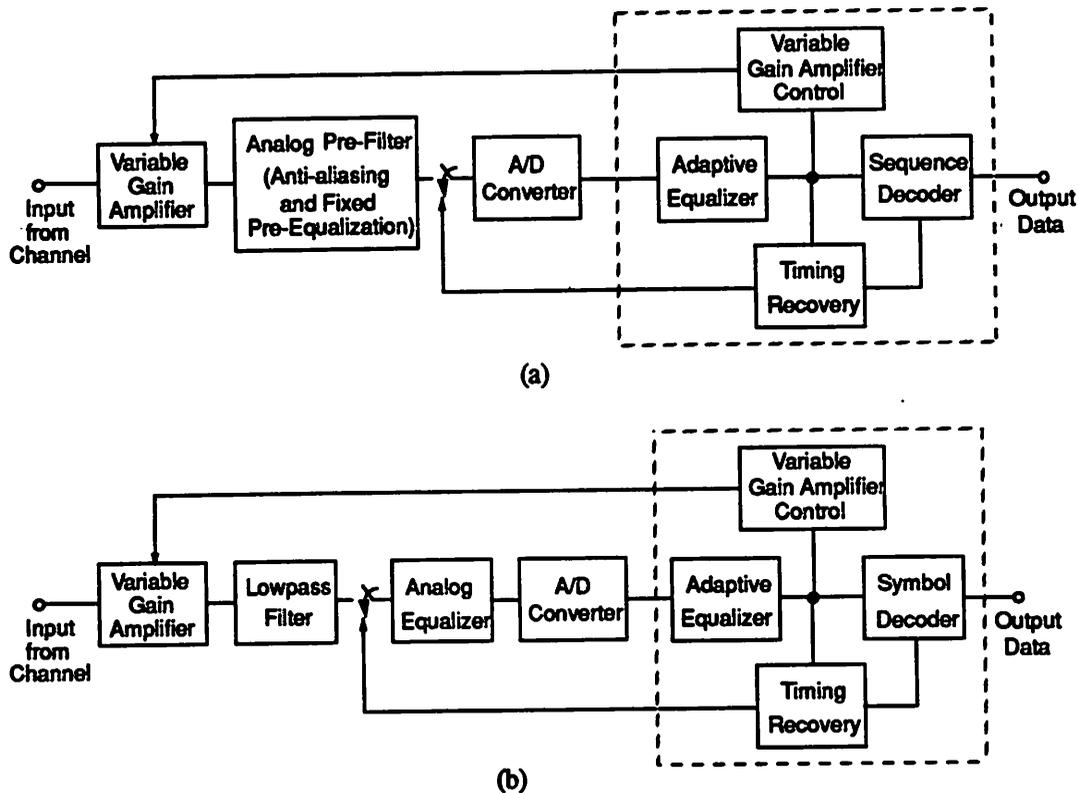
All of the signal processing in the all-digital receiver is performed in the digital domain, providing the advantages of robustness to supply and system noise and ease of programmability and adaptability. As feature sizes of CMOS technologies continues to decrease, the trend will continue to be toward more and more processing in the digital domain because of the ease of scalability of digital circuitry and the inherent robustness associated with digital implementation. However, due to the inherent analog nature of the signal received from the channel, analog processing will always be required in the front-end. In the following sections we shall examine approaches with differing degrees of analog signal processing which can impact both the analog-to-digital conversion and the timing recovery implementation.

#### 4.4.4 Digital Receiver with Analog Pre-equalization

In the digital receiver with analog pre-equalization, some of the equalization is performed by either a fixed or programmable equalizer in the analog domain. An adaptive digital equalizer is still required to remove residual ISI. Analog adaptive equalization has been shown to be useful only in limited situations [28,29]. The complexity of digital equalizers goes up less than linearly with the number of taps making it practical for equalizers with more than just a few taps. The complexity of analog adaptive equalizers grows much more quickly than linearly as we shall see in Chapter 5. Since further adaptive equalization is still required, it is still necessary to have digital signal processing. Two approaches for mixed signal receivers are shown in Figure 4.19. In the approach shown in Figure 4.19a, the equalizer is folded into the anti-aliasing filter resulting in the Analog Pre-Filter block as shown. This would be implemented using continuous-time filter techniques. These filters may or may not be combined depending on the requirements and the implementation. In the approach shown in Figure 4.19b, the analog equalization is shown in discrete-time. This would probably use switched-capacitor techniques. One approach for CMOS implementation is described in Chapters 5 and 8.

The choice between continuous- and discrete-time processing would depend on factors such as bandwidth, technology, power, and performance requirements. There is a window in which discrete-time techniques are attractive as the performance of continuous-time techniques becomes limited due to parasitic capacitance and finite tuning control. At this point, the switched-capacitor circuit techniques described in later chapters may be advantageous since the frequency response

depends mostly on capacitor ratios. However, at speeds where the amplifier settling becomes an issue, once again continuous-time approaches will become viable. However, filters in this region of speed will be limited to simple non-critical filters.



**Figure 4.19** Digital Receiver shown with two analog approaches to pre-equalization. (a) Continuous-time analog pre-filter. (b) Discrete-time analog equalizer.

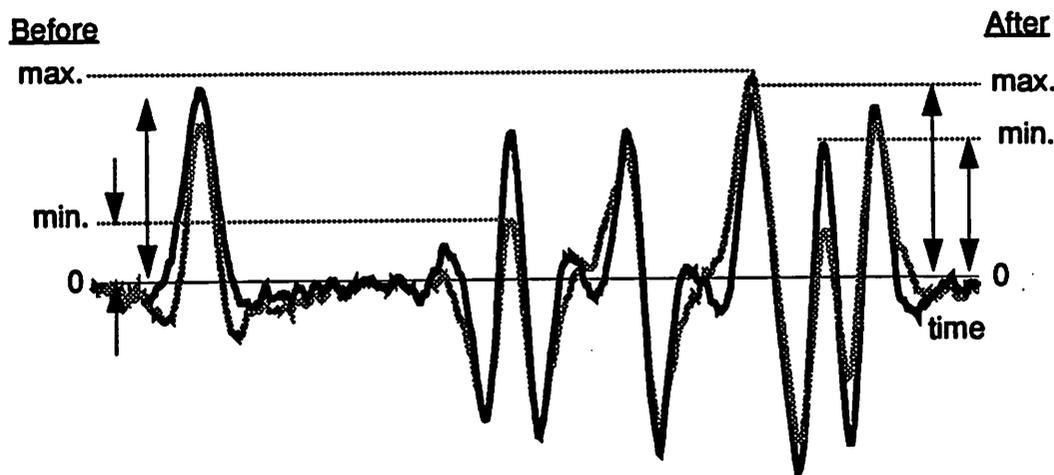
There are some important advantages to performing some of the equalization in the analog domain in systems based around digital adaptive equalizers. The key function of equalization is to process the spectrum of the receive signal so as to reduce or remove the effects of intersymbol interference. Since we assume a linear channel, the equalizer can be viewed as an extra filter included before the decision device which performs its correcting function on the received signal spectrum. Depending on the channel, the equalizer may be a complicated composite of different filters such as a magnitude correction filter and a phase correction filter. When analog pre-equalization is used, portions of these filters are performed in the analog domain. Suppose for instance that a well known amount of boost is required at a certain frequency. By providing some boost in those frequencies in the analog domain reduces the amount of boost required in the digital domain resulting in possibly a simpler, shorter filter in the digital domain. Most digital adaptive equalizers

use the LMS algorithm for convergence. The convergence time of these filters is proportional to the filter length [30] and so the use of analog equalization may be advantageous from a complexity as well as performance standpoint. So the first key advantage is that a simpler, faster converging digital filter may result with the use of analog pre-equalization.

As discussed earlier in this chapter, there is quantization noise associated with the process of A/D conversion which can be modelled as an additive white noise source at the input of the A/D. In cases where the equalizer provides magnitude boost at certain frequencies, noise at these frequencies is enhanced. In digital implementation of adaptive filters providing magnitude boost, the filters boost quantization noise of the A/D converter. However in cases where the boost is provided in the analog domain, enhancement of the quantization noise does not occur.

The last advantage to be discussed deals with the reduction of intersymbol interference that may occur as a result of pre-equalization. Intersymbol interference has the effect of reducing and/or enhancing signal amplitudes in sections of the signal in which there are many data transitions. There are instances where the dynamic range or peak amplitudes are changed sufficiently that extra bits are required in the A/D in order to deal with the extra processing required as a result of ISI. Analog pre-equalization may result in the reduction of the number bits required in the A/D converter by reducing the dynamic range of the signal into the converter. This is very significant because of the high speed requirements of many communication systems. Even a reduction of one bit as a result of pre-equalization can be important from a hardware standpoint. An example of this is shown in Figure 4.20. The gray line corresponds to a signal from an actual drive before pre-

equalization. The solid line is the signal that results after convolution with a 3-tap raised cosine equalizer. Note the difference in dynamic range before and after equalization.



**Figure 4.20** Signal before and after convolution with a 3-tap raised-cosine equalizer impulse response of  $-0.35, 1.0, -0.35$ .

#### 4.4.5 Digital Receiver with Adaptive Analog Pre-equalization

The digital receiver with adaptive analog pre-equalization has the advantages of analog pre-equalization presented in the previous section. It has the added advantage that when the equalization provided is sufficient to derive a timing function in the analog domain from samples of its output, the signal need not pass through the long loop of Figure 4.17 in order to feed back to the local oscillator. Removal of this latency results in a much more stable, fast acquisition timing recovery loop that can have a higher tracking bandwidth as well.

A block diagram of this approach is shown in Figure 4.21. To date, no practical continuous-time adaptive equalizers have been presented and so a discrete-time analog adaptive equalizer is assumed (Techniques for such a filter are presented toward the end of Chapter 5). It is rather difficult to realize an adaptive equalizer with more than about 3-taps so an adaptive equalizer may still be required in the digital domain for further equalization prior to sequence detection.

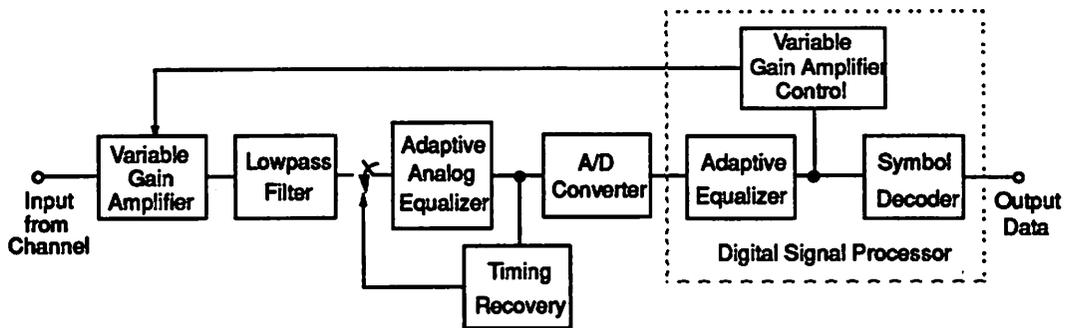
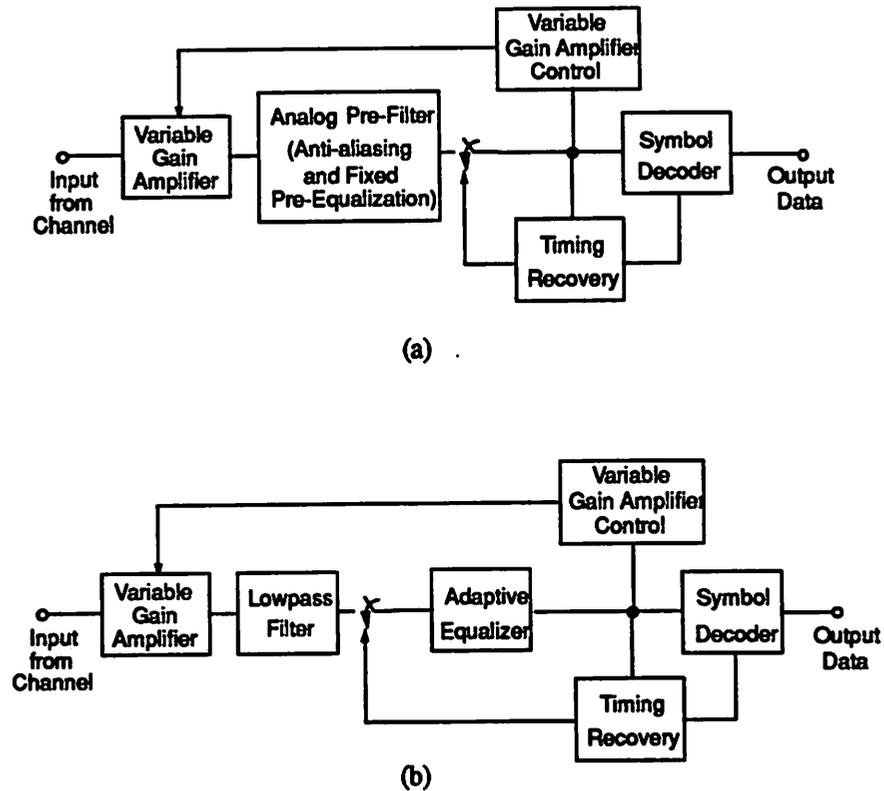


Figure 4.21 Block Diagram of Digital Receiver with Analog Adaptive Equalization.

#### 4.4.6 All Analog Receiver

The all analog receiver performs all the functions including equalization, timing recovery, and signal detection in the analog domain. Block diagrams of two such receivers are shown in Figure 4.22 where the receiver shown in Figure 4.22a uses continuous-time equalization and the approach in Figure 4.22b uses a discrete-time approach. Both approaches can be made programmable for applications where the channel is well known. The filter itself can have a local control circuit which chooses the setting or it can be set externally by a controller. For applications requiring adaptive equalizers, currently, only discrete-time approaches seem viable. The key problem with discrete-time approaches is that it is costly to make equalizers with impulse responses that are sufficiently long.



**Figure 4.22** Block diagram of two all-analog receivers. (a) Continuous-time analog pre-filter. (b) Discrete-time analog adaptive equalizer.

## 4.5 Comparison of Approaches

Each of the approaches described in the previous section have both advantages and limitations and find their rightful place in different applications. In this section we shall compare the different approaches in order to point out some of the key differences and to suggest when each might be more applicable than the others.

The all-digital receiver is best when the channel characteristics vary to the degree that no single compromise analog equalizer exists. The advantages of pre-equalization discussed in this chapter certainly warrants its consideration. There are certainly cases in which the complexity of a

useful analog equalizer prohibits its use and therefore all processing should be done digitally. On the other hand are applications where the channel is well known along with the expected channel variations. In these situations, fully analog approaches become very attractive. Keeping all the signal processing in the analog domain tends to keep latency in the feedback loops low and circumvents the need for an analog-to-digital converter which saves both power and area.

Since digital communication receivers are at some point inherently discrete-time systems and at other points, inherently continuous-time, one of the key questions is at which point in the system should the signal processing move from continuous-time into discrete-time and back if necessary. Some analog continuous-time based approaches require signal levels that may need to be reduced in order to minimize distortion through a block. These approaches have the disadvantage that they require amplification in the back-end which amplifies noise along with the signal. There is the issue of digital versus analog control of the time changing parameters of the system. Analog control parameters take more time to set up, but can be done so with more resolution. Another issue is that of latency in the feedback loops of adaptive and control algorithms that result from parallelism and pipelining in the architectures. In some cases, extra analog processing in the front end can eliminate the need for feedback from the digital domain back into the analog domain as in the timing recovery example presented earlier. As a result more parallelism and pipelining can occur in a larger part of the channel.

In order to make good choices, it is useful to have a good understanding of the channel as well as a knowledge of both analog and digital circuit design. These decisions can be very complicated because of the number of issues that need simultaneous consideration and the fact that the trade-offs can be extremely non-linear.

## **4.6 Proposed Architecture for PR-IV Disk Drive Read Channel**

Circuit techniques were developed as a part of this research in order to implement a read channel in CMOS technology capable of achieving speeds higher than otherwise possible using conventional circuit architectures. In this section, we will examine the architecture of a prototype

integrated circuit designed to demonstrate the concepts developed in this research from a block level view. In particular, we will present the overall architecture of the prototype and quantitatively examine the effects of pre-equalization in DSP-based PR-IV channels with adaptive digital equalization.

The overall architecture selected was that of the digital receiver with analog pre-equalization. A block diagram of the proposed read channel architecture is shown in Figure 4.23. After the

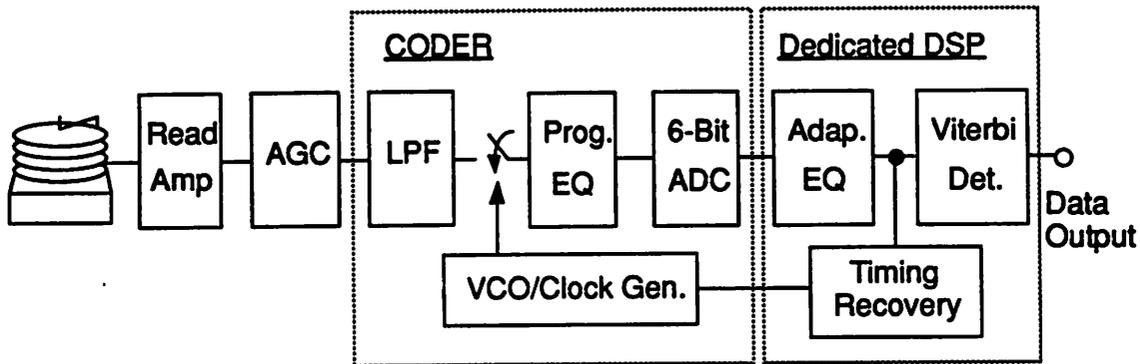


Figure 4.23 Block diagram of the proposed read channel.

AGC function, are the two major blocks; the *CODER* which is the analog-to-digital interface and a dedicated DSP performing the functions required in the digital domain. The coder provides the functions of lowpass filtering, pre-equalization, sampling, and quantization (by the A/D converter). The DSP contains a linear adaptive equalizer, Viterbi sequence detector, and timing function generator which controls the VCO in the coder and defines the sampling phase. Since the disk drive channel is lowpass in nature, high frequency boost is required in this channel to reduce ISI. Some of this high frequency boost is provided by the programmable equalizer [32] preceding the ADC.

#### Block Diagram of Prototype Coder

The key design goal of this research is implementation of the CODER function at a 100 MHz sampling rate in a 1.2  $\mu\text{m}$  CMOS process. A block diagram of the prototype coder is shown in Figure 4.24 and contains a composite lowpass filter comprised of a non-critical continuous-time

filter followed by a 3:1 decimation filter, programmable equalizer, all required clock generation, and a 6-bit ADC. Since preliminary testing and evaluation of the coder will be with sinusoidal inputs, the on-chip phase-lock loop in the clock generation circuitry has been designed to lock onto an external input clock at one-half the output rate so that a timing recovery loop is not needed. The motivation for and implementation of the composite lowpass filter is presented in detail in later chapters along with detailed design descriptions of each of other the blocks in the prototype. Since many detection techniques are possible in the digital domain (PR-IV being only one of them), the coder architecture is somewhat of a generic analog-to-digital interface for disk drive channels employing discrete-time processing. However, it will be described in the context of the block diagram shown in Figure 4.23 which we will assume to be that of a PR-IV channel.

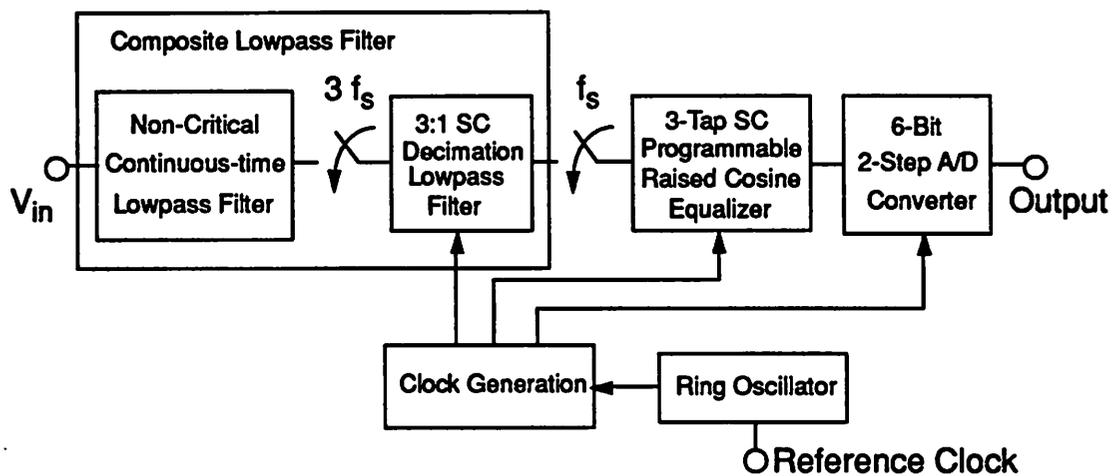


Figure 4.24 Block diagram of the prototype coder.

### Partitioning of the Signal Processing

As described earlier in this chapter, the spectral shaping provided by the equalizer before the A/D converter can have an important impact on overall performance and requirements on the ADC. Let us now examine through simulation the effect of the programmable equalizer on the mean-square error at the adaptive equalizer output for different amount of quantization (ADC resolution) and different quantization in the digital adaptive equalizer structure. We will use the simplified read channel architecture shown in Figure 4.25 containing only a sampler, programmable equalizer, quantizer (A/D converter), and adaptive equalizer following the quantizer. The program-

mable equalizer is constrained to be a 3-tap raised cosine equalizer using the switched-capacitor FIR architectures described in the following chapters.

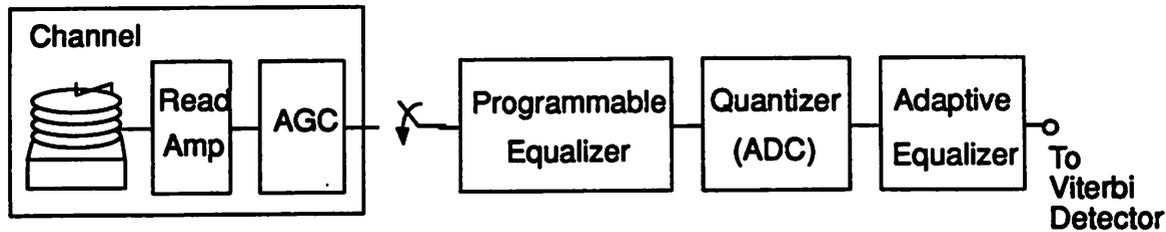


Figure 4.25 Simplified read channel model to study the effect of equalization before the quantizer in PR-IV channels.

The channel model is shown in Figure 4.26 and assumes conventional PR-IV signalling with a Lorentzian pulse shape. Simulations are performed with and without additive noise which is added to the samples from a noiseless channel model. The “user” data is output from the random number generator and no precoding is used resulting in a zero constraint code. Ideal sampling phase is assumed and occurs at the symbol rate.

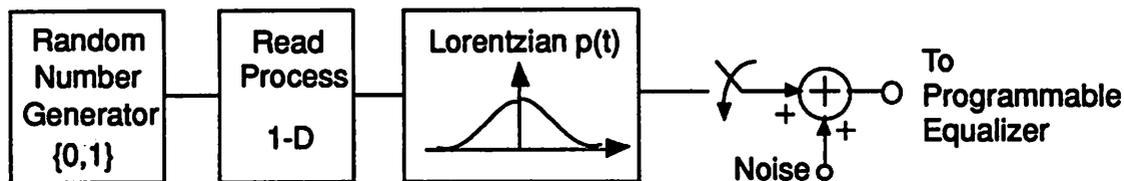


Figure 4.26 Channel model.

The simulations were performed under the following conditions:

- Lorentzian PW50 equal 2.5.
- ADC quantization of 5, 6, 7, and 10 bits.
- 15 Taps in the adaptive equalizer. Tap weights are stored with double precision but before the multiplies, are truncated to a finite number of bits. This models the situation where many bits are stored for the coefficient representation in the update algorithm but only the N-MSBs are used in an N-by-N multiply. Coefficient quantization of 5,6,7, and 10 bits are modelled.
- For the 3-tap raised cosine equalizer impulse response of  $K, +1.0, K$ , the middle tap is kept constant while the outside taps are varied together from  $+0.05$  to  $-0.5$  in steps of  $0.05$ . The case for  $K$  equal  $0.0$  corresponds to the no equalization case. As  $K$  becomes more negative, more boost is provided by the equalizer.

Let us define signal power to be

$$S = \frac{A^2}{2} \tag{4.6}$$

where  $A$  is the amplitude of an isolated transition sampled in the correct phase as shown in Figure 4.27.  $A$  will be less than unity.

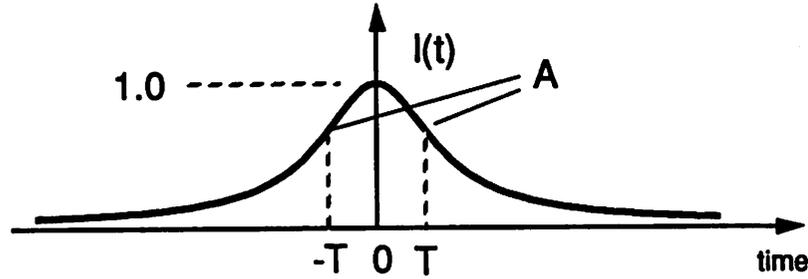


Figure 4.27 Definition of  $A$  in signal power calculation.

The additive noise is zero mean gaussian noise with variance  $\sigma_n^2$ . Therefore, the input SNR is defined as

$$SNR_{input} = 10\log\left(\frac{A^2}{2\sigma_n^2}\right). \tag{4.7}$$

For a desired input SNR at a specified PW50,  $A$  is first calculated plugging  $T$  into the equation for the Lorentzian pulse. Then, the  $\sigma_n^2$  is found to satisfy Eq. 4.7.

Suppose the output of the equalizer was labelled  $y_k$ . It is useful to quantitatively describe the SNR in  $y_k$  for measuring the effects of quantization and equalization at different points in the channel. For this purpose, let us define the signal power as the time averaged signal power

$$\bar{S}_y = \frac{1}{2N} \sum_{k=1}^N (y_k)^2. \tag{4.8}$$

In a similar manner, we define the time averaged noise power to be

$$\bar{S}_n = \frac{1}{2N} \sum_{k=1}^N (\hat{y}_k - y_k)^2 \tag{4.9}$$

where  $\hat{y}_k$  is the ideal equalized noiseless output which includes the effects of residual ISI as well as noise. Therefore, the resultant SNR is defined as

$$SNR = 10 \log \left( \frac{\overline{S_y}}{\overline{S_n}} \right). \quad (4.10)$$

Let us now examine a summary of the key simulation results.

In the simulation results shown in Figure 4.28, the adaptive equalizer output SNR is plotted as a function of the outside coefficient  $K$  in the programmable raised cosine equalizer which precedes the ADC. The ADC quantization was set to 10 bits and simulations were performed to simulate coefficient quantization of the tap weights in the digital equalizer to 5, 6, 7, and 10 bits with and without additive noise. In the case with noise, an SNR of 20 dB was used. Using 10 bits in the ADC allows the study of the effects of quantization in the digital equalizer.

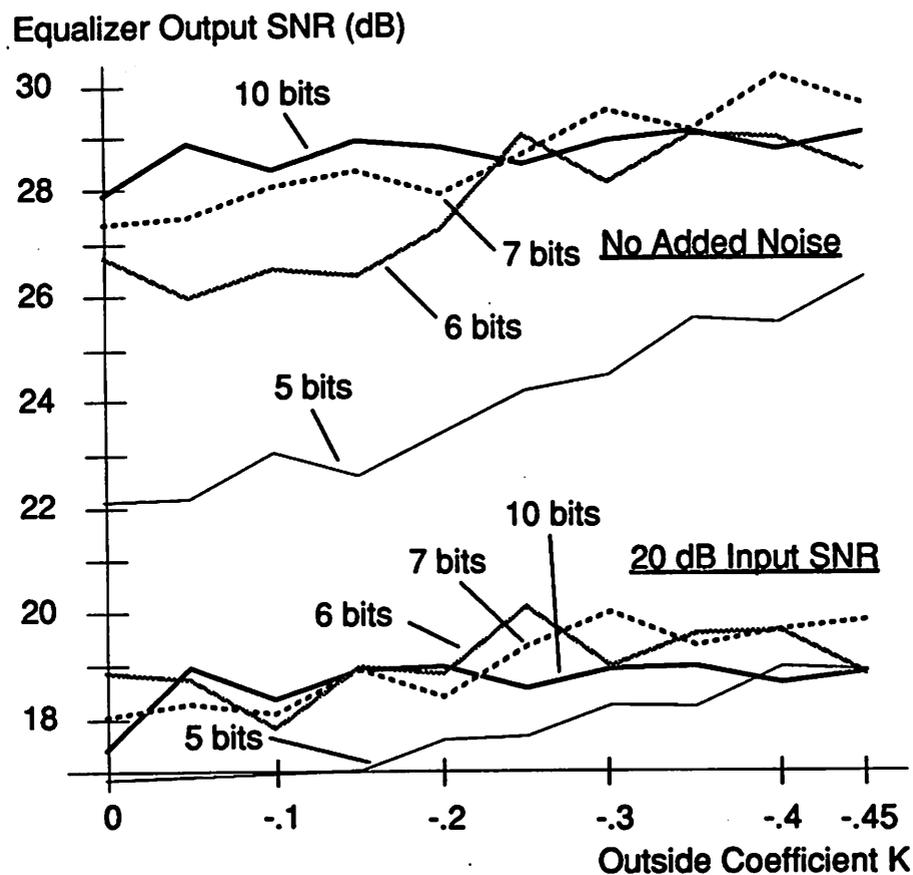
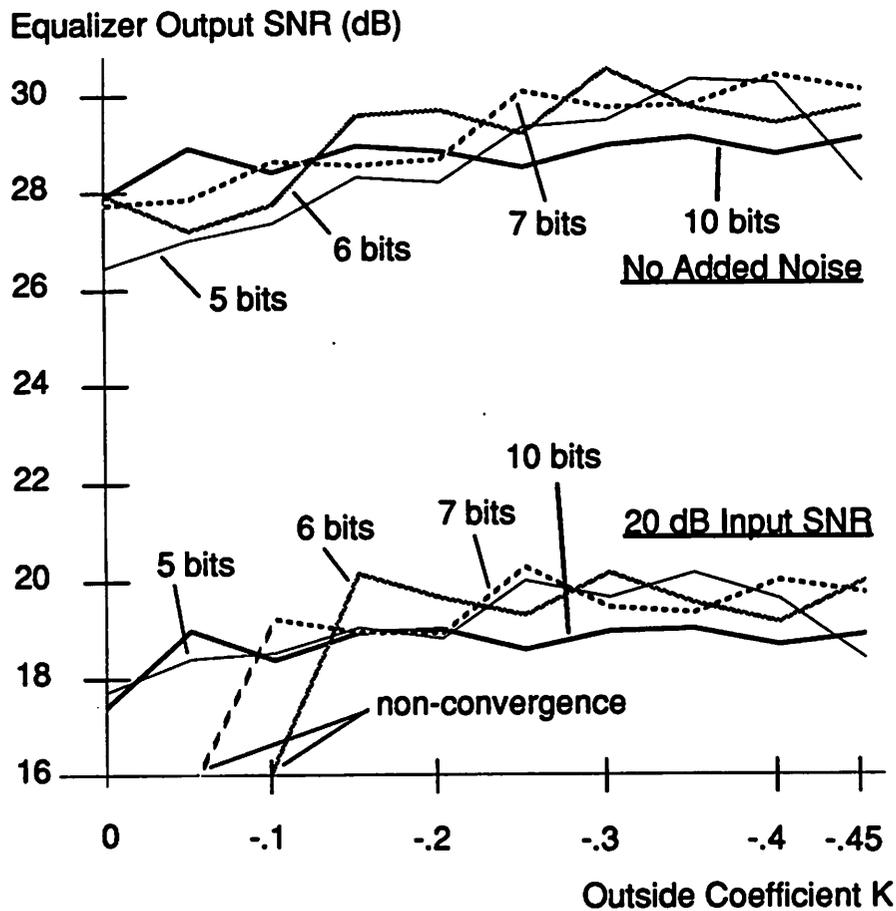


Figure 4.28 Plot of Adaptive Equalizer output SNR as a function of the outside coefficient  $K$  in the programmable raised cosine equalizer for a 10 bit ADC and quantization of the tap weights in the adaptive equalizer (prior to multiplication) of 5, 6, 7, and 10 bits.

The condition for  $K$  equal zero corresponds to the condition in which the analog equalizer provides no filtering. As the value of  $K$  increases in the negative direction, more high frequency boost is provided by the equalizer. For the case with no noise, there is little difference with 7 or more bits of quantization of the filter tap weights. However, there is a large difference between 5 and 6 bits. Notice particularly in the case with 5 bits, there is an increasing SNR advantage as the outside coefficients in the programmable equalizer increase in magnitude. As a result of increased high frequency boost in the analog domain, less boost needs to be provided by the digital adaptive equalizer resulting in less noise enhancement of quantization noise within the filter. In the case of the input SNR of 20 dB, there is almost no difference with 6 or more bits of quantization and a small difference with as few as 5 bits. The noise contributed due to tap weight quantization tends to be below the noise floor for an input SNR of 20 dB with 6 or more bits in the representation of the tap weights prior to the multiplies in the filter.

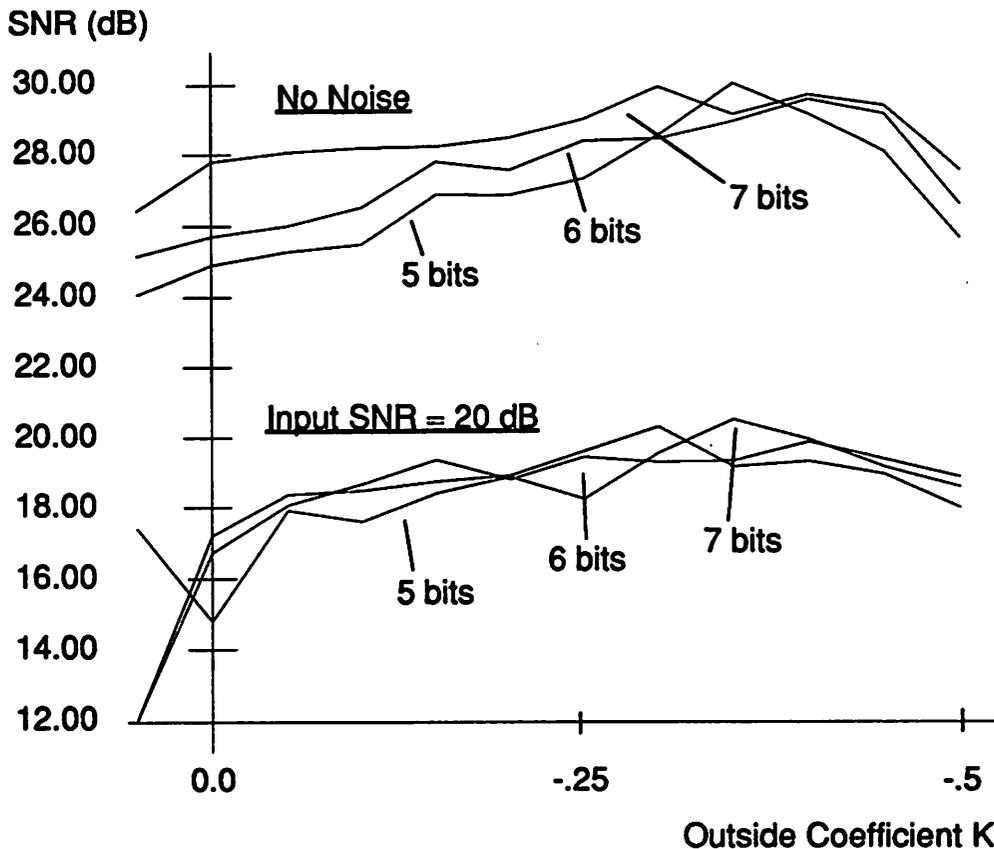
In the simulation results shown in Figure 4.29, the adaptive equalizer output SNR is once again plotted as a function of the outside coefficient  $K$  in the programmable raised cosine equalizer which precedes the ADC. However, this time, quantization of the equalizer tap weights prior to multiplication in the filter is fixed at 10 bits and different values of ADC quantization are used. For the case of no noise, it appears that 5 bits may be sufficient provided that a cosine equalizer preceding the ADC with outside coefficients of  $-0.35$  is present. In the case of an input SNR of 20 dB, it appears that there is very little difference between 5 and 10 bits provided the equalizer is able to converge. There does however appear to be an SNR advantage with the presence of analog equalization as can be seen by the net positive slope in the SNR plot.

It is interesting to note in these two figures that performance of the system as measured by SNR at the equalizer output is more sensitive to tap weight quantization prior to multiplication in the adaptive equalizer than the number of bits used in the ADC. Quantization in the tap weights corresponds to a quantization of the allowable equalizer responses. As fewer bits are used in the tap weight representations, the resulting filters may have responses that are further away from the ideal resulting in a lower achievable SNR.



**Figure 4.29** Plot of Adaptive Equalizer output SNR as a function of the outside coefficient  $K$  in the programmable raised cosine equalizer for ADC quantization of 5, 6, 7, and 10 bits and a quantization of the tap weights of the adaptive equalizer of 10 bits.

Plotted in Figure 4.30 is the output SNR as a function of the programmable tap weight varying from +.05 to -.5 for different values of ADC quantization and tap-weight quantization in the adaptive equalizer. In this plot, the quantization of both change together.



**Figure 4.30** Plot of SNR as a function of the outside coefficient K for the case of no noise and an input SNR of 20 dB with equal quantization in the number of bits in the ADC and the tap weight quantization in the adaptive equalizer.

The key results of the simulations are as follows:

- There is a distinct advantage of using pre-equalization (analog equalization) even in the presence of a digital adaptive equalization of 3dB in the case of the input SNR equal to 20 dB and 5 dB in the case of no noise.
- Performance is more sensitive to the effect of quantization of the filter tap weights than the resolution of the A/D converter, i.e., using a 5-bit ADC and 6-bit multipliers in the adaptive equalizer performs better than the opposite.
- For an input SNR of 20dB, the 5-bit ADC performs as well as the 6-bit ADC.

The simulations suggest that for a PW50 of 2.5 and input SNR of 20dB, a good value for the outside taps of the programmable equalizer is .35.

## 4.7 Summary

In this chapter, we examined some of the important considerations for block level design of digital baseband receivers. We began with an examination of some pertinent basics such as noise shaping through the receiver and issue of latency in feedback loops. Since filtering is one of the key signal processing functions required in receivers, we examined some of the key analog filter approaches used in CMOS implementations. Most texts in digital signal processing contain a section or chapter on hardware implementation and therefore implementation of digital filters was not presented here. With this background, we examined different partitioning of the signal processing between the continuous and discrete-time domains and found that in instances where the channel varies considerably over the operating range, digital solutions are desirable because they are most flexible and robust. And when the channel is well known, analog approaches become attractive because of the absence of the analog-to-digital converter and the shorter latency in feedback loops. We concluded the chapter with a read channel architecture proposed for magnetic disk read channels employing PR-IV signalling. In particular, we examined through simulation the effects of a 3-tap raised cosine equalizer preceding a quantizer (A/D converter) on the signal to-noise ratio at the output of a linear equalizer which follows the quantizer. We found there to be important signal-to-noise ratio advantages due to the reduction in high frequency boost required in the digital domain that boosts quantization noise.

Starting in the next chapter, we will begin presentation of the circuit techniques employed in the prototype. In Chapter 8, we will present a comprehensive circuit design description of the prototype CODER whose architecture was presented and examined in this chapter.

## References

- [1] R. Philpott, R. Kertis, R. Richetta, T. Schmerbeck, and D. Schulte, "A 7 MB/sec (65 MHz) Mixed-Signal Magnetic Recording Channel DSP Using Partial Response Signaling with Maximum Likelihood Detection," *Custom Integrated Circuits Conference 1993*, pp. 10.4.1-10.4.4, May 1993.
- [2] J. Hong, R. Wood, and D. Chan, "An Experimental 180 Mb/sec PRML Magnetic Recording Channel," *IEEE Transactions on Magnetics*, Vol. 27, No. 6, pp. 4532-4537, November 1991.
- [3] R. H. Katz, G. A. Gibson, and D. A. Patterson, "Disk System Architectures for High Performance Computing," *Proceedings of the IEEE*, Vol. 77, No. 2, pp. 1842-1858, December 1989.
- [4] S. Haykin, *Communication Systems: Second Edition*, John Wiley & Sons, 1983.
- [5] P. R. Gray and R. G. Meyer, *Analysis and Design of Analog Integrated Circuits: Second Edition*, John Wiley & Sons, 1984.
- [6] E. A. Lee and D. G. Messerschmitt, *Digital Communication*, Kluwar Academic Publishers, 1988.
- [7] J. Proakis, *Digital Communications, Second Edition*, McGraw-Hill, 1989.
- [8] D. Sallaerts, D. H. Rabaey, R. F. Dierckx, J. Sevenhans, D. R. Haspeslagh, and B. J. De Ceulaer, "A Single-Chip U-Interface Transceiver for ISDN," *IEEE Journal of Solid-State Circuits*, Vol. SC-22, No. 6, pp. 1011-1021, December 1987.
- [9] P. R. Gray, B. A. Wooley, and Robert W. Brodersen, *Analog MOS Integrated Circuits, II*, IEEE Press, 1989.
- [10] R. Gregorian and G. C. Temes, *Analog MOS Integrated Circuits*, John Wiley & Sons, 1986.
- [11] A. B. Williams and F. J. Taylor, *Electronic Filter Design Handbook: LC, Active, and Digital Filters (Second Edition)*, McGraw-Hill, 1988.
- [12] M. E. Van Valkenburg, *Analog Filter Design*, Holt, Rinehart and Winston, 1982.
- [13] G. M. Jacobs, D. J. Allstot, R. W. Brodersen, and P. R. Gray, "Design Techniques for MOS Switched Capacitor Ladder Filters," *IEEE Transactions on Circuits and Systems*, Vol. CAS-25, no. 12, pp. 1014-1021, December 1978.
- [14] E. A. Guillemin, *Synthesis of Passive Networks*, Wiley, 1957.
- [15] Y. P. Tsividis, *Continuous-Time Filters*, IEEE Press, 1992.
- [16] Y. Tsividis, M. Banu, and J. Khoury, "Continuous-Time MOSFET-C Filters in VLSI," *IEEE Journal of Solid-State Circuits*, Vol. SC-21, No. 1, pp. 15-29, February 1986.
- [17] J. Khoury, "Design of a 15-MHz CMOS Continuous-Time Filter with On-Chip Tuning," *IEEE Transactions on Solid-State Circuits*, Vol. 26, No. 12, pp. 1988-1997, December 1991.
- [18] V. Gopinathan, Y. P. Tsividis, K.-S. Tan, and R. K. Hester, "A 5V 7th-order Elliptic Analog Filter for Digital Video Applications," *ISSCC 1990 Digest of Technical Papers*, vol. 33, pp. 208-209, Feb. 1990.

- [19] R. Castello, et. al., "A Very Linear BiCMOS Transconductor for High-Frequency Filtering Applications," Proceedings of the International Symposium on Circuits and Systems, pp. 1364-1367, 1990.
- [20] C. A. Laber and P. R. Gray, "A 20-MHz Sixth-Order BiCMOS Parasitic-Insensitive Continuous-Time Filter and Second-Order Equalizer Optimized for Disk Drive Channels," *IEEE Journal of Solid State Circuits*, Vol. 28, No. 4, pp. 462-470, April 1993.
- [21] H. Khorramabadi and P. R. Gray, "High-Frequency Continuous-Time Filters," *Journal of Solid-State Circuits*, Vol. SC-19, No. 6, pp. 939-947, December 1984.
- [22] V. Gopinathan, Y. P. Tsvividis, K.-S. Tan, and R. K. Hester, "Design Considerations for High-Frequency Continuous-Time Filters and Implementation of an Antialiasing Filter for Digital Video," *Journal of Solid-State Circuits*, Vol. 25, No. 6, pp. 1368-1378, December 1990.
- [23] Y.-T. Wang and A. A. Abidi, "CMOS Active Filter Design at Very High Frequencies," *Journal of Solid-State Circuits*, Vol. 25, No. 6, pp. 1562-1574, December 1990.
- [24] R. W. Brodersen, P. R. Gray, and D. A. Hodges, "MOS Switched-Capacitor Filters," *Proceedings from the IEEE*, Vol. 67, pp. 61075, January 1979.
- [25] K. Martin, "Improved Circuits for the Realization of Switched-Capacitor Filters," *IEEE Transactions on Circuits and Systems*, Vol. CAS-27, No. 4, pp. 237-244, April 1980.
- [26] G. D. Forney Jr., "Maximum-Likelihood Sequence Estimation of Digital Sequences in the Presence of Intersymbol Interference," *IEEE Transactions on Information Theory*, Vol. IT-18, No. 3, pp. 363-378, May 1972.
- [27] F. Gardner, *Phaselock Techniques*, Wiley & Sons, Inc., 1979.
- [28] H. Takatori, T. Suzuki, F. Fujii, and M. Ogawa, "A CMOS Line Equalizer for a Digital Subscriber Loop," *IEEE Journal of Solid-State Circuits*, Vol. SC-19, No. 6., pp. 906-912, December 1984.
- [29] R. W. Wood and R. W. Donaldson, "Decision Feedback Equalization of the DC Null in High-Density Digital Magnetic Recording," *IEEE Transactions on Magnetics*, Vol. MAG-14, No. 4, pp. 218-222, July 1978.
- [30] M. L. Honig and D. G. Messerschmitt, *Adaptive Filters: Structures, Algorithms, and Applications*, Kluwar Academic Publishers, Boston, 1984.
- [31] D. G. Messerschmitt, "Echo Cancellation in Speech and Data Transmission", *IEEE Journal on Selected Areas in Communications*, SAC-2(2), p. 283-297, March 1984.
- [32] T. Kameyama, S. Takanami, and R. Arai, "Improvement of Recording Density by Means of Cosine Equalizer", *IEEE Transactions on Magnetics*, Vol. MAG-12, No. 6, pp. 746-748, November 1976.

## CHAPTER 5

# High Speed Switched-Capacitor Transversal Filters

---

### 5.0 Introduction

Switched-Capacitor (SC) transversal or finite impulse response filter structures have been explored by numerous authors for different applications. They have been used in adaptive filter structures for equalizers [3] and used to perform signal processing in the analog domain [2]. SC FIR decimation filters have been combined with infinite-impulse response (IIR) filters for anti-aliasing applications, easing requirements on the preceding continuous-time filter [4]. SC implementation of polyphase decimation filters has also been proposed [5]. In each of the examples above, the operational amplifiers were required to settle at the output rate or even faster, limiting the signal frequencies with which these filters can operate.

#### ***Important Application for High Speed Filters: Anti-aliasing in Data Communications***

Many emerging applications in data communications such as digital magnetic recording and high speed data transmission over twisted-pair will be operating with symbol rates in excess of 100 MHz. Due to the scaling of feature sizes in CMOS technologies and clever architectures, digital signal processors (DSPs) are able to clock at these rates allowing the application of sophisti-

cated processing techniques to these systems. Such systems require an interface which takes samples of the analog signal from the media and converts it to a digital representation for the DSP. Contained in this interface is a filter which bandlimits the analog signal (limiting the amount of aliasing noise that may result from sampling) and may perform some form of equalization prior to sampling and analog-to-digital conversion. Although the applications are quite varied, there are some filter requirements that are common. In many of these applications, due to the inherent low-pass nature of the channel, most of the required attenuation is located just after the passband edge and only modest amounts of attenuation is required far outside of the passband. Quite often, only a limited dynamic range is required through the filter due to limited input SNR. Delay distortion must normally be kept to a minimum, requiring linear phase or near linear phase structures. And some applications such as digital magnetic recording have variable data rates which dictate the need for filters with scalable frequency characteristics that track the data rate.

The two conventional approaches for fully integrated anti-alias filters in CMOS are switched-capacitor IIR and continuous-time filters. For filter bandwidths above 10MHz, switched-capacitor IIR approaches are not attractive for many reasons. Since they are by nature sampled-data filters, they also require a continuous-time filter to precede it to supply attenuation near the sampling rate. Due to the nature of IIR filters, the amplifiers are required to settle at the sampling rate of the filter. In order for the majority of spectral shaping to occur in the SC filter as opposed to the continuous-time filter preceding it, sampling must occur at a rate higher than the symbol rate with a decimation possible at the filter output. The problem with IIR filters is that the amplifiers must settle at the higher sampling rate which limits the operation of such an approach to symbol rates which are less than one-over the settling time of the amplifiers. In summary, the speed limitation of SC IIR filters is the amplifier settling time.

The majority of CMOS anti-alias filters for bandwidths above 10MHz tend to use continuous-time filtering techniques. The two basic viable continuous-time filtering approaches as described in Chapter 4 are transconductance-C filters and MOSFET-C filters. Transconductance-C filters suffer from parasitic sensitivity since all parasitics on the output of the transconductor add to the capacitance and thus affect the time constant of the transconductance-C element. Another problem is that for linearity considerations, amplitudes in the 100-250 mV range are used through the filter

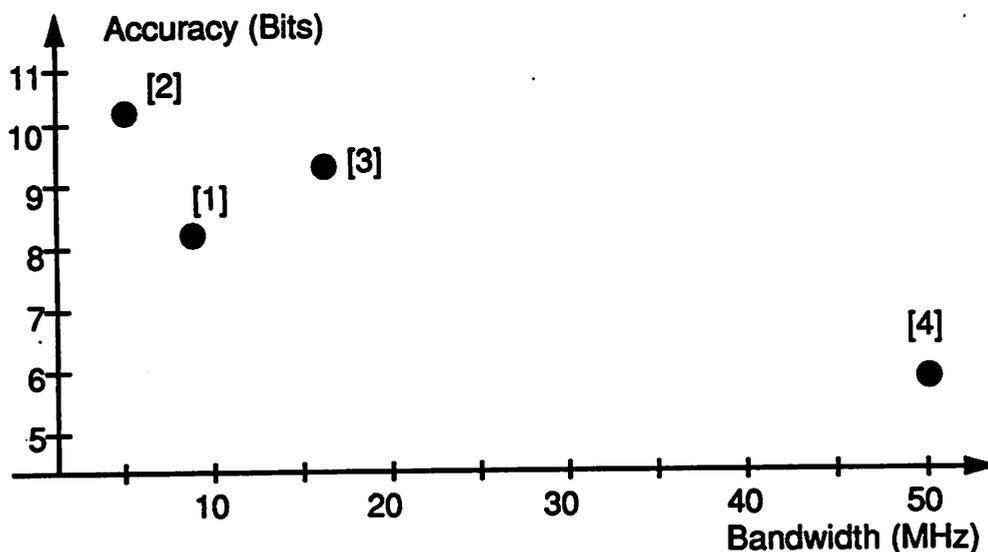
with an appropriate wideband gain stage at the end. Unfortunately, this gain stage amplifies noise as well as signal. Very recently, operational amplifier based approaches have been reported which greatly reduces parasitic sensitivity and allows signal amplitudes in the 1-2 V range [9]. Still, very sophisticated control loops are required to tune both the frequency and the Q of the integrators. Also, BiCMOS technology is required because device  $f_s$  much greater than the signal bandwidth of interest is needed. In summary, continuous-time filters require tuning circuits that correct for the presence of unwanted poles and zeros that occur due to parasitic and second-order effects. As a result, practical filters need technologies with intrinsic bandwidths that are much higher than the desired filter bandwidth. Even with tuning, practical circuits have a finite tuning range due in particular to mismatches that arise in master-slave tuning architectures which ultimately limit the attainable bandwidth.

Transversal or finite impulse response (FIR) filters are uniquely different from their continuous-time filter counterparts because rather than be defined by a pole-zero constellation, they are specified by their discrete time impulse response. Most practical FIR filter implementations tend to be performed in the digital domain where it is straight forward to implement different structures such as those which minimize delay through the filter (latency) or minimize the hardware requirements. Analog FIR filter implementations, just as with other signal processing implementations in the analog domain tend to require much more care and not have the modularity nor inherent robustness of their digital counterparts. There are, however, windows of opportunity where analog processing provides important advantages. And for certain algorithm parameters (i.e., short impulse response filters, low resolution systems, and-subtract and compare (Viterbi), etc...), analog approaches may be viable and even advantageous.

#### Where Does This Approach Fit In?

In Figure 5.1 below, some previously reported filters in CMOS are plotted with the vertical axis of dynamic range in effective number of bits and the horizontal axis is filter bandwidth. Two of the filters are SC IIR filters for IF applications and one is a continuous-time filter for the magnetic disk drive application. Also plotted on the diagram is the expected performance of a prototype filter using the proposed approach presented later in this chapter and described in detail in Chapter 8. Note that using SC IIR and continuous-time filtering approaches, dynamic range

greater than the 8 bit level is easily achieved. However, in applications such as the magnetic disk drive read channel which is an important application needing high speed filters, the dynamic range requirements are around the 5-6 bit level since the off-channel SNR is between 16-25 dB. The proposed approach trades resolution for speed. Note in the diagram that although the prototype target bandwidth of 50 MHz is much higher than the bandwidth of the previously reported filters, the target resolution is only 6 bits.



	Name	Filter Type	Application	Technology	Year
[1]	Martins and Franca [4]	SC Fir/IIR LPF	Video	2.4 $\mu\text{m}$	'89
[2]	Gopinathan, et al. [8]	Continuous-time LPF	Video	1.0 $\mu\text{m}$	'90
[3]	J. Khoury [16]	Continuous-time LPF	Magnetic Disk	.9 $\mu\text{m}$	'91
[4]	Uehara and Gray	SC Decimation FIR LPF	Magnetic Disk	1.2 $\mu\text{m}$	'93

Figure 5.1 Comparison of the new approach with previously reported high speed CMOS filters.

In this chapter, we will describe new architectures for the implementation of high speed analog transversal filters using switched-capacitor techniques in CMOS technology. The architectures

employ the use of parallelism in the processing path in order to implement filters with output sampling rates that can be much larger than the inverse of the settling time of the operational amplifiers. This is an important objective because the key speed bottleneck of previously reported switched-capacitor filters is the amplifier settling time. We will present architectures that can be used for the implementation of both short and long decimation filters. In Chapter 8, we will examine a prototype IC employing the proposed architectures for the implementation of a decimation lowpass filter and a pre-conditioning equalizer for application in magnetic storage read channel front-ends.

This chapter will begin with an examination of the transversal filter structure. Switched capacitor implementation of FIR filters have been reported and we shall examine some of the previous work in this area. Following this, we will then present the new architecture for switched-capacitor circuit realization of both single- and multi-rate filters. Architectures for both short and long impulse responses along with decimation filters will be presented. Following this will be an in-depth analysis of performance limitations of the proposed architectures.

## **5.1 Transversal Filters**

In this section, we will review the basic structure of a transversal filter and consider some of the advantages of these structures. Following this, we will examine previous work that has been reported in the application of switched-capacitor circuit techniques to the implementation of transversal structures.

### **5.1.1 Basic Transversal Filter Structure**

The transversal or finite impulse response filter is a sampled-data filter characterized simply by its impulse response. The input signal is a discrete-time sequence which may be analog or digital and the output is another discrete-time sequence which is the convolution of the input sequence and the filter impulse response. Any particular output is the inner product of a vector made up of

the most recent  $N$  inputs in succession with the transpose of a vector made up of the tap weights of the filter. The relationship between the input samples  $x[n]$  and output samples  $y[n]$  is given by

$$y[n] = x[n]C_0 + x[n-1]C_1 + \dots + x[n-N+1]C_{N-1} \quad (5.1)$$

A block diagram of an FIR filter is shown in Figure 5.2 and includes delay elements, multipliers, and an  $N$ -input summing block. The output is equal to an *inner product* of a vector of input samples including the current through the input at time  $(N-1)$  for an  $N$ -tap filter and a coefficient vector made up of the filter tap weights. As can be seen in the diagram, there is no feedback in an FIR filter and thus it is an all-zero filter which means that the response is shaped by placement of transmission zeroes in the frequency domain.

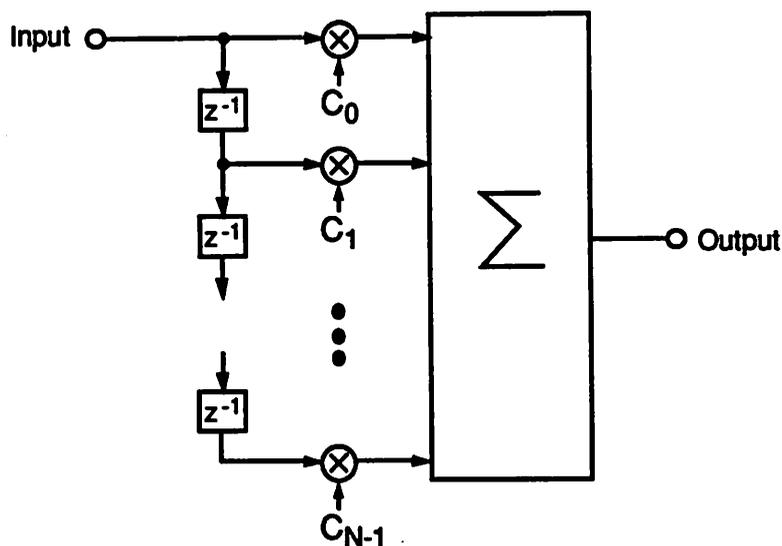


Figure 5.2  $N$ -tap FIR Filter Block Diagram

There are many techniques for designing filters FIR filters which may be found in the rich set of literature on the subject [17]. There are different ways of achieving desired frequency domain responses as well as architectures which exploit characteristics of classes of impulse responses. We shall limit our discussion to a presentation of some of the more important advantages and limitations of FIR filters in order to understand the trade-offs between the implementation of continuous-time filters and analog discrete-time FIR filters.

### 5.1.1.1 Advantages of FIR Filters

One of the key advantages of FIR filters [17,18] is the ease with which one can design linear phase filters. For a filter with an odd number of taps, the impulse response needs to have even symmetry about the center tap in order to have linear phase. For filters with  $N$  even, the filter needs odd symmetry i.e., the taps on the right half must be the negative of the taps on the left. Another advantage of FIR filters are that they are robust to component variations. Yet another important advantage is that they facilitate the implementation of adaptive filter structures [15].

Let us compare continuous-time filters with FIR filters from an analog IC implementation standpoint in order to get some intuition about how and when discrete-time processing may be a viable alternative.

From the signal processing standpoint, a significant difference between continuous-time filters and discrete-time FIR filters is simply that the time domain response of continuous-time filters is continuous whereas discrete-time filters need only to calculate outputs at discrete-time intervals. Continuous-time filters need sufficient integrity throughout the entire bandwidth of interest. Unwanted poles and zeroes directly affect the frequency response, and the high frequency operation of these filters is limited by the ability to cancel and remove the effects of these non-idealities (see Chapter 4). On the other hand, discrete-time FIR filters require samples to be taken in accurate time intervals (needs a good time base), but beyond this, the required operations are simply multiplication and addition. They are inherently robust to the effects of parasitic poles and zeroes which affect continuous-time filters. The key requirement is that amplifiers used in the addition operation are given sufficient time to settle and that the multiplications are performed with sufficient accuracy.

We shall see in this chapter that from a practical implementation standpoint, high speed discrete-time filters are limited to impulse responses that are relatively short (3-7 output periods); and therefore, to the implementation of filters that are relatively simple. The primary reason for this is that when analog parallel processing is employed, noise due to voltage offsets and charge injection and clock feedthrough from switches get added in every stage. Therefore, the number of stages must be limited, limiting the complexity and length of the filters. Since filters with as few as 3 taps

can provide useful amounts of signal processing, discrete-time analog FIR filters have windows of opportunity where they are a viable alternative in high frequency filtering applications.

### 5.1.1.2 Example: Raised Cosine Filter

The cosine filter or raised cosine filter gets its name from the fact that its frequency response looks like a cosine function. Since it is a sampled-data filter, its frequency response is periodic. It is particularly important in analog FIR filters because although there are only three taps in the impulse response, both high-pass and lowpass frequency response characteristics are attainable with these filters.

The tap weights of a cosine equalizer is simply  $K, 1.0, K$  where  $K$  can be a positive or negative coefficient. Since there is an odd number of taps and there is even symmetry about the center tap, the cosine filter has linear phase. In the frequency domain, the response to this filter is given by

$$H(f) = 1 + 2K \cos(2\pi ft) \quad (5.2)$$

This expression is plotted in Figure 5.3 for different values of  $K$ . For positive  $K$ , the response is lowpass in nature. When  $K$  is negative, a high-pass response results.

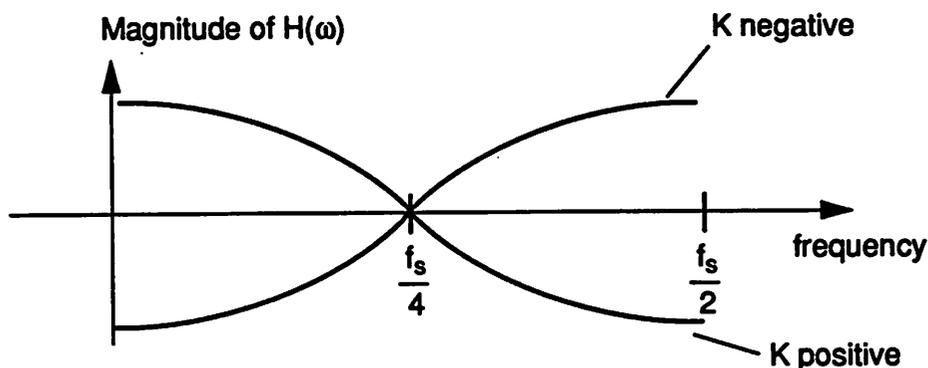


Figure 5.3 Frequency response of a cosine equalizer with different values of  $K$ .

There is an intuitive relationship between the impulse and frequency responses. As we know from signals and systems, the Fourier transform of a cosine waveform is two impulses in the frequency domain at positive and negative values of the cosine frequency. Also, an impulse in the

time domain has all frequency components present and thus has a Fourier transform which is a constant over all frequencies. Therefore, we expect an impulse response of the form  $K, 1.0, K$  to have a frequency response which is a cosine in the frequency domain with a d.c. offset (equal to zero in the example) which is what we see in the figure.

We shall find both lowpass and high-pass cosine filters implemented in the prototype described in Chapter 9. The lowpass response is used in a decimation anti-aliasing filter and the high-pass response is used in a symbol-rate equalizer.

## 5.1.2 Previous Work in SC Transversal Filters

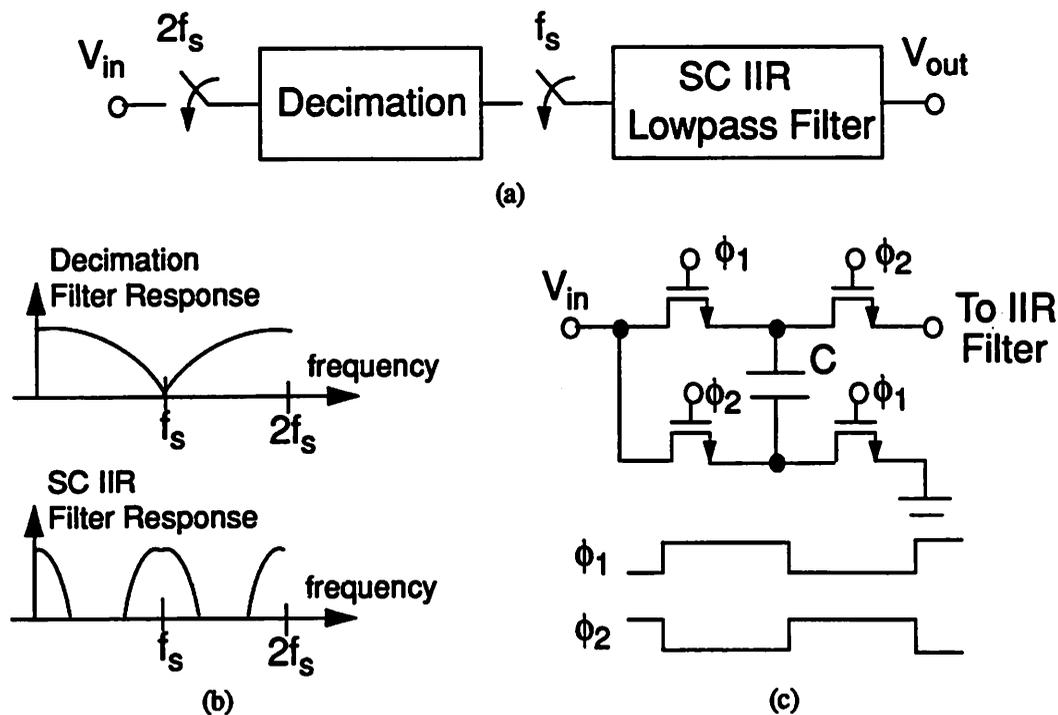
Let us now examine some previous work in SC transversal filters with focus on the different approaches that have been used to implement the delay elements, tap weights, and accumulator. We will see that the speed performance of all previously proposed approaches were limited by amplifier settling time.

### 5.1.2.1 Double Sampling Front-End

A technique proposed in [1] that was very important in telephone codecs was the double sampling front-end filter. In this filter, a response of

$$H(z) = 1 + z^{-1/2} \quad (5.3)$$

is achieved which is similar to the 1+D obtained in Partial Response coding (see Chapter 2). This provides a lowpass response with a zero at one-half the sampling rate. This simple transfer function was placed in front of lowpass SC IIR filters where the sampling rate was on the order of 32 times the lowpass filter bandwidth. This greatly reduced requirements on the front-end anti-aliasing filter. A schematic of this front-end is shown in block diagram form in Figure 5.4a with the associated frequency response of both blocks in Figure 5.4b, and a circuit implementation in Figure 5.4c. Referring to Figure 5.4c, we see that on the falling edge of  $\phi_1$ , the input is sampled and at the end of  $\phi_2$ , the output of the following IIR filter will appear to be preceded by the transfer function described by Eq. 5.3. This simple circuit provides a very useful function without affecting the amplifier settling time. However, it is limited to applications where the signal frequency of interest is much less than the output rate of the IIR filter which is ultimately limited by amplifier settling.



**Figure 5.4** Conceptual illustration of the double sampling front-end filter. (a) Block diagram. (b) Frequency domain characteristics. (c) Circuit implementation.,

### 5.1.2.2 FIR-IIR Structures

An approach folding FIR decimation filter structures into IIR filter architectures was proposed in [4] as a means to allow the operating rate of the amplifiers to be below that of a high sampling rate front-end. This is conceptually similar to the approach previously described. However, instead of implementing a simple response as given by Eq. 5.3, a 3 tap FIR decimation was used followed by an SC IIR filter. This approach realized a 40.5 MHz sampling rate with a decimation down to 13.5 MHz in a 2.4  $\mu\text{m}$  CMOS process. The application was for video anti-aliasing applications. However, as in the previously described approach, the speed limitation is again the amplifier settling time in the IIR filter.

5.1.2.3 Polyphase Filter Structures

Polyphase filter structures were first proposed for implementation of digital decimation filters in order to reduce the speed requirements on the multipliers. This structure was proposed for implementation of switched-capacitor decimation filters in [5]. In the polyphase filter structure, the filter is decomposed into  $M$  sub-filters which operate at the output rate, thus allowing the multipliers to operate at  $f_s$  instead of the input rate  $Mf_s$ . This is an important advantage in digital filter implementation. In a switched-capacitor implementation, the architecture is an efficient approach using the minimum number of capacitors (one capacitor per filter tap) and a single opamp.

The basic structure of the polyphase filter is shown in Figure 5.5a where  $M$  parallel filters, each with a different transfer function is shown. The commutator in the front-end operates at  $Mf_s$ . The filter is simply decomposed with the first tap going to the first filter, second tap to the second filter, up onto the  $M$ th tap which goes to filter  $M$ . Then tap  $M+1$  goes to the first filter,  $M+2$  to the second, and so forth until all the taps are distributed. An example with  $M$  equal to 3 and a total of 9 taps is shown in Figure 5.5b.

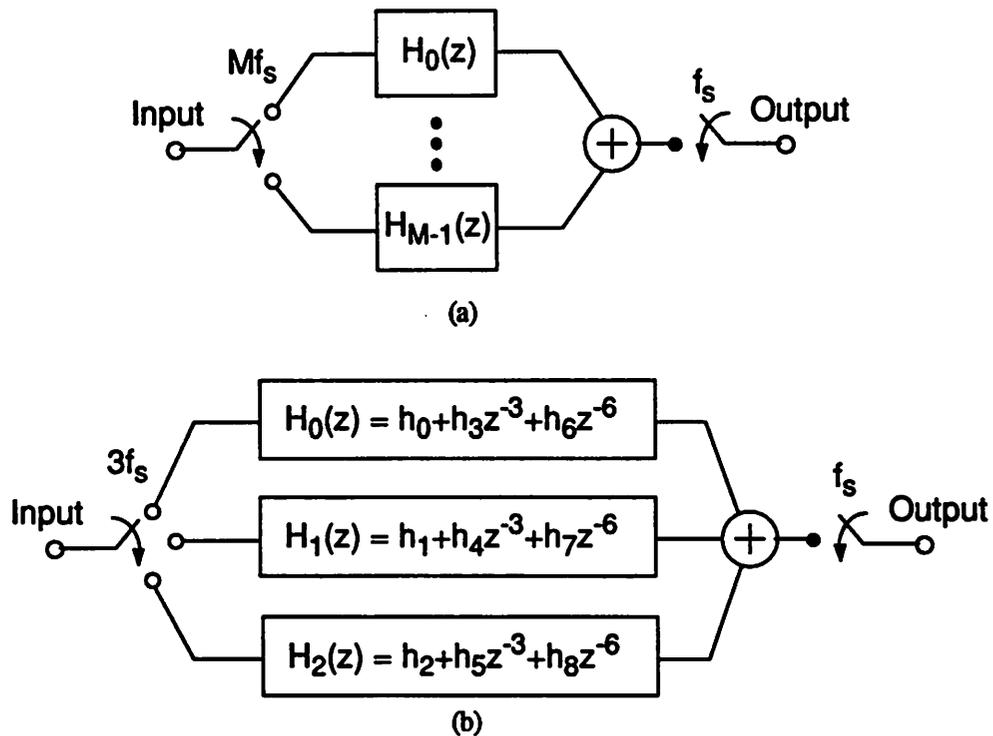


Figure 5.5 (a) General block diagram for a polyphase structure. (b) Filter example with  $M=3$  and 9 total taps.

In the SC implementation proposed in [5], the summer implemented with an SC integrator that is initialized or zeroed at the output rate in order to realize a zero-order hold. The tap weights are realized by ratioed capacitors and  $M$  capacitors are connected to the amplifier at the input rate. Therefore, the opamp is required to settle at the input rate. The advantage of this approach is that the minimum number of capacitors may be used. The speed performance is clearly limited by the amplifier settling time.

#### 5.1.2.4 Rotating Switch-Based FIR Structures

A transversal filter structure to generate the real parts of a running discrete-time Fourier transform was proposed in [2]. A total of 32 input samples are used each time period to calculate eight outputs given by

$$X_k(n) = \frac{1}{8} \sum_{i=0}^{31} x(n-1) \cos\left(\frac{2\pi ik}{32}\right) \quad (5.4)$$

for  $k = 2, 4, 6, \dots, 16$  where  $x(n)$  is the input sequence.

The basic overall structure for the processor is shown in Figure 5.6. One of the 32 sample-and-hold circuits takes a new sample every period and therefore each sample-and-hold is updated every 32 periods. The sample-and-hold circuits are comprised of sampling capacitors followed by voltage followers which act as buffers. These buffers drive a 32 x 32 rotating switch which multiplexes all 32 inputs to the 8 scaling/summing amplifiers which follow. The scaling/summing amplifiers have ratioed capacitors which scale the 32 inputs by the respective coefficient weights given by Eq. 5.4 prior to summing these scaled inputs with an opamp. All 8 outputs  $X_0(n)$  through  $X_7(n)$  appear at the end of every clock period.

In summary, during each clock cycle one of the sample-and-hold circuits takes in a new input while the rotating switch reconfigures itself to appropriately pass the current 32 sampled voltages

to the scaling/summing amplifiers. Therefore, the 8 amplifiers settle during the same time each period. Once again, high speed operation is limited by the amplifier settling time.

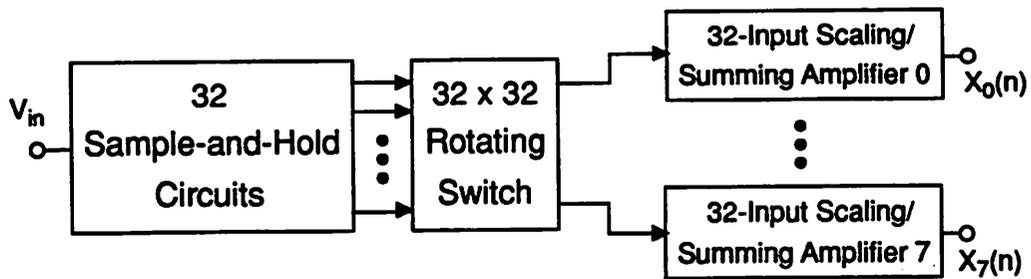


Figure 5.6 Block diagram of the rotating switch-based FIR filter structure.

### 5.1.2.5 Analog Delay Line-Based FIR Structures

An architecture for implementation of a fixed FIR equalizer for discrete-time magnetic read channels using a switched-capacitor circuit approach was proposed in [6]. A block diagram of this equalizer approach is shown in Figure 5.7 for the 7-tap equalizer. Simple unity gain sample-and-hold amplifiers (realizing the  $T/2$  delays) are clocked at twice the output rate and connected in a chain making up an analog delay line. Two interleaved delay lines operating with one output period of offset results in the settling of alternate delay lines each output period. The outputs are tapped off and sampled on two identical capacitor arrays which realize the tap weights of the filter. The sampled voltages across these arrays are passed to two summing trees which generate alternate equalized outputs.

