

Copyright © 1987, 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.

**MULTI-CHANNEL PCM A/D INTERFACES
USING OVERSAMPLING TECHNIQUES**

by

Bosco Hok-Chung Leung

Memorandum No. UCB/ERL M87/88

17 November 1987

COVER PAGE

**MULTI-CHANNEL PCM A/D INTERFACES
USING OVERSAMPLING TECHNIQUES**

by

Bosco Hok-Chung Leung

Copyright © 1987

Memorandum No. UCB/ERL M87/88

17 November 1987

ELECTRONICS RESEARCH LABORATORY

College of Engineering
University of California, Berkeley
94720

TITLE PAGE

**MULTI-CHANNEL PCM A/D INTERFACES
USING OVERSAMPLING TECHNIQUES**

by

Bosco Hok-Chung Leung

Copyright © 1987

Memorandum No. UCB/ERL M87/88

17 November 1987

ELECTRONICS RESEARCH LABORATORY

College of Engineering
University of California, Berkeley
94720

Multi-channel PCM A/D Interfaces Using Oversampling Techniques

Ph.D

Bosco Leung

EECS

Abstract

An important objective in telephony systems is the reduction of silicon area per line on central office line cards serving analog subscriber loops. Multiplexing of hardware is an attractive approach to this problem, but current practice is to use per line coders and filters because of crosstalk issues and other problems. However, with the use of digital filtering instead of analog, the crosstalk problem between signal paths multiplexing the same hardware becomes much less severe. Such multiplexing can result in dramatic reductions in silicon area in such systems, as well as in a wide variety of other types of applications.

The purpose of this research project is to explore the architectural and circuit design issues associated with putting multi-channel voice coders with multiplexed digital filters on a single chip. Such a coder will consist of multiple sigma-delta front ends followed by a time-multiplexed digital decimation and anti-aliasing filter. The oversampling scheme dramatically relaxes the analog components' requirements.

A chip with 4 channels on has been designed and fabricated using a 3 micron double metal single poly CMOS process. It has been tested up to full speed and has a dynamic range of 79db on a 5V supply, consumes 50 mW per channel, has an idle channel noise of 11.6dbmC0, crosstalk of -83db and a frequency response that meets the D3 specifications in a silicon area of 8250 sq. mils per channel.


Committee Chairman

Acknowledgement

I would like to express my thanks towards Professor Gray for his guidance and support throughout my Ph.D. Study. I would also like to extend my gratitude towards Professor Brodersen for his special role in advising me on this project. Their technical expertise is a constant source of enlightenment for me.

I would like to thank fellow colleagues Yuh-Min Lin and Robert Neff for their contributions to making the chip work. Many thanks to Steve Lewis, Sehat Sutarja, Joey Doernberg, Hyun Shin and all the other I.C. group students for their technical discussions. Also Steve Lewis has been most helpful throughout my prelim studies.

Finally, I would like to thank my wife and our parents for their encouragement as well as the sacrifice put up by them through all these years.

This research was supported by DARPA, the California MICRO program, as well as Bellcore, Fairchild, AMD, Rockwell International and Honeywell. Special thanks go to Bell Northern Research for supporting me under their sponsorship program.

Table of Contents

CHAPTER1: INTRODUCTION	4
1.1 Goals	3
1.2 Outline	3
CHAPTER 2: INTRODUCTION TO OVERSAMPLED A/D CONVERTERS	5
2.1 Introduction	5
2.2 Alternative oversampling schemes	5
2.3 Predictive Coding	9
2.4 Interpolative Coder	11
2.5 Sigma-delta coders	11
2.6 Modeling of the sigma-delta coder	13
CHAPTER 3 ARCHITECTURAL CONSIDERATIONS FOR A MINIMUM COST CODER	17
3.1 Introduction	17
3.2 Discussions on the selection of the structure of the oversampled coder	18
3.3 First order versus second order loop	19
3.4 Requirements on the decimation filter	19
3.5 Basic filter architecture	21
3.6 Optimum filter for multi-channel sigma-delta coders	23
3.7 Significance of the L/D ratio	28
3.8 Other considerations	30
3.9 Conclusions	30
CHAPTER 4 DESIGN TECHNIQUES FOR A MULTI-CHANNEL PCM CODEC	32

4.1 Digital decimation filter	32
4.1.1 The three filters architecture	32
4.1.2 ROM-based FIR filter	34
4.1.3 Custom FIR filter by Candy	39
4.1.4 Prototype Custom FIR filter for a triangle window	42
4.1.5 Circuit designs for the FIR filter	43
4.2 Second stage filter design issues	49
4.2.1 Bit serial approach	50
4.2.2 Microprogrammed approach	53
4.2.3 Lager system	56
4.2.5 Modifications to the address arithmetic unit	57
4.3 Analog front end design	58
4.3.1 Alternative analog architectures	58
4.3.2 Overall analog circuits requirements and solutions	60
4.3.3 Requirements on the operational amplifier	61
4.3.4 Prototype operational amplifier	65
4.3.5 Requirements on the comparator	72
4.3.6 Prototype comparator	73
CHAPTER 5 EXPERIMENTAL RESULTS	76
5.1 Introduction	76
5.2 Test setup	76
5.3 Experimental results	77
5.3.1 Experience from chip1	77
5.3.2 Experience from chip2	78

5.3.3 Experience from chip3	78
5.3.4 Test results and methods	79
CHAPTER 6 CONCLUSIONS	86
6.1 Introduction	86
6.2 Specific Functions of the coder	86
6.3 Comparisons between different architectures	86
6.3.1 Conventional approach	86
6.3.2 Single channel sigma-delta coder	88
6.3.3 Multi-channel sigma-delta coder	90
6.4 Major research results	93
6.5 Future directions	94
APPENDIX: CAD TOOLS	95

CHAPTER 1

INTRODUCTION

As the cost of VLSI circuits decreases there is a greater demand on the analog-digital interfaces between these VLSI circuits and the analog world. The importance of developing low cost analog-digital interfaces in applications ranging from instrumentation, speech recognition to voice coding cannot be overemphasized. Due to the large number of analog-digital interfaces in modern telecommunications systems, the development of low-cost monolithic MOS analog-digital conversion techniques has been an important element in reducing the cost of digital transmission and switching systems, particularly in telephony where a very large number of interfaces are required. One of the most important analog-digital interfaces from a commercial viewpoint is that of the subscriber loop in digital switching systems. Presently, most telephone digital switching systems are implemented with per-line codec/filter chips on the line card, the chip containing the A/D, D/A, anti-alias filtering, line driver, and other digital control functions. Most such codec chips utilize companding charge-redistribution coders with switched capacitor filters, but as MOS technology feature sizes continue to fall oversampled coders with digital filters are becoming more attractive. This is particularly true when more than one oversampled coder are put on the same chip with appropriate multiplexing of the decimation filter. Such a mult-channel codec chip will demonstrate substantial savings in terms of area, especially as the technology scales down.

To better understand the significance of the per line codec, a system overview is necessary. Fig 1.1 shows a simplified version of an analog telephone connection, in which switching is done locally, with the signals in a digital PCM format. The digital switch must inherently separate the two directions of transmission; that is it is four wire. Thus the hybrid moves to the subscriber side of the local switch, and must be implemented on a per subscriber line basis. A simplified block diagram of the interface between a local digital switch and a subscriber line is shown in fig. 1.2 The line interface performs the hybrid function, provides dc power to the phone, and other functions. The transmit low-pass filter prevents aliasing distortion resulting from any frequency components above 4 kHz, half the subsequent sampling rate of 8 kHz. The A/D converter(ADC) then converts individual samples to the digital format used internally in the

switch. On the receive side, the D/A converter(DAC) generates analog samples and the low-pass filter reconstructs the analog speech waveform. The ADC and DAC together are called a coder-decoder(codec). In the alternative implementation of metallic crosspoint analog switch, most of the functions in fig. 1.2 are not required. Hence these functions must be implemented very inexpensively to result in an economically competitive switch. Fortunately, there are offsetting savings in the local digital switch. The switching function itself is more economically realized, and the interface to digital PCM transmission trunks and subscriber line multiplex systems are much cheaper. As a result, the cost of the entire local digital switch is dominated by the cost of the functions on fig. 1.2 together with the related functions of battery feed, overload protection, and ringing and test access. Because the functions on fig. 1.2 must be realized inexpensively, and because they are replicated on every subscriber line and therefore will be produced in high volume, they are a natural application for custom integrated circuits where an area efficient solution can be attained.

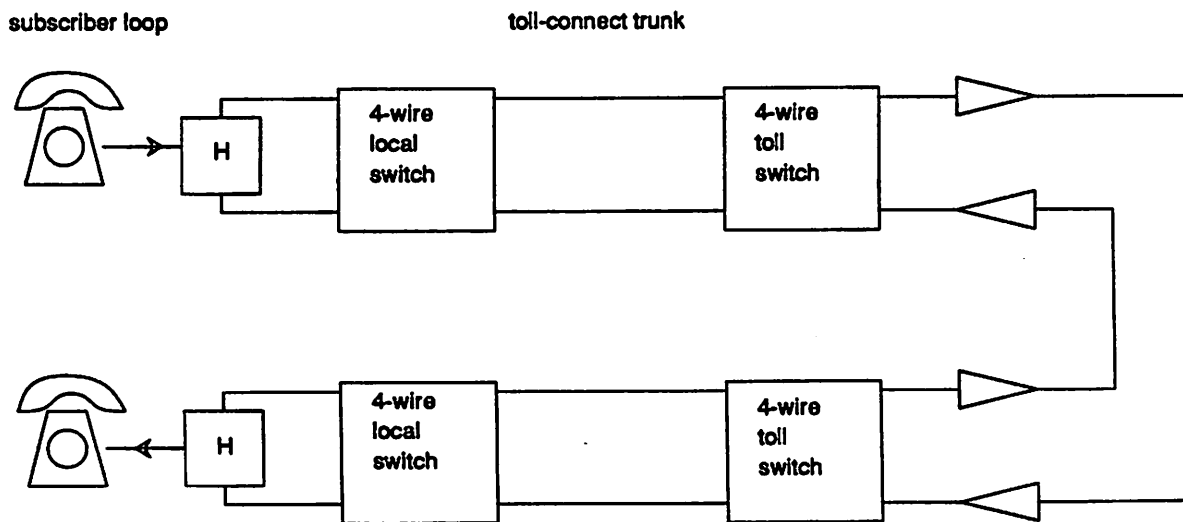


Fig. 1.1 Telephone system

Fig. 1.1

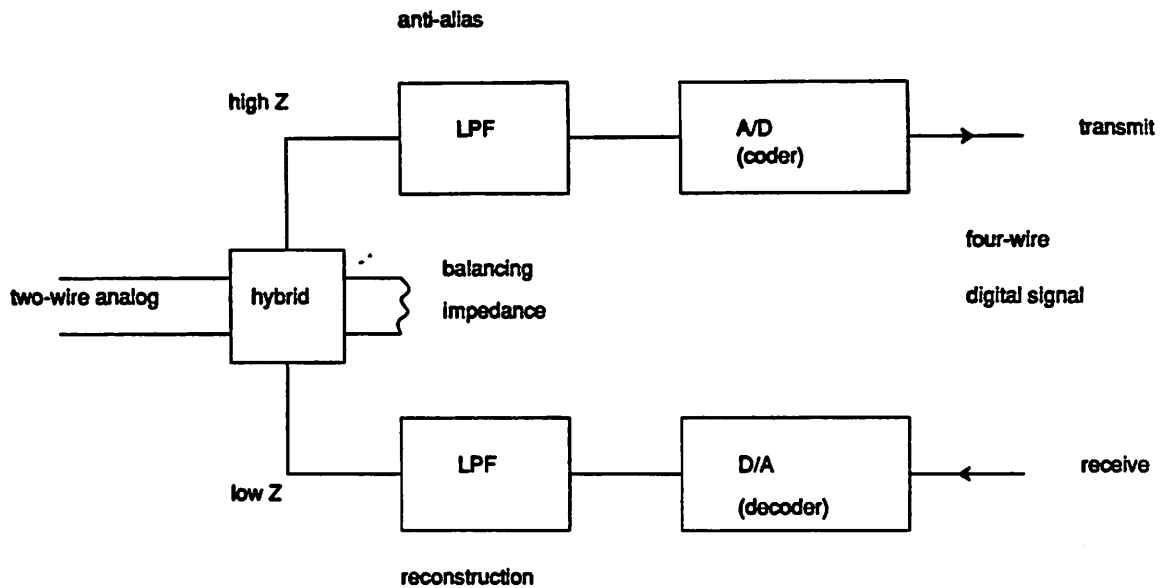


Fig. 1.2 Subscriber line interface

Fig. 1.2

1.1 Goals

The purpose of this research is to explore the use of oversampled techniques in developing low cost integrated circuits to fulfil the A/D functions in the subscriber loop. An area efficient solution using a multi-channel coder on the same chip is presented. The prototype employs an oversampled multi-channel coder which incorporates four A/D converters with transmit low pass filters on one single chip. The goal of this research is to make the total chip area to be smaller than that offered by the conventional approach in scaled technologies. At the same time the channel to channel crosstalk has to be kept below an acceptable level.

1.2 Outline

In chapter 2, an overview of alternative oversampled structures is discussed. This includes the predictive coding and the noise shaping coding.

Chapter 3 deals with the architectural issues which affect the design of a minimum cost oversampled coder in the context of present day CMOS technology. Comparative studies are made of the two particular

structures of oversampled modulators, the first order and the second order coder in terms of their area efficiency.

Chapter 4 discusses the design and implementation of a prototype four channel coder.

In chapter 5 experimental results are given on the prototype.

Finally in chapter 6 conclusions are drawn about this project and gives possible future directions of applications using oversampled coders.

CHAPTER 2

INTRODUCTION TO OVERSAMPLED A/D CONVERTERS

2.1 Introduction

In this chapter the concept of oversampling A/D converters is explained. The basic principles of the predictive coding and the noise shaping coding will be looked at. Then the interpolative coder and the sigma-delta coder, which use the above coding principles, will be discussed. This will enable us to investigate the advantages and the disadvantages of the various schemes, especially when implemented in the VLSI technology and allow us to make judicious choices.

2.2 Alternative oversampling schemes

Under most practical situations the quantization noise introduced in the A/D conversion process is approximately white and is uniformly spread from dc to half the sampling rate, assuming the signal amplitude is large compared to the step size. Since the total energy contained in the quantization noise is a constant determined by the step size, that portion of the noise lying in a given spectrum can be reduced by increasing the sample rate, at the rate of 3dB per octave of sample rate increase as shown in fig.2.1. Here the area is the total quantization noise power. As the sampling frequency is doubled the density is reduced by two. If the band of interest is still the same that means only half of the original noise power is inband and so the quantization noise is reduced by 3dB

Looking at it in another way, if the number of samples taken per unit time is doubled and each pair of samples produced is averaged to produce new samples at the original sample rate, the signal component of the samples will add linearly while the quantization noise component in the samples will add as uncorrelated random variables. Thus the signal-to-noise ratio will improve by 3db.

However there are two drawbacks to this oversampling scheme. The first is that the rate of improvement is relatively slow; In order to get the equivalent of 1 additional bit of resolution the sample rate must be increased by a factor of 4. A second reason is that the underlying assumption that the signal amplitude

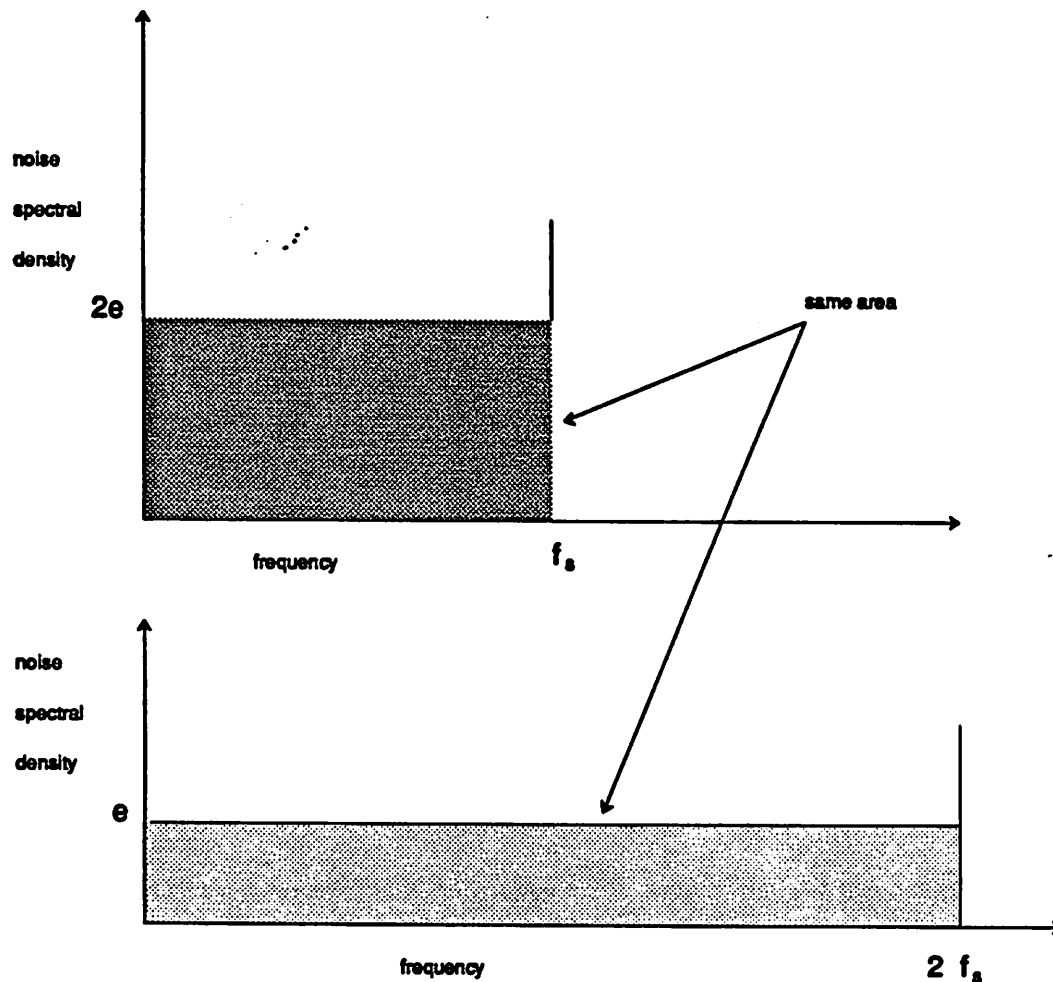


Fig. 2.1 PCM oversampling

is much larger than the step size will be violated for small signals if this process is carried very far. Both of these drawbacks can be solved by incorporating the quantizer with a low pass filter in a feedback loop. The location of the low pass filter defines two classes of coding schemes: the noise-shaping coding and the predictive coding. Whereas in a coder using noise-shaping coding the low pass filter is in the forward path, in a coder using predictive coding the low pass filter is in the feedback path. This difference determines the principle of operation of the two oversampling schemes. Noise-shaping coding works by moving the quantization noise out of band and employing a digital filter to remove it. An example coder using noise-shaping coding is shown in fig. 2.2 where the low pass filter is realized by an integrator.

The loop presents different paths to the signal x and the quantization noise e generated from the N-

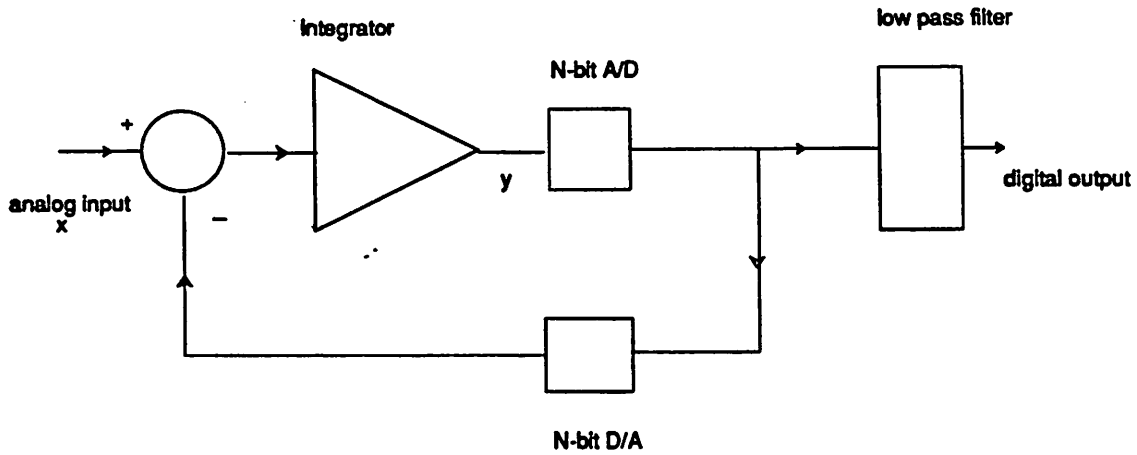


Fig. 2.2 noise-shaping coding

bit A/D. The signal is not shaped by the overall loop but the noise is shaped by the loop to have a high pass nature as shown in fig. 2.3

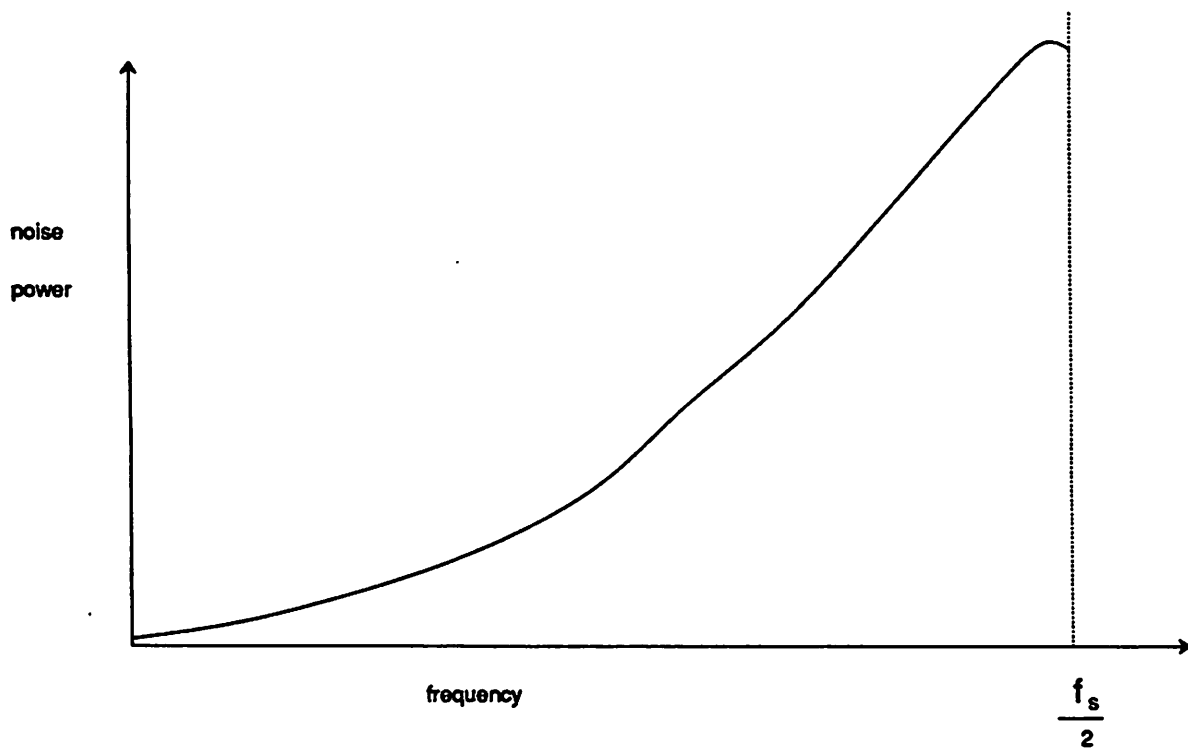


Fig. 2.3 Quantization noise power spectrum

Predictive coding, on the other hand, works on reducing the size of the quantization error. Fig. 2.4

shows an example coder using predictive coding where the low pass filter is an integrator. Here the feedback loop works by making the integrator output y tracks the signal input x , thus the error e is reduced.