Once again, the speed limitation of this approach is amplifier settling time. The amplifiers are required to settle at twice the output rate since they realize one-half period delays.

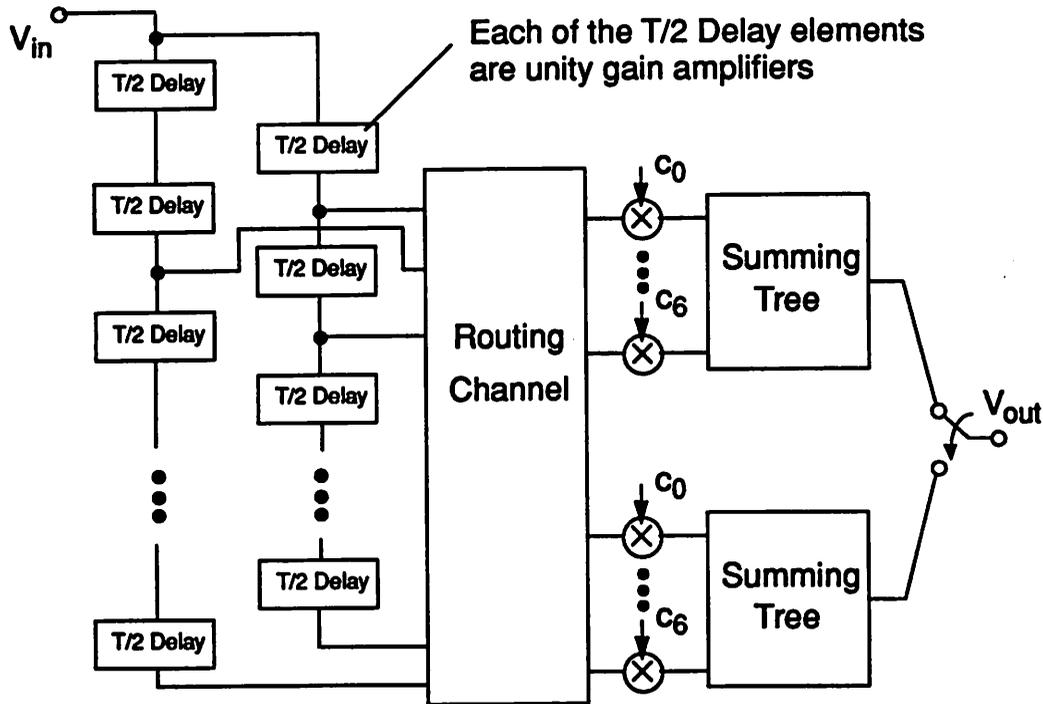


Figure 5.7 Block diagram of the 7-tap FIR filter for equalization in discrete-time magnetic disk read channels.

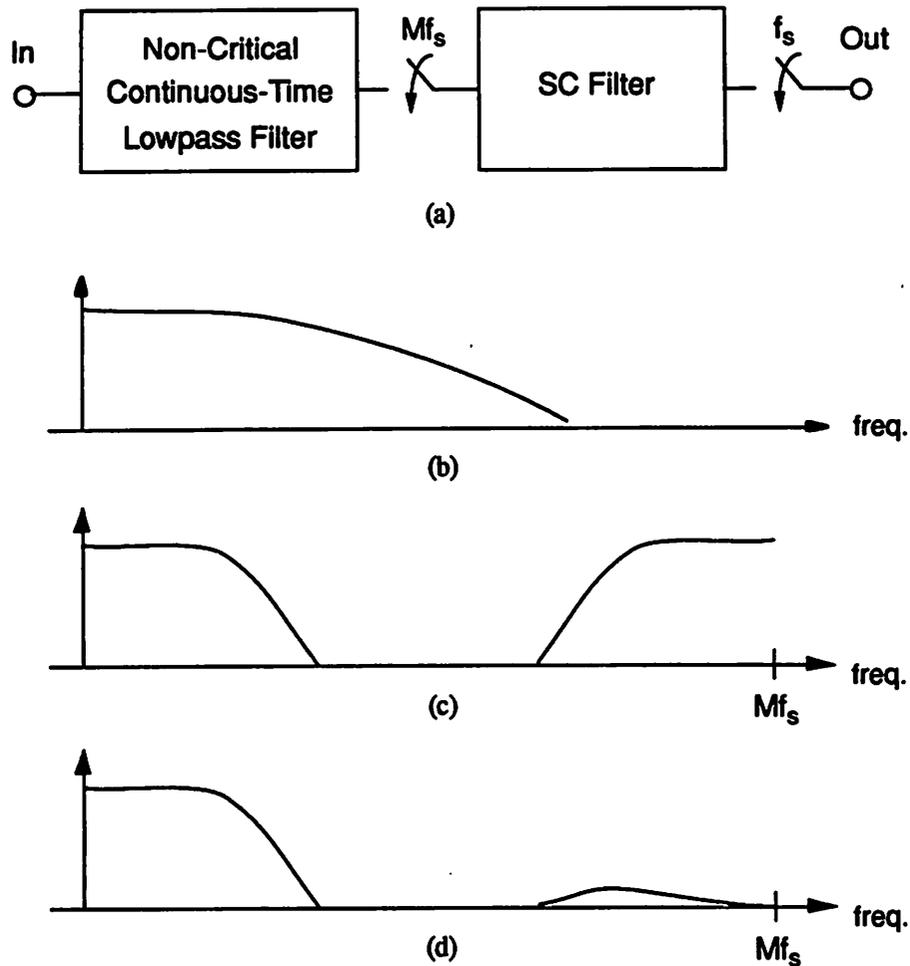
## 5.2 High Speed SC FIR Architectures

In this section, we shall present a new architecture suitable the implementation of high speed decimation and single-rate FIR filters. We shall develop the architecture for the implementation of decimation filters and later simply it for single-rate filter applications. The decimation filter is important because lowpass anti-alias filtering is one context in which these filters may have a distinct speed advantage over alternative filtering approaches. Following this will be an examination of an approach for the implementation of decimation filters with longer impulse responses.

### 5.2.1 SC Decimation Filter Block Diagram

The new switched-capacitor FIR filter approach [20] takes advantage of the fact that FIR filters have no inherent feedback within the filter structure and thus, parallelism and pipelining may be employed in their implementation. With the use of parallelism, amplifiers may be allowed multiple output periods for settling, breaking the key bandwidth limitation of SC IIR filters.

A block diagram of the proposed approach for the implementation of high speed anti-aliasing filters is shown in Figure 5.8. As is evident in the figure, as with all discrete-time filters, the frequency response of the SC filter is periodic in the sampling rate. However, this reduces requirements on the preceding continuous-time filter which only needs to provide adequate noise suppression near the sampling frequency. The composite frequency response of the two filters in cascade is shown in Figure 5.8d. The attenuation near  $Mf_s$  is supplied by the continuous-time filter.



**Figure 5.8** (a) Block diagram of the composite filter. (b) Frequency response of the continuous-time filter. (c) Frequency response of the SC FIR filter. (d) Frequency response of the composite filter.

There are a couple of key advantages to this architecture. First, the requirements on the continuous-time anti-aliasing filter are relaxed due to the high sampling rate of the decimation filter. This filter may be implemented with simple low  $Q$  second-order filter sections in cascade. Second, the bulk of the noise shaping in the filter passband and above is provided by the SC filter. Thus, the frequency response is defined by capacitor ratios and the response automatically scales with clock frequency.

### 5.2.2 SC Decimation Filter Architecture

The circuit architecture for the SC decimation filter will now be presented. We shall begin with a conceptual presentation of the approach and follow with some circuit implementation issues. In Chapters 6 and 8, circuits used in a prototype demonstrating the viability of the approach will be presented along with experimental results in Chapter 8.

The time domain output of an FIR filter is the convolution of the input sequence with the filter impulse response. Each individual output is equal to the inner product of a vector of the last  $N$  inputs with a tap weight vector. For input vector  $x$  and coefficient vector  $h$ , the output  $y[n]$  at time  $n$  can be written as

$$y[n] = [x[n] \ x[n-1] \ \dots \ x[n-N+1]] \cdot \begin{bmatrix} h_0 \\ h_1 \\ \vdots \\ h_{N-1} \end{bmatrix} \quad (5.5)$$

In the new approach, each output is generated independently and therefore the inner product interpretation will prove useful.

Consider the capacitor array shown in Figure 5.9. The input voltage is applied across the entire array which has one side of each capacitor tied to ground and the other connected to the input voltage through a switch. Each of the switches open in succession, sampling the instantaneous voltage across the array at each particular sampling instant. At the end of  $N$ -samples, a total charge  $Q_{total}$  exists across the array which can be described by

$$Q_{total} = \sum_{n=0}^{N-1} V_{in}(t_n) \cdot C_n \quad (5.6)$$

In words, this charge is equal to the sum of all the products of each of the individual capacitors and the voltage sampled across the capacitor at the instant the corresponding switch turned off. By weighting the capacitors as an impulse response, the total charge on the capacitor array after taking  $N$  samples is equal to the inner product between a vector made up of the  $N$ -input samples and the  $N$  capacitors.

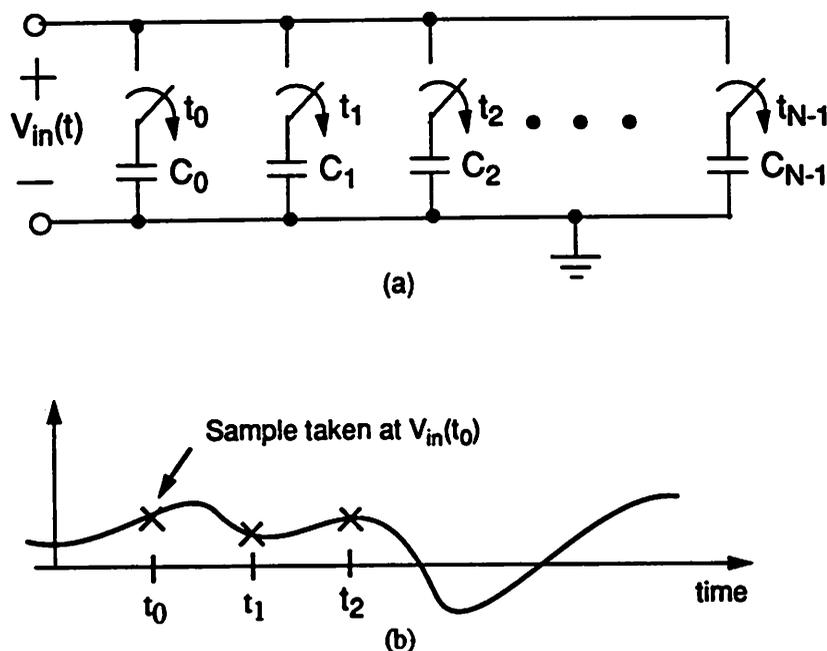
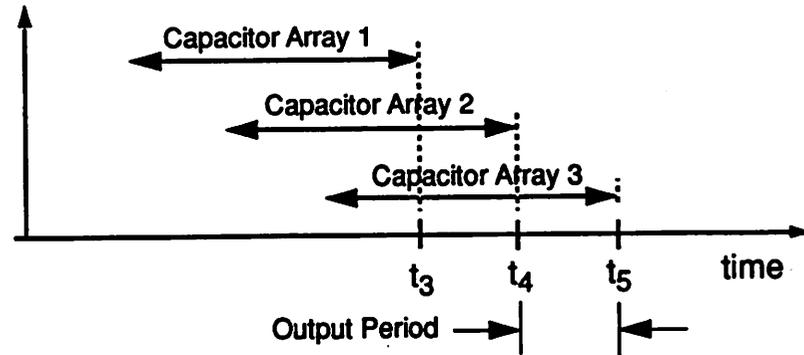


Figure 5.9 Capacitor array sampling input voltage. (a) Array with capacitors weighted as impulse response. (b) Example indicating sampling instants.

In order to generate an output at the desired rate, a number of these capacitor arrays must operate in parallel with a phase shift between them as shown conceptually in Figure 5.10. The minimum number of parallel arrays required and the phase shift required depends on the length of the filter impulse response and the decimation factor (more on this in a later section). Each two-sided arrow represents the period over which a capacitor array samples the input. In a decimation filter, the input sampling rate is higher than the output sampling rate. The output rate of the filter shown in the figure is the rate at which the total charge on each array is sampled. Whereas, the input rate is the rate at which the sampling switches open in each of the capacitor arrays shown in Figure 5.9. By simply taking more than one input sample per output period, a decimation filter is realized

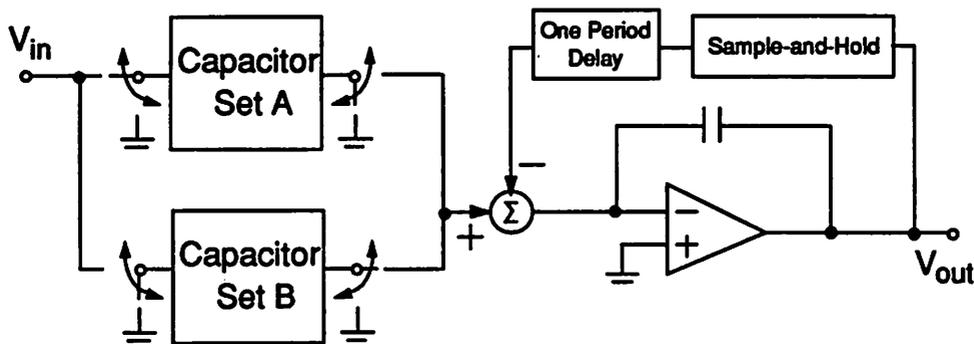


**Figure 5.10** Conceptual timing diagram of parallel arrays. In the case of a decimation filter, multiple inputs are taken each output period.

At the end of each period, one of the capacitor arrays has taken its last sample and is ready to *dump* its accumulated charge to a circuit in order to convert the charge  $Q_{total}$  to a proportional voltage. In a decimation filter, more than one sample is taken every output period by each capacitor array in the sampling mode. However, the charge to voltage conversion need occur at a rate of only once per period.

Consider the example shown in Figure 5.10 where 3 capacitor arrays are used for an impulse response length of three periods. At time instants  $t_3$ ,  $t_4$ , and  $t_5$  the output of each respective array is available for conversion. However, more arrays are needed in order to have outputs at times after  $t_4$ . One solution is to have two interleaved arrays within each parallel channel, operating such that while one array is sampling, the other is in the dump or conversion mode. A conceptual schematic of a circuit which can perform this function in one of the parallel channels is shown in Figure 5.11. Note the two capacitor arrays. The switch phases are such that while Capacitor Set A is in the sampling mode, Capacitor Set B is in the dump mode. And in the next cycle, the arrays switch modes. During the dump mode, the capacitor arrays are connected to a switched-capacitor integrator (opamp with a capacitor in negative feedback) which convert the input charge to a voltage. A key problem with using the integrator in this sample-and-hold like application is that it must be reset after every conversion or it will hold charge across the feedback capacitor from conversion to conversion and thus act like an integrator. The time needed for this reset reduces the time allowed

for operational amplifier to settle. One solution is to use a switched-capacitor circuit technique where the current output voltage is sampled on a capacitor equal in value to the feedback capacitor. In the following cycle, this capacitor subtracts a voltage equal to the previous output which removes the integrating effect or cycle-to-cycle memory of the feedback capacitor. The net effect is that the circuit of Figure 5.11 has a zero-order hold output and no reset period is required for the amplifier. We shall examine a practical circuit to implement this function in Chapter 6.



**Figure 5.11** Schematic of interleaved capacitor sets sharing one amplifier and capacitor zeroing circuit.

Let us refer to the circuit shown in Figure 5.11 as a *switched-capacitor inner product generator* since its output is the inner product between a vector of  $N$  input samples and a tap weight vector made up of ratioed capacitors. Negative coefficients can be realized using switched capacitor techniques described in Chapter 6. Suppose Capacitor Sets A and B are identical and define a  $1 \times N$  vector  $C$  having capacitor values  $[C_0, C_1, \dots, C_{N-1}]$ . Now define a  $1 \times N$  vector  $V_{in}$  having input voltage values  $[V_{in}[n], V_{in}[n-1], \dots, V_{in}[n-N+1]]$ . The output voltage  $V[n+m]$  where  $m$  is the number of input periods given the amplifier for settling can be written as  $V_{in} * C * 1/C_{fb}$ .

A conceptual diagram of parallel arrays coupled with interleaving within each array to achieve the desired throughput is shown in Figure 5.12. The output is taken from consecutive channels by a commutator which takes the output first from Channel 1, then Channel 2 then Channel 3, then again from Channel 1 and so forth. Within each array are two interleaved capacitor arrays which alternate between taking samples and dumping to the operational amplifier. A key advantage of

this approach is that the amplifiers are allowed a time equal to the total sampling period of a capacitor array to settle which may be significantly longer than the effective output period of the filter. And thus, the output rate of this filter is not limited by amplifier settling as is the case with other SC filter architectures.

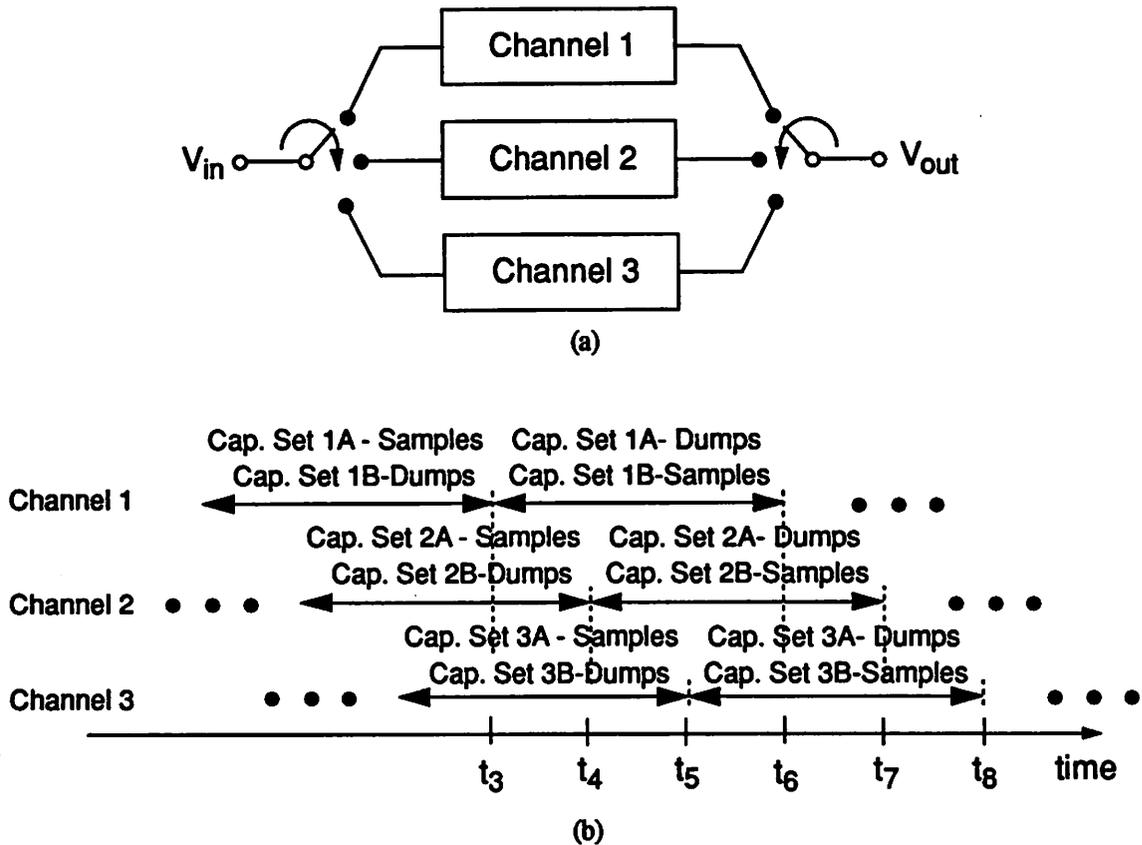
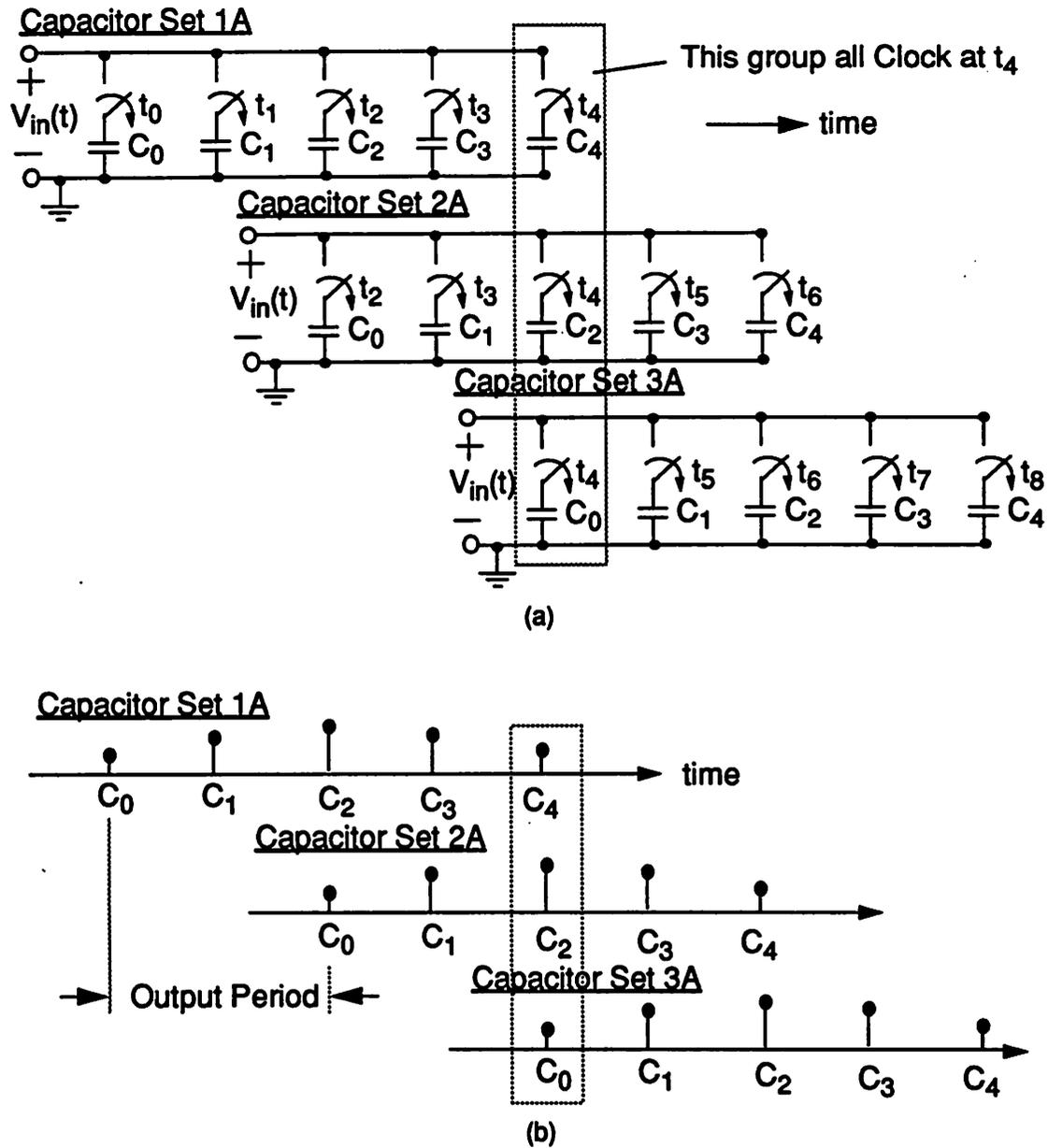


Figure 5.12 Conceptual schematic showing (a) parallel channels and (b) operation of the timing.

A simple example illustrating the operation of a 5-tap 2:1 decimation filter is shown in Figure 5.13a. Only Capacitor Sets 1A-3A are shown. The effective time domain impulse response is shown in Figure 5.13 where each impulse represents the relative magnitude of each tap weight. Note the corresponding distribution over time. This conceptual notation of the parallel impulse responses will be used again in Chapter 8. This illustration is meant to give a clearer picture of the operation of the parallel capacitor arrays (in Figure 5.13a) and to introduce the simplified description shown in Figure 5.13b.



**Figure 5.13** Diagrams illustrating operation of capacitor arrays in a 5-tap 2:1 decimation filter. (a) Parallel capacitor arrays with a phase offset. Sampling capacitors stacked vertically sample at the same instant. (b) Conceptual sampling and multiplication with relative tap weights.

Let us now summarize the key advantages of this approach. The passive sampling capacitor arrays easily have a sampling bandwidth greater than that of any amplifier, limited by the on-resistance of the MOS sampling switch and the sampling capacitor value. These passive, high band-

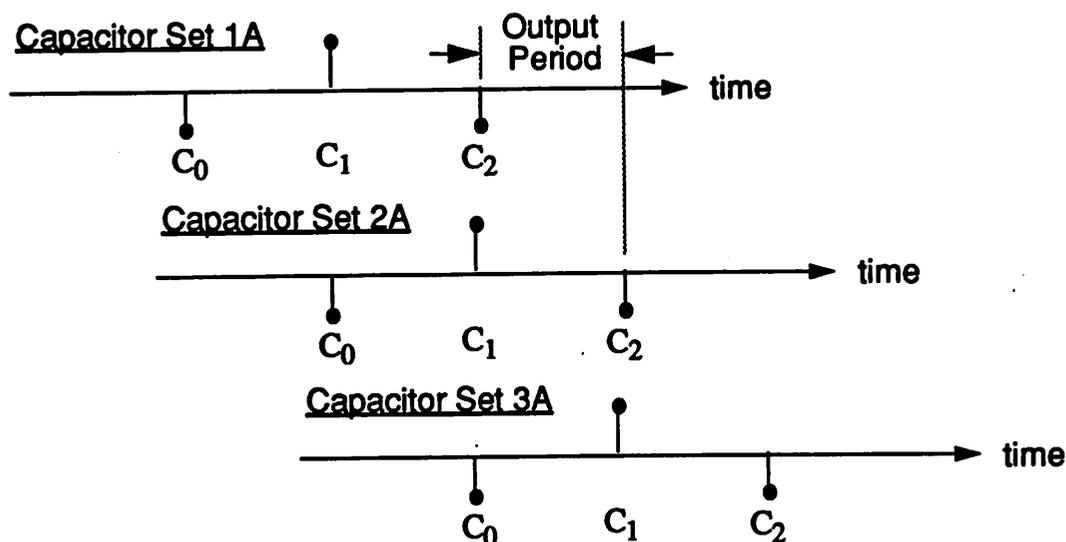
width sampling circuits appear in the front-end of the switched capacitor filter. The clock waveforms which control the sampling switches are inherently digital waveforms and the highest effective sampling rate achievable is limited by gate delays (In practice, sampling phase jitter and offsets require thoughtful design of the sampling clock generators.). High speed operation of conventional switched-capacitors filters is limited by the amplifier settling times because in conventional approaches, amplifiers must settle at the output rate. However, in the proposed architecture, the amplifier settling time bottleneck is broken due to the parallel nature and amplifiers are allowed multiple output periods to settle.

In principal, only switches and capacitors are needed to realize many switched-capacitor signal processing functions such as sample-and-holds. However, an operational amplifier in a closed loop configuration is present in most practical SC signal processing circuits in order to reduce distortion and sensitivity to parasitic capacitance. Due to the parallel nature of the proposed approach, unlike other switched-capacitor filter approaches, the speed is not necessarily limited by the amplifier settling time. Instead, there is a speed versus complexity trade-off because in principal, more parallelism can be used to reduce amplifier settling time requirements. This is the key feature which differentiates this architecture from those previously reported.

### 5.2.3 Single-Rate Filters

Single-rate filters are the simple case where the decimation factor is one so the input rate is equal to the output rate. These filters are not useful in baud-rate sampled systems as anti-alias filters, but they can be as analog equalizers.

A good example of a useful single-rate analog equalizer is the cosine equalizer discussed earlier in this chapter. The implementation of a programmable cosine equalizer is described in detail in Chapter 9. A conceptual diagram illustrating the operation of Capacitor Sets 1A-3A is shown in Figure 5.14 for a cosine equalizer. Note that the output period is equal to the sampling period.



**Figure 5.14** Conceptual diagram showing operation of Capacitor Sets 1A-3A for a 3-tap cosine equalizer as an example of a single-rate filter.

### 5.2.4 Long Impulse Response Decimation Filters

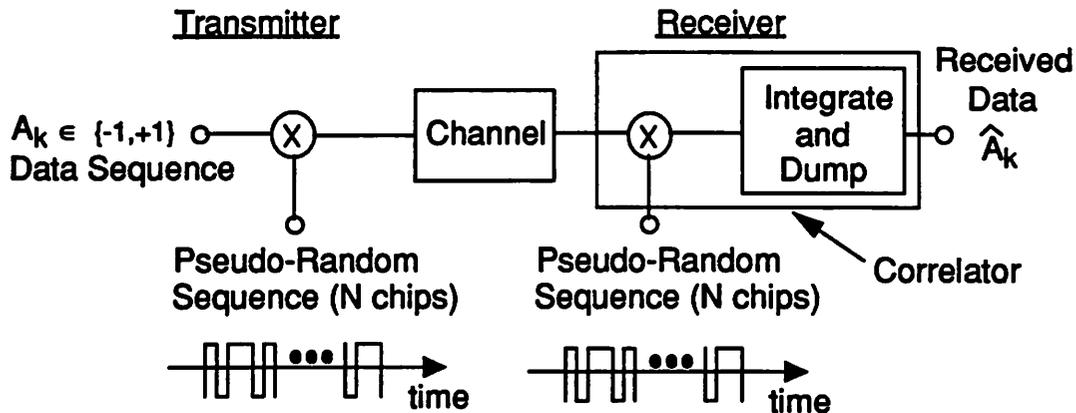
We shall refer to the SC FIR topology described in previous sections as the *brute force* approach since it follows from a straight forward implementation utilizing the SC inner product generator or (IPG). This approach is more suited to shorter impulse responses. Particularly, those where the impulse response length in output periods  $L$  is less than or equal to the decimation factor  $M$ . As the impulse response length increases, the sampling capacitor arrays can become quite large and complicated, with increasing amounts of parasitic capacitance due to interconnect that can greatly compromise performance. Also, in the brute force approach, the number of sampling phases equals twice the number of taps, further increasing the cost of long impulse response implementation. Thus, for long impulse responses, particularly with larger values of  $M$ , a different architecture may be appropriate.

As discussed earlier, when analog signal processing is employed, each block in the signal path contributes voltage offsets. When these offsets contribute periodically to the output, a fixed pattern

noise results and can greatly reduce the signal-to-noise ratio attainable through the processor. In principal, there are techniques that can be used to cancel these offsets but they can be extremely complicated and costly to implement. Therefore, in general, high performance filters requiring long impulse responses may best be performed in the digital domain. However, there are applications requiring the use of long decimation filters such as correlators in spread spectrum receivers where absolute accuracy is not important due to the robustness inherent in the correlating process. Let us examine the example of the spread spectrum receiver as a vehicle to present an architecture which was developed for the implementation of particularly long impulse responses.

#### 5.2.4.1 Example: Spread Spectrum Decorrelator

In a spread spectrum [21] transmitter, the data sequence  $\{-1,+1\}$  is modulated by a pseudo-random sequence of values  $\{-1,+1\}$  and length  $N$ . The elements of the pseudo-random sequence are referred to as chips. The data rate is the rate at which the data sequence is sent and the chip rate is  $N$  times the data rate. In these systems, a correlator is required in the receiver which multiplies the received signal by the same pseudo-random sequence prior to an integrate-and-dump function over one bit period. In effect, such a receiver front-end performs a correlation of the pseudo-random sequence used to modulate the signal with a local pseudo-random sequence. The ideal result in the absence of noise is  $\pm N$  at the output of the correlator when the transmit and receive pseudo-random sequences are identical and the sampling phase is correct, and zero otherwise. This is because ideally, an orthogonal set of pseudo-random sequences are used with auto-correlation functions that are the impulse function. A simplified block diagram of a spread spectrum receiver is shown in Figure 5.15.



**Figure 5.15** Simplified block diagram of a spread spectrum communications channel. The correlator can be viewed as a decimation filter with decimation factor equal to the number of chips in the pseudo-random sequence.

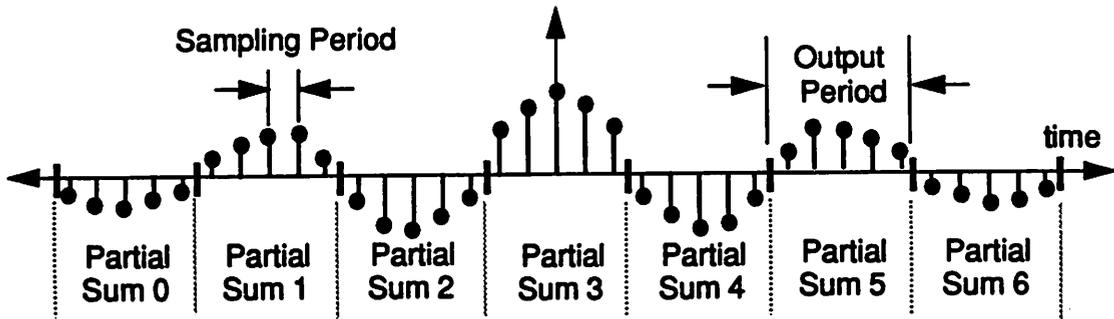
The correlator can be viewed as a decimation filter with taps weights  $\pm 1$  (corresponding to the particular pseudo-random sequence) and a decimation factor equal to the number of chips per bit (also referred to as the spreading factor). This decimation factor may be in the tens, hundreds, or even thousands (For SC implementation, decimation factors in the tens are all that are practical). In these instances, it may be rather costly in terms of hardware to use the brute-force realization introduced earlier in this chapter.

#### 5.2.4.2 Partial Sum-Accumulator Bank Architecture

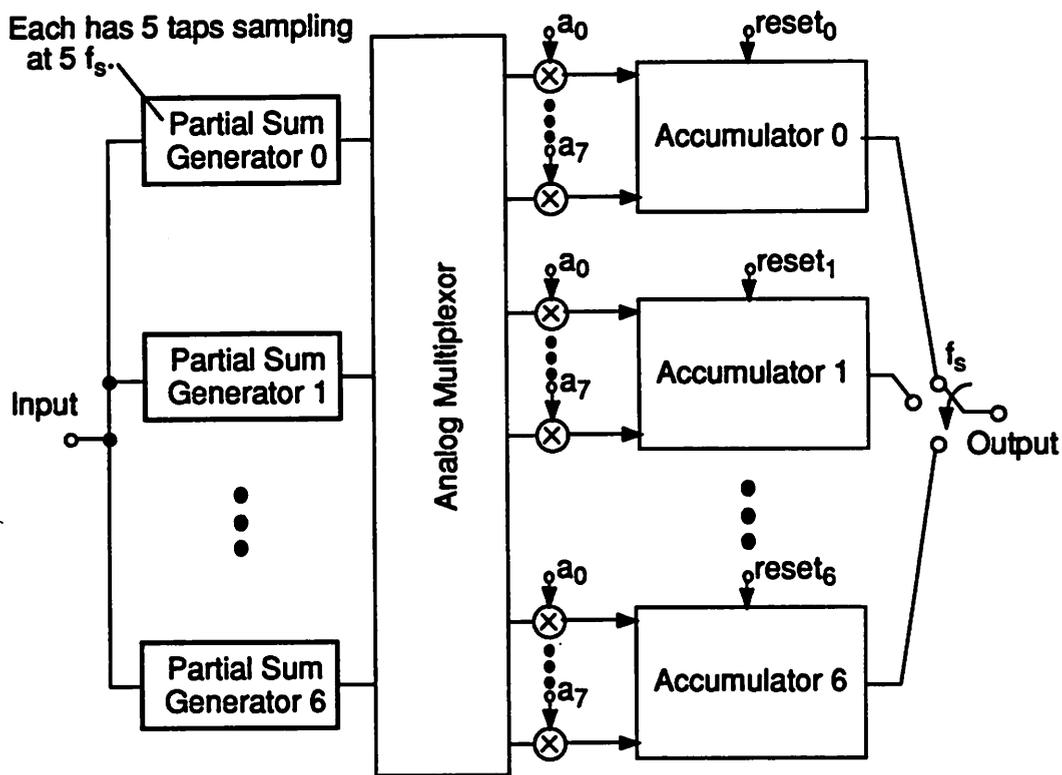
Let us now examine what we shall call the *Partial Sum-Accumulator Bank* approach for the realization of long impulse response filters. Suppose we wish to implement an impulse response with  $L$  output periods and a decimation factor of  $M$  (total number of taps is  $LM$ ). The key point of this approach is that instead of generating the inner product to create an output all at once, the inner product is decomposed into partial sums which are summed together in a second stage made up of a bank of accumulators. Recall that a filter output can be written as an inner product and rewrite the inner product as the sum of  $L$  partial sums, each with  $M$  filter taps (where  $M$  is the decimation factor). Each of these partial sums can be generated using an SC IPG realizing the inner

product with only  $M$  coefficients (instead of  $ML$ ). Let us refer to such a structure as a *partial sum generator* (PSG).

We shall describe the partial sum-accumulator bank approach through an example. Consider the architecture shown in Figure 5.16 for implementation of the 35-tap 5:1 decimation filter impulse response shown at the top of the figure. Note in the figure that there are  $L$  PSGs where in this example  $L$  is equal to 7, each realizing in sequence  $M$  coefficients ( $M$  equal to 5) of the  $ML$  ( $7 \cdot 5 = 35$ ) tap long impulse response, are followed by a  $T$ -rate sampling second stage which accumulates in succession, 7 partial sums over  $LT$  seconds. The second stage is comprised of 7 accumulator stages operating in an interleaved manner  $T$  seconds apart. During every output period, all 7 partial sums are computed and one of the accumulators sums in its 7th partial sum. The output of this accumulator is then sampled by the commutator before it is re-initialized to zero and begins the accumulation process once again.



(a)



(b)

**Figure 5.16** Block diagram of Partial Sum -- Accumulator Bank Approach for the implementation of long decimation filters. (a) Impulse response with  $L=7$  (impulse response length) and  $M=5$  (decimation factor). (b) Realization using Partial Sum -- Accumulator Bank Approach.

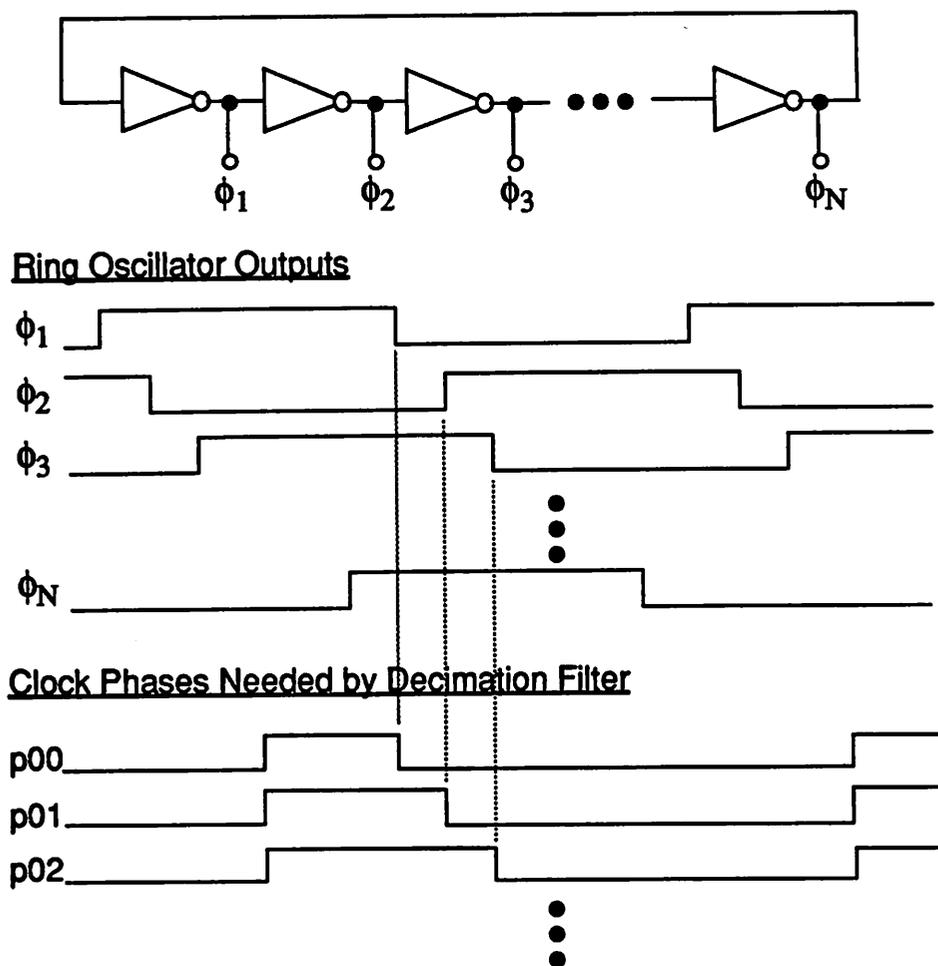
The partial sum - accumulator bank approach avoids the problems associated with generating the entire inner product all at once by decomposing it into partial sums of length  $M$  (which are

generated every output period) and appropriately accumulating them in a second stage. The outputs of the PSGs can be scaled prior to accumulation (using scaling factors  $a_0$  through  $a_{L-1}$  as shown in the figure, facilitating the implementation of larger coefficient spreads. Another big advantage is that the number of sampling phases required is reduced from  $2LM$  to  $2M$  which in this example would be a reduction from 70 down to 10. This reduction in the number of clocks also is indicative of a reduction in the number of switches and capacitors. Although the total opamp count may be increased, it may be well worth the reduction in overall complexity.

### 5.3 Sampling Clock Generation

In typical DSP-based communication systems, there is a local clock in the receiver usually in a timing recovery loop locked on to the incoming data stream. This clock is normally at the symbol rate and is used to define the sampling phase as well as clock the A/D converter and the DSP. Thus, all clocks including those which control interleaving as well as those which define the sampling instants of the decimation filter need to be generated on chip. Generation of these clocks is facilitated by use of either a delay line locked to an external clock (delay locked loop- DLL) or by connecting the delay line back to itself creating a ring oscillator controlled by a phase-lock loop (PLL). By tapping off at appropriate points along the delay line, the staggered timing phases required to clock the interleaved channels can be easily generated. These staggered timing phases can also be used to generate the sampling waveforms, enabling input sampling rates greater than the external clock frequency. The sampling waveforms are the most critical because jitter in the sampling phase results in noise at the output as we shall see later. In Chapters 6 and 8, implementation issues of high speed clock generation for the SC decimation filter based upon a ring oscillator PLL are addressed.

A simplified block diagram of a ring oscillator-based sampling clock generator circuit is shown in Figure 5.17. Each of the phases in a delay line or ring oscillator have a 50% duty cycle which makes them inappropriate for direct use as sampling clocks. Therefore, some logic must be performed on the delay line outputs in order to generate suitable sampling waveforms.



**Figure 5.17** Ring oscillator with output phases shown along with clock phases required by the SC decimation filter. Clearly, the oscillator outputs cannot be used directly.

Noise and offsets in the delay line result in jitter in the sampling phase and non-uniform sampling, respectively. Supply and device noise are the primary noise sources. Circuit techniques have been developed to combat their effects [12,19]. The use of a fully differential delay line minimizes the effect of supply noise. Device noise effects are minimized by careful design of the device size and currents of the transistors in the delay elements. Non-uniform sampling is the result of mismatches in parasitics capacitances and voltage offsets in the circuits which generated the sampling signals. Each device in the path from the delay line through to the sampling signal has an input-

referred voltage offset which when divided by the slope of the signal driving that device results in a time offset that accumulates through the path. As a result, it is desirable to bias devices throughout the path at high current. In order to minimize non-uniform sampling, it is important to maintain a fully balanced layout (including parasitics), minimize the amount of circuitry from the delay line to the generation of the sampling signal, and keep the slopes of signals through this path sufficiently large to minimize the effects of voltage offsets to each sampling phase. Implementation issues of low jitter sampling phase design are discussed in detail in Chapter 7.

Because of the power and area cost of generating a low jitter sampling phase, the number of sampling phases required must be a factor in the choice of an architecture. For example, the brute-force approach requires  $2LM$  phases as compared with  $2M$  required by the partial sum-accumulator bank approach.

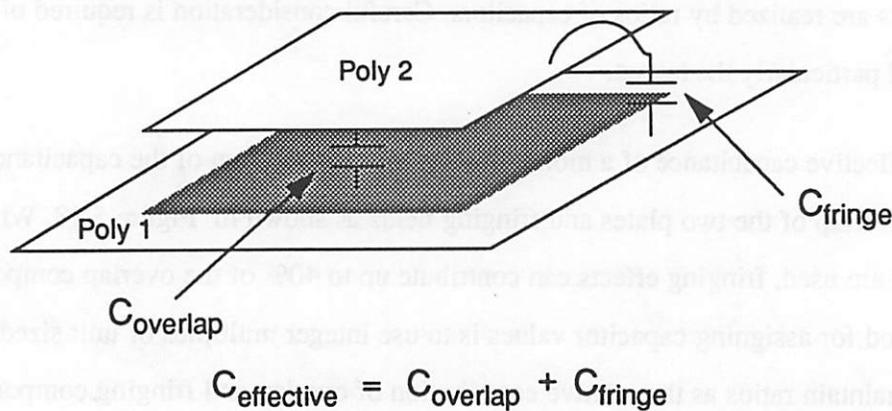
## 5.4 Capacitor Design

In the switched-capacitor FIR filter implementation approach described in Section 5.2, filter tap weights are realized by ratios of capacitors. Careful consideration is required of the capacitor design and particularly the layout.

The effective capacitance of a monolithic capacitor is the sum of the capacitances associated with the overlap of the two plates and fringing fields as shown in Figure 5.18. When small unit capacitors are used, fringing effects can contribute up to 40% of the overlap component [22,23]. One method for assigning capacitor values is to use integer multiples of unit sized capacitors in order to maintain ratios as the relative contribution of overlap and fringing components changes with process variations. The granularity introduced by the strict use of unit capacitors is analogous to a digital filter where the filter taps are implemented using finite precision. One of the key design challenges of specifying the capacitor sizes using unit capacitors are the trade-offs between granularity of the tap weights, capacitive loading on the amplifier, maintaining an appropriate feedback factor for amplifier speed and stability, and capacitor matching. Fine granularity in the capacitor ratios requires a large number of unit capacitors. For example, suppose a ratio of 9.5:1 is required. This could be accomplished with 19 units for one capacitor and 2 for the other. The finer the granularity, the larger the total number units that will be needed. However, the larger number of unit

capacitors means more total capacitance which results in a need for large amplifiers consuming more power. As a result, it is desirable to use small unit capacitors. But capacitor matching is roughly inversely proportional to the capacitor area [14] which means that smaller capacitors don't match as well.

Therefore, it is important to find an appropriate design with the minimum amount of granularity required in the tap weight ratios meeting the filter specifications. Next, find the minimum size of unit capacitor such that when the total capacitance of the input capacitor array is summed, the feedback factor of the amplifier is such that the amplifier can be designed to drive the load capacitance with this resulting feedback factor. If the capacitive load on an amplifier is much larger than both the total input (sampling) capacitance and feedback capacitance, increasing the size of the input and feedback capacitors have the effect of increasing the feedback factor which can increase the overall bandwidth of the circuit.



**Figure 5.18** Schematic showing effective capacitance being the sum of the overlap capacitance due to the overlap area between the parallel plates and the fringing capacitance due to fringing fields.

Since the dynamic range performance of SC FIR filters is limited,  $kT/C$  noise is generally not an issue and capacitor size is driven primarily by matching considerations. The random mis-

matches due to processing variations affect both the frequency response and distortion. Performance limitations of SC FIR filters due to capacitor mismatch are discussed in the next section.

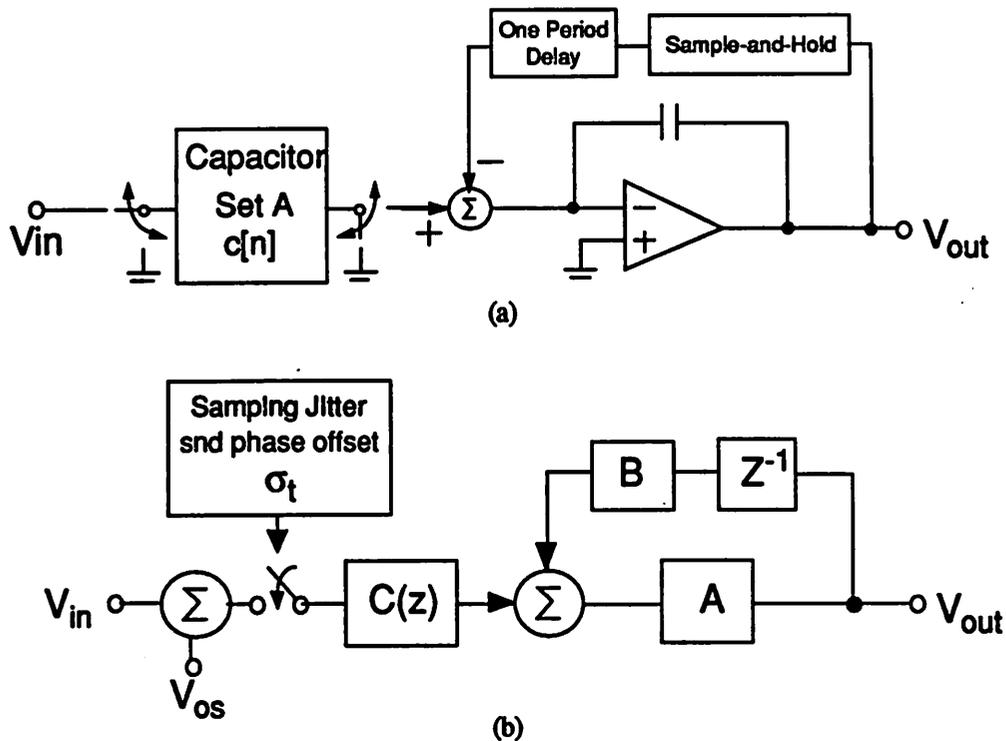
## 5.5 Performance Limitations of High Speed SC FIR Filters

Parallel analog signal processing has been applied to high speed analog-to-digital converters and recently to an all CMOS fiber optic receiver circuit in order to achieve processing rates that are much higher than otherwise achievable using pipelining or other serial approaches. However, when parallel architectures are applied to analog signal processing, gain, offset, and timing mismatches between parallel channels contribute noise and distortion products which limit the attainable SNR. The effects of these non-idealities on the SC FIR filter architectures presented earlier in this chapter are similar to those encountered in parallel A/D converter architectures and much can be gained from previous analyses of these circuits [10,11,13]. The major difference between the parallel filter and A/D converter architectures is that the analyses previously performed have assumed the gain of each converter to be constant over the frequency range of interests, i.e., the frequency response is flat. However, in the case of the filter, the ideal frequency response is that of the filter being realized. It turns out that this difference can be easily taken into account and will be presented in this section.

Let us now examine the key factors which limit the achievable speed and performance of the proposed SC FIR filter architectures.

### 5.5.1 Model for Sources of Error

Shown in Figure 5.19 is a simplified model of a switched-capacitor inner product generator together with the model we shall use to analyze the effects of non-idealities of this block on performance of filters using the parallel SC filter architecture. The model of the inner product generator has a key simplification in that there is only one coefficient set that is shown where in practical implementation there will be at least 2 interleaved sets. We will return to this point when necessary.



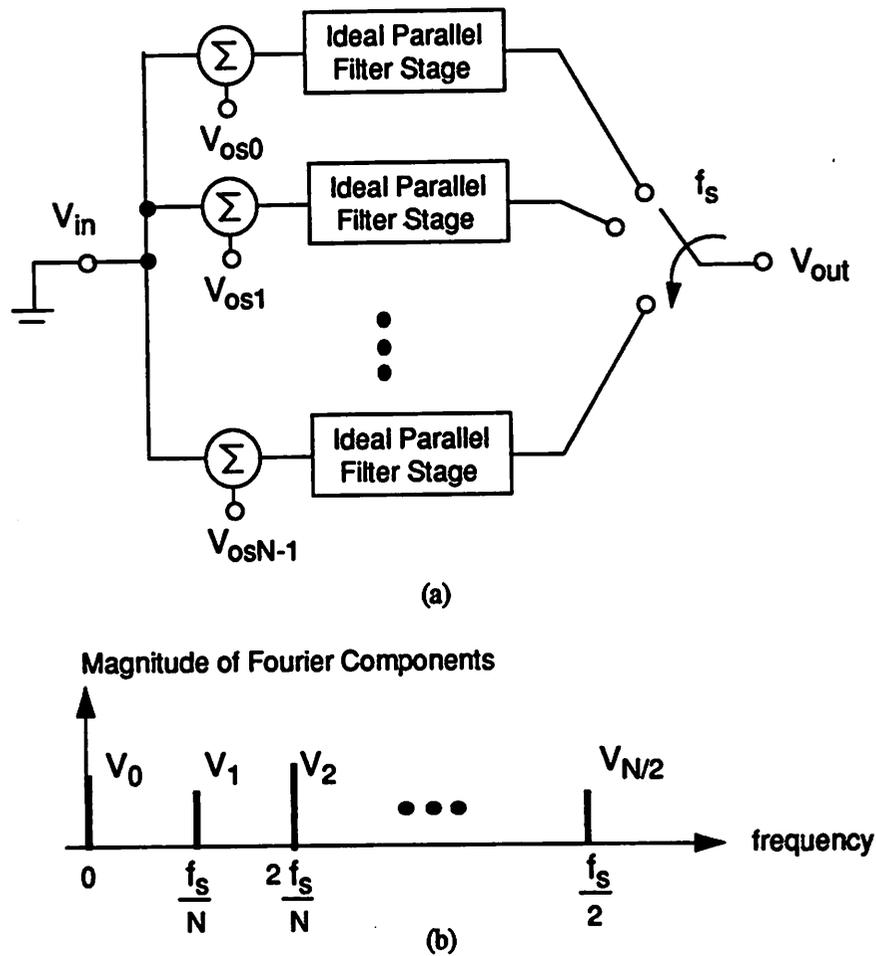
**Figure 5.19** Model for analysis of non-idealities of the partial sum generator on the performance of parallel SC FIR filters. (a) Simplified partial sum generator with only one input capacitor set. (b) Model for analysis.

The 4 key components in this model include an input offset voltage  $V_{os}$ , sampling jitter and phase offset  $\sigma_t$ , tap weight vector  $C(z)$ , forward d.c. gain factor  $A$ , and feedback gain factor  $B$ . We shall examine the sources of each of the components and their effect on filter performance individually. We will assume that each of the components are independent which will allow each of the effects to be considered individually. Let us assume that  $N$  parallel stages are being used to implement an  $M:1$  decimation filter, and that the output sampling frequency is  $f_s$ .

### 5.5.2 Input-Referred Offset Voltage

The input-referred offset voltage  $V_{os}$  has contributions from the offset of the amplifier and the charge injection and clock feedthrough from the switches. Each of the capacitor array/amplifier combinations have a different  $V_{os}$ . Assuming zero input, the output commutator effectively sam-

ples each of the offsets as it traverses the outputs resulting in a periodic voltage pattern with period  $N$  as shown schematically in Figure 5.20a. The result is in the frequency domain, there are tones independent of the input signal at multiples of  $f_s/N$  and d.c. as shown in Figure 5.20b.



**Figure 5.20** (a) Model of input-referred voltage offsets. (b) Resulting frequency domain response for  $N$  even.

The output spectrum as a result of the voltage offsets for  $N$  parallel stages is

$$Y(j\omega) = \frac{1}{T} \sum_{n=-\infty}^{\infty} V_n \delta\left(f - \frac{nf_s}{N}\right) \quad (5.7)$$

the magnitude of the Fourier components  $V_n$  is given by

$$V_n = \frac{1}{N} \sum_{k=0}^{N-1} Vos_k e^{-\frac{j(2\pi)nk}{N}} \quad (5.8)$$

The expected value of these components has been shown to be

$$E\{|V_n|\} = \frac{\sigma_{vos}}{2} \sqrt{\frac{\pi}{N}} \quad (5.9)$$

where  $\sigma_{vos}$  is the standard deviation of the offsets.

In order to have these components 45dB down below an output signal tone normalized to unity in a system with 3 parallel stages, we find

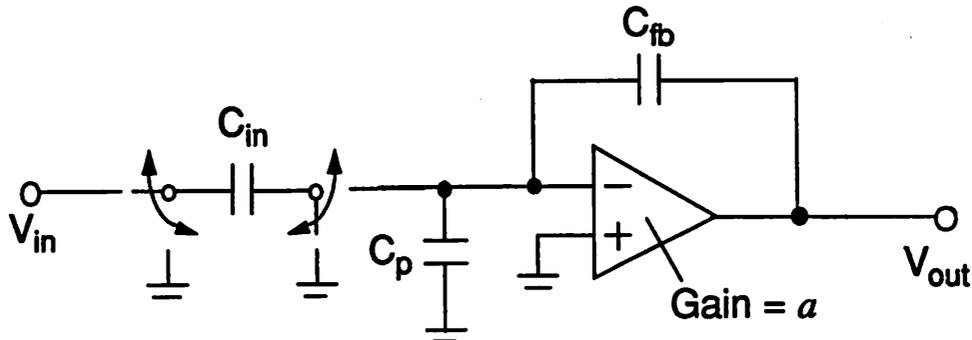
$$\sigma_{vos} = \frac{2 \cdot 10^{-45/20}}{\sqrt{\frac{\pi}{3}}} = 0.01099 \quad (5.10)$$

This means that the standard deviation of the channel offsets must be kept below 11mV in order to meet the specification.

The amplifiers are usually shared by more than one capacitor set and so the contribution of the input-referred offset of the amplifier is a common term in more than one of the offset voltages in the model. Thus, each of the offsets are not totally independent and as a result the magnitude of the frequency components may not be equal but will vary with the periodicities of the offset components. The relative contribution to the input offset from the amplifiers and the switches will vary with specific design approaches and processing quality. Some circuit techniques to reduce the magnitude of voltage offsets are presented in Chapter 6.

### 5.5.3 D.C. Gain

The forward gain factor  $A$  in the model of Figure 5.22 models the effect of gain mismatch between adjacent channels. This is due both to capacitor mismatch and gain mismatch between parallel amplifiers. Let us simplify the SC inner product generator to a one-tap gain element which reduces to an SC sample-and-hold as shown in Figure 5.21.



**Figure 5.21** Circuit for calculating closed loop gain from input to output including effect of finite opamp gain and summing node parasitic  $C_p$ .

The gain from input to output can be written as:

$$V_{out} = V_{in} \cdot \frac{C_{in}}{C_{fb} + \frac{C_{in} + C_{fb} + C_p}{a}} \quad (5.11)$$

where  $a$  is the open loop gain of the operational amplifier,  $C_p$  is the parasitic capacitance on the summing node,  $C_{in}$  is the input capacitor, and  $C_{fb}$  is the feedback capacitor. The third term on the right side of Eq. 5.11 contains the effect of finite amplifier gain which results in a gain error from the ideal transfer function of  $C_{in}/C_{fb}$ . This term is the forward gain factor  $a$ . A model containing only the d.c. gain mismatch in parallel channels is shown in Figure 5.22a where an ideal filter with transfer function  $C(z)$  is followed by  $N$  parallel interleaved gain stages, each with a different gain  $a_k$ .

The output spectrum for input  $x(t)$  is

$$Y(f) = \frac{1}{T} \sum_{n=-\infty}^{\infty} A_n X \left[ j \left( f - \frac{n}{NT} \right) \right] \quad (5.12)$$

where

$$A_n = \frac{1}{N} \sum_{k=0}^{N-1} a_k e^{-j \frac{(2\pi)mk}{N}} \quad (5.13)$$

For a sinusoidal input of  $\sin(2\pi f_{in}t)$ , the output spectrum is

$$Y(f) = \frac{1}{T} \sum_{n=-\infty}^{\infty} \frac{A_n}{2j} \left[ \delta \left( f - f_{in} - \frac{nf_s}{N} \right) - \delta \left( f + f_{in} - \frac{nf_s}{N} \right) \right] \quad (5.14)$$

As can be seen in Eq. 5.14, for a sinusoidal input, d.c. gain mismatches between parallel stages results in sidebands at the input frequency about  $f_s/N$ ,  $2f_s/N$ , on up to  $f_s/2$  as shown in Figure 5.22b. The sidebands come in pairs and their amplitudes were given in Eq. 5.17.

The expected value of the Fourier components has been shown to be

$$E \{ |A_n| \} = \frac{\sigma_{ak}}{2} \sqrt{\frac{\pi}{N}} \quad (5.15)$$

where  $\sigma_{ak}$  is the standard deviation of the channel-to-channel gain mismatch.

In order to have these components 45dB down below an output signal tone normalized to unity in a system with 3 parallel stages, the channel-to-channel mismatch must be better than

$$\sigma_{ak} = \frac{2 \cdot 10^{-45/20}}{\sqrt{\frac{\pi}{3}}} = 0.01099 \quad (5.16)$$

or 1 percent.

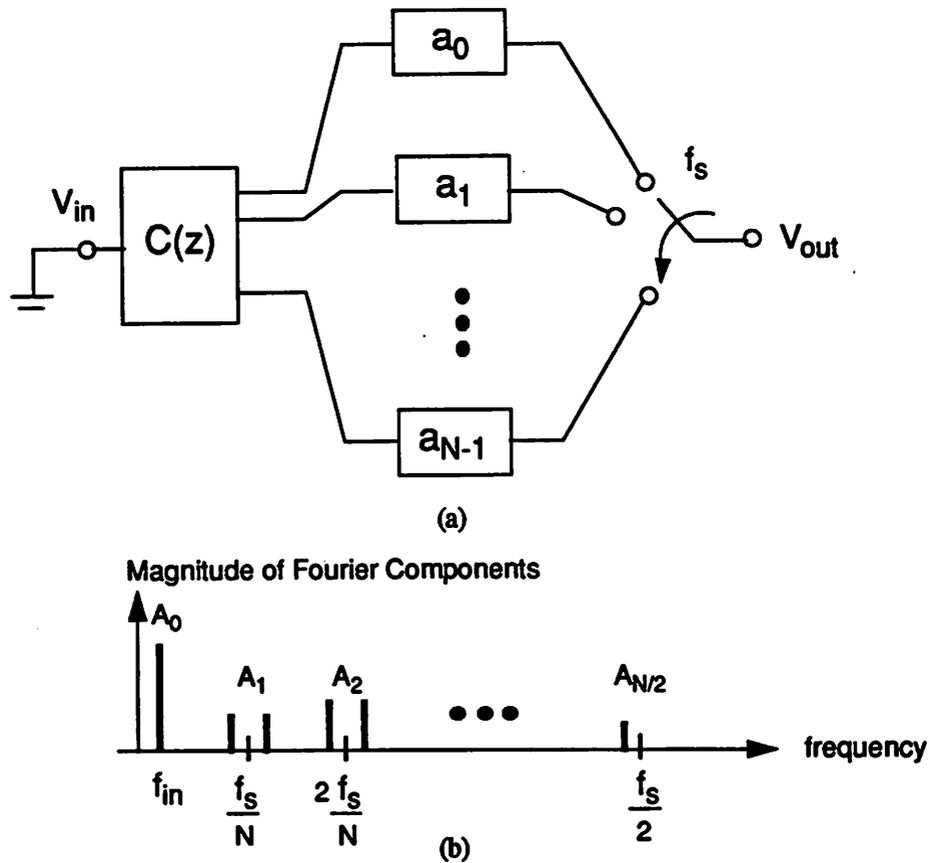
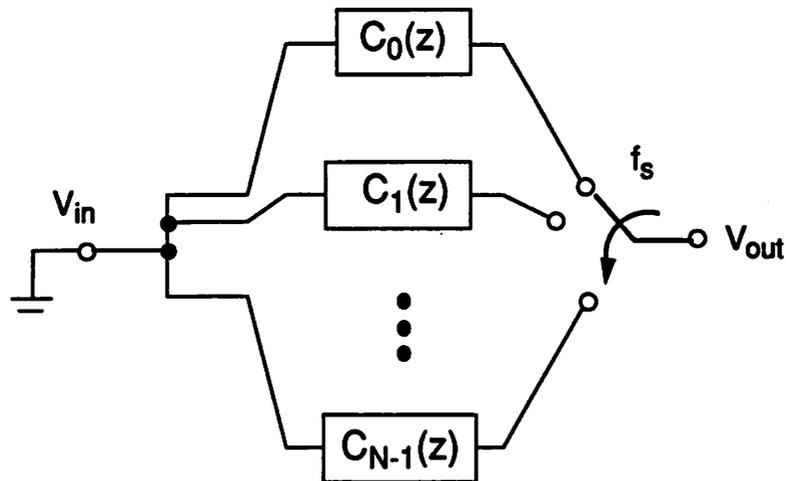


Figure 5.22 (a) Model for analyzing d.c. gain mismatch. (b) Effect on the frequency spectrum for a sine wave input.

### 5.5.4 Variation in Response of Parallel Filters

In the parallel architecture, an FIR filter response is realized using parallel inner product generators each operating in a different phase. Each parallel filter may have a slightly different frequency response. This situation is shown conceptually in Figure 5.23 where  $N$  parallel filters  $C_0(z)$  through  $C_{N-1}(z)$  are shown. Ideally, all  $N$  of these filter sections have the exact same frequency response but due to two primary effects, there may have different responses. First, processing variations result in variations of the capacitor ratios which set the tap weights of the filter. Second, the feedback gain factor  $B$  of Figure 5.19 which models the non-ideal cancellation of

charge in the integrating capacitor from cycle-to-cycle may not equal the forward gain  $A$ . As a result, some residual voltage will remain from cycle-to-cycle and the sample-and-hold appears as a recursive filter.



**Figure 5.23** Effect of variations in the frequency response of each parallel filter and the effect on the frequency domain response of the filter to an input at  $f_{in}$

The effect on the filter impulse response is similar to that of d.c. gain mismatch. The key difference is that since each filter is slightly different, the gain error is a function of frequency and thus the magnitude of the sidebands is a function of the input signal frequency. Eq. 5.17 can be changed to reflect the frequency dependence of the sideband magnitudes. This is given by

$$A_n(f_{in}) = \frac{1}{N} \sum_{k=0}^{N-1} a_k(f_{in}) e^{-j \frac{(2\pi)mk}{N}} \quad (5.17)$$

where  $a_n(f_{in})$  is the response of filter  $n$  at input frequency  $f_{in}$ . The expected value of the Fourier components can be found modifying Eq. 5.15 to be a function of the input signal frequency in a similar manner.

### 5.5.5 Sampling Jitter and Fixed Phase Offsets

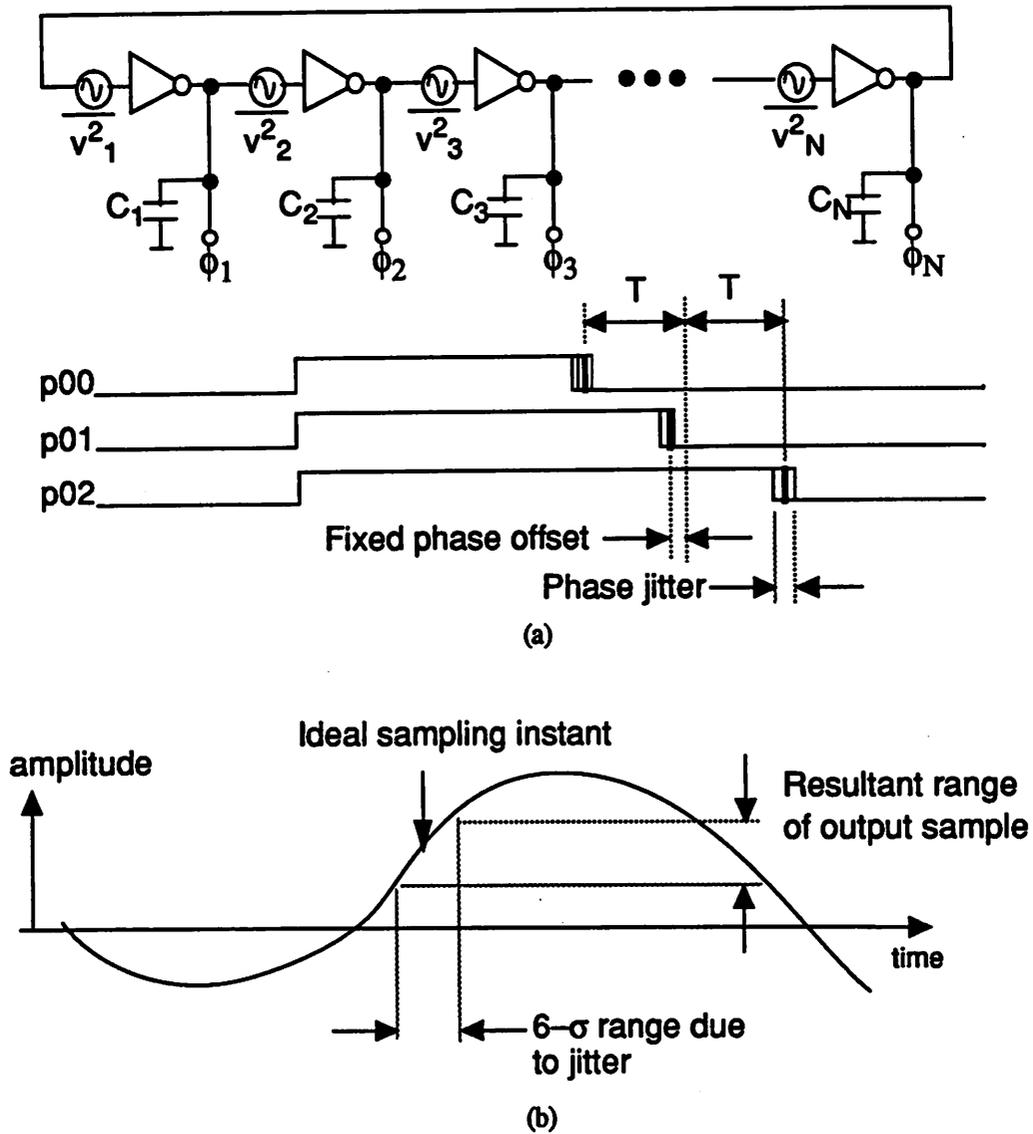
Two time domain non-idealities are important and shown conceptually in Figure 5.24. First, device noise in the ring oscillator transistors results in a randomness of the actual phase generated by each delay element output. This randomness results in a jitter in the resultant sampling phase as shown in the figure. The effect on the output sample is shown in part [b] of the figure where the ideal sampling phase is indicated by the arrow and the range of randomness of the sampling phase due to jitter (horizontal variation). The result is a range for the output samples (vertical variation) that clearly depends on the instantaneous slope of the input signal in the vicinity of a particular sampling instant and the variance of the jitter.

Second, variations in the capacitive loading result in *fixed* phase offsets of the sampling phases. The result is a non-uniform sampling that is periodic in the number of parallel sampling stages which will be a minimum of two times the number of parallel filters stages. This is due to each parallel stage having at least two interleaved sets of sampling capacitors. The effect is similar to the case of sampling jitter because a signal dependent noise results but their effects on the frequency domain are different. Let us now examine the effect of these phenomenon on the frequency domain response.

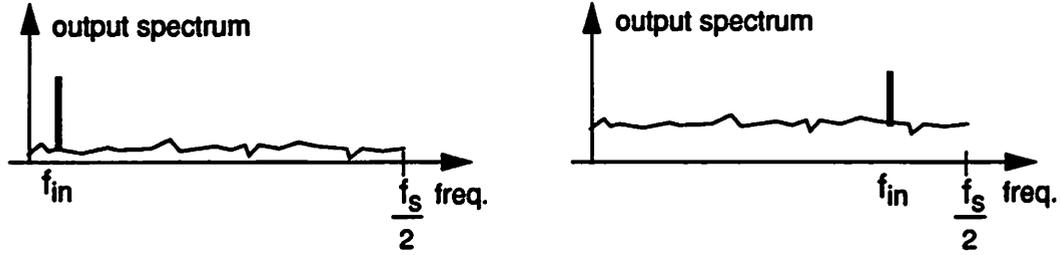
The effects of sampling jitter and fixed phase offsets are shown in Figure 5.24. Sampling jitter results in an additive noise component added to the output sample depending on the average slope of the input signal near the sampling instant and the root mean square value of the sampling jitter. Assuming the jitter spectrum is white, the effect on the frequency domain response will be an increase in the noise floor across all frequencies. The magnitude of this increase depends on the slope of the input signal and therefore will increase with increasing signal frequency as shown in Figure 5.24a.

Fixed phase offsets are equivalent to periodic non-uniform sampling. It can be shown that the effect of fixed phases offsets for a sinusoidal input results in a frequency dependent gain error. This effect is similar to that of frequency response mismatch in parallel channels with the same result that sidebands appear about the multiples of the frequency of non-uniformity. The magni-

tudes of these sidebands depends on the actual sampling phase offsets and the input signal frequency. This is shown in Figure 5.25b.



**Figure 5.24** (a) Conceptual schematic of ring oscillator with input-referred-noise in the delay elements and loading mismatches (Capacitors not necessarily equal) shown with the effect these non-idealities have on the sampling phases. (b) The result of sampling jitter and the resulting variance of the output sample.



a. Note: Plot on the right has higher input frequency.

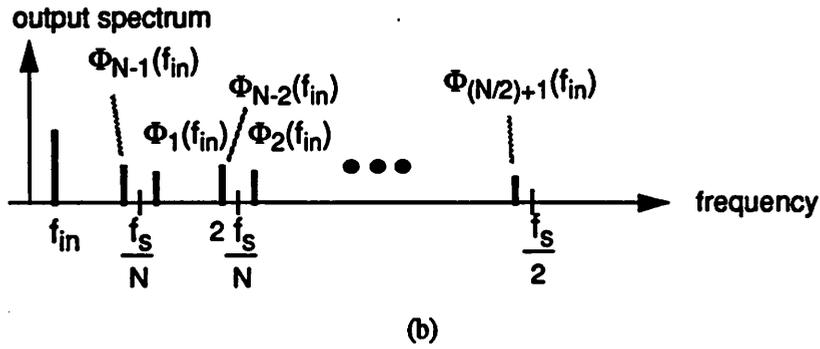


Figure 5.25 Frequency spectra for an input tone at  $f_{in}$  with (a) parallel sampling phase jitter and (b) sampling phase offsets.

The output spectrum for input  $x(t)$  is

$$Y(f) = \frac{1}{T} \sum_{n=-\infty}^{\infty} \Phi_n(f) X\left[j\left(f - \frac{n}{NT}\right)\right] \tag{5.18}$$

where

$$\Phi_n(f) = \frac{1}{N} \sum_{k=0}^{N-1} a_k e^{j\left(2\pi f - \frac{2\pi n}{NT}\right) \Delta t_k} e^{-j\left(\frac{2\pi}{N}\right) nk} \tag{5.19}$$

For a sinusoidal input of  $\sin(2\pi f_{in} t)$ , the output spectrum is

$$Y(f) = \frac{1}{T} \sum_{n=-\infty}^{\infty} \Phi_n \delta\left(f - f_{in} - \frac{nf_s}{N}\right) \tag{5.20}$$

where

$$\Phi_n(f_{in}) = \frac{1}{N} \sum_{k=0}^{N-1} (1 + j2\pi f_{in} \Delta t_k) e^{-j\left(\frac{2\pi}{N}\right) nk} \tag{5.21}$$

For a sinusoidal input, sampling phase mismatches between parallel channels results in sidebands at the input frequency about  $f_s/N$ ,  $2f_s/N$ , on up to  $f_s/2$  as in the gain mismatch condition. Two differences are that the magnitude of the sidebands are proportional to the input frequency and magnitudes of the sideband pairs about multiples of the input frequency are not equal.

The expected value of the Fourier components can be shown to be

$$E\{|\Phi_n(f_{in})|\} = \pi f_{in} \sigma_{\Delta t} \sqrt{\frac{\pi}{N}} \quad (5.22)$$

where  $\sigma_{\Delta t}$  is the standard deviation of the sampling phase mismatch.

In order to have these components 45dB down for an input tone at 30 MHz normalized to unity in a system with 3 parallel stages, the channel-to-channel mismatch must be better than

$$\sigma_{\Delta t} = \frac{10^{-45/20}}{\pi (30M) \sqrt{\frac{\pi}{3}}} = 58.3 \times 10^{-12} \quad (5.23)$$

or 58 psec.

It has been shown that for N-bits of dynamic range and a sinusoidal input frequency of  $f_{in}$  with the peak amplitude normalized to unity, sampling jitter should be kept below

$$\sigma_t = \frac{1}{\pi f_{in} 2^{N+1}} \quad (5.24)$$

For a 6-bit system and an input frequency of 30 MHz, the sampling phase jitter needs to be kept below a root mean square value of 83 psec.

### 5.5.6 Summary of Non-Idealities

As a result of the non-idealities discussed in this section, these filters are best suited for applications where only modest resolution but high speed is required. The effects of coefficient mismatch due to capacitor variations was examined in [7]. Issues of complexity and the associated non-idealities discussed in this section will probably limit the performance before the effects of

coefficient mismatch on attainable stop band rejection will become a limiting factor to performance.

In Table 5-1 below, the important non-idealities are listed with qualitative effects on filter performance.

**Table 5.1 Non-idealities and Qualitative Effect on Filter Performance**

Non-ideality	Input Signal Dependent?	Effect on Frequency Domain Response
Voltage offsets	No	Tones at multiples of $f_s/N$ .
Variation in Channel Frequency Responses and D.C. Gain Mismatches	Yes	Sidebands at the input frequency about multiples of $f_s/N$ .
Sampling Phase Jitter	Yes	Input signal frequency dependent noise floor (rise with increasing frequency).
Sampling Phase Offsets	Yes	Sidebands at the input frequency about multiples of $f_s/N$ .

In Chapter 8, a prototype integrated circuit is described which includes a 3:1 SC decimation filter and a 3-tap cosine equalizer using the architectures described in this section. The prototype was designed for an output rate of 100 MHz and 6 bits of resolution. Experimental results are presented in that chapter where the effect of some of the non-idealities discussed here are present.

## 5.6 Adaptive SC FIR Filters

Adaptive equalizers are very important in communications receivers because they enable high speed transmission over time varying channels. The motivation for adaptive equalization can be found in Chapter 2 and receiver architectures including adaptive equalizers can be found in Chapter 4.

In this section, a new architecture will be presented for analog adaptive equalization based upon the parallel switched-capacitor FIR filter structure described earlier in this chapter. We will begin this section with an examination of two alternative approaches before presenting the new proposed approach.

To date, no adaptive continuous-time filters suitable for application in the disk drive or other high data rate channel has been reported. Adaptive equalizers are usually performed in discrete-time and in the digital domain using transversal filter architectures. Use of transversal filter architectures enables employment of the Least Mean Square or LMS algorithm for recursively updating the coefficients. In this scenario, the filter is simply a transversal filter whose coefficients are individually controlled through a feedback path by the LMS algorithm as shown in Figure 5.26 (described in Chapter 2).

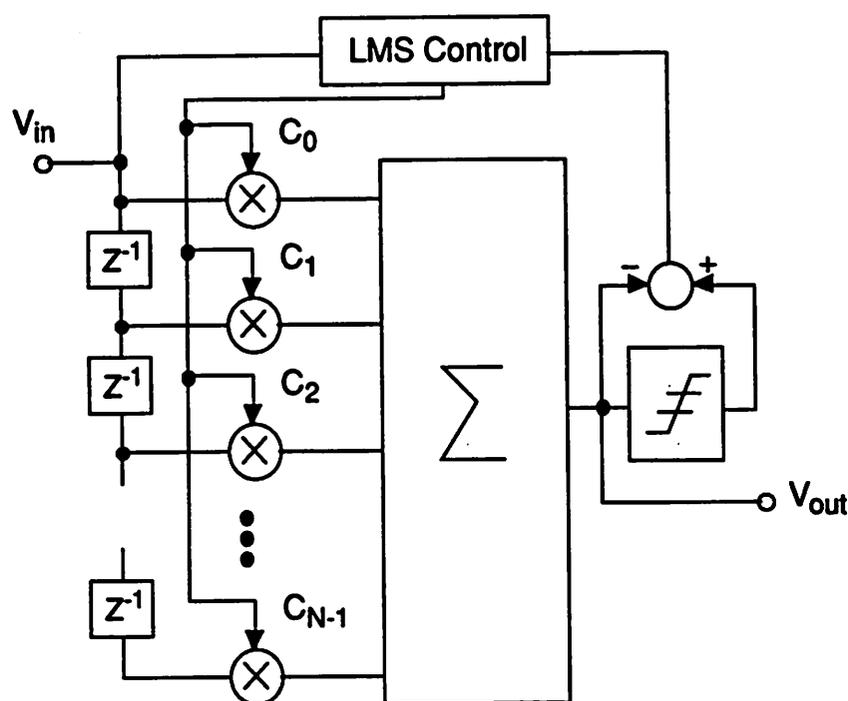


Figure 5.26 Block diagram of adaptive FIR filter employing the LMS algorithm for tap weight control.

The key problems for analog implementation of the structure shown in the figure above are realization of a high speed variable multiplier and implementation of a high speed transversal filter

structure. The latter problem can be addressed using the architecture proposed earlier in this chapter for high speed single-rate filters. However, there are different ways in which the programmable tap weights may be realized which we shall now examine.

### Gilbert Multiplier Cell

One way to perform the multiplication function in the analog domain is with use of a Gilbert Multiplier Cell [24] which requires a technology with a high speed bipolar transistor. The output voltage of a Gilbert Multiplier Cell is the product of two input voltages. The input signal is applied to one pair of inputs and a programmable voltage corresponding to the tap weight is applied to the other. Control of the tap weight depends on the particular implementation which might be from a continuous-time integrator [25] or the output of a D/A converter. A number of Gilbert Cells equal to the number of taps in the filter must be sampled appropriately and the output products must be summed to create one output of the adaptive FIR filter. By allowing the tap weights of the filter to be controlled by the LMS algorithm, an adaptive filter can be realized. There are a few limitations of this approach. First, it requires the use of a BiCMOS technology. Second, the linear input range of the Gilbert Cell is limited requiring the use of small signal levels which require the use of subsequent amplification which amplifies both noise and offsets. Lastly, control of the tap weights are in the analog domain. Digital control is desirable but requires a separate DAC which can be costly in terms of silicon area and power.

### Switched-Capacitor DAC

The switched-capacitor DAC provides another approach for implementation of the variable tap weights. One implementation uses a binary weighted capacitor array as shown in Figure 5.27. By digital control, the effective capacitor value can be changed resulting in a different effective tap weight. The granularity and range of the resultant tap weight is set by the size of the smallest capacitors and the total number of binary weighted capacitors in the array. An array is required for every variable tap in the filter which means that using the parallel filter architecture for high speed operation, multiple sets of programmable capacitor arrays are needed.

In order to achieve 6-7 bits of resolution in the filter tap weights, a large amount of parasitic capacitance results due to routing and the addition of transmission gates into the array. These

arrays can take large amounts of silicon area and the associated parasitic capacitance reduces feedback factors of the closed-loop amplifiers which increase required settling times. The key disadvantages of this approach are the large amount of silicon area required by the many numbers of programmable arrays that are needed and the speed degradation of the amplifiers as a result of the parasitic capacitance associated with the capacitor arrays.

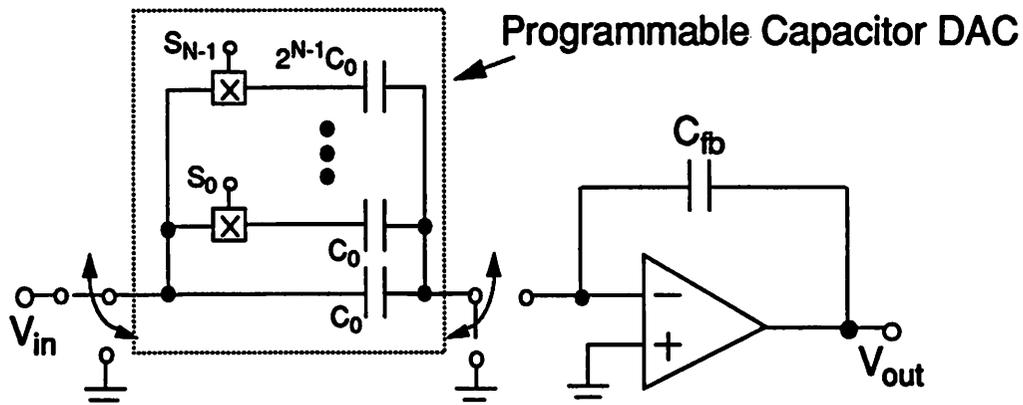
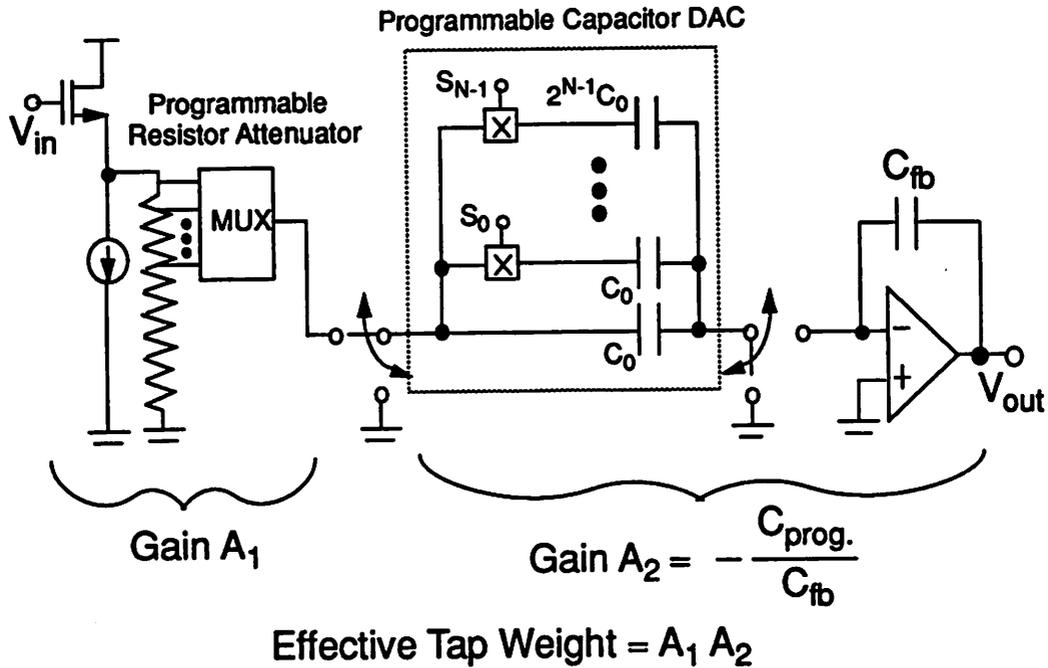


Figure 5.27 Switched-Capacitor DAC for realization of one filter tap.

### Resistor Attenuator/Capacitor DAC-Based Adaptive Filters

Another approach is to realize the tap weight in a two step manner with a programmable attenuator preceding a capacitor array. The capacitor array performs the coarse scaling and the attenuator provides the fine. The attenuator may be realized with a voltage follower with an inherent gain around unity driving a resistor string with a multiplexing network able to tap off the resistor string at different points. Attenuation is achieved through the resistor divider action of this circuit. An example of this approach is shown in Figure 5.28. Suppose the desired range is a tap weight in the range between .25 and .5 with approximately 6 bits of resolution. By having the range of the capacitor array provide a coarse levels of .3, .4, and .5 and preceding this with an attenuator with a range of .9 through 1.0 with 8 steps, a somewhat monotonic tap weight function may be realized. The corresponding effective gain as a function of first and second stage settings is shown in the figure.



**Figure 5.28** Conceptual diagram of the two stage resistor attenuator -- capacitor DAC approach for realization of variable filter tap weight.

The major advantage of this approach is that although the signal processing is performed in the analog domain, the tap weights may be stored digitally. Also, the large parasitic capacitance associated with a full SC DAC implementation is not present since the DAC is used only for coarse tuning. In practice, design of the analog multiplexor which chooses the attenuation can be quite difficult due to its associated capacitance and resistance. The advantage of the two step approach is that all the burden of realization of a wide range is not placed on one circuit block.

A block diagram of the proposed architecture is shown in Figure 5.29. The two key signal processing blocks in the signal path are the programmable resistors arrays and the programmable capacitor DACs. The output of the capacitor DAC stage is the filter output which drives a decision circuit which makes the tentative decisions used by the LMS control block and possibly an A/D converter for digital processing or an analog Viterbi decoder. The sign-LMS algorithm might be a good algorithm candidate for the LMS control.

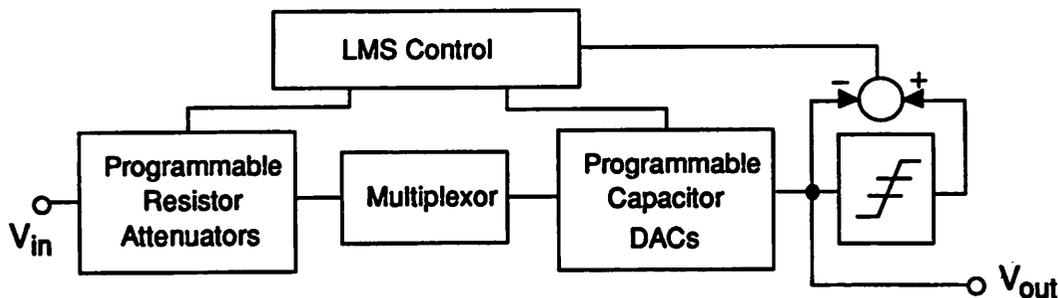


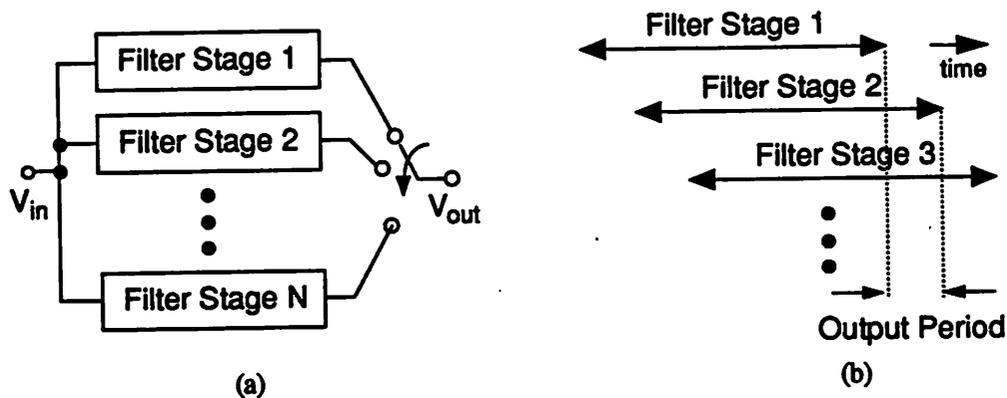
Figure 5.29 Block diagram proposed for implementation of analog adaptive equalizers.

## 5.7 Application of the Architecture to the Implementation of Digital Filters

The power of a CMOS digital circuit is equal to  $CV^2f$  where  $C$  is the capacitance being switched,  $V$  is the supply voltage (logic swing) and  $f$  is the frequency the circuit is being clocked. For short channel devices, due to velocity saturation the speed of a circuit is roughly proportional to the power supply voltage. In order to reduce the power consumption of high throughput digital circuits, one approach that has been proposed is to lower the supply voltage and employ parallelism and/or pipelining [26].

The parallel filter architecture proposed for implementation of high speed switched-capacitor transversal filters may also be applied to the implementation of high speed digital transversal filters. Two key problems associated with high speed digital filter implementation are the power consumed by the multipliers and the speed with which they can operate. By implementing the filter with parallel filter stages operating with a different phase as shown in Figure 5.30, a high throughput filter can be realized. Due to the reduced speed requirements of each parallel stage, simple cir-

cuit approaches may be used. Furthermore, power savings may be obtained by reducing the power supply voltage.



**Figure 5.30** Parallel Filter Architecture (a) with staggered phase operation as shown in (b). Each Filter Stage is comprised of multipliers and an accumulator. The input delay line is shared.

In a prototype design currently in fabrication, a 100 MHz 8-tap adaptive equalizer/Viterbi detector for the application of the magnetic disk read channel employing PRML signalling has been designed in a 1.2  $\mu\text{m}$  CMOS process [27,28]. The prototype was designed to operate with a 3.3 V power supply and consume less than 250 mW. The multipliers in the filter multiply two 6-bit numbers. Using the simple multiplier structure employed in the prototype, it was not possible to achieve 100 MHz throughput with a 5 V supply. The parallel architecture allowed use of the simple multiplier structure due to the reduced speed requirements of each stage. A block diagram of the prototype is shown in Figure 5.31.

The filter is realized with 4 parallel stages, each operating at 25 MHz. The output of only Filter Stage 1 is used to update the filter coefficients. The same set of filter coefficients are used by all four filters. The signed LMS algorithm was employed to minimize the time required for the coefficient update (There are two multiplies in the regular LMS algorithm which are simplified to an exclusive OR in the signed LMS algorithm). Since the partial response polynomial for Class IV partial response signalling is  $(1-D^2)$ , the even and odd samples are independent and the Viterbi detector can be decomposed into two independent interleaved  $(1-D)$  channels. Therefore, the out-

put of Filter Stage 1 and 3 feed one Viterbi decoder while the outputs from Filter Stage 2 and 4 feed another. The outputs of the two decoders are multiplexed to get the received bit stream.

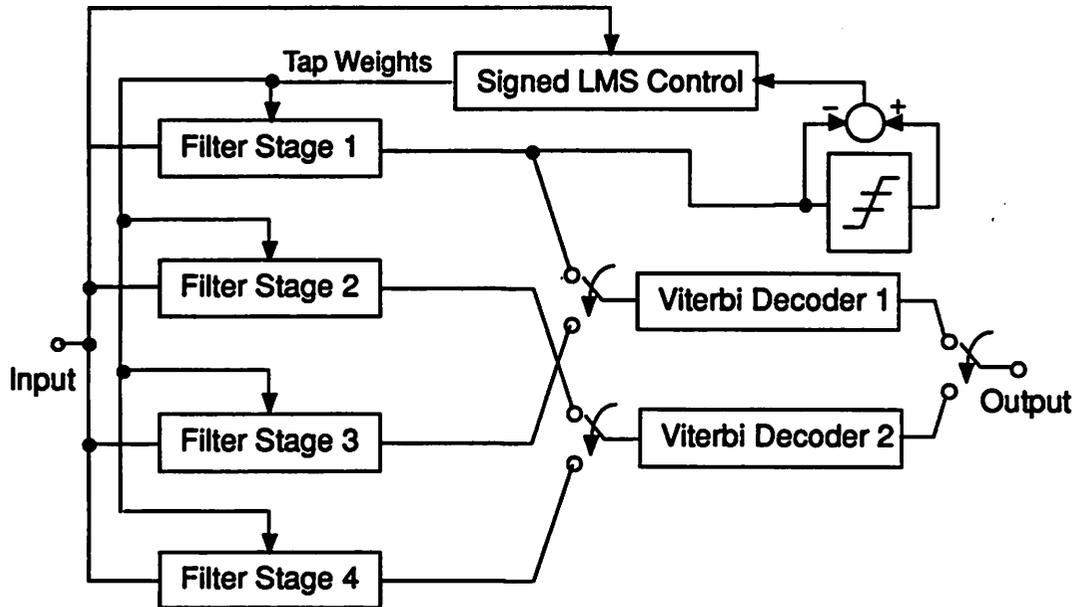


Figure 5.31 Block diagram of the prototype adaptive equalizer/Viterbi detector for 100 MHz operation with a 3.3 V supply.

## 5.8 Summary

In this chapter, a new architecture for implementation of high speed switched-capacitor transversal filters was presented. Architectures for both decimation filters and single-rate filters were presented. The architectures exploit the feedforward-only nature of FIR filters. Since there is no feedback, parallelism is used to overcome the speed limitation of conventional switched-capacitor filters which is amplifier settling time. Due to the parallel architecture, amplifiers are allowed multiple output periods to settle.

The effect of non-idealities of the new architectures were then examined. Effects such as voltage offsets, mismatch in parallel filter frequency responses, sampling offset and jitter, and d.c. gain offsets were examined. Only the voltage offsets result in an output noise independent of the input

signal. The other effects result in sidebands about multiples of the channel periodicity or an increase in the noise floor of the system. Both of these effects depend on the input signal frequency.

A proposal was made for a method of making the switched-capacitor FIR filter adaptive based on a two step approach for realization of the variable tap weights. Since the tap weights are realized in two steps, the approach does require large capacitor arrays and contain the associated parasitic capacitance otherwise needed to get a similar range and granularity. Finally, the block level architecture of a digital adaptive equalizer/Viterbi detector architecture was presented. An architecture similar to that proposed for the SC transversal filters has been used in this chip to obtain high throughput at a relatively low power consumption. In Chapters 6 and 8, we shall examine practical implementation issues associated with the proposed architecture for high speed SC FIR filters. In Chapter 8, an extensive design overview of a prototype demonstrating many of the proposed techniques will be presented.

## References

- [1] R. G. Gregorian and W. E. Nicholson, "Switched-Capacitor Decimation and Interpolation Circuits," *IEEE Transactions on Circuits and Systems*, Vol. CAS-27, No. 6, June 1980, pp. 509-514.
- [2] Y.-S. Lee and K. W. Martin, "A Switched-Capacitor Realization of Multiple FIR Filters on a Single Chip," *IEEE Journal of Solid-State Circuits*, Vol. 23, No. 2, April 1988, pp. 536-542.
- [3] B. K. Ahuja, M. A. Copeland, and C. H. Chan, "A Sampled Analog MOS LSI Adaptive Filter," *IEEE J. Solid-State Circuits*, vol. SC-14, no. 1, pp. 148-154, Feb. 1979.
- [4] R.P. Martins and J.E. Franca, "A 2.4  $\mu\text{m}$  CMOS Switched-Capacitor Video Decimator with Sampling Rate Reduction from 40.5 MHz to 13.5 MHz," *Proceedings IEEE Custom Integrated Circuits Conference*, San Diego, CA, May 1989.
- [5] J. E. Franca, "Nonrecursive Polyphase Switched-Capacitor Decimators and Interpolators," *IEEE Trans. Circuits and Systems*, vol. CAS-32, no. 9, pp.877-887, Sept. 1985.
- [6] R. Gomez, M. Rofougaran, and A. A. Abidi, "A Discrete-Time Analog Signal Processor for Disk Read Channels," *ISSCC 1993 Digest of Technical Papers*, vol. 36, pp. 212-213, Feb. 1993.
- [7] A. Petraglia and S. K. Mitra, "Effects of Coefficient Inaccuracy in Switched-Capacitor Transversal Filters," *IEEE Trans. Circuits and Systems*, vol. 38, no. 9, pp. 977-983, Sept. 1991.
- [8] V. Gopinathan, Y. P. Tsividis, K.-S. Tan, and R. K. Hester, "A 5V 7th-order Elliptic Analog Filter for Digital Video Applications," *ISSCC 1990 Digest of Technical Papers*, vol. 33, pp. 208-209, Feb. 1990.
- [9] C. A. Laber and P. R. Gray, "A 20-MHz Sixth-Order BiCMOS Parasitic-Insensitive Continuous-Time Filter and Second-Order Equalizer Optimized for Disk Drive Channels," *IEEE Journal of Solid State Circuits*, Vol. 28, No. 4, pp. 462-470, April 1993.
- [10] W. C. Black, "High Speed CMOS A/D Conversion Techniques," *Ph.D. Dissertation*, Memorandum No. UCB/ERL M80/54, Nov. 1980.
- [11] C. S. G. Conroy, D. W. Cline, and P. R. Gray, "An 8-b 85-MS/s Parallel Pipeline A/D Converter in 1- $\mu\text{m}$  CMOS," *IEEE Journal of Solid State Circuits*, Vol. 28, No. 4, pp. 447-454, April 1993.
- [12] B. Kim, D. N. Helman, and P. R. Gray, "A 30 MHz High-Speed Analog/Digital PLL in 2  $\mu\text{m}$  CMOS," *IEEE Journal of Solid State Circuits*, vol. 25, no. 6, pp. 1385-1394, Dec. 1990.
- [13] Cormac Conroy, *Ph.D. Dissertation, Electronics Research Laboratory, UC Berkeley*, to be published fall of 1993.
- [14] M. J. M. Pelgrom, A. C. J. Duinmaijer, A. P. G. Welbers, "Matching Properties of MOS Transistors," *IEEE Journal of Solid-State Circuits*, Vol. 24, No. 5, pp. 1433-1439, October 1989.
- [15] M. L. Honig and D. G. Messerschmitt, *Adaptive Filters: Structures, Algorithms, and Applications*, Kluwar Academic Publishers, Boston, 1984.

- 
- [16] J. Khoury, "Design of a 15-MHz CMOS Continuous-Time Filter with On-Chip Tuning," *IEEE Transactions on Solid-State Circuits*, Vol. 26, No. 12, pp. 1988-1997, December 1991.
- [17] L. R. Rabiner and B. Gold, *Theory and Application of Digital Signal Processing*, Prentice-Hall, Inc., 1975.
- [18] L. B. Jackson, *Digital Filters and Signal Processing*, Kluwer Academic Publishers, 1986.
- [19] B. Kim, "High Speed Clock Recovery in VLSI Using Hybrid Analog/Digital Techniques," *Ph.D. Thesis*, University of California at Berkeley, ERL Memorandum No. UCB/ERL M90/50, June 1990.
- [20] G. T. Uehara and P. R. Gray, "Practical Aspects of High Speed Switched-Capacitor Decimation Filter Implementation," *1992 IEEE International Symposium on Circuits and Systems*, San Diego, pp. 2308-2312, May 1992.
- [21] C. E. Cook and H. S. Marsh, "An Introduction to Spread Spectrum," *IEEE Communications Magazine*, March 1983.
- [22] T. Sakurai and K. Tamaru, "Simple Formulas for Two- and Three-Dimensional Capacitances," *IEEE Transactions on Electron Devices*, Vol. ED-30, no. 2, pp. 183-185, February 1983.
- [23] M. I. Elmasry, "Capacitance Calculations in MOSFET VLSI," *IEEE Electron Device Letters*, Vol. EDL-3, No. 1, pp. 6-7, January 1982.
- [24] P. R. Gray and R. G. Meyer, *Analysis and Design of Analog Integrated Circuits, Second Edition*, John Wiley & Sons, 1984.
- [25] H. Takatori, T. Suzuki, F. Fujii, and M. Ogawa, "A CMOS Line Equalizer for a Digital Subscriber Loop," *Journal of Solid-State Circuits*, Vol. SC-19, No. 6., pp. 906-912, December 1984.
- [26] A. P. Chandrakasan, S. Sheng, R. W. Brodersen, "Low Power CMOS Digital Design," *IEEE Journal of Solid State Circuits*, Vol. 27, No. 4, pp. 473-484, April 1992.
- [27] C. S. H. Wong, Masters Thesis (to be published in the fall of 1993).
- [28] J. C. Rudell, Masters Thesis (to be published in the fall of 1993).



## CHAPTER 6

# High Speed Analog Multiphase Sampler and Summer

---

### 6.0 Introduction

One of the key functions required in the high speed SC decimation filter architecture is the front-end which samples the continuous-time input signal at the high sampling rate prior to decimation. The key elements which contribute to the functionality of this block are shown in Figure 6.1 and includes the interleaved switched-capacitor sampling networks, the zero-order hold circuitry, the operational amplifier, and the clock generators which generate the high sampling rate clocks along with the non-overlapping clocks important in switched-capacitor applications. Due to the large number of sampling clocks necessary and the stringent requirements for low phase jitter, all the clocks must be generated on-chip. The simplified schematic shown in the figure is for a 3-tap decimation filter such as the one used as the lowpass filter in the prototype.

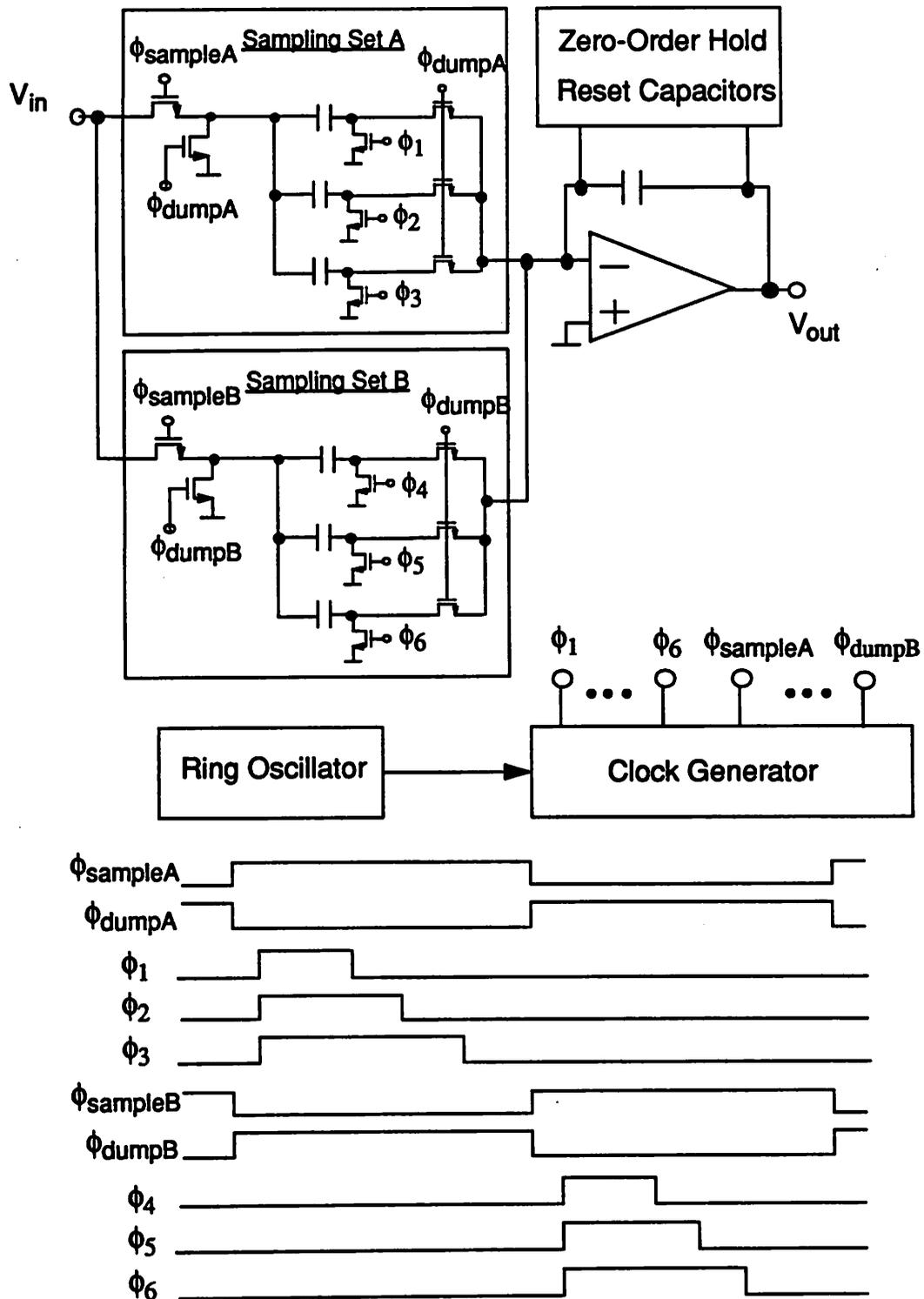


Figure 6.1 Block diagram of key functions required to realize high sampling rate front-end.

In this chapter, we shall examine issues relevant toward the implementation of these functions and present the techniques used in the prototype. We will begin with an examination of the switched-capacitor sample-and-hold with multiple inputs. This will be followed by a description of a switched-capacitor technique to realize a zero-order hold block and the associated non-idealities. In the following section, we will present key issues for high speed operational amplifier design and examine two transconductance amplifier approaches. Then we will examine issues pertinent to the generation of multiphase clocks using outputs off of a ring oscillator in order to create low jitter sampling phases. And finally, we shall present a new method for the generation of non-overlapping clocks and place this block in different scenarios.

## 6.1 Summing Zero-Order Sample-and-Hold

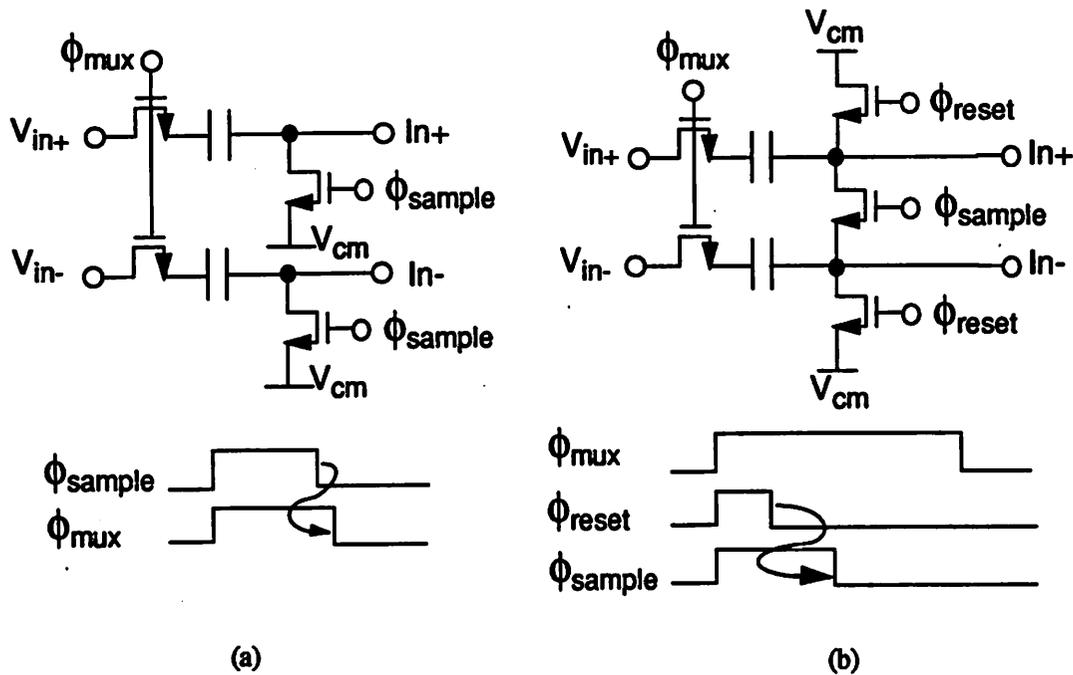
The motivation for this block in the context of high speed SC transversal filters is discussed in Chapter 5. The key requirement of this circuit is to sum the accumulated charge on a capacitor array corresponding to input samples, each of which has been scaled by ratioing capacitors in order to realize the tap weights of a transversal filter impulse response. A straight forward implementation of a circuit providing this function was shown in Figure 6.1. The sampled voltages result in a total charge that is dumped to the summing node of an opamp with a capacitor connected in a negative feedback configuration. The voltage at the output of this circuit once the amplifier settles is the total input charge divided by the value of the feedback capacitor. After every clock period, the voltage across the integrating (opamp feedback) capacitor must be reset or initialized. Using the simplest clocking scheme, the opamp is allowed one half the output period to settle since the other half period is required to reset or initialize the circuit. At the high output rates required, this is a big penalty because amplifier settling time can limit the achievable speed.

This function can be broken into two sections. The front-end sampling and summing portion, and the zero-order hold portion. We will discuss these separately in the next two sections.

### 6.1.1 Sampling and Summing Network

The sampling network shown in Figure 6.1 is simply a conceptual view. The sampling network in practice needs to be fully differential with a very wide bandwidth and charge injection and

clock feedthrough as a result of sampling must be kept to a minimum. Figure 6.2 a shows a simple sampling network and a more complicated, but robust, fully differential network.



**Figure 6.2** Two switched-capacitor implementations of sampling networks. (a) Parallel single-ended samplers. (b) Center switch sampler with common-mode reset.

In the implementation shown in Figure 6.2a, the differential sampling network is comprised of two parallel single-ended sampling switch and capacitor sets. The network in Figure 6.2b is a bit more complicated and works as follows. The LO to HI transition of all three clocks is non-critical and may go HI at anytime. The first clock to change is  $\phi_{reset}$ . The purpose of the switches driven by  $\phi_{reset}$  are to set the common-mode voltage of  $In+$  and  $In-$ . Since  $In+$  and  $In-$  connect either through pass gates or directly to the summing nodes of a differential opamp which is a high impedance, the switches controlled by  $\phi_{reset}$  set the common-mode voltage on those nodes by initializing these nodes to  $V_{cm}$  which is the input common-mode level. These switches may be small. Their purpose is to charge nodes  $In+$  and  $In-$  to the common-mode voltage during the duration that  $\phi_{reset}$  is HI. Clock signal  $\phi_{sample}$  remains ON during the HI to LO transition of  $\phi_{reset}$  resulting in the  $\phi_{sample}$  switch performing two roles. First, it equilibrates any charge differential between nodes

In+ and In- as a result of unequal charge injection by the reset switches. Second, it becomes the sampling switch. While  $\phi_{\text{sample}}$  is HI, the *center* switch it controls defines the sampling bandwidth. Since the midpoint of the center switch is an a.c. ground to differential input voltages, each sampling capacitor sees a resistance of one-half the on-resistance of the sampling switch to ground. Therefore, for a given center switch size, the effective on-resistance of this switch for differential input signals is one-half the total on-resistance of the switch. This allows the use of a smaller switch for a given sampling bandwidth than otherwise possible if two of the sampling networks in Figure 6.2a are used in parallel to realize a differential configuration. Finally, as  $\phi_{\text{sample}}$  makes its HI to LO transition and the center switch is turned off, the sampling instant is defined. It has been shown that this center switch sampling technique may result in less differential offset since smaller switches may be used and the sampling instant is defined by the turning off of just one device [1].

The overhead of this sampling approach is a bit more complicated layout and the requirement of an extra clock ( $\phi_{\text{reset}}$ ). However, in Chapter 8, we shall see that the added complexity is small for the resultant reduction in differential charge injection which is important because of the number of sampling switches that are placed in parallel in the SC FIR architecture.

### Realization of Negative Tap Weights

Changing the polarity of an effective tap weight in the differential structures just described simply entails switching the polarity of *either* pair of lines into or out of the sampling networks of Figure 6.2. Programmable tap weights which require the possibility of both polarities require a multiplexor to choose between polarities.

## 6.1.2 Zero-Order Hold Circuit

Figure 6.3 shows a single-ended integrator and its switched-capacitor equivalent. This circuit is the basic building block of switched-capacitor filters. The input is sampled on capacitor  $C_{\text{in}}$  during  $\phi_1$  and dumped to the non-inverting input of the opamp during  $\phi_2$ . The output voltage at time  $t_n$  is equal to

$$V_o[t_n] = V_o[t_n - T] + V_{in}\left[t_n - \frac{T}{2}\right] \cdot \frac{C_{in}}{C_{fb}} \quad (6.25)$$

where  $T$  is the period of the master clock.

In this circuit, the amplifier input is applied during  $\phi_1$  and has the duration of  $\phi_2$  to slew and settle to its final value. Assuming a 50% duty cycle clock, the amplifier has one-half clock period to settle. The feedback capacitor  $C_{fb}$  is always present and is reset during  $\phi_1$ . The reset phase is important in order to affect a sample-and-hold and not an integrator.

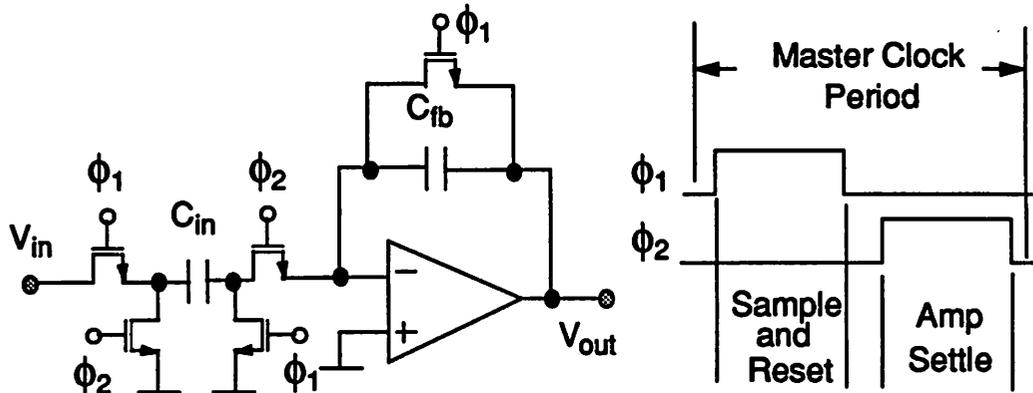


Figure 6.3 Two phases of a simple switched-capacitor sample-and-hold.

The key drawback of the use of this circuit in the high speed switched-capacitor transversal filter architecture is that the opamp is allowed only one-half of the master clock period for settling. The other half period is used to initialize the feedback capacitor and therefore no output can be generated by the amplifier during this time, although power is still consumed by the opamp. The sampling capacitor array has two main modes of operation, sampling and dumping (evaluation). In order to accommodate the two interleaved capacitor arrays as shown in Figure 6.1 another approach is used which allows two outputs to be generated during one master clock period. In this other approach, the output voltage after settling (which is the voltage across the feedback capacitor) is sampled onto a capacitor of equal value to the feedback capacitor. During the next clock phase, this capacitor is switched back to the summing node with an appropriate polarity so that it contributes the negative of the last output to the current output. As a result, the integrating effect of the feedback capacitor is removed. Thus, the circuit operates as a zero-order sample-and-hold. Let

us now examine the circuit used for this purpose and study the non-ideal effects associated with this approach

### 6.1.3 Switched-Capacitor Summing Circuit with On-the-Fly Reset

The output of a zero-order sample-and-hold circuit changes with the sampling rate as shown in Figure 6.4. The output is the input at the sampling instant held for the duration of the sampling period.

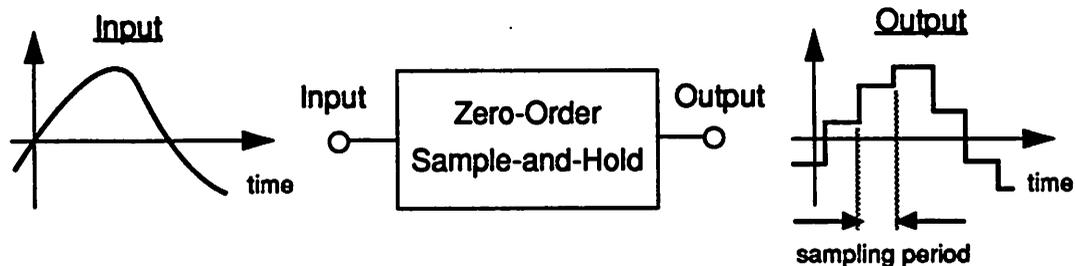
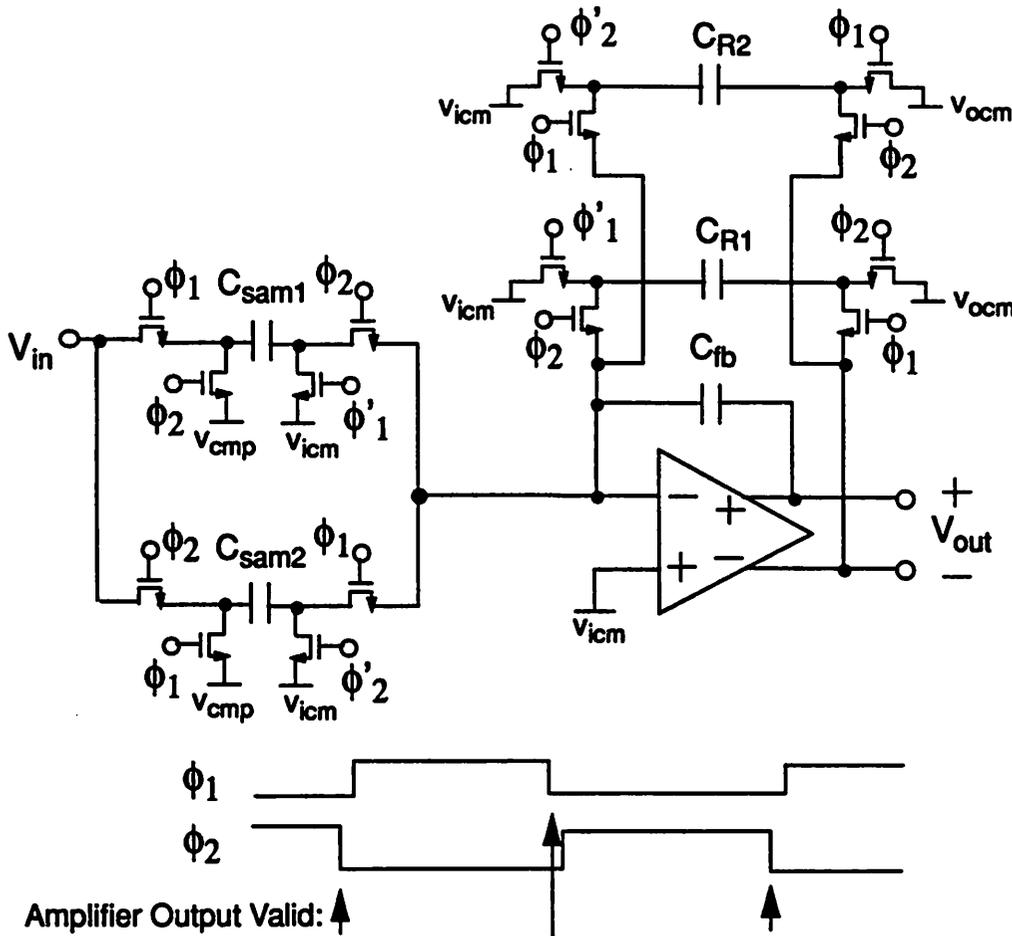


Figure 6.4 Zero-Order Sample-and-Hold.

A conceptual schematic of the circuit used to perform the summing operation was shown in Figure 6.1. Capacitor Arrays A and B operate in the interleaved manner described earlier. Two sample-and-hold circuits which also operate in an interleaved manner appear in the feedback path. Each of the feedback sample-and-hold circuits alternately samples the present output on capacitors of value equal to the integrating capacitors, and then adds the negative of this sample to the output during the following period. This has the effect of removing the period-to-period memory effects of the integrating capacitors. By interleaving two of these circuits, one operates in the sample mode while the other in the output mode. If Capacitor Arrays A and B are single capacitors as in the case of a sample-and-hold, this circuit would operate as a zero-order hold.

A single-ended version (shown for simplicity) of a switched capacitor implementation of this circuit is shown in Figure 6.5. Note in the figure that a differential amplifier is shown in the single-ended schematic. It turns out that in order for the polarity to be correct, the negative of the output voltage must be used which is obtained by sampling the negative output of a differential amplifier. This is not a problem in the actual implementation since a fully differential implementation is used. The reset capacitors are equal in value to the feedback capacitor  $C_{fb}$  and are shown as

$C_{R1}$  and  $C_{R2}$ . During  $\phi_1$ ,  $C_{R1}$  samples the negative of the output voltage. Note that the output voltage appears across  $C_{FB}$  since the opamp input is a virtual ground. Then in  $\phi_2$ , the polarity is such that  $C_{R1}$  contributes a charge to the summing node to cancel off the charge due to the previous output voltage. Capacitor  $C_{R2}$  simply operates in the opposite phases.



The prime superscript (') denotes an early HI to LO transition.  
 $V_{icm}$  is the input common-mode voltage of the opamp.  
 $V_{ocm}$  is the output common-mode voltage of the opamp.  
 $V_{cmp}$  is the output common-mode voltage of the previous stage.

Figure 6.5 Single-ended example of the zero-order sample-and-hold.

The major advantage of this circuit is that the opamp can be used to evaluate a sum every clock cycle, i.e., no reset period is required. The added complexity does however reduce the

feedback factor of the opamp and thus affect the gain requirements and settling behavior, as well as introduce other non-ideal effects. We shall examine these issues in the following sections.

### 6.1.4 Non-idealities of SC Summer

There are two primary sources of error in the switched-capacitor summer previously described. They are capacitor mismatch and offsets in the charge injection and clock feedthrough from the switches. Let us first examine the effect of capacitor mismatch.

There are two primary concerns due to the mismatch in capacitors  $C_{fb}$ ,  $C_{R1}$ , and  $C_{R2}$ . First, there can be a stability issue if the mismatch is sufficiently large. Second, there will be a contribution from the previous output to the current output. In this way, the sample-and-hold looks like a lossy integrator. We will examine both of these issues in this section.

In the absence of capacitor mismatch and assuming a large opamp open loop gain, the output voltage can be written as

$$V_o[n] = V_{in}\left[n - \frac{1}{2}\right] \cdot \frac{C_{sam}}{C_{fb}} - \left( V_o[n-1] + \left( V_o[n-1] \cdot \frac{C_{R1}}{C_{fb}} \right) \right) \quad (6.2)$$

So if  $C_{R1}=C_{fb}$ , the clocking of  $C_{R1}$  contributes in such a way as to remove the previous output.

Suppose that due to capacitor mismatch there is the following relationship between  $C_{fb}$  and  $C_{R1}$ :

$$\Delta C = C_{R1} - C_{fb} \quad (6.3)$$

where  $\Delta C$  is a mismatch term. The output during  $\phi_2$  can be written

$$V_o[n] = V_{in}\left[n - \frac{1}{2}\right] \cdot \frac{C_{sam}}{C_{fb}} + V_o[n-1] \cdot \frac{\Delta C}{C_{fb}} \quad (6.4)$$

The capacitor mismatch contributes a small error term to the output.

We can view this system with the imperfect cancellation due to capacitor mismatch as a simple discrete-time feedback system as shown in Figure 6.6a. It can be easily shown that for a bounded input, the output is bounded and thus the system is stable if the magnitude of  $\Delta C/C_{fb}$  is less than unity which it surely should be. Converting Eq. 6.4 to the z-domain we have

$$V_o[z] = V_{in}[z] \cdot \frac{C_{sam}}{C_{fb}} \cdot z^{-1/2} + V_o[z] \cdot \frac{\Delta C}{C_{fb}} \cdot z^{-1} \quad (6.5)$$

The transfer function is thus

$$\frac{V_o[z]}{V_{in}[z]} = \frac{z^{1/2} \cdot \frac{C_{sam}}{C_{fb}}}{z - \frac{\Delta C}{C_{fb}}} \quad (6.6)$$

The resulting pole-zero plot is shown in Figure 6.6b where there is a zero at 0 and a real pole at  $-\Delta C/C_{fb}$ . In the case where  $\Delta C$  is zero, the pole and zero cancel and all that is left is a phase term due to the half period delay from input to output. However, if it is non-zero and positive, the result is a pole at  $\Delta C/C_{fb}$  which results in an overall response that is slightly high-pass in nature where the high frequency magnitude response is slightly larger than at low frequencies. The opposite happens if  $\Delta C$  is non-zero and negative with a response that is somewhat lowpass in nature. Even for capacitor matching of 1%, the difference between the low and high frequency response should be less than 0.1dB.

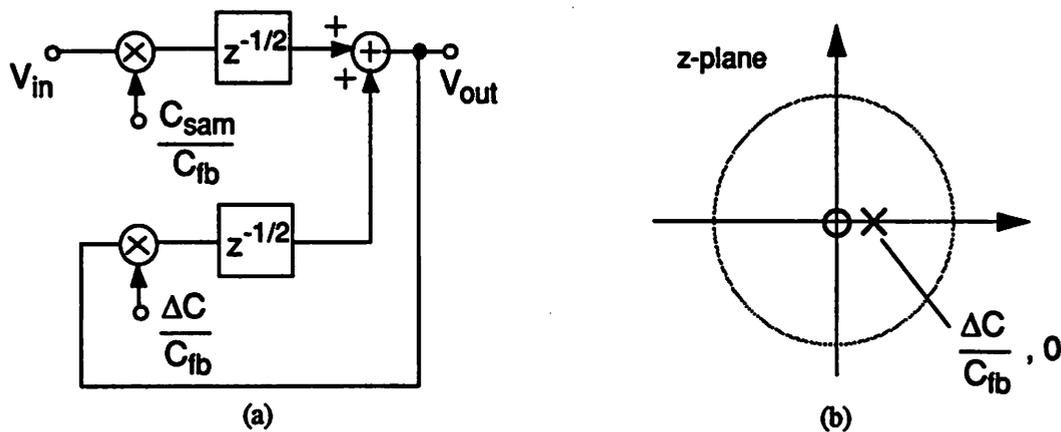


Figure 6.6 (a) Block diagram of sample-and-hold with mismatch in zeroing capacitors. (b) Resulting z-plane pole-zero diagram.

Since there are two sets of reset capacitors, the  $\Delta C$  will actually alternate between two values. Although the stability will not be affected, the effect on the performance of the filter will be similar to that described in Chapter 5 for frequency response variations in the parallel circuit blocks.

There are many extra switches connected to the summing nodes of the opamp as a result of the zero-order hold circuitry. Bottom plate sampling techniques can be used with sampling capacitors  $C_{R1}$  and  $C_{R2}$  to eliminate the effects of signal dependent charge injection. However, voltage offsets will still result due to device mismatches resulting in two different input-referred offset voltages corresponding to each set of reset capacitors and switches. A fully differential approach with careful layout and switching must be used to keep the effect of these offsets at a tolerable level.

## 6.2 Operational Transconductance Amplifier (OTA)

The operational amplifier is perhaps the most important circuit block in analog IC design. It is a key component in most switched-capacitor circuits such as sample-and-holds as in the present situation, as well as A/D converters and switched-capacitor filters. It alone often sets the allowable performance limitations of these circuits. In this section we will discuss the key specifications of the opamp and focus on the important trade-offs for achieving opamps with fast settling times. We will begin with a discussion motivating the use of transconductance amplifiers over voltage amplifiers for high speed switched-capacitor applications. Then, we will examine performance specifications such as gain and settling time. Following this, folded cascode and telescopic OTA architectures will be discussed. Finally, we shall examine a composite OTA approach with a wide-band preamp preceding a telescopic OTA and consider some important advantages for this architecture in high speed switched-capacitor circuit applications.

### 6.2.1 Advantages Over Voltage Amplifiers

In switched-capacitor circuits, the high gain operational amplifiers needs only drive loads that are capacitive (no resistors). It is very attractive in these applications to use operational transconductance amplifiers (OTAs) as opposed to traditional two stage opamps. In this section, we shall briefly examine the advantages of the OTA for driving capacitive-only loads. We will then examine the telescopic implementation of the cascode OTA and follow this with advantages of using a preamp in front of a telescopic OTA in high speed applications.

An often used block in switched-capacitor circuit design is the capacitive feedback amplifier configuration is shown in Figure 6.7 with sampling capacitor  $C_{sam}$ , feedback capacitor  $C_{fb}$ , load capacitor  $C_{load}$ , and the total summing node shunt capacitance  $C_p$ . The amplifier is often a voltage amplifier with voltage gain  $A$  and both the input and output variables are voltages. We shall assume that the input signal is a step input as encountered in SC circuits performing discrete-time signal processing. The necessary switches are not shown and their on-resistance is assumed to be negligible which in most cases is a good assumption.

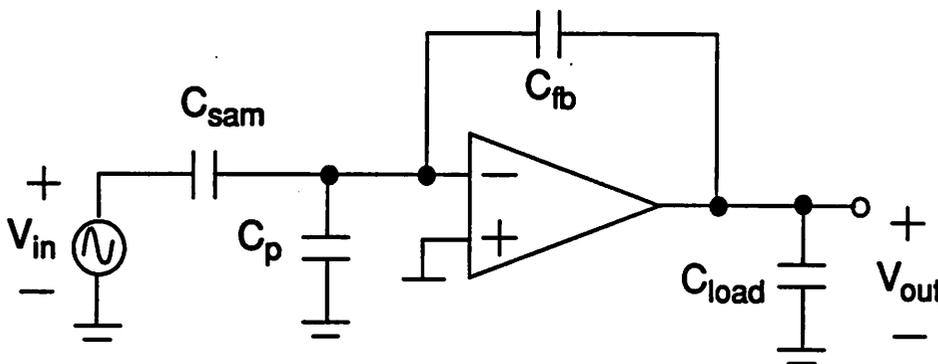


Figure 6.7 Basic capacitive feedback amplifier found in analog discrete-time MOS circuits (switches are not shown).

There are many advantages to using an operational transconductance amplifier where the output variable is a current as opposed to a voltage in these applications [3]. Models of voltage both a voltage amplifier and a transconductance amplifier are compared in Figure 6.8.

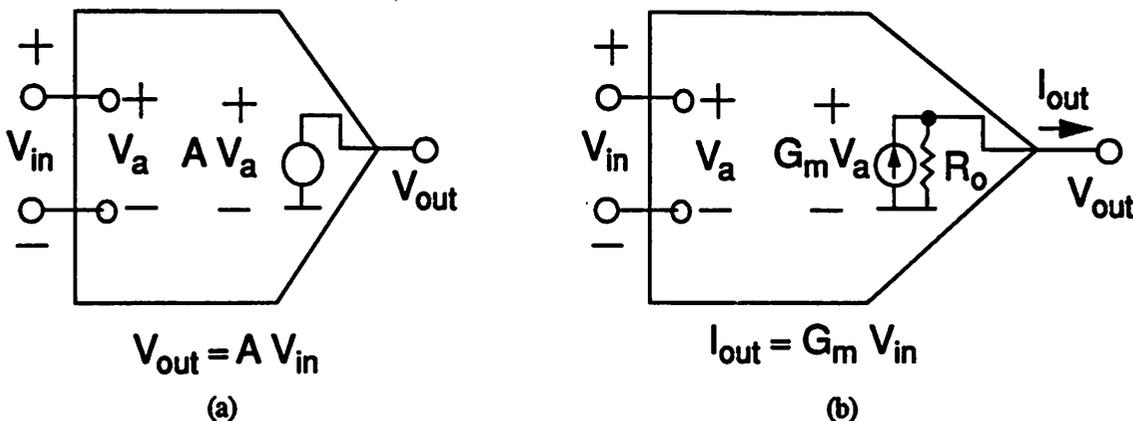


Figure 6.8 Comparison between voltage and transconductance amplifiers. (a) Voltage amplifier. (b) Transconductance amplifier.

The fundamental difference between the two amplifiers is that the voltage amplifier output is a voltage source with value  $A$  times the differential input, whereas the transconductance amplifier

output is a current with magnitude equal to the input voltage times the transconductance. However, if the output of the OTA is driving a capacitive-only load, then at d.c. the open loop amplifier exhibits a voltage gain equal to  $G_m R_o$  where  $R_o$  is the small signal output resistance of the voltage controlled current source. Thus, in a capacitive feedback circuit as shown in Figure 6.7, an OTA appears as a voltage amplifier with gain  $A$  equal to  $G_m R_o$ . Clearly, this is valid only for capacitive feedback and capacitive loads because any equivalent finite shunt output resistance appears in parallel with  $R_o$  and will reduce the effective open loop gain.

### 6.2.2 Feedback Factor Effect on Amplifier Performance

One of the most important characteristics of a circuit in its closed loop configuration is the *feedback factor*. This factor as we shall see impacts the required opamp gain, the output offset voltage and the circuit settling time. The feedback factor describes how much of the output signal is fed back to the feedback point. The loop at the point of feedback is broken and the feedback factor is the impedance back to the feedback point divided by the total impedance as shown in Figure 6.9a. The feedback factor for the circuit in Figure 6.7 is shown in Figure 6.9b. The input capacitance  $C_{in}$  looking into the amplifier must be included in the calculation of the feedback factor even though the loop is broken at that point.

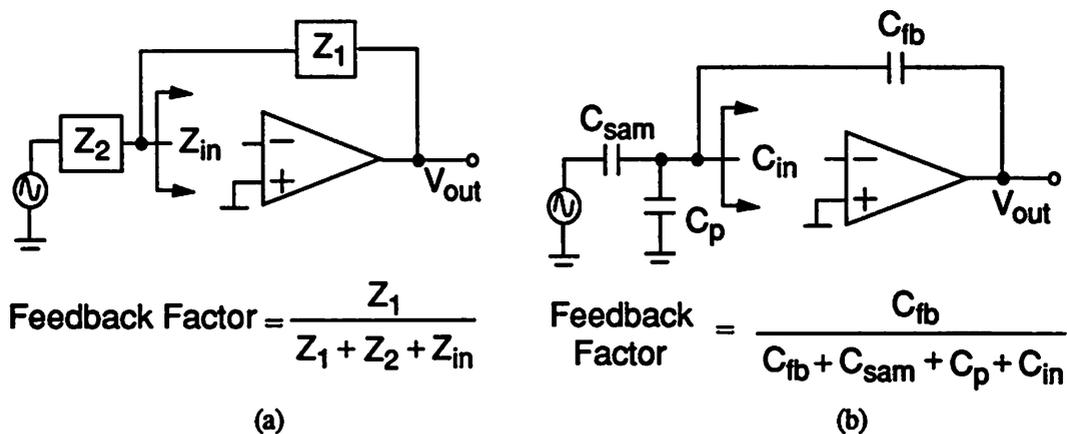


Figure 6.9 Breaking the loop in a closed-loop amplifier for the calculation of the feedback factor.

(a) General case. (b) Capacitive feedback amplifier case.

### 6.2.2.1 Settling Behavior

Let us now consider the settling behavior of the OTA in the circuit of Figure 6.8b to a voltage step at the input. In high speed circuits, the high currents required for high speed operation and the resultant large  $V_{dsat}$ s results in circuits that do not usually spend much time if any in slew rate limited conditions. Therefore, the settling behavior is dominated by small signal settling and can be analyzed and characterized using a linear model for settling.

We will begin by considering the output resistance of the OTA in a closed-loop gain configuration. In Figure 6.10, an OTA is shown with the corresponding small signal output resistance labelled near the amplifier as  $R_o$ . Let  $f$  be the feedback factor. The output impedance is increased over that of a transconductance amplifier in unity gain by a factor of  $1/f$ . For the circuit shown in Figure 6.10, the output resistance is

$$R_o = \frac{1}{G_m} \left[ \frac{C_{fb} + C_{sam} + C_p + C_{in}}{C_{fb}} \right]. \quad (6.7)$$

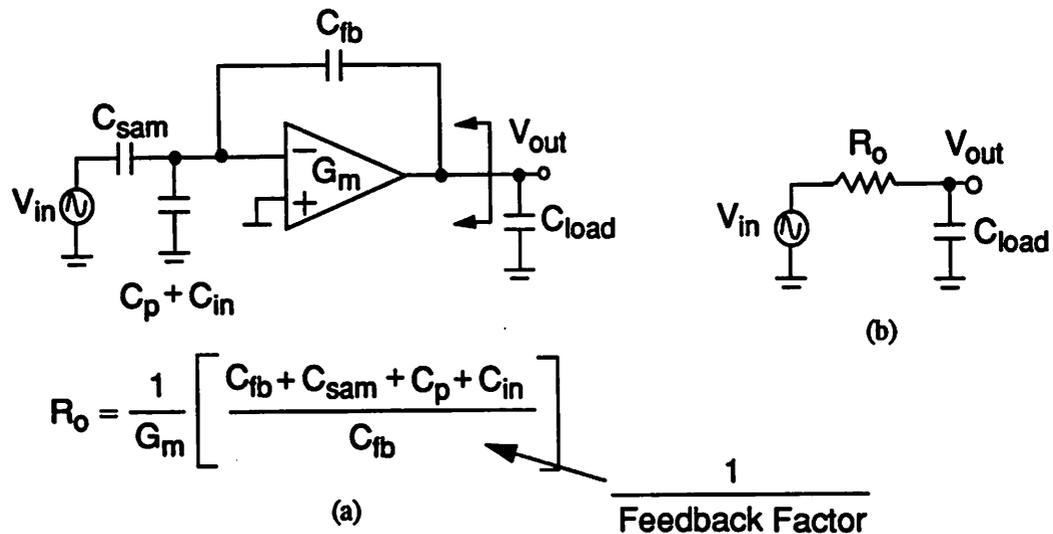
In unity gain, the output resistance is simply  $1/G_m$ .

If we neglect the effect of non-dominant poles in the amplifier, the OTA driving the load capacitor appears as a Thevenin equivalent voltage source driving the capacitive load through a resistor of value equal to the closed-loop small signal output impedance. As a result, the circuit of Figure 6.10 settles with a time constant of value

$$\tau_{settling} = R_o \cdot C_{load} \quad (6.8)$$

$$= \frac{1}{G_m} \left[ \frac{C_{fb} + C_{sam} + C_p + C_{in}}{C_{fb}} \right] \cdot C_{load} \quad (6.9)$$

As the feedback factor decreases, particularly due to an increase in  $C_p$  or  $C_{in}$ , the time constant of the settling time increases.



**Figure 6.10** Equivalent circuit for calculation of the time constant for OTA settling. (a) Circuit for calculation of output impedance. (b) Thevenin equivalent circuit.

### 6.2.2.2 Open Loop Gain

The closed loop gain from the input to the output including the finite opamp gain  $A$  is

$$V_{out} = \frac{V_{in} \cdot C_{sam}}{C_{fb} + \frac{C_{sam} + C_{in} + C_p + C_{fb}}{A}} \quad (6.10)$$

In Eq. 6.10, the right hand term in the denominator is the term which accounts for the effect of finite opamp gain. This changes the overall gain from the ideal which would be  $C_{sam}/C_{fb}$ . In circuits such as A/D converters, the gain must in many cases be accurate to the resolution of the converter.

In the SC FIR filter architecture, finite opamp gain affects the filter performance in two important ways. First, it results in an overall d.c. gain error as just described. This in general is not a problem since it will not affect the frequency response of the filter. However, if  $A$  is sufficiently small, variations of  $A$  between opamps in parallel channels will cause what is effectively gain mis-

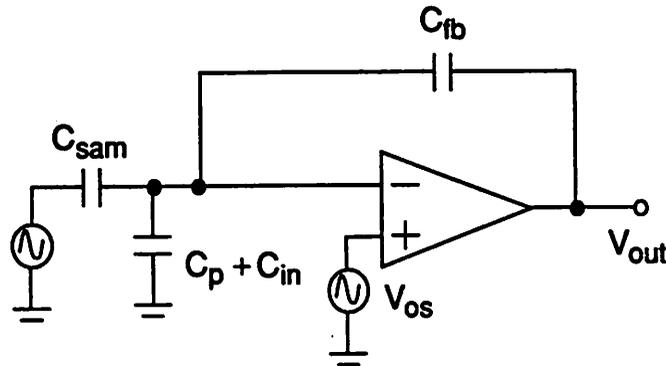
match in parallel channels which will impact the filter performance as discussed in Chapter 5. Therefore, the open loop gain must be large enough to insure sufficient gain matching between parallel channels. Second and more importantly is the effect of finite  $A$  on the zero-order hold circuitry. Rewriting Eq. 6.2 taking into account finite opamp gain, after some algebra we get

$$V_o[n] = V_{in}\left[n - \frac{1}{2}\right] \cdot \frac{C_{sam}}{C_{fb} + \frac{1}{fA}} - \left( V_o[n-1] \left( 1 - \frac{C_{R1}}{C_{fb} + \frac{1}{fA}} \right) \right). \quad (6.11)$$

The first term on the right side of Eq. 6.2 is the ideal input-output relationship for a sample-and-hold with finite  $A$ . The second term describes the cancellation of the previous output for the zero-order hold response. For  $A$  very large, the condition  $C_{R1}$  equal  $C_{fb}$  makes this term zero. However, with finite opamp gain, when the capacitors are equal this term is *not* zero. This will result in a small residue on the integrating capacitor from cycle to cycle and add a lowpass response to the transfer function of the filter. For a product of the feedback factor the amplifier gain ( $fA$ ) equal to 50, the result on the overall transfer function will be less than a .2dB droop from d.c. to one-half the output frequency of the filter. Once again, the effect of gain matching in parallel channels must also be considered.

### 6.2.2.3 Input-Referred Voltage Offsets

An input-referred voltage offset appears in the single-ended version of the sample-and-hold block as shown in Figure 6.11. This is the non-inverting amplifier circuit. Unfortunately, by definition, this circuit has a gain greater than one which means the input-referred voltage and noise will be amplified.



**Figure 6.11** Input-referred voltage offset appears as the input to a non-inverting amplifier configuration.

The gain of this circuit including  $C_p$  and  $C_{in}$  is given by

$$\frac{V_{out}}{V_{os}} = 1 + \frac{C_{sam} + C_p + C_{in} + C_{fb}}{C_{fb}}. \quad (6.12)$$

The input-referred offset is clearly enhanced by a factor equal to the inverse of the feedback factor or  $1/f$ . Input-referred noise is amplified in a similar way. Because of the use of large signal levels into and the modest dynamic range requirements of the filter, amplifier and device noise should not be of much consequence.

We have just seen how the feedback factor affects the settling time of the sample-and-hold, can increase the open loop gain requirements, and increases the effect of the input-referred offset voltage. Capacitors  $C_{sam}$  and  $C_{fb}$  control the closed loop gain while  $C_p$  and  $C_{in}$  are both parasitic capacitors and their presence clearly has undesirable consequences. Later in this section, we will see how a small feedback factor can allow the addition of a wideband preamp which increases the effective transconductance and open loop gain by taking advantage of the reduction in closed loop bandwidth.

### 6.2.3 Folded-Cascode Operational Transconductance Amplifier (OTA)

An important implementation of the OTA is the folded-cascode architecture [3] as shown in Figure 6.12. An input transconductance stage made up of transistors M1 and M2 drive common-gate amplifiers M9 and M10. The output is taken at the high impedance cascoded nodes as shown in the figure. Some advantages of the folded-cascode OTA include a large common-mode input range and large output swing possible with proper biasing of the cascode devices at  $V_{pbias2}$  and  $V_{nbias2}$

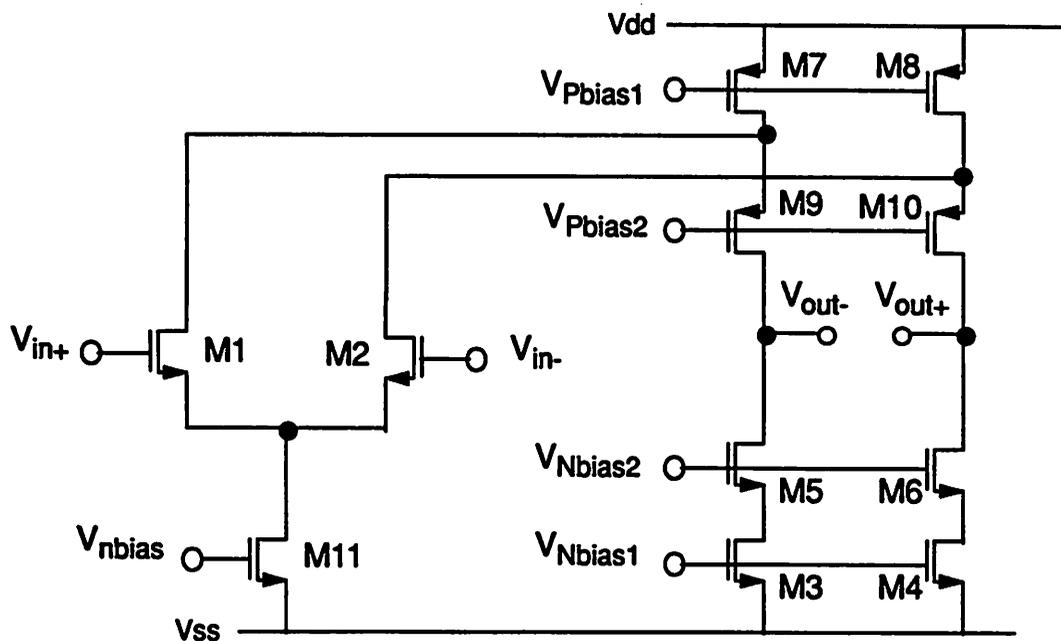


Figure 6.12 Folded-Cascode OTA with NMOS input stage.