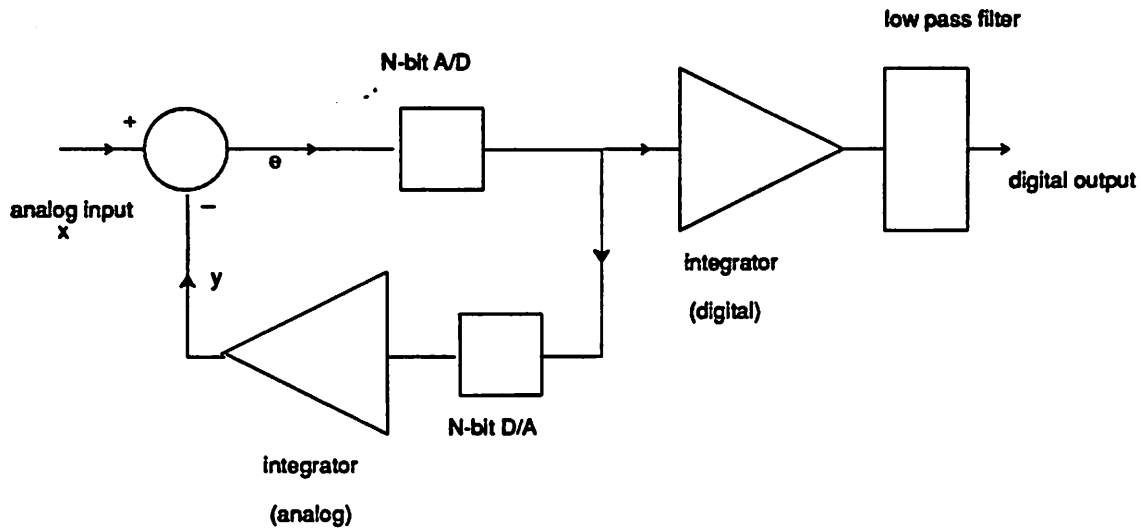


Fig. 2.4 Predictive coding

A successful design of the coder depends on the choice of the proper structure, the order of the coder, the amount of oversampling required and the proper design of the subsequent decimation filter for a particular application. Furthermore the noise-shaping coding and the predictive coding can be combined in one coder to achieve the necessary performance.

Before a comparison is made among various oversampling schemes, let us try to define some means of measuring their performance. The most important performance benchmark is a plot of the signal-to-noise ratio (SNR) versus signal amplitudes for various input frequencies of a sine wave. Signal to noise ratio is defined as the signal power to noise power. Let us take a look at a typical signal-to-noise ratio curve as shown in Fig. 2.5.

Normally the curve is a straight line, indicating that the quantization noise is relatively constant with respect to the signal amplitude. After the signal amplitude reaches a certain point the SNR drops, usually due to overloading. The predictive coding has a SNR curve that is dependent on the input frequency due to slope overloading. The higher the frequency, the lower the SNR curve. The overload point is defined as

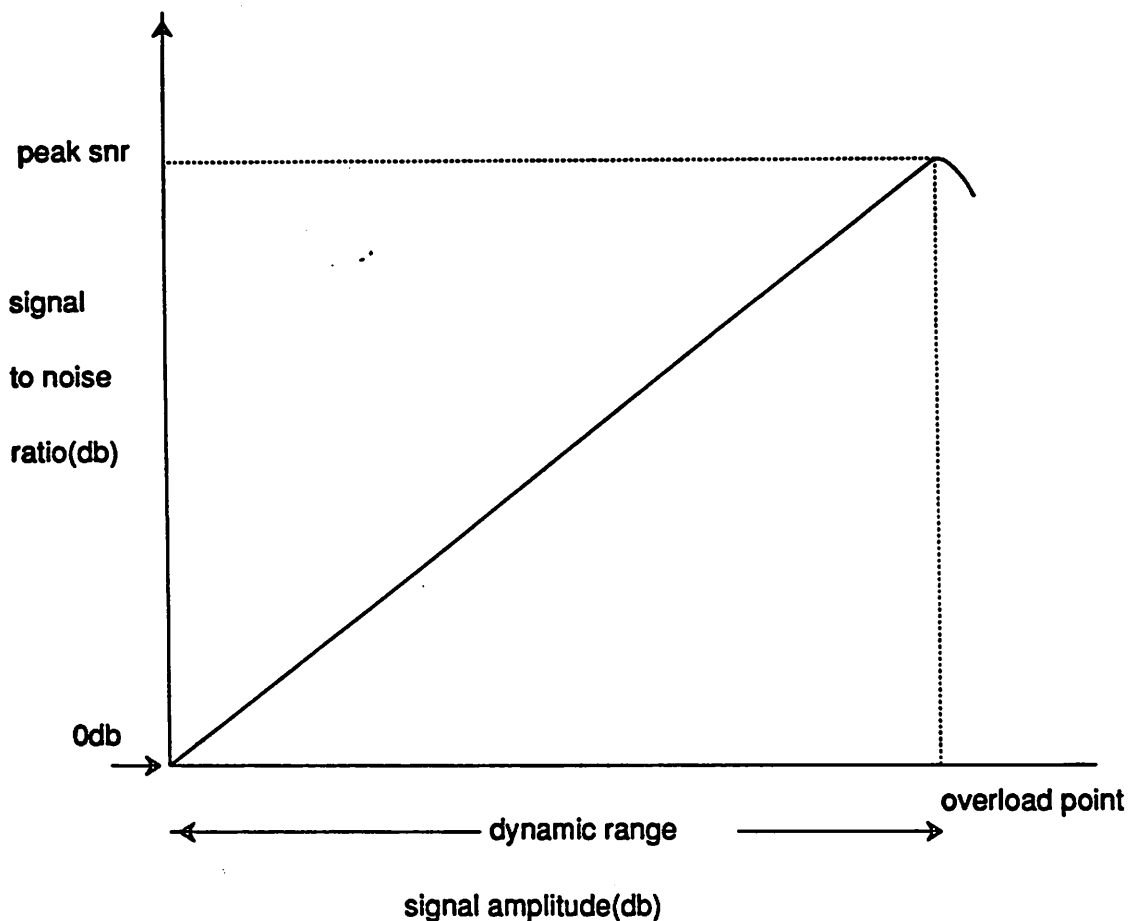


Fig. 2.5 Typical signal to noise ratio

the input amplitude when the SNR starts to drop. This is due to clipping in a conventional PCM coder as well as in a first order sigma-delta coder and occurs when the input signal amplitude is larger than the reference voltage. For higher order sigma-delta coders this point occurs below the voltage reference. The point at which it occurs depends on the exact noise-shaping characteristics as well as the number of bits employed in the A/D, D/A in the feedback loop. Dynamic range is defined as the range of signal amplitude from the point the coder overloads to the point when the SNR is 0db. Peak SNR, however, is defined as the maximum signal-to-noise ratio that one can get out of the coder and is usually the SNR at the point when the coder goes into overload. Depending on the slope of the SNR curve and the nature of the coder, the peak SNR of a coder may or may not be the same as the dynamic range.

2.3 Predictive Coding

A coder uses predictive coding by taking the quantized output, passes it through a digital to analog converter and then a low pass filter, and feedbacks the resulting signal. Note that the low pass filter is in the feedback path. The order of the low pass filter determines the order of the predictive coder. Fig. 2.7 shows the general form of predictive coding.

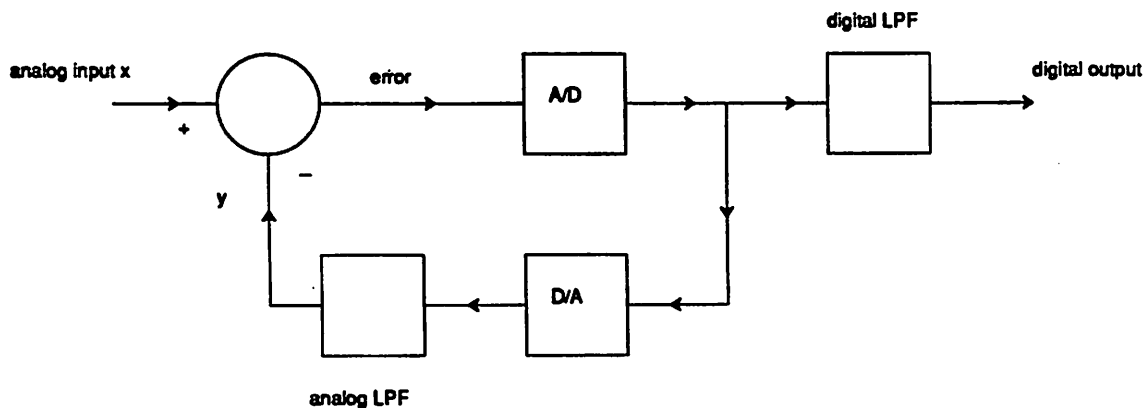


Fig. 2.6 Predictive coding

In the simplest case of the first order predictive coding the low pass filter is just an integrator. If the coder uses 1-bit A/D and D/A then the structure reduces to that of a delta modulator. In general for higher order coders, poles of the low pass filter are chosen to reduce the prediction error σ_e and therefore to reduce the input full scale range of the A/D converter. Different methods have been proposed to select these poles to minimise the errors.¹ Following the analysis given in¹ σ_e^2 decreases by 6.N db per octave increase in the sample rate where N is the order of the low pass filter. Therefore N bits are eliminated for each doubling of the sample rate. Furthermore there is the decrease in the inband coding noise power and it can be shown that the inband noise decreases by 3db per octave. Therefore the level separation can be increased by a factor of 1.4 for each doubling of the sample rate. Therefore N+1/2 bits are eliminated by an Nth-order predictive coding with out-of-band noise filtering for each doubling of the sampling rate. This improvement is similar to that achieved by a noise-shaping coding.

Implementation issues

Even though the theoretical signal-to-noise ratio obtained using predictive coding is comparable to

that from noise-shaping coding a number of factors make the noise-shaping coding more desirable, especially in the integrated circuit environment. First, a coder using predictive coding is subjected to the slope overload problem. To understand the slope overload problem let us take a look at the first order predictive coder in fig. 2.4. When the input signal x changes, the feedback signal y tries to predict what x should be and adjusts itself to track it. The signal y can only change by a fixed step in a given clock period. If the signal x changes by more than the step size so that the difference between x and y , or the error signal e is larger than the dynamic range of the N -bit A/D, then the coder will fail. This is called the slope overload problem. Again using the first order predictive coder as an example, there is the extra integrator(digital) in the coder using predictive coding and any mismatch between that and the analog integrator in feedback path may create problems(e.g. if the gain element is different). In terms of just hardware the output of the digital integrator is multi-bit which has to be processed by the subsequent decimation filter. Hence the decimation filter is harder to design than that of the noise-shaping coding which needs to process just a one-bit code.

2.4 Interpolative Coder

There is a class of coder reported in the literature called the interpolative coder. This coder uses both the predictive coding and the noise-shaping coding. A block diagram of an example of an interpolative coder² is shown fig 2.7.

On each clock cycle the comparator decides if the integrator output is positive or negative, and increments the shift register up or down. During the following clock period, the integrator integrates the difference between the instantaneous input signal and the current DAC output value. At the start of the next clock cycle the shift register is again clocked up or down. The result is a waveform at the DAC output which constantly interpolates between the output levels of the DAC. The average value of this waveform is a representation of the signal. The sample rate for this coder is in the 256kHz range for voiceband signals.

2.5 Sigma-delta Coders³

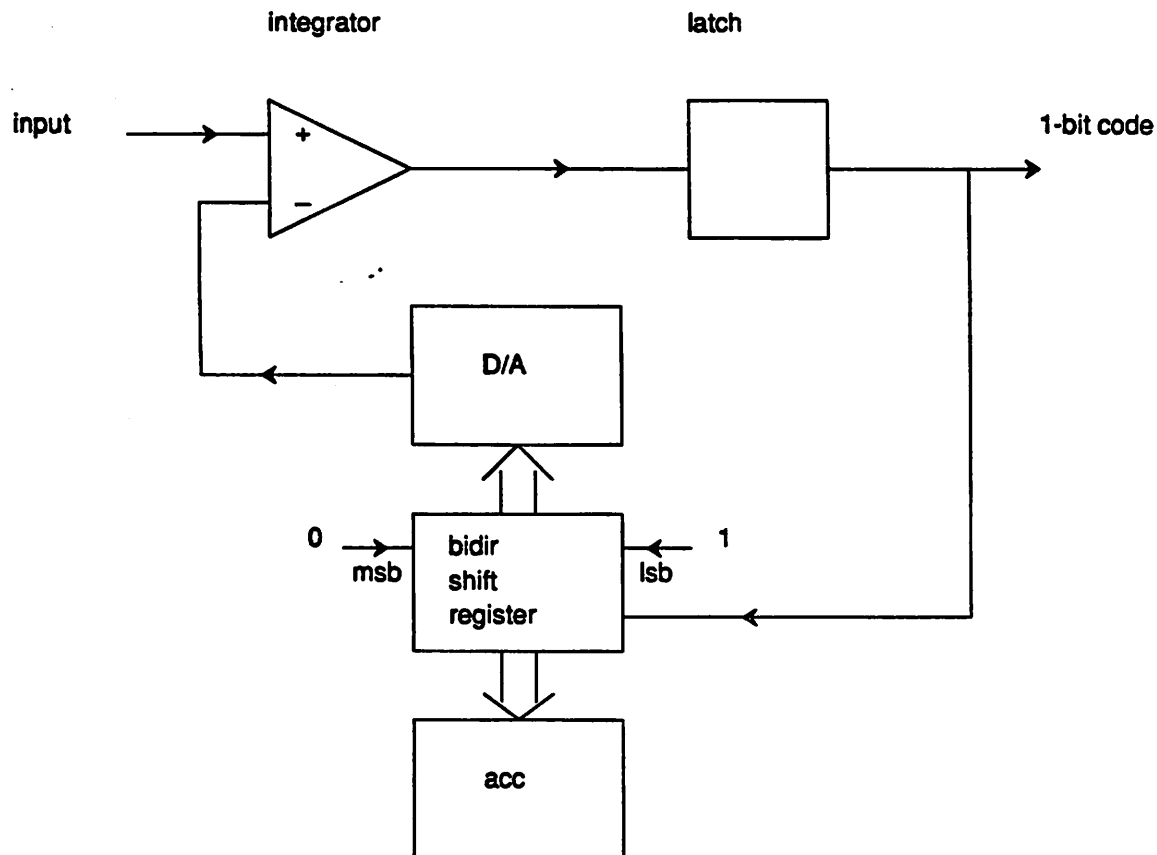


Fig. 2.7 Block diagram of an example interpolative coder

A sigma-delta coder is a coder using only noise-shaping coding and no predictive coding. Also only a 1-bit A/D and a 1-bit D/A are used. The resulting waveform when a sinusoid is encoded is a square wave of amplitude equal to the full scale reference voltage, and as a result the total energy in the waveform is quite large. Subsequent decimation filter has to be able to filter this high frequency noise out.

The sigma-delta coder has advantages in hardware simplicity for the analog front end and high sampling rate, which allows more use of digital anti-alias filtering. Most of the quantization noise can be moved out-of-band by the use of an appropriate filter in the forward path. However the large amplitude of the out-of-band quantization noise requires a more complex anti-alias filter prior to decimation. Another important aspect of sigma-delta coder performance is that for low order (primarily first order) structure, when signal amplitude is low, the DAC output waveform can contain "noise" components which are not random, but which are concentrated at one frequency and result in spurious tones. This can however be randomised by adding a dither signal at the input. ^{4,5,6} The order of a sigma-delta coder is determined by

the order of the low pass filter in the forward path.

Implementation issues

The principal practical advantage of sigma-delta coders is the fact that their sampling rate is high and more digital anti-alias and reconstruction filtering can be used. Since digital filters can directly take advantage of scaled technologies, oversampled coders with digital filters is playing a more important role in telecommunications analog-digital interfaces. Even though the order of the decimation filter is higher due to more out of band noise, the fact that a 1-bit code is present eliminates the use of multipliers and simplify the filter. The use of switched capacitor integrators for both the first and the second order loop have been reported ^{4,7,8} As digital technology continues to scale this trend of trading digital circuit complexities for simpler analog front end is likely to continue.

2.6 Modeling of the sigma-delta coder

The design methodology is based on a discrete-time model of the sigma-delta coder. The basic configuration of a typical sigma-delta coder is shown in Fig.2.8.

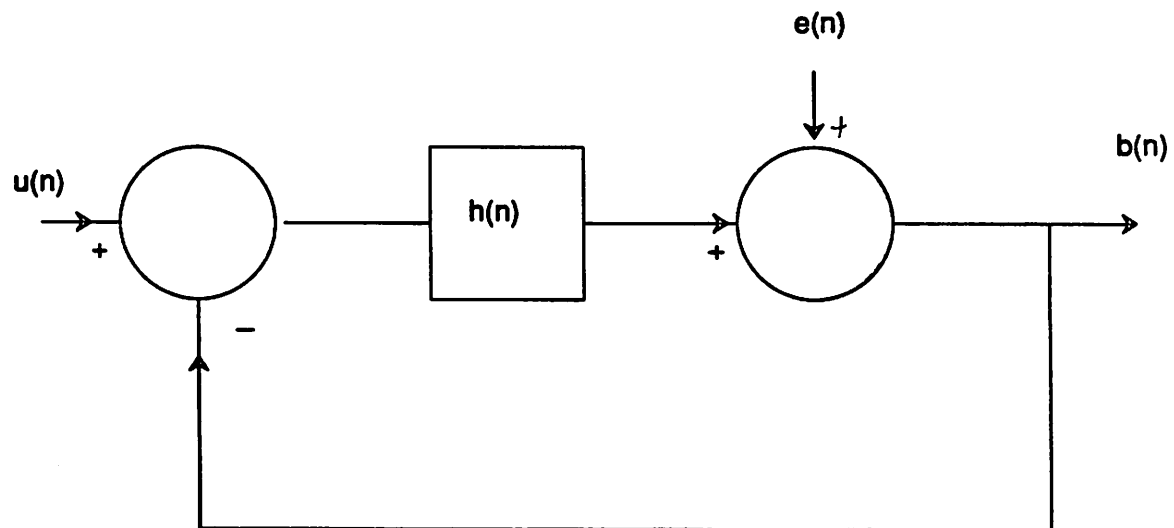


Fig. 2.8 Mathematical model of a sigma-delta coder

It consists of a filter $H(z)$, an A/D converter, and a D/A converter which is embedded in the feedback

loop. The filter can be analog(RC type) $H(s)$ or discrete time(switched-capacitor) $H(z)$. In the following discussion it is assumed to be of the $H(z)$ form. In the sigma-delta coder, the A/D converter is implemented as a comparator followed by a latch and the D/A converter is an array of switches that switch the proper voltage reference as analog output corresponding to the latch's state. The order of the sigma-delta coder is determined by the order of $H(z)$.

From fig. 2.8 two transfer functions can be derived; one between the input $u(n)$ and the output $b(n)$, called the $W(z)$ and the second between the noise $e(n)$ and output, called the noise transfer function $T(z)$.

$$W(z) = \frac{H(z)}{1+H(z)} \tag{2.1}$$

$$T(z) = \frac{1}{1+H(z)} \tag{2.2}$$

Examining the above two functions it is found out that the input signal is confined to a small band around dc. Consequently, the $W(z)$ should be flat at low frequency. The noise transfer function, on the other hand, should be of a high pass nature, effectively attenuating the quantization noise at low frequencies, although at the expense of amplifying the quantization noise at high frequencies. A first-order sigma-delta coder have a noise transfer function given by

$$T(z) = (1-z^{-1}) \tag{2.3}$$

The spectrum $N(n)$ at the output is given by

$$S_{nn}(f) = 4\sigma_e^2 \sin^2\left(\pi \frac{f}{f_s}\right) \tag{2.4}$$

The total quantization noise in the band of interest $[-fc, fc]$ is given by

$$\sigma_n^2 = 2 \frac{\sigma_e^2}{\pi} \left[2\pi \frac{fc}{f_s} \sin\left(2\pi \frac{fc}{f_s}\right) \right] \tag{2.5}$$

Using a two-term expansion for sin (assuming $f_s \gg fc$)

$$\sigma_n^2 = \frac{8}{3} \pi^2 \sigma_e^2 \left(\frac{fc}{f_s}\right)^3 \tag{2.6}$$

Therefore doubling the sampling rate decreases the noise power by 9dB. A second order sigma-delta have a noise-transfer function given by

$$T(z) = [1 - z^{-1}]^2 \quad (2.7)$$

Then

$$S_{nn}(f) = 16\sigma_e^2 [\sin(\pi \frac{f}{f_s})]^4 \quad (2.8)$$

The inband noise power is given by

$$\sigma_n^2 = 24\pi^4 \sigma_e^2 (\frac{f_c}{f_s})^5 \quad (2.9)$$

Note that doubling the sampling rate decreases the inband noise power by 15dB. With this mathematical formulation a design for the multi-channel coder will be presented in the subsequent chapters.

References

1. Tewksbury , "Oversampled, Linear Predictive and Noise-Shaping Coders of Order $N > 1$," *IEEE Transactions on Circuits and Systems*, pp. 436-447, July 1978 .
2. Wooley , "An Integrated Per-channel Encoder Based on Interpolation," *IEEE Journal of Solid-State Circuits*, pp. 14-20, Feb, 1979 .
3. Candy , "Using Triangularly Weighted Interpolation to Get 13-Bit PCM from a Sigma-Delta Modulator," *IEEE Transactions on Communications* , pp. 1259-1275, Nov 1976 .
4. Hauser , "MOS ADC-Filter Combination That Does Not Require Precision Analog Components," *ISSCC'85 Digest of Technical Papers*, pp. 80-81..
5. Everhard , "A Singel Channel PCM Codec," *IEEE Journal of Solid-State Circuits*, pp. 25-37, 1979, sc-14 .
6. Misawa , "Single-Chip per Channel Codec with Filters Utilizing Delta-sigma Modulation," *IEEE*

Journal of Solid-State Circuits, pp. 333-341, vol. SC-16, 1983 .

7. Defraeye , "A 3-um CMOS Digital Codec with Programmable Echo Cancellation and Gain Setting," *IEEE Journal of Solid-State Circuits*, pp. 679-687, June 1985 .
8. Koch , "A 120kHz Sigma/Delta A/D Converter," *ISSCC'86 Digest of Technical Papers*, pp. 138-139.

CHAPTER 3

Architectures For A Minimum Cost Multi-Channel CMOS Coder

3.1 Introduction

From chapter 2 the basic principles underlying the various oversampled coders have been established. In this chapter a comparison will be made between different oversampled coders. The goal is achieve minimum cost which in terms of the integrated circuit technology means the minimum chip area.

A coder is consisted of both the ADC and the anti-alias filter in front of it. Therefore the area comparisons will include the area contributions from both of these components.

Shown in fig. 3.1 is a block diagram of an example of a oversampled multi-channel coder.

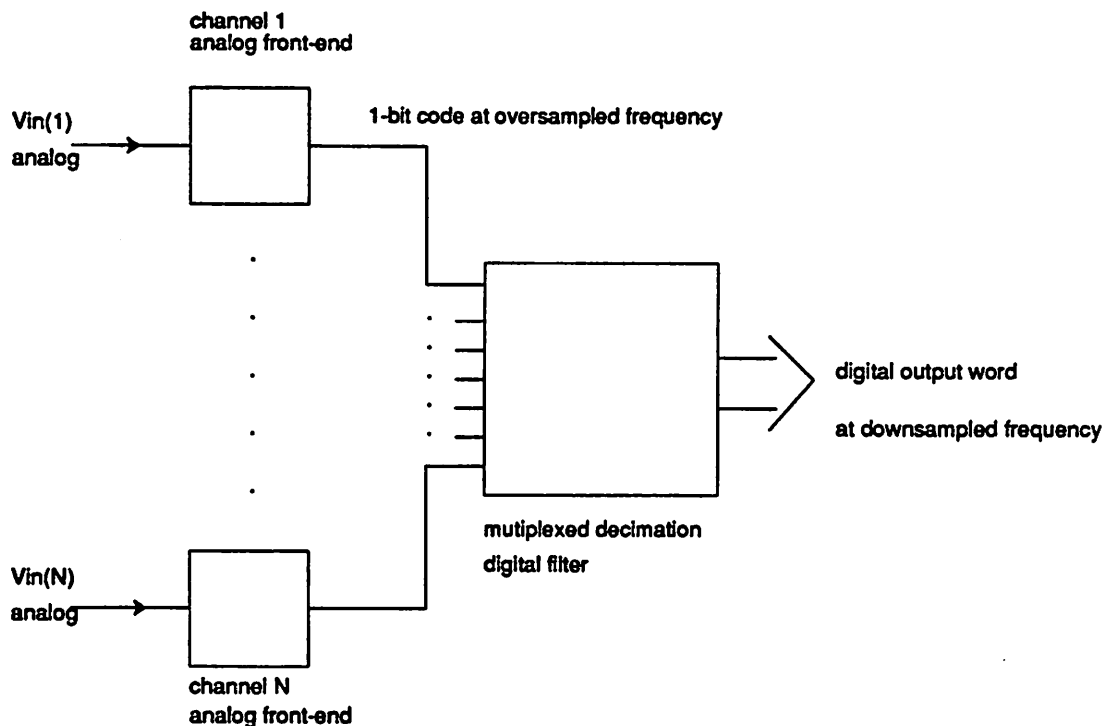


Fig. 3.1 Example of a Multi-channel coder

The analog front end is an integrator loop as shown in fig. 2.2 and the multiplexed digital decimation filter is the one labelled low-pass filter in fig. 2.2. The reason for having separate analog front ends is to

minimize channel to channel crosstalk. Since the analog front end takes up less than 20% of the total chip area, such an approach does not put a heavy penalty on the total chip area. The digital filter however, takes up 80% of the total area and so multiplexing it can result in a big saving. Since it is digital the filter can be multiplexed without having to worry about the crosstalk.