The key limitation of the folded-cascode OTA is the trade-off between an NMOS (PMOS) input stage and a PMOS (NMOS) common-gate stage. With the NMOS input stage, the non-dominant pole of the PMOS common-gate device can be near the unity gain frequency of the open loop amplifier thus affecting the settling of the feedback amplifier. On the other hand, the OTA with the PMOS input stage has the NMOS common-gate device which pushes the non-dominant pole out relative to the unity gain frequency due to the transconductance ratios of the NMOS to the PMOS

devices. However, the transconductance of the PMOS device is inherently lower than that of the NMOS device. This trade-off does not exist in the telescopic OTA which we shall next examine.

### 6.2.4 Telescopic Operational Transconductance Amplifier (OTA)

A schematic of the telescopic OTA [2] is shown in Figure 6.13. It receives its name from its resemblance to a submarine telescope with the inputs as the handles of the telescope. This is a simple NMOS differential pair with cascode devices M3 and M4 and cascoded current sources M5 and M6. Not shown is a common-mode feedback circuit needed to bias the output common-mode voltage at the desired level.

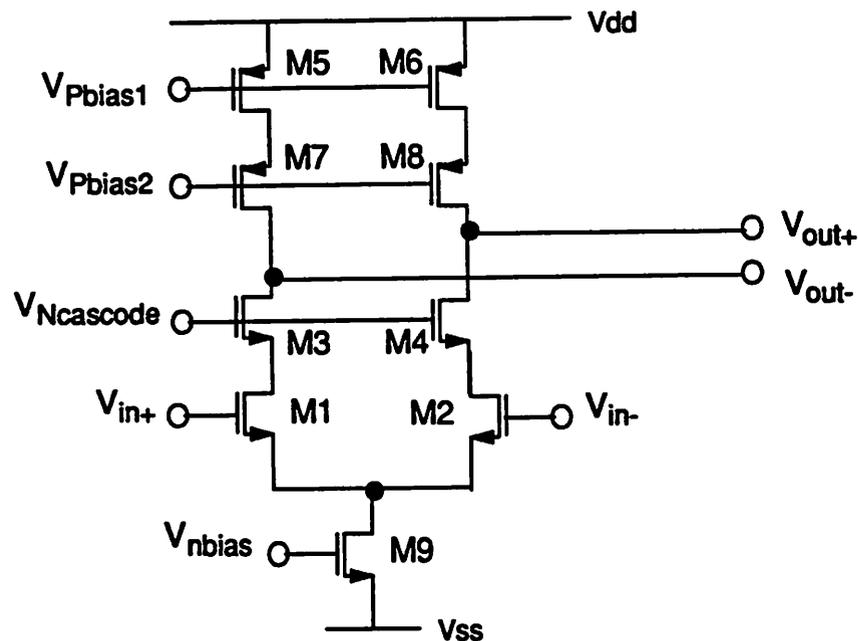


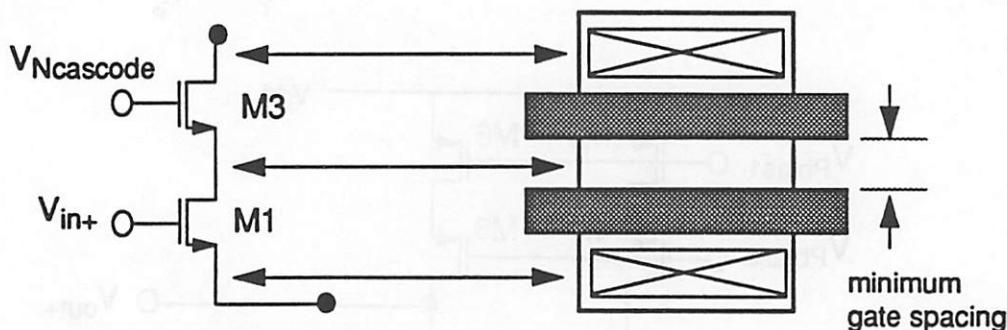
Figure 6.13 Schematic of a telescopic OTA.

#### 6.2.4.1 Speed and Voltage Swing

The telescopic OTA architecture is particularly advantageous from the standpoint of speed because in the implementation shown, it has NMOS input devices for large small signal transconductance and NMOS cascode devices which are the only other devices in the signal path. PMOS devices M5-M8 simply make up biasing current sources. As a result, the time constant of the non-dominant pole created by the impedance and capacitance at the drains of the input device pair is

$$\tau_{non-dominant} = \frac{1}{G_{m_3}} \cdot C_{shunt} \quad (6.13)$$

where  $C_{shunt}$  is the total shunt capacitance at the source of the cascode devices including  $C_{gs3}$ ,  $C_{s3}$ ,  $C_{d1}$ , and  $C_{gd1}$ . By making the widths of devices M1 and M3 the same, the two gates may be laid out in strips with minimum diffusion between the two gates as shown in Figure 6.14. This pushes out the non-dominant pole near the device  $f_t$  of the NMOS cascode devices.



**Figure 6.14** Schematic and layout of input and cascode devices of the telescopic OTA for minimum parasitic capacitance on the drain node of M1.

The telescopic OTA has the best of both folded cascode approaches: NMOS input devices for large transconductance and NMOS cascode (common-gate) devices for high frequency non-dominant poles. In applications where the feedback factor is small, the non-dominant pole can be sufficiently far out that the step response is that of a single pole system. Thus, the ringing that is often associated with second-order systems is not present. However, for fast settling, a second-order system can be designed to do better than one of first-order. We will see in the next section that the addition of a preamp reduces the phase margin by reintroducing a non-dominant pole and can result in increased overall performance including reduced settling time.

#### 6.2.4.2 Common-mode Input Range

The common-mode input range of the telescopic OTA implementation is limited. The  $V_{ds}$  of the tail current device M9 must be maintained because M9 determines the common-mode rejec-

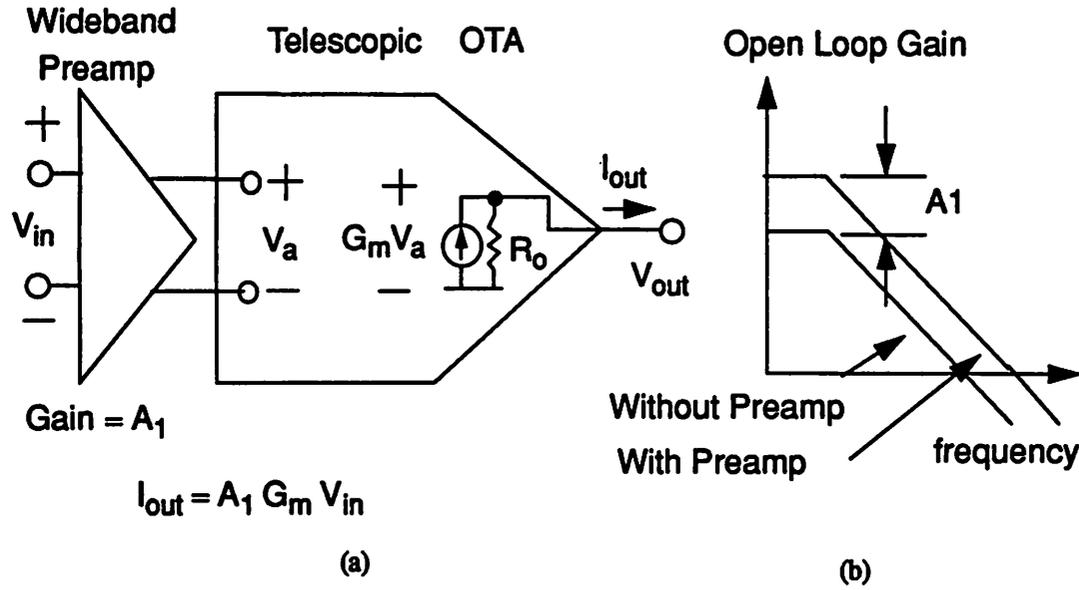
tion. The bias voltage of the cascode devices sets the drain voltage of the input devices while the input common-mode voltage sets the gate voltage. The  $V_{gs}$  of the input devices is determined by the sum of the threshold voltage and the  $V_{dsat}$ . As a result, the  $V_{ds}$  of the input devices depends on the cascode bias, the input common-mode voltage, the input device  $V_t$ , and the tail current which sets the  $V_{dsat}$ . We shall see later in the implementation section in Chapter 8 how replica biasing can be used to set the cascode bias and similarly, the common-mode input voltage. The common-mode input range is limited because as the common-mode input voltage is increased, the  $V_{ds}$  of the input devices is reduced which will eventually take these devices out of saturation. In the other direction, as the input common-mode voltage is reduced the  $V_{ds}$  of M9 is reduced and M9 is eventually taken out of saturation.

### 6.2.5 Telescopic OTA with Preamp

The feedback factor affects the closed loop bandwidth of a feedback system. In order for a feedback system to have its poles in the left half of the  $s$ -plane, the loop gain of the closed loop system with the loop broken must drop below unity before the phase shift around the loop reaches 180 degrees. The amount of margin of the phase shift from 180 degrees at the unity gain point is referred to as the *phase margin*. Each pole introduces a drop-off in the magnitude response of 20dB/decade starting at the pole location and a phase shift in the phase response of 45 degrees (from zero at d.c.) at the pole location, asymptotically approaching 90 degrees at higher frequencies. Therefore, in a two pole system with one pole at very low frequencies, in order to have 45 degrees of phase margin, the second pole must be at the unity gain frequency of the closed loop system.

In applications where the feedback factor is small (less than 1/4), the addition of a wideband preamp in front of a telescopic OTA may have important benefits. A conceptual schematic of the telescopic OTA with a wideband preamp with gain  $A_1$  is shown in Figure 6.15 together with a frequency response plot of the open loop gain with and without the preamp. As can be seen in the figure, the preamp has the desirable effect of increasing both the open loop gain and the bandwidth. The assumption is that the non-dominant pole of the telescopic OTA by itself is at a very high fre-

quency such that it is out of the picture. However, a new pole is introduced by the wideband amplifier with a location that is inversely proportional to the preamp gain. With a careful design trading off preamp gain with the location of the non-dominant pole, an amplifier with a larger d.c. gain and a faster settling time can be achieved.



**Figure 6.15** Block diagram and open loop gain of OTA with and without preamp. (a) OTA with preamp. (b) Open loop frequency response with and without the preamp.

The faster settling time comes about for two reasons. First, the effective transconductance is increased by the preamp gain. Second, the telescopic OTA by itself has a phase margin of nearly 90 degrees because of the distant location of the non-dominant pole. With the addition of the preamp, a new non-dominant pole created by the product of the resistance of the preamp load device and the total capacitance on the output node of the preamp (including input capacitance into the telescopic OTA) introduces phase shift which lowers the phase margin to something less than 90 degrees. The settling time as a function of phase margin is plotted qualitatively in Figure 6.16. The settling time is defined to be the time it takes for the closed loop system to settle within a certain percentage of the final value depending on the required accuracy of the system. Therefore, the plot of Figure 6.16 really depends on the resolution of the system. However, the qualitative trade-

off is clear. In order to minimize settling time, the preamp gain should be increased until the settling time begins to increase.

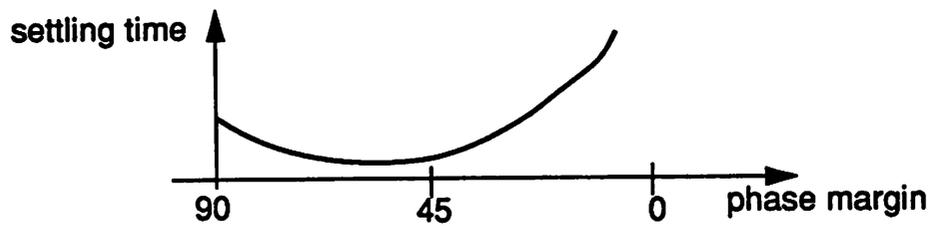


Figure 6.16 Qualitative plot showing settling time as a function of phase margin.

### Reduced Input Capacitance

The input capacitance of the amplifier appears as a shunt capacitance on the summing node of the amplifier and has the deleterious effect of reducing the feedback factor which increases the open loop gain requirements and slows the settling. The input capacitance of the telescopic OTA is the sum of the input device gate-to-source capacitance  $C_{gs}$  and the Miller multiplied  $C_{gd}$ . The effect of  $C_{gd}$  on the input capacitance as a function of frequency is considered in [4]. The half circuit required for calculation of the input capacitance is shown in Figure 6.17. Since the output

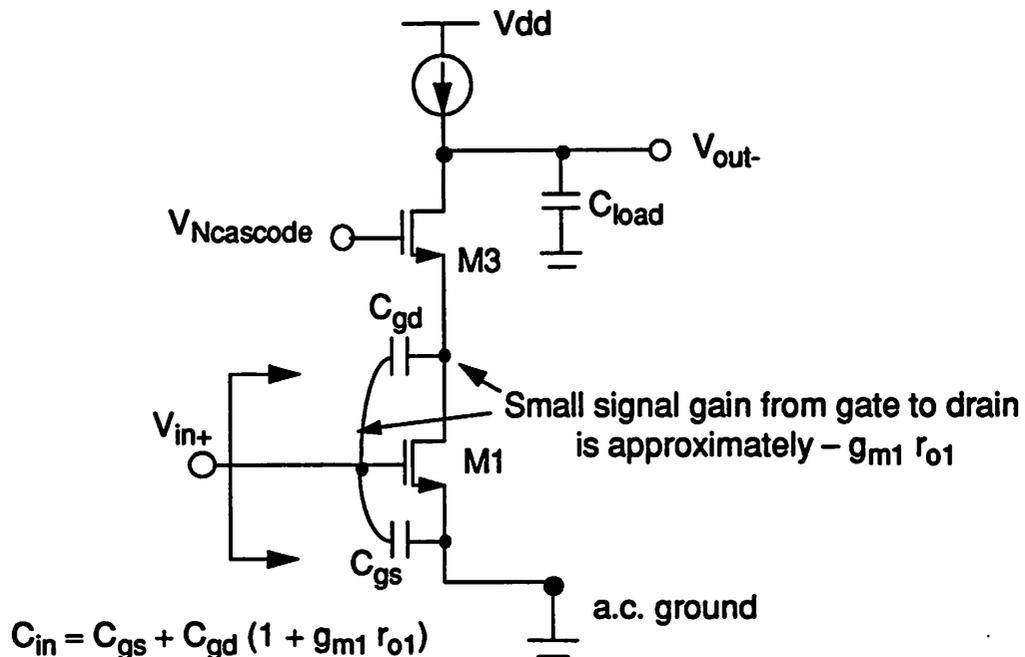


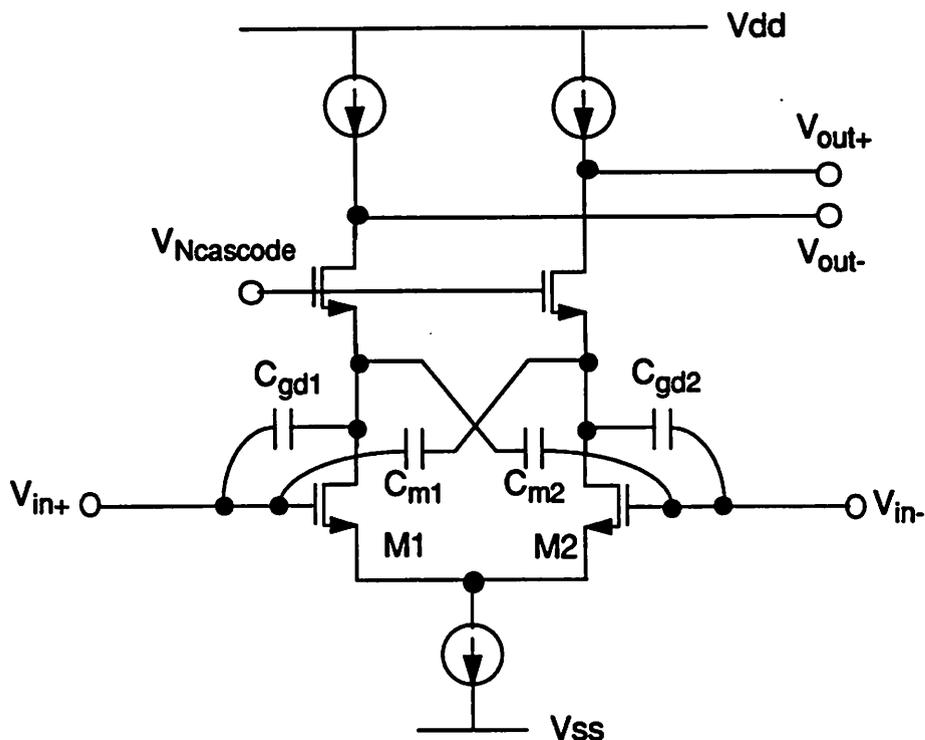
Figure 6.17 Half circuit for calculating low frequency input capacitance.

node is a high impedance node, the impedance looking up into the source of the cascode device M3 is large resulting in a small signal gain from the gate to the drain of the input device of approximately  $g_m r_o$  of the input device. This gain increases  $C_{gd}$  by the factor  $(1+g_m r_o)$ . Since  $C_{gd}$  is nominally between 10 and 20 percent of  $C_{gs}$  and  $g_m r_o$  for an NMOS device nominally around 30,  $C_{gd}$  at d.c. is roughly between 3 to 6 times  $C_{gs}$ . The equation for the closed-loop gain including the effect of the input capacitance can be written as

$$A_{closed-loop} = \frac{C_{in}}{C_{fb} + \frac{C_{in} + C_{fb} + C_p + C_{gs} + C_{gd} \cdot (1 + g_m r_o)}{A_{open-loop}}} \quad (6.14)$$

where  $C_p$  is the parasitic capacitance on the amplifier summing node and  $A_{open-loop}$  is the open loop gain of the amplifier. At high frequencies, the capacitance shunting the output node reduces the impedance at that node resulting in a reduction in the impedance looking into the source of M3. In the limit as this impedance gets very small, the impedance looking into the source of M3 approaches  $1/g_{m3}$  reducing the Miller multiplication factor. Thus, the feedback factor is small at low frequencies and increases at higher frequencies to an asymptotic value. This scenario puts the maximum requirements on the OTA when used in an S/H amplifier because it must have adequate phase margin at high frequencies where the feedback is greatest and the low frequency gain requirements are increased due to the increased input capacitance due to the Miller multiplication of  $C_{gd}$ . In practice, the phase margin is not a problem due to the fact that the non-dominant pole of the telescopic OTA can be made quite high. But the open loop gain requirements remains a problem.

One solution to the increased input capacitance due to Miller multiplication is the addition of cross coupled capacitors  $C_{m1}$  and  $C_{m2}$  as shown in Figure 6.18 [5]. The value of these equal sized capacitors is that of the  $C_{gd}$  of transistors M1 and M2. For differential inputs and matched capacitors, displacement current flows through these capacitors from the gate to the opposite drain with a polarity opposite to that of the displacement current due to the Miller multiplied  $C_{gd}$  which flows from the drain to the gate. These currents nearly cancel one another and the effect of Miller multiplication of  $C_{gd}$  is greatly reduced.



**Figure 6.18** Addition of auxiliary capacitors to reduce effects of Miller multiplied  $C_{gd}$ .

Refer now to Figure 6.19 and let us take a more detailed look at the reduction of the Miller multiplied  $C_{gd}$  as seen from input  $V_{in+}$ . Let  $I_1$  be the total displacement current that flows into both capacitors as the result of a small input voltage  $\Delta V$  where

$$I_1 = I_{c_{gd1}} + I_{c_{m1}} \quad (6.15)$$

Both currents  $I_{c_{gd1}}$  and  $I_{c_{m1}}$  are proportional to the small signal voltage that appears across capacitors  $C_{gd1}$  and  $C_{m1}$ , respectively. Assume the capacitors are equal. The small signal voltage across  $C_{gd1}$  is

$$\Delta V_{c_{gd1}} = \Delta V - (-\Delta V \times A) \quad (6.16)$$

$$= \Delta V \cdot (1 + A), \quad (6.17)$$

while the small signal voltage across  $C_{m1}$  is

$$\Delta V_{cm1} = \Delta V - (\Delta V \times A) \quad (6.18)$$

$$= \Delta V \cdot (1 - A). \quad (6.19)$$

Since currents  $I_{cgd1}$  and  $I_{cm1}$  are proportional to the small signal voltage that appears across the respective capacitors, the sum of the currents is proportional to the sum of voltages  $\Delta V_{cgd1}$  and  $\Delta V_{cm1}$  which combining Eq. 6.17 and Eq. 6.19 is  $2\Delta V$ . Therefore, an input of  $\Delta V$  results in an effective small signal voltage across  $C_{gd1}$  of  $2\Delta V$ , and the input source *sees* an effective capacitance looking up into capacitors  $C_{gd1}$  and  $C_{m1}$  of  $2C_{gd1}$ . Thus, when the auxiliary capacitors are equal to the gate-drain capacitance of the input devices, the result is a reduction of the Miller multiplication factor of  $(1+A)$  down to 2.

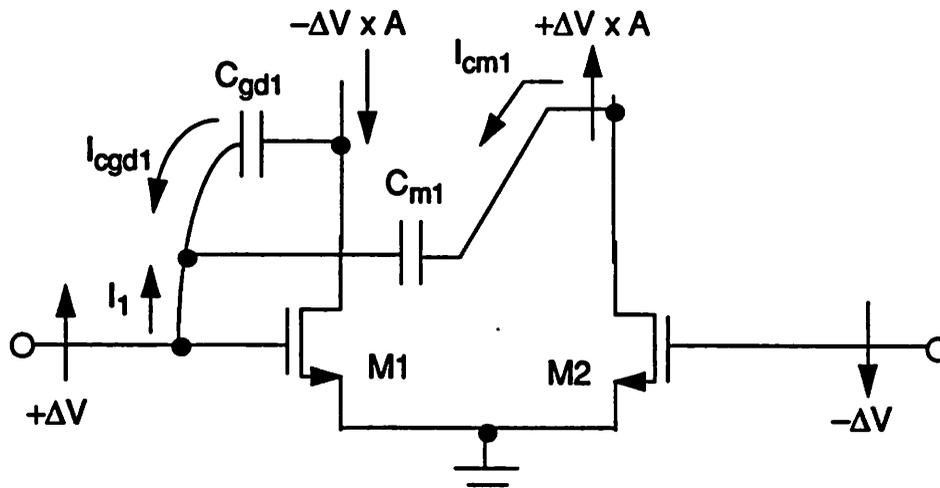


Figure 6.19 Schematic for analyzing the effect of auxiliary capacitor  $C_{m1}$ .

The major disadvantage with this solution is the increased capacitance on the source of the cascode device which device contributes directly to a decrease in the location of the non-dominant pole. In order for the auxiliary capacitors to match the gate-drain capacitance, they are often laid out as transistors of one-half the size of the input pair with the source and drain connected which greatly complicates the layout of the input stage of the opamp.

The preamp automatically reduces the effect of the Miller multiplication of  $C_{gd}$  since the gain  $A$  in the Miller multiplication factor of  $(1+A)$  is the preamp gain as opposed to  $g_m r_o$ . This circumvents the need for Miller capacitance cancellation techniques as just described and is yet another benefit of the use of the preamp with the telescopic OTA.

### Key points of this section

In this section, we have examined many advantages of the use of a wide-band preamp preceding a telescopic OTA. These advantages include:

- Increased transconductance for faster settling.
- Increased open loop gain.
- Reduced input capacitance for reduce open loop gain requirements.
- Added phase shift for second-order loop dynamics and reduced settling time.

In the discussion of the research prototype in Chapter 8, a full schematic and discussion of an OTA with a wideband preamp is described along with the necessary bias circuitry.

## 6.3 Sampling and Amplifier Clock Generation

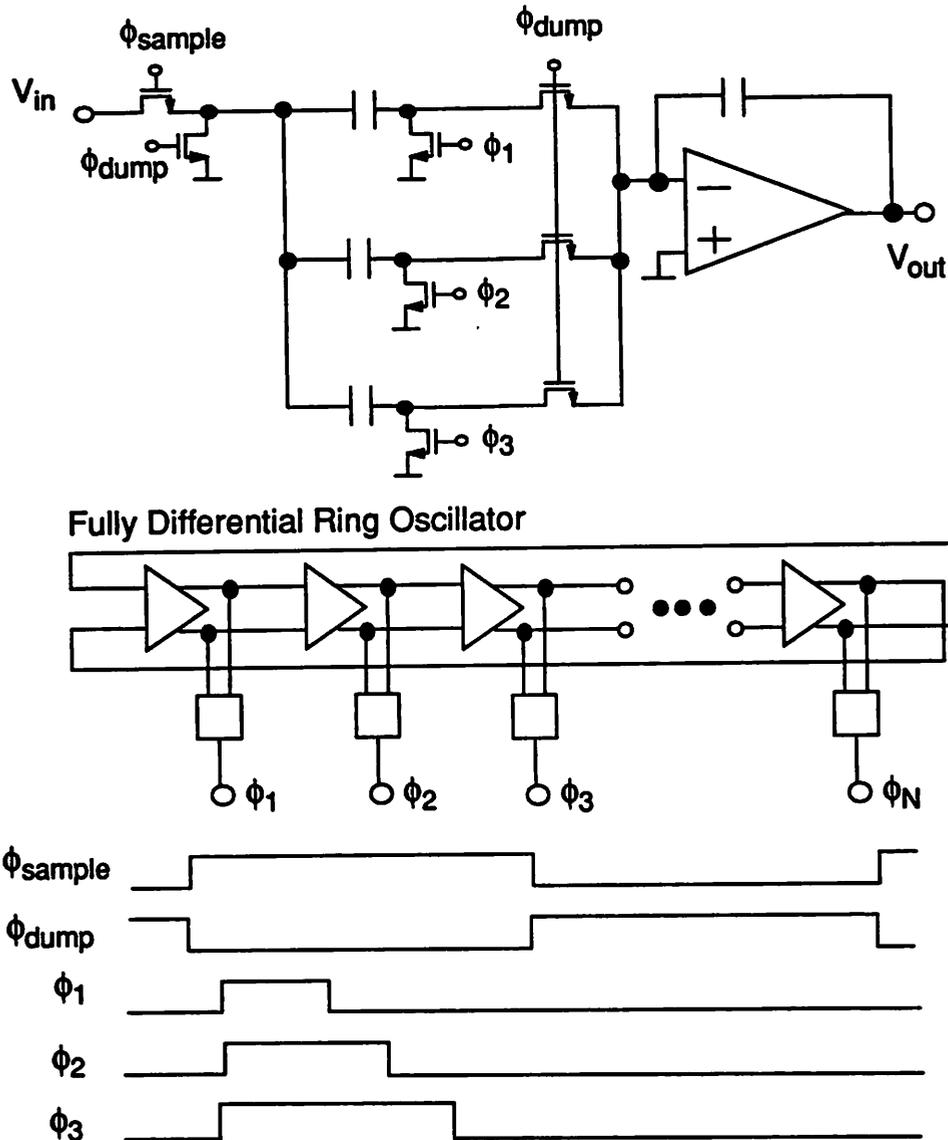
One of the more difficult design challenges of high speed mixed signal integrated circuits is that of generation of the clocks which control the switched-capacitor circuits. In particular, clocks which define the sampling instants in the interface from the continuous-time to the discrete-time domain and the generation of non-overlap clocks which properly control the flow of charge through the signal path are of critical importance. Although the voltage levels at the output of this circuitry include both supplies as in digital circuits, due to the critical nature of the precise location (in absolute time and sometimes in reference to one another) of the transitions between these levels makes generation of these clocks an analog circuit design problem. In this section, we will examine techniques for the generation of clocks in high speed analog signal processing applications.

### 6.3.1 Sampling Phase Generation

The effects of timing jitter or jitter in the phase of a sampling phase are becoming more important as operating frequencies increase. Signal frequencies in excess of 1 GHz are routinely dealt with in fiber optic systems with the use of high bandwidth technologies such as Bipolar or GaAs.

Due to the relatively high bandwidth and transconductance of devices in these technologies as compared with CMOS, simple structures may be used to implement signal processing functions at high speed. Unfortunately, CMOS architectures capable of high speed operation tend to require circuit techniques and architectures which are a bit more complicated than their bipolar counterparts. These more complicated architectures are necessary to overcome differences in inherent device speed. Increased complexity results in an increase in the number of sources of error. The complexity associated with clock generation for the high sampling rate front-end filter is an example of a technique that allows the processing of signals at much higher rates than possible using conventional techniques but has the cost of increased complexity.

In order to achieve high bandwidth filters, a parallel architecture was introduced in Chapter 5 which greatly increased the allowable settling time of the amplifiers at the expense of increased hardware (parallel channels) and the need for relatively high speed sampling clock generation. One set of sampling switches and capacitors along with the necessary clock waveforms are shown in Figure 6.20. A ring oscillator is also shown along with the outputs of different delay elements along the ring. The ring oscillator is attractive for this application because it facilitates generation of the different clock phases in applications requiring multiphase clocks. Clock phases in a ring oscillator have a fifty percent duty cycle making the direct application of a waveform off the ring oscillator incompatible for the sampling phase generation. With regards to sampling jitter and offset, the critical edge of the sampling clock (assuming NMOS switches) is the falling edge. The rising edge is non-critical.



**Figure 6.20** Simplified diagram showing required sampling waveforms for the high sampling rate front-end. The waveforms are generated off of the ring oscillator.

At this point, we have established two key points regarding sampling phase generation with the ring oscillator. First, a ring oscillator-based design is desirable for ease of generation of the sampling clock phases but the outputs cannot be used directly. Second, the falling edge of a sam-



has noise coupling to the local Vdd point as shown in Figure 6.22. Define the magnitude of the noise on Vdd at the instant at which the input waveform turns ON device M1 as  $V_{\text{noise}}$ . Assuming nothing else has changed, a time perturbation  $\Delta t$  will occur in the sampling instant due to  $V_{\text{noise}}$  where  $\Delta t$  is given by

$$\Delta t = \frac{V_{\text{noise}}}{m} \quad (6.20)$$

If  $V_{\text{noise}}$  is a d.c. event, then  $\Delta t$  will simply be a time offset. But if  $V_{\text{noise}}$  changes from cycle to cycle,  $\Delta t$  will be time varying and will appear as a jitter component in the sampling phase. Clearly, the statistics of  $\Delta t$  will depend on the statistics of  $V_{\text{noise}}$ .

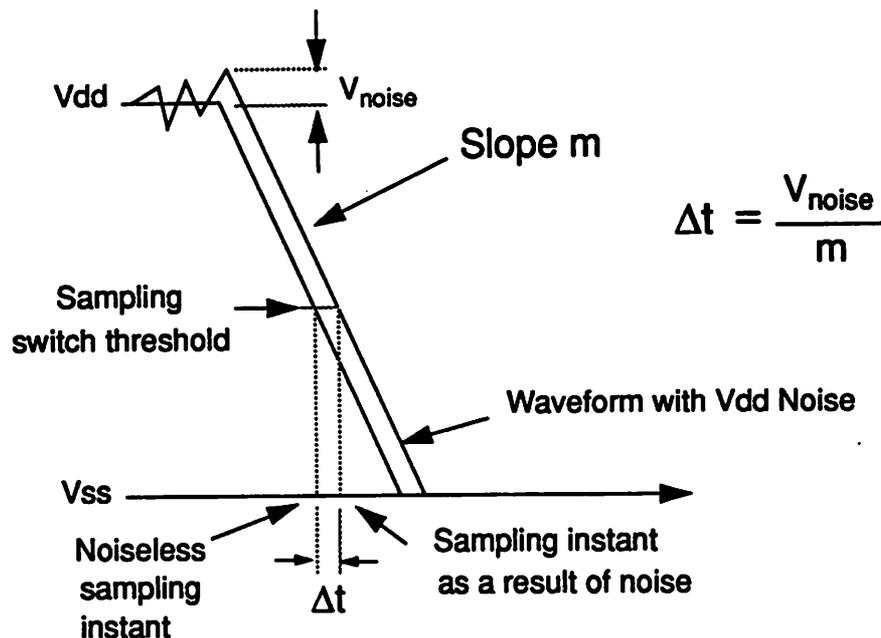


Figure 6.22 Illustration of the effect of supply noise on sampling phase.

Let us now consider the effect of device noise. It turns out that device noise can be treated in a similar manner as supply noise. We begin by defining a transitional voltage at each stage starting from the delay line or ring oscillator on through to the actual sampling switch. The *transitional voltage* is an arbitrary voltage threshold assigned at each stage where in the absence of noise, that particular stage is said to change state when the signal driving this stage passes through this thresh-

old. The input-referred device noise near the sampling instant is placed at the input of each of the blocks and the slope of the signal driving each block results in a variance in the instant of each transition equal to

$$\Delta t^2 = \frac{\overline{v_n^2}}{m} \quad (6.21)$$

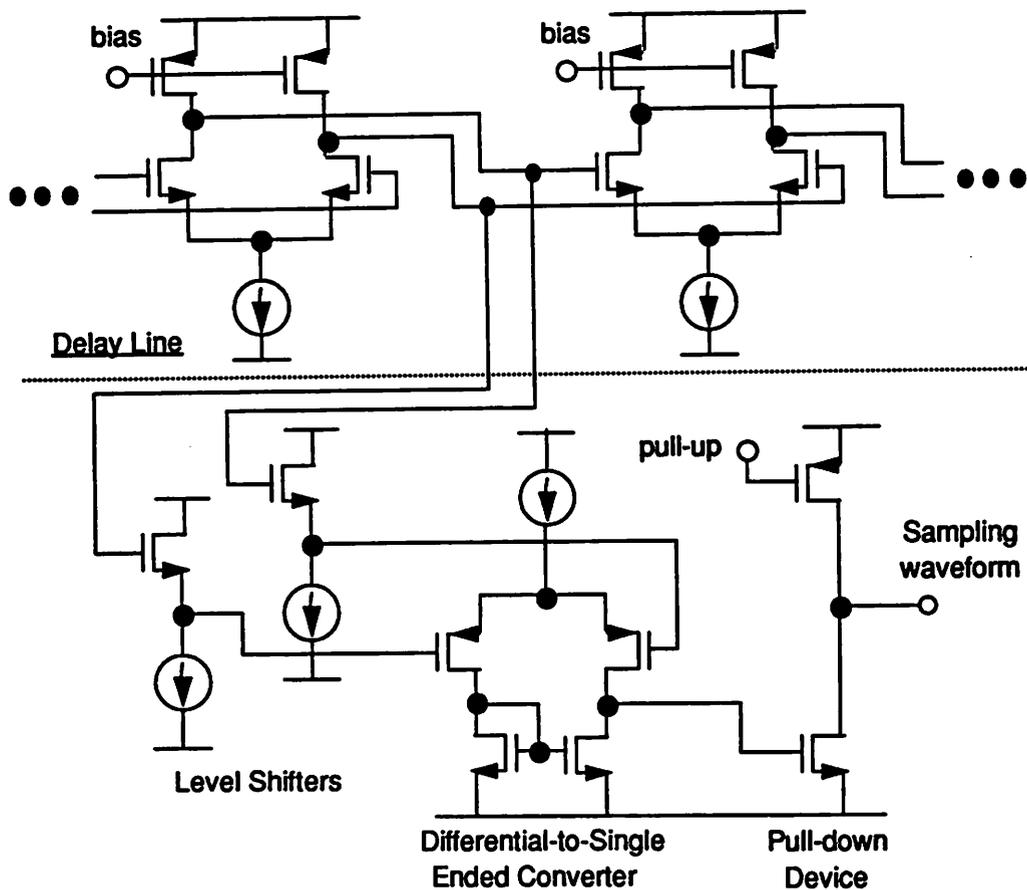
where  $\overline{v_n^2}$  is the input-referred noise variance of each stage,  $m$  is the slope of the driving signal, and  $\Delta t^2$  is the resulting jitter variance of that stage. Assuming each of these variances are independent, the variances at each stage can be summed to get the total variance or jitter due to device noise. In channels where parallel clocks are generated with different devices, voltage offsets contribute to non-uniform sampling or systematic timing offset from sample to sample. These offsets can be treated in the same manner but the input-referred offset of each block is used instead of the input-referred noise.

From this discussion, we can conclude a few things about sampling phase generation. First, the effects of supply noise on the sampling phase can be reduced with appropriate supply decoupling to all circuitry involved in generation of the sampling waveforms and the use of a differential signal path where ever possible. And second, the effects of supply noise and device noise and offsets can be minimized with the use of signals with large transition slopes.

The design developed and used in the prototype begins with a fully differential ring oscillator. A design similar to that described in [6] was employed for the delay element where differential pair NMOS devices drive PMOS triode loads. The voltage swing in the ring oscillator is 1 Volt, kept constant by controlling the resistance of the PMOS load devices. As a result of the fully differential architecture, noise from the power supply appears as common-mode noise and does not affect (to first order) the time location of the transitions through the oscillator. An effort is made to keep the circuitry from the ring oscillator to the sampling phase generation at the absolute minimum in order to minimize jitter and offset accumulation.

One of the key challenges of this design is to generate a single sampling phase that swings from the positive to the negative supply taking as input the differential signals from the ring oscillator that swing only 1 Volt in either direction. This circuit must also reject the common-mode noise that may be on the differential signal as a result of power supply noise.

The circuit shown in Figure 6.23 was used to meet these requirements. The signals from the ring oscillator are first buffered and level shifted by the source followers and then passed on to a differential-to-single ended converter which rejects common-mode noise. The output of this circuit drives a pull-down NMOS device optimized to result in a sharp slope in the pull-down signal. Since the rising edge of the sampling phase is non-critical, it is performed off-line by a PMOS pull-up device. The coordination of this pull-up device to the pull-down is described in detail along with many other pertinent details regarding the design of this stage in Chapter 8.



**Figure 6.23** Simplified schematic of circuitry used to transform the differential outputs from the delay line into a single-ended sampling waveform.

### 6.3.2 Non-Overlap Clock Generation

In switched-capacitor circuit design, the order in which switches turn on and off are of critical importance. There are *local* (within a block) issues as well as *global* issues (block-to-block) requiring careful consideration. In this section we will examine these requirements more closely and present circuit techniques which were developed to meet them.

#### 6.3.2.1 Local Non-Overlap Clock Generation

Let us begin with a discussion of the issues regarding local non-overlap clock generation. Consider the switched-capacitor integrator and associated clocks shown in Figure 6.24. There are 3 phases associated with both  $\phi_1$  and  $\phi_2$  for a total of 6 phases in all. The  $\phi_1$  set is used during the sampling phase and the  $\phi_2$  set during the dump or integrate phase. Associated with each set are  $\phi_n$  and  $\phi_p$  phases which control the input analog transmission gate. The extra phase  $\phi'$  controls the switch which defines the sampling instant. This technique is referred to as *bottom plate sampling* [7]. Since the node that this switch is connected to is either connected to ground (or common-mode input level) or one of the summing nodes of the opamp, the voltage across this switch remains small. In fact when the switch is on, the only time there is a voltage across it is in the sampling mode when displacement current flows through it to charge the capacitor. The device can be sized to keep this voltage relatively small. Since the voltage across this switch is always near zero, the amount of charge in its channel remains nearly constant while it is on. This is advantageous because when it is turned off, the amount of charge injected from the switch into the node connected to the capacitor is to first order independent of the input signal. Therefore, signal dependent charge injection occurs if the  $\phi_p$  switch turns off first. If either  $\phi_n$  or  $\phi_p$  were to turn off first or create a perturbation to the input signal while  $\phi_p$  had the sampling switch on, the charge on the opamp side of the capacitor would change resulting in an error term in the integration. When the input transmission gate is on, the left side of the input capacitor is connected to the input signal. Since the charge in the channel is equal to the product of the gate capacitance and  $(V_{gs} - V_t)$ , the charge in the channel is a function of the input signal. As a result, signal dependent charge injection can occur if the sampling switch does not turn off first.

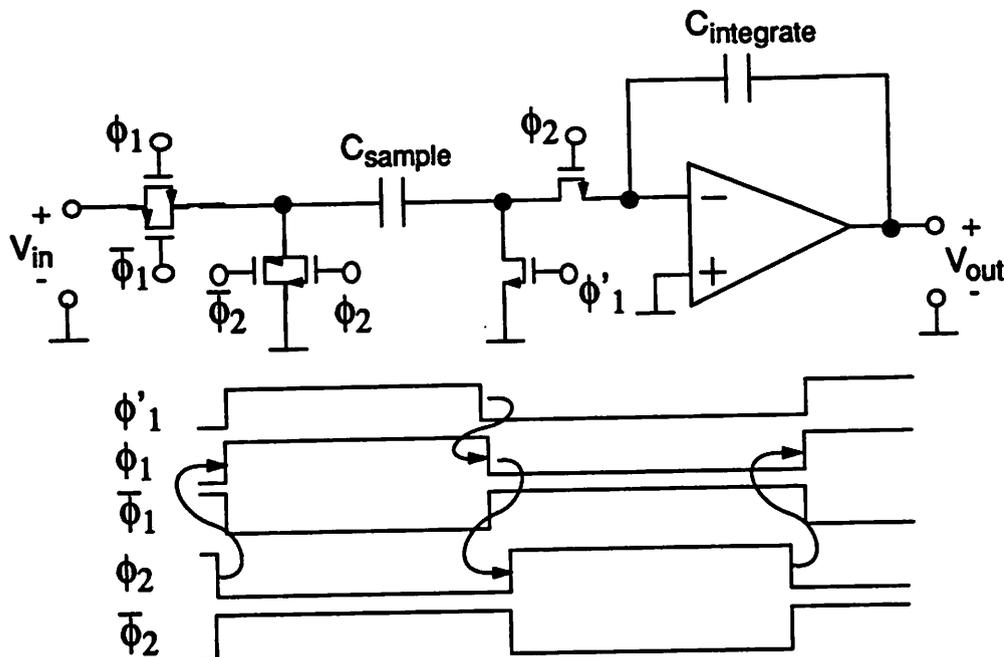


Figure 6.24 Switched-capacitor integrator and associated clocks.

At the instant the sampling switch is turned off, the charge on the sampling switch side of the capacitor is equal to the product of the input voltage at the sampling instant and the sampling capacitor plus some charge associated with clock feedthrough and charge injection. It is important to note that charge corresponding to clock feedthrough and charge injection are independent of the signal and thus contribute to a d.c. offset. After the  $\phi_1$  switches are all off, the  $\phi_2$  switches may be turned on. This is the classic 2 phase non-overlap clocks required by switched capacitor circuits. The extra phases of clocks are improvements which reduce signal dependent charge injection.

### 6.3.3 Global Non-Overlap Clock Generation

Issues of global or block-to-block non-overlap clock generation are similar to those of local clock generation in that the objective is to retain bottom plate sampling to reduce effect from signal dependent charge injection from the switches. In Figure 6.25, two blocks in a pipelined analog signal processing path are shown. The Stage 1 block takes in the input, processes the signal, and

holds the output for Stage 2 to sample it. The basic problem is that the sampling switch in Stage 2 must turn off before the output from Stage 1 changes.

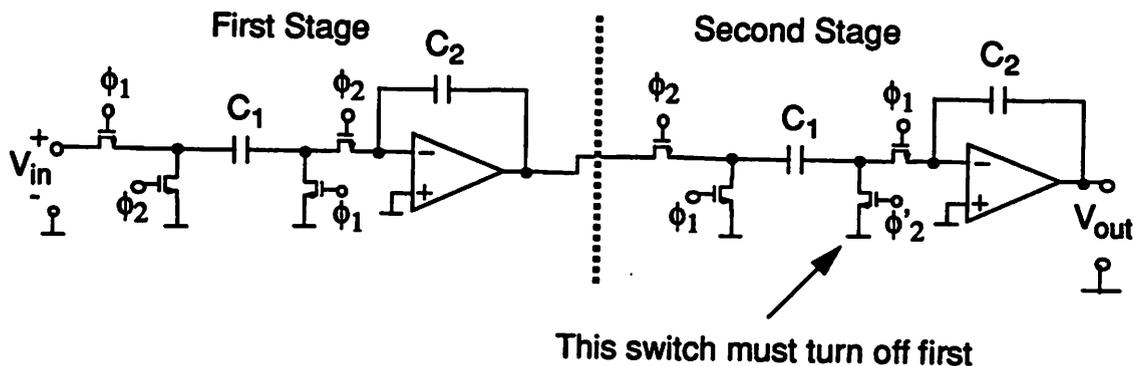


Figure 6.25 Schematic of pipelined analog processing illustrating the need for global synchronization.

We shall consider two approaches to solving this problem. The first is open loop clock generation and the second is a feedback based clock generation.

### 6.3.3.1 Circuit Approaches

One of the key problems in generation of non-overlapping clocks is that it must be done independent of capacitive loading and process, temperature, and supply variations. In *open loop clock generation*, worst case scenarios for the non-overlap clocks are found by inspection and delays are included where necessary to ensure that the desired clocking results over all scenarios. This approach has its limitations is that often the delays in a path must be mimicked in another which may have limitations since it may be very difficult to emulate the loading on a long line on a complicated chip accurately and efficiently. An approach that is often used is to place a lot of margin into the design for robustness. At high speed, this is very costly because the dead time involved to assure non-overlapping clock takes time away from amplifier settling and other functions for which may limit high speed performance.

There are risks involved with open loop clock generation because the worst case delay must be found and accounted for accordingly. Another disadvantage is cost of the extra time delay added to assure proper operation over different conditions is time that comes from the time budget of

another function. This extra cost must be included as overhead but it is desirable to minimize this cost. The next approach to be discussed uses feedback between the clock generator or within the blocks and therefore is closed loop. This allows the time overhead of the non-overlap clocks to be minimized.

In *closed-loop clock generation*, feedback within the clock generator or between the blocks exists. As a result, by design certain events must occur before others are allowed to happen. This allows the time overhead of the non-overlap clocks to be minimized. An example of both open- and closed-loop clock generation is shown in Figure 6.26 where non-overlap clock generators based on the cross-coupled NOR gates are shown.

Let us first consider the open-loop case. In the case of the input making the HI to LO transition, the  $\phi_1$  clocks must all complete their transitions before the  $\phi_2$  clocks change from OFF to ON. Thus, a delay is placed in parallel with the generation of the  $\phi_1$  clocks to ensure that  $\phi_{1n}$  occurs before the  $\phi_2$  clocks begin their transitions. Another way is to use closed-loop clock generation where  $\phi_{1n}$ , the last  $\phi_1$  clock to change is fed back ensuring that it completes its transition prior to the  $\phi_2$ ' change. This simple example indicates the fundamental difference between the two approaches. The use of closed-loop clock generation ensures by design that one set of transitions occurs before another. As can be seen in the figure, when using the closed-loop approach, the proper transition sequence should happen independent of loading and process variations.

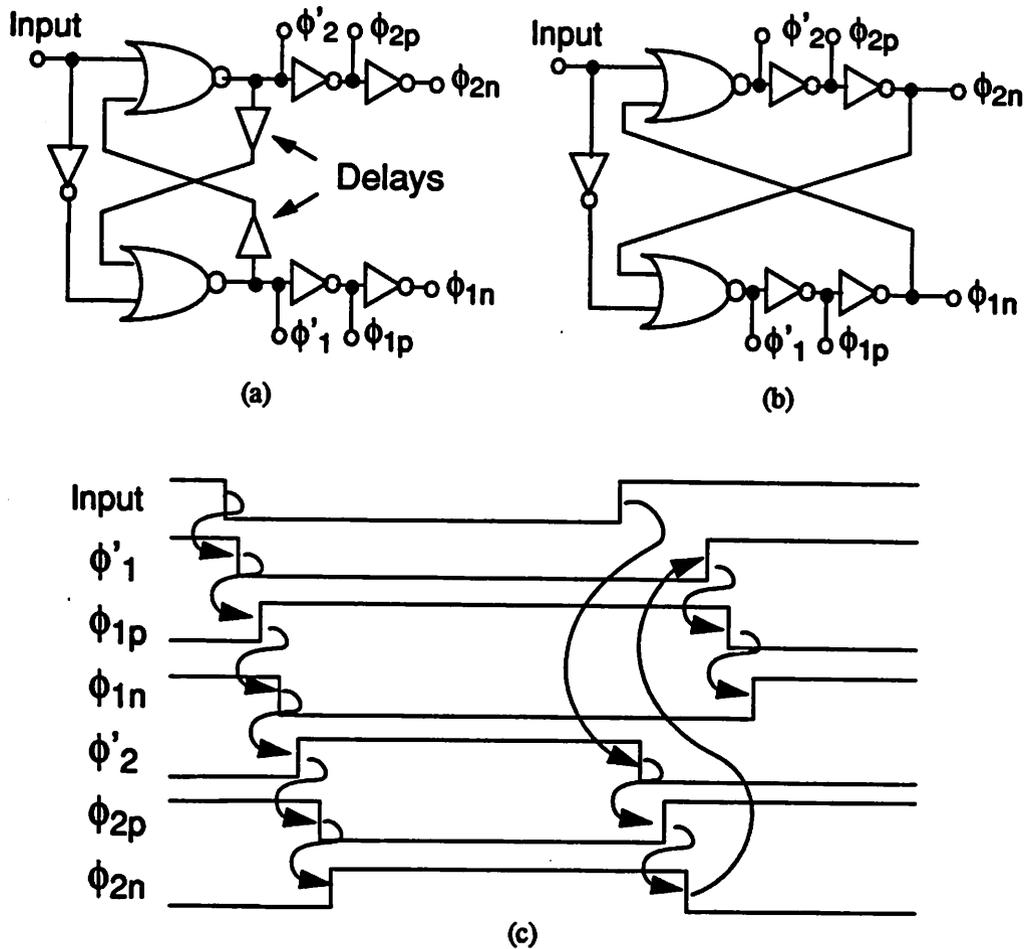


Figure 6.26 Two approaches to the generation of signals in (c). (a) Open-loop. (b) Closed-loop.

The cross-coupled NOR approach using the closed-loop approach works fine for local clock generation. However, there is not a convenient way to use this approach to meet global clocking issues. Furthermore, there is an asymmetry in the way the two phases are generated off of the input signal due to the extra inverter in the  $\phi_1$  path. And lastly, in situations as found in the analog-to-digital converter clocks, the duty cycle of many of the clocks may not be 50 percent but may yet require non-overlapping issues as found in a two-phase clocking scheme to be met. Use of the cross-coupled NOR in these situations require a very complicated set of waveforms to be generated which would act as the control signals to the cross-coupled NOR-based clock generators.

These waveforms would have to satisfy the some of the non-overlapping issues and interaction between clock generators would still be required.

In order to overcome these limitations and simplify the clock generation process, the circuit of Figure 6.27 was developed. As opposed to the cross-couple NOR based generator requiring just one input control signal, this approach requires two. To generate a transitions in the clocks, one of the control signals is pulsed HI as shown in the figure. Let us examine the  $\phi_1$  sampling ( $\phi_1$  HI to LO) transition. On the LO to HI transition of  $\phi_{1\text{fire}}$ ,  $\phi_1$  is pulled immediately LO by the NMOS pull-down. This results in the inverters propagating the transitions to the LO to HI transition of  $\phi_{1p}$ . At the time,  $\phi_{1\text{fire}}$  is still high so that both inputs into the NAND gate are HI resulting in the turn-on of the pull-up device which results in a LO to HI transition of  $\phi_2'$  which starts the transitioning of the  $\phi_2$  clocks. Transitions end off the HI going transition of  $\phi_1$  fire when  $\phi_{2p}$  goes LO. This process is repeated off of the activation of the  $\phi_{2\text{fire}}$  signal. The requirement is that the  $\phi_{\text{fire}}$  signals remain high for the time it takes for the other  $\phi'$  signal to go HI. An approach to generate the  $\phi_{\text{fire}}$  signals are described in Chapter 9.

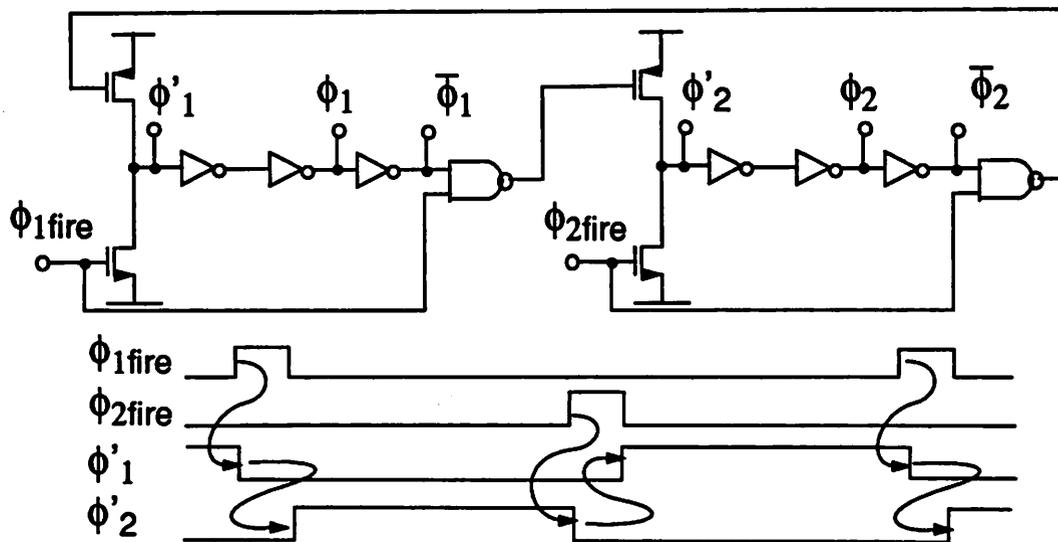


Figure 6.27 Circuit for generation of non-overlapping clocks.

The circuit of Figure 6.27 results in a symmetry in the way both phases of clocks are generated. We will see very shortly how this circuit can easily be modified to meet global clocking issues. And although more control signal may need to be generated, it is quite intuitive to think of

each of the control signals as fire signals that result in a set of transitions off of a LO to HI transition.

The circuit described above can be used in a feedback configuration to assure proper operation of clocks between two blocks. Let us assume both Stages 1 and 2 require local 6-phase non-overlap clocks and that Stage 2 samples the output of Stage 1 requiring that the sampling switch in Stage 2 turn off before the output of Stage 1 changes or equivalently the clocks in Stage 1 change. Suppose we want this to occur off the rising edge of  $\phi_n$ .

In Figure 6.28, the scenario described above is shown with the associated waveforms and the circuit which can generate these waveforms. Locally, the 6-phase non-overlap clock generator described in an earlier section is used. The difference is that instead of firing off of the rising edge of  $\phi_{fire}$ , the Stage 1 clock generator fires off the logical AND combination of a line fed back from the Stage 2 clock generator and  $\phi_{fire}$ . Upon the rising edge of  $\phi_{1fire}$ ,  $\phi'_{1ss}$  in the Stage 2 generator goes from HI to LO sampling the output from Stage 1. Clock  $\phi'_{1ss}$  then drives an inverter whose output is the feedback signal which is passed back to the Stage 1 clock generator. This feedback signal is logically ANDed with  $\phi_{1fire}$  resulting in the falling edge of  $\phi'_{1fs}$  in the Stage 1 generator. Generation of the other clocks follows as in the previous case.

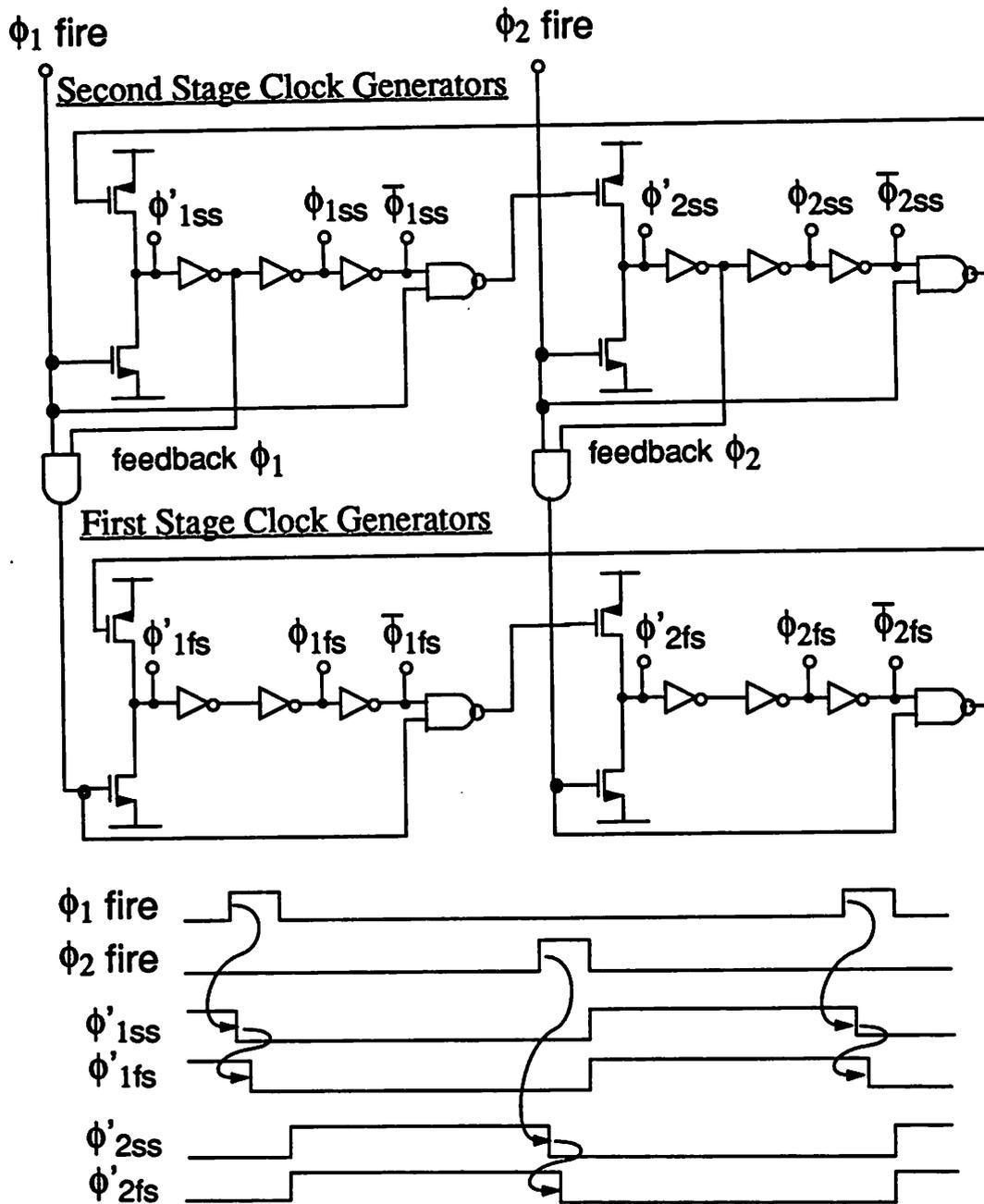


Figure 6.28 Circuit for generating global and local non-overlap clocks. Subscripts *ss* and *fs* denote second-stage and first-stage, respectively.

This circuit satisfies both local and global non-overlap clocking issues while minimizing the time overhead required in non-overlap clock generation. Because of the closed loop nature, proper operation occurs independent of loading and variations in process and operating conditions.

Let us consider one more application of this clock generation approach. In the analog-to-digital converter architecture developed for the prototype, many clock phases are required. One set of the clocks occurs as shown in Figure 6.31. Associated with each phase that is shown is a sampling phase clock  $\phi'$  and two multiplexor clocks  $\phi$  and  $\bar{\phi}$ . The non-overlapping requirement is that the previous  $\phi$  signal set turns OFF before the next can turn ON.

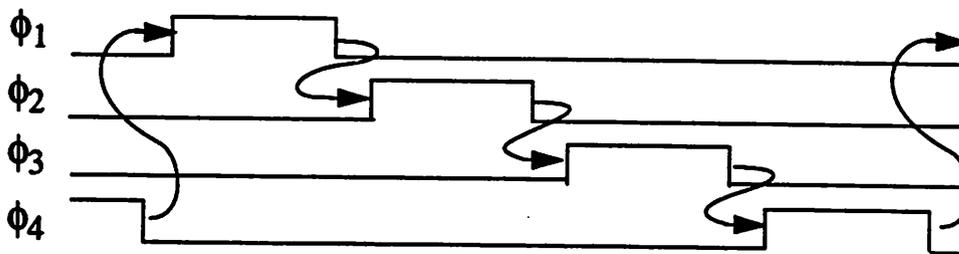


Figure 6.31 Sampling of clocks required in the analog-to-digital converter described in Chapter 7.

The circuit of Figure 6.27 can be applied to this application as well using the connections as shown in Figure 6.32. Each  $\phi_{\text{fire}}$  signal results in a HI to LO transition of the corresponding  $\phi'$  signal and the resulting LO to HI transition of the following phase. Note in the figure that as desired, all three phases associated with a particular clock phase need to occur before the following clock phase set can begin changing. This further demonstrates the flexibility of the clock circuit of Figure 6.27.

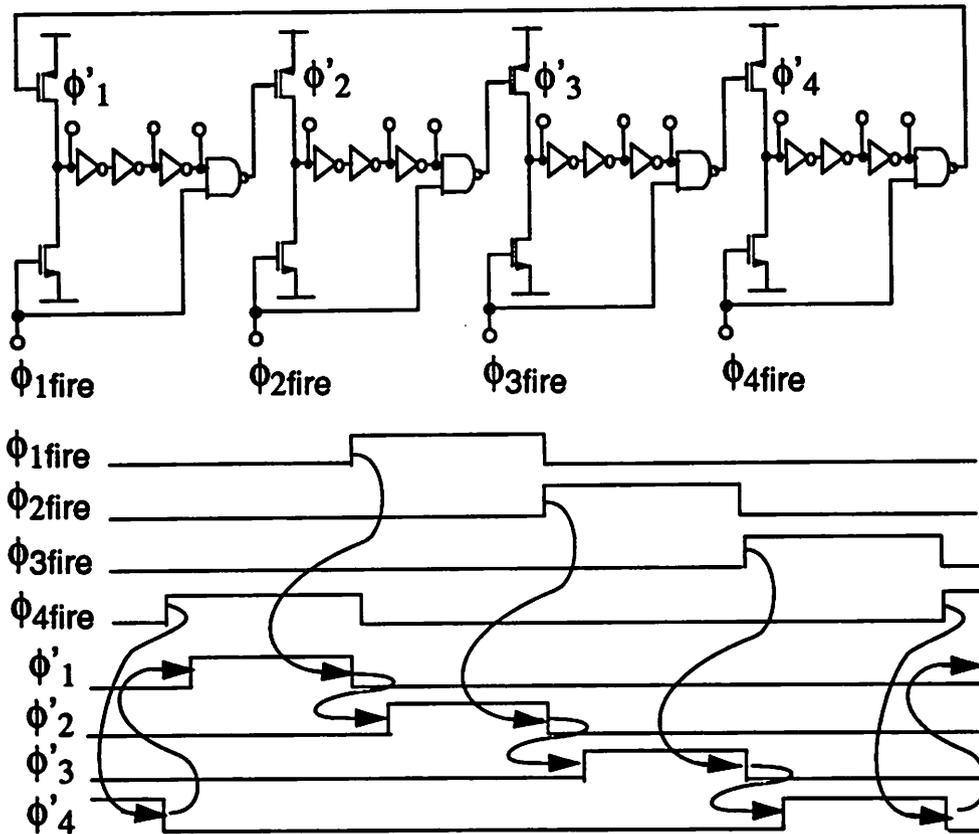


Figure 6.32 Generation of the clocks of Figure 6.31 using a permutation of the circuit shown in Figure 6.27.

## 6.4 Summary

In this chapter, we have presented some of the key circuit considerations for implementation of the high speed switched-capacitor transversal filters described in Chapter 5. We examined a switched-capacitor approach for implementation of the zero-order sampled-and-hold function. Then we considered the effect of the feedback factor of a closed loop amplifier on key amplifier performance specifications such as settling time and the required open loop gain. We found the telescopic amplifier preceded by a wideband amplifier to have many important performance

advantages in the application of the zero-order sample-and-hold. This is particularly important because of the small feedback factor that the amplifier sees in the parallel FIR filter architecture, particularly due to parasitic capacitance on the summing nodes due to connectivity to the interleaved sampling capacitor arrays and the extra capacitance due to the zero-order hold circuitry. We then examined circuit approaches for generating the multiphase sampling clocks required in the front-end of the decimation filter architecture and the key considerations that need to be addressed in the design of these circuits. We also presented a circuit building block for generation of non-overlapping clocks in pipelined analog processing channels.

In Chapter 8, we shall examine circuits used in the prototype A/D front-end employing many of the ideas presented in this chapter.

---

## References

- [1] Y.-M. Lin, B. Kim, and P. R. Gray, "A 13-b 2.5-MHz Self-Calibrated Pipelined A/D Converter in 3- $\mu$ m CMOS," *IEEE Journal of Solid-State Circuits*, pp. 628-636, April 1992.
- [2] G. Nicollini, P. Confalonieri, and D. Senderowicz, "A Fully Differential Sample-and-Hold Circuit for High-Speed Applications," *IEEE Journal of Solid-State Circuits*, Vol. 24, No. 5, pp. 1461-1464, October 1989.
- [3] T. C. Choi, R. T. Kaneshiro, R. W. Brodersen, P. R. Gray, W. B. Jett, and M. Wilcox, "High-Frequency CMOS Switched-Capacitor Filters for Communications Application," *IEEE Journal of Solid-State Circuits*, Vol. SC-18, No. 6, pp. 652-664, December 1983.
- [4] A. A. Abidi, "On the Operation of Cascode Gain Stages," *IEEE Journal of Solid-State Circuits*, Vol. 23, No. 6, pp. 1434-1436, December 1988.
- [5] B.-S. Song, M. F. Tompsett, and K. R. Lakshmi Kumar, "A 12-bit 1-M Sample/s Capacitor Error-Averaging Pipelined A/D Converter," *IEEE Journal of Solid-State Circuits*, Vol. 23, No. 6, pp. 1324-1333, December 1988.
- [6] B. Kim, D. N. Helman, and P. R. Gray, "A 30 MHz Hybrid Analog/Digital Clock Recovery Circuit in 2- $\mu$ m CMOS," *IEEE Journal of Solid-State Circuits*, Vol. 25, No. 6, pp. 1385-1394, December 1990.
- [7] K. Martin, "Improved Circuits for the Realization of Switched-Capacitor Filters," *IEEE Transactions on Circuits and Systems*, Vol. CAS-27, No. 4, pp. 237-244, April 1980.

1. The first part of the document is a letter from the author to the editor, dated 10/10/1954. The letter is addressed to the Editor of the Journal of the American Medical Association, Chicago, Illinois. The author is Dr. J. H. [Name obscured] of the University of California, Los Angeles. The letter discusses the author's interest in the journal and the possibility of contributing to it.

2. The second part of the document is a letter from the editor to the author, dated 10/15/1954. The editor is Dr. [Name obscured] of the American Medical Association. The editor's letter is a response to the author's letter and discusses the journal's policies and the possibility of accepting the author's work.

3. The third part of the document is a letter from the author to the editor, dated 10/20/1954. The author is Dr. J. H. [Name obscured] of the University of California, Los Angeles. The letter discusses the author's interest in the journal and the possibility of contributing to it.

4. The fourth part of the document is a letter from the editor to the author, dated 10/25/1954. The editor is Dr. [Name obscured] of the American Medical Association. The editor's letter is a response to the author's letter and discusses the journal's policies and the possibility of accepting the author's work.

5. The fifth part of the document is a letter from the author to the editor, dated 10/30/1954. The author is Dr. J. H. [Name obscured] of the University of California, Los Angeles. The letter discusses the author's interest in the journal and the possibility of contributing to it.

6. The sixth part of the document is a letter from the editor to the author, dated 11/5/1954. The editor is Dr. [Name obscured] of the American Medical Association. The editor's letter is a response to the author's letter and discusses the journal's policies and the possibility of accepting the author's work.

10/10/1954

---

# CHAPTER 7

## High Speed Analog-to-Digital Converters

---

### 7.0 Introduction

The analog-to-digital converter (ADC or A/D converter) is a key building block in digital communication receivers employing digital signal processing techniques. In bridging the analog and digital domains, performance of the ADC often limits the achievable performance of the receiver. Many architectural choices in a receiver are affected by the A/D converter architecture. The key parameters which generally characterize A/D converter architectures are speed, resolution, and power. In the context of full CMOS mixed signal solutions to communications receivers, other important A/D converter parameters include input capacitance and the time allowed for a front-end sample-and-hold to settle to its final value while driving this capacitance, latency through the converter, and comparator meta-stability and output sparkle codes.

The requirements for the magnetic disk drive channel employing digital filtering and detection are as follows. Roughly 5-6 bits of resolution for an output signal-to-noise ratio in the vicinity of 30 dB or so. Speed requirements are quickly approaching 100 samples/sec (S/s) and will eventually increase above this rate. Latency is important because timing recovery is normally performed

in the digital domain and therefore must be kept to a minimum. These specifications must be met at the minimum power consumption. As the form factor of drives continues to decrease, heat dissipated by the electronics becomes a problem due to the proximity to the media.

In this chapter, we will focus on A/D converter considerations and approaches that are candidates to meet the requirements of the magnetic disk drive channel employing discrete-time processing in the digital domain. This is an application where the resolution is modest (5-6 bits) but where speed and power are important. We will begin this chapter with some important system level architectural issues. We will then examine different ADC architectures and discuss the advantages and disadvantages of each approach for the disk drive read channel application. And finally, we will develop an architecture for a highly efficient parallel A/D converter implementation approach employing a two-step pipeline architecture. This approach meets the key objective of allowing the maximum settling time for a preceding set of time-interleaved sample-and-hold amplifiers. This approach is used in the prototype described in Chapter 8.

The input signal throughout the remainder of this chapter is assumed to be a voltage. The use of A/D converter approaches that operate in the current domain have been proposed [1]. However, they all require a linear transconductor with large input voltage range and a bandwidth much greater than that of the input signal which limits performance at high frequencies.

There are basically two classes of converters; Nyquist rate converters where the input rate is equal to the output rate and Sigma-Delta converters where the input sampling rate is much higher than the output rate. Sigma-Delta converters [2,11,12] are referred to as oversampled converters since the sampling rate is much greater than the Nyquist rate. They trade-off the complexity associated with a high sampling rate for increased resolution at signal bandwidths much lower than the sampling frequency. However, this comes at the cost of oversampling and increased latency which are not appropriate for very high speed data communication applications. Therefore we limit our discussion to Nyquist rate converters only.

## 7.1 Analog-to-Digital Converter System-Related Issues

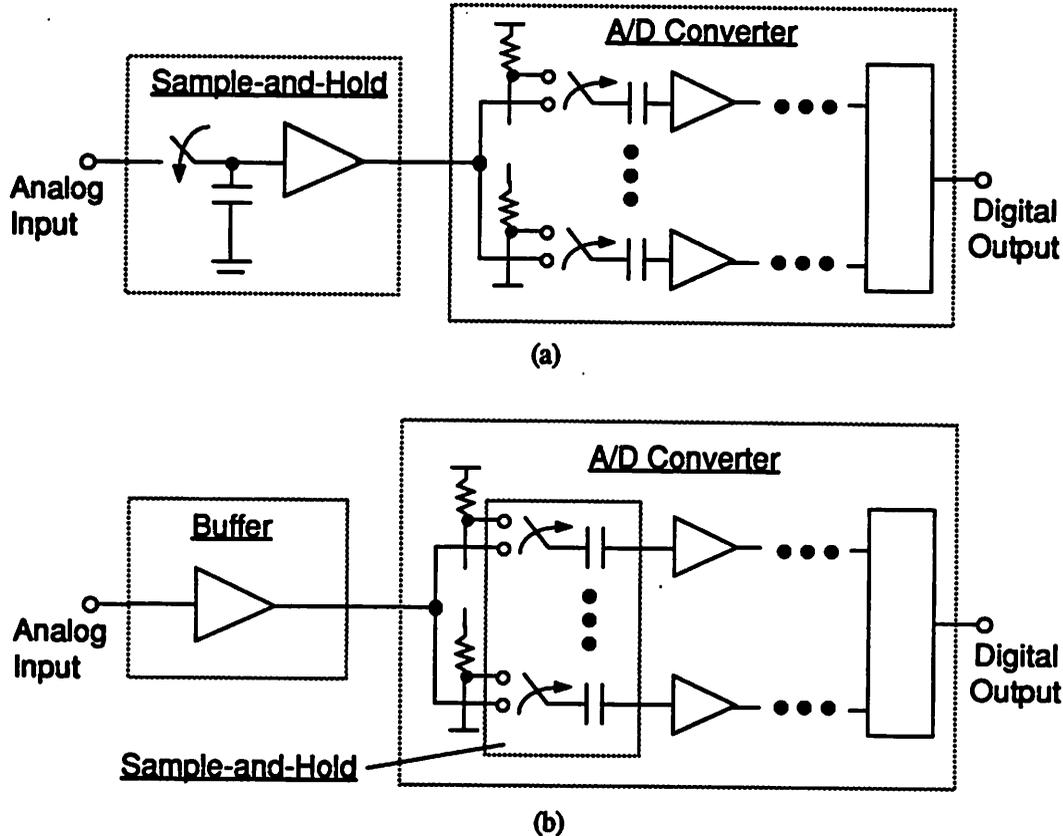
In this section, we will take a closer look at some of the key architecture and performance issues associated with choosing an A/D converter architecture for the read channel application. In particular, we will examine loading on the sample-and-hold amplifier, noise and distortion performance, and comparator metastability and latency.

The A/D converter block operates on samples of a signal in the analog domain and converts it to a digital code (number) which is proportional to the relationship of the input signal to a reference level. A reference range is broken up into  $2^N$  different levels. The A/D converter attempts to find the reference level closest to the input signal and outputs the code corresponding to that level. The larger (smaller) the input signal amplitude, the larger (smaller) the resultant code.

### 7.1.1 Loading on the Previous Block

The ADC function begins with the sample-and-hold block which may be an explicit block or folded into the architecture of the A/D converter. This block samples the analog input voltage and then holds this voltage at its output for conversion by the ADC. The bandwidth of the sampling circuit must be great enough so that it has a negligible effect on the input signal throughout the bandwidth of interest. When an explicit sample-and-hold block is used, an amplifier-based circuit is usually used to reduce the effects of parasitic capacitances. This amplifier must be able to drive the input impedance of the ADC in the allowed time to the sampled input voltage. When the sample-and-hold function is folded into the A/D converter, a continuous-time buffer is required to drive the sampling circuitry in the ADC.

Two scenarios of the input block of the ADC are shown in Figure 7.1. In Figure 7.1a, the sample-and-hold function is a separate block which precedes the ADC. In Figure 7.1b, the sample-and-hold function is folded into the first stage of the ADC requiring a high bandwidth low impedance buffer. The choice would depend on circumstances such as whether or not discrete-time processing is being performed prior to the ADC and whether or not a sample-and-hold function fits conveniently into the ADC architecture.



**Figure 7.1** Two sample-and-hold/ADC scenarios. (a) Sample-and-Hold precedes ADC. (b) Sample-and-Hold folded into ADC requiring buffer.

Most commercially available high speed ADCs have the sampled-and-hold folded into the converter which is the scenario shown in Figure 7.1b. They typically have an input capacitance between 15 and 35 pF, which is driven by high bandwidth buffers from off-chip. Each of the sampling networks within the ADC which sample the continuous-time input signal must have a bandwidth larger than the input signal bandwidth. The key problems with this approach for a fully integrated single chip solution in CMOS are that it may be difficult to implement such a large high bandwidth buffer on-chip and each of the sampling networks within the ADC needs a very large bandwidth.

By using the approach shown in Figure 7.1a, the sampling network in the sample-and-hold block can be optimized for bandwidth and the ADC from end can be optimized for other criteria

other than bandwidth. The key problem of this approach is that a high speed amplifier is now needed with can settle to the final value in the allocated time. This is one of the key problems addressed by the architecture used in the prototype.

### 7.1.2 Quantization Noise and Distortion

In the absence of noise and offsets, the A/D converter effectively finds which of the  $2^N$  reference voltages the input voltage is nearest and more positive than. For a given reference  $V_{\text{ref}}$  and  $N$  bits, each reference voltage is roughly  $V_{\text{ref}}/2^N$  apart. Therefore, there are twice as many reference levels for each added bit. The result is that the input is quantized with greater granularity as the number of bits is increased.

Unless the input is at some arbitrarily small value above one of the  $2^N$  reference voltages during a conversion, there is an error associated with the conversion equal to the actual input voltage and the voltage corresponding to the output code determined by the converter. This noise is a random component and is often modelled as an additive white noise source at the output of an ideal converter. This noise really depends on the statistics of the input signal and is usually not white. However, treating quantization noise as additive white gaussian noise greatly simplifies analysis. For our purposes, we need the relationship between the number of bits and the resultant signal-to-noise ratio. Since the statistics of the quantization noise is a function of the statistics of the input signal, it follows that the SNR depends on the input signal statistics. For a sinusoidal input, the SNR is equal to [3]

$$SNR_{dB} = 6 \cdot N + 1.76dB \quad (6.22)$$

where  $N$  is the number of bits.

Harmonic distortion is another signal degradation that may be introduced in the A/D converter or by other blocks in the signal path. One way that analog signal paths with back-end A/D converters are tested is to input a sinusoid and observe the digital output using digital signal processing techniques. The power of the signal at the input frequency is then compared to the signal power at all other frequencies excluding d.c. resulting in a signal-to-noise plus distortion (S/N+D)ratio. This number is stated in dB and for the disk drive channel, S/(N+D) of around 30 dB is required. Actual

distortion specifications in general depends on the particular signalling method being used and the off-channel SNR.

### 7.1.3 Converter Latency

In communication receivers using digital signal processing, the A/D converter is usually present so that techniques such as adaptive equalization, sequence detection, and timing recovery may be performed in the digital domain. Timing recovery is often performed in the digital domain because many of the often used decision-directed algorithms require equalized samples. In these situations, latency in the path from the sampler to the update of the voltage controlled oscillator defining the sampling must be minimized for loop stability. The allowable latency depends on the actual implementation of the receiver. It is important to keep in mind when surveying approaches that latency in this application is rather critical.

### 7.1.4 Comparator Metastability

Comparator metastability has to do with a comparators inability to “make up its mind”. The key objective of a comparator is to make a decision whether or not an input voltage is larger than or smaller than a reference voltage. Its output is a logic 1 or 0, depending on the result of the comparison. In order for its output to be a logic 0 or 1, its input must be sufficiently large to push it in one direction or the other toward a decision. In the case where regenerative latches are used, the input must be sufficiently large simply to start it in one direction and the positive feedback present in the regenerative latch will drive it to a decision. There is always a non-zero probability that the input will be such that a comparator is unable to make a decision (right or wrong). When this happens, a metastable output has occurred and rather than be a logic HI or LO the output may be somewhere in between. It turns out that a metastability occurrence can do more harm than if a comparator is to make a wrong decision.

Although there are many architectures for implementation of A/D converters, one similarity found in most converters is a point at which an input voltage is compared to a number of equally spaced reference voltages by the same number of comparators placed in parallel. Each comparator compares the input to a different reference voltage. The output of each of these comparators is then processed by a circuit block which for the moment we shall call a *decoder*. The decoder deter-

mines the location of the input voltage with respect to the reference voltages. It may determine for example, which reference voltage the input is *closest* to. Decoders are normally designed to take as input the binary logic levels 0 or 1. In the case of a meta-stable condition where an input is somewhere in between, the result can be catastrophic and the decoder may *decide* that the input is closest to a reference that is far from the correct. Suppose an input is near the bottom-most reference voltage and a metastable condition occurs in the comparator associated with this reference. The metastable input level from this comparator into the decoder may result in the decoder decision that the input is nearest the top-most reference. This is extremely undesirable in applications in data communications and magnetic data storage because a decision error on a symbol is likely to occur. Output codes such the one that would result in this worst case example in which the A/D converter output code is far away and seemingly unrelated to the actual output code are called *sparkle codes*. Sparkle codes can also be caused by noise related events that result in random errors in the decoder. Careful circuit design and layout techniques can minimize the probability of occurrence of such events to negligible levels. Comparator offsets in some A/D architectures can result in sparkle codes when simple decoders are used. There are different approaches such as the use of grey codes which are designed to constrain the magnitude of these errors [4].

The probability of a comparator remaining in a metastable condition is an exponentially decreasing function of time [5,6]. Thus, the longer a comparator is given to make a decision, the chances of it staying in a metastable state decreases very quickly. This is because device noise, clock noise, and other feedthrough effects tend to drive comparators out of this condition should they start there. Therefore, it follows that it is desirable for a comparator to be allotted the maximum decision time possible in order to decrease the probability of a metastable condition.

In summary, if a technique is used in the decode process that can correct for or constrain the severity of sparkle code errors, it may be better for a comparator to make a wrong decision than to stay in a metastable state. From a design standpoint, it is desirable to allow comparators maximum decision time and to use some form of redundancy in the decoder to limit the severity of wrong comparator decisions.

## 7.2 A/D Converter Architectures

In this section, we shall examine candidate architectures for high speed A/D converters. We shall briefly examine 4 different architectures including the flash, sub-ranging, pipeline and 2-step pipeline approaches. This is by no means a thorough overview but merely some important candidate approaches which will help to justify design decisions made during the design process of the prototype. Throughout this chapter, we shall assume  $N$  to be the number of bits at the converter output.

### 7.2.1 Flash Converters

*Flash* converters [7] derive their name from the one-step nature of the conversion process. They are conceptually simplest of the A/D converter architectures and its basic structure is imbedded in just about all Nyquist rate converters. A simplified schematic diagram of a 3-bit flash converter is shown in Figure 7.2. Bipolar reference voltages  $V_{ref+}$  and  $V_{ref-}$  are placed across a resistor string with evenly spaced taps creating 8 ( $2^N$  where  $N$  equals the number of bits which in this case is 3) references. The input signal amplitude is compared with each of the reference voltages by a bank of comparators. Since the input is applied to the positive input of the comparators, each comparator asks the question “Is the input larger than its reference voltage? (Answer 1 for yes, 0 for no)”. Thus, the comparators find the reference voltage that the input is closest to without being greater than.

An example input voltage is shown with the input level indicated in the figure. Note in the figure that as we ascend the comparator outputs, they change from all 1s to all 0s. This is referred to as a *thermometer code* since the location of the transition from 1s to 0s moves with the location of the input amplitude. A decoding circuit then translates the thermometer code to a binary equivalent. One way to perform the translation is to have the thermometer code converted into a *1-hot* code which then drives a ROM. The 1-hot code gets its name from characteristic that the only bit that is 1 is the active bit. The thermometer-to-1-hot code converter is simply a differentiator which looks for the transition in the thermometer code and outputs a 1 in that location. All other outputs are 0. A circuit which implements this function is presented in Chapter 8.

The input signal range is assumed to be within  $V_{ref+}$  and  $V_{ref-}$ . Therefore, the output of the bottom most comparator in Figure 7.2 is 1 by definition. Thus, this comparator is redundant and may be removed. As a result, only  $2^{N-1}$  references and comparators are needed to resolve  $N$  bits.

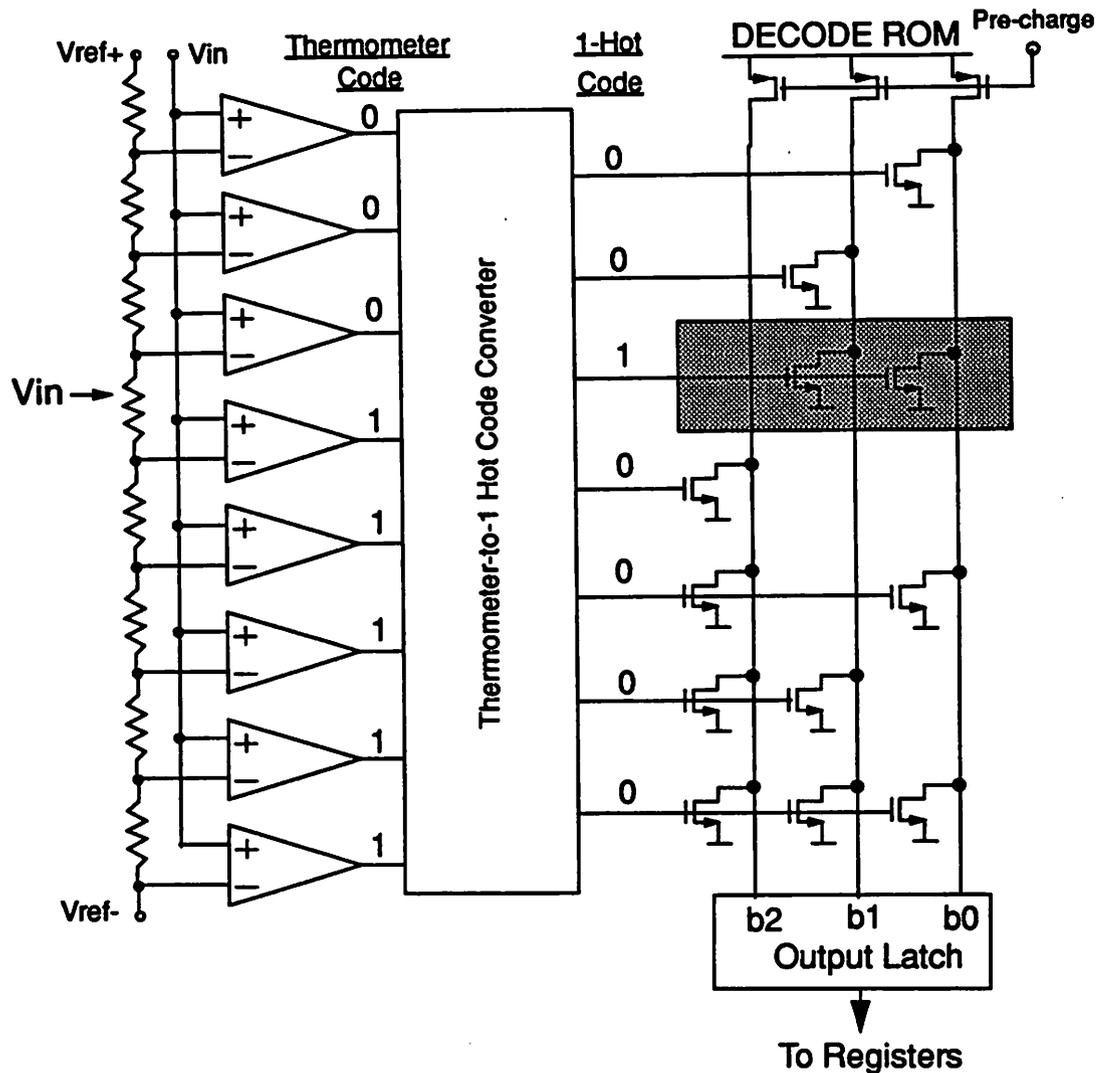


Figure 7.2 Simplified schematic of a 3-bit flash converter.

A sample-and-hold circuit samples the input and holds it while the comparators make decisions. With the flash converter architecture, at some point a circuit needs to drive the inputs to  $2^{N-1}$  comparators. Although this architecture allows very high speed processing and low latency, the large input capacitance associated with the architecture places a heavy burden on the circuit

driving the comparator inputs. Unless an extremely high speed buffer can be designed, the full speed potential of the architecture may not be achievable. Another limitation of the flash architecture is that the comparator count grows exponentially with the number of bits in the conversion, resulting in large area and high power consumption.

### 7.2.2 Subranging Converters

In the *subranging* converter [8], the conversion is performed in more than one step with the primary advantage of fewer comparators (less total hardware) than the flash converter. Two steps are about as many as is practicable for reasons which will soon be evident. In Figure 7.3, a block diagram for a two step subranging converter is shown where a *coarse* quantization of  $n$  bits is followed by a *fine* quantization of  $m$  bits ( $n + m = N$ ). In the example shown in the figure,  $n=m=2$ . The range of  $2^{N-1}$  levels is broken into  $2^n$  chords and during the coarse quantization, it is determined in which chord the input lies. A chord is a range of  $2^m$  reference levels. The LSB references above the reference corresponding to the MSB decision are then passed to the second stage converter which finds which LSB reference the input sample is closest. The structures used for both conversions can be similar to that used in the flash converter where the input is compared with  $2^n-1$  or  $2^m-1$  references.

The key advantages of the subranging converter over the flash converter architecture are that of lower input capacitance and fewer comparators. The input capacitance is lower because the load presented the driving circuit includes only  $2^n-1$  comparators as opposed to  $2^N-1$ . The total number of comparators is reduced from  $2^N-1$  down to  $2^n+2^m-2$ . For a 6-bit converter broken up into two 3-bit converters, this results in a reduction from 63 down to 14 comparators.

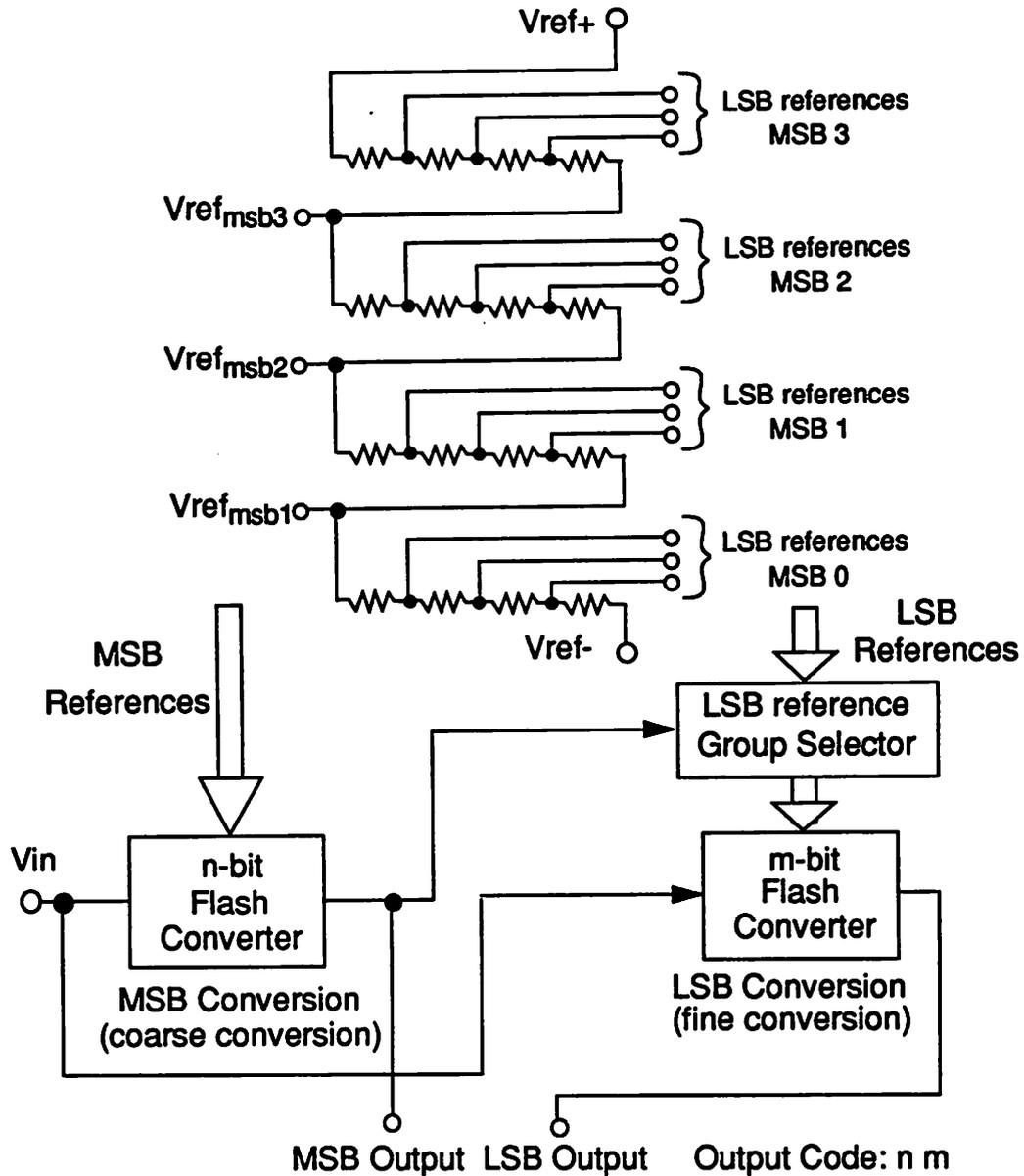


Figure 7.3 Block diagram of subranging converter (2+2=4 bits).

Subranging converters in practice are limited to two stages because multiplexing of the reference voltages quickly becomes unmanageable beyond two stages. One solution to this is the use of pipeline architectures described in the next section.

### 7.2.3 Pipeline Converters

*Pipeline* converters [9] break the conversion process up into many stages. A block diagram of a pipeline converter is shown in Figure 7.4a with  $P$  pipelined stages, each resolving  $m_p$  bits. The resolution of the converter is  $N=m_1+m_2+\dots+m_p$  bits. Let us refer to the expanded diagram of a pipeline stage in Figure 7.4b. In each stage, an  $m_p$ -bit flash conversion is performed on the analog input to that stage. The voltage corresponding to the MSB code is subtracted by a DAC from the input voltage. The result is a *residue* which is then amplified by an interstage amplifier by a factor of  $2^{m_p}$  (that is  $m_p$ ) and then output to the following stage.

This architecture overcomes the problems of extending the subranging converter beyond two stages by placing an interstage amplifier between stages which amplifies the residue so that the maximum amplitude in each stage is the same. In this way, the same set of reference voltages can be used in each conversion. Another advantage of pipeline ADCs is that it allows the use of calibration techniques which reduce the offset requirements of the comparators. An often used calibration technique is that referred to as *digital error correction* described in [9]. Redundancy is used in this technique by resolving an extra bit in a stage of the pipeline. Rather than use this bit as data, it is used to remove the effects of comparator offsets.

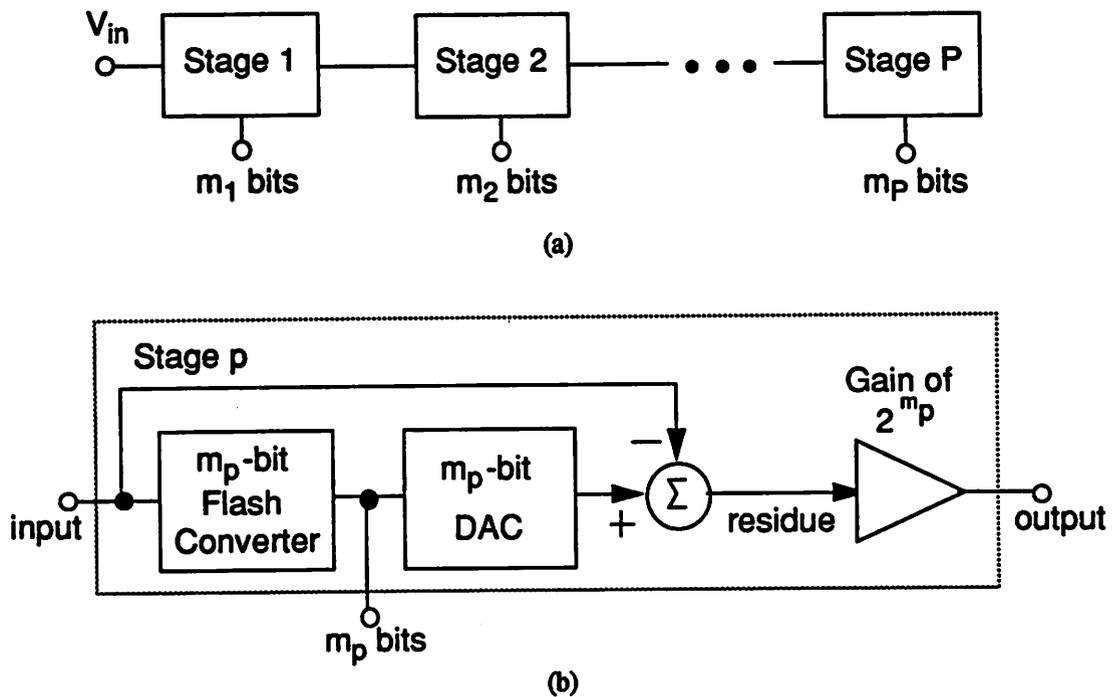


Figure 7.4 Conceptual block diagram of a pipeline converter ( $m_1+m_2+\dots+m_p=N$ ).

(a) Converter is made up of  $P$  stages. (b) Block diagram of stage  $p$ .

The modularity and the ease with which digital error correction can be incorporated into pipeline architectures make it very attractive for converters with resolutions between 8 and 13 bits. Ultimately, the resolution and speed performance tends to be limited by the finite gain and speed performance of the interstage amplifiers. The latency through these converters is relatively high which may be an issue in data communications applications.

At the 6 bit level, 5 V supply, and signal swings on the order of 2-3 Volts, pipeline architectures are a bit of an overkill. In particular, there is no need for an interstage amplifier at these signal levels. The prototype converter to be presented in the next section is a combination of both subranging and pipeline architectures without the interstage amplifier. At lower supply voltages and smaller signal levels that may become standard with deep submicron CMOS technologies and low power applications, pipeline converters may become important in read channel architectures.

### 7.2.4 Two-Step Pipeline Converters without Interstage Gain

The two-step pipeline ADC without interstage gain [10] is very similar to the pipeline converter just described except that there is no interstage gain between the conversion of the MSBs and the LSBs. We will refer to this approach as the *Two-Step Pipeline Converter*. A block diagram of a two-step pipeline is shown below in Figure 7.5. The conversion is broken up into an  $m_p$ -bit MSB conversion and an  $m_q$ -bit LSB conversion for a total of  $m_p+m_n=N$  bits.

Like the subranging architecture, these converters are limited to resolutions in the 6-8 bit range because of noise and offsets associated with generation of the residue.

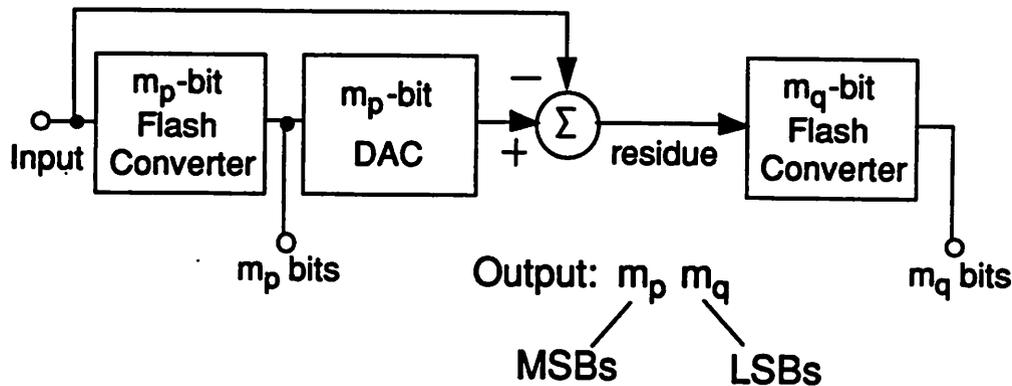


Figure 7.5 Conceptual block diagram of the two-step pipeline A/D converter.

### 7.3 Highly Parallel Two-Step Pipeline Implementation

The architecture used in the prototype is the two-step pipeline described at the end of the previous section. Both this architecture and the subranging architecture were the strongest candidates because they have far less input capacitance and comparator count than the flash converter, and they do not require the use of interstage amplifiers as does the pipeline approach. The reason for choosing the two-step over the subranging will become clear in this section as implementation details and considerations are presented.

We will begin this section by examining the context in which the converter is to be used on the prototype. In particular, we shall take into account the architecture of the equalizer stage which precedes the ADC. Then, we will examine some considerations for the implementation of a basic building block required in two-step A/D converters which is a difference amplifier followed by a regenerative latch. We will then consider application of this building block in the A/D converter required by the prototype, i.e., one driven by the parallel filter architecture. The key new implementation aspect of the converter is the high level of parallelism employed to interface to the parallel filter architecture.

### 7.3.1 System Context

The block level architecture for which the prototype was designed was presented near the end of Chapter 4 and is shown again below in Figure 7.6. In this section, we will develop the architecture used in the prototype for the high throughput 6-bit analog-to-digital converter needed to convert samples from the programmable equalizer into the digital domain. Circuit schematics and implementation details of the converter are presented in Chapter 8.

During development of the algorithm/architecture, an attempt was made to optimize the converter architecture to that of the driving stage, i.e., the programmable equalizer, while keeping a high level view on the cost-benefit trade-offs of such an optimization. An important objective of the filter approach (presented in Chapter 5 and used in the prototype) was to break the speed limitation of SC filters which was amplifier settling time. The parallel architecture used in the programmable equalizer allows the amplifiers in this block 3 output periods for settling. The ADC approach maintains this allowed settling time.

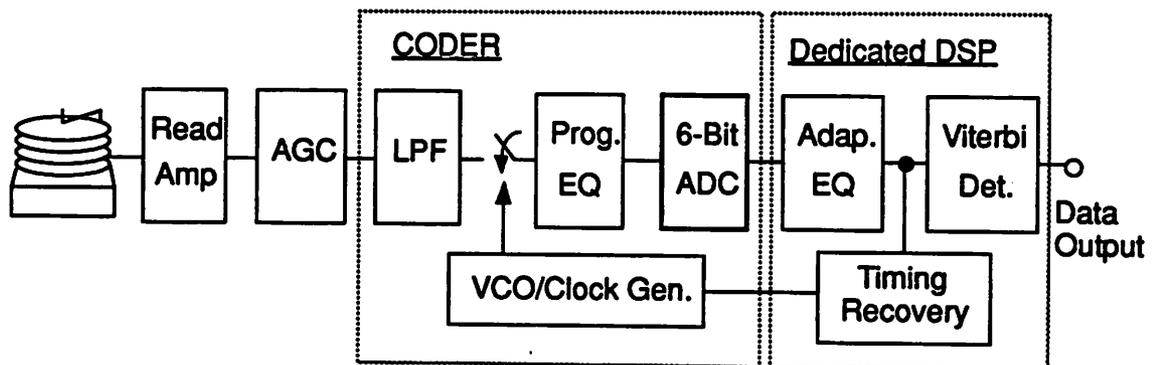


Figure 7.6 Block diagram of proposed read channel.

The general architecture for the equalizer was first presented in Chapter 5 in the example of the single-rate FIR filter. The equalizer is realized with 3 parallel filter sections, each operating one period out of phase as shown in Figure 7.7.

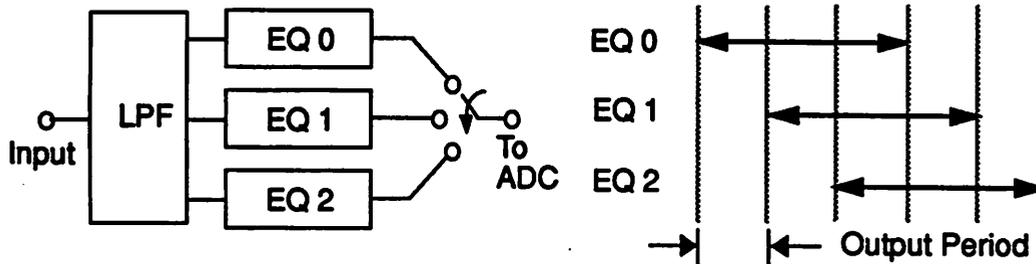


Figure 7.7 Parallel architecture of the equalizer and diagram illustrating interleaved operation.

There are at least two ways in which the ADC can sample the equalizer outputs and they are shown in Figure 7.8. In Figure 7.8a, the ADC is shown sampling the equalizer output at the effective output rate of the equalizer. In this approach, the ADC operates at the effective equalizer output rate. Another approach is shown in Figure 7.8b where each of the three equalizer sections drives its own ADC. This allows all the electronics to operate at one-third the output rate and is the approach we shall pursue.

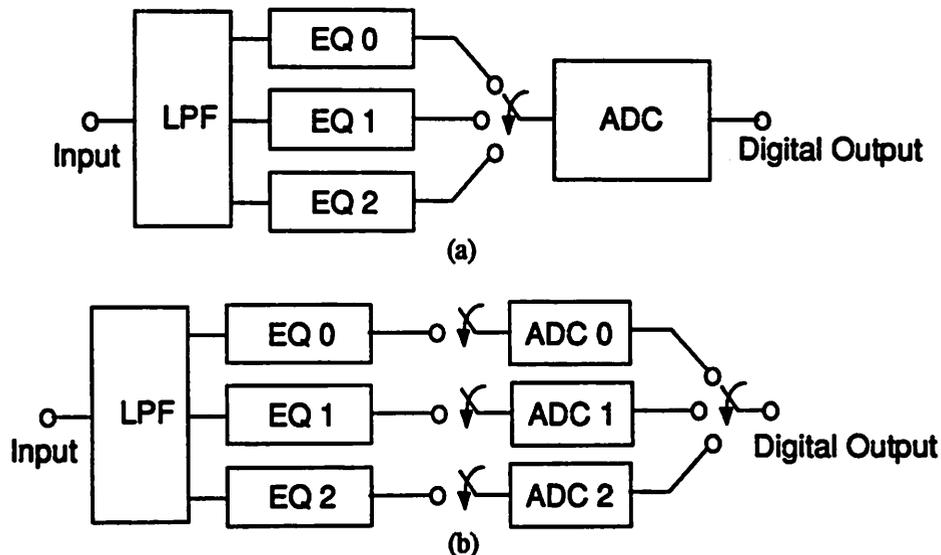
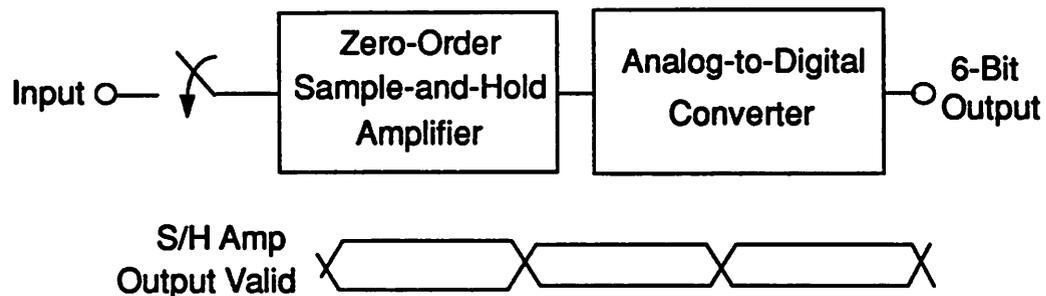


Figure 7.8 Two approaches for the A/D converter interface to the equalizer. (a) ADC operates at the effective output rate of the filter. (b) Each equalizer drives its own ADC. Each ADC operates at one-third the effective output rate.

In the remainder of this section, we will develop the ADC algorithm used in the prototype. We shall assume the block preceding the A/D converter to be an SC FIR filter employing the parallel architecture described in Chapter 5. Thus, each ADC is driven by an S/H amplifier with a zero-order hold output. This means that there is no reset or initialization period required by the previous stage amplifier and also that the sampling bandwidth issues are addressed in a previous block. We shall focus on developing an efficient architecture for the A/D converter. Let us refer to the amplifier in the equalizer as the sample-and-hold (S/H) amplifier. We shall also assume that throughput requirements are high, such that the time allocated for settling of the S/H amplifier has already been maximized.

The diagram in Figure 7.9 illustrates the scenario which we are addressing. One of the key objectives of the ADC design is to maintain the allowed settling time allocated to the S/H amplifier.

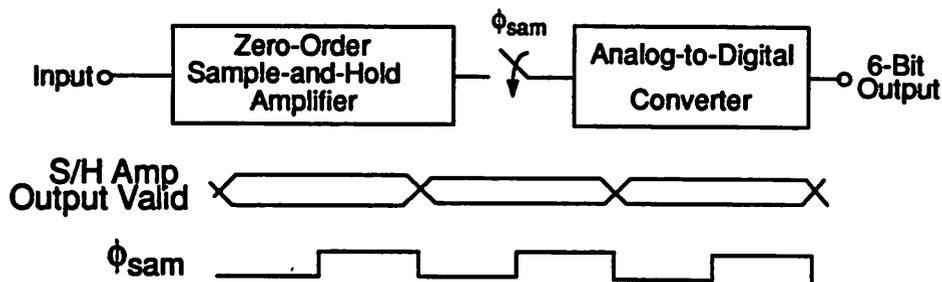


**Figure 7.9** Block Diagram of the S/H amplifier and A/D converter. The output of the S/H is valid every clock cycle so the A/D receives a new input each clock period.

### 7.3.2 A/D Converter Sampling of the S/H Amplifier

Let us examine ways in which the A/D converter can sample the output of the S/H amplifier and the impact of this on the overall A/D architecture. There are at least two ways in which the A/D converter can sample the S/H amplifier output. In the first, the output of the S/H amplifier is sampled by the A/D converter as shown in Figure 7.10. The problem with this approach is that the settling of the S/H amplifier is disturbed. The sampling process results in the input capacitance of the A/D converter to be connected to the settling amplifier which disturbs the loop of the amplifier,

thus prolonging the required settling. Amplifier settling generally has two parts: slewing and small signal settling. If the capacitive load is large, the amplifier may return to slewing followed by a period for settling upon addition of the loading of the ADC to the settling amplifier. Even if the load is small, it can still have a big effect on the settling of the closed loop amplifier. Due to the closed loop configuration of the amplifiers used in the SC FIR filter architecture, the majority of the time allowed the amplifiers tends to be spent on settling rather than slewing. As a result, an additional load in the form of the A/D converter input capacitance added to the amplifier output during the final stages of settling results in a disturbance of the loop, requiring the amplifier to begin settling all over again. It is thus desirable for the input capacitance of the A/D to be present throughout the settling time of the S/H amplifier.



**Figure 7.10** Sampling frequency of the A/D converter equal to output rate of the S/H amplifier. The rising edge of  $\phi_{sam}$  results in a disturbance of the settling of the S/H amplifier.

Another way for the A/D converter to sample the S/H amplifier is to interleave two half-rate converters as shown in Figure 7.11. While one converter is connected to the S/H amplifier output, the other is evaluating the previous S/H output. In this way, the settling time for the S/H amplifier is not affected and the converters can operate at a lower rate. Clearly the cost for this solution seems rather high. The architecture of the A/D must take advantage of the lower operating rate and

do so with significant hardware savings or the cost of two half-rate converters will be difficult to recoup.

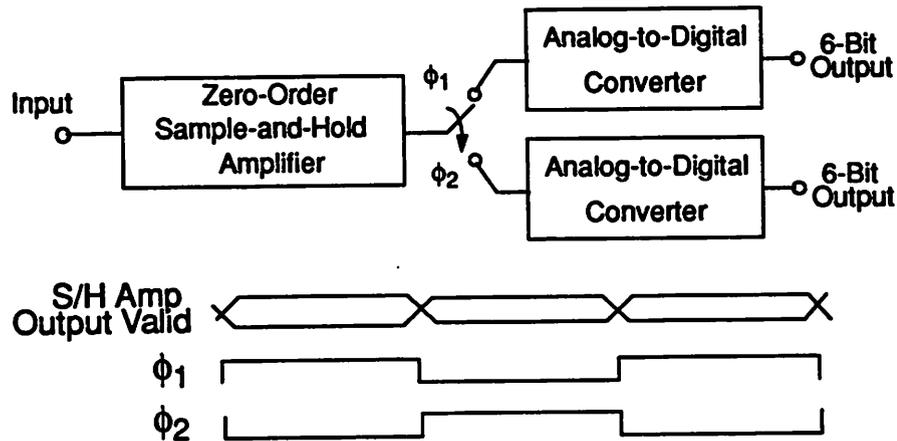


Figure 7.11 Interleaved ADCs share a single S/H amplifier.

Before presenting the ADC algorithm/architecture used in the prototype, it is useful to digress and examine circuit implementation of the *comparator* which is the key building block in an ADC. As mentioned earlier, pipeline ADC architectures allow the use of a technique called digital error correction which enables comparators to have rather large offset voltages. For example, in a pipeline converter employing digital error correction where 3 bits are resolved in the first stage, for a reference voltage of 3 V the comparators can have an offset of up to 187.5 mV. However, flash and subranging converters which have no interstage amplification need to have comparators with offset voltages below the resolution of the converter. For example, for a 6-bit converter and a reference voltage of 3 V, the comparators need offset below 23 mV.

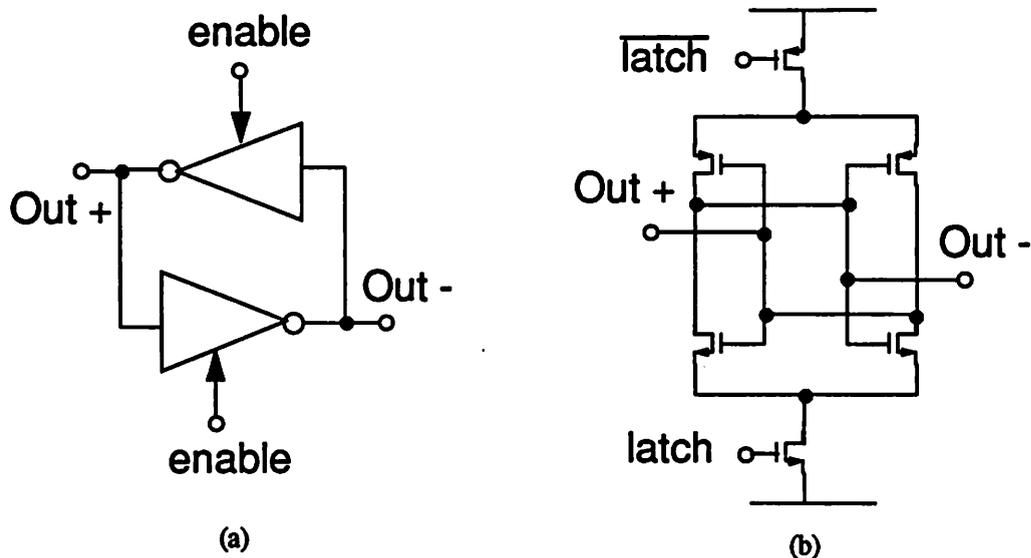
Pipeline ADC architectures tend to be used for resolutions in the 8-12 bit range. Although it allows comparators with somewhat large offsets, the complexity of the architecture makes it unattractive for only 6-bits of resolution. Our discussion of comparators will concentrate on approaches that will be good to the resolution of the converter at the 6-bit level.

### 7.3.3 Difference Amplifier – Regenerative Latch Comparators

There are many approaches for performing the comparison between the input voltage and the reference voltages. Choice of the comparator approach has important consequences from the stand point of the number of clock phases necessary for operation. CMOS logic levels including both power supplies consume no d.c. power and thus it is desirable for the comparator decisions to be the positive supply for logic 1 and the negative supply for logic 0.

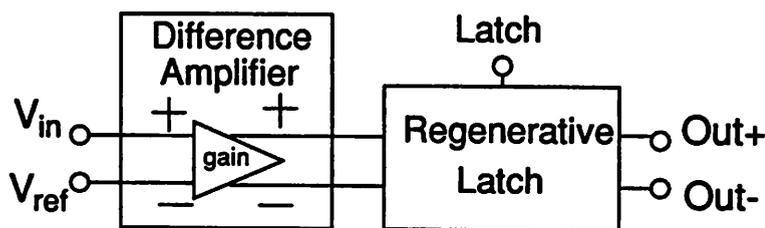
The function of the comparator is to compare the amplitudes of two voltages or currents and to output a logic 1 or 0 depending on whether or not the input signal is greater than or less than the reference signal (depending on the sense of the logic). We shall assume voltage comparison in the following as well as take the logic sense the if the input signal is larger (smaller) than the reference, the output is logic 1 (0). One approach is to use a difference amplifier -- regenerative latch comparator which achieves this function in two steps. It first generates and amplifies the difference quantity  $V_{in} - V_{ref}$ , then performs a *limiting* function on this difference which results in a logic 1 or 0 at the output, depending on the sign.

We shall first consider implementation of the limiter block with the use of a regenerative latch because of the consequences on the differencing circuit. The *regenerative latch* uses positive feedback to increase a small initial voltage difference to the supply rails in a relatively short time. A regenerative latch is shown functionally along with a schematic of a CMOS implementation in Figure 7.12. This circuit can be thought of as back to back inverters with enable switches connecting/disconnecting the inverter pair from the power supplies. Operation begins with disconnecting the inverter pair from the supplies. The output nodes Out+ and Out- are then initialized to the voltage difference between  $V_{in}$  and  $V_{ref}$ . The latch signals (*enable* in the conceptual schematic) are then activated and the positive feedback inherent in this circuit drives the outputs to the supply rails. Suppose  $V_{in}$  was larger than  $V_{ref}$  and these signals are connected to Out+ and Out-, respectively. Upon the *enable* signal becoming active, Out+ would rise to the positive supply while Out- would drop to the negative. This circuit clearly has two modes, an *initialize* mode and a *latch* or *evaluate* mode.



**Figure 7.12** Conceptual and circuit implementation of a latch. Signals *enable* and *latch* have the same logic sense. (a) Conceptual schematic. (b) CMOS implementation.

Device mismatch results in an input-referred offset for the latch as a result of threshold voltage offsets and  $W/L$  and  $\mu C_{ox}$  variations. As a result, the latch must be preceded by a circuit which amplifies the  $V_{in} - V_{ref}$  difference before initializing the latch to this quantity. This circuit must have enough gain so that the minimum  $V_{in} - V_{ref}$  difference to be resolved ( $1/2$  LSB) is amplified to a voltage large enough to overcome the input-referred offset of the latch. A block diagram of this scenario is shown in Figure 7.13 where a *difference-amplifier* block is shown driving the latch. This illustration is for a single-ended input signal and reference. These signals will be differential in the actual implementation.



**Figure 7.13** Difference-amplifier and regenerative latch combination.

Let us recall that the input signal comes from a zero-order hold circuit and assume the reference voltage comes from a resistor ladder which makes available evenly distributed voltages across the range from  $V_{ref+}$  to  $V_{ref-}$ . We shall consider two fully differential approaches for implementation of this circuit. In the first, a differencing amplifier is created using the differential pair amplifier and in the second, switched capacitor techniques are employed.

Shown in Figure 7.14 is a differencing amplifier made up of a pair of differential pair amplifiers in parallel. The inputs to one differential pair is the input signal and the other is the reference voltage. The outputs of each differential pair are summed at the output nodes resulting in a voltage which is a function of the difference between the differential inputs. Since only the sign of the difference is important, linearity is not an issue.

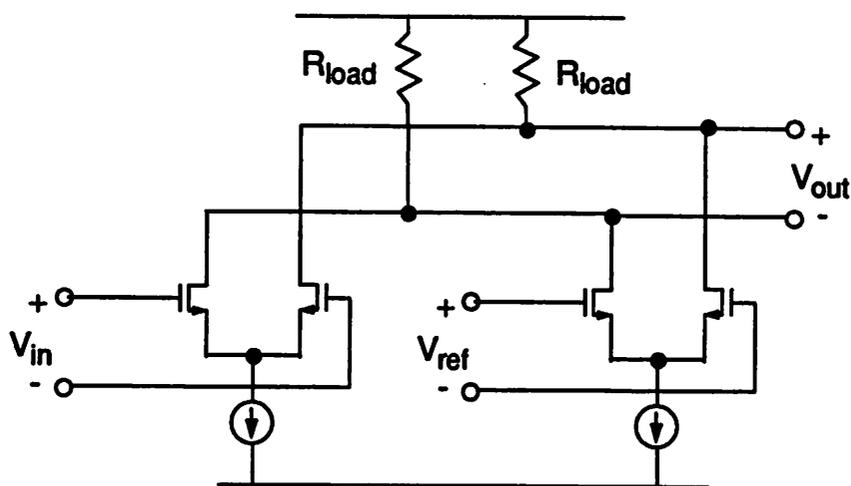


Figure 7.14 Difference amplifier using two differential pairs.

There are a number of disadvantages of this approach. The bandwidth of this difference amplifier is limited by the time constant at the output node. Having two differential pairs results in roughly one-half the bandwidth attainable if only one pair were used (The other solution to be discussed uses only one differential pair). Another problem is that this circuit is sensitive to the common-mode of the input and reference voltages which may result in a weighting difference between the input pairs and result in erroneous outputs. And yet perhaps the most important disadvantage is

that in a two-step or subranging converter, it is desirable to sample the input and pipeline the signal processing. This approach requires that the input signal always be present, i.e., the presence of a dedicated sample-and-hold which holds the input throughout the entire multi-phase conversion process. These disadvantages are overcome with the use of the switched-capacitor difference circuit shown in Figure 7.15.

This circuit operates in two phases. In the first, the input is connected to the two capacitors labelled  $C_{\text{sample}}$  and sampling switches M1 and M2 are turned ON. At the end of  $\phi_1$ , M1 and M2 are turned off and the input signal is sampled differentially across the two capacitors. During  $\phi_2$ , the capacitors are connected to the reference voltage pair resulting in a voltage difference proportional to the difference between the differential input signal and reference to appear at the differential pair input. Suppose the input signal was positive and the reference was smaller than the input signal. The result would be a negative deviation on node A relative to node B in the figure. The polarity of the amplifier is defined so that this condition results in a positive output voltage out of the difference amplifier.

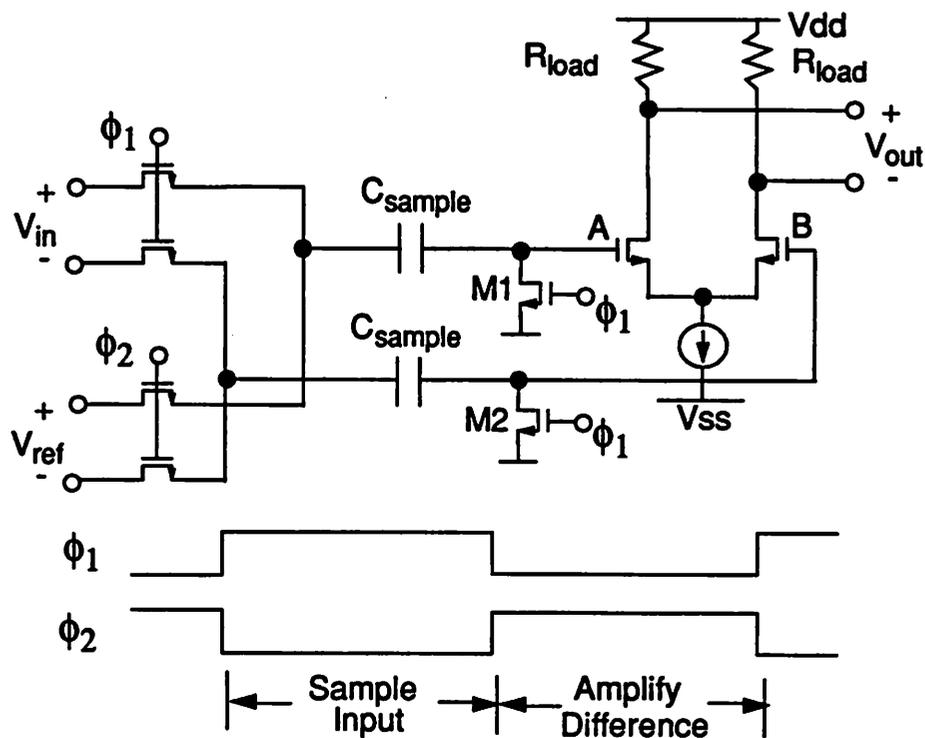


Figure 7.15 Switched-Capacitor Difference Amplifier

In this approach, only one differential pair is used and so a higher bandwidth is achievable as compared with the approach described previously. This also reduces the common-mode sensitivity because the subtraction is done in the charge domain on the amplifier input nodes as opposed to the differential pair outputs. Another important advantage of this approach is that by adding another pair of capacitors and switches, another level of subtraction is possible, i.e., to realize an equation such as  $V_{in} - V_{ref} - V_{ref2}$  where  $V_{ref2}$  is another reference voltage. The motivation for two layers of subtraction become clearer after the presentation of the algorithm used in the prototype. This approach is described in detail in Chapter 8.

In summary, practical comparator blocks made up of a difference amplifier and regenerative latch require at least three phases of operation. There is the sampling mode where the input voltage is sampled, the difference mode where a voltage proportional to the difference of the input signal and a reference is generated and precharges a regenerative latch, and the latch mode where the difference is made into one of the two logic levels by a regenerative latch. These functions can be pipelined such that while the present input is being sampled onto the sampling capacitors, a decision on the previous input is made by the regenerative latch. The key point to take into the next section is that two phases are required by the front-end of the comparator. A sampling phase and a differencing phase.

### 7.3.4 Key Implementation Considerations

Let us now examine the ADC algorithm used in the prototype. Recall that each of the three interleaved amplifier blocks equalizer will drive its own ADC block. We will present the algorithm used by each of the ADC blocks. The key considerations for the approach include:

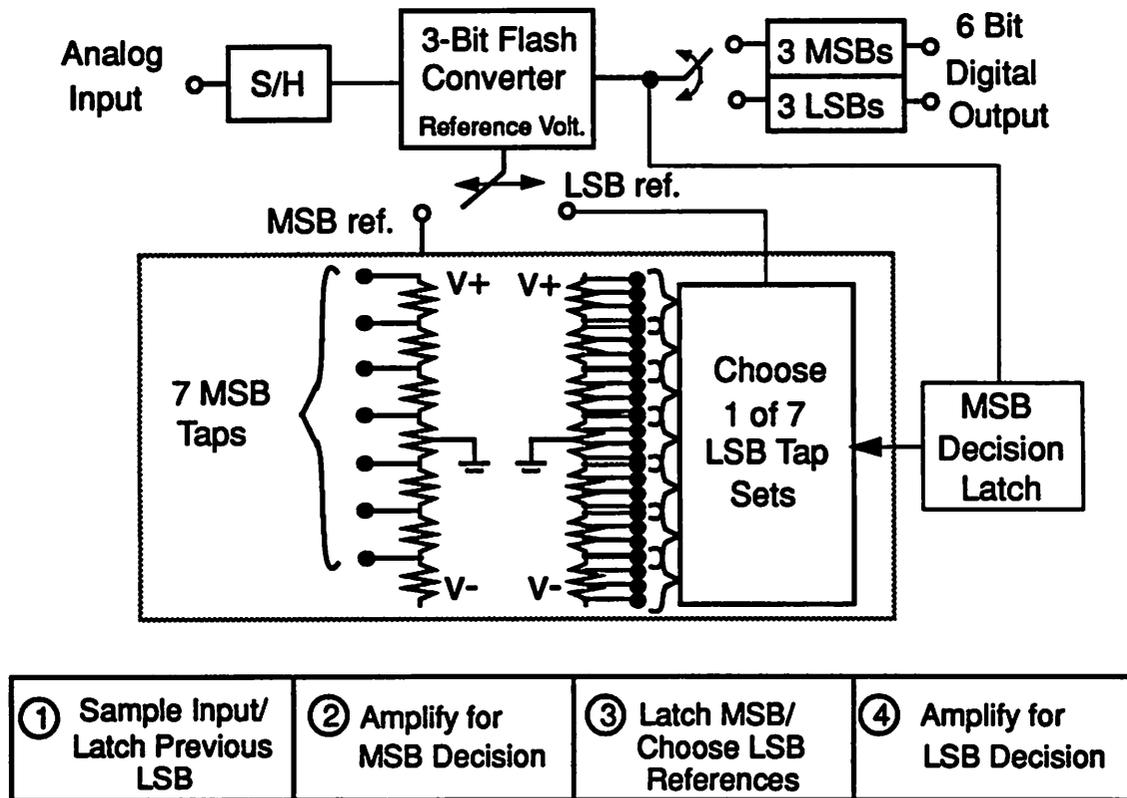
- Allow previous stage amplifier (Equalizer amp) 3 effective ADC output periods to settle. Sampling circuit in the ADC needs to be connected for the entire duration.
- No interstage amplification (Ruling out pipeline approach).
- Avoid a full flash architecture which requires 63 comparators and references resulting in a large input capacitance. Also difficult to perform differencing function other than with two interleaved flash converters which would be overly excessive in terms of hardware.
- Use difference-amplifier and regenerative latch for reliable high performance comparators.

- Allow maximum time for the comparators to make decisions to minimize the probability of metastable conditions at comparator outputs.

In light of the considerations above, the 2-step approaches become rather attractive. Breaking up the conversion into two 3-bit conversions means a total of  $2 \times 7$  or 14 comparators as compared with 63. Practical considerations increase this number to 28 in the actual implementation which is still quite a savings of hardware. Another important benefit of realizing the A/D conversion 3 bits at a time is that the input capacitance into the A/D converter is that associated with 7 comparators as opposed to 63 as present in the flash architecture.

### 7.3.5 Highly Parallel Implementation

Before presenting the implementation approach used in the prototype, we shall first examine an approach using the difference-amplifier and regenerative latch circuit in the subranging architecture described earlier and illustrated in Figure 7.16. The algorithm used in the prototype is very similar to this approach. The block labelled 3-Bit Flash Converter contains 7 difference-amplifier and regenerative latch circuits. Three periods are required for the first 3-bit comparison. The first period is the sampling mode in which the S/H amplifier drives the sampling capacitors in the difference-amplifier front-end. The second period is the difference-amplify mode, where the MSB references are subtracted from the input signal. During this period, the 7 difference voltages are amplified and used to precharge the regenerative latches. Following this is the third period which is the MSB latch mode where the MSBs are decided. During the fourth period, the LSB references above the MSB decision are multiplexed to the *reference* input of the 3-Bit Flash Converter by the *Choose 1 of 7* block and subtracted from the input signal. These difference voltages are amplified and used to precharge the regenerative latches. Then during the first period of the next cycle (while the next input is being sampled), the LSBs are latched. The regeneration and latching process can occur while a new input is being sampled because the difference-amplify and regenerative latch functions can easily be pipelined with transmission gates.



**Figure 7.16** Block Diagram of Two-Step A/D Converter using subranging ADC architecture.

Two resistor strings are shown for clarity (One will suffice). Flash converter block contains 7 comparators.

The major disadvantages of this approach is that all 63 reference voltages must be routed throughout the A/D converter and it is difficult to implement the LSB reference multiplexor in an efficient manner. Particularly where 3 ADC blocks are required, routing and multiplexing the reference voltages can be rather inconvenient.

Another approach is to use the two-step pipelining approach which takes a detour in the algorithm after the MSB decision as shown in Figure 7.17. In this approach, the voltage corresponding to the MSB decision is subtracted from the input signal. This results in a residue that is always between 0 and a level one LSB below the MSB above zero. A key advantage of this approach is that only a total of 14 references needs to be routed throughout the converter using this approach

because the residue is always in a known range. Thus the same LSB references can always be used for the second conversion.

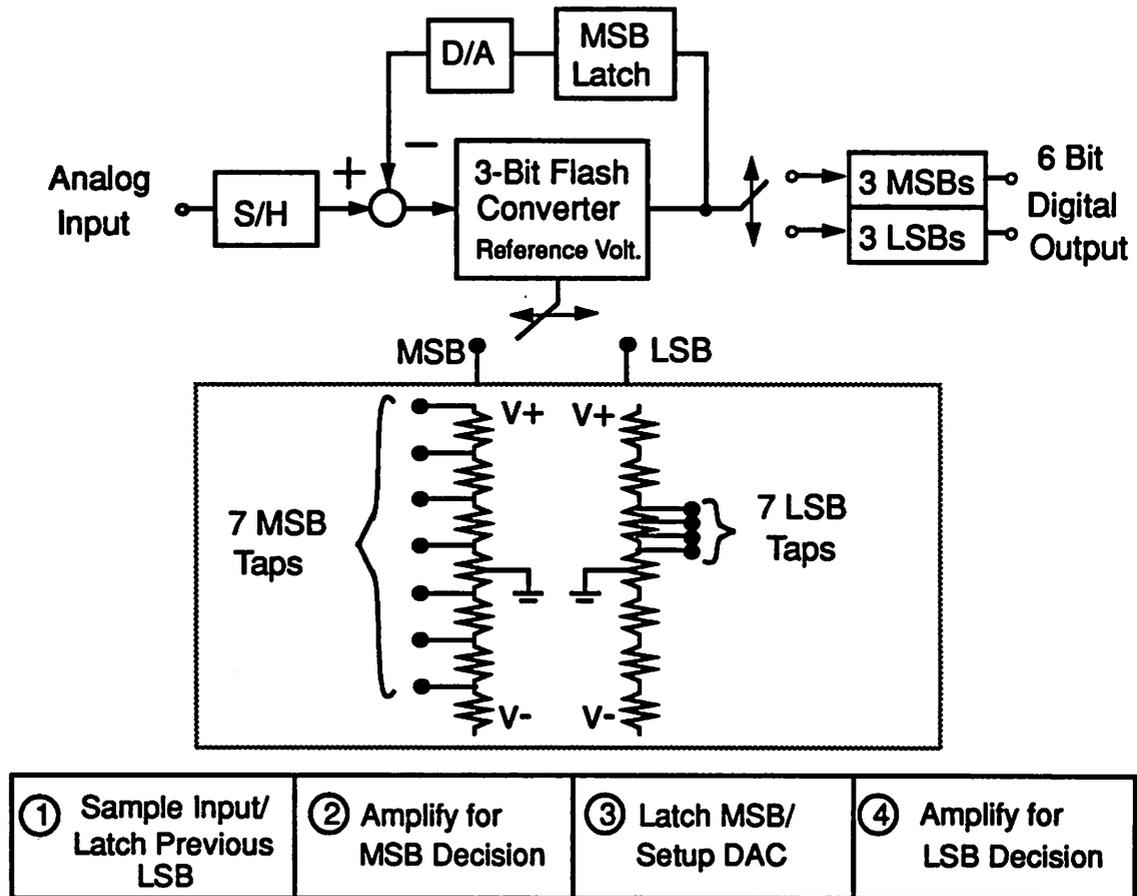


Figure 7.17 Block Diagram of Two-Step ADC algorithm used in the prototype. Two resistor strings are shown for clarity (One will suffice). Algorithm works over 4 periods.

Let us recall the discussion of the switched-capacitor front-end of the difference-amplifier and regenerative latch block. The motivation for two levels of subtraction can now be made clear. With this algorithm, during the LSB difference generation the signal presented to the input of the amplifier is  $V_{in} - V_{msb\_decision} - V_{lsb\_reference\_k}$  for comparator  $k$ .

Let us refer to the hardware implementing the algorithm shown in Figure 7.17 as a *SubConverter*. Each SubConverter contains 7 comparators and the associated hardware. The algorithm has four distinct periods. Each of these periods is the length of the settling time of the S/H amplifier

which precedes the ADC. In order to realize a conversion at the input rate, the ADC block associated with each S/H amplifier is realized with four SubConverters implementing the algorithm described in Figure 7.17. Each of the 4 SubConverters operates in one of the four distinct phases as shown in Figure 7.18.

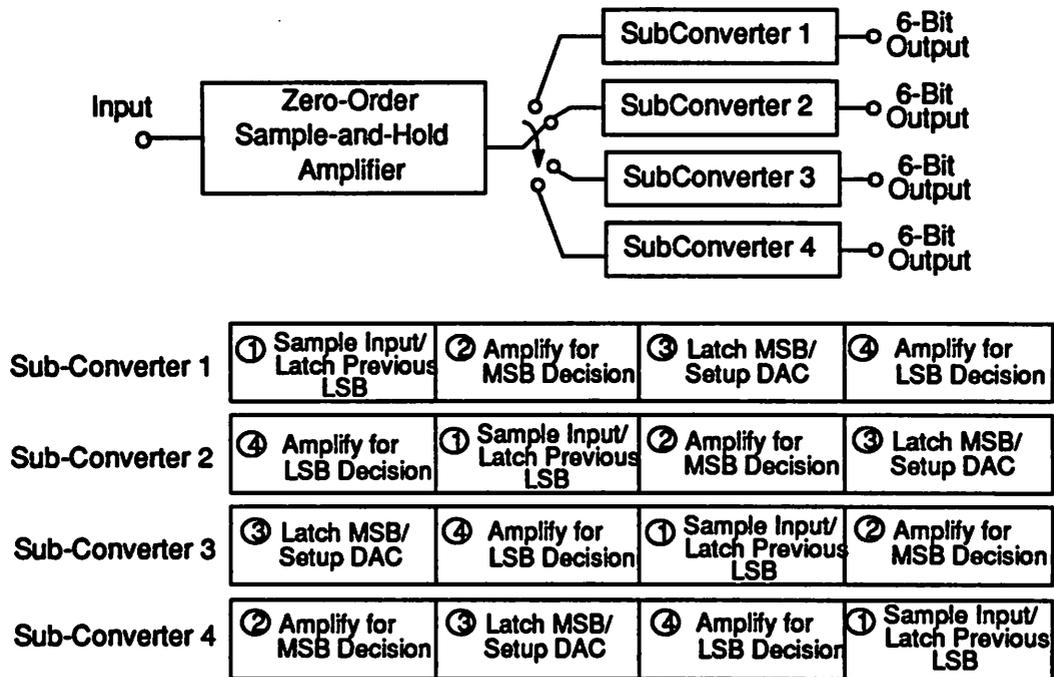


Figure 7.18 Four SubConverters share one S/H Amplifier. Each operates in 1 of the 4 algorithm phases.

The programmable equalizer stage preceding the A/D converter is realized with 3 parallel sections. Each of these sections drives an ADC block comprised of 4 sub-converters. Therefore, the composite A/D converter is realized with a total of 12 SubConverters operating in each of 12 phases as shown in Figure 7.19. The result is a total of 84 comparators and 14 reference voltages (28 differential lines) which must be passed throughout the converter.

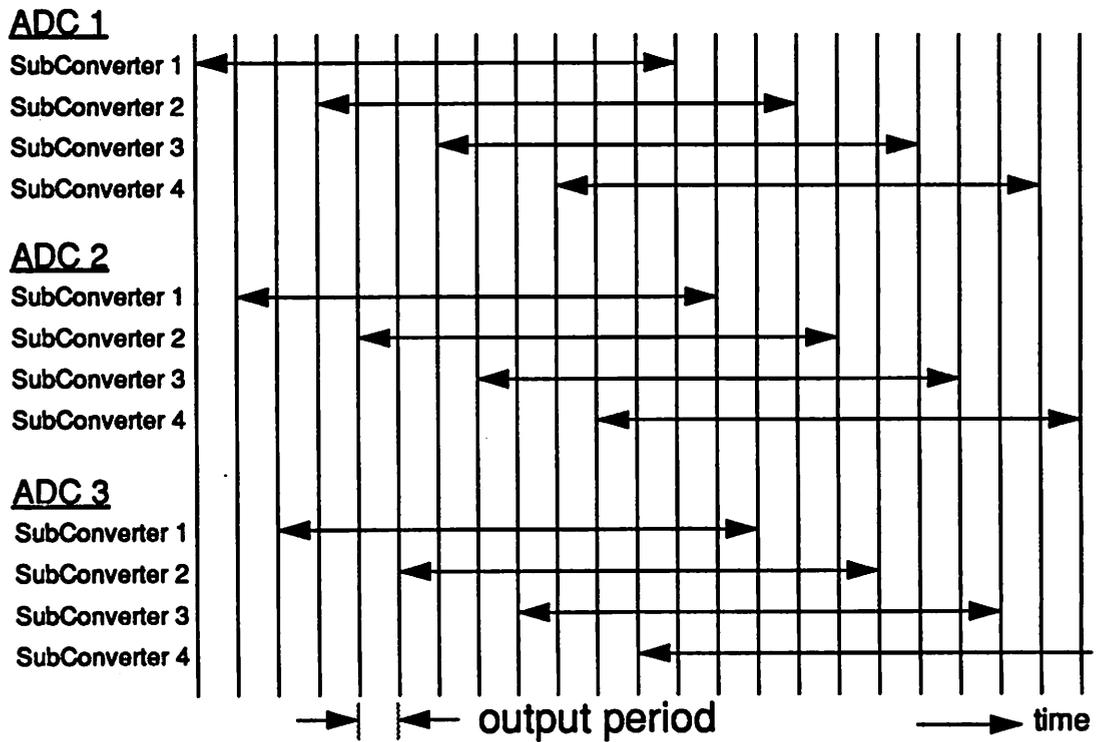
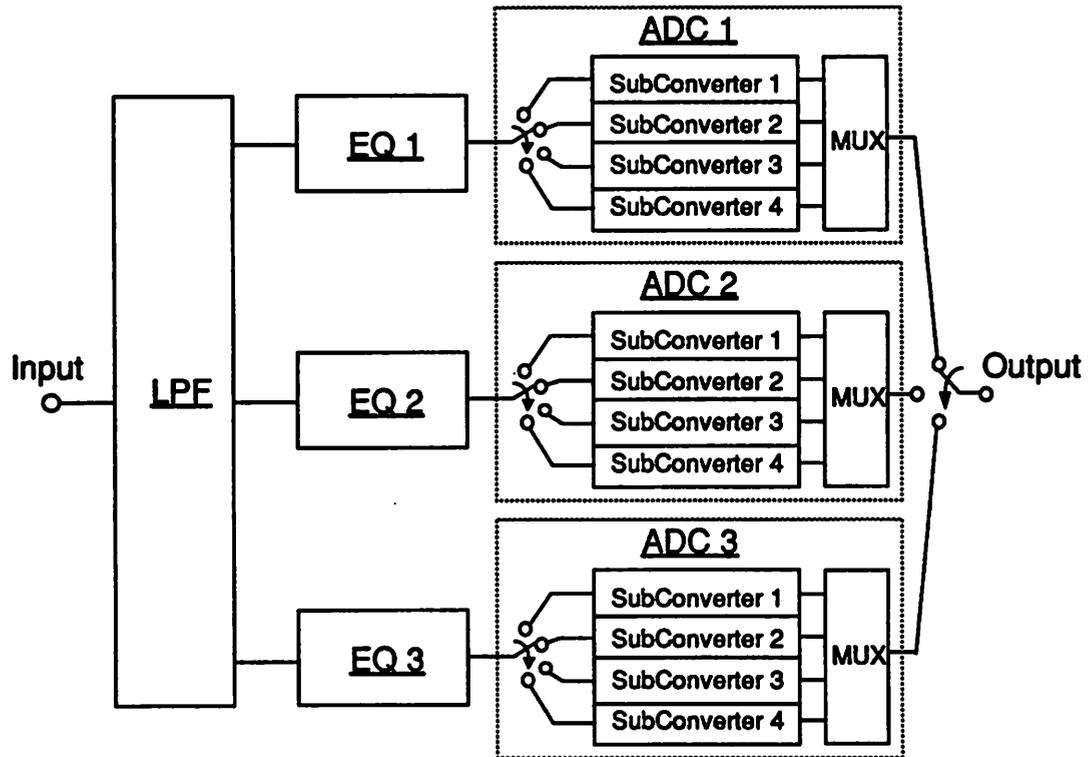


Figure 7.19 Block diagram of A/D Converter with diagram illustrating interleaved approach.

This is the approach used in the prototype. The high degree of parallelism allows each of the operating periods to be 3 output periods long. Therefore, all the analog functions associated with this converter including settling of the equalizer amplifiers driving the input capacitance of the converters; subtraction, amplification, and precharging of the regenerative latch; actual regeneration and latching of the latch outputs; are all given 3 output periods to perform their functions. This means that for a 100MHz output rate, all processing functions are allocated 30 nsec instead of just 10 nsec for an all-serial approach. This time advantage is especially important when the time overhead of non-overlapping clocks is factored in. For example, for serial processing at 100 MHz, 3 nsec of non-overlapping clock overhead is 30% of the 10 nsec period leaving only 7 nsec for circuit operation. Whereas 3 nsec is only 10% of 30 nsec, allowing another 27 nsec for processing time. This is particularly important for amplifier settling and the occurrence probability of metastability in the comparators.

The circuit implementation details of the analog-to-digital converter are presented in the prototype description in Chapter 8.

## 7.4 Summary

We started this chapter by presenting some key considerations for implementation of high speed ADCs and then focusing on the requirements of the ADC required in the prototype. One of the key implementation considerations for an ADC which is part of a signal processing IC is the way in which it interfaces to the other blocks. Since the ADC required in the prototype gets its input from the programmable equalizer stage employing the parallel filter structure, a high degree of parallelism is used in the ADC implementation as well. We then examined some candidate architectures for high speed converters and decided that the two-step approaches would be best for the current application.

One approach for implementation of the difference amplifier -- regenerative latch which is a key building block in two-step ADCs was presented. This block was then placed in the context of both the subranging and the two-step pipeline architectures. The two-step pipeline was chosen pri-

marily because it requires only a fraction of the voltage references to be routed throughout the converter, reducing both area and parasitic capacitance. The ADC architecture has a high degree of parallelism. In fact, the 100 MHz A/D converter is realized with 12 8.333 MHz converters, each operating in a different phase.

Circuit level implementation issues for the analog-to-digital converter will be presented in detail in the section on the A/D Converter in Chapter 8.

## References

- [1] P. Real, D. H. Robertson, C. W. Mangelsdorf, and T. L. Tewksbury, "A Wide-Band 10-b 20 MS/s Pipelined ADC Using Current-Mode Signals," *Journal of Solid-State Circuits*, Vol. 26, No. 8, pp. 1103-1109, August 1991.
- [2] B. Del Signore, D. Kerth, N. Sooch, E. Swanson, "A 20b Delta-Sigma A/D Converter," *1990 ISSCC Digest of Technical Papers*, vol. 33, pp. 170-171, February 1990.
- [3] N. S. Jayant and P. Noll, *Digital Coding of Waveforms: Principles and Applications to Speech and Video*, Prentice-Hall, 1984.
- [4] U. Fiedler and D. Seitzer, "A High-Speed 8 Bit A/D Converter Based on Gray-Code Multiple Folding Circuit," *IEEE Journal of Solid-State Circuits*, Vol. SC-14, No. 3, pp. 547-551, June 1979.
- [5] B. Zojer, R. Petschacher, and W. A. Lushnig, "A 6-Bit/200 MHz Full Nyquist A/D Converter," *IEEE Journal of Solid-State Circuits*, Vol. SC-20, No. 3, pp. 780-786, June 1985.
- [6] C. W. Mangelsdorf, "Improving Bandwidth and Error Rate in Flash Converters," *1989 Symposium on VLSI Circuits Digest of Technical Papers*, pp. 53-54, 1989.
- [7] T. Wakimoto, et. al., "Si Bipolar 2 GS/s 6-b Flash A/D Converter LSI," *ISSCC Digest of Technical Papers*, pp. 232-233, February 1988.
- [8] J. Doernberg, "A 10b 5MSample/s CMOS Two-Step Flash ADC," *IEEE Journal of Solid-State Circuits*, Vol. SC-24, pp. 241-249, April 1989.
- [9] S. H. Lewis and P. R. Gray, "A Pipelined 5-Msample/s 9-Bit Analog-to-Digital Converter," *IEEE Journal of Solid-State Circuits*, Vol. SC-22, No. 6, pp. 954-961, December 1987.
- [10] B. Razavi and B. A. Wooley, "A 12b 5 MS/s Two-Step CMOS A/D Converter," *1992 Digest of Technical Papers, ISSCC*, San Francisco, pp. 36-37, February 1992.
- [11] M. W. Hauser, P. J. Hurst, and Robert W. Brodersen, "MOS ADC-Filter Combination That Does Not Require Precision Analog Components," *ISSCC Digest of Technical Papers*, pp. 80-81, February 1985.
- [12] J. C. Candy, "A Use of Double Integration in Sigma Delta Modulation," *IEEE Transactions on Communications*, Vol. COM-33, No. 3, pp. 249-258, March 1985.

## CHAPTER 8

# Prototype Description and Design

---

### 8.0 Introduction

In this chapter, we will take a detailed examination of a prototype IC demonstrating the architectures proposed in Chapter 5 for high speed switched-capacitor FIR filters and Chapter 7 for a 6-bit analog-to-digital converter. Both a decimation filter and a programmable equalizer were implemented using the new high speed filter architectures. The performance targets of the prototype are compatible with those of the magnetic disk drive read channel employing sampled-data PRML techniques. In particular, functions required by the analog-to-digital interface in the read channel were addressed.