3.2 Discussions on the selection of an oversampled coder

Within the realm of oversampled ADCs there are numerous design options available. To best select the type of oversampled coder in a multi-channel application subject to the constraint of the technology requires a careful analysis of the system aspect of the design. Since our design goal is to minimize area, hardware between different channels should be multiplexed as much as possible. Since analog circuitries cannot be easily multiplexed without introducing crosstalk, separate analog front ends for different channels will be preferred. Therefore oversampling techniques that require simple analog circuits are favored. Following this argument the noise shaping and the predictive coder that use multi-bit ADC and/or DAC are excluded. This is because the multi-bit ADC and DAC require more complicated analog front end than a one-bit version. Consequently the interpolative coder¹ will not be a good candidate. The delta modulator, the coder that uses predictive coding and a one-bit quantizer, is very simple in terms of hardware but suffers from the slope overload problem as discussed in chapter 2. On the contrary, a sigma delta modulator, which uses noise shaping coding and a one-bit quantizer, does not have this problem and has simple analog front end. Therefore it is the coder best suited for this application. Only two sigma-delta architectures are considered: the first order and the second order because higher order coders can be unstable.² Another type of sigma-delta coder, the feedforward coder³ that needs to depend on the cancellation of the first order noise from two first order sections to achieve good performance. The theoretical achievable performance of such a coder can be no better than that of a second order coder. Also its quantization noise transfer function is identical to that of a second order coder discussed in chapter 2. Hence the discussion on the second order coder suffices to cover it.

Both the first order and the second order coders need to have a digital filter to attenuate the noise that is out of the voice band. Compared to the second order coder a first order coder has a simpler analog front end and demands lower digital filter requirement. However a second order coder needs a smaller

oversampling ratio that translates into a lower speed requirement. With a 3u CMOS technology circuits can be built fast enough to handle both the first and the second order coders. Since the design goal of the project is to minimize area and since the 80% of the chip area comes from the digital filter, therefore the first order coder is selected. The following discussions go into the details of the comparisons between the first and the second order approaches and how the digital filter is selected.

3.3 First order versus second order loop

This section tries to determine how a first order sigma-delta coder compares with a second order sigma-delta coder in the implementation of a PCM voiceband coder. The comparison is viewed in the framework of minimising the total silicon area taken. Shown in fig. 3.1 is a block diagram of the proposed oversampled multi-channel coder.

As discussed above the bulk of the area comes from the digital decimation filter, therefore the size of the decimation filter needed in the first and the second order case will be compared examined.

3.4 Requirements on the decimation filter

There are various ways of implementing the decimation filter, but they all have to fulfil the following three functions:

- 1) suppress the out of band noise
- 2) suppress the aliased component to more than 33db down from the signal
- 3) maintain a passband ripple that is less than 0.25db from 300hz to 3.4khz

To understand the significance of these three functions, let us take a look at fig. 3.7 which has the frequency response of the filter and the noise power. First the oversampling frequency is defined as F_s , the downsampling frequency as f_s and the signal bandwidth is f_c which is less than $\frac{f_s}{2}$. The frequency labelled f_a is $f_s - f_c$ and is the lowest frequency that will be aliased into the inband. Then $N(n)$ is defined as the quantization noise spectral density which is a function of frequency f . However if the spectral density is broken up into segments as shown in fig. 3.6, then $N(n)$ will be constant within each segment and $N(n)$ will just be a function of n , the segment number.

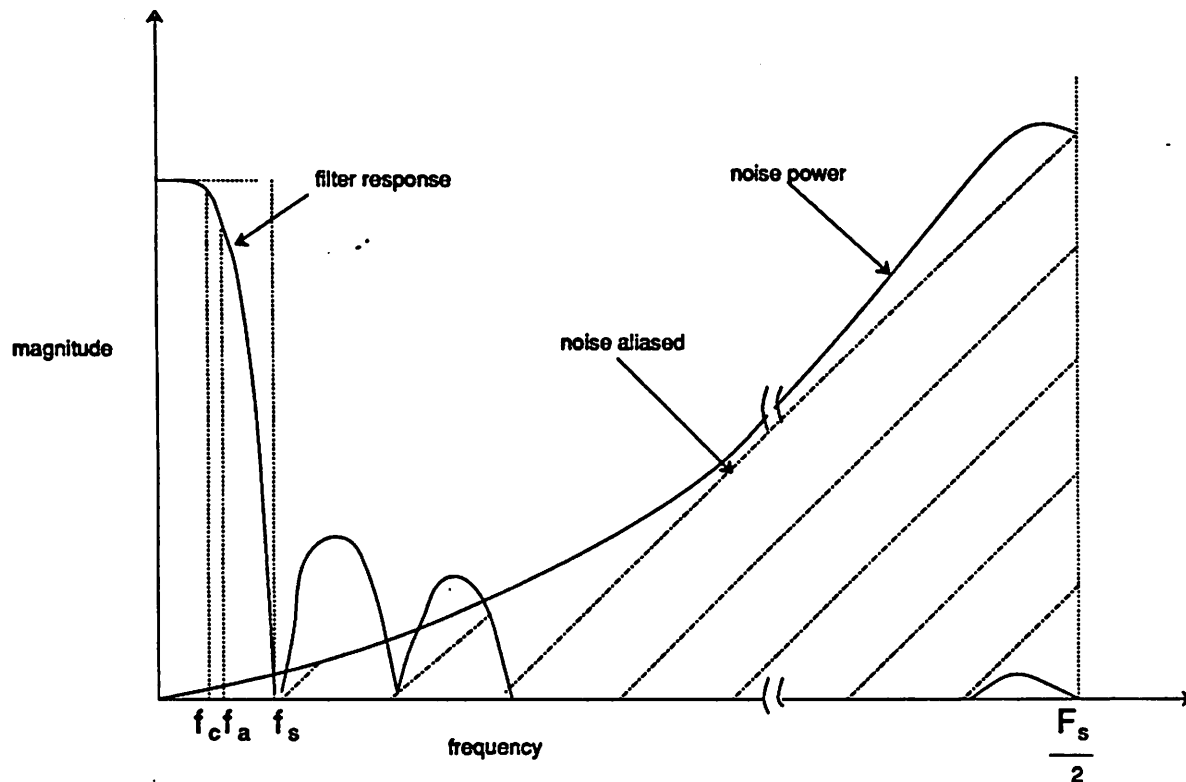


Fig. 3.2 Example of Frequency response of filter and noise spectral density of coder

Let us start with requirement number one: suppression of out of band noise. It has been noted that all the noise power in the shaded part will be attenuated by the filter and then aliased back into the baseband f_c after decimation. The formulae developed in section 2.6 assume the filter has infinite suppression in the stopband so all the noise power that is out of band will be suppressed and zero noise power will be aliased back. Hence the only quantization noise is the one that is inband to begin with. In reality such a filter does not exist and for any real life filter the effectiveness of the filter is gauged by the ratio of the inband noise power to the aliased noise power. The criterion can be arbitrarily set such that the ratio has to be more than one for an acceptable filter. In terms of SNR what that means is the peak SNR will be 3db less than that predicted by section 2.6. Now obviously the transfer function in the analog front end determines the spectrum of the noise power. To a first order approximation the $N(n)$ of a first order coder has n^1 dependency and $N(n)$ of a second order coder has a n^2 dependency as shown in fig. 3.6a.

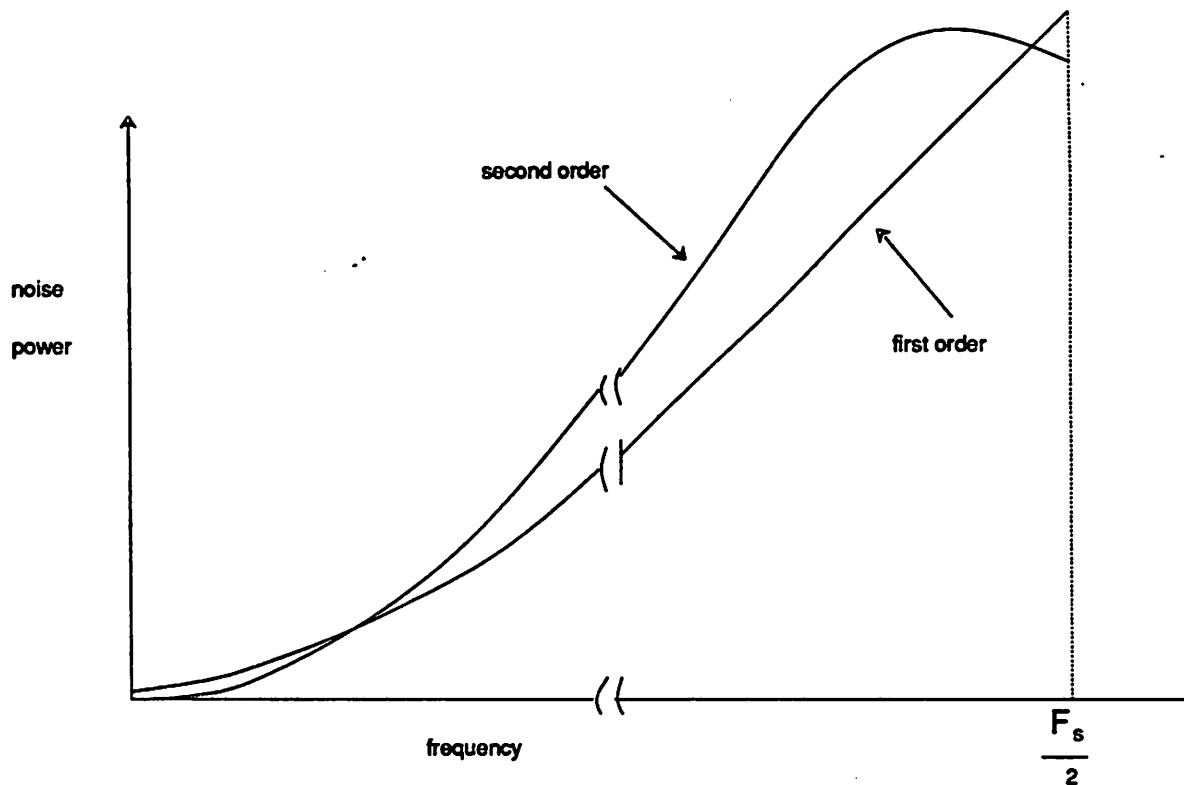


Fig. 3.3 Quantization Noise spectra of first- and second-order sigma-delta coder

Let us take a look at requirement number two: suppress the alias component. Other than the quantization noise that will be aliased in band, spurious signals generated inside the circuit at high frequencies can also be aliased inband, some notable examples are the clock noise of the analog circuits and the switching noise in the digital circuits. These signals have to be attenuated or else once aliased inband they will be indistinguishable from the input signal. For a PCM voiceband coder there is a specific requirement of a 33db suppression of all these signals, meaning that in fig. 3.2 the filter response at f_a has to be at least 33db down from that at DC.

As for requirement number three it can be seen that the filter response falls off from DC to f_c . The codec specs demands such a droop to be less than 0.25db

3.5 Basic filter architecture

To design a filter that satisfies the above three criteria there are a few degrees of freedom, two most notable ones being:

1) there is a choice between an FIR(finite impulse response) filter and an IIR (infinite impulse response) filter

2) since the filter involves a decimation or a change in frequency the filtering can be done in one stage or multi-stages as shown in fig. 3.8

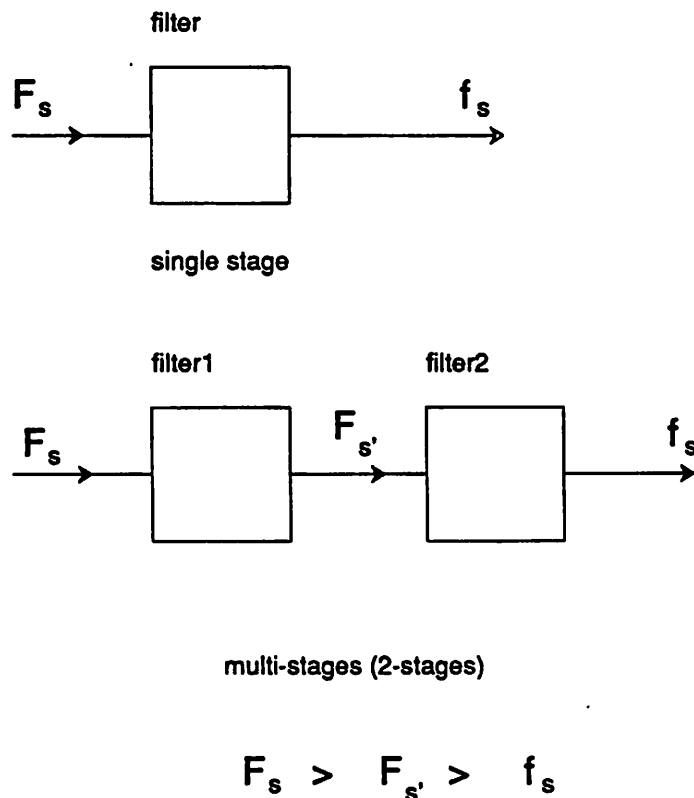


Fig. 3.4 single stage versus multi-stages

Before these degrees of freedom have been examined some observations will be made that will narrow the search space. First because the filter is a decimation filter it is best implemented as an FIR filter. This is because there is no feedback term in an FIR filter and so computations needed to be done at the downsampled frequency f_s instead of the oversampled frequency F_s . To be complete there is a class of IIR filter⁴ that has a specific form in which the denominator of the transfer function only has terms that are powers of D (that is the denominator is of a form like z^{-D}) where $D = \frac{F_s}{f_s}$ and is the decimation ratio. This will in principle enables the computation of the IIR filter to be done at f_s only. However several problems arise from this implementation. One is the fact that a certain restriction is put on the frequency response of

this kind of filter and the filter tends to have a lot of peaking internally. More importantly is the fact that in a sigma-delta coder the input signal to the decimation filter is a one bit code. That means only an AND gate is needed instead of a multiplier to form the product of the input and the coefficient. This is true in an FIR filter. With the IIR filter because there is feedback, so the multiplication that is necessary in the feedback loop is no longer between a one-bit code and the coefficient. This is true even for the class of IIR filter mentioned above.

Once it has been decided that an FIR filter is the choice the advantage of having a multi-stage over a single stage is obvious. That is because of the passband requirement. For a 0.25db ripple inband, implementing the FIR filter in one stage will be very inefficient. The obvious choice is to separate the filtering into two parts. In the front is a decimation filter that takes care of the out of band noise suppression and the anti-alias requirement, followed by an IIR filter to shape up the passband. Furthermore since the first FIR filter has to do its computation at the incoming oversampling frequency F_s , with a 3 μ technology each channel needs to have its own FIR filter whereas for the second stage IIR filter the computation rate can be done at the downsampled frequency and so it can be multiplexed by the different channels. The filter architecture will then be as shown in fig. 3.5 With this basic architecture the comparisons between a first order and a second order coder can be made.

3.6 Optimum filter for multi-channel sigma-delta coders

As shown in fig. 3.5 the bulk of the area comes from the FIR filter since the datapath of this filter cannot be multiplexed at the present technology. Consequently the choice between a first order and a second order coder depends on how much area these two approaches will lead to in implementing the FIR filter. Now an FIR filter is consisted of two things, the control path that generates the coefficients (for example, a ROM or a PLA) and the datapath (the adder and accumulator) that computes the output. For a multiplexed filter the coefficients needed for the different filters are the same so the control path of the FIR filter can be shared. Accordingly the bulk of the area comes from the datapath. This has a lot of impact on the design because the size of the PLA depends on the number of coefficients and hence on L , the filter length, whereas the size of the datapath depends on the number of adders which equals the number of

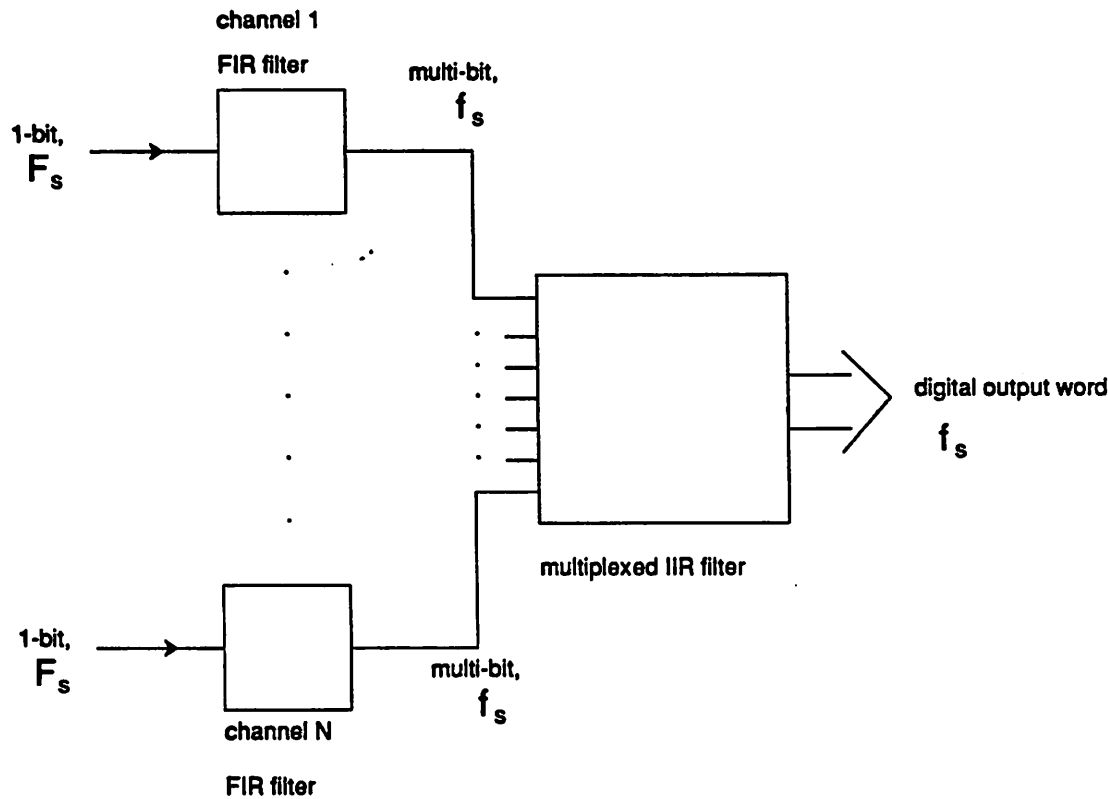


Fig. 3.5 Proposed filter architecture

addition needed per sample and this equals L/D , where D is the decimation ratio as defined above. The distinction between these two numbers is very significant as will be shown in the following sections.

In this section the relation between the system architecture, that is whether it is first order, second order, one stage or multi-stage and the area parameter, that is the L/D ratio will be established the general frequency response of the filter required will be established. Starting with the goal of trying to find a filter that will satisfy the requirement on suppressing the out of band noise leads to the investigation of the roll-off characteristics of this filter in the stopband, because that determines how much the out of band noise is suppressed. This rolloff would be normalised to the inband frequency f_s . Also it is assumed that for all practical purposes that $f_c = \frac{f_s}{2}$ in our derivation to simplify the analysis. In other words it is to be found out every time the frequency is increased by f_s , how much the filter response should drop. As shown in fig. 3.6 if it is assumed that the filter response and the $N(n)$ is constant in a segment of frequency whose width is f_s , then the total out of band noise that is going to be aliased back as shown in fig. 3.6 can be computed.

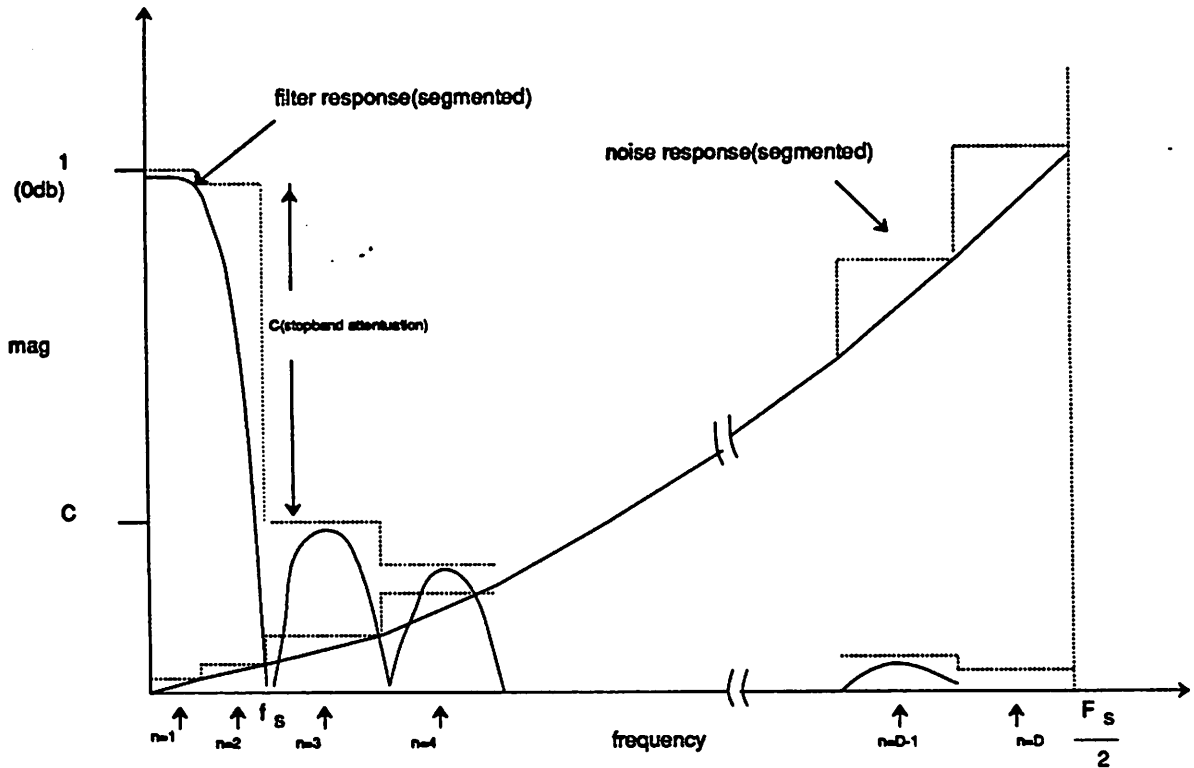


Fig. 3.6 Diagram showing the segmented filter response and noise density

Let us define N_{total} as the total rms noise aliased back into the baseband. The noise in each segment is defined as $N(n)$ which is a function of n , the segment number. The filter response, $filter(n)$ in each segment will be defined as a function of n , the segment number. Then it can be shown that

$$N_{total} = \sum_{n=2}^{n=D} (filter(n))^2 (N(n))^2 (\Delta W) \tag{3.1}$$

Since for a first order coder if the $N(n)$ is measured and if it is assumed to be constant within one segment of the frequency with width = f_s , then $N(n)$ increases linearly with the segment number. As shown in fig. 3.6 the segments are divided into two regions. For $n=3$ to D

$$N(n) = \frac{n}{D} \frac{\sigma_e}{\sqrt{F_s}} \tag{3.2}$$

where

D =decimation ratio

σ_e = variance of noise

F_s =oversampling frequency

For $n=1, 2$

$$N(n) = \frac{n}{2D} \frac{\sigma_e}{\sqrt{F_s}} \quad (3.3)$$

For a second order coder $N(n)$ increases as the square of the segment number. Again for $n=3$ to D

$$N(n) = \left(\frac{n}{D}\right)^2 \frac{\sigma_e}{\sqrt{F_s}} \quad (3.4)$$

For $n=1,2$

$$N(n) = \left(\frac{n}{2D}\right)^2 \frac{\sigma_e}{\sqrt{F_s}} \quad (3.5)$$

Let us take a look at the filter response. As will be shown later, in order to satisfy the anti-alias requirement the first zero is put at the resampling frequency, that is at f_s as shown in fig. 3.6. Then filter frequency response is separated into three parts: one for $n=1$, one for $n=2$ and the other from $n=3$ to $n=D$.

For $n=1$

$$\text{filter}(n) = 1 \quad (3.6)$$

For $n=2$, as shown in fig. 3.6 there is a sharp curve and the step approximation is not accurate so numerical integration has to be done when try to compute the aliased noise.

For $n=3$ to D

$$\text{filter}(n) = C \left(\frac{1}{(n-2)^y} \right) \quad (3.7)$$

where C =stopband attenuation that depends on the type of filter (window) but not on L .

and y =order of the filter.

From equations 3.1, 3.2, 3.4, 3.7 the aliased noise in general will have a form like

$$N_{total} = N(2) + C \sum_{n=3}^{n=D} \left(\frac{\left(\frac{n}{D}\right)^2 \frac{\sigma_e}{\sqrt{F_s}}}{(n-2)^y} \right)^2 \frac{F_s}{D} \quad (3.8)$$

where N_{total} = total aliased noise

$N(2)$ = noise in segment 2 that depends on the exact window shape in segment 2 as shown in fig. 3.6

x =order of the coder ($x=1$ for first order and 2 for second order coder)

y =order of the filter

This sum N_{total} should be smaller than the inband noise which is $N(1)$ or the noise in segment 1 given as

$$N(1) = \sum_{n=1}^{\infty} \text{filter}(n)^2 (N(n))^2 (\Delta W) \tag{3.9}$$

From equations 3.9, 3.3, 3.5, 3.6

$$N(1) = 1^2 \left(\left(\frac{1}{2D} \right)^x \frac{\sigma_f}{\sqrt{F_s}} \right)^2 \frac{1}{2D} \tag{3.10}$$

where x =order of the coder ($x=1$ for first order and 2 for second order)

In general, for most filters numerical integration shows that $N(2)$ is about equal to the inband noise $N(1)$. Therefore the rest of the terms has to sum up to zero. The exact calculation depends on C but for most practical purposes the sum should converge as n goes up. For a practical system n can be from 100 to 1000 and it can safely be replaced the sum by an integral. As it is known if the integrand $\frac{1}{n^{y-x}}$ has $y-x$ just less than 1, the integral will converge. If $y-x$ is larger than 1 the integral will diverge and so $y-x$ is picked to be 1 as the final choice. That means for a first order coder a window that has a response going down as n^2 should be selected. For a second order coder the response should go down as n^3 . The former can be realised by a triangular window and the latter by a so called *sinc*³ window which is just a triangular window followed by a rectangular window. Note that the above analysis can be extended to higher order sigma-delta coders as x is not restricted to one or two. That will permit us to predict how complicated the FIR filter needed to be for higher order filters.

Instead of looking at the frequency domain Candy has come up with a derivation based on the time domain for the optimum filter for a first order coder.⁵ The window that is optimum is a parabolic window but a triangular window is very nearly optimum. The derivation is based on a L/D ratio of one but can be extended easily to other L/D ratios. No such derivation has been published for a second order case.

3.7 Significance of the L/D ratio

Now if the above two windows are selected what will the L be? Since the rolloff has to be quadratic for every segment, it turns out at every increase of n which corresponds to a step of f_s in frequency as shown in eq. 2 and 4, the window has to go down by 1/4 or 12db, that means 2 zeroes have to be placed since each zero contribute a falloff of 6db. But $\frac{F_s}{f_s}=D$ the decimation ratio. Therefore the total number of zeroes= 2D which in turn equals L. Consequently L/D=2. Turning to the second order case the rolloff for every segment should be cubic, therefore every time f is increased by f_s the rolloff has to be 1/8 or 18db, meaning 3 zeroes have to be placed now. Arguing as above L/D=3. The importance of the above discussion lies in the fact that not only does it apply to sigma-delta coders of any order, but also to interpolative coders that have multi-bit A/D and D/A, as well as having a predictor in the feedback loop. The only thing that one has to watch out for is in this case the σ_e is a lot smaller to begin with, but the same argument holds true as far as the ratio of out of band noise to inband noise is concerned. Actually both the use of multi-bit A/D, D/A and the use of predictive coder serves to reduce σ_e so all that is changed in the above discussion is σ_e . As far as the noise-shaping part is concerned everything remains the same.

A more direct way of coming up with the above L/D ratios is from the observation that since decimation is done in two stages not all the noise is aliased back in band after the first stage. Let the resampling frequency be F_{s1} as shown in fig. 3.4. Only the noise that is centered around $F_{s1}2F_{s1}...nF_{s1}$ is aliased. The rest will be suppressed once more by subsequent filtering which can be done at a much lower rate. In order to suppress the noise that will be aliased back the noise should be suppressed around multiples of F_{s1} . The most direct way of doing that is to put zeroes at these frequencies. How many zeroes should be put then? Since the general rolloff should be proportional to n^2 , 2 zeroes should be placed at each spot for a first order coder and 3 for a second order coder. But the number of spots is just D since $F_{s1}=\frac{F_s}{D}$. Consequently for a first order coder L/D=2 and for a second order coder L/D=3. This method also points out another crucial point, that is regardless of the number of stages the FIR filter is split into, the L/D ratio is the same. That is because if L is reduced by increasing F_{s1} , D is also reduced, since $D=\frac{F_s}{F_{s1}}$. Consequently L/D remains the same for a particular order of sigma-delta coder. So it pays to use as few FIR filter stages as

possible, since the L/D ratio is the same even though the L will increase. This is contrary to the conventional approach of a multi-stage FIR filter implementation for a decimation filter and is something to watch out for in the design of a multi-channel coder.

Having established the fact that a first order coder needs an L/D of 2, the second order coder an L/D of 3 for the out of band noise suppression and that a second stage IIR filter is used to do passband shaping, there is a need to check out the anti-alias requirement. This is very dependent on the application's specification. If at the frequency f_a there is a zero, the suppression is generally larger than 33db. For a triangular filter the first two zeroes are at $2/L$. Actually for most windows the first zeroes (or the width of main lobe) are at $2/L$ except for the rectangular window. That means if $L/D=2$, then $D=2/L$ and F_{s1} is right at the point where there is a zero and consequently the anti-alias requirement is satisfied. This is independent of the coder, or in other words, to satisfy the anti-alias requirement, even for a second order coder, a triangular window is good enough.

Knowing that a minimum of L/D of 2 is needed for a first order and a L/D of 3 for a second order, it can be concluded that in terms of area the first order coder will be better off because that means a first order coder needs two adders and accumulators whereas a second order coder needs three adders and accumulators. Even though a first order coder needs to sample at 4Mhz to get the codec performance whereas a second order coder needs 1Mhz and might be able to multiplex some more of the adders, the state variable cannot be multiplexed, meaning the number of accumulators remains the same. Besides the L/D ratio the number of bits necessary in the coefficients is also very important because that translates into the number of bits of the adder and the accumulator. That number depends on the decimation ratio D and the filter complexity. For a first order coder it is around 14 bits whereas for the second order coder it is around 18 bits. That means both the adder and the accumulator will be larger. The area of the FIR filter for the second order approach comes out to be about 50% more than a first order coder. The extra complexity of control path of the second order coder has not been considered. Because of the simple structure a triangular window can be implemented easily as an updown counter. But generating a sinc^3 window is more involved and may need a PLA. Finally there is also the extra area coming in from the more complicated analog front end. Comparing the areas taken by both approaches the second order analog front end can

occupy 50% to 100% more than that of the first order.

3.8 Other considerations

Besides area there are other factors that come into the comparison picture. A first order coder has the main disadvantage of having to sample at a higher frequency, at 4 Mhz in order to satisfy the codec specs. That means faster analog and digital circuits are required. In a 3u CMOS technology analog and digital circuits can be built fast enough to handle a 4Mhz sampling frequency. There is also the dither consideration. For a first order coder the frequency response exhibits more line like spectrum than the second order coder. For an input signal smaller than $1/D$ of the full scale, some of the lines can appear inband and generate tones. This is not acceptable for a codec, especially in the idle channel condition. To get rid of the tone, a dither signal is applied to break up the line. Additional technique like putting the codec in an auto-zero feedback loop will also help the situation.

For the second order coder the main complexity comes from having more complicated analog front end. Also to maintain stability there is a requirement on how well the components match. Finally there is the possibility of the coder going into overload. Because of more than one feedback loop, the signal at the input to the comparator may exceed a certain range and the coder goes into overload. For a typical coder² this happens when the input signal is larger than 0.5 of full scale and this overload characteristics will cut into the peak SNR of the coder. However in a codec environment because the SNR requirement is less stringent when the signal is at full scale, this is not a serious problem.

3.9 Conclusions

With the above discussion the conclusion that can be drawn is that for a multi-channel codec application using a 3u-CMOS technology a first order sigma-delta coder can be implemented in the least amount of area. In addition, it offers additional advantages of being compatible to digital processes and can exploit the full advantages of scaling easier than other conventional analog approaches.

References

1. Wooley, "An Integrated Per-Channel Encoder Based on Interpolation," *IEEE Journal of Solid-State Circuits*, pp. 14-20, Feb 1979.
2. Candy , "A Use of Double Integration in Sigma Delta Modulation," *IEEE Transactions on Communications*, pp. 249-258, Mar 1985 .
3. Hayashi , "A Multistage Delta-Sigma Modulator without Double Integration Loop," *ISSCC' 86 Digest of Technical Papers*, pp. 182-183.
4. Liu , "Efficient Sampling Rate Alteration Using Recursive (IIR) Digital Filters," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, pp. 1366-1373 , Dec, 1983 .
5. Candy , "Using Triangularly Weighted Interpolation to Get 13-bit PCM from a Sigma-Delta Modulator," *IEEE Transactions on Communications*, pp. 1268-1275, Nov 1976 .

CHAPTER 4

Design Techniques For A Multi-Channel PCM Coder

INTRODUCTION

This chapter is concentrated on looking into various design techniques for a multi-channel voiceband coder. The design techniques include both the architectural and the circuit design level. As concluded in chapter 3 the optimum oversampled coder, in terms of minimum area and crosstalk for a multi-channel voiceband coder, is a first order sigma-delta coder followed by a multiplexed decimation filter as shown in fig. 3.1. The analog front end will be separated for each channel. The decimation filter will be separated into two stages, the first stage being a FIR filter which is a triangular window and the second stage being an IIR filter. The following discussions will be directed towards the design aspects of both the analog front end and the digital filter. Specifically the different filter architectures for the FIR filter and the IIR filter will be discussed. The comparisons will again be made under the area constraint. Various design approaches for the analog front end will be explained. The requirements on the operational amplifier and comparator will then be discussed. Finally the circuit implementation details of an experimental prototype in a 3 μ CMOS process will be explained.

4.1 Digital decimation filter design

4.1.1 The three filters architecture

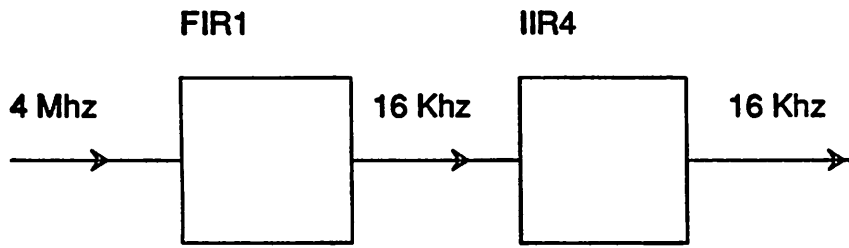
As discussed in chapter 3 the decimation filter is best divided into two filters as shown in fig. 3.5. As it turns out the triangle window used for the first filter cannot satisfy the anti-alias requirement at 16 KHz. Since it is not desirable to increase the complexity of the first FIR filter, the best way to resolve this problem is to introduce another FIR filter. The new structure is as shown in fig. 4.1. where FIR2 decimates from 4 Mhz to 32 KHz, FIR3 decimates from 32 KHz to 16 KHz and the IIR4 does its computation at 16 KHz and decimates its output to 8 KHz. As shown in fig. 4.2 the filter response for FIR2 at f_a (28 KHz) is more

attenuated than the filter response for FIR1 at f_a (12 KHz). Therefore FIR2 can satisfy the anti-alias requirement of 33db down from passband at $f_a = 28\text{kHz}$. The frequency response for FIR3 will be as shown in fig. 4.3 where it will be 33db down at $f_a = 12\text{ KHz}$. FIR3 is an FIR filter with only 5 coefficients. Since its main purpose is to suppress signal around 16 KHz so 3 zeroes are put at that frequency. Since FIR3 operates at 32 KHz it can be multiplexed. IIR4 has frequency response as shown in fig. 4.4. It has a peak up response to compensate for the droop in the passband of the FIR2 and FIR3. Its passband response has a ripple that is less than 0.25db so that the overall filter's passband response will be equiripple and is less than 0.25db. Finally it has stopband suppression of more than 33db around 8 KHz (from 4.6 KHz to 8 KHz). With this arrangement both FIR3 and IIR4 can be multiplexed and the area is increased just by a small amount. FIR2 has about the same complexity as FIR1 because now the L/D ratio is $256/128=2$, same as that of FIR1. If we had used just FIR1 and IIR4, the complexity of FIR1 has to be increased to have a ratio of $L/D=3$ to satisfy the anti-alias requirement of 33db at $f_a = 12\text{ KHz}$. The above argument is particularly valid in a multi-channel coder environment but can also be extended to a single channel case.

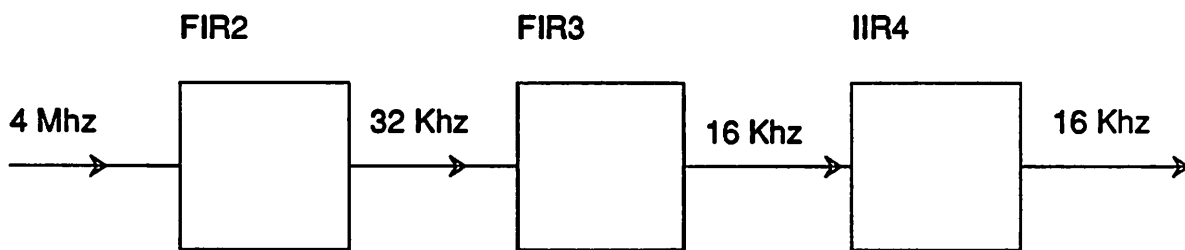
Having shown the frequency responses for the three filters, a full discussion of the issues involved in the implementation of each one of them will follow. Since FIR2 needs to sample at a high rate it will be implemented using a custom approach and this is called the first stage. This means the property of the triangle window should be exploited to reduce area. FIR3 and IIR4 can be multiplexed and they will together constitute a second stage.

Alternative first stage FIR architectures

Since a multi-channel approach is used the control can be done relatively easy. It can be done using a ROM, a PLA or just random logic. Since a triangle window used, a counter seems to be the best choice. Also attention has to be paid to the fact that the adder should be multiplexed to save hardware. So the random logic should be realized to implement this efficiently.



Scheme 1



Scheme 2

Fig. 4.1 Three filters architecture

4.1.2 ROM-based FIR filter

This approach has been widely used¹ as it can be used for windows for arbitrary length and shape, that is, windows more complicated than a triangular window. An example is included in fig. 4.5. where the L/D is two. The number of accumulators is equal to the L/D ratio. The important advantage here is that only half of the ROM needed be stored because the FIR filter's coefficients are symmetric. Also the use of the shuffler is to activate the proper adder when the L/D ratio is larger than one.

The operation of the circuit is as follows: For an FIR filter as shown in fig. 4.6 the output at sample n is defined as:

$$y_n = \sum_{i=1}^{i=L} c_{n-i} x_{n-i} \tag{4.1}$$

where $c_n = \text{coefficient}$

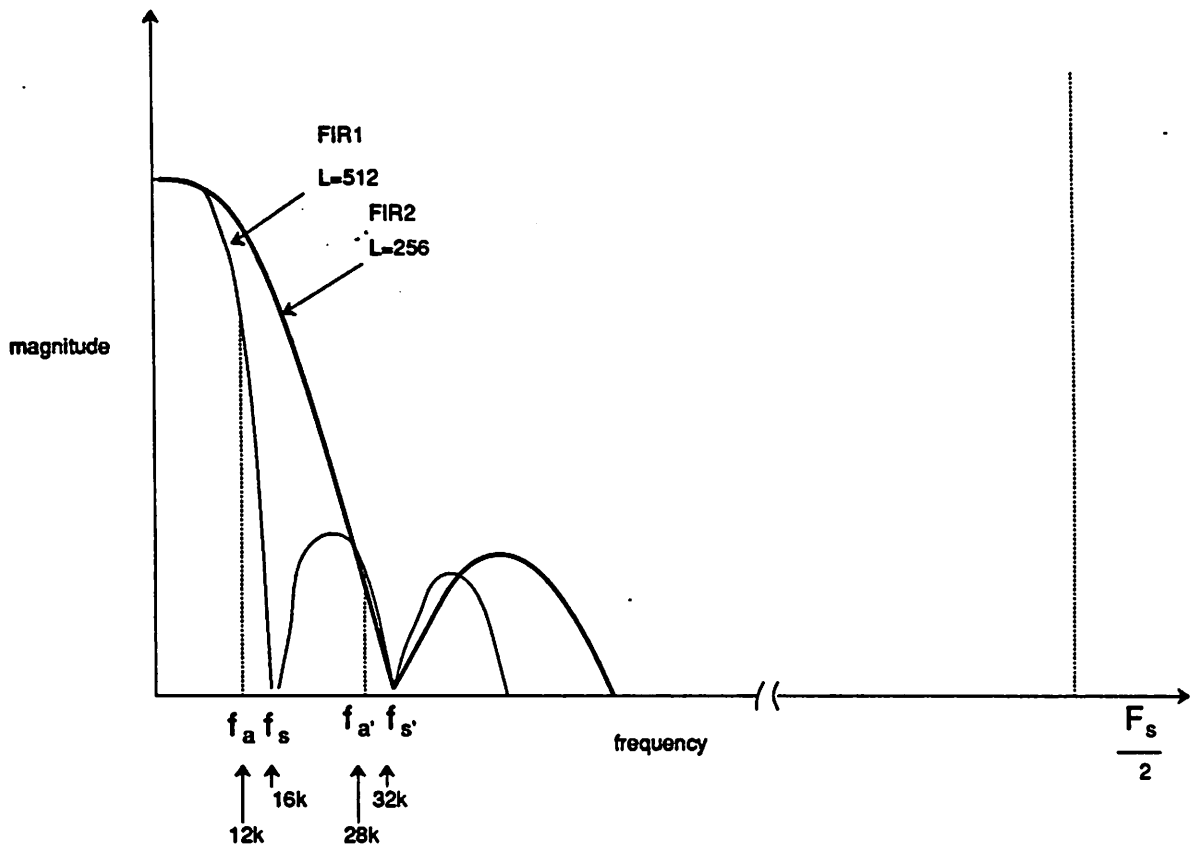


Fig. 4.2 FIR1 and FIR2 frequency responses

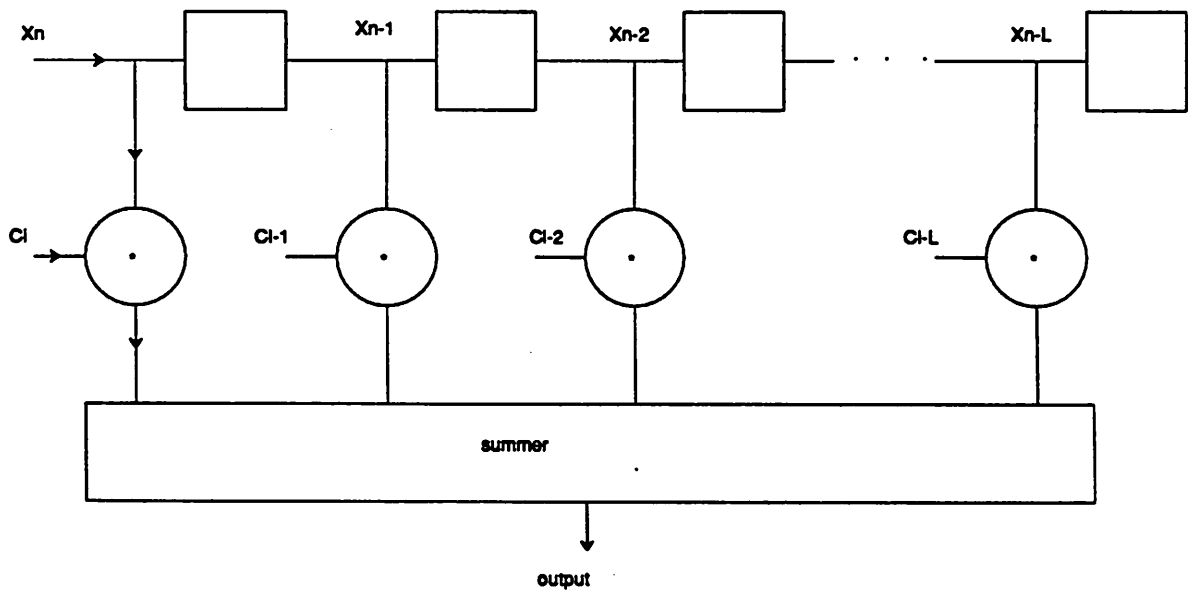


Fig. 4.6 conventional FIR filter

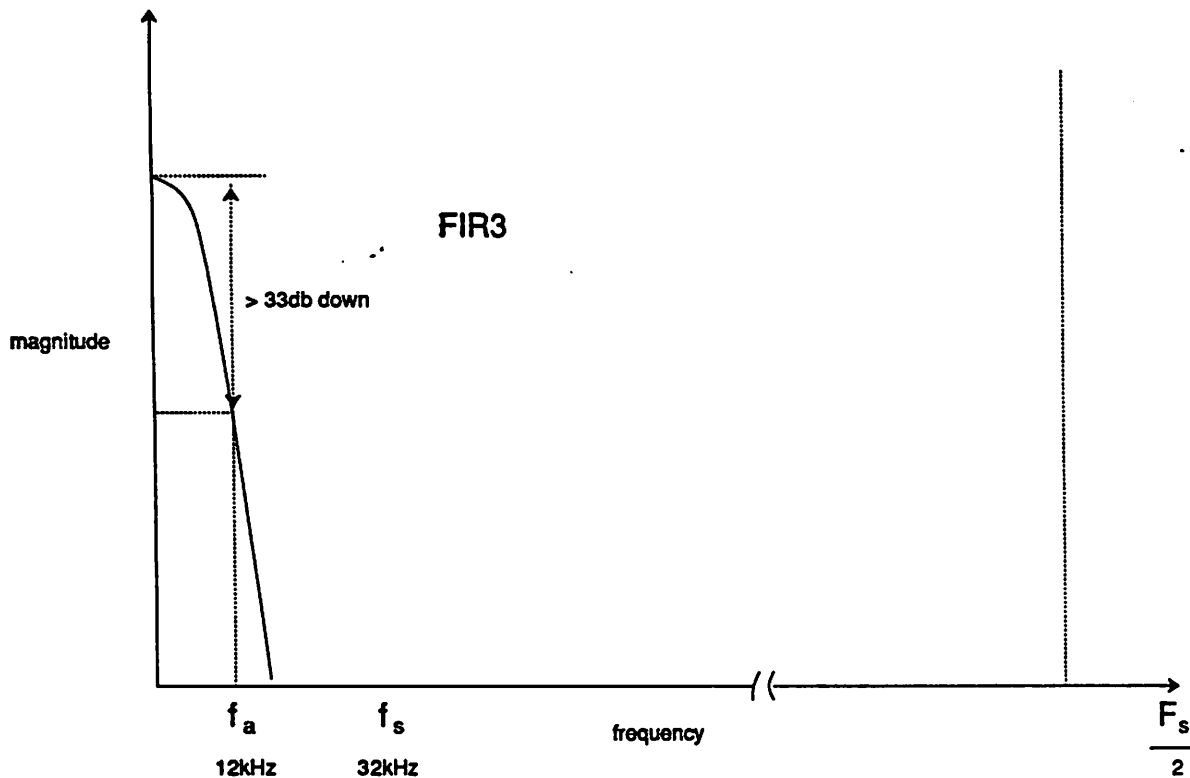


Fig. 4.3 FIR3 frequency response

That means L adders (or alternately one L-input adder) and multipliers are needed because for any input sample L multiplies are done and one addition with L inputs are done. Now since the input is a one-bit code the multiplier is replaced by an AND gate. Furthermore if there is decimation and if the the decimation ratio $D=L$ there is a need to compute only once out of L. That is, y_{n-1} to y_{n-L} can be skipped. There is a need to compute only y_n . But y_n can be computed in L samples. With L additions to be done in L samples, only one binary adder is necessary. That adder has to perform the sum at eq. 4.1. Now since only one addition can be done per sample the coefficient c_n needed to be generated once per sample. Referring to fig. 4.5, that is what the coefficient ROM is doing, that is, at every sample the address is updated by the counter. Extending to L/D larger than one case, in this case the $L/D=2$ means that at every sample two additions needed to be done. This can be accomplished by having two adders. Now since the filter outputs once every L/2 times the sum that needed to be computed is

$$y_n = \sum_{i=1}^{i=L} c_{n-i} x_{n-i} \tag{4.2}$$

$$y_{n+D} = \sum_{i=1}^{i=L} c_{n+D-i} x_{n+D-i} \tag{4.3}$$

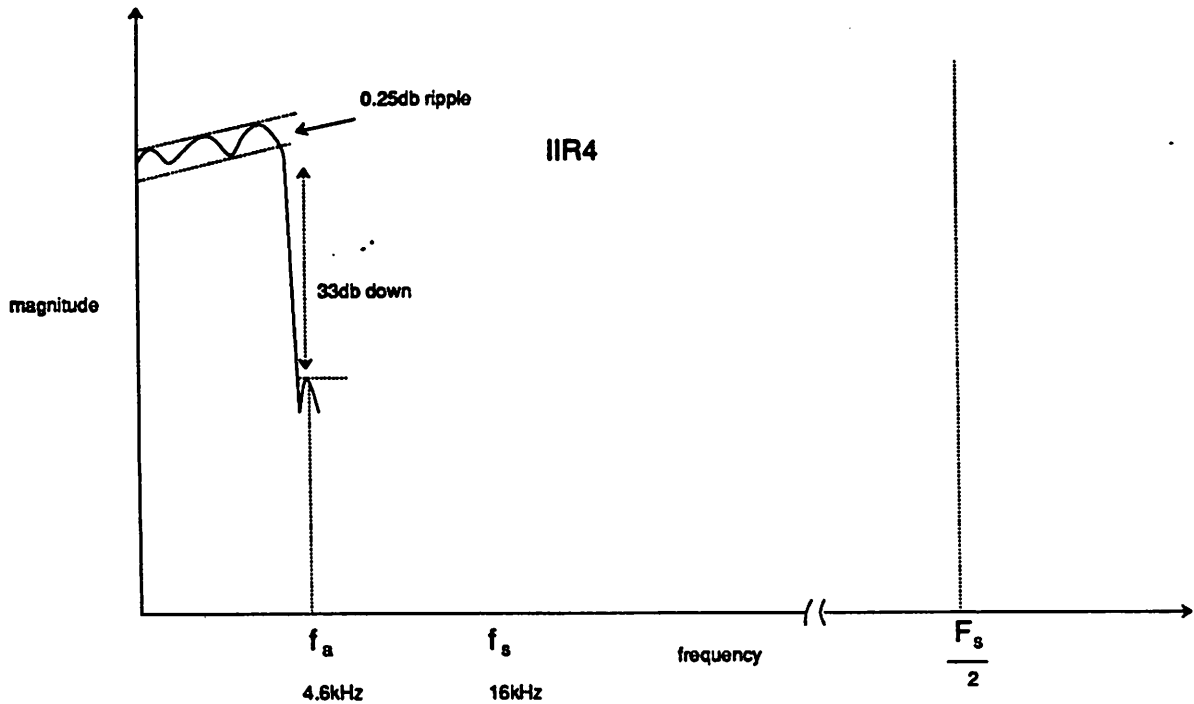


Fig. 4.4 IIR4 frequency response

Since two sums are needed there are two sets of coefficients that needed to be supplied to the two adders per input sample. These two sets of coefficients are, however, spaced from one another by $D=L/2$. So that is what the shuffle network is doing. The ROM is divided into two columns, one with coefficients from c_1 to $c_{L/2}$, the other from $c_{L/2+1}$ to c_L . At any sample the shuffle network will switch the correct coefficient to the proper adder. Similarly the multiplexer will decide which adder sum is shifted to the output, depending on whether y_n or y_{n+D} is needed. There is one further simplification that can be achieved due to the symmetry of the FIR filter. Therefore only half of the coefficients needed to be stored. When the coefficient reaches $c_{n+L/2}$ we just count down again. This is accomplished by the up/down counter.