In Figure 8.1, a block diagram of the prototype is shown. The first block is a non-critical programmable cascade of second-order lowpass filter sections which provides anti-alias filtering for the discrete-time analog filter which follows. There are 3 bits which control the cutoff frequency of this filter. Following this is a 3-tap 3:1 decimation filter using the FIR architecture described in Chapter 5. The parallel outputs of the SC lowpass filter feed a programmable 3-tap raised cosine equalizer which provides 3 degrees of high frequency boost. This symbol-rate equalizer also

employs the parallel FIR structure. The operational amplifiers within the equalizer act as the sample-and-hold amplifiers for 3 parallel 6-bit 2-step analog-to-digital converters. An on-chip ring oscillator is slaved to an external reference clock at one-half the output rate. All clocks are generated on-chip derived from different phases of the ring oscillator.

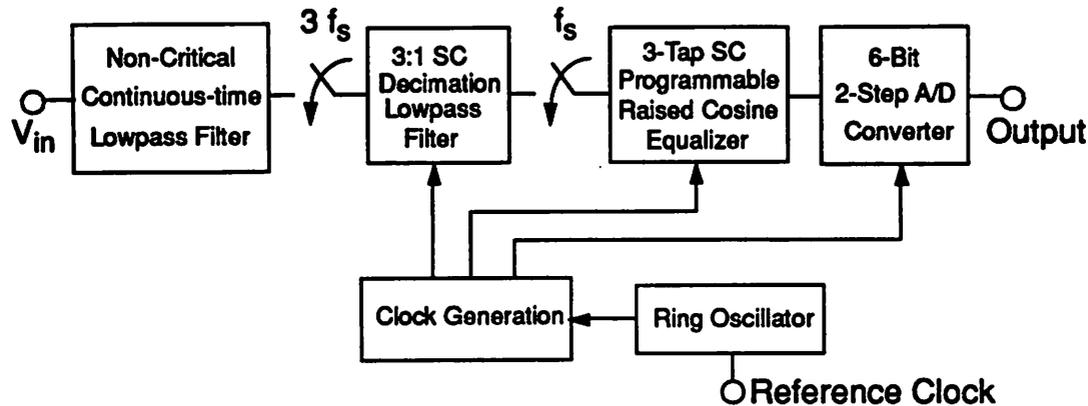


Figure 8.1 Block diagram of prototype IC.

The continuous-time filter and the SC decimation filter were designed for a composite response that compares with fifth-order Bessel and Butterworth responses. The cosine equalizer with 3-tap impulse response  $-K, 1.0, -K$  has programmable  $K$  values of .25, .35, and .45. In an actual read channel, the ring oscillator would be in a timing recovery loop driven by a decision-directed algorithm operating on equalized samples in the digital domain. In the prototype, a fixed external clock is used so that the integrity of the signal path can be evaluated as a pre-conditioning filter and A/D converter.

From an architecture and implementation standpoint, some of the key ideas demonstrated by the prototype are as follows. First, implementation of the high speed lowpass anti-alias filter using a composite filter comprised of a non-critical continuous-time filter followed by a decimation filter using the new parallel filter architecture allows CMOS filter bandwidths that are much larger than those attainable using conventional filtering techniques. Second, inclusion of the raised cosine equalizer before the ADC (using the parallel filter architecture) is of interest both from an implementation and an architectural standpoint. The signal preconditioning performed by this equalizer in the analog domain has been shown in Chapter 4 to result in a lower  $S/N+D$  in the presence of

quantization effects after the adaptive digital filter converges making the high frequency boost provided by this equalizer before the ADC important from a system standpoint. Implementation in discrete-time using SC techniques results in a frequency response that is primarily a function of capacitor ratios. Continuous-time implementation of these equalizers are very sensitive to process variations and require very accurate tuning loops which may be quite complicated. Third, generation of the sampling waveforms required for a 300 MHz sampling rate (100 MHz output rate) meeting sampling jitter and offset specifications were implemented.

Design considerations for each of the blocks will be presented in this chapter along with detailed design descriptions. The prototype was designed in a 1.2 $\mu$ m CMOS double-metal double-poly process. The prototype was designed to operate at a 100 MHz output rate with an output SNR of 30 dB at frequencies up to one-half the output rate. The input signal peak amplitude is assumed to be 4 V differential.

## 8.1 Continuous-Time Anti-Alias Filter

The key function of this block is to provide anti-alias filtering with a linear phase response for the discrete-time SC decimation filter which follows. An input buffer is also contained in this circuit block to provide a high input impedance at the prototype input. When running at full designed speed, the sampling frequency of the SC decimation lowpass filter is 300 MHz with an output rate of 100 MHz. Although the noise spectrum from the magnetic medium rolls off at high frequencies, high frequency noise exists in the signal due to coupling from high speed electronics present in the system. Thus, attenuation is still required at high frequencies somewhere in the signal path prior to sampling. The SC decimation filter provides the lowpass channel shaping and the continuous-time filter provides attenuation near the sampling rate of 300 MHz. The response of the continuous-time anti-alias filter is programmable with 3 external bits of trim providing 8 different filter settings. The bandwidth trim range is roughly 4-1. The programmable response is necessary in order to demonstrate the feasibility of this approach in channels employing zone-bit recording where the data rates off the disk can vary as much as 2-1.

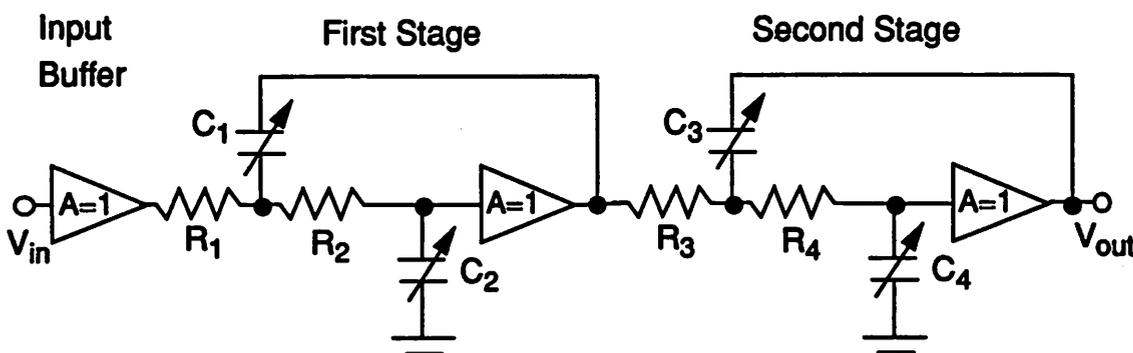
One of the key problems with continuous-time filter implementation for frequencies in the tens of mega-hertz is that it is difficult to control the frequency response. However, the continuous-

time filter required by the decimation filter is non-critical and needs only to provide filtering at frequencies in the vicinity of 200 MHz and beyond. Therefore, second-order filter sections with low filter  $Q_s$  can be used to meet the system requirements.

### ***Implementation***

A cascade of two Sallen-Key second-order sections were chosen for this function. The second-order response is important since it allows a linear phase response. The bandwidth of the first section was designed to be slightly less than that of the second in order to minimize effects of the composite filter on the passband. An advantage of the Sallen-Key structure for this application is that a voltage follower can be used as the active element. As a result of the use of voltage followers, the differential structure was realized with two parallel stages and thus no common-mode rejection occurs in this block.

A schematic of the filter used is shown in Figure 8.2. The capacitors are realized by binary weighted capacitor arrays which provide programmability of the response. Capacitor arrays were used instead of resistor arrays. It was found through simulation that for the filter topology shown in the figure, the effect of the parasitic series impedance of the transmission gates is worse using fixed capacitors and programmable resistor arrays on both the frequency response and distortion than if fixed resistors and programmable capacitors are used.



**Figure 8.2** Schematic diagram of continuous-time lowpass filter.

A detailed schematic of the implementation used in the prototype is shown in Figure 8.3. Devices M1-M3 and one of the 2 mA current sources make up a unity gain follower that overcomes the gain loss due to body effect and an otherwise present signal level-shifting problem associated with the use of a simple source follower. The input signal goes down a  $V_{gs}$  and back up again in M2 resulting in a gain of about unity. The cost of this circuit is reduced bandwidth due an increase of the output impedance to  $2/g_m$ . This effect is made small by using NMOS only devices and running the devices at rather large currents.

In the figure, each of the programmable capacitors is shown to be 3 binary weighted capacitors with associated switches. A box with an "x" denotes a CMOS transmission gate and a single device denotes an NMOS pass-gate. Note in the figure that the capacitor array for  $C_2$  uses only NMOS transistors in series with the capacitors. This is because of the large  $V_{gs}-V_t$  possible (and resultant low on-resistance) for the switches since the connection is to Vss. The capacitor array for  $C_1$  was configured as shown because it minimized the parasitic capacitance at node A and isolates the node from the array when the switches are off. The large parasitic capacitance associated with the array is now connected to the output node which is the least sensitive node in the filter section. Polysilicon resistors have been used and provision was made to account for the large variations expected in the absolute resistance over processing variations.

The gain through the continuous-time filter is less than one due to the absence of the level shifting device in output buffer device M8. This trade-off was made in order to maximize the bandwidth and drive capability of this device since it drives the high-rate sampling capacitor arrays in the SC FIR filter which follows. A dedicated current source biases this circuit independently from the rest of the chip to facilitate testing. In this way, current can be changed in this block independent of the rest of the chip.

The power consumed by this circuit is nominally 105 mW and the active area is roughly 2 k square mils.

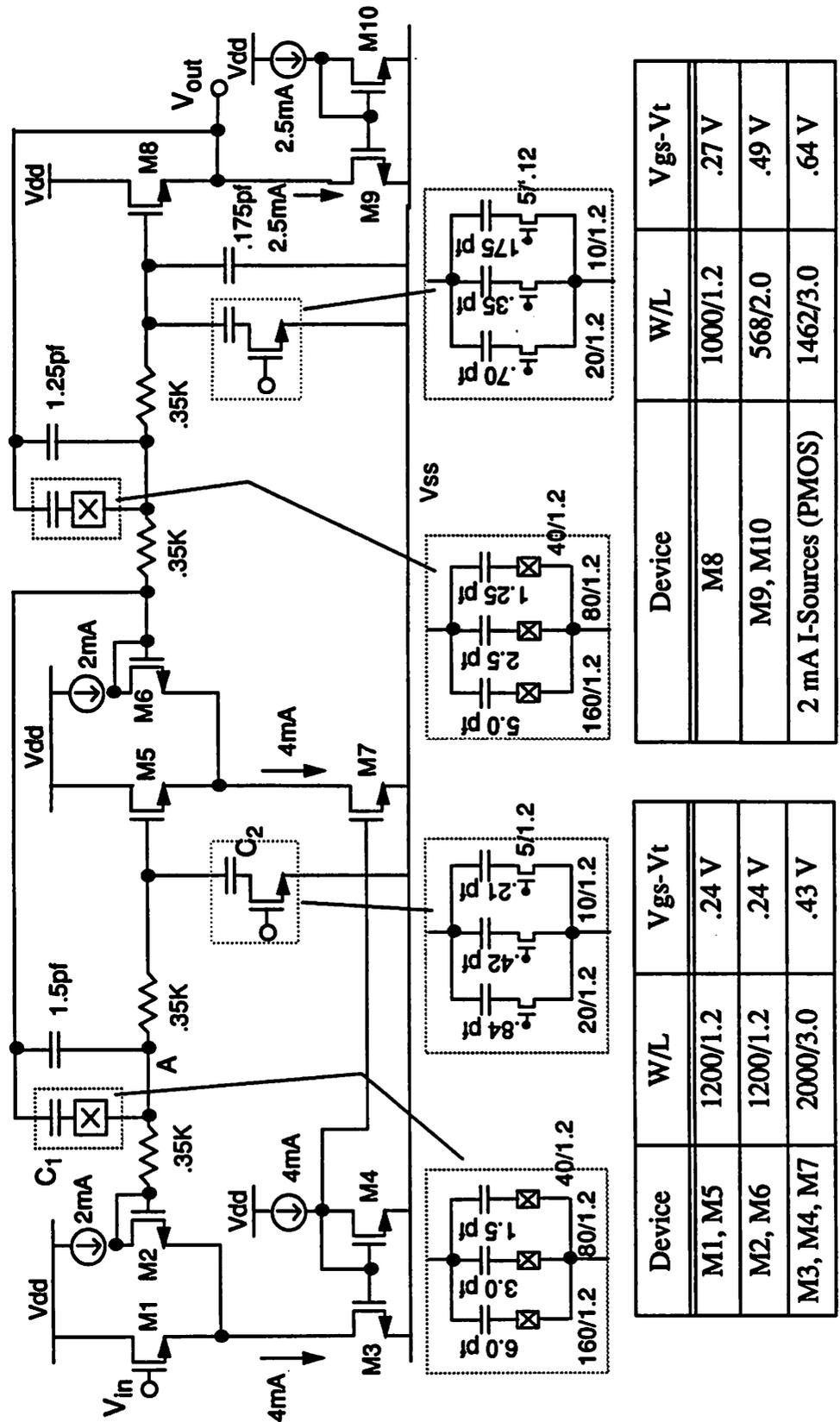


Figure 8.3 Schematic of one half of the continuous-time filter.

## 8.2 Operational Amplifier and Analog Bias Generation

Before continuing on the through the signal path, let us digress with a discussion of the operational amplifier and its associated bias voltage generator along with techniques that were employed in order to obtain high speed operation.

### *Amplifier Gain and Bandwidth*

Finite opamp gain results in an error term in the closed loop gain of a feedback circuit with the end result being a gain error through the circuit block. This gain error is detrimental in applications such as pipelined A/D converters where the gain accuracy must be better than the number of bits yet to be resolved at any point in the pipeline, and in switched-capacitor recursive filters where gain errors affect the location of poles and zeros and thus the overall filter response. In the case of transversal filter structures, since there is inherently no feedback, gain errors appear only as a gain error through the filter which has no consequence on the filter frequency response. However, the switched-capacitor zero-order hold approach used in the prototype is in fact sensitive to finite amplifier gain. The effect of finite gain is a zero-order hold with a droop at high frequencies. The overall effect on the frequency response of the filter is to fold in a lowpass response into the overall frequency response of the channel and should not be much of an issue. Far more important in the parallel FIR approach is channel-to-channel gain matching. The opamp gain must be large enough so that channel-to-channel closed loop gain mismatch as a result of opamp gain mismatch is sufficiently small. In Chapter 5, the effect of gain mismatch on filter performance was examined. Amplifier gain on the order of around 500 should be acceptable for this application.

The amplifier bandwidth needs to be maximized in order to achieve a high filter throughput. A conceptual block diagram of the amplifier in its closed-loop configuration is shown in Figure 8.4a. The amplifier core is made up of a wideband preamp followed by a telescopic transconductance amplifier. Under the condition that the preamp bandwidth is sufficiently high such that the composite amplifier appears as a single transconductance amplifier, the settling time of the closed-loop amplifier is inversely proportional to the total load capacitance. In the scenario of Figure 8.4, both capacitance  $C_{sam2}$  and the loading of the common-mode feedback network contribute to the total load capacitance  $C_{load}$  on the amplifier.

The approach shown in Figure 8.4b was employed to remove the amplifier settling time dependency on the load capacitance. In this case, the outputs of the telescopic amplifier drive the feedback network along with unity gain buffers which drive the capacitive load as well as the common-mode feedback network. In this way, the amplifier core is nested in a tight feedback path and buffered from the capacitive load. The key advantage of this approach is that the potentially large load capacitance of this circuit comprised of the sampling network of the following stage and the parasitic capacitance associated with interconnect lines are driven by the buffer and the feedback amplifier is allowed to settle independent of this load. The load capacitance of the feedback amplifier is merely the input capacitance of the buffer. This approach facilitates the use of switched-capacitor based common-mode feedback techniques without the concern of the additional capacitive load to the amplifier normally associated with these techniques.

The key disadvantage of this approach is that the differential offset voltage of the voltage followers adds directly into the signal path. However, by using large transistors as dictated by the bandwidth requirements, differential offsets in the 2-3 mV range can be achieved which has little impact on a signal path with requirements of only 6 bits of accuracy.

### **Amplifier and Bias Schematics**

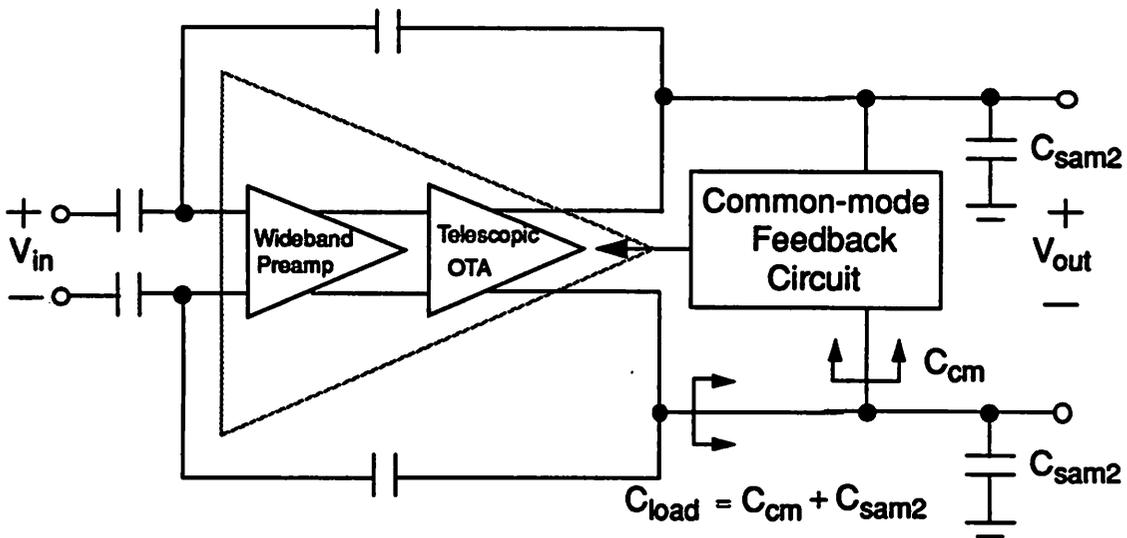
A detailed schematic of the amplifier and bias together with device sizes is shown in Figure 8.5. The amplifier architecture is a wideband preamp preceding a telescopic transconductance amplifier as discussed in Chapter 6. The preamp gain is set by the ratio of the input device to the load device transconductance and has been designed for a gain of about 3. The bias voltage to the load devices  $pa\_lb$  sets the input common-mode level into the main telescopic amplifier. Transistor M3, the current source for the preamp differential pair has a longer channel than M15 because its output impedance is more critical for common-mode rejection. The common-mode input to a telescopic OTA is usually quite critical because if it is too low, the tail current source device output impedance can drop off and if it is too high, the input device pair can be driven into the linear region by the cascode devices or the voltage swing can be compromised. The preamp has a larger input common-mode range since there are no cascode devices directly above the input source-coupled pair. Thus, this composite amplifier has a wider common-mode input range than a telescopic amplifier by itself.

A single bias voltage generator was shared by 6 amplifiers, three in the lowpass filter and three in the equalizer. Note in the telescopic amplifier that the bias voltage for the NMOS cascode devices are generated inside the opamp. There are two reasons this was done. First, the cascode devices are in the signal path and cross-talk could occur between amplifiers through the cascode bias voltage if they were all connected. Second, since the bias voltage is referenced to the center point of the source-coupled pair, it tracks movement of the input common-mode.

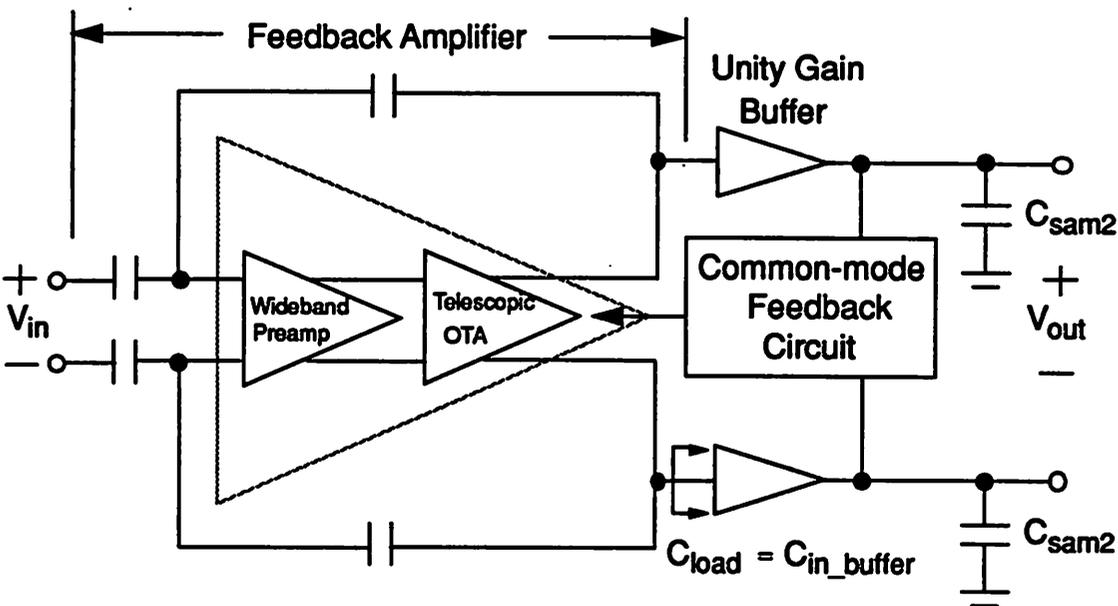
As described earlier, the voltage followers (M27 and M28) are not in the feedback path but drive the capacitive load and the common-mode feedback network. The common-mode feedback circuit is the standard switched-capacitor approach used in switched-capacitor filters [2].

There are two things in the bias generator that we shall discuss. Referring to the figure, let us consider the generation of the PMOS cascode bias *pb2*. Four PMOS devices are shown stacked with their gates all connected to the drain of M74. A simplified version of this circuit would be two stacked PMOS devices with both gates connected the drain of the lower. With a current source at the bottom, the bottom PMOS device is in saturation and the top device is in the linear region. With appropriate sizing of the devices and current, a voltage can be generated which is the sum of the  $V_{gs}$  of a saturated device and the voltage drop across the linear region device. This is useful in a high swing cascode bias situation where a voltage equal to the sum of a  $V_{gs}$  of a saturated device and a fixed voltage equal to the  $V_{ds}$  to be forced across the current source is needed [1]. In order to get the desired fixed voltage across the current source device, transistors with non-minimum channel lengths were required. However, since only devices with channel lengths of 1.2  $\mu\text{m}$  had been fully characterized, four devices with 1.2  $\mu\text{m}$  channel lengths were stacked as shown in the figure.

Referring to the figure once again, consider now the generation of current source bias *pb1* made up of devices M75 and M76. Transistor M76 mere sets the  $V_{ds}$  across M75 to mirror that which are seen by the current source devices in the amplifier. By connecting the gate of M75 to the drain of M76, the gain voltage settles to the voltage required by the PMOS device to carry the desired current at the  $V_{ds}$  set by M76. This is current matching in the presence of relatively low output resistance in the current source devices as present in short channel MOS transistors.



a. OTA in closed-loop configuration.



b. Inclusion of buffers to drive the load capacitance and common-mode feedback network.

Figure 8.4 Closed-loop amplifiers with (b) and without (a) buffers to drive capacitive loading.

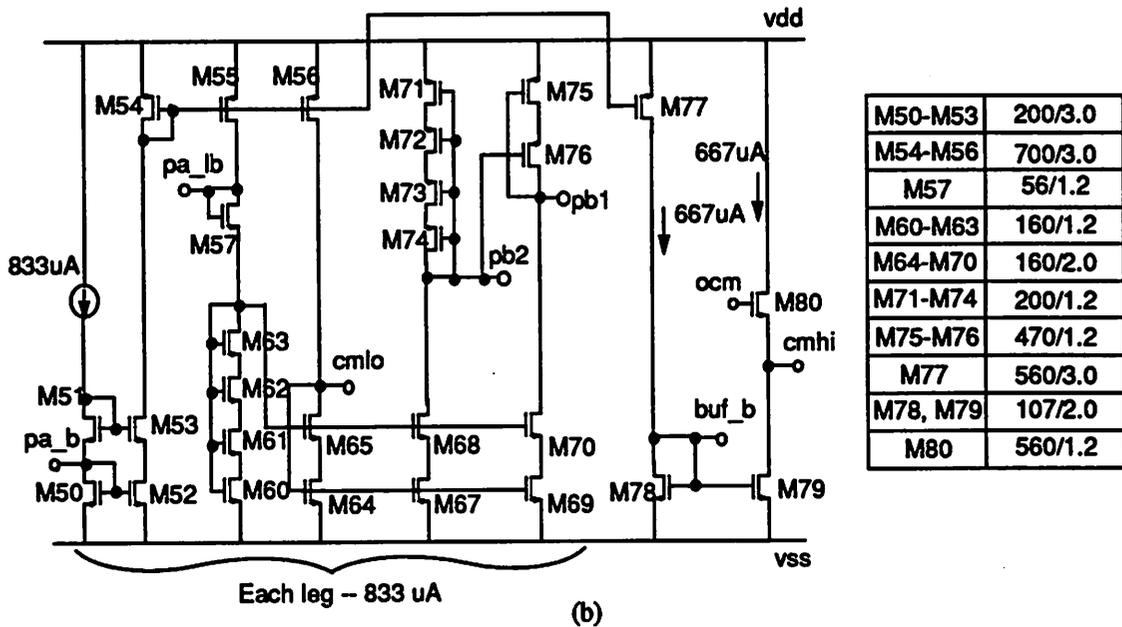
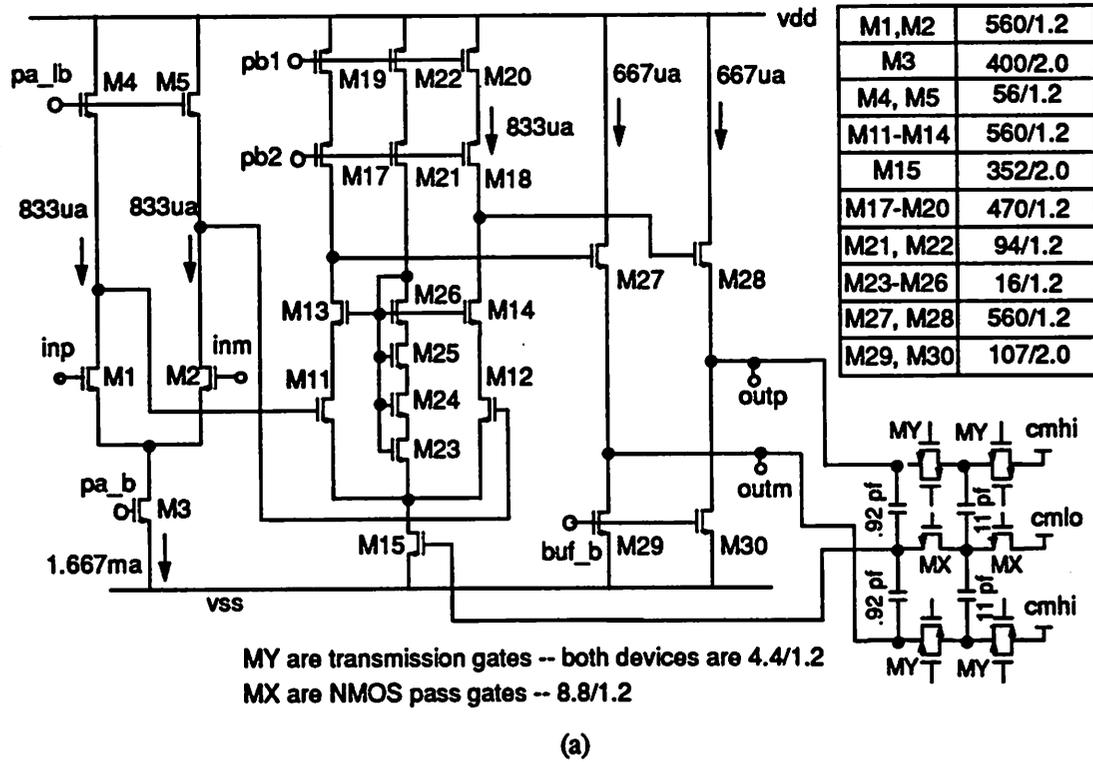


Figure 8.5 Schematic of amplifier and bias. (a) Telescopic transconductance amplifier with preamp. (b) Bias circuit shared by 6 amplifiers.

***More on the Use of the Voltage Follower Buffers***

Another more subtle advantage of the use of voltage followers has to do with pipelined analog stages. In Chapter 6, a technique for closed-loop non-overlap clock generation satisfying global clocking issues was presented. This approach ensures that sampling occurs in a following stage of a pipeline before the previous stage removes its output. Also in Chapter 6, a method for switched-capacitor implementation of a zero-order hold was presented where the output voltage of an SC integrator is sampled on a capacitor in one time period and the capacitor is switched back to the summing node during the next time period in such a way as to remove the integrating effects of the integrating capacitor. In a similar way, the zeroing capacitors likewise must take their samples before the amplifier removes the output.

A problem arises because the following stage sampling clocks are designed to fire before the previous stage opamp clocks. This results in charge being dumped back into the previous stage when the following stage takes its sample. During the resulting transient, the amplifier output that is then sampled by the zeroing capacitors can be corrupted.

The buffers have the added advantage that they reduce the charge “kick-back” into the core amplifier output and thus maintain the integrity of the core amplifier output for sampling by the zeroing capacitors. This is illustrated in Figure 8.6 where two capacitors of a following stage are shown sampling the outputs of a previous stage zero-order hold circuit. The following stage sampling clock  $\phi_{\text{stg2\_sample}}$  is shown turning off first, followed by  $\phi'_1$  and  $\phi_1$  before the low to high transition of  $\phi_2$ . As can be seen in the figure, the buffers act to decouple the main amplifier outputs from the charge injection and clock feedthrough effects of the second stage sampling process ensuring the integrity at the core amp output for the  $\phi'_1$  sampling event.

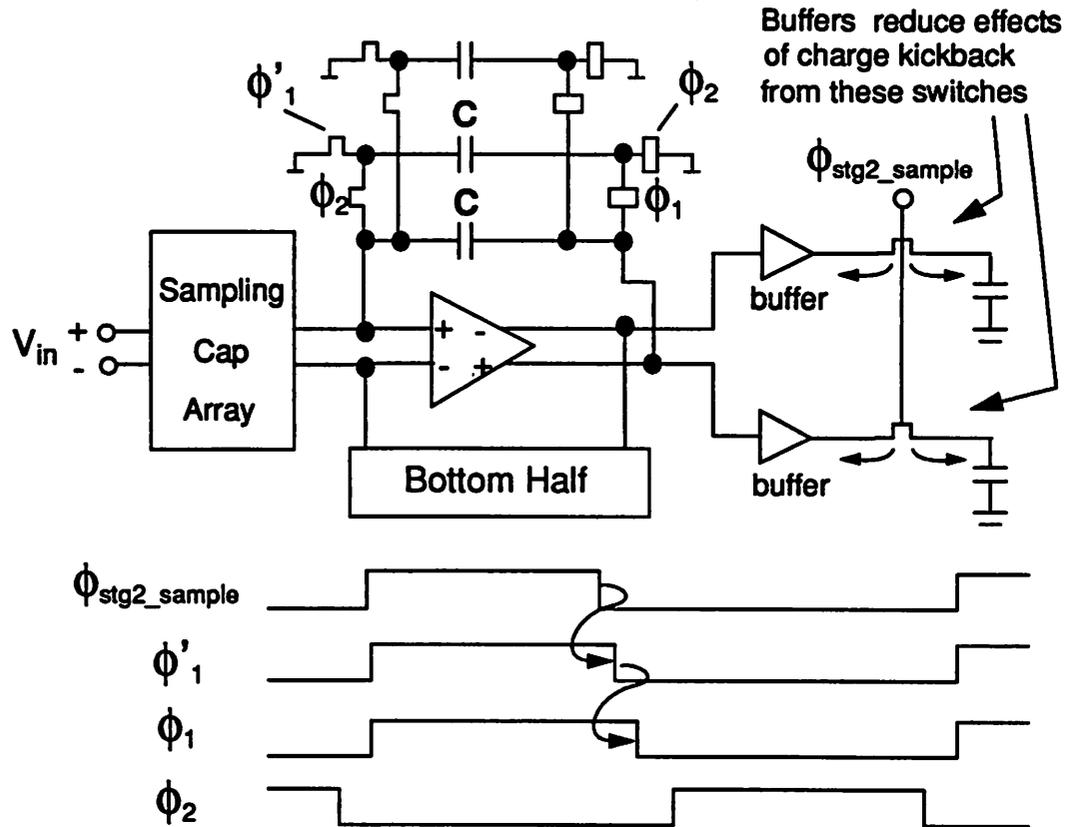


Figure 8.6 Block diagram illustrating the decoupling effect of the buffers on charge kickback from second-stage sampling.

### Amplifier Common-Mode Voltages

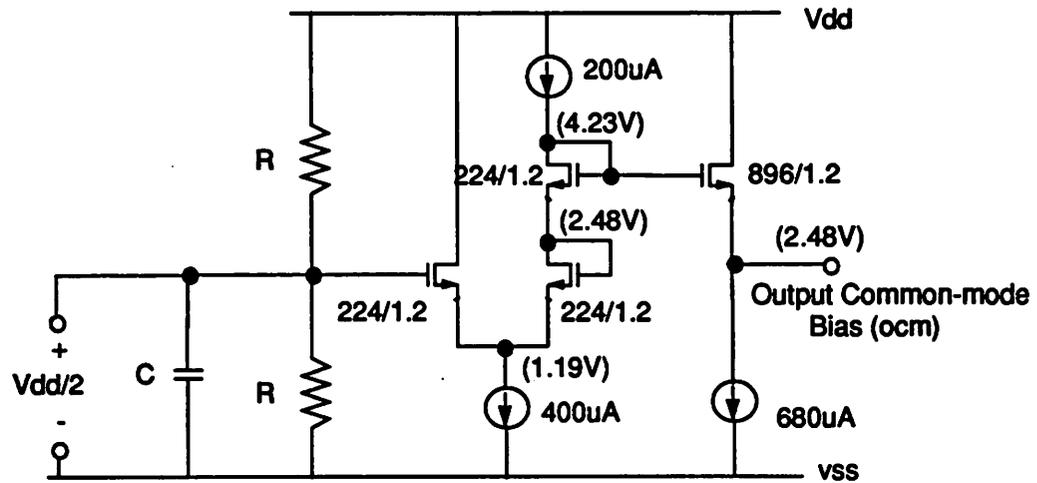
The switched-capacitor techniques being employed allow the amplifier to be used during both cycles of the clock. As a result, the input and output common-mode voltages of the fully-differential amplifier must be set by the switched-capacitors and the common-mode feedback network. At the input, this is achieved by sampling the input voltage with the bottom plate of the sampling capacitors connected to a reference voltage equal to the desired common-mode input voltage ( $icm$ ). The common-mode Thevenin equivalent network presented the amplifier input by the sampling network is a voltage source of value  $icm$  with an output resistance of  $1/fC$  where  $f$  is the frequency of the switching and  $C$  is the capacitance of the sampling capacitors.

The amplifier output common-mode voltage is set by the common-mode feedback circuit shown in Figure 8.5. As can be seen in the figure, reference voltage  $ocm$  is required by the bias circuit. This is the voltage to which the common-mode feedback circuit will drive the core amplifier output.

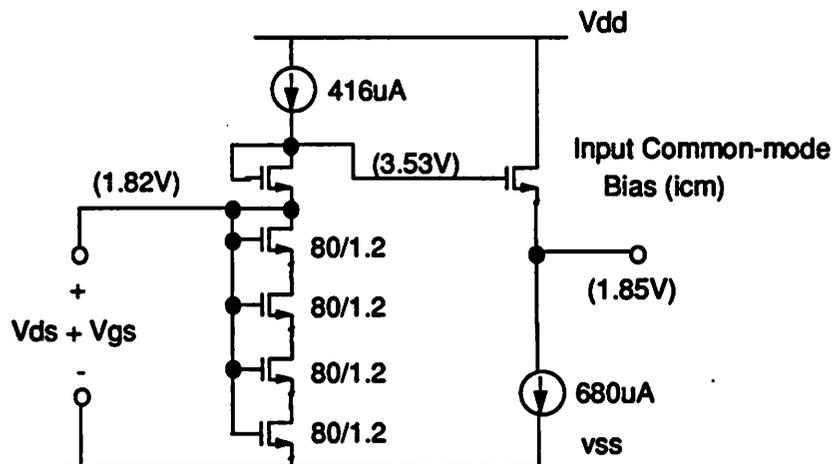
The input common-mode voltage ( $icm$ ) should be equal to the sum of the gate-source voltage of the devices in the input source-coupled pair and the drain-source voltage required across the current source device which assures it is in saturation. The output common-mode voltage should be roughly between the two supplies. Since the output resistance of the PMOS devices are much lower than those of the NMOS devices in the technology used in the prototype design, more  $V_{ds}$  is allocated to PMOS half.

Both of these reference voltages need to be sampled at a 100MHz rate. If they are provided off-chip, series inductance results in large amounts of ringing upon sampling. Therefore, it is desirable to generate these voltages on-chip. The circuits used to generate these voltages are shown in Figure 8.7. In both cases, voltage followers drive the bias lines to provide a low impedance high bandwidth source. No decoupling capacitors are added to these lines in order to maximize the bandwidth of these bias circuits.

In the output common-mode bias generator of Figure 8.7 a, a voltage equal to half of  $V_{dd}$  is generated with the resistor divider. Since the common-mode feedback circuit in the amplifiers are referenced to  $V_{ss}$ , a decoupling capacitor is added from the mid-point of the resistors to  $V_{ss}$ . This voltage is taken down a  $V_{gs}$  and then up two  $V_{gs}$ s, before finally coming down a  $V_{gs}$  in the voltage follower. In the input common-mode bias generator shown in Figure 8.7b, the target voltage is generated by forcing a current into four stacked devices with their gates tied together and connected to the drain of the top most device as done in the amplifier bias generator to generate the cascode bias voltages. This voltage is again taken up a  $V_{gs}$  so that it may come down a  $V_{gs}$  in the output voltage follower.



(a)



(b)

**Figure 8.7** Amplifier input and output common-mode reference voltage generators with nominal node voltages shown in parentheses. (a) Output common-mode bias generator. (b) Input common-mode bias generator.

### 8.3 Switched-Capacitor 3:1 Decimation Lowpass Filter

The switched-capacitor decimation filter provides most of the lowpass shaping of the spectrum at frequencies up to around 150% of the output rate of the filter with a linear phase response. The SC decimation filter architecture described in Chapter 5 was used to attain the high filter bandwidth. A 3-tap filter sampling at 3 times the output rate was implemented.

The number of taps and the decimation factor are both important parameters and need to be chosen carefully. From an implementation standpoint, the analog implementation is not modular like its digital counterpart. In fact, implementation complexity goes up almost exponentially as the number of taps or decimation factor is increased because the number of critical conditions needing to be satisfied simultaneously is increased. It is important to keep the filter parameters as simple as possible.

#### ***Selection of the Response: 3:1 Decimation Filter with 3 Taps***

In principal, with the use of a ring oscillator, extremely high effective sampling rates are possible. However, for reasons stated earlier, it is prudent to use the lowest decimation factor which enables the design objectives to be met. In the disk drive application, the lowpass filter bandwidth needs to be roughly one-half the output rate. When Class IV Partial Response signalling is employed, coding results in a zero at one-half the symbol rate and therefore a bandwidth less than one-half the output rate is sufficient. For the 100 MHz output rate objective, the filter bandwidth needs to be in the vicinity of 50 MHz.

A decimation factor of 3 was chosen because it allows the realization of filters that have minimal attenuation at 50 MHz while placing zeros at 150 MHz which results in a sufficiently sharp roll-off in the transition region. Also important is that preceding a 3:1 decimation filter with a cascade of two second-order sections results in minimal attenuation in the passband and sufficient attenuation at frequencies above 150 MHz where the magnitude response of the discrete-time filter begins to increase.

#### ***Filter Impulse Response***

A block diagram of the filter is shown in Figure 8.8. The relative tap weights  $C_0$  through  $C_2$  are .5, 1.0, and .5, respectively. The input sampling rate is three times the output sampling rate of  $f_s$ .

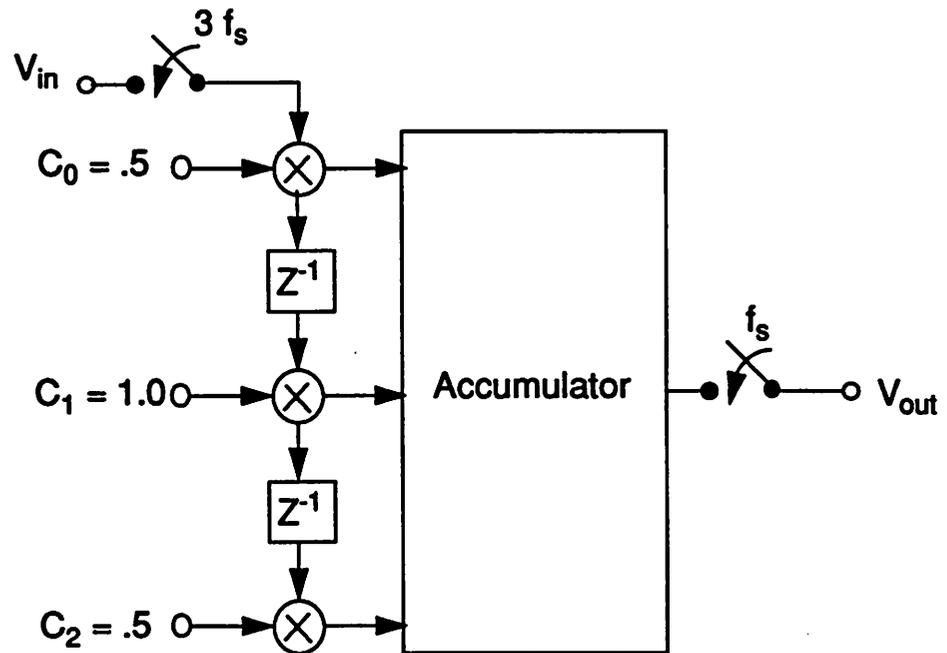
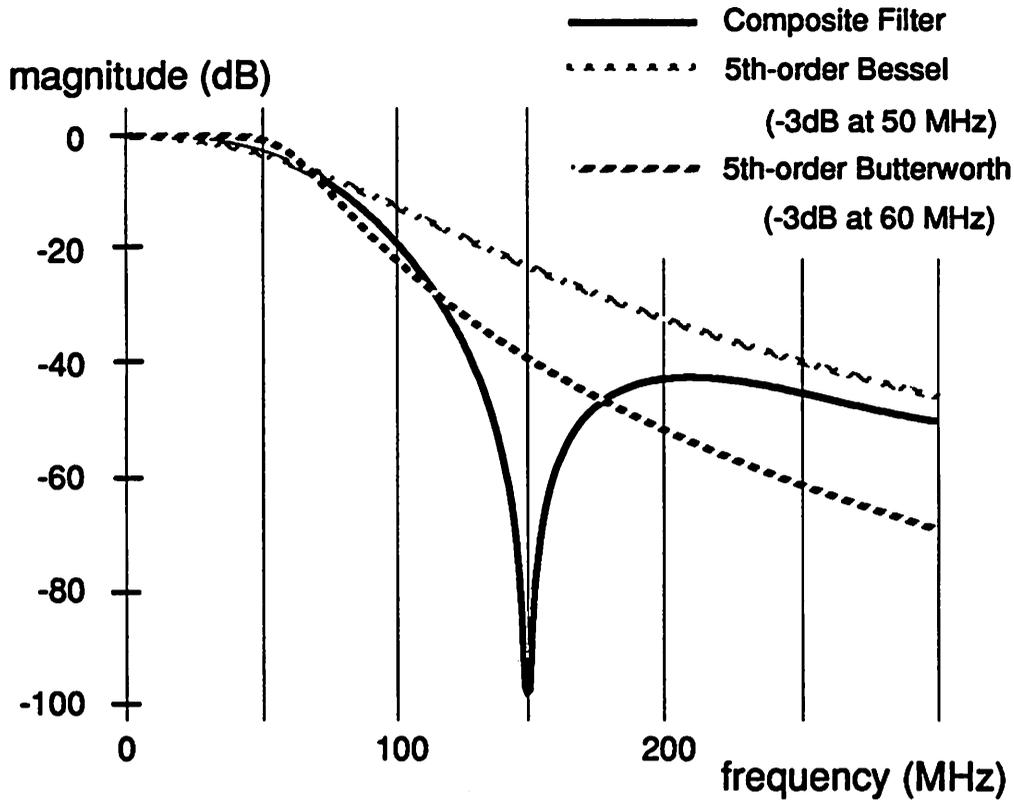


Figure 8.8 Block diagram of 3:1 3-tap lowpass transversal filter.

A three tap response was chosen since it is short and yet provides shaping of the spectrum similar to a fifth-order Bessel or Butterworth filter response. In Figure 8.9, a plot showing a comparison between the composite filter made up of a cascade of the continuous-time filter and the 3-tap lowpass filter before decimation is shown with a fifth-order Bessel filter with a 50 MHz bandwidth and a fifth-order Butterworth filter with a 60 MHz bandwidth. The Butterworth filter has a higher bandwidth because of the extra phase distortion incurred by the Butterworth filter near the pass-band edge.



**Figure 8.9** Comparison of response of proposed composite filter with Fifth-order Bessel and Butterworth magnitude responses.

The first architectural decision to be made was the number of parallel stages to be used in the implementation. An estimate was made on the load capacitance, and the total input and feedback capacitance needed in order to achieve the desired filter accuracy.

#### Capacitor Selection and Design

A choice was made to use integer multiples of unit capacitors in order to have the design be less sensitive to process variations in the relative contributions of parallel plate and fringing capacitance [4,5]. The magnitudes of these tap values were added to get sum which would be equivalent to the normalizing feedback capacitor in an SC multiple input sample-and-hold.

### Selection of the Number of Parallel Stages

Three parallel stages allows 33.3ns periods for the opamps. Approximately 8 ns were allocated for non-overlap clock generation leaving a total of about 25 ns for amplifier settling. Under the same assumptions, if 2 parallel stages were chosen only 15 ns would be available for opamp settling. Initial simulations indicated that amplifier settling times in the low 20 ns range was easily obtainable and therefore the choice for 3 parallel stages was made.

### Clocking Issues for One Stage

Shown in Figure 8.10 are the clocks required for operation of filter stage 1. Clocks  $l0xp$ ,  $l0xpp$ , and  $l0xppp$  have non-overlapping relationships with  $l0yp$ ,  $l0ypp$ , and  $l0yppp$ . The single  $p$  suffix signifies the early clock. The  $pp$  and  $ppp$  suffixes control the multiplexor switches and are active when the corresponding  $p$  clock is active. Clocks  $rs0$  and  $rs3$  reset the common-mode of the sampling networks employing center switch sampling. Clocks  $p00$ - $p02$  and  $p30$ - $p32$  are the sampling clocks which define the sampling instants.

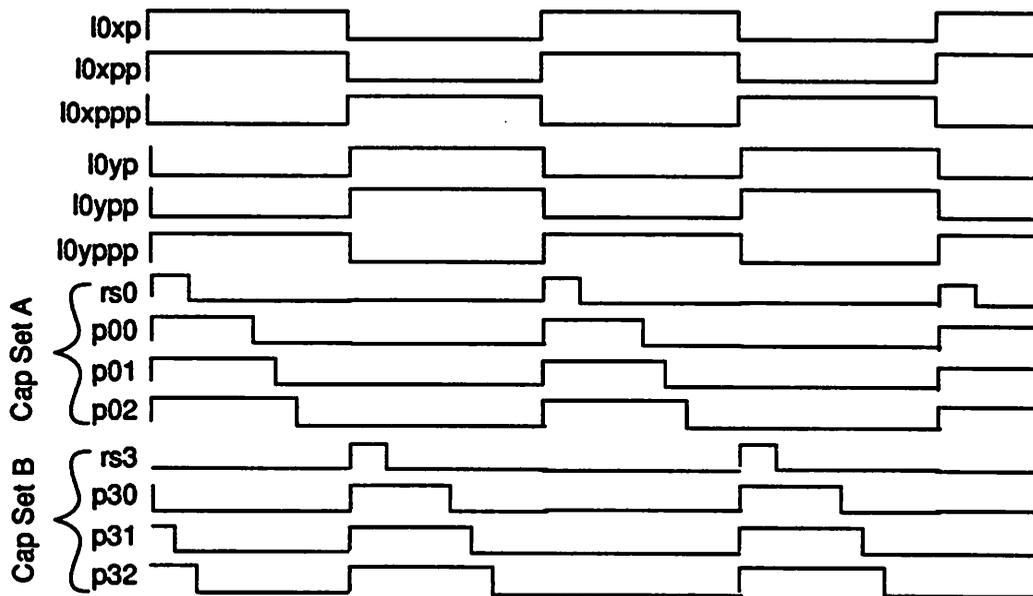


Figure 8.10 Clocks for Filter Stage 1.

**3 Stages in Parallel**

In order to get the desired throughput, three parallel stages are required as illustrated in Figure 8.11. In each of the stages, there are two interleaved capacitor sets A and B. While Set A is in the sampling mode, Set B is in the dump or evaluate mode. The impulses in the figure signify sampling on a capacitor ratioed by the relative magnitude of the impulse (Shorter impulses are half the magnitude of the longer). Two sets of capacitors share one amplifier. The double arrow indicates the settling period of the amplifiers. Note that the filters/amplifiers operate one output period out of phase, three samples are taken every output period, and the amplifiers have three output periods for settling.

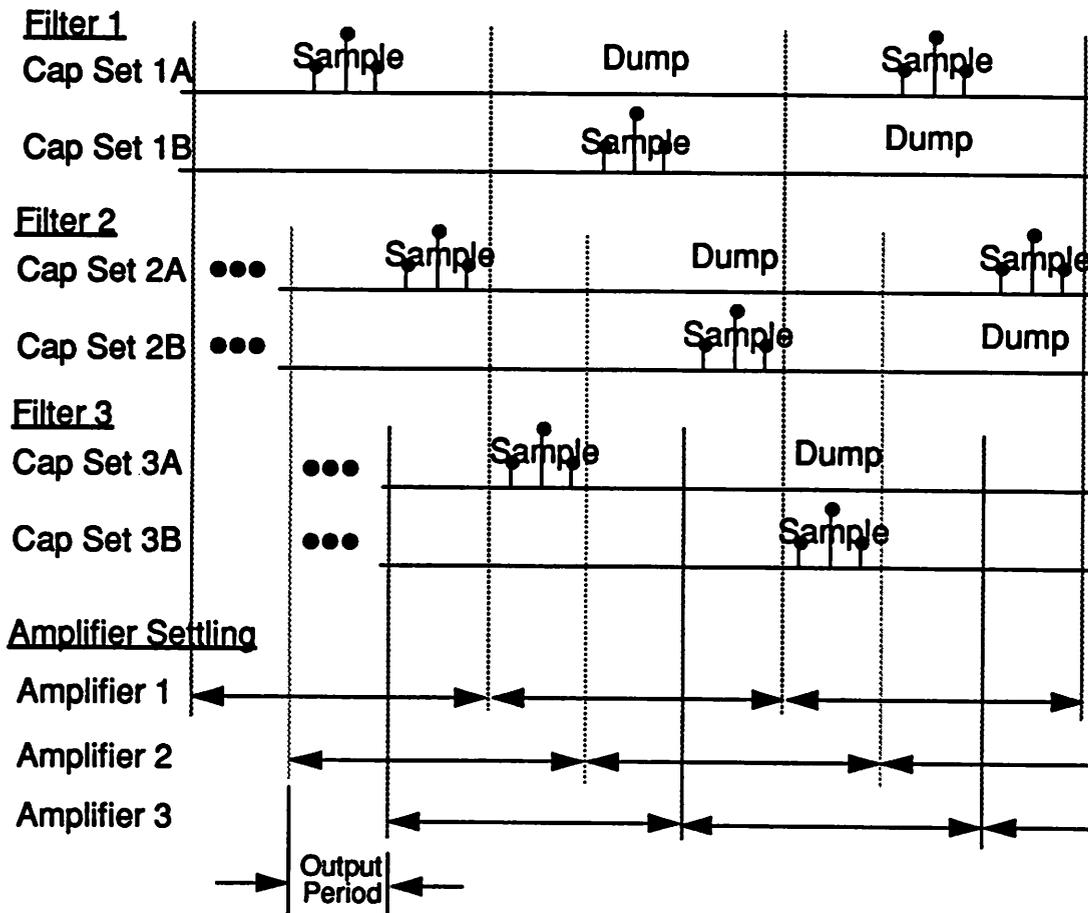


Figure 8.11 Operation of 3 parallel filters, each with 2 capacitors sets.

A simplified schematic of the entire decimation filter is shown in Figure 8.12 with the key clocks waveforms in Figure 8.13.

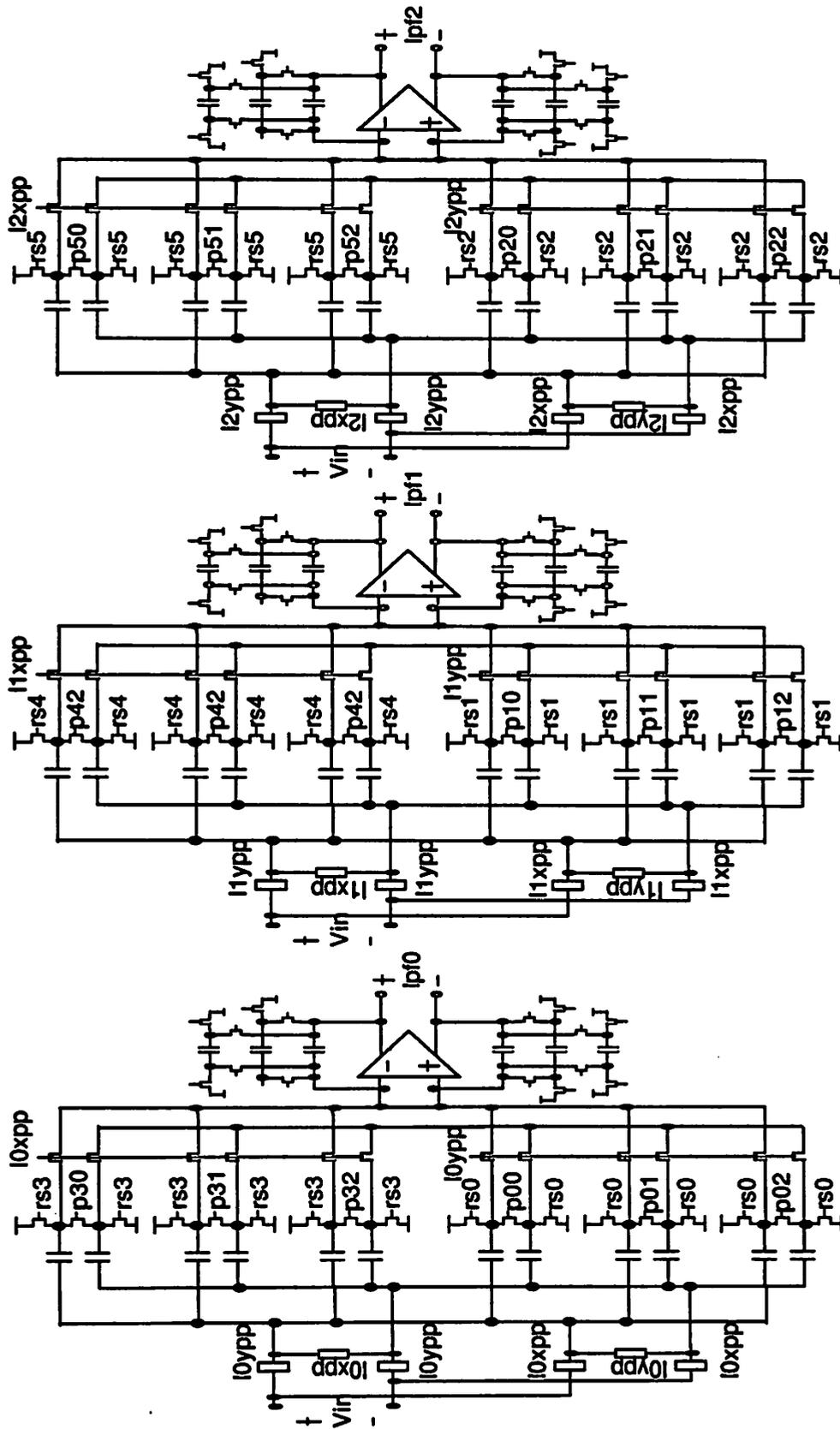


Figure 8.12 Circuit schematic of the SC Decimation Lowpass Filter.

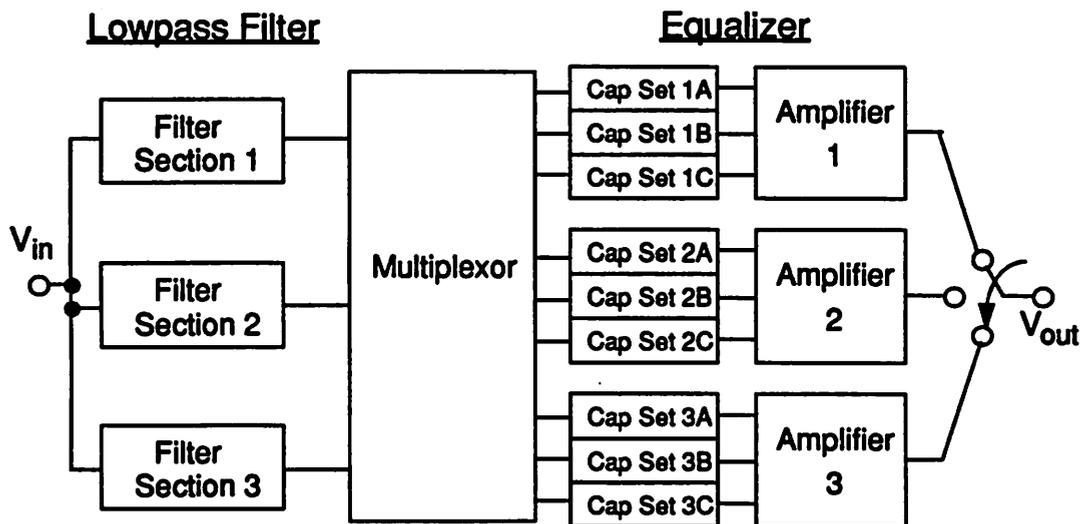


## 8.4 Programmable Switched-Capacitor Raised Cosine Equalizer

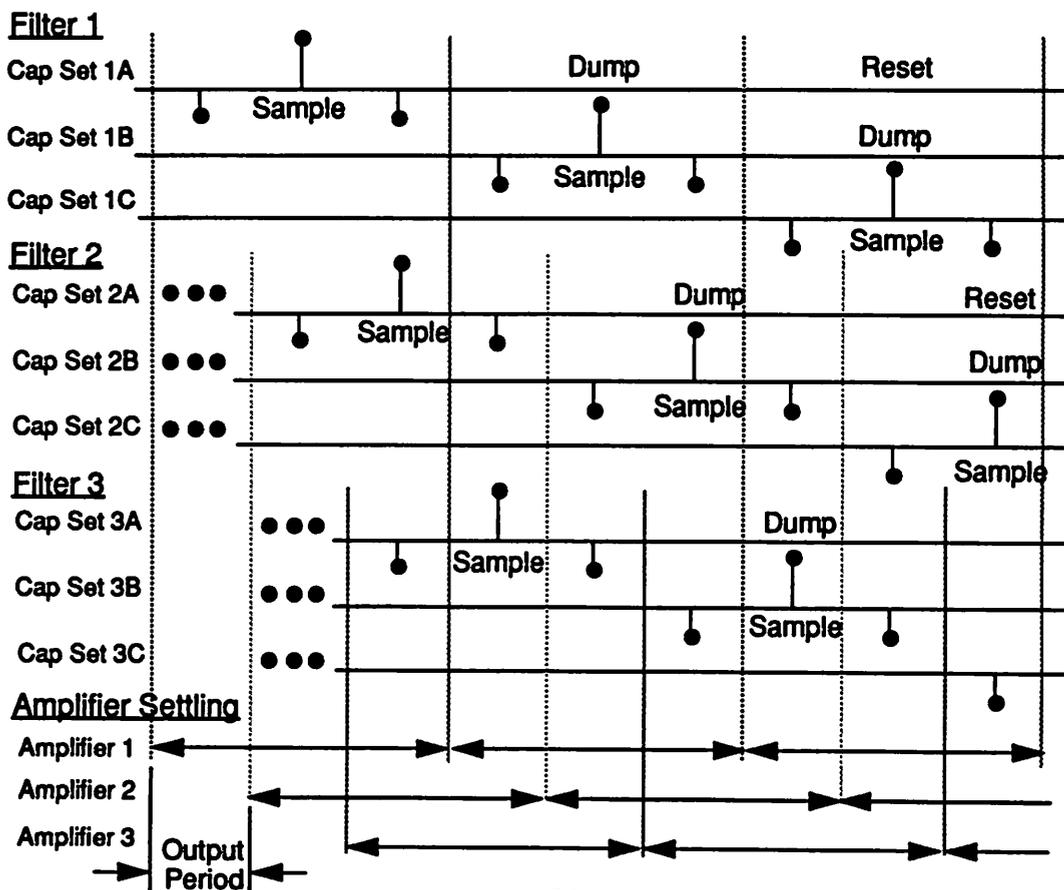
The equalizer block provides 3 programmable settings of high frequency boost. This equalizer operates at the output rate of the lowpass filter. It is a 3-tap equalizer with symmetric impulse response  $-K, 1.0, -K$  where the choices of  $K$  include .25, .35, and .45. These impulse responses result in different levels of peaking at one-half the output rate of the filter. The setting with  $K$  equal to .35 results in an equalizer close to that desired to equalize an ideal Lorentz pulse with a  $PW50/T$  of 2.5 to a Class IV Partial Response target.

### *Parallel Architecture*

In order to use the same amplifiers and similar sets of amplifier clock phases as used in the SC lowpass filter, the architectural approach for the equalizer also employs 3 parallel filter stages. A conceptual block diagram of the equalizer and a diagram illustrating scheduling of the capacitor arrays and amplifiers are shown in Figure 8.14.



(a)



(b)

Figure 8.14 (a) Equalizer block diagram with diagram. (b) Scheduling of components.

### Input Capacitor Configuration

In switched-capacitor circuits where the gain is programmed through the addition or subtraction of capacitance, an important consideration is the way in which changing the gain changes the feedback factor of the closed-loop amplifier. This can result in variations of the stability or speed of the amplifier with different gain settings.

Since one of the taps is always fixed, the input capacitors should be variable as opposed to the feedback capacitor which would affect the gain of all three inputs. The approach used in the prototype is shown in Figure 8.15. Referring to the figure, there are three possible settings and at any one time, only one of the three control signals *more*, *less*, or *nominal* can be ON or HI. In the *nominal* setting, there are 7 units of capacitance from the differential inputs to the amplifier inputs. Two units are added in parallel for 9 units in the case of *more*, and cross-connected in the case of *less* which has the effect of subtracting two units from the nominal 7 to get an effective total of 5. The advantage of this approach is the effective capacitance on each of the summing nodes is effectively 9 units to a.c. ground from the standpoint of the feedback factor independent of the setting.

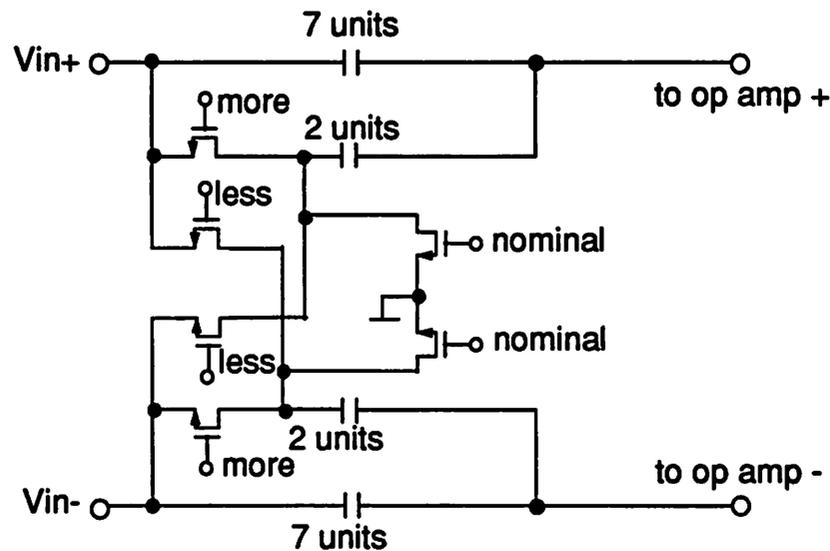


Figure 8.15 Schematic for programmable tap weight realization.

### Frequency Response

In Figure 8.16 are plots of the frequency response of the equalizer at each of the programmable settings. The settings as labelled in Figure 8.15 using *more*, *nominal*, and *less* are used in the figure. Since this is a discrete-time filter, the frequency response will track the clock rate. The maximum amount of boost will always occur at one-half of the output rate of the filter.

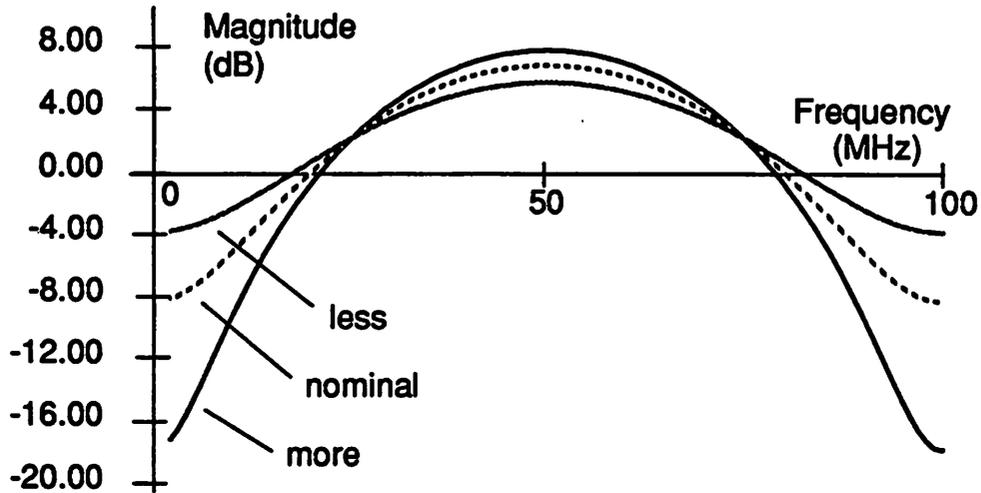


Figure 8.16 Frequency response of the programmable equalizer at each of the 3 settings.

A circuit schematic of the equalizer is shown in Figure 8.17 with the associated clock waveforms in Figure 8.18. Signals ADC0-ADC2 are the differential inputs to the analog-to-digital converter block which follows.



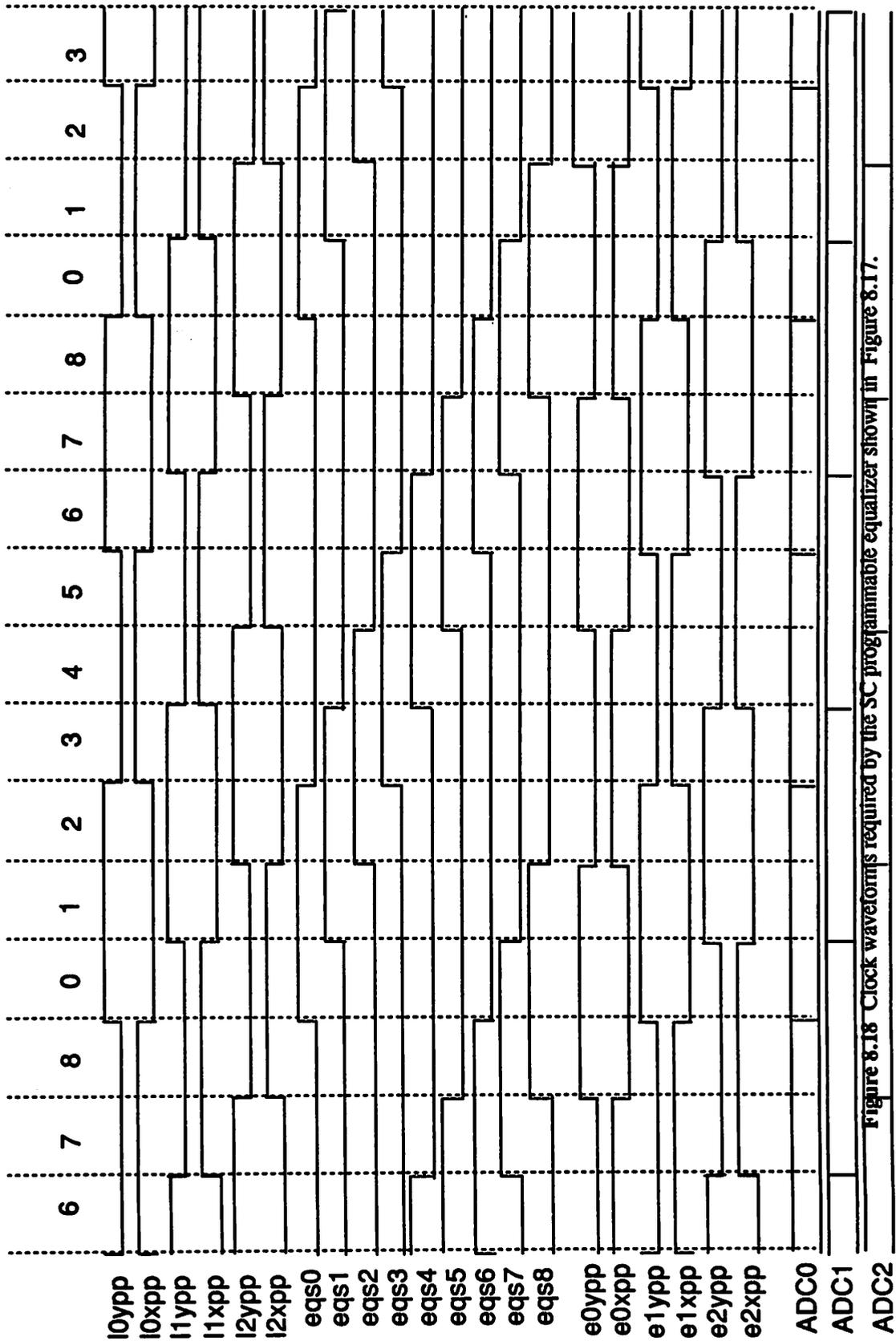


Figure 8.18 Clock waveforms required by the SC programmable equalizer shown in Figure 8.17.

## 8.5 6-Bit 100 MHz Analog-to-Digital Converter

In this section, we shall examine design details of the implementation of the A/D Converter. The algorithm for the A/D converter in the prototype is a two-step parallel architecture and was described in Chapter 8.

Preceding the A/D converter is the equalizer block which is implemented using three parallel SC multi-phase summers. At a 100 MHz effective output rate, this means that one of the three outputs of the equalizer are ready every 10 nsec. Since the signal path is independent at this point, the approach adopted is to have each equalizer stage feed its own A/D converter as shown in Figure 8.19. As a result, clocking and connectivity is simplified because each equalizer amplifier feeds one of the three ADCs. This approach also allows the amplifiers in the equalizer section the full three periods for settling so that the settling time allotted these amplifiers need not be reduced by the presence of the A/D converter.

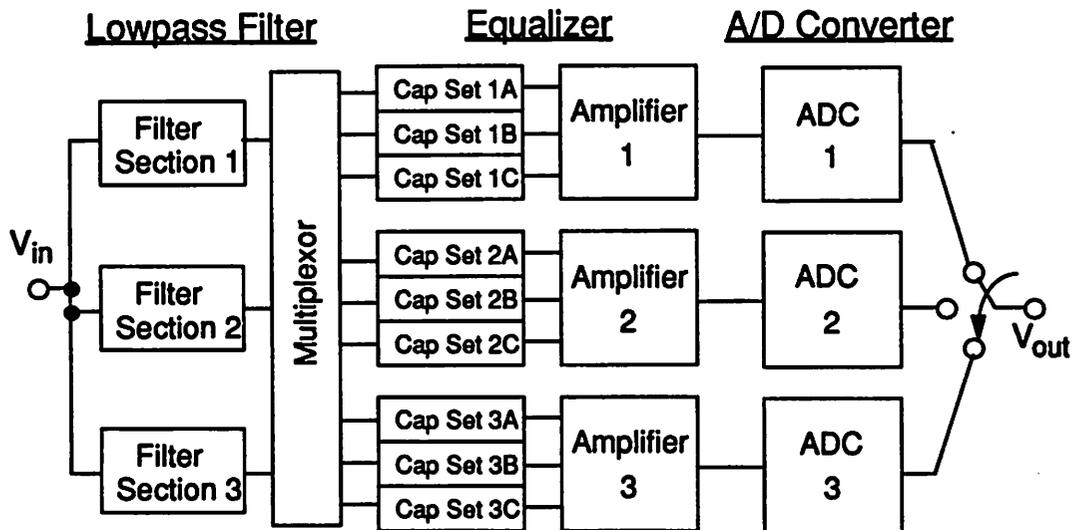
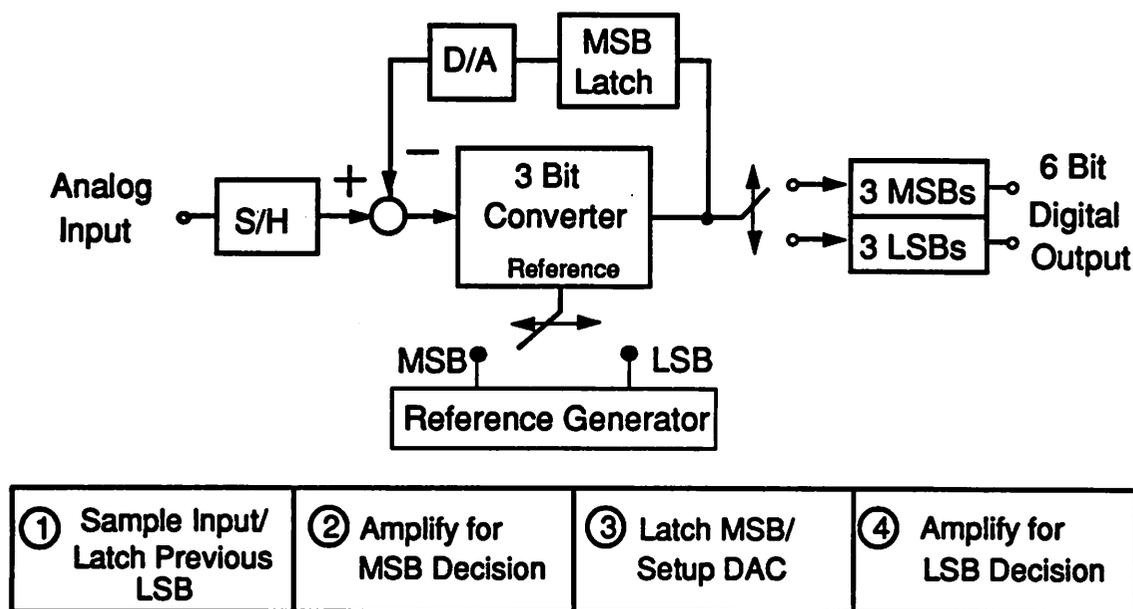


Figure 8.19 Each amplifier in the equalizer block feeds its own A/D converter.

The A/D architecture presented in Chapter 8 is a two-step converter architecture which requires four clock phases. A conceptual diagram of the algorithm is repeated below in Figure

8.20 for convenience. One of the key design considerations of the algorithm was to maximize the allowed settling time of the preceding sample-and-hold stage.

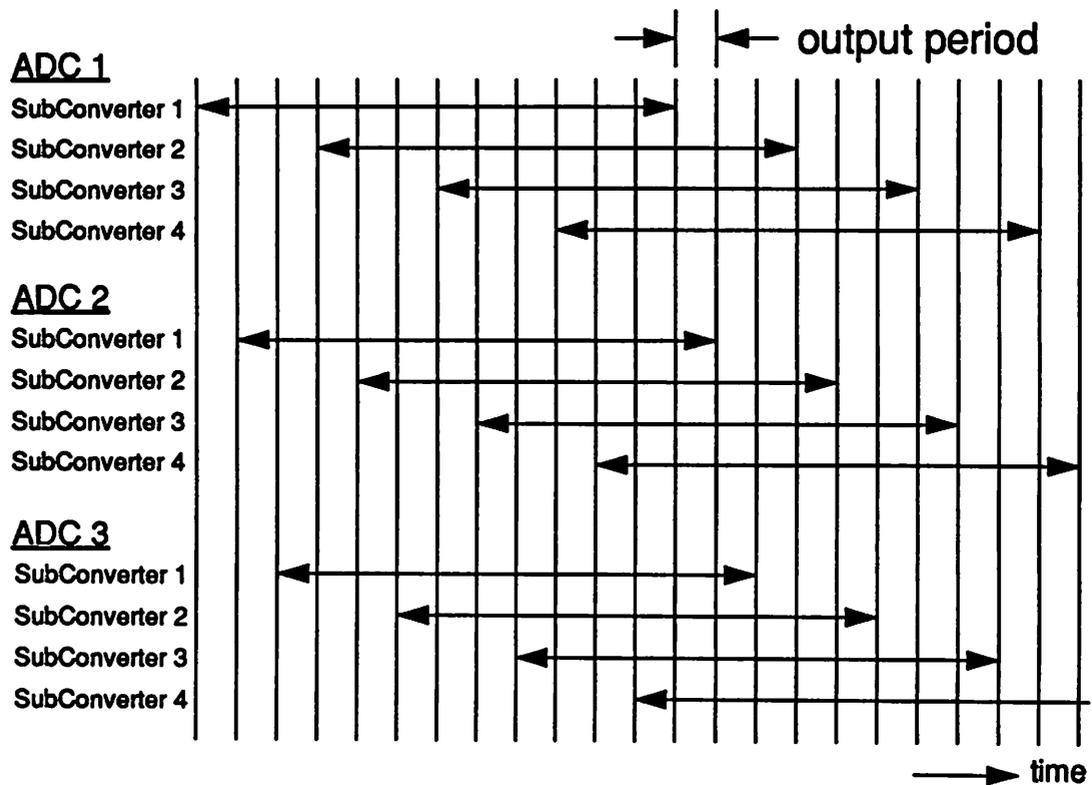
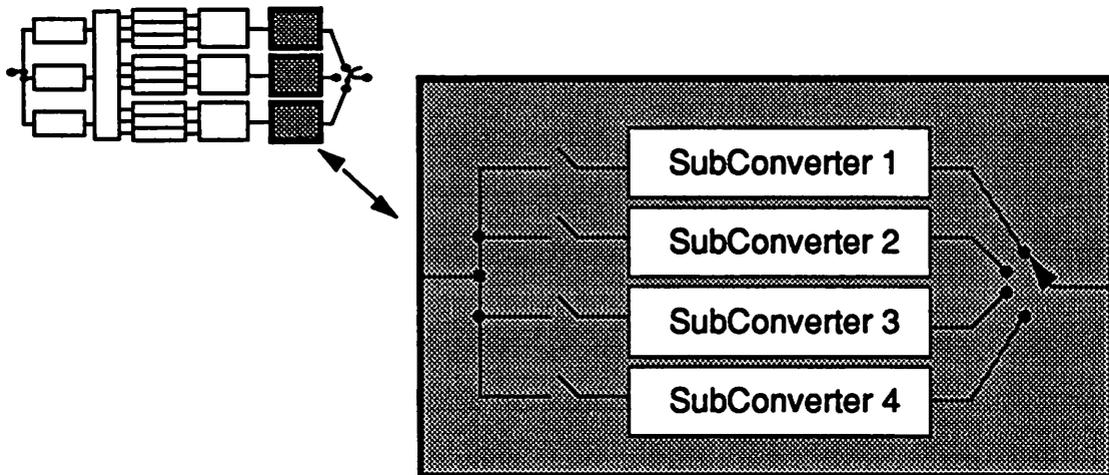


**Figure 8.20** Conceptual diagram describing A/D converter algorithm with 4 phases of operation. Each phase is 3 output periods long.

Each of the three ADCs (ADC1-ADC3) shown in Figure 8.19 are made up of 4 A/D converters, each operating in one of the four phases of the algorithm shown in Figure 8.20. Let us refer to a group of four converters as an *ADC block* and each of the individual converters which implement the 4 phases of the algorithm as a *SubConverter*. The composite A/D converter is made up of a total of 12 SubConverters operating in parallel as shown in Figure 8.21. The equalizer amplifiers synchronously drive one of the three ADC blocks which operate with a one period phase shift, and each of the ADC blocks are comprised of 4 interleaved SubConverters. Note that each of the interleaved SubConverters within an ADC block lags the previous SubConverter by 3 periods.

This architecture has many important advantages. First of all, the amplifiers have 3 output periods to settle. Next, the regenerative latches have 3 periods to make decisions which reduces the probability of the occurrence of a metastable condition. The entire converter is comprised of a

total of only 84 comparators with no interstage amplifiers. And finally, only 7 pairs of MSB and 7 pairs of LSB references need to be routed throughout the converter.



**Figure 8.21** Three parallel ADC blocks, each with 4 SubConverters operating in 1-of-4 phases of the algorithm results in a total of 12 parallel SubConverters.

***A/D Converter Slice***

Let us now examine the circuits used in the implementation of the algorithm. We shall first examine the basic functional block of the two-step converter architecture which includes a comparator made up of a regenerative latch and associated preamp and sampling capacitors and switches. This block will be referred to as a *slice* and a simplified single-ended schematic along with timing information describing operation of a slice is shown in Figure 8.22. Three bits are resolved during each conversion requiring 7 parallel slices, each using a different set of voltage references. The MSB and LSB references in the figure refer to the particular set of references used by that slice. Let us go through each of the four phases of the algorithm.

During  $\phi_1$  capacitor  $C_1$  samples the input  $V_{in}$  from the equalizer stage. Capacitor  $C_2$  during this period is initialized to zero volts. Once the MSB decision is made,  $C_2$  will be the capacitor used to subtract off the MSB reference from  $V_{in}$ . During the  $\phi_2$  period, the left side of  $C_1$  is connected to an MSB reference. The effect is that the voltage on the right side of  $C_1$  goes negative if  $V_{in}$  is larger than the reference and positive if it is smaller. This voltage excursion is amplified by the preamp and sampled at the end of the period by the following latch stage. Upon the start of  $\phi_3$ , the regenerative latch amplifies the voltage sampled at the preamp output and the regenerative nature of the latch results in a full CMOS logic level which includes either of the supplies ( $V_{dd}$  or  $V_{ss}$ ). The output of 7 of the latches in parallel and a logic circuit signals at which MSB reference level is closest to the input signal without exceeding it. This is the MSB decision whose location is sensed by a decode ROM for a binary output. Also important is that this decision be passed to each of the seven slices for the next period. This next period is  $\phi_4$  where  $C_2$  is connected to the MSB reference corresponding to the MSB decision. Capacitor  $C_1$  is then connected to the LSB reference. The net result is that the resultant voltage excursion on the comparator input node depends on the sign of  $V_{in}$  minus the MSB decision minus the LSB reference. If this is positive, then the voltage excursion will be negative meaning that  $V_{in}$  minus the MSB decision is still larger than the LSB reference. Naturally, the excursion will be negative if it is smaller than the LSB reference. This voltage excursion is amplified by the preamp and again sampled by the latch at the end of  $\phi_4$ . Then during the next  $\phi_1$ , while the next input  $V_{in}$  is sampled on  $C_1$ , the latch amplifies the LSB decision to be decoded by the ROM for 3 more bits of binary output. The 6 bits are then resampled

and multiplexed to the output pads. The transmission gate controlled by  $\phi_{24}$  in the figure pipelines the process so that the latch can be in regeneration while a new input is being sampled.

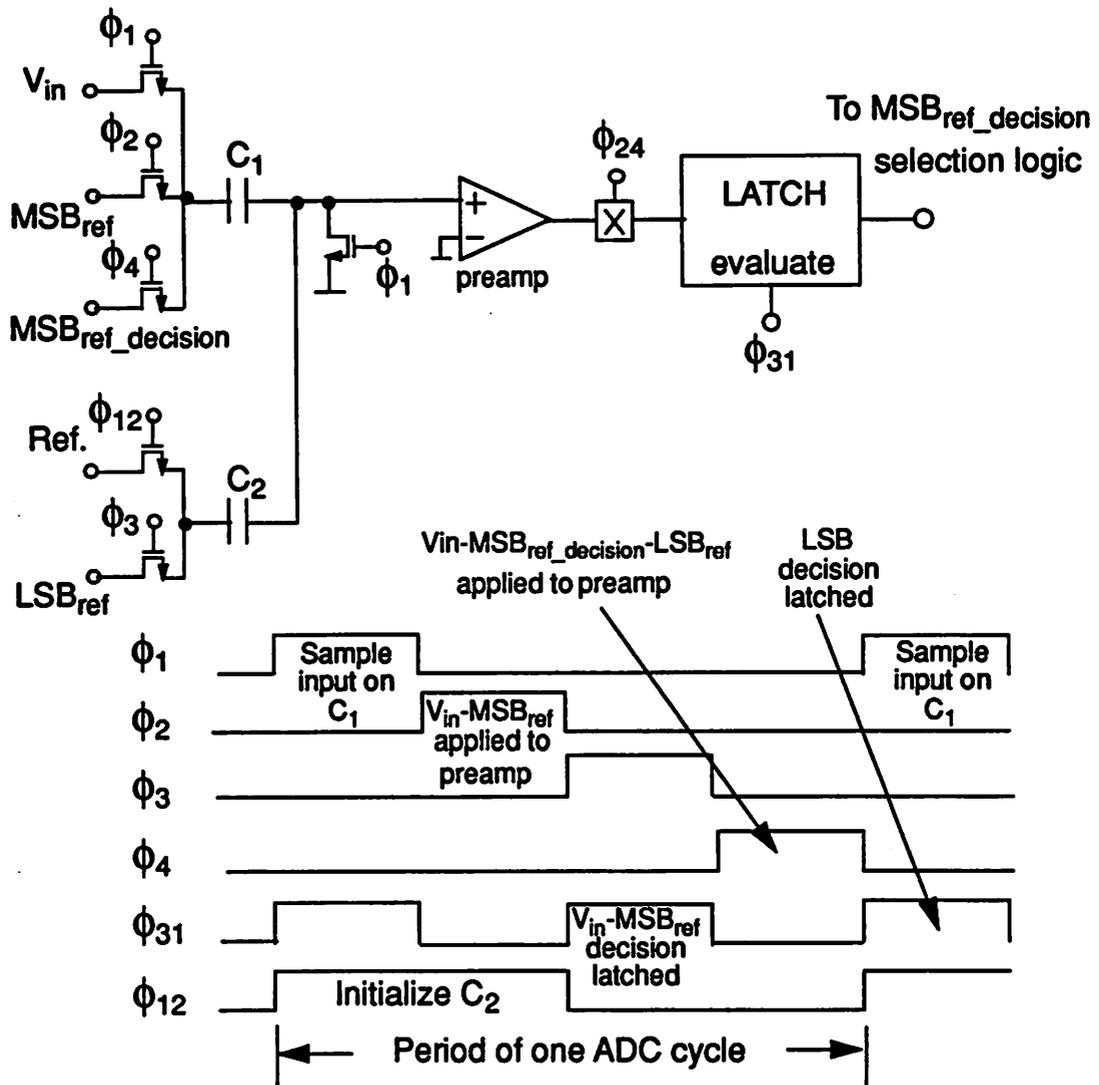


Figure 8.22 Simplified clocking of an A/D converter slice.

### Parallel Slices Make Up Flash Converter

Seven slices are placed in parallel, each with a different set of voltage references as shown in Figure 8.23. After a comparison, the outputs of the slices make up a thermometer code. This code must then be converted to a binary equivalent. The approach used in the prototype (one way of many) is shown in the figure and comprised of an array of two-input NOR gates which performs

the thermometer to 1-hot code conversion. This is followed by a ROM which converts the 1-hot code to a 3-bit binary equivalent.

An example input voltage is shown schematically in the figure with its location between slices 4 and 5. As a result of the decisions output from the slices (note the differential output where the minus output is the compliment of the logical comparator output), only the output of the NOR gate corresponding to the location between these slices is HI which turns on the two NMOS pull-down devices in the ROM and results in the binary output 100 or the equivalent of decimal 4.

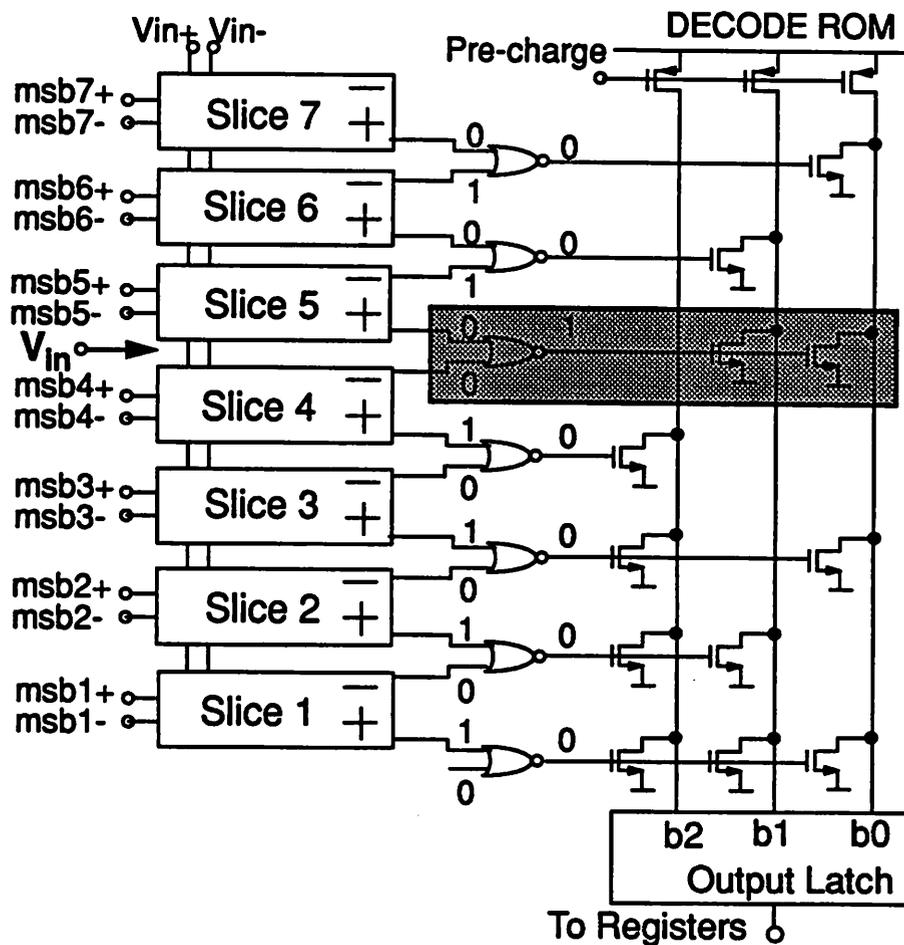


Figure 8.23 Schematic diagram for MSB decision showing parallel slices and decode circuitry. Example shows operation of decode circuitry.

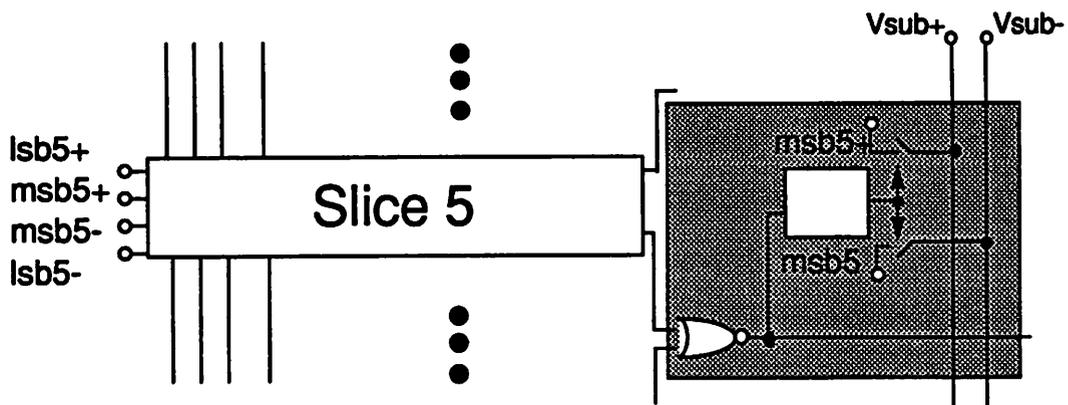
The MSBs and the LSBs are decoded using the same circuitry. After the MSB decision is made, the slices reconfigure themselves for the LSB conversion. Another higher level view of the



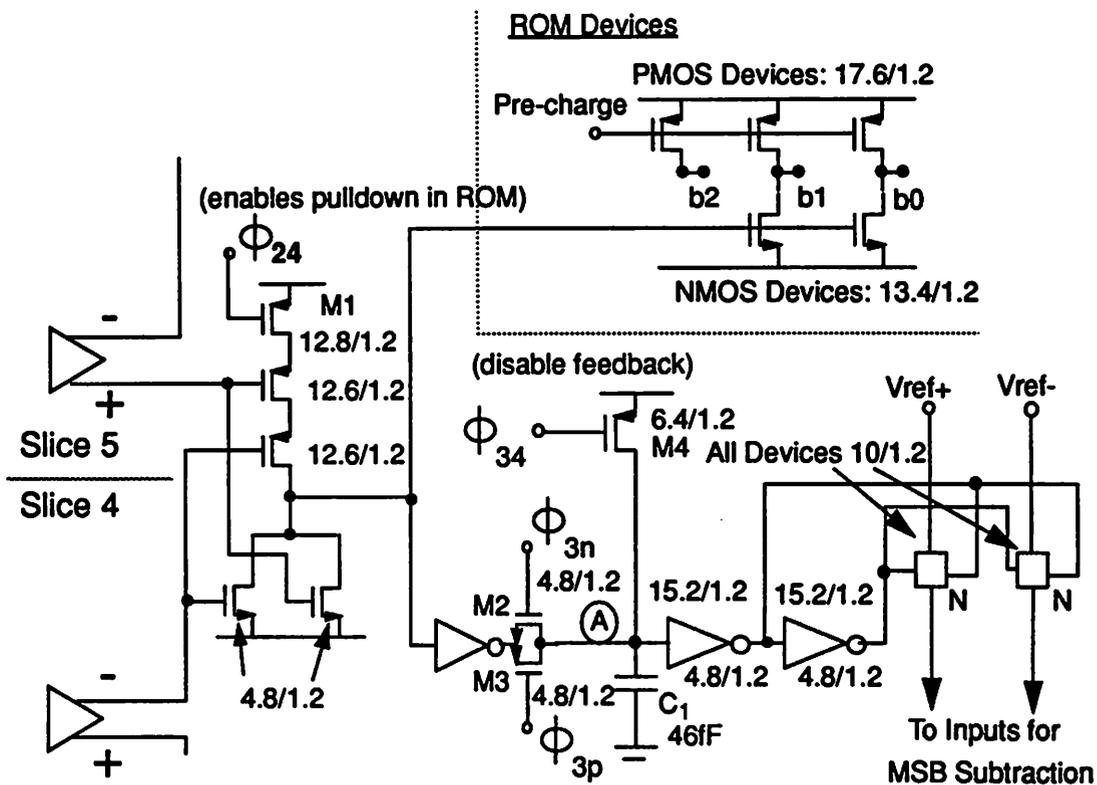
Let us now examine the circuitry used to implement the NOR function as well as the latching function which when the output of a particular NOR is HI connects the MSB references to the subtraction lines. The functions we are now considering is shown in the shaded box of Figure 8.25a where a portion of Figure 8.24 is repeated.

The outputs of the slices are shown in Figure 8.25b driving the inputs of a NOR gate with an enable HI PMOS transistor M1 in series with the PMOS transistors of the NOR gate. The portion of the ROM that this particular NOR gate drives is also shown. Since the ROM uses dynamic logic with a pre-charge HI, the NOR gate output must not be allowed to go HI during a transient in the decode process. The enable HI PMOS transistor in the NOR gate assures that the NOR output does not go HI during the pre-charge phase. We will see in a later section that during the precharge of the latch, the signal levels driving the latch are roughly at the midpoint of the supplies. When the latch is enabled, the voltage at the outputs of the latch and into the NOR begin near the midpoint prior to regeneration which causes the latch outputs to head to opposite supplies. Since the ROM is precharged HI, it is important that the NMOS inputs to the ROM never inadvertently turn on during a glitch from the output of the NOR gate. The NOR structure between the latches and the ROM is very compatible. Since the latch outputs prior to regeneration sit at the midpoint of the supplies, the NMOS devices in the NOR keep its output LO. In order for a HI to occur at the output of a NOR, two LO latch outputs (corresponding to the input voltage being between two slices) along with the enabling of M1 is required to activate a pull-up in the NOR. Thus, there is a guard band by design against inadvertent pull-down signals going into the ROM. This guard band is important because dynamic logic is unforgiving to such events.

Operation of the MSB reference feedback is relatively straightforward. The transmission gates connecting the reference voltages are ON only if the NOR output is latched HI. Transistors M2-M3 and capacitor  $C_1$  make up a dynamic latch which holds the decision during  $\phi_4$  when the decision is used. Transistor M4 drives node A HI during  $\phi_1$  and  $\phi_2$  to ensure that all transmission gates which enable the MSB reference feedback are OFF.



(a)



(b)

Figure 8.25 Inside view of circuitry in shaded box in (a) shown in (b).

*Detailed Description of A/D Slice*

Let us now examine a detailed schematic of an A/D converter slice shown in Figure 8.26. The operation of this circuit was described in Figure 8.22. The PMOS source-followers on the left provide buffers which lowers the input capacitance into this block. This is important of two reasons. First, it reduces the load capacitance on the previous stage sample-and-hold amplifier, thus reducing the settling time required for this amplifier. Second, it reduces the total capacitance switched into the resistor string which generates the reference voltages. This is significant due to the number of slices connected to the resistor string at any one time. Since the voltage followers are d.c. coupled through the sampling capacitors into the preamp input nodes, the voltage changes that occur on their gates is what is reflected to the preamp inputs. As a result, the input-referred offset voltage of the followers are inconsequential since they are sampled across the sampling capacitors.

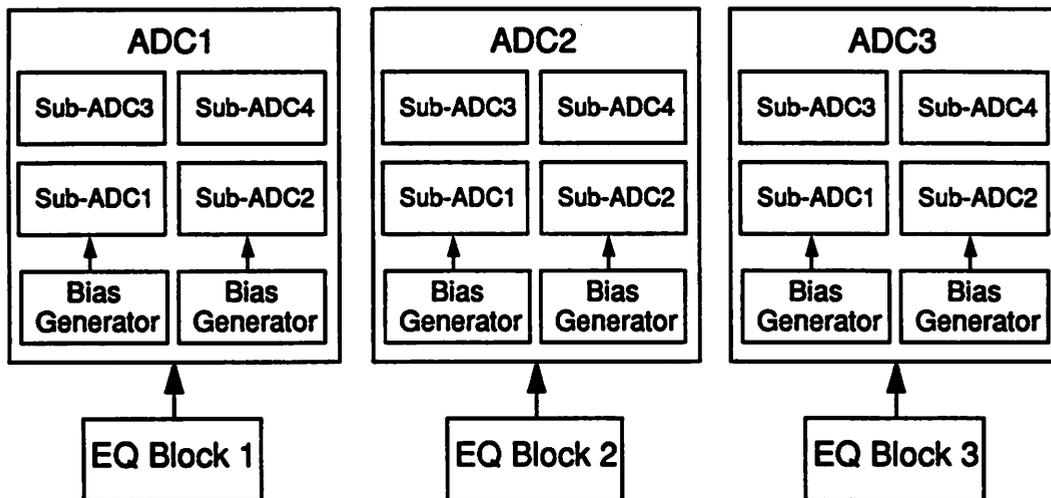
The center switch sampling approach is used. Upon the falling edge of  $\phi_1$ , the input nodes to the preamp becoming the summing nodes for the switched operations that follow which realize the algorithm. The preamp is a simple differential pair with a PMOS triode load. The cascode devices are present to reduce voltage variations in the drain voltages of the input devices. This is important to reduce coupling back from these nodes into the summing nodes through the  $C_{gs}$  of the input pair which could affect operation of the algorithm. The PMOS shorting switch between the preamp output nodes is present to help reset the differential output between cycles. The NMOS followers between the preamp output and the latch provide buffering and level shifting into the latch. Without these buffers, the preamp takes a relatively long time to precharge to latch back from the rail-to-rail voltage swing due to latching during the previous cycle. As can be seen from the relatively large bias currents flowing in these followers of 125 mA, a large driving capability is needed to precharge the latch in sufficient time.

The input-referred offset voltage of the preamp-latch combination must be approximately 12 mV to be about one-quarter of an LSB. The contributors to this are the input device pair of the preamp, and the sum of the offsets from latch buffers and the regenerative latch divided by the preamp gain. The input device pair was sized in order to achieve about 5 mV of offset and to achieve a gain which would be large enough to reduce the input referred offset of the buffers and latch to about 7 mV. The device sizes in the input pair required to achieve this offset and gain



***Bias Generation***

Let us now examine bias voltage generation for the A/D converter. Each of the three opamps in the previous equalizer stage acts as the sample-and-hold amplifier for an ADC comprised of four 7-comparator sub-ADCs as shown in Figure 8.27. The sub-ADCs operate in the order as numbered in the figure. Sub-ADC pair 1 and 3 and pair 2 and 4 share their own local bias generator in each of the ADCs. Thus, there is a total of 6 bias generators in the entire A/D converter.



**Figure 8.27** View of local bias generators within ADC blocks.

An external current of 2.5 mA is supplied to a master bias generator which generates bias currents that are sent to the local bias generators. Sending currents from the Master Bias Generator and creating voltages locally at each converter makes the bias generation less sensitive to variations in the supply voltages and transistor characteristics throughout the converter. This approach is shown schematically in Figure 8.28.

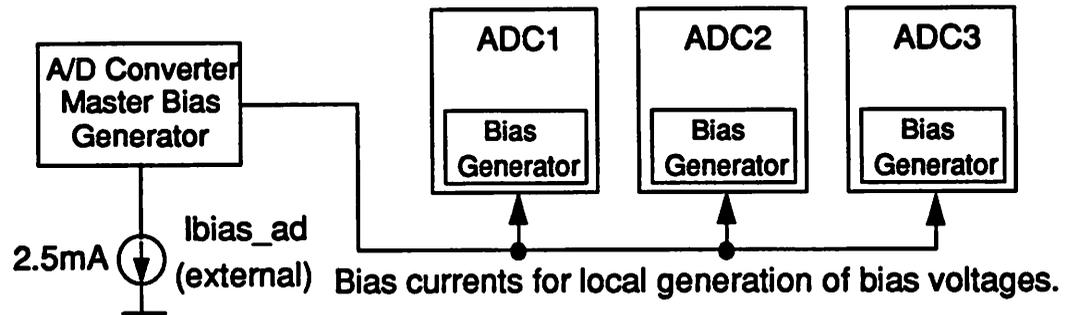


Figure 8.28 Master-slave current bias generation approach.

A schematic of the Master Bias Generator and one of the local Bias Generators is shown in Figure 8.29 along with a sampling of the circuitry the bias generators drive in the ADC slices.

There is one bias voltage input externally which is a reference voltage off of which the load bias for the preamps in the slices is generated. Since the PMOS load devices are biased in the triode region at a  $V_{gs} - V_t$  on the order of 2 volts, the accuracy with which this voltage is supplied is not very critical. Therefore, a single external reference voltage is input and locally buffered and level shifted with a voltage follower. This resultant voltage labelled *load bias* in the figure is locally coupled with 12 pf of capacitance to the local analog Vdd.

Voltage *nbias* biases the preamps and the latch buffers, while *pbias* is the bias for the input buffers and the voltage labelled *cascode bias* biases up the cascode devices in the preamps.

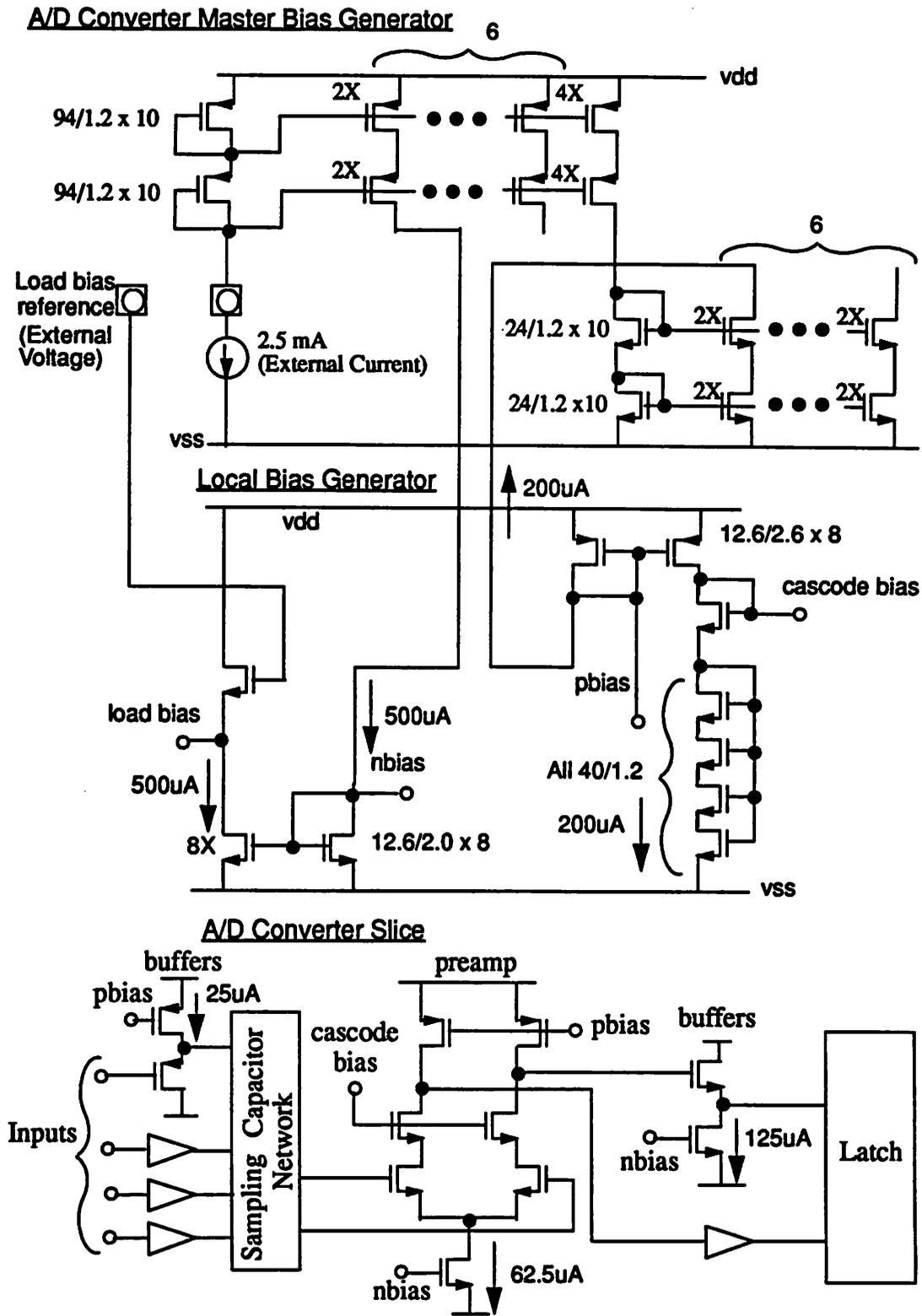


Figure 8.29 Detailed schematic of bias generation starting with the generation of currents (top) into voltage (middle) and the nodes they drive.

### Clock Generation

The next block to be examined is the clock generation in the A/D converter. There are many clock phases that are required in this architecture. However, due to the parallel nature of the approach, the hardware to generate to clocks in one channel is simply repeated four times in each of the three ADC blocks. It is important to have an approach that can easily be repeated. Use of the clock generation block presented in Chapter 6 is used extensively in the ADC clock generator.

Shown in Figure 8.30 is a simplified schematic showing the key blocks in the signal path and some of the key relationships between clock edges required for proper operation. Clock signal *resetl* must surely turn off before *p1an* which is the sampling clock. Next, it is important that *p1an* turn off before *p1bn/p* (*p1bn* and *p1bp*) in order for the sample to be taken without signal dependent charge injection. During phase 2, each MSB reference voltage is subtracted from the input voltage and this difference is presented to a comparator. There are no major clocking issues during this transition. On the next phase, *p13an* needs to go HI to LO before *p13pgn/p* because it results in the start of the latching process before the transmission gates which connect the buffered preamplifier outputs to the latch are removed. These transmission gates are controlled by *p13pgn/p*. It is important to enable the regenerative action in the latching process before turning off the transmission gates because charge injection and clock feedthrough that results from turning off transmission gates can bias the latches in the wrong direction. Also enabled at this time is the decode NOR gate which allows the MSB decision to be latched. This also allows the 3-bit output to occur in the ROM. During phase 4, *p13pgn* goes HI which disables the NOR gate output HI which disallows the accidental pull-down of a bit in the decode ROM. The MSB feedback latch is not allowed to operate during phases 3 and 4.

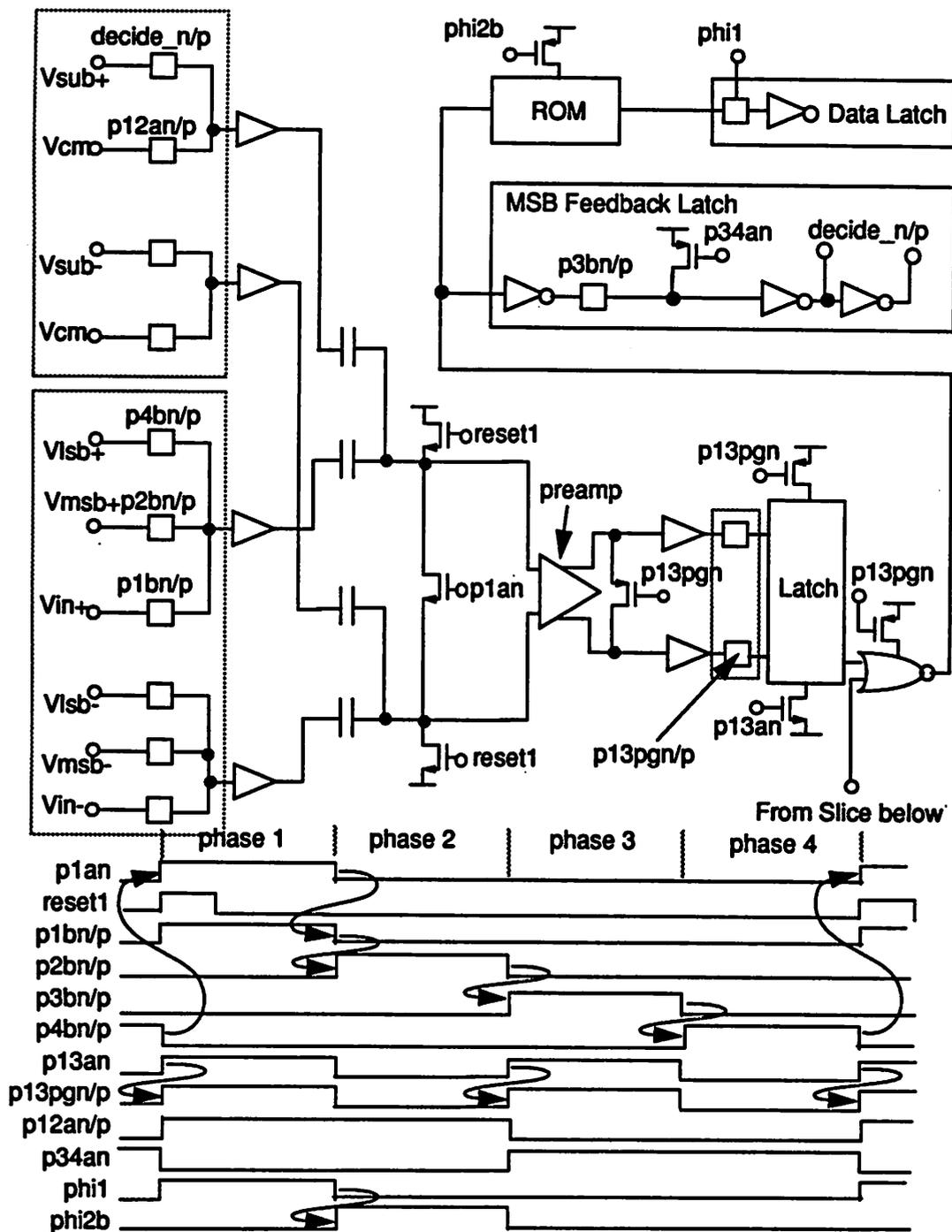


Figure 8.30 Key clocks and their relationships for an A/D converter slice.

Each of the clock phases shown in Figure 8.31 are required for each of the 3 ADC blocks. The notation  $p_{xxx\_cn}$  is used to denote clock phase  $xxx$  for ADC block  $n$  of which there are three. Shown for example are the key sampling clocks for each of the 12 converters with the notation used. Note that there are 4 SubConverter sampling clocks with the  $\_cn$  for  $n$  equal to 1, 2, and 3. Also note that there are 12 different complete phases running simultaneously within the converter.

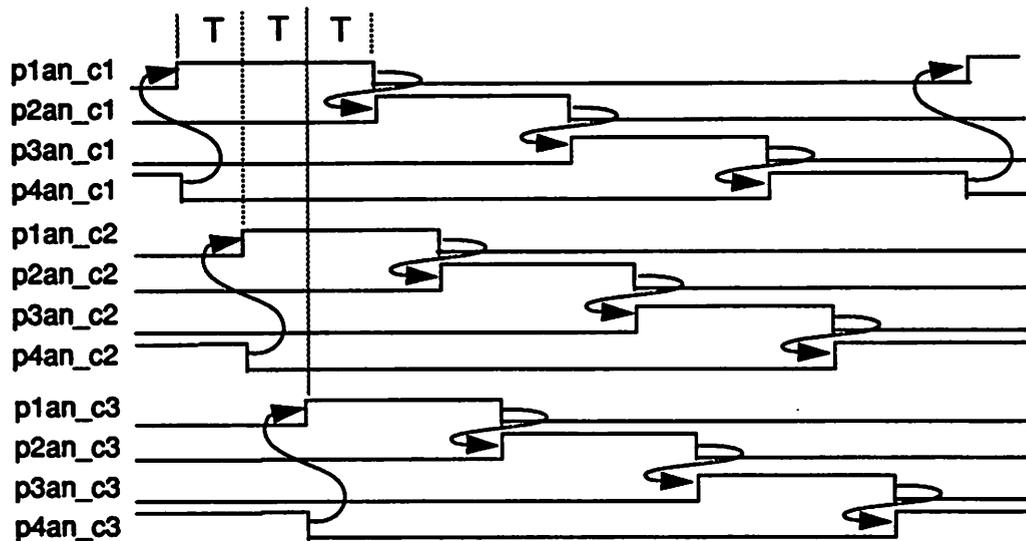


Figure 8.31 Each of the twelve SubConverter sampling clocks.

The last A/D converter block we shall discuss is the output latch block which resamples the outputs of the SubConverters to provide 3 6-bit words at one-third the effective output rate. The notation introduced above will be helpful because some of the locally used converter clocks are used for global resampling of the digital outputs for synchronization prior to output of the digital data.

In order to test the chip, the outputs of the analog-to-digital converter must be sent off-chip through a set of output buffers. Since the converters operate one output period out of phase, at a 100 MHz effective output rate one of the 3 sets of output buffers would need to be sampled every 10 nsec. The approach adopted in the prototype instead incurs resampling of the data on-chip so that 3 words are available from 3 sets of output buffers at one-third the output rate.

***Latching of Data***

Each ADC block contains 4 SubConverters. Let us now examine how the outputs of SubConverter 1 in each of the three ADC blocks makes it to the output. In Figure 8.32, a schematic diagram of one ADC block is shown with a simplified version of the output latches along with the key clocks used to resample the SubConverter outputs. The time during which the MSBs and LSBs of the SubConverter 1 in each ADC blocks is also shown.

Clock waveform *p4an\_c3* is used to sample the MSBs of each SubConverter 1 into point A. Two periods later, *p2an\_c3* samples the LSBs into point B. Each converter block has its own points A and B. After the LSBs are moved into point B, there is a time in which both the MSBs and LSBs are valid in parallel latches. Clock waveform *p3an\_c1* controls the multiplexor which chooses from each of the subconverter outputs selecting the output data words made up of the MSBs and LSBs of the data from each of the SubConverter 1 blocks. A total of 18 output buffers are always active.

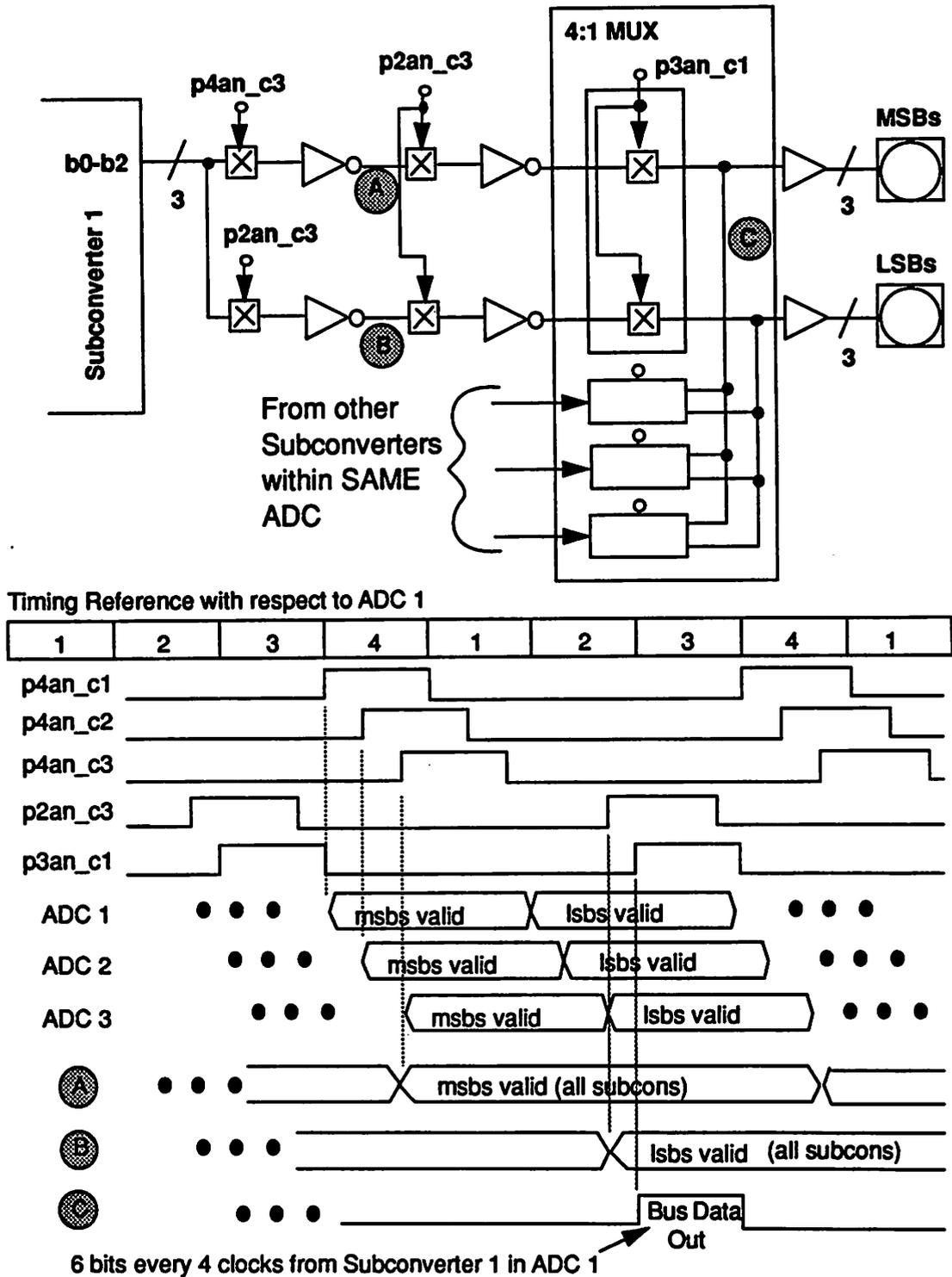


Figure 8.32 Clocking diagram illustrating resampling of output data for output to pads.



referred offset results in a phase error between the output of the last delay element back to the first. Assume the delay of each element is  $T$  seconds, by definition the total delay through  $N$  elements is  $TN$  seconds. Suppose the phase detector has an input referred offset of  $V_{os}$  Volts and the slope of the signal from the ring oscillator is  $m$  Volts/sec. The corresponding time offset of  $V_{os}/m$  from end-to-end of the delay line is divided evenly such that each delay is now  $T+(V_{os}/m)$ . So the error appears to be distributed evenly throughout the delay line. However, in applications such as the SC decimation filter where many of the taps off the delay line are used, it is important for the delay between the last tap and the first to be the same as that of each delay element in the delay line. Therefore in the example of time offset  $V_{os}/m$ , this time offset, though being distributed throughout the delay line appears as a lumped offset between the last tap and the first. The result may be waveforms as shown in Figure 8.34 where the performance appears good through the delay line but a large offset appears between the last and first taps.

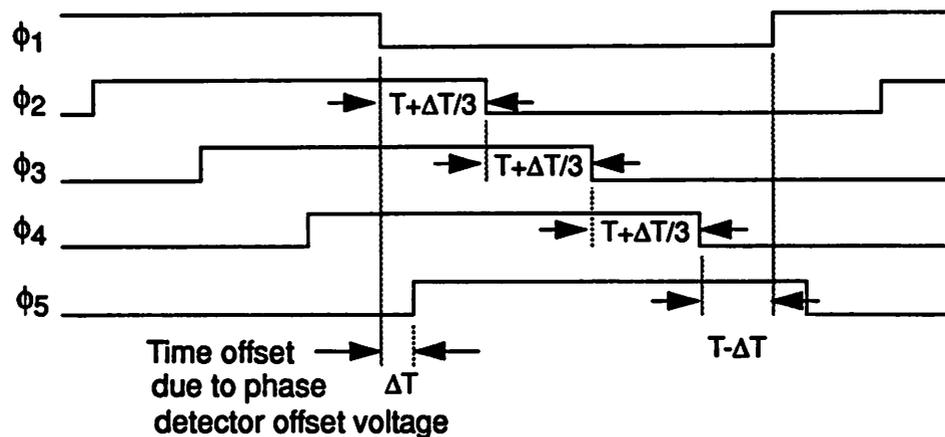


Figure 8.34 Waveforms demonstrating effect of phase offset due to offset voltage in phase detector. Assume phase detector aligns falling edge of  $\phi_1$  with rising edge of  $\phi_5$ .

Another limitation of delay-lock loops is that in sampled-data communication systems the received symbol rate is often different from the frequency of the local clock. And although a frequency offset between the incoming signal and the local clock can be overcome by a time changing phase, the phase granularity required can result in excessive hardware requirements.

Delay-lock loops are attractive from the standpoint of jitter accumulation, but practical limitations of implementation of the phase detector need to be solved before this important advantage can be fully utilized for generation of multiphase clocks in signal processing applications.

Ring oscillators are simply delay lines tied back to itself with a net inversion such that oscillation occurs. By controlling the delay through the delay elements, a voltage controlled oscillator is realized. This VCO can be placed in a conventional phase-lock loop as shown back in Figure 8.33b. One output is tapped off the ring oscillator and compared with an input reference clock by a phase detector. The phase detector output is filtered and the filtered signal acts as the control signal to the VCO. One of the biggest drawbacks of ring oscillators for high speed analog sampled-data applications are that they suffer from accumulation of jitter since a phase perturbation ripples down the delay line and then is fed back around to the beginning of the loop. However, with a careful design approach, the requirements of a 6-bit 50 MHz signal bandwidth which requires roughly 40 psec of rms jitter can be met.

#### *Choose Ring Oscillator-Based Phase Lock Loop*

The first design choice was to use a ring oscillator-based phase-lock loop. Primarily, this choice was made because it results in an overall simpler design, particularly in the phase detector.

#### *Survey of Key Clocks Over Chip*

Let us refer to Figure 8.35 where a sampling of the key clocks in each of the switched-capacitor analog sections is shown. As discussed earlier, there are both local and global clocking issues. For the moment, we shall focus more on global issues.

Referring to the figure, sampling phases *p00*, *p01*, and *p02* are the sampling clocks for three taps making up one impulse response. As can be seen in the figure, these sampling phases need to be centered within the ON period of the multiplexor control clock *l0xp*. From a global clocking point of view, the A/D converter which is the last stage in the pipeline must sample the output of the equalizer blocks before the equalizer opamp clocks change (for example: *plan\_c1* OFF before *e0xp*), the equalizer opamp clocks must switch before the equalizer sampling clocks change (example: *e2xp* OFF before *eqs8*), the equalizer sampling clocks must sample before the lowpass

filter opamp clocks change (example: *eqs0* OFF before *l0xp*), and once again, the lowpass filter sampling clocks must be centered within the ON time of the lowpass filter opamp or multiplexor clocks (example: *p00*, *p01*, and *p02* ON then OFF during *l0xp*).

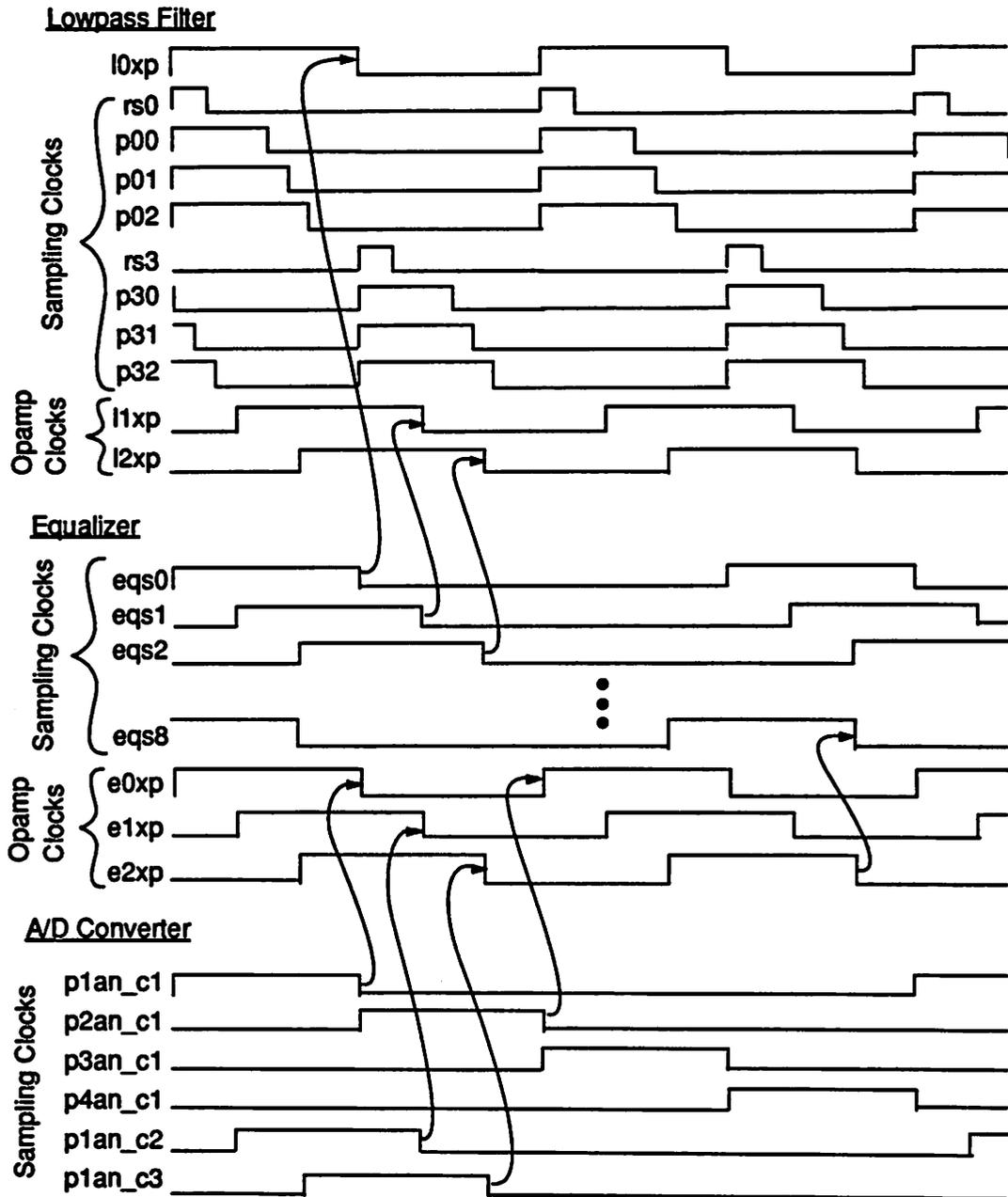
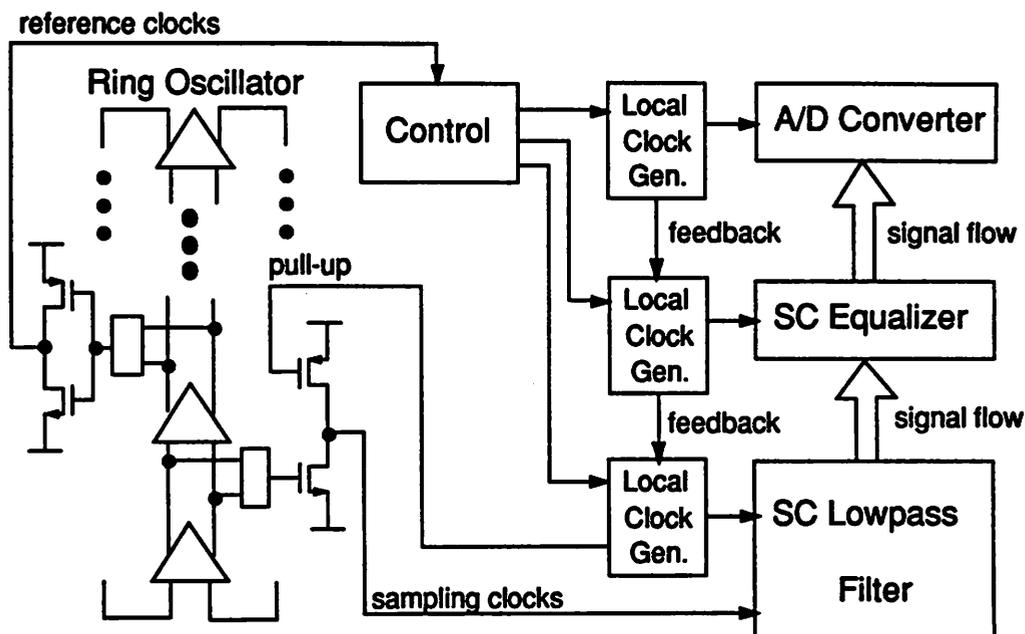


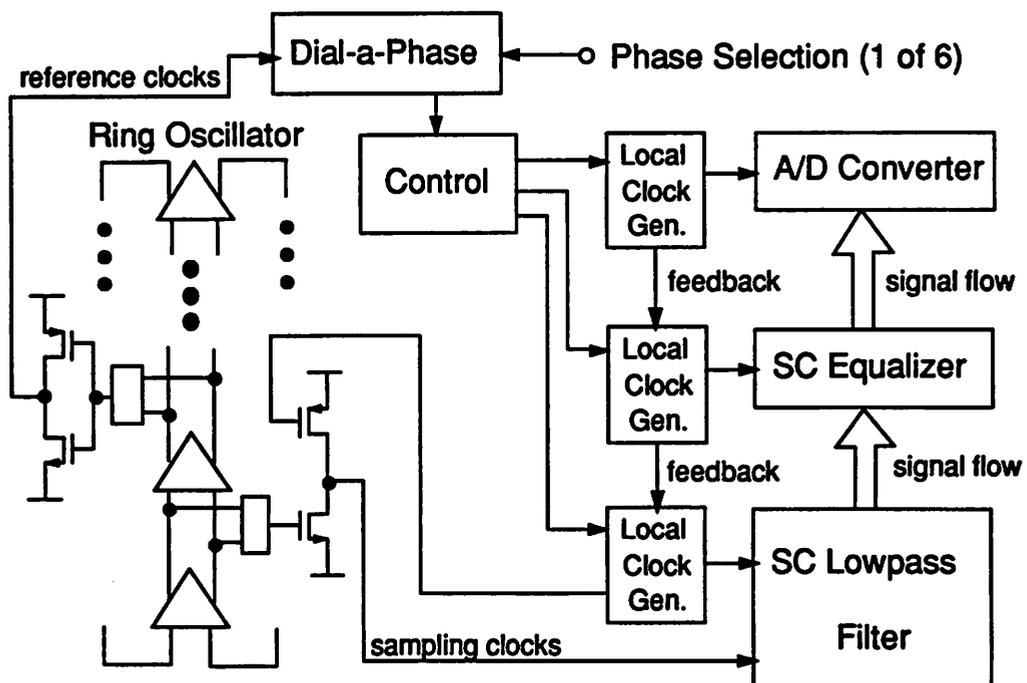
Figure 8.35 Sampling of key clocks required in the prototype.

A high level conceptual diagram of one approach to generation of the clocks in Figure 8.35 is shown in Figure 8.36. The ring oscillator is shown driving circuits that generate *reference clocks* and *sampling clocks*. Note in the figure below that the lowpass filter sampling clocks are generated directly off of the ring oscillator and need to be synchronized with clocks that need to satisfy many layers of non-overlapping conditions. These issues result in delays that cannot be predicted reliably and which make synchronization very difficult. There are actually 6 reference clocks and 18 sampling clocks although only one of each signal is shown. The reference clocks drive a *Control* block which sends control signalling to *local clock generators* for each of the three major switched-capacitor blocks. Information notifying the SC Equalizer local clock generator that the ADC has sampled is shown as *feedback*. A similar feedback signal is shown between the Equalizer clock generator and the SC Lowpass Filter clock generator. In the figure it can be seen how the clock feedback signals feed downward while the signal flow is upward. Notice that the sampling clocks come directly from the ring oscillator and the *pull-up* signals for the PMOS devices in the sampling clock generator which initialize the sampling phases HI comes from the lowpass filter clock generator.



**Figure 8.36** Conceptual diagram showing clock generation in each of the blocks. Note that the sampling clocks come directly off of the ring oscillator whereas the amplifier clocks generated by the local clock generator have a much longer path.

The prototype was designed to operate in a range from 50 MHz to 100 MHz. This wide range makes difficult and unreliable any approach which assumes a particular feedback delay because what may be marginal at 50 MHz may be out of spec at 100 MHz. The solution was the use of a “*dial-a-phase*” circuit as shown in Figure 8.37 which allows the selection of any of the six reference clock phases to be the primary clock through the use of 3 digital selection inputs. This assures that correct alignment of the sampling clocks and multiplexor clocks in the lowpass filter is always possible.



**Figure 8.37** Conceptual diagram of the solution to the problem indicated in Figure 8.36. An externally controlled Dial-a-Phase block aligns the Local Clock Generator block for the SC Lowpass Filter with the sampling clocks.

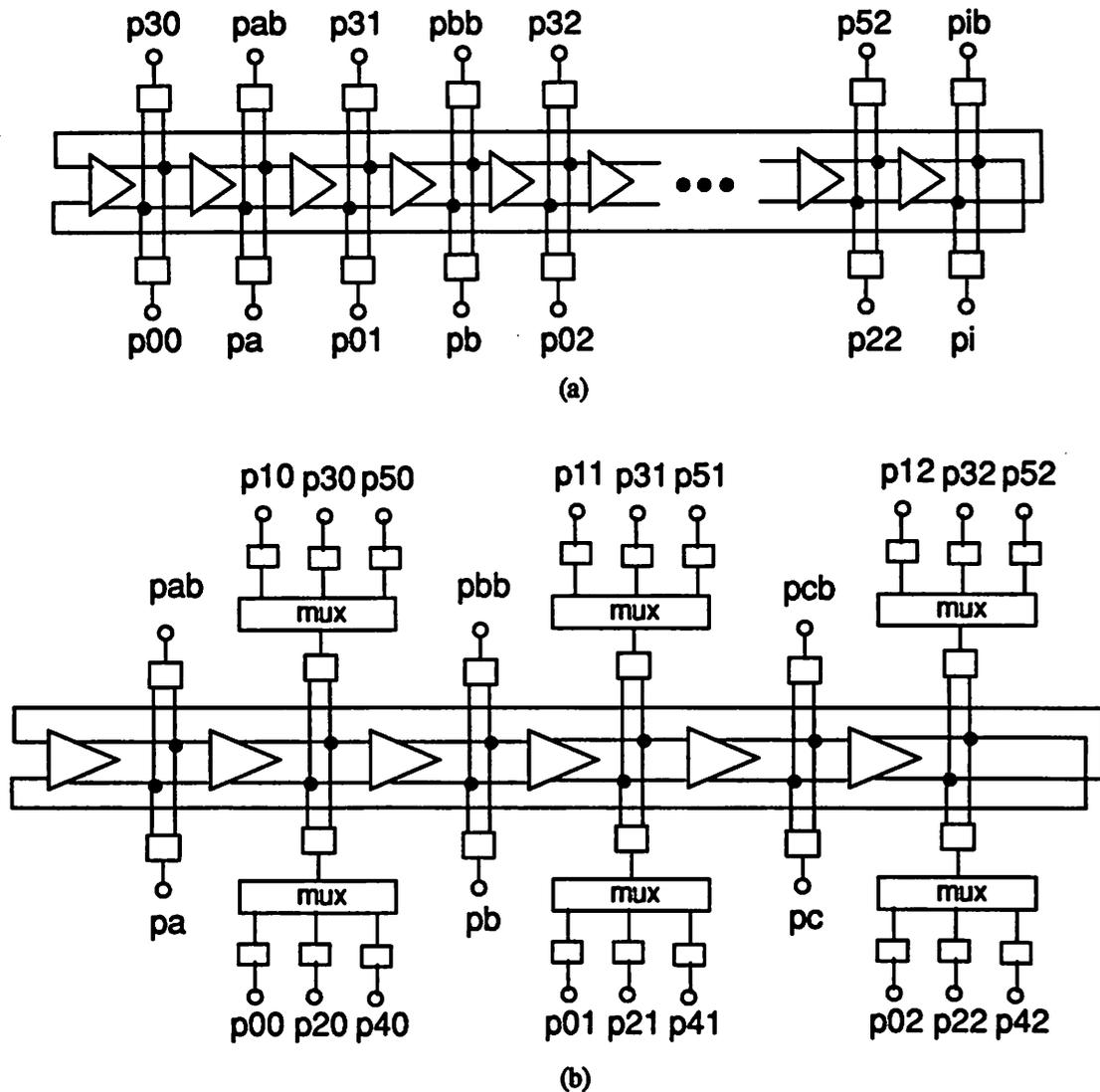
### Ring Oscillator - High Level Design Issues

One of the key considerations and most critical of all the clocks are the input sampling clocks which drive the sampling switches in the decimation filter. They have the most stringent jitter and offset requirements. It is important that the signal path from the ring oscillator to the generation of the sampling phase have as few devices as possible in order to reduce to amount of noise which

will result in sampling jitter and offsets. This explains why the sampling clocks in Figure 8.36 and Figure 8.37 are shown going directly to the lowpass filter block.

From the discussion on the SC lowpass filter earlier in this chapter and Figure 8.13, 18 sampling phases are required in the decimation filter. The required rms jitter for a 6-bit system for each of the sampling phases must be kept below 50 psec. In order to keep jitter to a minimum, it is desirable to keep the delay through a delay element in the ring oscillator on the order of a nanosecond or so. The reason for this is that the signal slopes through the ring oscillator decrease as the delay of an element is increased. The desired delay can be calculated by determining the expected slope of the signals at the output of a delay element at the transition point of the following delay element and dividing this into the expected r.m.s. noise at the input of a delay element. This gives the expected jitter in a delay element which must be multiplied by a jitter enhancement factor for PLLs given in [6]. This is an iterative procedure. In the prototype, each delay element at full speed was designed for a 1.667 nsec delay. Therefore, two elements are needed to get the 3.333 nsec delay as required for a 300 MHz sampling rate. This is attractive because alternating delay element outputs can be used to generate sampling clocks and reference clocks. The pulses created by the differential-to-single ended converters off the ring oscillator activate dynamic pull-down devices for a low jitter sampling phase with a large slope. In the case of the reference clocks generators, these pulses instead drive CMOS inverters.

Two ways the 18 sampling phases and reference clocks can be generated are shown in Figure 8.38 where 18 and 6 delay elements are used. Signals  $p00, p01, \dots$  are the sampling phases and signals  $pa, pab, \dots$  are the reference phases. Since a differential ring oscillator is used, an even number of stages can be used while still achieving a net inversion around the loop. Also, the differential signals provide two phases as the signal passes through the ring oscillator followed by its inverse the next time around. Therefore in the case of 18 delay elements, each delay element providing a differential input to a sampling phase generator has two generators on its output. Each of the generators is connected with a different phase. In the case of 6 delay elements, 3 sampling phase generators are connected in a particular phase to a delay element through a multiplexor. As a result, 3 of the delay elements drive a total of 6 generators for a total of 18 clocks. The other delay elements drive reference clocks.

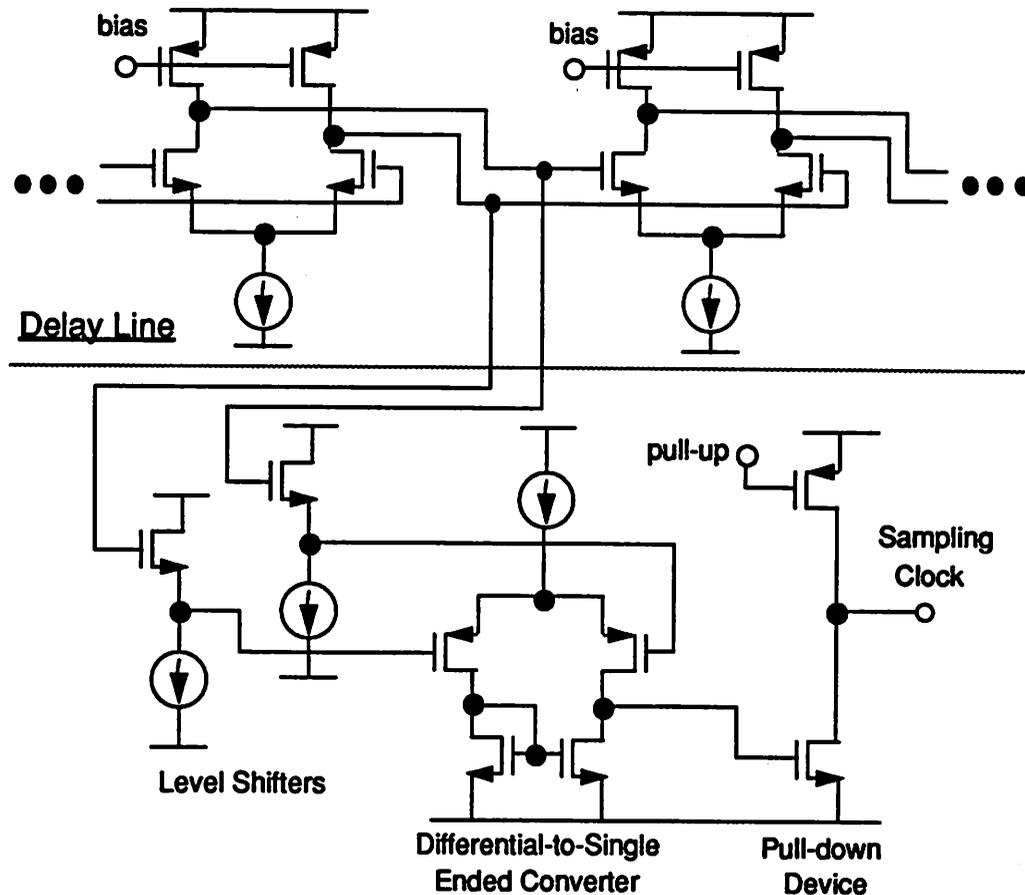


**Figure 8.38** Two ring oscillator approaches for generation of the 18 sampling phases. (a) No sharing delay elements -- uses 36 delay elements. (b) Sharing delay elements -- uses 6 elements.

### Ring Oscillator - Low Level Design

The differential delay element presented in Chapter 6 was used in the ring oscillator and is shown in Figure 8.39. The primary goal of this circuit is to create sampling phases with low jitter. The differential delay element was used in the ring oscillator because it rejects supply noise prevalent in mixed-signal environments. Due to the differential nature of the delay element, this noise

appears as a common-mode component to the signal propagating through the delay line. The resistance of the triode PMOS devices in the delay element controls the voltage swing to be around 1 Volt. As discussed in Chapter 7, in order to create a low jitter sampling clock that rejects supply noise to first order, the differential outputs from the delay line are buffered, level shifted, and drive a differential to single-ended converter which drives a large pull-down device.



**Figure 8.39** Schematic of circuits which generate the single-ended sampling clocks from the differential delay line.

In Figure 8.40, a full schematic of a delay element is shown along with the circuitry required to generate the low jitter sampling phases. In order to assure a wide operating range, the delay element was designed with two bits of control to independently configure both the load and the tail current in a binary weighted manner. The independent control of the load and tail current results in

more flexibility for testing. The input device sizes, size of the load with control signals  $pb1$  and  $pb2$  both LO, and maximum tail current ( $nb1$  and  $nb2$  both HI) were designed to meet the jitter requirements at a signal frequency of 50 MHz using the techniques outlined in [kim and helman].

The voltage followers that follow the delay line buffer the delay line from the loading of the later circuitry as well as provide a level shift from the swing in the ring oscillator which is roughly about a common-mode voltage of 4.5 V to about 2.5 V. These buffers then drive differential-to-single ended converters which are designed for a signal swing of roughly  $\pm .5$  V. The purpose of the differential-to-single ended converters is to create a single-ended driving signal for the NMOS pull-down devices in the sampling clock generators from the differential inputs from the ring oscillator. This function is important in order to reject the common-mode noise on the signals from the delay line.

The PMOS differential pair in the differential-to-single ended converter are biased at a  $V_{gs}-V_t$  of roughly .5 V. In this way, the devices are always drawing some current. This is important because the input signal is quickly changing and it would otherwise take some time, assuming the differential pair went into a slewing condition where all the current was flowing in one device and one was in cutoff, for the device that is in cutoff to come out of that region of operation. As a result, the common-mode rejection of high frequency noise on the supply would be reduced. There is trade-off associated with the design of this block which has to do with the margin with which  $V_{gs}-V_t$  exceeds the single-ended voltage swing in the delay line. The more  $V_{gs}-V_t$  exceeds the voltage swing, the more likely it is that even over process variations, both devices will be drawing current. However, this results in the diversion of more of the tail current in the differential pair away from the load which is the gate of the pull-down devices in the sampling phase generator block. It is important to maintain a sufficiently large slope in the charging voltage to these gates in order to reduce the effects of threshold mismatch in these pull-down devices. This is because the threshold mismatch in the pull-down devices divided by the slope of the driving signal results in sampling phase offsets. Thus, the cost of design robustness to process variations is increased power consumption.