Differential encoding

There is the choice of whether one should encode the coefficients or whether one should do encoding differentially. That means in the ROM instead of putting c_n

$\delta c_n = c_n - c_{n-1}$ is placed. The advantage of doing this is the reduction of the number of bits required. That is because the FIR filter coefficients vary slowly. However an additional adder is needed to recover the coefficient before it get multiplied by the input sample. The final circuits also contain some extra control

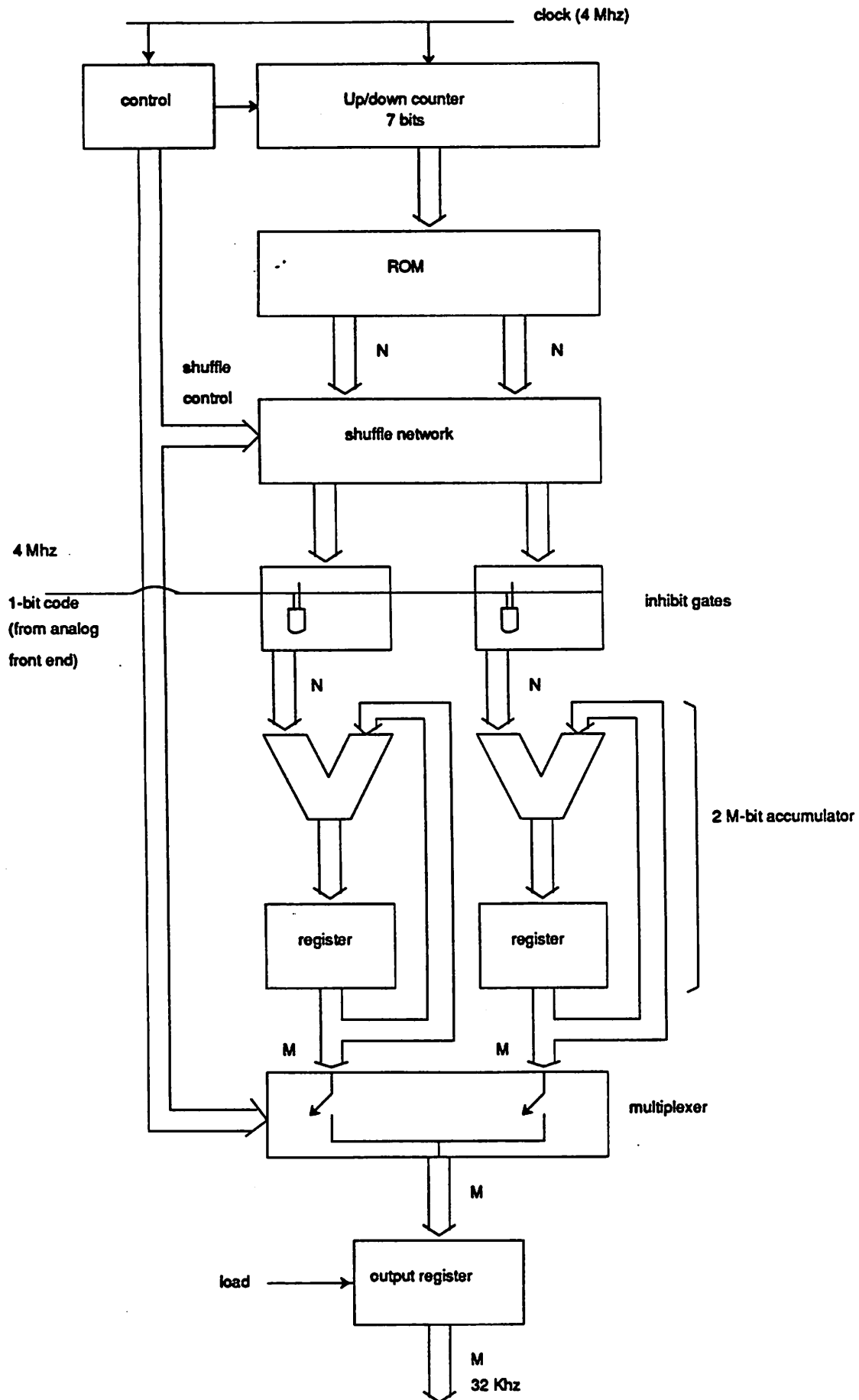


Fig. 4.5 ROM-based architecture

logic. The merit of going into differential encoding depends on how many bits is needed in the coefficient to begin with. Under the present circumstance the extra circuits and routing do not justify such a scheme.

Custom FIR filter for a triangle window

The above approach can be used for any FIR filter as long as it is symmetric, which is true for most of the FIR filter used for sigma-delta coder. However for a first order coder, if one is going to use a triangle window, then the simple shape of the time domain impulse response of the FIR filter can be further exploited. Two such approaches will be outlined.

4.1.3 Custom triangle window by Candy

This design is originated by Candy.² As shown in fig. 4.7 it consists of no counter and no ROM. Instead it exploits the property of a triangle window as a rectangular window followed by another rectangular window. Now a rectangle window that has a decimated output can be implemented by an accumulator and so the circuit is consisted of two accumulator stages. The basic operation of the circuit is as follows: The first stage is used to accumulate the input sample through a register and an adder to get node0. The signal C_D is the reset signal. It enables the addition for D samples, then it resets the register and starts a sequence of operations to compute the final output. This sequence is consisted of multiplying node0 by D to get node1 which is the area of the shaded part as shown in fig. 4.8. Simultaneously the sum is passed to node2 which is the shaded area in fig. 4.8. It can then be shown

$$node\ 1 = D \sum_{i=0}^{i=D} X_i \quad (4.3)$$

$$node\ 2 = \sum_{i=0}^{i=D-1} (D-i) X_i \quad (4.4)$$

$$node\ 3 = node\ 1 - node\ 2 = \sum_{i=0}^{i=D-1} i X_i \quad (4.5)$$

Node3 will be as shown in fig. 4.8 and if it is shifted and added to node2, the output will be generated which is the output of a triangle window outputted every D samples.

Note that this approach has the advantage of having no coefficient ROM. Also the first accumulator needs to be about half the length of the second accumulator and some savings can be achieved. Another important advantage is that it does not assume the input code is one bit, so if a noise-shaping coder using

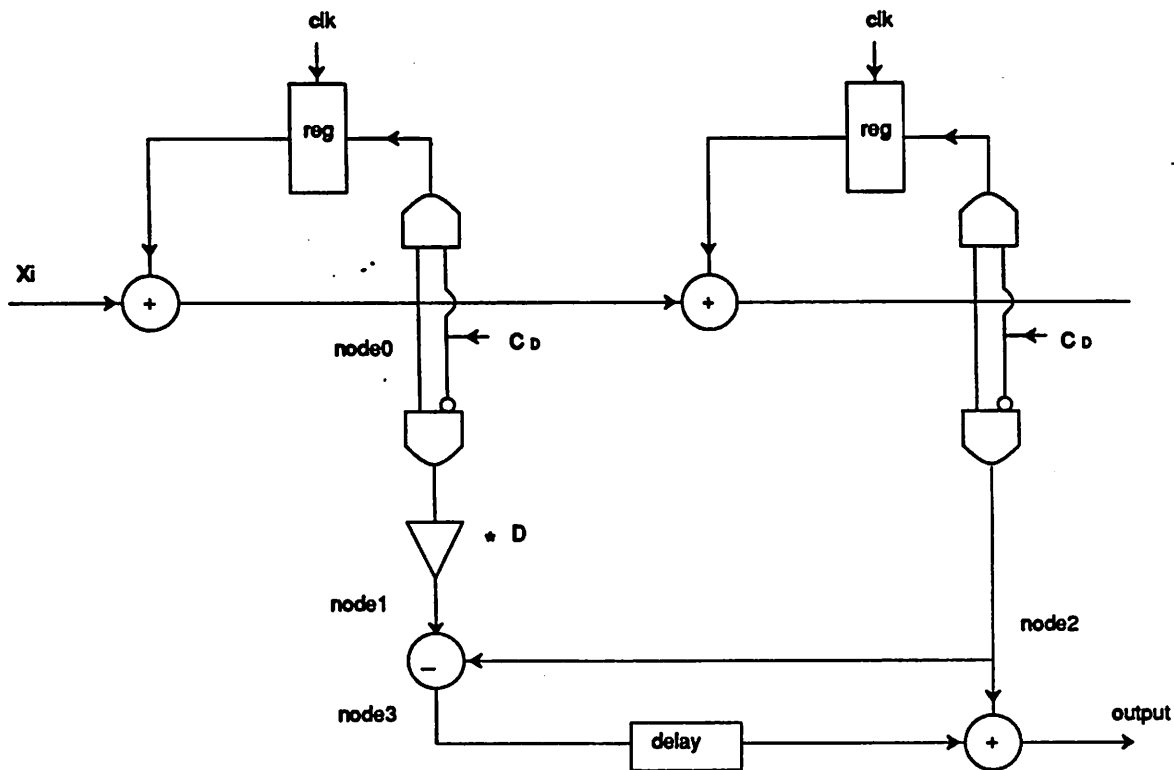


Fig. 4.7 Candy's FIR filter

multi-bit ADC and DAC is used this scheme can also be used without introducing any multiplier. Similarly if a multi-stage FIR filter approach is used the latter stage will have a multi-bit input and this scheme will be useful. The main hardware consists of the two accumulators that needed to be implemented at high speed. All the other logic like the subtraction, delay and final addition can be done at low speed and can be multiplexed with the hardware for the second stage FIR filter and IIR filter.

Disadvantages

This scheme works well if one is going to use two adders for the filter. However for a 3 μ CMOS technology a 16bit adder can be designed to run at around 10 Mhz. Since the sampling frequency is only 4 Mhz the adder should be multiplexed to save area. Unfortunately this approach does not lend itself to multiplexing easily without introducing extra registers as temporary storage elements as well as a host of control lines. The advantage of having one smaller adder as presented above is lost.

4.1.4 Prototype Custom FIR filter for a triangle window

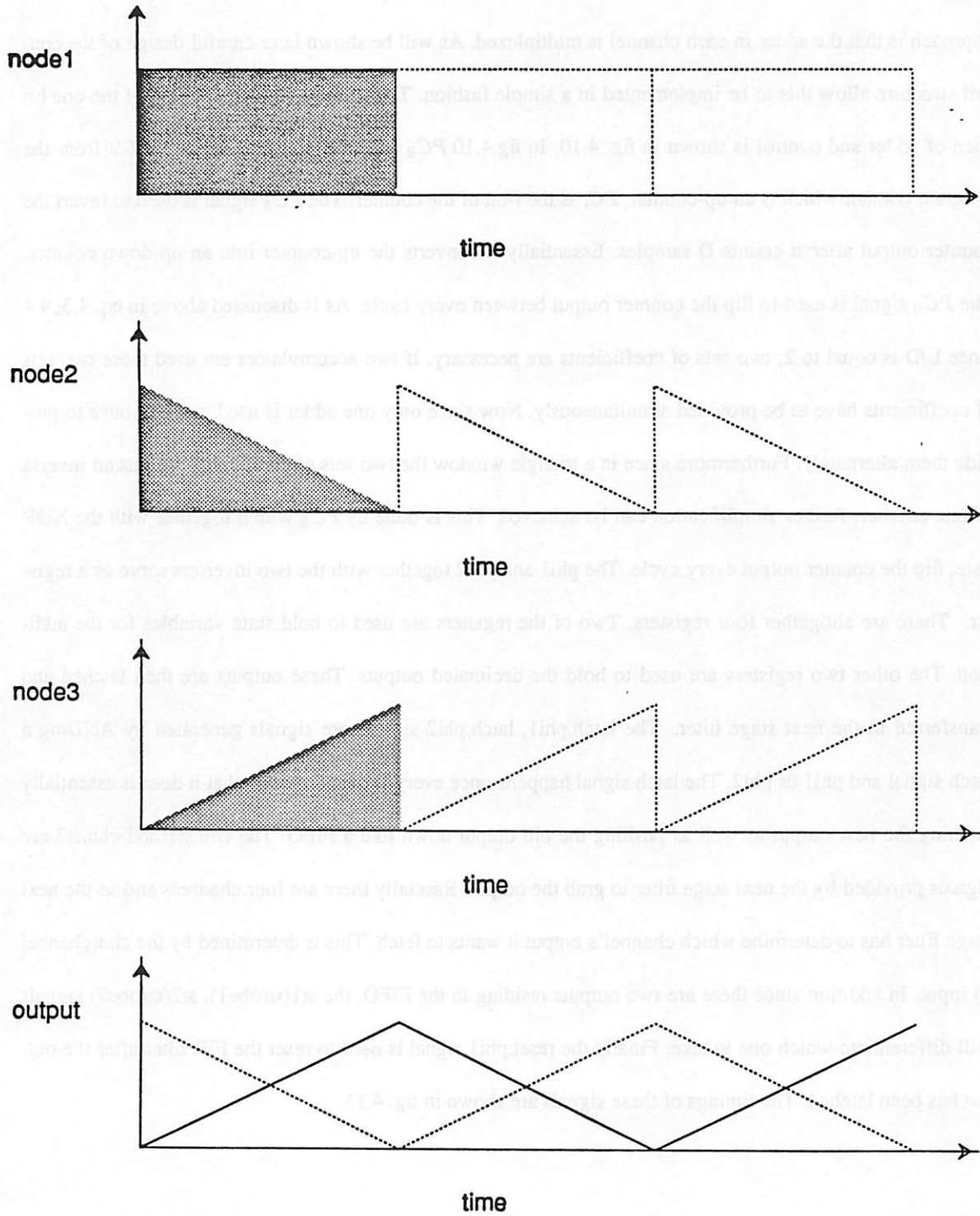


Fig. 4.8 timing diagram

In this section the prototype custom FIR design will be discussed. Shown in fig. 4.9 is the architecture of the FIR filter. It is shown for a four channel filter where one counter provides all the coefficients.

The X1 through X4 are just the one bit code from the four analog front ends. The distinct feature of this approach is that the adder in each channel is multiplexed. As will be shown later careful design of the control structure allow this to be implemented in a simple fashion. The detailed implementation of the one bit slice of adder and control is shown in fig. 4.10. In fig.4.10 PC_0 and PC_8 are the LSB and MSB from the program counter which is an up-counter. PC_i is the i-bit of the counter. The PC_8 signal is used to invert the counter output after it counts D samples. Essentially it converts the up-counter into an up-down counter. The PC_0 signal is used to flip the counter output between every cycle. As is discussed above in eq. 4.3, 4.4 since L/D is equal to 2, two sets of coefficients are necessary. If two accumulators are used these two sets of coefficients have to be provided simultaneously. Now since only one adder is used, we will have to provide them alternately. Furthermore since in a triangle window the two sets of coefficients are just an inverse of one another, further simplification can be achieved. This is done by PC_0 which together with the XOR gate, flip the counter output every cycle. The phi1 and phi2 together with the two inverters serve as a register. There are altogether four registers. Two of the registers are used to hold state variables for the addition. The other two registers are used to hold the decimated outputs. These outputs are then latched and transferred to the next stage filter. The latch.phi1, latch.phi2 signals are signals generated by ANDing a latch signal and phi1 or phi2. The latch signal happens once every D samples and what it does is essentially latching the new output as well as pushing the old output down like a FIFO. The chn.st1 and chn.st2 are signals provided by the next stage filter to grab the output. Basically there are four channels and so the next stage filter has to determine which channel's output it wants to fetch. This is determined by the chn(channel n) input. In addition since there are two outputs residing in the FIFO, the st1(strobe1), st2(strobe2) signals will differentiate which one to take. Finally the reset.phi1 signal is used to reset the FIR filter after the output has been latched. The timings of these signals are shown in fig. 4.11.

4.1.5 Circuit designs for the FIR filter

The major speed limitation of the FIR filter is the ripple carry chain. Therefore efforts are made to make the adder runs as fast as possible. Two adder designs have been used in two different chips and their

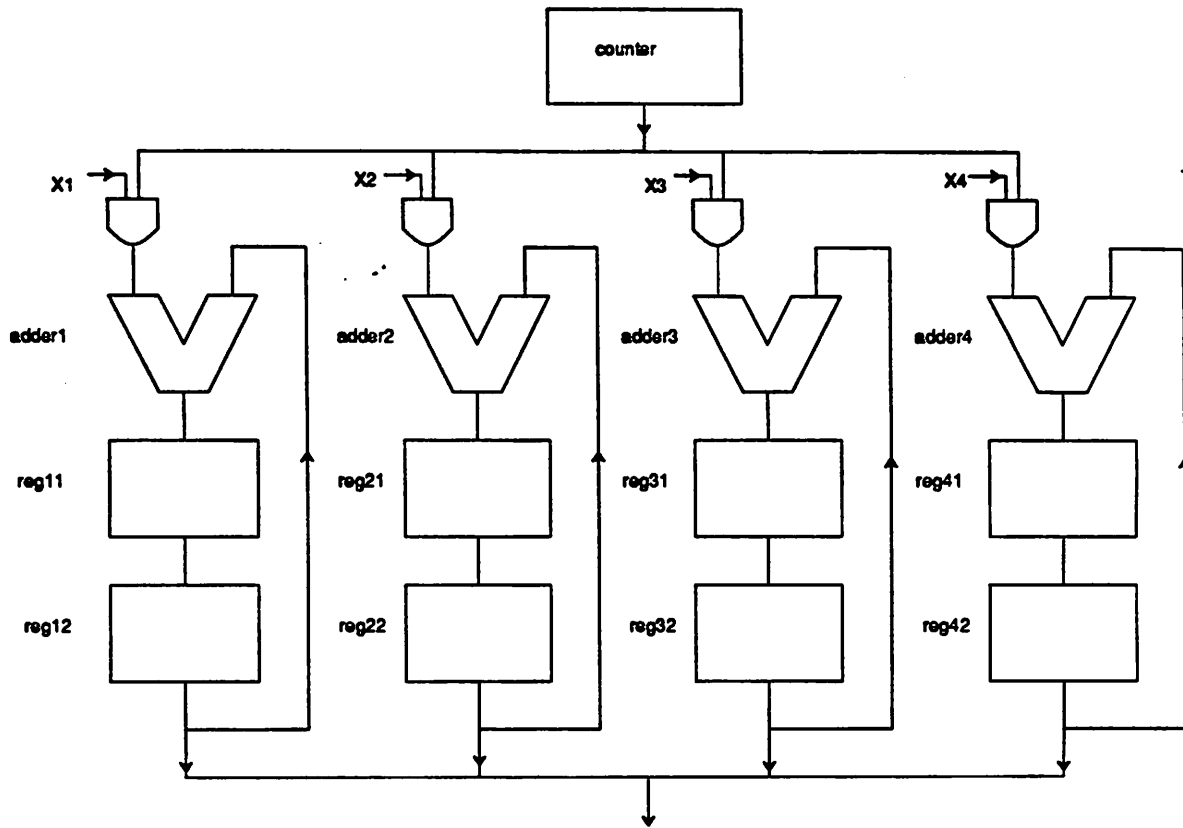


Fig. 4.9 Prototype FIR filter for a triangle window

performances compared. Adder1, as shown in fig. 4.12 is of the basic transmission gate type. The W/L of the transistors are optimised for the speed requirement and is shown in table 4.1