As described earlier, each differential-to-single ended converter output is multiplexed between three sampling clock generators to reduce the hardware requirements. The multiplexor is com-

prised simply of an NMOS pass transistors and an NMOS discharge device which is active after a pull-down event. Operation begins with *ENO* signal turning ON *Mpass0* for the generation of clock *p00*. When *ENO* turns off, the voltage at the output of the differential-to-single ended converter is sampled on the gate of the pull-down device and there is a possibility that this voltage is larger than the threshold voltage. The result is that there will be a fight between the dynamic pull-up device and the pull-down device during the *p00* (active LO) period. Then signal *p00* will simply prematurely fall to the negative supply once the pull-up signal goes HI. Therefore, discharge device *Mdis0* is active during the clock phase opposite *ENO* (*EN3*) assuring the pull-down device to be OFF during the precharge HI of its respective output.

The sampling phase clock generator is comprised of a large NMOS pull-down device, a PMOS pull-up, and the top-half of a static latch. Without the inverter and PMOS pull-up, the large pull-up device brings the sampling clock *pNM* to *Vdd* and then turns off. Since *pNM* in this case is a dynamic node, the charge injection and clock feedthrough due to the PMOS device turning off results in *pNM* exceeding *Vdd*. Variations in this overvoltage from sampler to sampler results in a timing variation when divided by the slope of the falling edge of each respective clock. The addition of the inverter and PMOS pull-up brings the sampling clock voltage back to *Vdd* after the pull-up device is turned off. This additional circuit affects the pull-down process because there is a momentary fight between the pull-down device and the weak PMOS pull-up driven by the inverter during a pull-down event. However, the increased uniformity in the sampling clocks justifies its inclusion.



An important consideration in the layout was that of displacement current resulting in transient IR drops in the supply lines. During the time that an NMOS device is discharging a capacitance (associated with the gate of a sampling device), displacement current flows through it, resulting in transient current in the ground line. If a neighboring NMOS pull-down device begins switching during the time that displacement current is still flowing, this device may require a larger gate voltage to turn on than the first. The IR drop resulting from the transient current flowing in the first device may increase the source voltage of the second, requiring a larger gate voltage to turn the second device on. A simple schematic illustrating this is shown in Figure 8.41. It is important to consider second order phenomenon at these high operating speeds. In the prototype, the impedance back to the off-chip ground will not only be resistive but inductive with distributed capacitance. It is important to consider these parasitic effects.

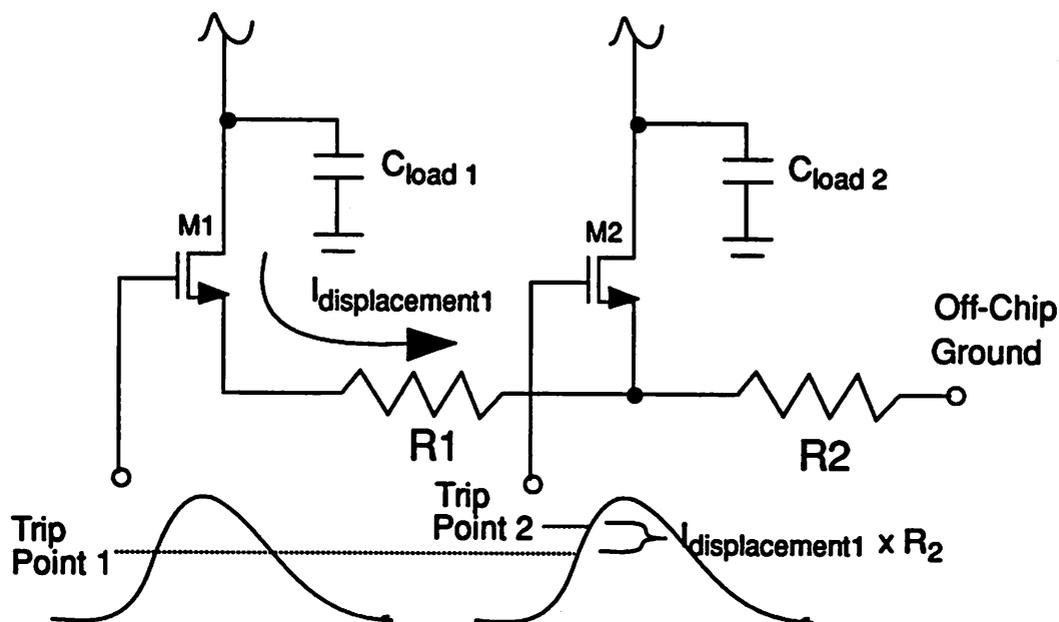


Figure 8.41 Illustration of effect of displacement current of devices on the trip points of others.

### Frequency and Phase Control

The frequency of the ring oscillator is controlled by the tail current in the delay elements. The gates of the PMOS triode load devices are controlled by a circuit which monitors the tail current in the delay elements and provides a bias voltage to these gates in order to maintain a voltage swing

of 1 Volt. A conceptual schematic of the tail current control circuit is shown along with a full transistor level implementation in Figure 8.42. Together with the ring oscillator, the circuits shown in Figure 8.42 make up a charge pump phase-lock loop [3]. An input clock at one-half the effective sampling rate of the ADC is the reference clock into the phase detector. The local clock is one of the reference clocks from the ring oscillator.

Let us first refer to Figure 8.42a and discuss the overall architecture of the phase-lock loop. The frequency and phase of the input clocks are compared by the frequency-phase detector which drives a charge pump. The charge pump converts the frequency or phase error into a current for a time and with a polarity dependant upon the error. This current drives an RC network which makes up the loop filter. The voltage across this filter is the filtered phase detector output which then drives a transconductor which converts this voltage to a current. The output of the transconductor is summed with a reference current source into an NMOS device, the current of which is mirrored in tail current generators of the delay lines. The reference current source controls the free-running frequency of the ring oscillator. The transconductor then add or subtracts current from the reference current source to lock the local clock to the reference clock.

It may appear inefficient to perform the current-to-voltage and voltage-to-current transformations required in this approach; however, this series of transformations greatly facilitates the signal processing. Performing the loop filtering with a current into an impedance allows the use of a passive filter requiring no operational amplifier. Furthermore, summing the currents into the diode connected transistor results in an efficient, high bandwidth summation again, requiring no operational amplifier.

Let us now refer to Figure 8.42b where the transistor implementation is shown. A gate level implementation of the frequency-phase-detector will be shown later. The output of the frequency/detector informs the charge pump to pump up, down, or both up and down which would result in an increase, decrease, or no change in the ring oscillator frequency, respectively. Note that the current from the charge pump either goes to the load or to a voltage reference  $V_{mid}$  which sits roughly between the two supplies. The purpose of this diversion is to keep the current source devices in saturation and to keep the voltage across the devices nearly constant. Consider the PMOS current source in the charge pump. Without the circuit that diverts the current to  $V_{mid}$  when the PMOS

switch to the load is off, the drain of the current source device would eventually charge up to  $V_{dd}$ . When the PMOS switch was eventually turned on, it would take a finite amount of time for the current source device to come out of the deep triode region. Furthermore, any capacitance on the drain node would be charged up to  $V_{dd}$  and this charge would be redistributed to the load during the ON transition of the PMOS switch. The diversion circuit greatly reduces these unwanted transients.

Since the loop filter is off-chip, it is very likely that supply and clock noise will couple into the filter resulting in jitter of the ring oscillator phase. A differential architecture was adopted as shown in the figure. A reference current into an external resistor sets the common-mode level of the transconductor input. In parallel with this resistor is a replica of the loop filter which presents a similar impedance at high frequencies to coupling phenomenon. Therefore, any coupling appears as a common-mode signal present on both the reference node and the signal node driven by the charge pump which is then reduced by the differential-to-single ended conversion performed by the transconductor during the voltage-to-current conversion. The external  $I_{ref}$  is simply mirrored and summed with the transconductor output into the tail current bias device.

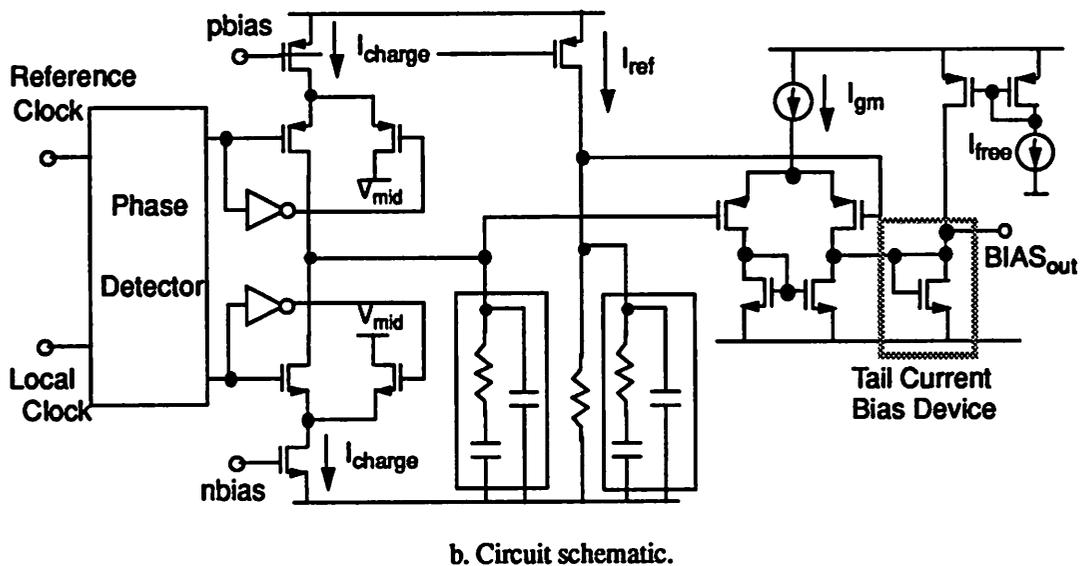
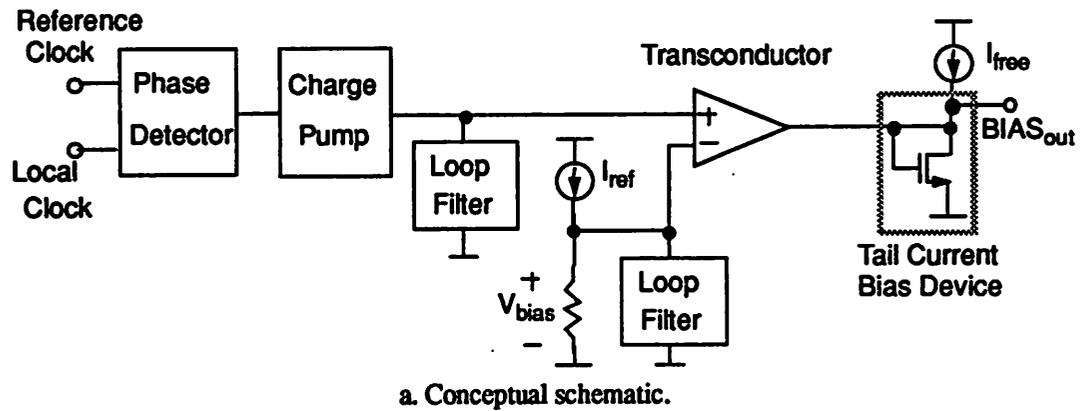


Figure 8.42 Conceptual and circuit schematics for tail current bias generation.

The bias voltage generator for the PMOS triode load device was presented in [7] and shown in Figure 8.43. This circuit biases the triode load bias in the delay element with a voltage which results in a voltage swing in the delay line defined by the NMOS transistor  $M_{\text{swing}}$ . Transistor  $M_1$  is a replica device of the triode load device in the delay element. Transistor  $M_2$  mimics the on-resistance of the selection switches which allow for switching in/out parallel transistors for a wide programmable operating frequency range. The tail current of the delay line generated by the circuit



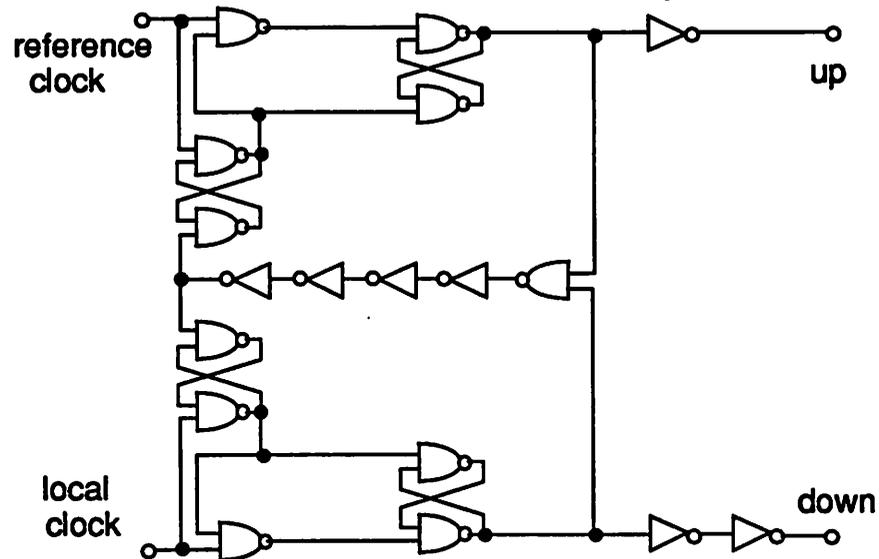


Figure 8.44 Frequency-phase detector.

Up to this point, we have examined the operation of the ring oscillator and how it generates the clocks which effect the high sampling rate. We have also discussed the non-overlap clock generator building block and how this block can be used to meet both local as well as global non-overlap clock issues. We shall now examine how reference clocks from the ring oscillator are used to generate control signals for the non-overlap clock generators.

Let us first consider the control clocks to the A/D converter. Refer to Figure 8.45 where the sampling clocks  $p1an-p4an$  for ADC Block 1 are shown with the circuits used to generate these four clocks along with the four input clocks required by the circuit. The signals  $\overline{a0}$ ,  $\overline{a3}$ ,  $\overline{a6}$ , and  $\overline{a9}$  are named so simply by convention. These signals need to be 3 output periods apart and for a later circuit, be HI for roughly one-third an output period.

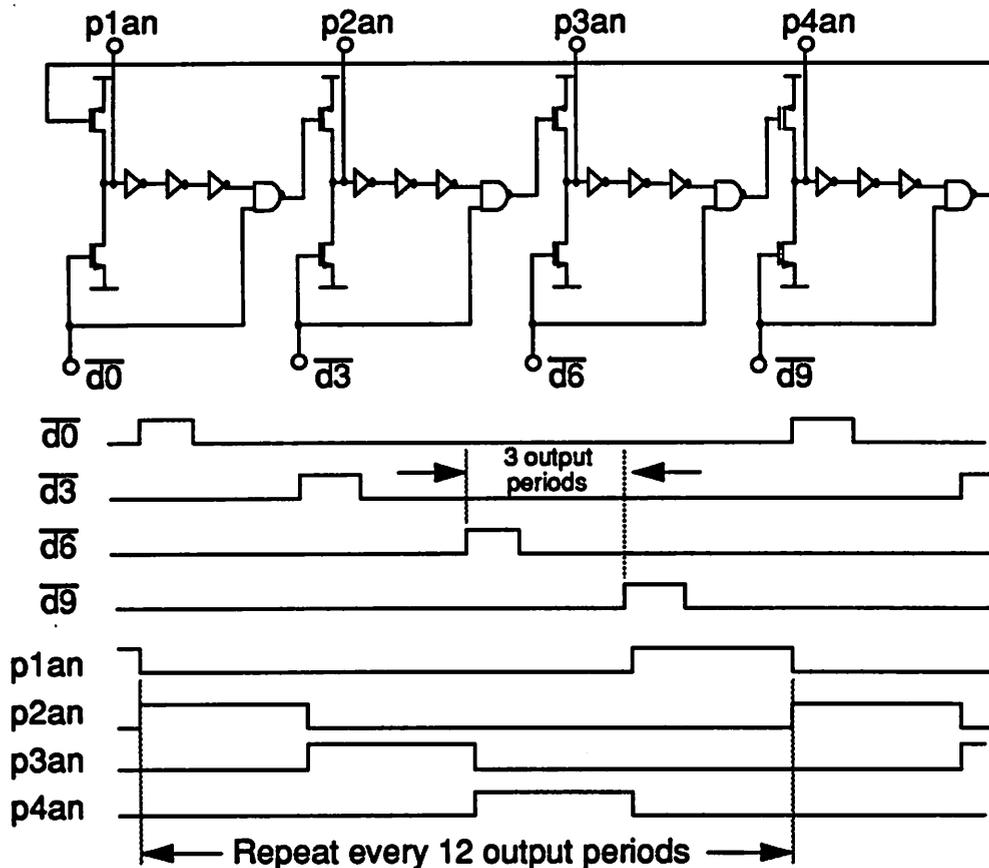


Figure 8.45 Control clocks  $\bar{d}0$ - $\bar{d}9$  and clocks they generate p1an-p4an.

The reference clocks from the ring oscillator give an output at one-half the output rate. Two of the reference clocks from the oscillator are labelled as  $pa$  and  $pab$  in Figure 8.46 along with a sampling of the compliments of the signals required by the generator circuits shown in the previous figure. The signals shown below are locally inverted in the A/D converter clock generation circuitry in order to regenerate the signals and locally buffer them. A total of 12  $d$ -clocks are required starting from  $d0$ - $d11$ . As we can see from Figure 8.46, 4  $d$ -clocks are required for each of the 3 ADC Blocks.

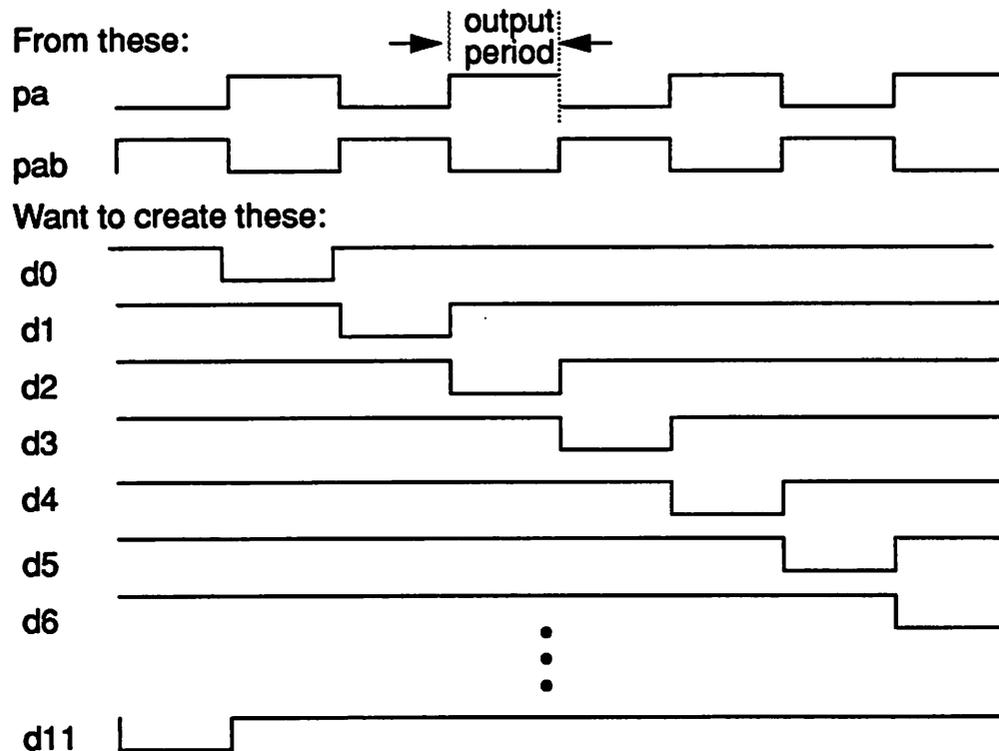


Figure 8.46 Reference clocks *pa* and *pab* used to generate *d0-d11*.

One way to generate the d-clocks given only reference clock *pa* and *pab* is shown in Figure 8.47. A counter (not shown) using *pab* as input counts to 12 (0-11). During every count of “1”, a HI signal is input to the left most d-latch clocked by reference clock *pa*. This logic HI signal acts as an enable signal to the buffers enabling a pull-down of the output. A schematic of the buffer with the pull-down-enable is shown in the figure. The pull-down-enable is simply enabled by an NMOS device in series with the inverter NMOS device. The output signal then transitions LO to HI on the next HI to LO transition of *pab*. As the logic 1 propagates down the delay line, it enables subsequent buffers to perform the pull-down function. Thus, every time *pa* transitions from LO to HI, a different output transitions HI to LO.

Using this approach, the even numbered clocks from *d0-d10* are generated.

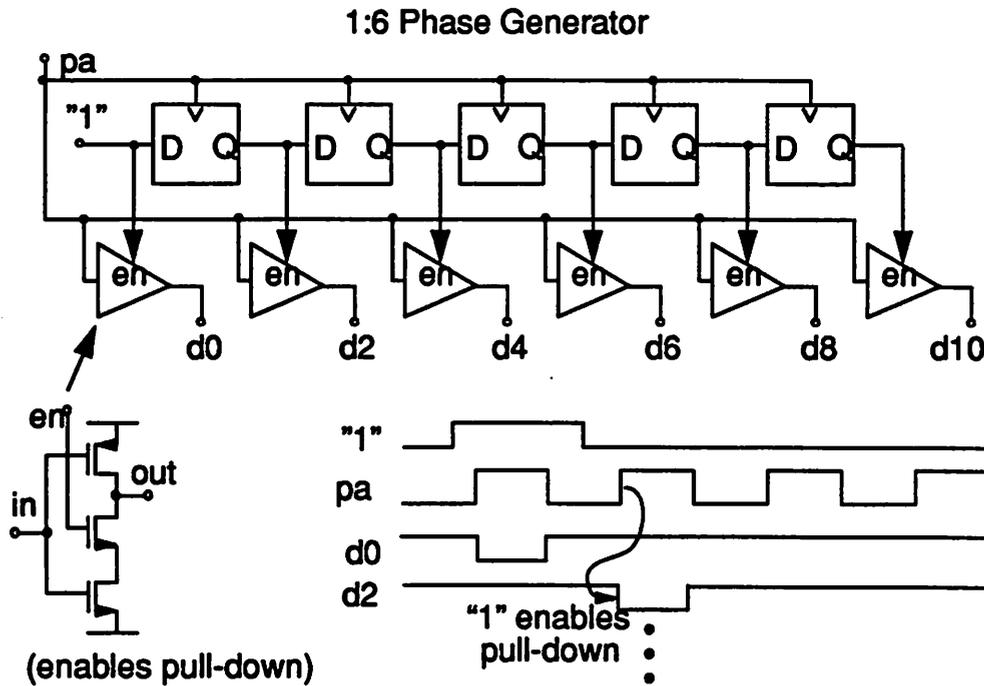


Figure 8.47 Block diagram of a 1:6 Phase Generator which generates  $d0-d10$  as shown from input clock  $pa$ .

By placing two of the clock generators shown in Figure 8.48 in parallel and clocking one off of  $pa$  and the other with  $pab$ , control clocks  $d0-d11$  can be generated as shown in Figure 8.48. Thus, all twelve  $d$ -clocks can be generated off of  $pa$  and  $pab$ .

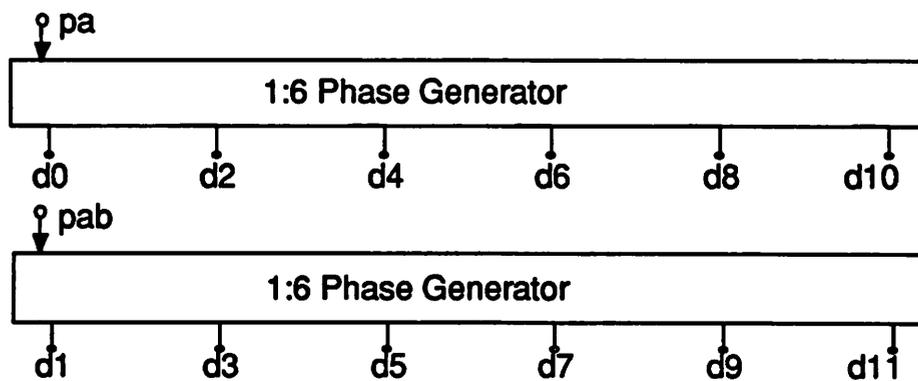
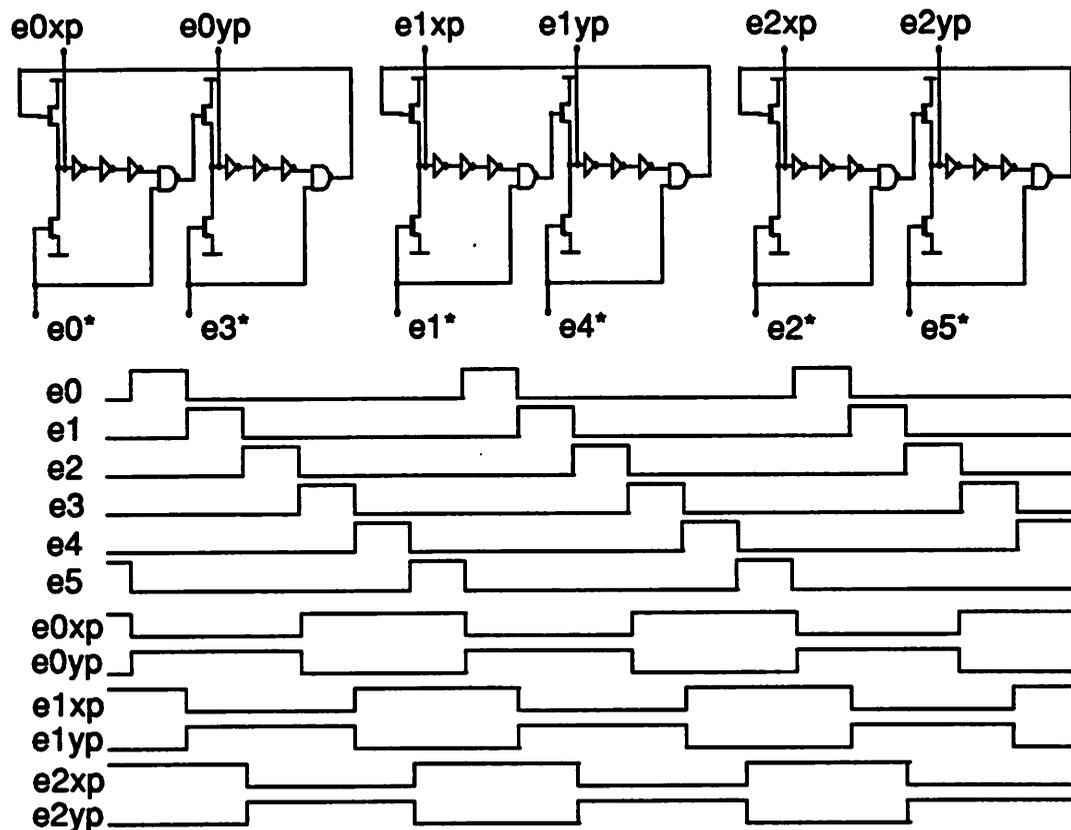


Figure 8.48 Two sets of 1:6 Phase Generators using input clock phases 180 degrees apart generate clocks  $d0-d11$ .

Let us next consider the signals which clock the opamps. These are the 6 phase clocks required by each amplifier section in both the SC lowpass filter and equalizer blocks. These require a total of 6 input clocks as shown in Figure 8.49 where the clocks for the equalizer section are shown. Associated with  $e0xp$  and  $e0yp$  are signals  $e0xpp$ ,  $e0xppp$ ,  $e0ypp$ , and  $e0yppp$ . Thus, only two of the 6 phases for each amplifier section are shown. The same input control clocks  $e0-e5$  go to the local clock generators of both the lowpass filter and the equalizer. In the equalizer, they are logically ANDed with the appropriate feedback signals from the ADC and in the lowpass filter, from the equalizer sampling clocks.



**Figure 8.49** Non-overlap clocks for the amplifiers in both the equalizer and SC lowpass filter generated off of clocks  $e0-e5$ . In both cases, clocks  $e0-e5$  are combined with appropriate feedback circuitry and this is indicated by the \* suffix. Equalizer opamp labels are shown.

Control clocks  $e0-e5$  can easily be generated by performing a NAND function on the “d” clocks used to control the A/D converter input clocks as shown in Figure 8.50.

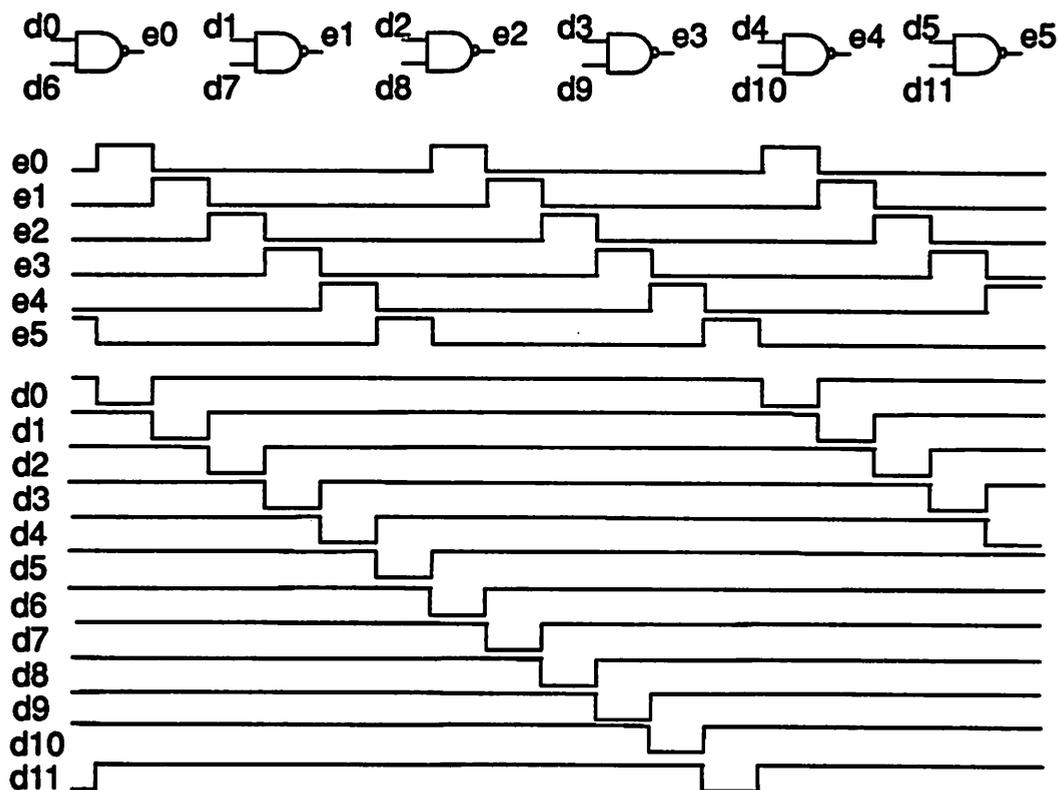


Figure 8.50 Clocks  $e0$ - $e5$  are easily generated from clocks  $d0$ - $d11$ .

The equalizer sampling capacitors require nine clock phases as shown in discussion of the equalizer design earlier in this chapter. A circuit block similar to that shown in Figure 8.47 is used to generate a set of 36 clock phases  $u0$ - $u35$ , each of which are active for three periods. A different set of four of the thirty-six clock phases controls one of nine circuit blocks shown in Figure 8.51. Each of these blocks generates one of the 9 clocks  $eqs0$ - $eqs8$  used as sampling clocks in the equalizer.

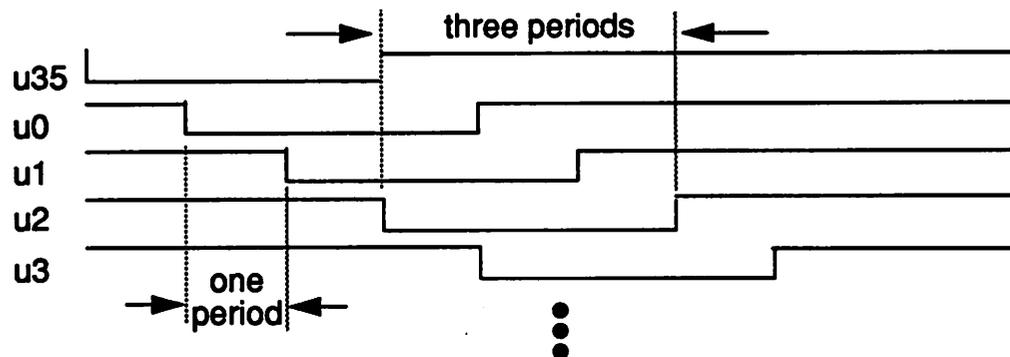


Figure 8.51 Clock phases  $u0$ - $u35$  from which  $eqs0$ - $eqs8$  clocks are generated.

The equalizer sampling clocks also function as the dump or evaluate clocks for the SC equalizer block. Therefore, the equalizer sampling clock phase preceding by 3 output periods another sampling phase must be LO before the latter phase can go HI to ensure that only one set of three capacitors are connected to the amplifier summing nodes at any one time. This is achieved as shown in the example of Figure 8.52 where  $eqs6$  is buffered through two inverters before enabling the LO to HI transition of  $eqs0$ . Further delay is incurred through additional buffering of the output of the NAND gate which assures the non-overlapping of phases 3 phases apart.

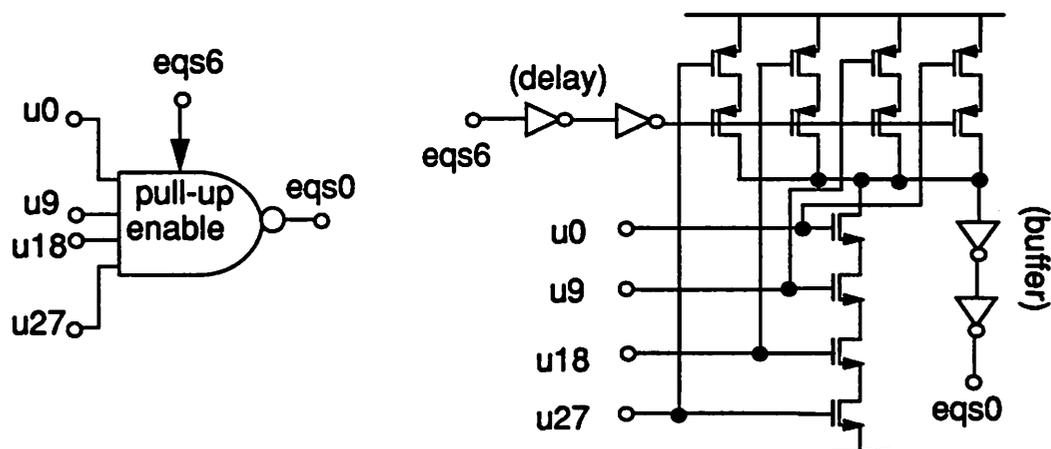


Figure 8.52 NAND gate with pull-up enable for generating clocks  $eqs0$ - $eqs8$ . The enable signal assures only one  $eqs$  signal is HI at one time.

Summary of Global Non-Overlapping Clock Issues

Global non-overlapping clock issues are met using the approach shown schematically in Figure 8.53. Let us discuss each of the key issues starting with the A/D converter sampling clocks. Signals  $d0-d11$  from the Clock Generation Control Block result in the firing of the sampling clocks in the A/D Converter. Feedback clocks from the ADC are passed back to the equalizer clock generator. Signals  $e0-e5$  are passed from the Clock Generation Control Block to both the equalizer and lowpass filter clock generators. In the equalizer section, these signals are ANDed with the feedback clocks from the ADC feedback lines to generate the equalizer opamp control clocks. Similarly, in the lowpass filter section, signals  $e0-e5$  are ANDed with feedback lines derived from the equalizer sampling clocks  $eqs0-eqs8$ .

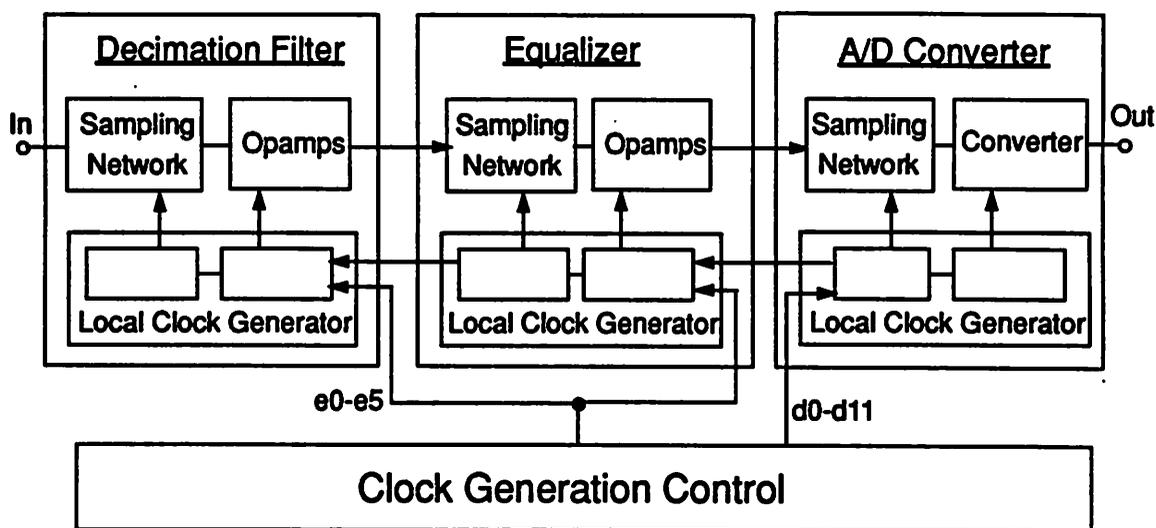
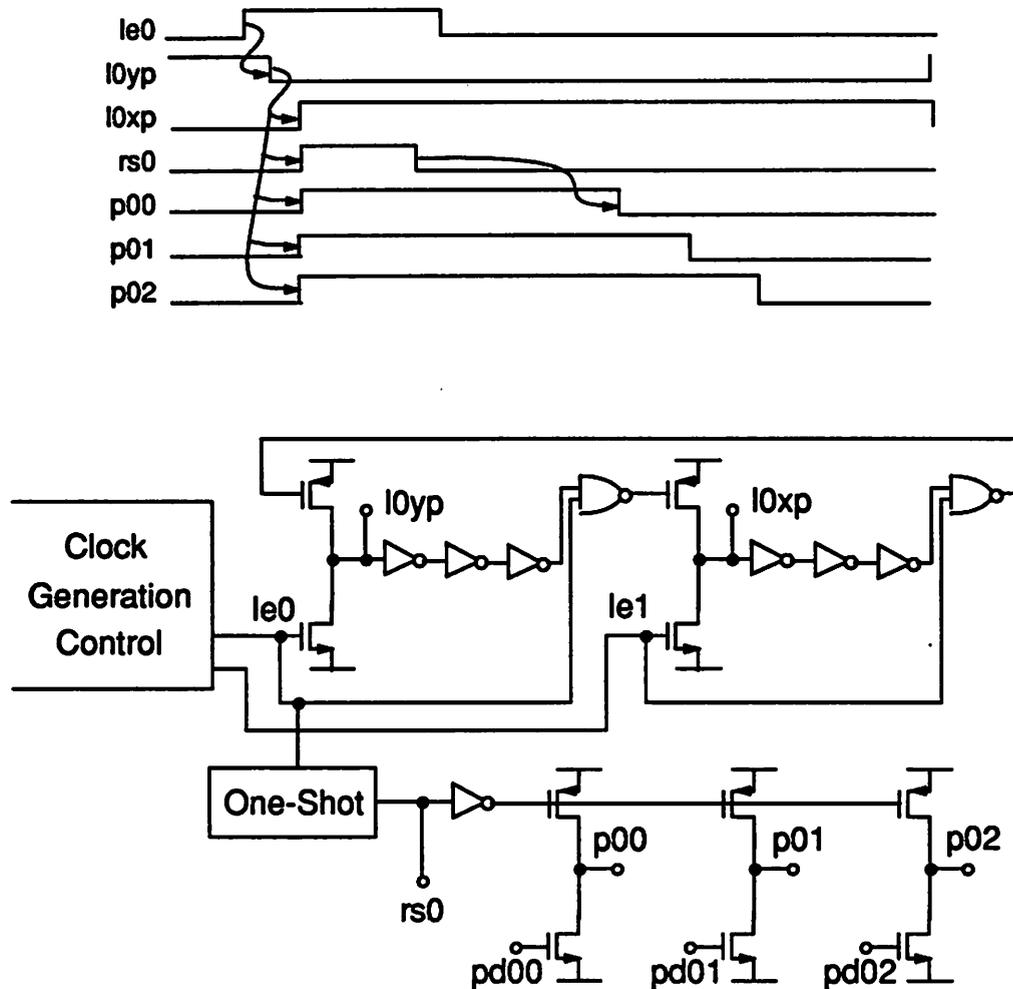


Figure 8.53 Routing of key control clocks and feedback lines to meet global clocking issues.

Two important clocking issues still need to be addressed. The first is the relationship between the equalizer sampling clocks  $eqs0-eqs8$  and the equalizer opamp clocks. These sampling clocks also act as the dump or evaluate clocks for the equalizer section and therefore must lag the opamp clocks so that the zero-order hold circuitry of an opamp can sample the opamp output prior to the removal of the input. This relationship is assured in an open-loop manner relying on the delay inherent between the generation of clocks  $e0-e5$  and sampling clocks  $eqs0-eqs8$ . A number of gate delays exists between these clocks which assures the required timing conditions are met.

The second is alignment of the input common-mode reset clocks for the sampling network of the decimation filter with the dynamic pull-up of the high sampling rate clocks  $p00$ - $p52$ . These waveforms are shown with their required relationships along with a simplified schematic of the circuitry used to realize these relationships in Figure 8.54.

### Sampling of Lowpass Filter Clocks



**Figure 8.54** Synchronization of the multiplexor clock  $l0xp$  with reset clock  $rs0$  and the pull-up of sampling clocks  $p00$ - $p02$ .

Clock  $le0$  which is the pull-down signal for  $l0yp$  results in the LO to HI transition of signal  $l0xp$  and is the ANDed version of signal  $e0$  with the appropriate equalizer sampling clock. Upon

the LO to HI transition of  $l0xp$ , the common-mode reset clock  $rs0$  must become active while the sampling phase clocks  $p00-p02$  turn ON. The  $le0$  signal drives a one-shot whose output  $rs0$  is active HI for 5-7 nsec. Clock  $rs0$  is then inverted and becomes the pull-up signal for the dynamic reset in the sampling clock generators. This approach ensures that the common-mode reset and turn ON of the sampling phases occurs after the turn ON of the input multiplexor ( $l0xpp$  and  $l0xppp$  clocks also active although not labelled).

## 8.7 Experimental Results

In this section, we will present experimental results taken from prototype devices. A diagram of the system used for testing is shown in Figure 8.55. The ring oscillator requires a clock at one-half the output rate while the three parallel A/D converters simultaneously output data at one-third the output rate. This means that for a 100 MHz output rate of the chip, this clock provided by the HP 8131A Pulse Generator must be divided by 2 for the PLL in the prototype to lock to and by 3 for the HP 16500 Logic Analyzer which acts as a data buffer for the ADC output. Each of the words is stored 512 words deep. With 3 words in parallel, this results in a total of 1536 words for processing. Only the first 512 are taken since this is sufficient for characterizing the 6-bit ADC. The memory of the logic analyzer is then dumped to the PC via a GP-IB bus at a much lower rate. The signal is then evaluated on the PC using a set of programs written by Monte Mar [9].

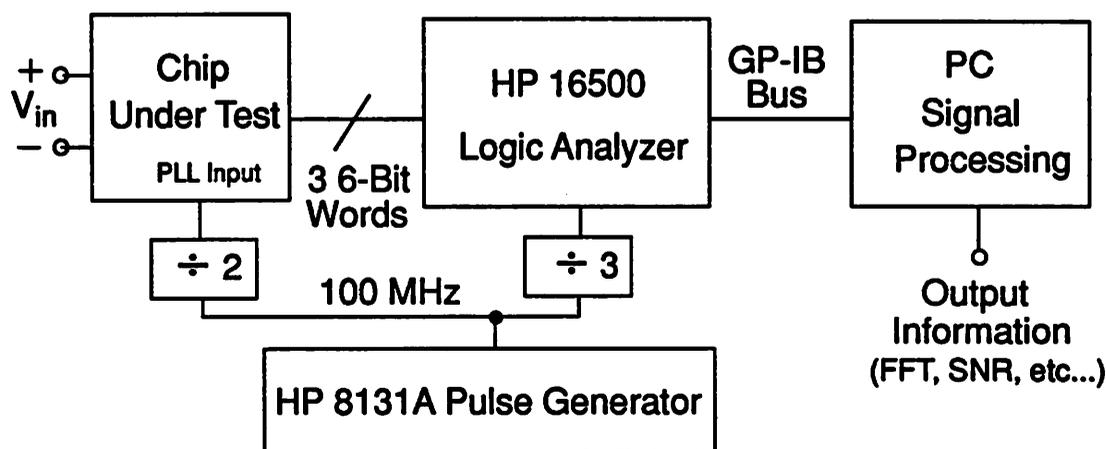
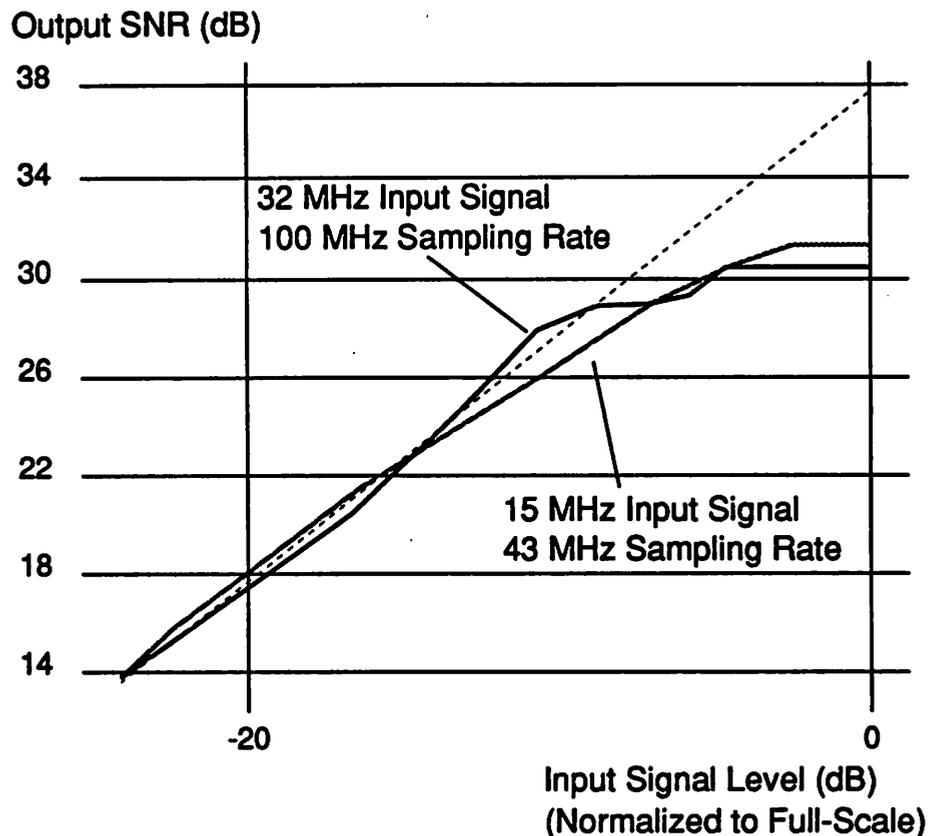


Figure 8.55 Block diagram of the test set-up.

***SNR vs. Input Signal Level***

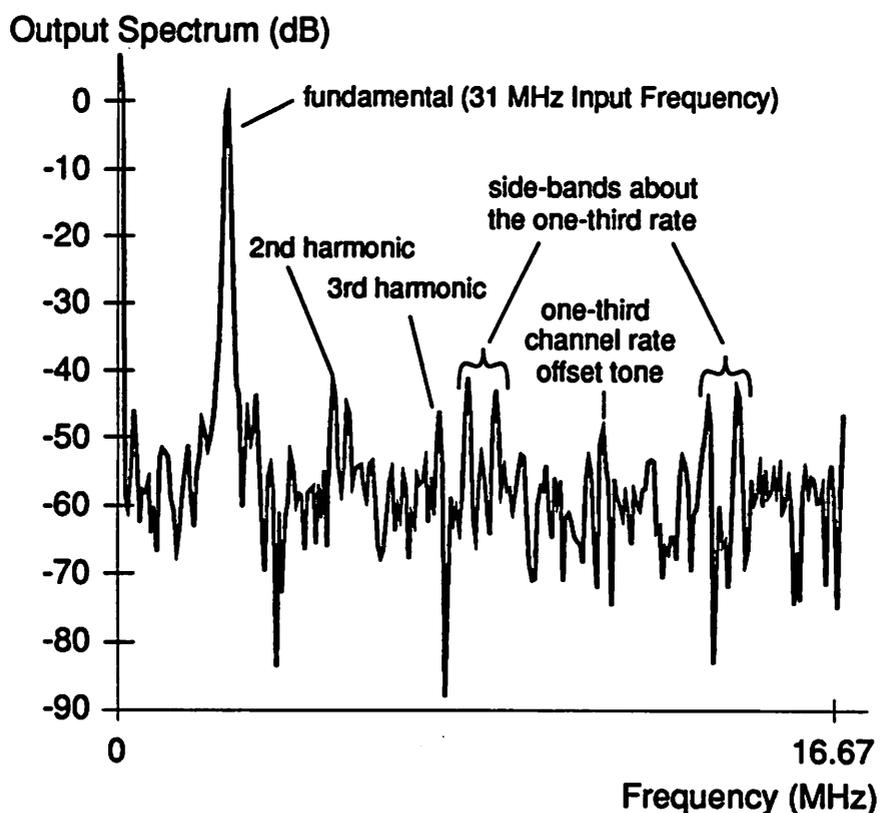
One way to characterize the linearity and dynamic range of an A/D converter is to plot the SNR versus the input signal level as shown in Figure 8.56. Ideally, this would be a straight line but due to distortion and other signal dependent non-ideal effects, the SNR saturates at higher input levels as can be seen in the figure. Shown plotted are the conditions where the output sampling rate is 43 MHz and 100 MHz with input signal frequencies of 15 MHz and 32 MHz, respectively. The plots at low signal levels extrapolates to 6-bits of linearity.



**Figure 8.56** Plots of the output SNR as a function of the input signal level for a 43 MHz and a 100 MHz output rate.

*Effects of non-idealities*

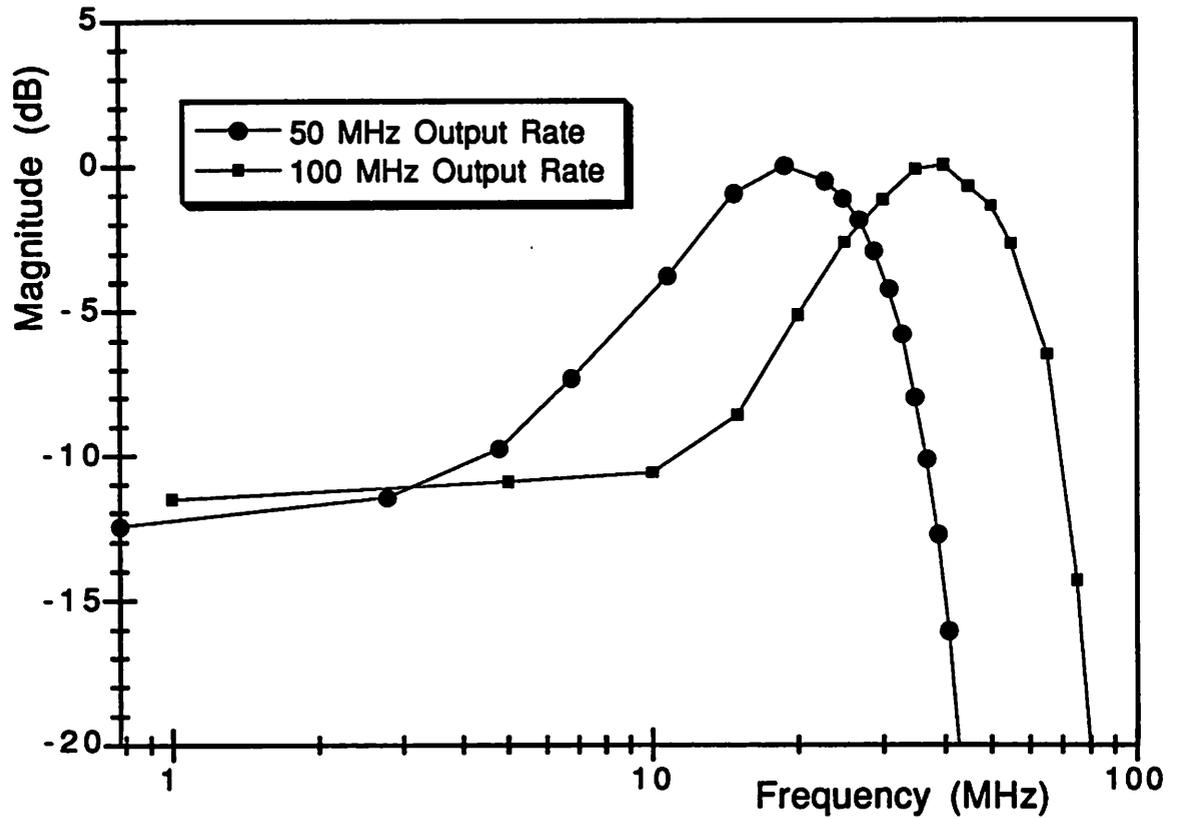
In order to illustrate some of the non-idealities, the output spectrum of one of the three parallel channels is plotted for an effective output rate of 100 MHz and an input signal frequency of 31 MHz in Figure 8.57. The signal has been sub-Nyquist sampled in order to facilitate analysis of the non-idealities. As the signal level is increased, spikes other than the fundamental begin to appear in the frequency domain. The second and third harmonic distortion are indicated in the figure. The other sidebands are due to gain mismatch as discussed in Chapter 5. Since these sidebands appear to be symmetric about one-third the single channel rate, this suggests that they are due to mismatches in the three sets of input sampling capacitors in the programmable equalizer stage.



**Figure 8.57** Output spectrum of one of the three parallel channels at an effective output rate of 100 MHz (33 MHz per channel). The input fundamental is at 31 MHz. Decimation is present in this spectrum.

***Frequency Response Plots***

The frequency response through the channel is shown for the NOMINAL equalizer setting at 50 MHz and 100 MHz output rates in Figure 8.58.



**Figure 8.58** Frequency response at 50 MHz and 100 MHz output rates.

### *Jitter Test*

One way to measure the jitter performance is to input a signal at different frequencies and observe the FFT for changes in the noise floor. As the signal frequency is increased, the noise floor would increase if jitter were limiting performance. The plots shown in Figure 8.59 are the output spectra for input signals at 12 MHz and 80 MHz. Since the noise floor appears independent of signal frequency, the jitter specification is being met.

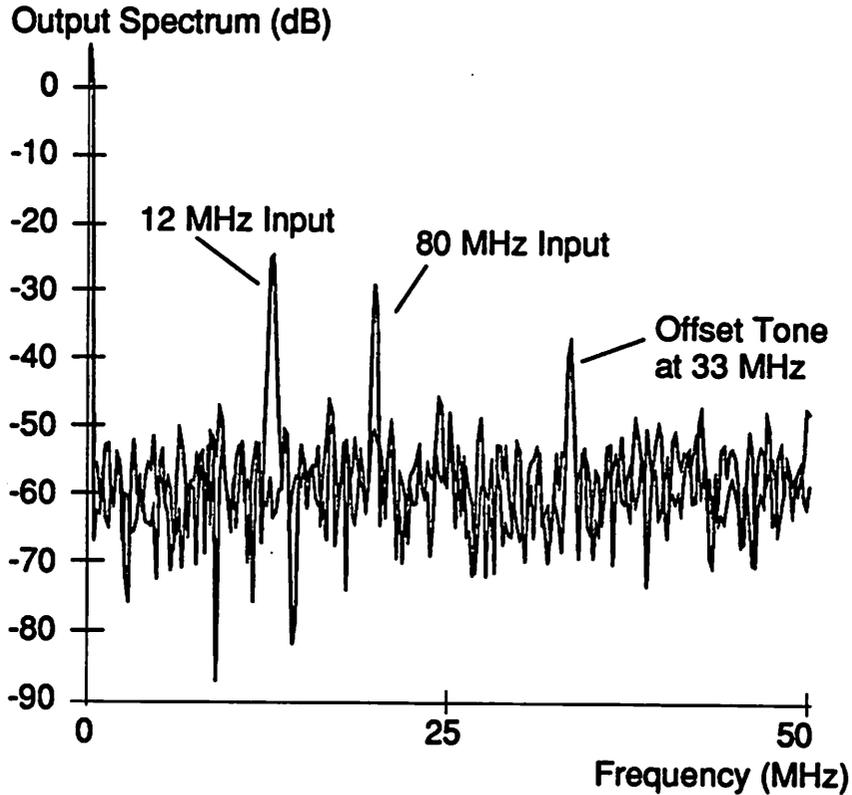


Figure 8.59 Plots to illustrate jitter performance. Output spectra for input signals at 12 MHz and 80 MHz.

## 8.8 Prototype Layout

In this section, we will present a layout of the chip and consider block-by-block power consumption.

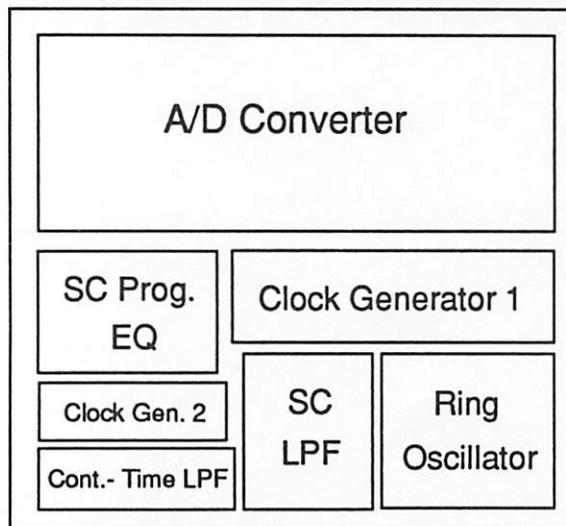
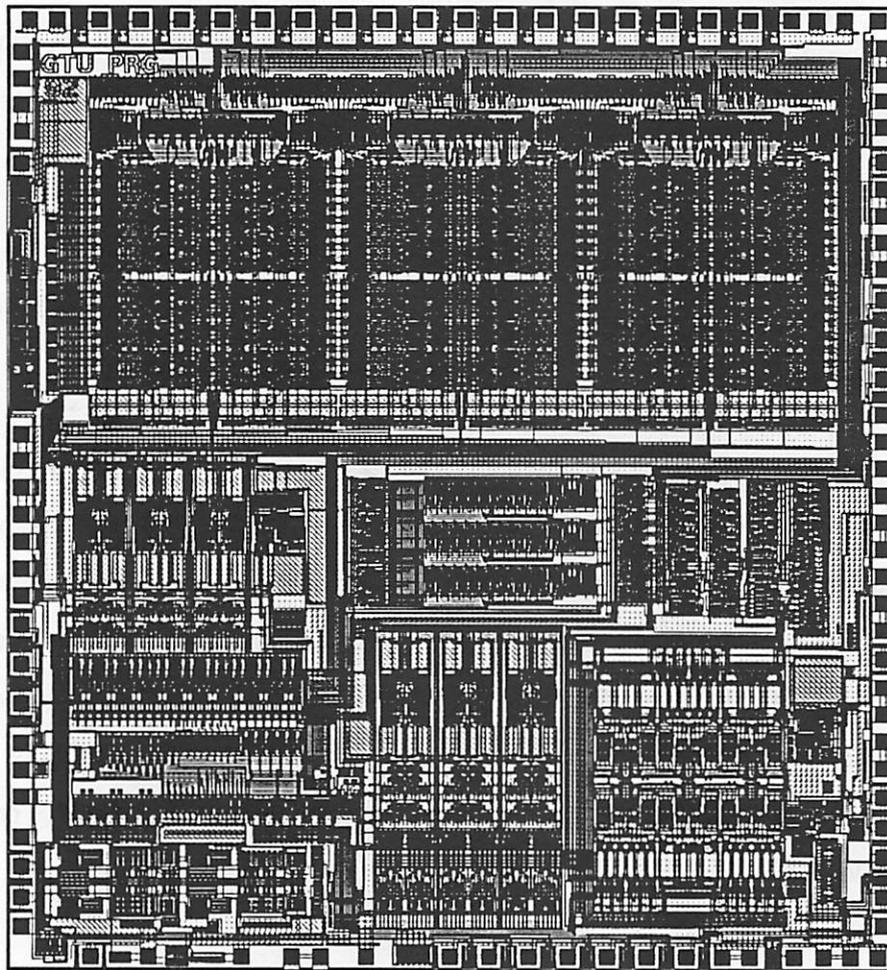
### 8.8.1 Layout

A cifplot of the layout is shown in Figure 8.60. The die size is 256 mils by 268 mils. The circuit was laid-out to be compatible with just about any 1.2  $\mu\text{m}$  CMOS process<sup>1</sup>. The design rules that were used were a set of 1.5  $\mu\text{m}$  CMOS design rules with 1.2  $\mu\text{m}$  gates.

Let us refer to Figure 8.60 where the key blocks are indicated in the key below the cifplot. Note the three parallel switched-capacitor sections in both the SC lowpass filter and SC programmable equalizer. The rectangular blocks in both these sections are the amplifiers. Large supply lines can be seen running on the top and bottom of the amplifier sections. These are sized to minimize IR drops in the local supply lines. The analog-to-digital converter is comprised of 3 large blocks, each of which contains four sub-converters which can be seen in the figure. The ring oscillator is placed next to the SC lowpass filter to minimize the parasitic capacitance on the high sampling rate clock lines.

---

1. This flexibility of design rules is important in the University environment.



Things to note:

1. 3 blocks in the A/D Converter.  
Within each, four smaller blocks can be seen.
2. Both the SC LPF and SC Programmable EQ are made up of 3 parallel sections.
3. Proximity of Ring Oscillator to SC LPF.
4. Proximity of Clock Generator 2 which generates the non-overlap clocks for both SC filters to the filters.

Figure 8.60 Plot of the chip with key blocks indicated in the key below.

### 8.8.2 Power Consumption by Block

It is useful to consider the power consumption block-by-block. This is shown below in Table 8-1.

**Table 8.2 Power Consumed by Each Block**

Circuit Block (5V 80 degrees C)	Power
Continuous-time LPF plus bias	100 mW
Switched-Capacitor Filters	170 mW
Analog-to-Digital Converter with Bias Circuits	250 mW
Ring Oscillator Circuitry	140 mW
Digital	240 mW
Output drivers (not counted in Total)	87 mW
Total	900 mW

The power consumption is somewhat evenly distributed amongst the different blocks. It is possible that the power in both the ring oscillator and the continuous-time lowpass filter could be reduced<sup>1</sup> in a more aggressive design.

1. Both of these block were conservatively designed in order to ensure 100 MHz operation at 6-bits.

## 8.9 Summary

In this chapter, the key circuit design considerations and techniques used in the experimental prototype were presented. We began with the continuous-time lowpass filter and followed the signal path through the switched-capacitor decimation filter, programmable equalizer, and A/D converter. The circuits used to generate the sampling phases for the decimation filter using a ring oscillator and the approach used to generate the non-overlapping clocks required by the analog signal path we discussed. The generation of the clocks and sampling phases was perhaps the most challenging part of the design because of the number of issues needing simultaneous attention.

In the lists below are the key circuit blocks included in the prototype, the key performance targets, and the key circuit contributions that came about during the design.

The key *circuit blocks* include:

- Cascade of Sallen-Key sections providing attenuation near 250 MHz and above
- 3-Tap 3:1 Decimation Lowpass Filter (300 MHz sampling rate)
- 3-Tap Programmable Symbol-Rate Equalizer (100 MHz output sampling/output rate)
- 6-Bit Analog-to-Digital Converter
- On-Chip Ring Oscillator with On-Chip PLL slaved to external 50 MHz input clock
- Clock Generation.

The key *performance targets* that were met include:

- 50 MHz filter bandwidth
- 100MHz output rate
- 6-Bits of linearity
- Power consumption less than 1 Watt.

The key *circuit contributions* include:

- Architectures for high speed switched-capacitor transversal filters
- Highly parallel implementation of a two-step pipeline converter
- Ring oscillator-based multiphase sampling clock generation
- Circuit block for generation of non-overlapping clocks.

Results taken from prototype devices in the laboratory were presented. The effects of non-idealities such as gain mismatch and channel-to-channel offset voltage mismatches in the parallel channels were present as expected. The chip performed to the design specifications of 6-bits of resolution at a 100 MHz output rate dissipating roughly 900 mW from a single 5V power supply.

## References

- [1] P. R. Gray and R. G. Meyer, *Analysis and Design of Analog Integrated Circuits, Second Edition*, John Wiley & Sons, 1984.
- [2] D. Senderowicz, S. F. Dreyer, J. H. Huggins, C. F. Rahim, and C. A. Laber, "A Family of Differential NMOS Analog Circuits for a PCM Codec Filter Chip," *IEEE Journal of Solid-State Circuits*, Vol. SC-17, pp. 1014-1023, December 1982.
- [3] F. M. Gardner, "Charge-Pump Phase-Lock Loops," *IEEE Transactions on Communications*, Vol. COM-28, No. 11, pp. 1849-1858, November 1980.
- [4] T. Sakurai and K. Tamaru, "Simple Formulas for Two- and Three-Dimensional Capacitances," *IEEE Transactions on Electron Devices*, Vol. ED-30, no. 2, pp. 183-185, February 1983.
- [5] M. I. Elmasry, "Capacitance Calculations in MOSFET VLSI," *IEEE Electron Device Letters*, Vol. EDL-3, No. 1, pp. 6-7, January 1982.
- [6] B. Kim, "High Speed Clock Recovery in VLSI Using Hybrid Analog/Digital Techniques," *Ph.D. Thesis*, University of California at Berkeley, ERL Memorandum No. UCB/ERL M90/50, June 1990.
- [7] B. Kim, D. N. Helman, and P. R. Gray, "A 30 MHz Hybrid Analog/Digital Clock Recovery Circuit in 2- $\mu\text{m}$  CMOS," *IEEE Journal of Solid-State Circuits*, Vol 25, No. 6, pp. 1385-1394, December 1990.
- [8] K. M. Ware, H. S. Lee, and C. G. Sodini, "A 200 MHz CMOS Phase-Locked Loop with Dual Phase Detectors," *IEEE Journal of Solid-State Circuits*, Vol 25, No. 6, pp. 1560-1568, December 1990.
- [9] Monte Mar, private communication.
- [10] T. Hu and P. Gray, "A Monolithic 480 Mb/s Parallel AGC/Decision/Clock-Recovery Circuit in 1.2  $\mu\text{m}$  CMOS," *ISSCC 1993 Digest of Technical Papers*, vol. 36, pp. 98-99, Feb. 1993.

## CHAPTER 9

# Summary and Conclusions

---

### 9.0 Introduction

In this chapter, we shall summarize the key contributions of this research and draw some conclusions and perspectives. We will begin with a summary of two general themes present throughout this research. They are the use of parallel signal processing to increase throughput and the examination of integrated circuit implementation of communication receivers from a system level view point. The latter is important because judicious partitioning of the signal processing between the analog and digital domains can have an important impact on both the performance and implementation of the overall system. Following this will be a summary of the key aspects of the experimental prototype. And finally, we will conclude with areas of future work and improvement.

Many disk drive designers and manufacturers believe that storage densities and transfer rates will continue to increase with the refinement of MR head technology and the increased application of Partial Response signalling to the read channel. As this continues, performance limitations may be economical implementation of the read channel electronics at a cost and power that keeps linear scaling competitive. At some point, one might imagine that there will be limitations that can only

be overcome with RAID<sup>1</sup>-like approaches which use a parallel disk arrays to achieve high storage densities and high transfer rates [1]. However, further development of CMOS and BiCMOS technologies together with the trend toward close alliances between disk drive manufacturers and IC design houses should result in continued improvements of read channel architectures and ICs. The collaboration that these alliances bring about between the circuit designer and disk drive channel designer enable the channels to be viewed in different and new perspectives. It appears that there will be an increasing volume of drives employing PRML signalling techniques. At the time of this writing, there are not yet commercially available PR-IV chips (although they have been reported [2] and many companies are working on them), there are companies that are beginning to consider chip development for higher order partial response polynomials.

The key point is that it appears certain that there will be the need for high speed discrete-time processing in the disk drive channel and the research prototype described in this thesis has targeted this exciting and commercially important application.

## **9.1 Key Research Themes**

There has been primarily two underlying themes throughout this research. First, has been to examine the application of parallel signal processing in both the analog and digital domains in order to achieve a higher throughput than otherwise possible in a given integrated circuit technology. This theme is present in the SC transversal architecture presented in Chapter 5 which was used in both the lowpass decimation filter and the programmable equalizer in the prototype. Parallel processing was also prevalent in the A/D converter, which achieved 6-bits of resolution at a 100 MHz output rate. Parallel processing was also employed in the digital domain in an adaptive filter project which was briefly described in Chapter 5 but which was not expanded in this thesis but will be reported elsewhere [3,4]. It was found that there is in fact definite advantages to attaining higher speed performance than otherwise possible as shown by the 100 MHz analog-to-digital interface experimental prototype which was implemented in 1.2 $\mu$ m CMOS, operated off of a 5 Volt supply, and consumed less than 900 mW.

---

1. RAID is an acronym for Redundant Array of Inexpensive Disks.

As silicon technology continues to scale, the trend particularly in CMOS is to do all the signal processing in the digital domain. At clock rates on the order of 100 MHz, implementation difficulty of the A/D converter is a non-linearly increasing function of the required resolution, i.e., a 5-bit converter is simpler than a 6-bit, whereas a 6 bit converter is much simpler than a 7-bit. Bringing this dimension into the architecture design and block specifications has been the second theme of this research. We have considered the cost-benefit trade-off of performing signal processing in the analog domain in a magnetic disk read channel implementation with large amounts of digital signal processing and found it advantageous to perform analog signal processing before the ADC. This theme is certainly not new and has been used in many architectures for high speed transmission over twisted pair [5]. However, this investigation in the PR-IV channel has not yet been reported.

## 9.2 Research Results

In this research, we have demonstrated the potential for high speed analog-to-digital interfaces using CMOS technology at frequencies in the 100 MHz range using parallel architectures. The prototype was designed in 1.2  $\mu\text{m}$  CMOS and operated off of a single 5 V supply. The key research results are as follows:

- Demonstrated that passive FIR filtering can be used to achieve high-sampling rate decimation filters for use in high speed lowpass filtering applications and single-rate filters for high speed analog equalizers.
- Demonstrated that parallel architectures can be utilized in the analog domain in order to achieve operating frequencies that are higher than otherwise possible using conventional approaches.
- Showed through simulation that there may be advantages to performing signal pre-conditioning (pre-equalization) in the analog domain before the A/D converter even in DSP-based systems implementing adaptive equalizers in the digital domain.
- Showed through analysis and confirmed through experiment that the limitations of the new parallel filter architecture are due primarily to gain and offset mismatch in the parallel channels.

### 9.3 Future Work and Improvements

There are many avenues for future work and improvements. Let us begin with the prototype. The DSP described in Chapter 5 is expected to run at an effective clock rate of 100 MHz off of a 3.3V supply. It contains an 8-tap adaptive equalizer and a Viterbi sequence detector and is expected to consume around 250 mW. The power consumption of the prototype A/D interface on the other hand is near 900 mW for 100 MHz operation. It must be stated that a rather conservative design approach was taken with each of the blocks to assure operation at 100 MHz. However, circuit techniques to reduce power consumption is a key area in which improvements can be made. The use of a more aggressive technology would certainly help in this area. The power efficiency of the ring oscillator-based clock generation needs to be improved. The amplifiers in the lowpass filter and programmable equalizer could be scaled down in a less conservative design requiring less power. And other A/D converter architectures requiring less power are certainly possible. However, with regard to the A/D converter, higher level considerations such as the settling time of the sample-and-hold amplifiers driving the A/D input needs to be factored into the architecture selection process.

Making the equalizer adaptive for the purpose of performing analog timing recovery is an area of future work. Achieving this would allow much more latency after the equalizer which means that more pipelining and parallelism could then be used to lower power consumption.

One of the key advantages of discrete-time switched-capacitor filters is their manufacturability as compared with their continuous-time counterparts. Since the filter response of a switched-capacitor filter is defined by capacitor ratios, it seems that it would be inherently more robust than a continuous-time filter to process, temperature, and supply variations. The ring oscillator is the most *fragile* block on the prototype. It would be interesting to study how these approaches perform in a production environment.

It is not obvious how bipolar devices which are standard devices in BiCMOS technologies would best be utilized. However, it may be possible that they may be used to lower overall power consumption, particularly as buffers and in the ADC.

---

The prototype achieved 6-bits of linearity and dynamic range with 18 output buffers changing state every 30 nsec with an output rate of 100 MHz (3 stages x 6 bits/stage output at one-third the output rate). This results in large current transients which apparently did not adversely affect performance of the chip. This suggests that the design techniques used in the prototype are robust and can be used in mixed-signal environments. A possible future project may be implementation of a single chip including both the DSP and the A/D front-end on a single substrate.

## References

- [1] R. H. Katz, G. A. Gibson, and D. A. Patterson, "Disk System Architectures for High Performance Computing," *Proceedings of the IEEE*, Vol. 77, No. 2, pp. 1842-1858, December 1989.
- [2] F. Goodenough, "DSP Technique Nearly Doubles Disk Capacity", *Electronic Design*, pp. 53-58, February 4, 1993.
- [3] C. S. H. Wong, *Masters Thesis* to be published in fall of 1993, UC Berkeley.
- [4] J. C. Rudell, *Masters Thesis* to be published in fall of 1993, UC Berkeley.
- [5] O. Agazzi and A. A. Adan, "An Analog Front End for Full-Duplex Digital Transceivers Working on Twisted Pairs", *IEEE Journal of Solid-State Circuits*, Vol. 24, No. 2, pp. 229-240, April 1989.