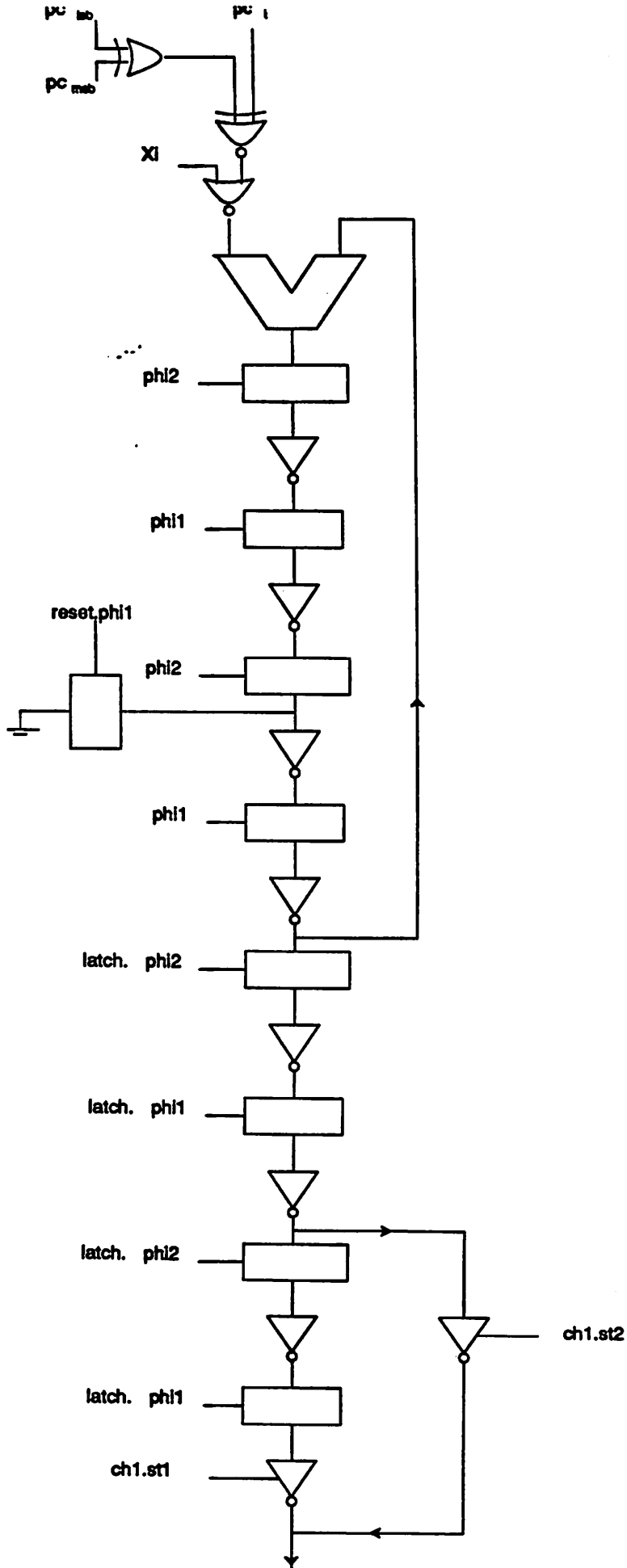


Fig. 4.10 Implementation of one-bit slice of the datapath

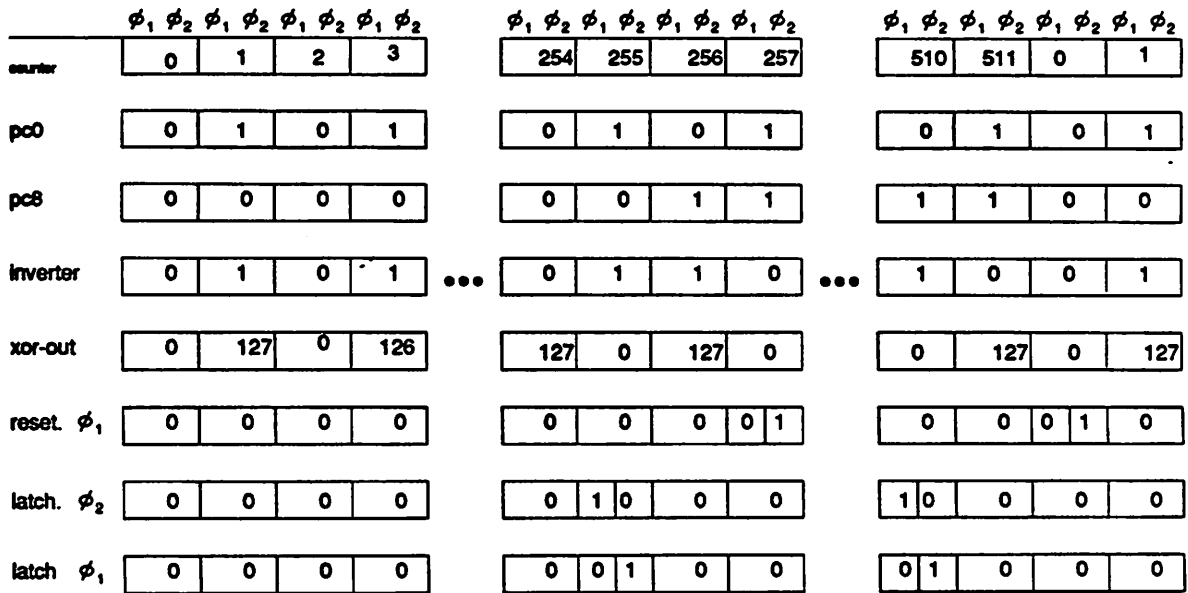


Fig. 4.11 timing for the FIR filter

Transistor	W/L	Transistor	W/L
M1	21/3	M2	9/3
M3	13.5/3	M4	10.5/3
M5	9/3	M6	27/3
M7	6/3	M8	10.5/3
M9	24/3	M10	10.5/3
M11	13.5/3	M12	27/3
M13	12/3	M14	13.5/3
M15-M24	6/3		

Table 4.1 W/L of AdderI

Spice simulation shows that this adder can run with a propagation delay of around 4.5ns per stage. As shown in the circuit diagram the major building block is the first circuit that generates that XOR functions. Using this the Sum and the Carry out function can be implemented. Ripple carry delay is determined by the gate loading of M2, M5, M9, M12 and drain capacitances of M11, M10 in normal cases. In worse

case when the transmission gate M11, M10 is on then the loading due to the drain source capacitances of M5, M2 as well as the gate capacitances of M7, M8 will also be present. The major advantage of this adder is it is smaller because it uses transmission gates. Because adders needed to be used a lot in the FIR filter, this design is selected to minimise area. However as can be seen the loading makes this adder slower than AdderII as described below. AdderII, as shown in fig.4.13 is an adder that minimises the loading of the carry chain. It is the adder designed for the cell library of the Berkeley Lager system. In addition it does not use any transmission gate in the feedback. The size of this adder is bigger than the AdderI but the propagation delay is smaller. This adder has been fabricated independently and the delay has been measured to be around 2.2ns per stage. The W/L of M5, M6 are 16.5/3, for M13, M15 are 21/3. For the rests the n transistors all have W/L=6/3 and the p transistors all have W/L=12/3.

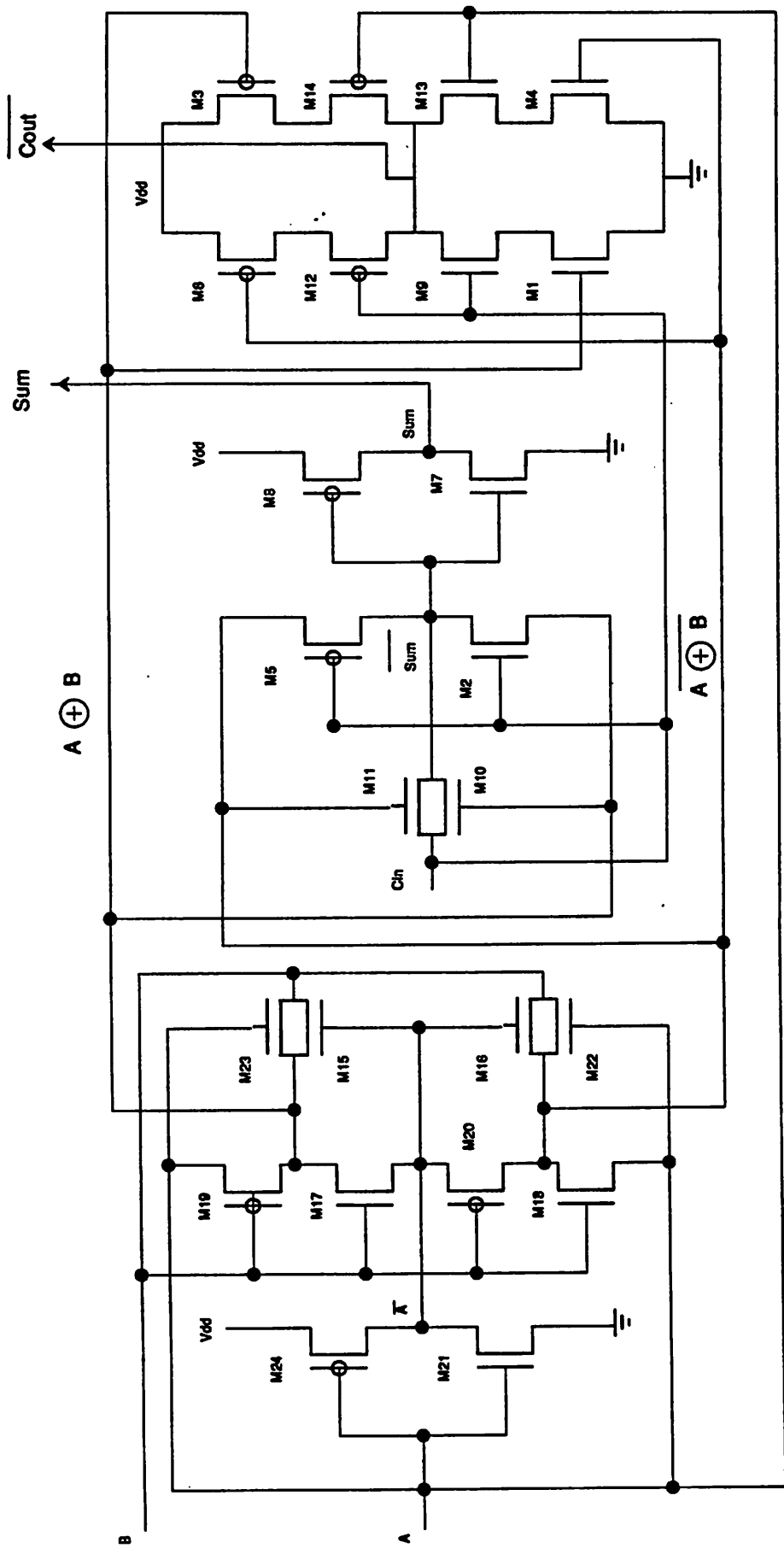


Fig.4.12 Adder1

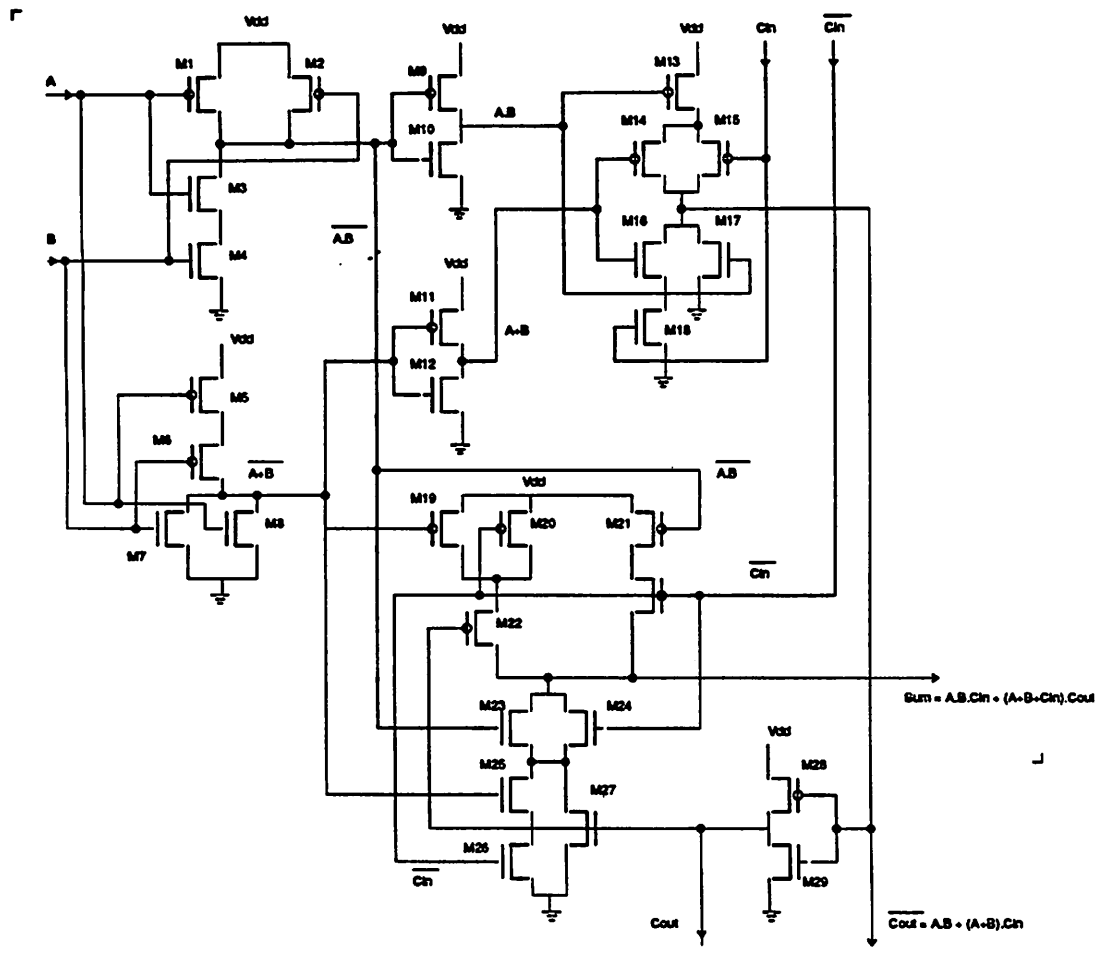


Fig.4.13 AdderII

The logical operation of this adder is as depicted on the diagram where the Sum and Cout is generated by two different parts of the circuitry. The major difference between this adder and the Adder1 is that the carry path is decoupled from the sum path and so no extra loading is present. Cin propagates through M15, M18 to get Cout. The carry propagation delay comes mainly from drain capacitances of M14, M15, M16, M17 as well as the gate capacitances from the next stage that corresponds to M15, M18.

4.2 Second stage filter design issues

For the second stage design we have to decide on the architecture of implementing both the FIR3 and IIR4 as shown in fig. 4.2. FIR3 is a five-taps FIR filter and IIR4 is an elliptic filter. To determine the structure of IIR4 a couple of options are available. For an IIR filter an elliptic filter can achieve the passband and stopband requirement in the lowest order. Furthermore since in the stopband there is attenuation from the previous FIR filters, so the requirement can be relaxed. In addition this filter has to satisfy the group delay specification for the coder. It has been found that a 4th-order elliptic filter will satisfy the filter response as shown in fig. 4.4. as well as the group delay requirement. Having decided on using the elliptic filter there is a choice between a cascaded structure and a wave-digital structure. A cascaded realization is shown in fig. 4.2.1 and a wave digital filter implementation is shown in fig. 4.2.2. The cascaded structure is simpler to implement but the wave-digital filter is less sensitive to rounding in the coefficients. However if the coefficients are expressed using CSD (canonical sign digit) and if the multiply is implemented using shift-and-add then it is the number of ones and minus-ones in the coefficients that counts, instead of the total number of bits in the coefficients. Comparing the two schemes using the above criterion it is found that the cascaded structure has about the same performance as the wave-digital filter and so the cascaded structure is selected.

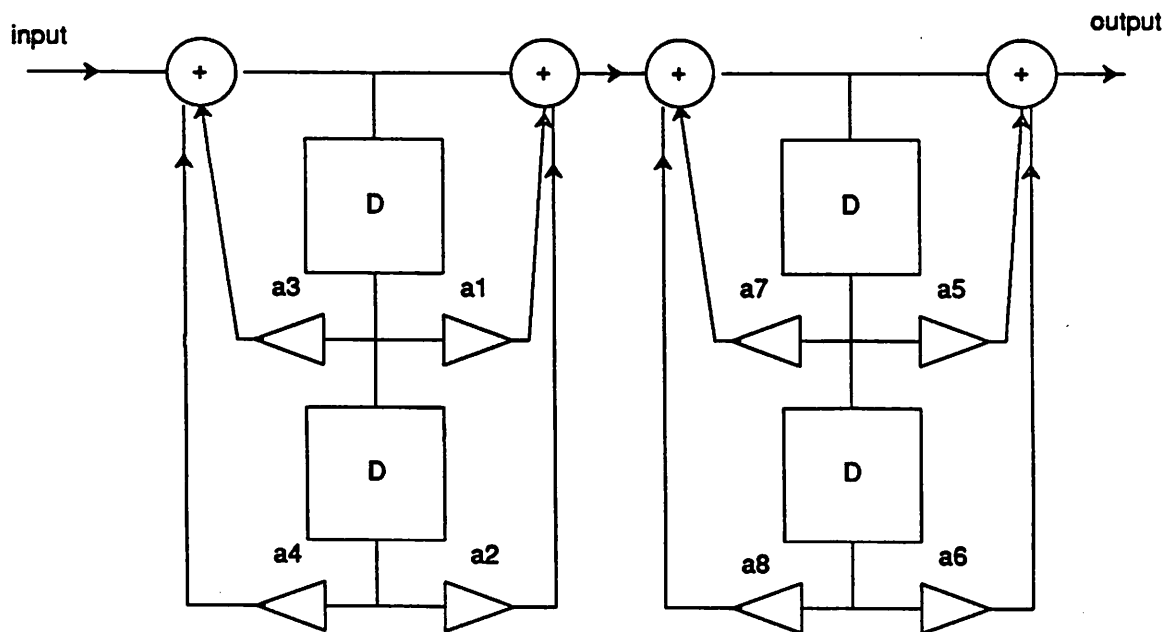


fig.4.2.1 cascaded structure

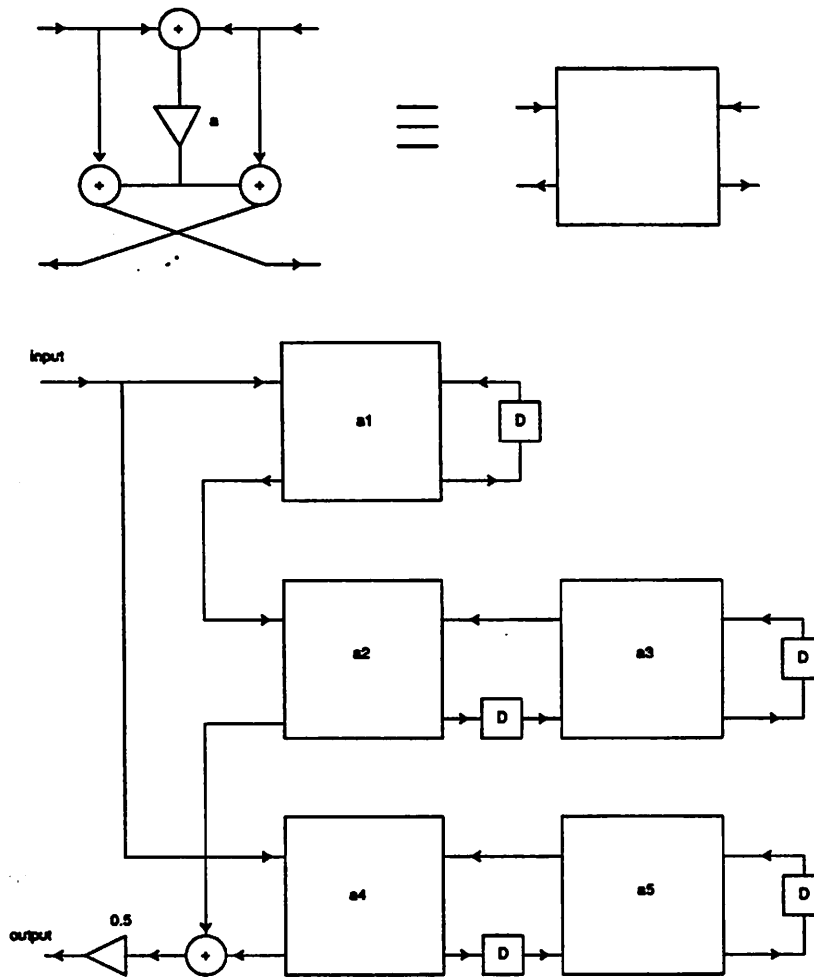


fig.4.2.2 wave-digital structure

4.2.1 Bit serial approach

The second stage design issue is concentrated on the filter architecture choice. Again the problem is one of minimising the area. The choices under consideration are bit serial,³ and microprogrammed approaches. It can be seen that for a multi-channel coder, the necessity to multiplex the second stage filter generates some interesting comparisons between the two approaches.

Description of the bit serial approach

As a simple example of this methodology, fig. 4.2.3 demonstrates the integration of a first-order recursive filter. Fig. 4.2.3a shows the block diagram of a first-order filter with a discrete coefficient $\alpha =$

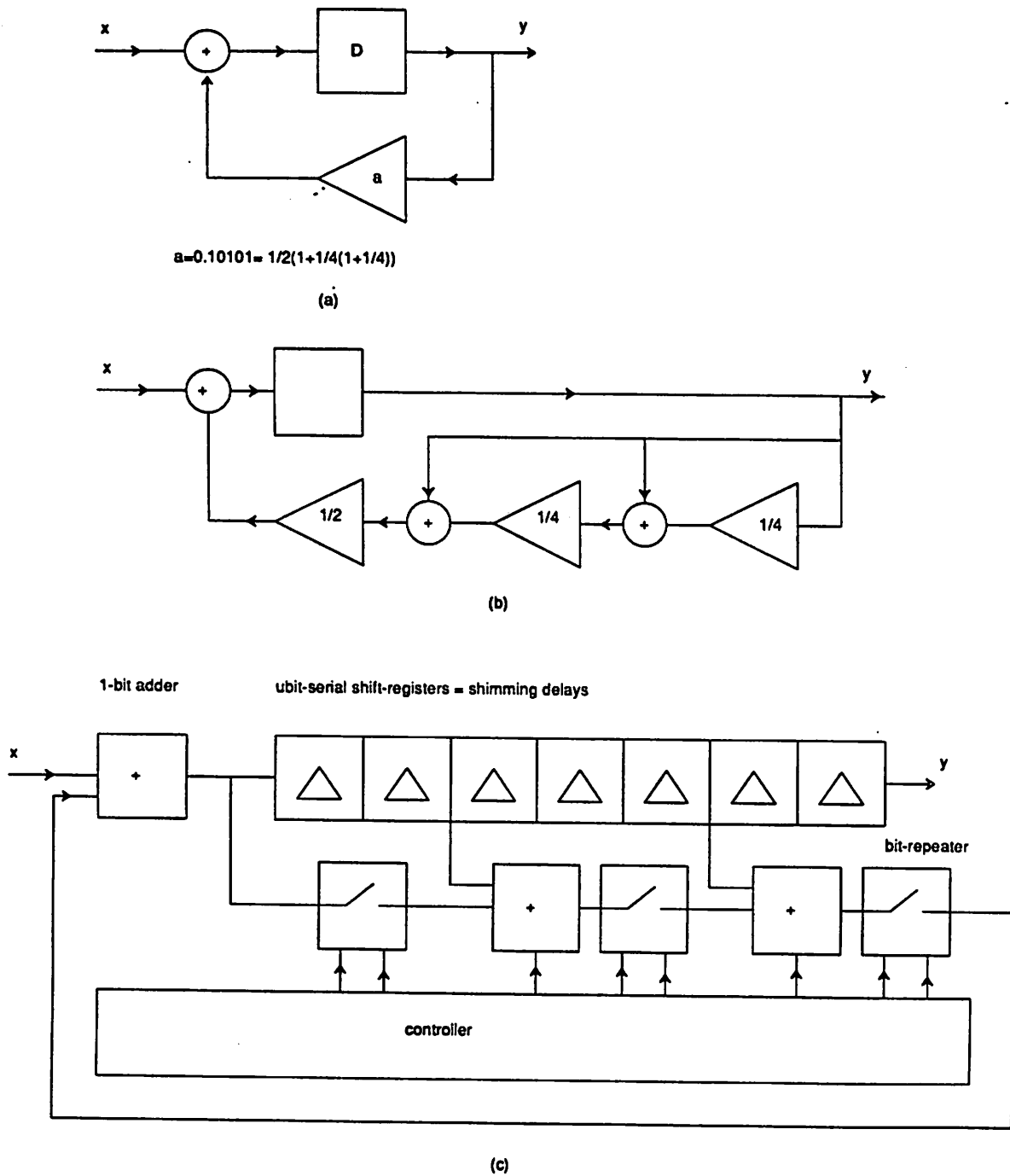


Fig.4.2.3 bit-serial architecture

0.10101. The signal word length is assumed to be 8 bits. The fixed nature of the coefficient is exploited by expanding the multiplication operation into power-of-two scaling (shift) and add operations as in fig. 4.2.3b. The synthesis of the expanded block diagram of fig. 4.2.3b into a bit-serial architecture is given in fig. 4.2.3c. The architecture is constructed by interconnecting 1-bit pipelined building blocks, such as 1-bit

adders, shift-registers, zero-injectors (used to perform a scaling with a negative power of two). A controller is added to the architecture to generate appropriate control signals to ensure correct synchronisation of these bit-serial building blocks. The circuit and layout descriptions of cells which implement the building blocks of the bit-serial architecture are stored in a predefined cell library.

Basically the bit serial approach runs at a speed that is independent of the complexity of the algorithm. However, the area is a direct function of the algorithm. So if the speed is not needed, such an approach will lead to a waste in terms of area. For bit-serial architectures, if F_c is the maximum clock rate at which the bits are propagated through the pipelined bit-serial building blocks (fig. 4.2.3c) and W is the signal word length then the maximum possible throughput is $\frac{F_c}{W}$. So for a given clock rate the throughput is inversely proportional to the number of bits representing the internal signals. On the other hand, the total number of shift register blocks required in the bit-serial architecture (fig. 4.2.3c) is directly proportional to the signal wordlength W . Hence a minimisation of signal word lengths maximises the throughput/unit area. The signal word length also determines the signal/noise ratio, dynamic range and relative amplitude of limit-cycle oscillations.

Disadvantages of the bit serial approach

Compared to the microprogrammed approach, the bit serial approach works at a higher speed but it suffers from two drawbacks: one is the fact that it uses shift registers rather than RAM cells as state variable storage elements and a shift register can take as much as 3 times more area than a 3T RAM cell when the amount of storage is large. In our case there is a need of a RAM of the size 40words*20bits which is roughly 1K bits. A comparison between the layout of a bit serial and a microprogrammed approach shows the difference. This is particularly true since the second stage filter is being multiplexed. In multiplexing, the datapath can be shared but not the storage for the state variables. For the microprogrammed approach the total amount of storage is roughly one-third the size of the total second stage filter. In a bit-serial approach this would increase the percentage to more than 60%. Furthermore in the second stage there is a need to implement both the FIR3 and the IIR4. A microprogrammed approach lends a lot of flexibility in implementing both types of filter while a bit serial approach precludes such options. In essence in a microprogrammed approach in order to implement an FIR and an IIR filter all that needs to be

done is to make sure that there are enough number of cycles to handle the computation. The area hardly increases(except for the amount for the state variables). In a bit-serial approach the hardware has to be increased everytime the filter complexity increases. Since our second stage filter can be implemented at 32 khz there is no need for the speed advantage.

4.2.2 Microprogrammed approach

The microprogrammed approach ⁴ is to encode the DSP algorithm into a microprogram and implement the program in an architecture that resembles an ordinary computer. In such it offers flexibility at the expense of speed. Unlike the bit serial approach, as long as the speed is adequate, the size of the hardware is independent of the complexity of the algorithm. However, the speed is now limited by the speed of the adder which in turn determines the clock speed, as well as the number of clock cycles available per input sample. This in turn is determined by the complexity of the algorithm. As long as the speed offered by a particular technology and a particular algorithm stays within the limit, no area penalty is paid. Also because we encode the algorithm as a general purpose program it is a lot easier to combine both the implementation of the FIR filter and the IIR filter together. Furthermore the decimation property of the FIR filter can be exploited to reduce the number of cycles needed.

Shown in fig. 4.2.4 is the basic architecture of the processor. The processor executes its microprogram (stored in ROM) once per sample interval. The PC(program counter) and ROM are collectively known as the control sequencer. The output of the control sequencer is a sequence of control words. The control word can be subdivided into a number of fields. One of these is the address offset field. It is this field that is used as input to the AAU (address arithmetic unit) macrocell. The AAU modifies the address offset to produce the effective address for the processor data memory (RAM). Viewed together, the ROM, PC and AAU present a stream of horizontal control and address words to the AUIO, RAM. The AUIO (arithmetic unit and I/O) consists of an arithmetic unit and an I/O interface. The arithmetic unit together with the RAM (data memory) form the signal data path. This is where all operations on signals are performed. This data path may be microcoded to perform fixed coefficient multiplies, add, subtract and accumulate operations.

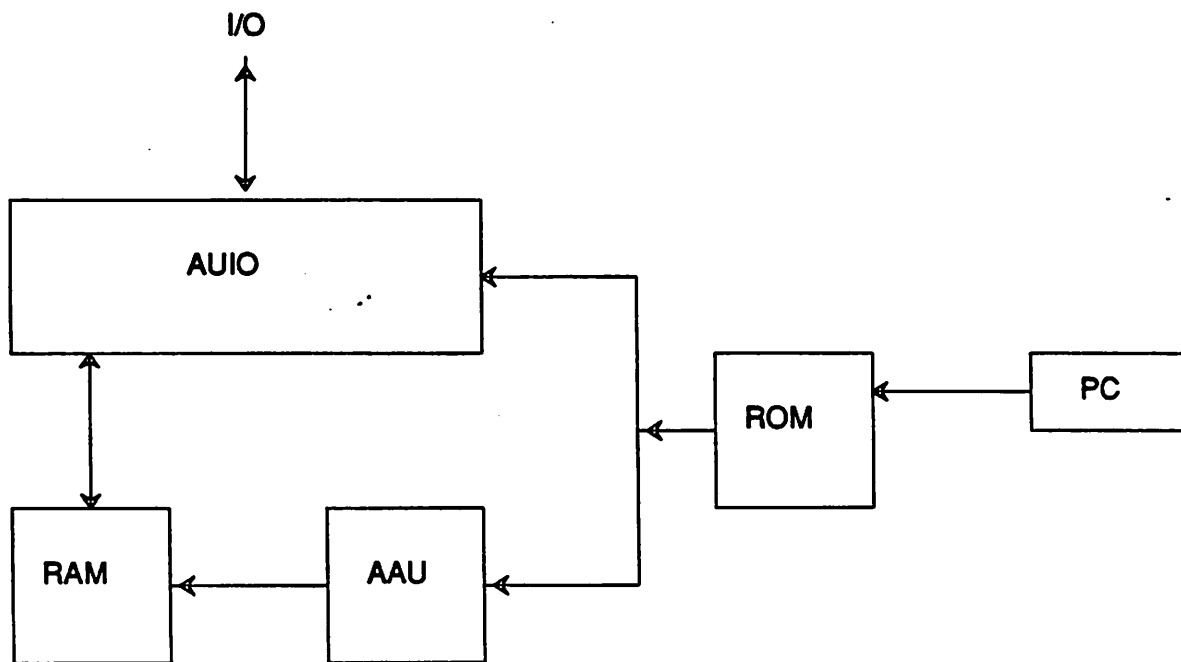


Fig. 4.2.4 microprogrammed architecture

The processor assembly executes its microprogram once per sample interval. Since primitive operations such as multiplies must be microcoded (due to the absence of a hardware multiplier which will take a lot of area) it follows that the sample rate must be substantially slower than the processor's clock rate. In order to make the clock rate as fast as possible heavy pipelining on the data-path is used. This is possible because in the filter implementation the execution sequence is known ahead of time and very few branches, if any occur. The pipelined datapath is shown in fig. 4.2.5. It can be seen that heavy pipelining is made possible by the MOR (memory output register), SOR (shift output register) and ACC (accumulator). With this pipelining the speed limitation comes from within the stages and the critical path will be limited by the adder stage. In a 3 μ CMOS technology, for an adder of 20bits, clock rate in excess of 8Mhz can be expected. That means for an input sampling rate of 16 kHz around 500 operations can be performed. If this is time shared between between 4 channels, then each channel will get around 128 cycles. Most of these cycles go into shift and add operations to implement a multiply and so the number of ones in the coefficients of the FIR filter and IIR filter will determine the number of cycles required. In actual coding, fewer than 64 cycles are needed for our application and we are well within the capability offered by this approach. For more general applications the ratio of the clock rate to sample rate is an important parameter.

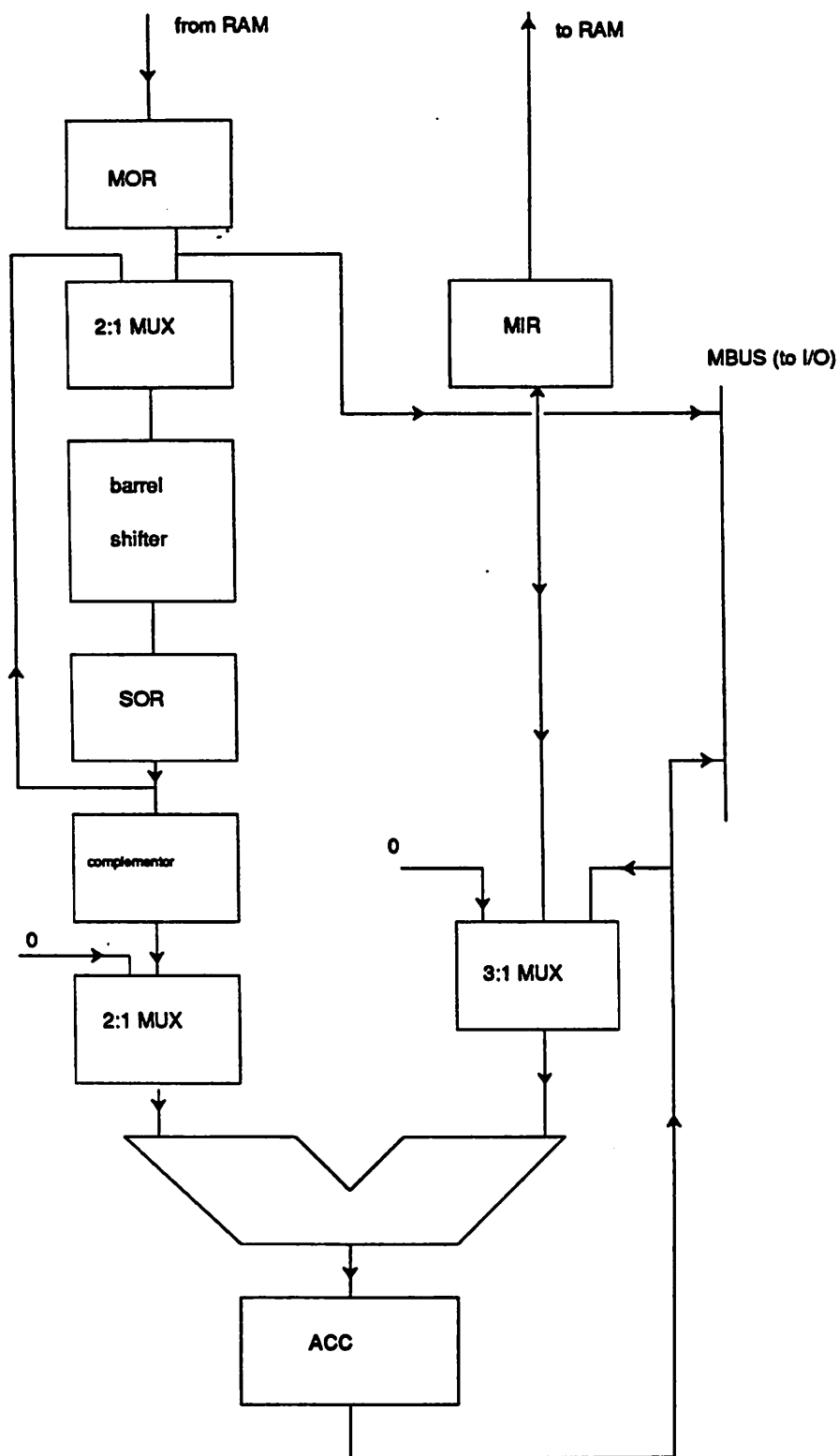


Fig. 4.2.5 pipelined datapath

If this ratio is too small, then there would not be time for a significant amount of processing during the sample interval. On the other hand, if the ratio of clock rate to sample rate becomes very large, the architecture exhibits an imbalance wherein the control ROM consumes nearly all of the silicon area, and the datapath only a small fraction. This situation is aggravated since the horizontal control words are not densely encoded.

4.2.3 Lager System

The microprogram approach that is used here is based on a silicon compiler system called the Lager system⁵ developed in Berkeley. The silicon compiler system, being general purpose does not produce the most area efficient solution for our application. Hence several modifications have been introduced to customise the design.

The Lager system is an approach which uses large, parameterized blocks of circuitry called macrocells. These macrocells differ from standard-cells in that they are large. Also standard-cell designs consist of regular rows of cells separated by wiring channels. Macrocells are assembled from a cell library by forming a two-dimensional array of cells. This leads to macrocells which are themselves very dense. The Lager design systems involve considerable supporting software for simulation and layout generation. There are basically the following software packages:

LAGER (LAYout GenerAtor)

The automatic layout system used to develop the digital signal processing circuits described in this paper. It includes a logic simulator, a design compiler, a module generator, and a place and route tool.

Demon

The logic simulator, used to test a control algorithm before it is ready for the physical layout.

Archer

The design compiler. This tool takes a valid design file, tested previously, and generates a parametric description of every macrocell which will be required by the layout.

TimLager

A tool which generates the macrocell from its parametric description. It requires a set of leafcells and a module-specific subroutine before it can tile a macrocell. TimLager creates both the physical layout and a hardware descriptor file used by the place and route tool.

Flint

The place and route tool used on macrocells generated by TimLager.

To implement the filter a design file has to be written by the designer. This file is an assembly language type program that instructs the hardware to implement the filter function as shown in the appendix. The design file is the input we need to pass to the program Archer. The output, ideally is a layout. There are however various modifications that are added to optimise for area. The details are described in⁶

4.2.5 Modifications to the Address arithmetic unit

Since there is a need to take input from four channels, one should use the sample counter in the AAU to index the memory for the correct state variables. Furthermore, since each channel takes two inputs from the first stage FIR filter, the AAU can be used to do proper indexing. However there are two reasons why this is not done: One, is the fact area should be minimised. Since the four channels have the same number of cycles the upper two bits of the program counter can be used to distinguish which channel is being executed. In order that the correct RAM locations will get accessed, special care has to be taken in the design file. In other words, when the locals are declared, the RAM locations have to be put in the order of the channel. Secondly, the timing of the data coming from the FIR2 makes it complicated to do simple indexing.⁶ Consequently special hardware buffers are built to match the timing requirement between the FIR2 and FIR3.

4.3 Analog Front End Design

The function of the analog front end is to implement the $H(z)$ function as discussed in chapter 2.10.

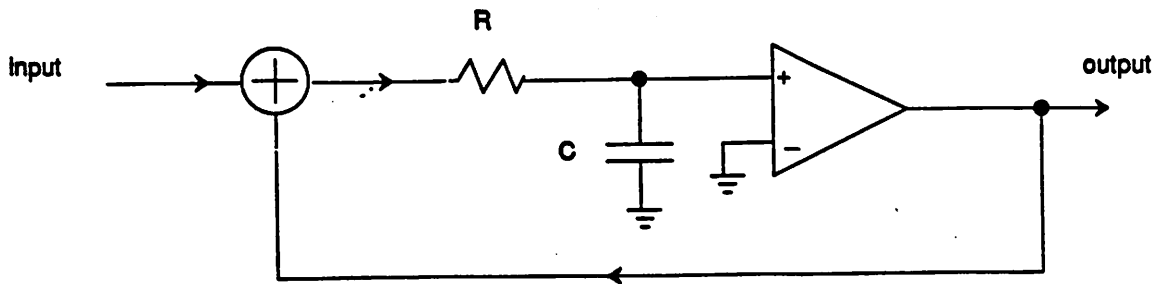


Fig. 4.3.1 RC implementation

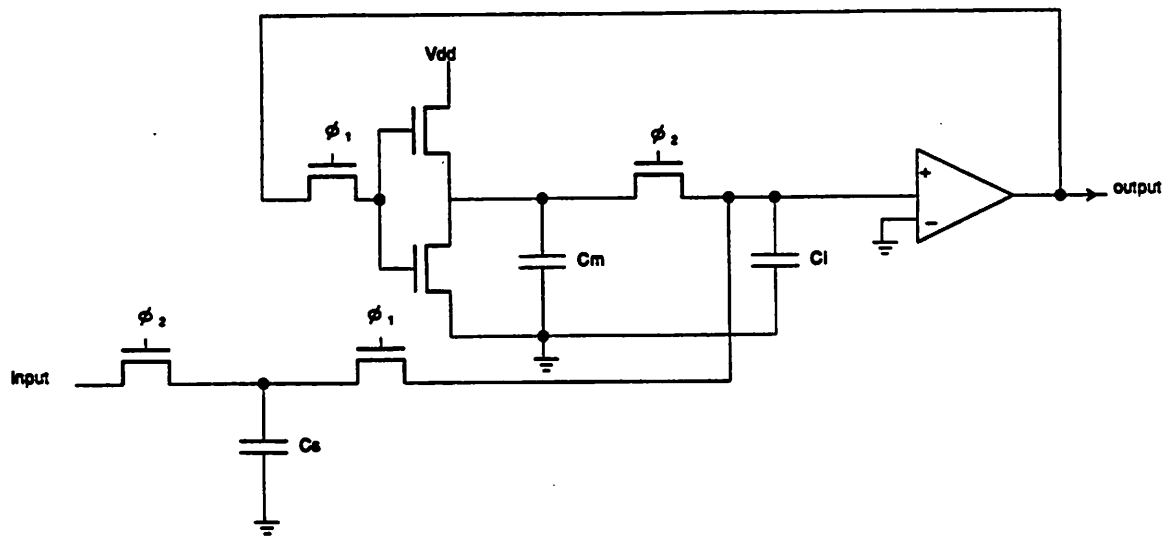


Fig. 4.3.2 Resistor implemented as switched capacitor

4.3.1 Alternative Analog Architectures

There are a variety of ways in which one can implement the low-pass function. The first way is to use an RC type implementation⁷ as shown in fig. 4.3.1. Another approach is to use the switched capacitor principle to simulate the resistor.⁸ as in fig. 4.3.2. This will be especially beneficial in a technology in which high gain opamp cannot be built easily, for example the NMOS technology. The problem here is that the capacitor ratios $\frac{C_i}{C_m}$ and $\frac{C_i}{C_s}$ are large, and are on the order of 400 for the voiceband application.

Both approaches however suffer from the lack of having an opamp and hence a summing node which will reduce the amount of noise injected from the substrate. This is especially important in a multi-channel coder where crosstalk from other channels as well as digital switching noise from the digital filter will appear at the same terminal as the input signal. Furthermore in both approaches the low-pass filter does not have gain which means the comparator has to provide large gain. This is due to the fact that at half the sampling frequency $\frac{f_s}{2}$ and at idle channel condition the output should be the idle pattern: 1, -1, 1, -1... . However since the RC time constant has a rolloff of 6db per octave, for a 5V supply the idle pattern is reduced to around 10mv input and the comparator has to be able to resolve this signal at 4Mhz.

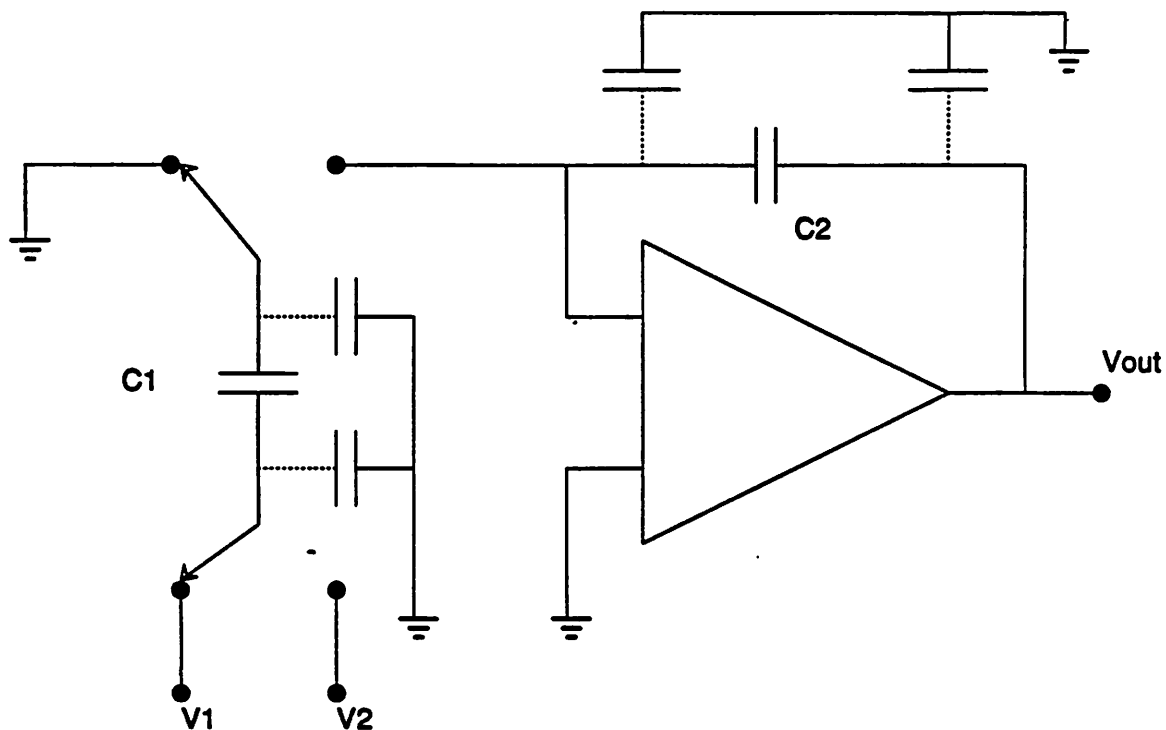


Fig. 4.3.3 switched capacitor integrator architecture

Prototype architecture

The prototype uses a MOS switched-capacitor integrator. The configuration as shown in fig. 4.3.3 is inherently insensitive to stray capacitances associated with the signal capacitors C_1 and C_2 . In a digital process, the capacitor exhibits a large parasitic capacitance from the bottom plate that may even considerably

exceed the value of the signal-path capacitor. Note that the ratio-accuracy of $\frac{C_1}{C_2}$ does not directly affect the converter accuracy.

4.3.2 Overall Analog requirements and solutions

Crosstalk

Since four analog front ends needed to be put on the same chip, there is a certain amount of crosstalk between different channels. The channel to channel crosstalk specification for a coder is less than 75db for a 1 kHz sine-wave. The crosstalk comes from different sources: power supply, substrate, ground, and any component that is shared. To minimise the crosstalk the following methods have been used in the prototype. One is to separate the voltage reference and bias generator between different channel. This increases the chip area but can achieve better isolation. The other is to use a fully differential architecture. In a fully differential architecture any crosstalk that is injected will become a common-mode signal if the matching is ideal.

Power supply rejection

This is another key element of design since there is so much digital circuitry sharing the same substrate with the analog circuit. In the prototype a fully differential architecture is used to help alleviate the situation. Another important factor is the bandwidth of the opamp. Power noise can be coupled to the output node of the opamp if the output impedance is large. At low frequency due to the high loop gain the output impedance is reduced by a lot. At high frequency the gain of the opamp drops and the output impedance increases. That means the coupling is more severe. Consequently an opamp has to have gain-bandwidth product at least several times larger than the frequency at which the digital circuit is working at. For the digital filter the FIR filter is working at 8 Mhz and the IIR filter at 4 Mhz. Therefore the opamp is designed to have gain-bandwidth product in excess of 8MHz.

Another source of power noise is from the output drivers. Since the drivers have to drive off-chip load, the switching current is large. One way in which the oversampling technique can be exploited is to arrange for the output pad drivers to switch at the downsampling frequency, as that is the time when the output is needed. Another improvement is by selecting the clock cycle in which the output pad driver

switches (ϕ_1) to be different from the clock cycle in which the opamp needs to settle (ϕ_2).

4.3.3 Requirements on the operational amplifier

Offset

Ideally in a coder environment the signal encoded is speech and so the dc offset is not significant. In fact, usually a coder is biased at a particular dc level by an auto-zero feedback loop to minimise the idle channel noise. The offset of the integrator is going to be the offset of the opamp modified by the ratio of the input capacitor to feedback capacitor. For a single-ended architecture the offset of an opamp comes from the following sources: a) mismatch between V_{ref+} and V_{ref-} . In other words, if the input is 0, the code should be 1 -1 1 -1 ...etc. However due to mismatch in V_{ref+} and V_{ref-} the output can be 0.9 -1 0.9 -1 ... Therefore there will be a dc offset problem. b) mismatch in input transistor sizes c) mismatch in input transistor bias current due to voltage drop across power and ground d) mismatch in charge injection due to mismatch in sampling capacitor and sampling switches. In a fully differential scheme, the mismatch between V_{ref+} and V_{ref-} will be cancelled as is shown in section 4.3.4. Offset can be minimized by careful layout and careful choice of input transistor sizes.

Gain

As far as the effect of the gain on the quantization noise is concerned, the main effect of the finite gain of the opamp is to modify the noise-shape of the integrator as shown in fig. 4.3.4. In other words the effect of the finite gain is to modify the frequency response of an integrator from having a pole at dc to having a pole at some finite frequency. Moreover the dc gain of an integrator is no longer infinite, but it is equal to A, the opamp gain. Since the noise transfer function is $\frac{1}{(1+H)}$ where H=integrator transfer function, the noise transfer function for large H is just 1/H and is as shown in fig. 4.3.4. Now the effect of this modification is to admit more inband noise as shown. Both from simulation and from calculation one can show that the opamp gain needs to be about the oversampling ratio. That this is the case can be explained by referring to fig.4.3.4. From the fig it can be seen that the response with finite opamp gain has the zero moved from dc to some frequency. If that frequency is f_c it can be ascertained how much extra noise is

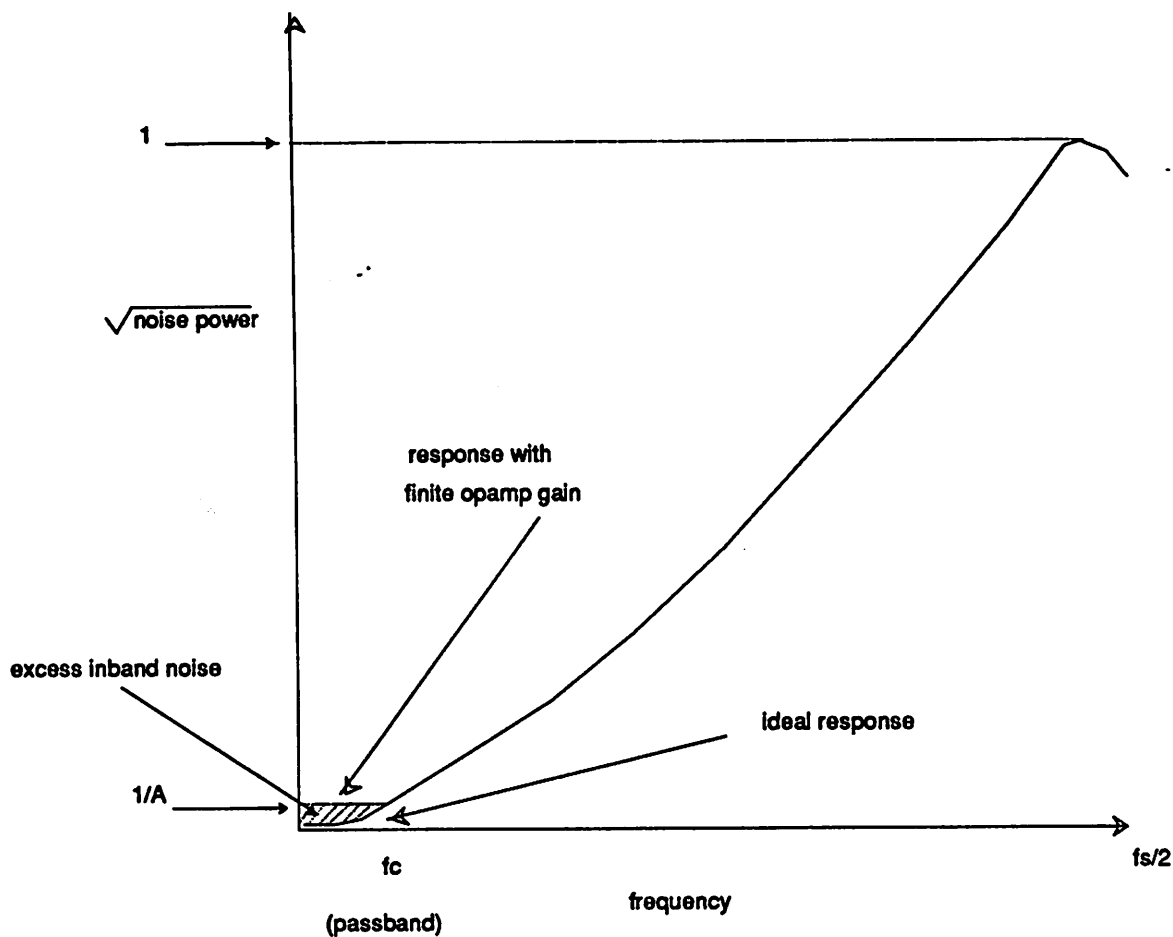


Fig. 4.3.4 Effect of finite opamp gain

introduced. Given that the spectral density of a first order coder is a function the frequency f , therefore with ideal response the inband noise = $1/3(fc^3)$. Whereas with modified response we have inband noise = fc^3 and the signal to noise ratio drops down by 4.8db. If this is defined as the acceptable loss then the zero should be placed at fc . Now using the geometric property of similar triangle it can be shown that the opamp gain $A = \frac{f_s}{2f_c} = \frac{D}{2}$. For the prototype the oversampling ratio is 512. Based on the above formulae together with a safety margin it is determined that a gain of 1000 is needed.

Harmonic distortion

For the second and third harmonic the requirement is 40db. The harmonic distortion comes from the following sources: a) signal dependent charge injection. b) capacitor voltage coefficient c) opamp non-linear gain. To eliminate error coming from the first source, a switching scheme similar to ⁹ is used. That means in fig. 4.3.5 $\phi_1, \phi_{1b}, \phi_2, \phi_{2b}$ are delayed by 2 inverter delays from $\phi_1, \phi_{1b}, \phi_2, \phi_{2b}$

respectively. For capacitor voltage coefficient, the important capacitor is the sampling capacitor. The feedback capacitor can have a large voltage coefficient because the D/A is just one-bit and so the voltage across the capacitor is constant. Simulation shows that for harmonic distortion less than 80db the voltage coefficient can be as high as 100ppm which is larger than that of metal-poly capacitors in most processes. Finally the nonlinearity of the opamp can be minimised by operating the opamp in a small voltage swing.

This can be accomplished easily by adjusting the ratio $\frac{C_i}{C_f}$ as shown in fig. 4.3.3.

Noise

Achievable dynamic range in monolithic A/D circuits is constrained by the available signal swing at one extreme and noise sources at the other. Noise can arise from the power supply or substrate coupling as discussed above, clock signal feedthrough, and from thermal and 1/f noise generation in the MOS devices. The components of importance for circuit noise are the opamp and the MOS analog switches.

A lower bound on quantization noise in sigma-delta converters arises from low-frequency components of induced power-supply and substrate noise, primarily through the opamp; 1/f noise and thermal noise in the input MOS transistors. 1/f noise is not a problem in the circuit because large input devices are needed for the speed otherwise. Thermal noise is a more fundamental problem. Because of thermal noise in the switch resistances, each voltage sampled onto a capacitance C exhibits an uncertainty of variance kT/C where k is Boltzmann's constant and T is absolute temperature. ¹⁰ For an input circuit like fig. 4.3.3, the input capacitor sees two switch paths per clock cycle, for a noise variance of $2kT/C$. The noise sources in the switches and opamp actually interact due to bandwidth effects, but it has been established that a lower bound for the combined opamp and switch thermal sources in this configuration is $2kT/C$. If D is the decimation ratio, the (uncorrelated) thermal noise variance falls by a similar amount. The maximum RMS

value of a sinusoidal signal, within a power supply voltage V_s , is $V_s/(2\sqrt{2})$. Combining these factors gives a maximum signal-to-thermal-noise ratio for an oversampling front end based on fig. 4.3.3, expressed as an amplitude ratio, of

$$SNR_{\max} = \frac{V_{dd}}{4} \sqrt{DC/kT}$$

where V_{dd} is the total power-supply voltage, D is the oversampling ratio, and C is the magnitude of the input sampling capacitor.

With a 5-volt power supply, $D=512$ and 0.5pf capacitors, SNR_{\max} corresponds to about 17 bits. This is more than enough for our present application which requires a SNR_{\max} of around 13 bits.

Settling time

Depending on the model of settling, different analytical results can be obtained to predict what the requirement is. If the settling is linear, meaning that the output voltage rises in an exponential fashion and never has ringing, then the effect is one of changing the integrator response and consequently the noise shaping response. Then it has been shown that at least for an LDI type integrator the transfer function of the integrator is changed so that under moderate and large settling error the integrator appear very lossy as if the opamp has a very low gain. Under such a condition computer simulation has established that settling error on the order of 10% of the full scale is tolerable¹⁰ If the model of the settling is such that the output rings and the error is random from sample to sample, then one can model the settling error as a random noise, just like the model of the quantization noise. In this case since the noise source follows the integrator, it is subjected to noise shaping and again the settling requirement will be very loose. However because there is slewing in the opamp the settling error will have some correlation with the input signal. If the opamp settling error is referred back to the input like signal, that is if everytime the opamp settles the error is made so that it is always accumulated, then it can be shown that a 13bit accuracy requires the settling error to be less than 13bit. That is because now the opamp settling error is referred back to the input and any error it makes is identical to an input error. It is this approach that has been chosen in the design of the prototype. With a 4Mhz clock and the clock cycle divided into ϕ_1 and ϕ_2 only half the clock cycle is available for the opamp to settle. That is the opamp has to settle to within 0.01% in 125ns.

4.3.4 Prototype Opamp

As discussed above the opamp used will have the following properties:

- 1) fast settling: 0.01% in 125ns
- 2) medium gain: 1000

In addition the crosstalk between opamp has to be low and the power supply rejection has to be good. Given the requirement and the choice of a 3u CMOS single poly, double metal technology, the opamp chosen is a fully differential, single stage class A opamp with a folded cascode¹¹ output stage. It uses dynamic common mode feedback¹² for establishing the common mode output voltage.

Integrator

The overall integrator is as shown in fig. 4.3.5. It is a bottom-plate integrator which minimises the effect of the parasitics. Note that there is a separate input for the dither signal. Because of the full differential arrangement any mismatch between V_{ref+} and V_{ref-} will be cancelled out. For example if V_{ref+} is 1.1 instead of 1 and V_{ref-} is -1, and if we use single ended approach and let us apply a 0 input the output will be 1.1, -1, 1.1 -1 ... and the dc value is 0.05. Now in a fully differential approach 1 means $V_{ref+} - V_{ref-} = 2.1$ and a -1 means $-V_{ref+} - V_{ref-} = -2.1$. so the average value of 2.1, -2.1, 2.1, -2.1 is still zero and there is no offset. All the n switches have W/L of 13.5/3 and all p switches have W/L of 30/3. Selection of the sizes of the switches and sampling capacitor C_1 and C_2 is a trade-off between the speed of the RC time constant and the amount of charge injection. Larger switch means faster sampling, but also means more clock feedthrough due to overlap capacitance. Furthermore this also means more charge injection during the turn-off of the switches, due to the flow of the channel charge. Note that the amount of charge injection depends on the impedance seen on the drain or source side of the switch and hence on the voltage difference. Because the input signal is different from cycle to cycle the source voltage will be different. On the other hand the drain voltage is kept at virtual ground and so the amount of charge injection will be different from cycle to cycle. This means the charge injection is signal dependent and will cause harmonic distortion. To remedy the situation another clock scheme is introduced. In addition to the original ϕ_1, ϕ_2 clock now we have ϕ_1', ϕ_2' clock which are derived from the original clocks by delaying them through two invert-

ers as shown in fig. 4.3.5. The effect of this clock scheme is to make sure that the switch which is connected to ground or virtual ground is switched off first. Hence the amount of charge injection, which depends on the voltage of the turnoff side, will always be constant. This will introduce an offset but no harmonic distortion. When the other side of the switch is turned off subsequently, it is like pumping charge into a floating capacitor and so no charge injection occurs.

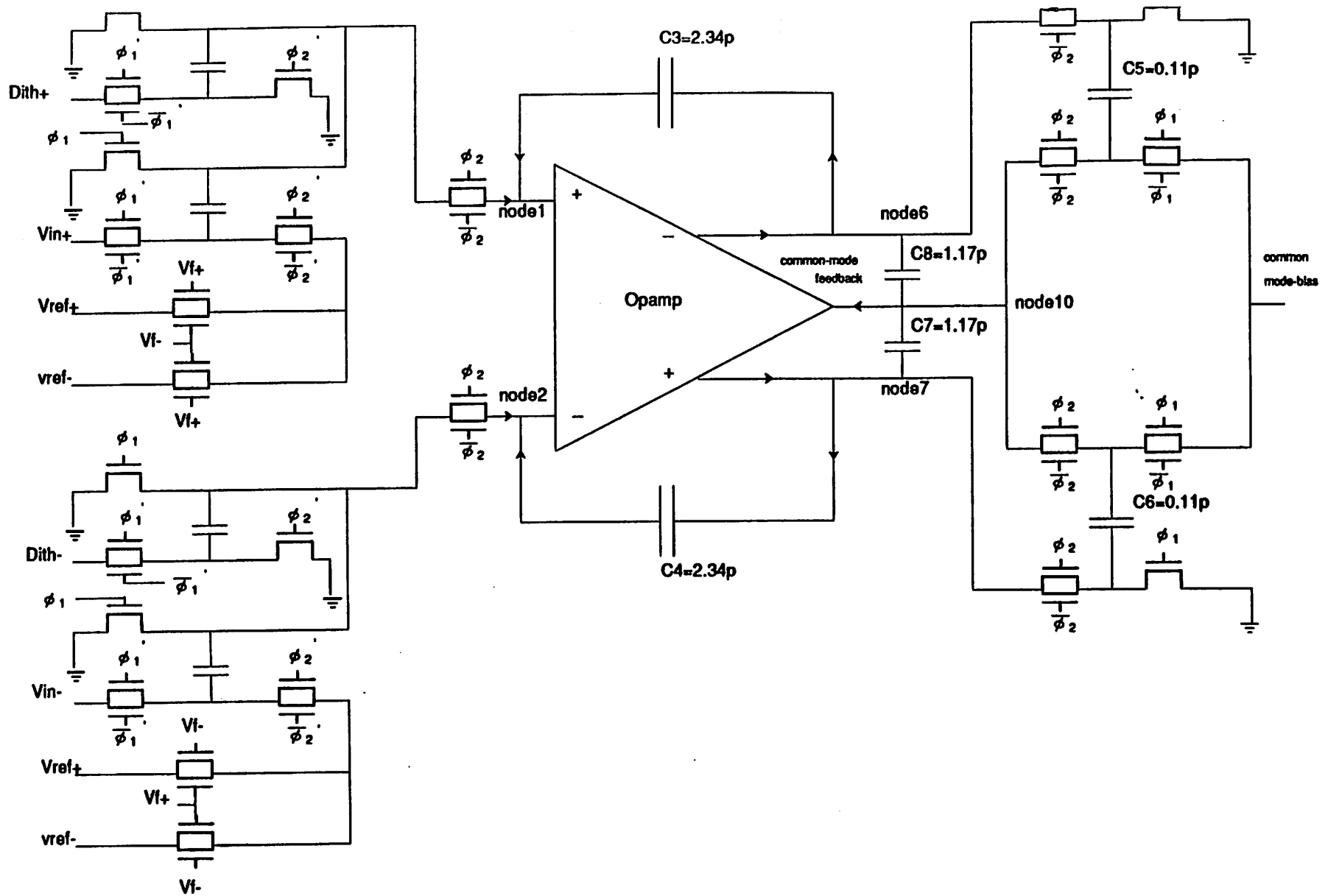


Fig.4.3.5 Prototype switch capacitor integrator

Transistor	W/L	Transistor	W/L
MD1	338/4.5	MD2	338/4.5
MD3	338/4.5	MD4	338/4.5
MD5	138/4.5	MD6	138/4.5
MD7	270/4.5	MD8	270/4.5
MD9	650/3	MD10	650/3
MC1	28/4.5	MC2	28/4.5
MC4	62/4.5	MC5	62/4.5
MC6	28/4.5	MC7	28/4.5
MC8	62/4.5	MC9	62/4.5
MC10	675/4.5		

Table 4.3.1 Device sizes for opamp

Opamp

Shown in fig. 4.3.6 is the circuit schematic of the opamp. The device sizes are as shown in table 4.3.1 The operation of the opamp is divided into the differential amplifier and the common mode feedback amplifier. The differential amplifier is a single stage amplifier with a folded cascode output branch. If we look at one branch of the amplifier it will be consisted of MC10, MD9, MD7, MD5, MD3, MD1. This half is like common source, common gate amplifier with the input coming in from MD9, and output from MD5. MD1, MD3, MD7 and MC10 serve as current sources. This is basically a current steering amplifier with the input voltage steering the amount of current flowing in the two branches. The output impedance of the double cascode branch is $(G_m R_{out}) \frac{R_{out}}{2}$, where R_{out} is inversely proportional to the drain current and increases with the device channel length while G_m is proportional to the square root of the current.

$$\text{voltage-gain} = G_{m_{MD9}} G_{m_{MD5}} R_{out_{MD9}} R_{out_{MD5}}$$

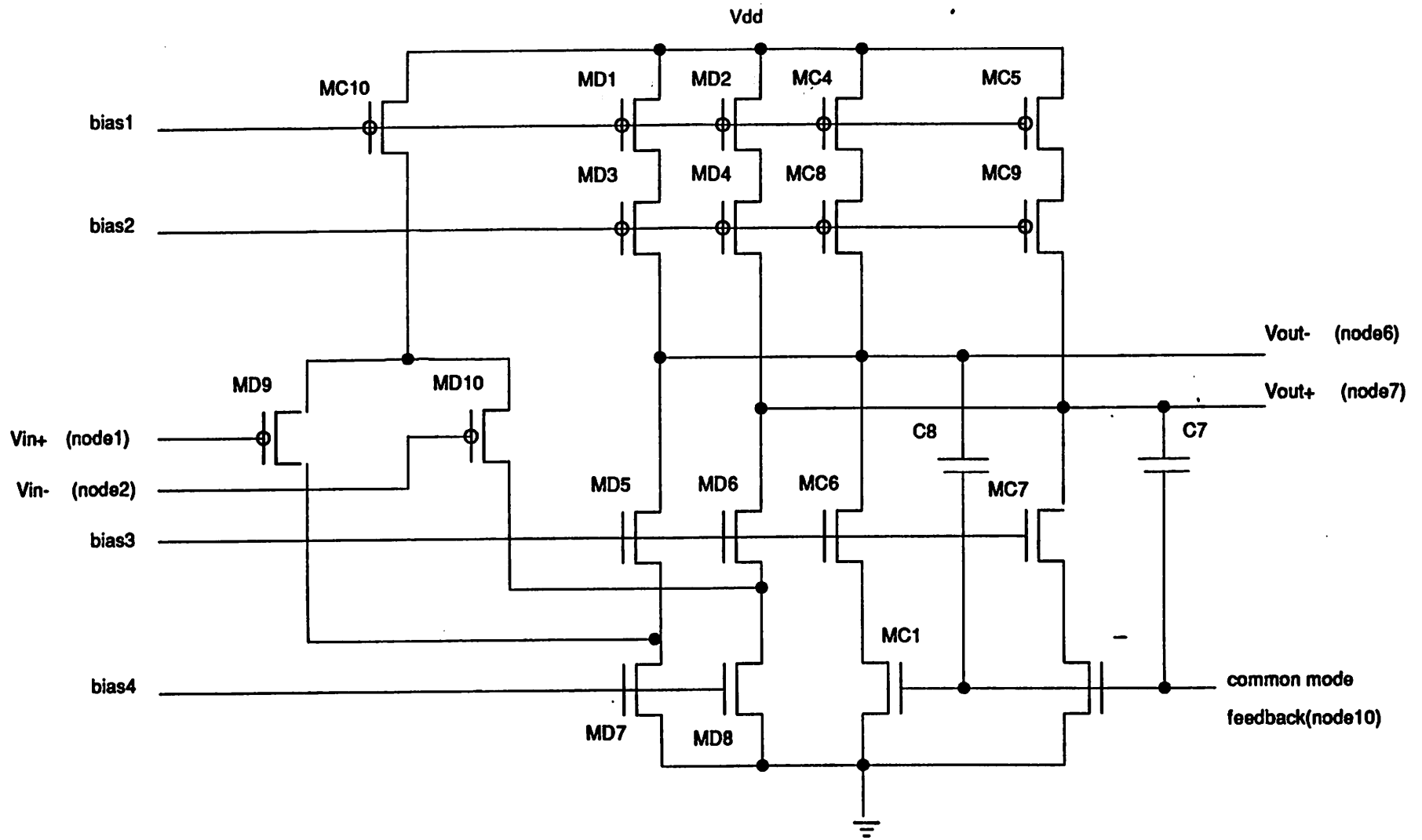


Fig. 4.3.6 Prototype folded cascode opamp

The high impedance node of the opamp is at the output node, therefore the load capacitance is used for compensation and no extra compensation capacitor is needed. The first non-dominant pole of the opamp is determined by $\frac{G_{m_{MD5}}}{C_p}$ where C_p is the total capacitance present at the cascode node and is comprised of the gate-to-source capacitance C_{gs} of the cascode device MD5, the sum of the junction capacitance C_j associated with the diffusions of the driver MD9, the n channel current source MD7 and the cascode device MD5. For a fixed C_p the non-dominant pole is proportional to $G_{m_{MD5}}$. This in turn decides how fast the opamp can settle. However since R_{out} is proportional to L, so the gain is also proportional to L. Major design effort is then used to tradeoff the gain versus the settling time by proper selection of the bias current and device sizes.

Shown in fig. 4.3.7 is a spice output of the frequency response of the opamp. It shows that opamp has a gain of 60db or 1000. The unity gain bandwidth is around 35Mhz. However due to the large sizes of the input transistors, the input capacitance is not negligible compared to the feedback capacitor and so the feedback factor is less than one, and we get only around 27 Mhz of bandwidth. Shown in fig. 4.3.8 is a spice output of the transient response of the whole integrator. It can be seen that the opamp settles to 0.01% by around 80ns.

Common mode feedback

The common mode feedback circuit works as follows: As shown in fig. 4.3.5 the ac voltage at node10(common-mode feedback) is given by the average of the output voltages, V_{out+} and V_{out-} . If The feedback capacitor C_7 and C_8 matches perfectly and if the parasitic capacitance at node10 is small, the the common-mode portion of the output signal is transmitted to to node10 unchanged while the differential portion has no effect. From node10 to the common mode output there is a negative gain whose gain is comparable to the forward gain of the amplifier. On the other hand the gain from node10 to the differential output is ideally zero. The loop gain is therefore very large and negative for any common mode signal while it is extremely small for any differential signal. This means that the common mode output voltage is kept at an almost constant value even in the presence of common mode output signal and at the same time the

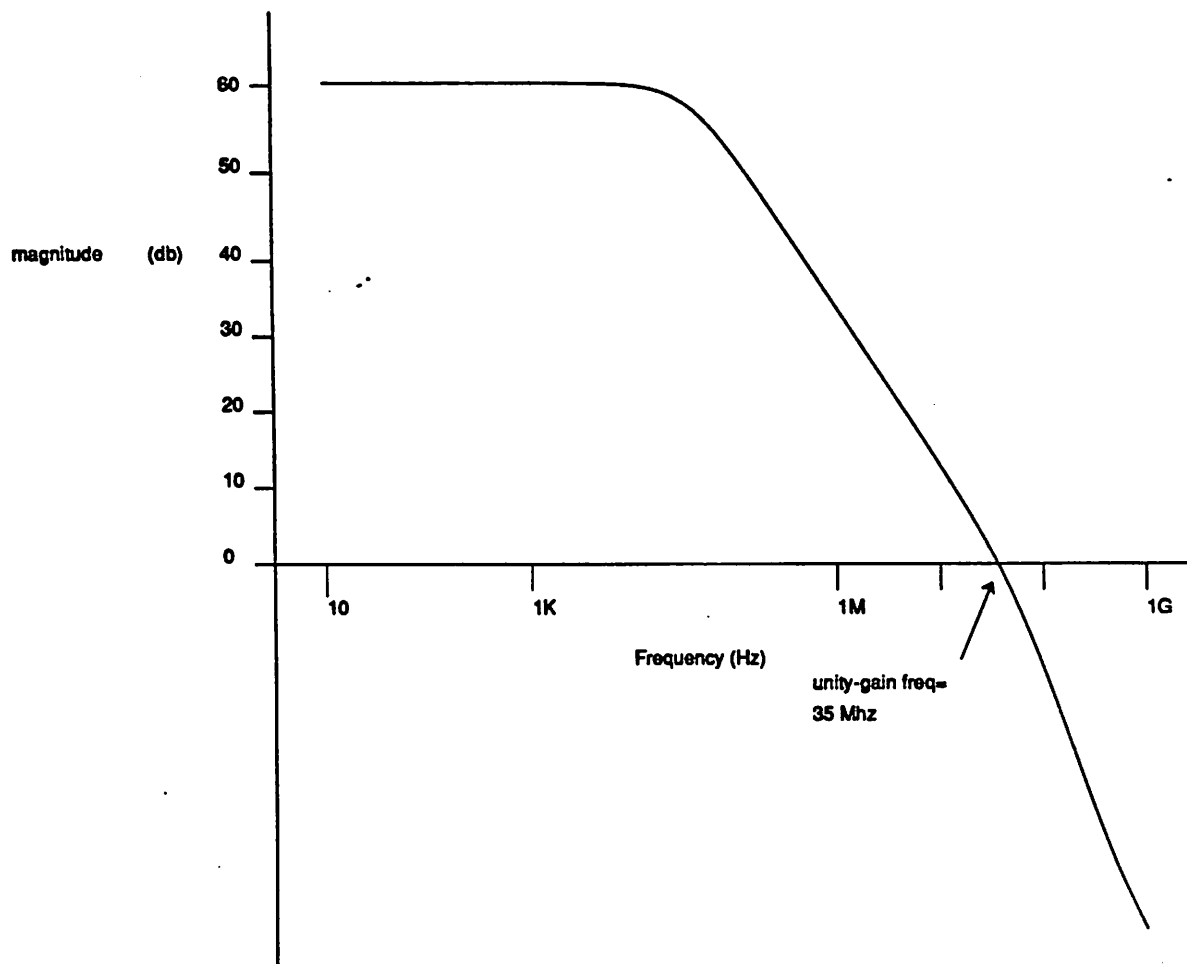


Fig. 4.3.7 Frequency response

opamp differential gain is totally unaffected. The dc value of the common mode output voltage is, however, not well defined, and is determined by the initial voltage across C_7 and C_8 . The purpose of C_5 and C_6 is to establish the voltage drop across C_7 and C_8 that gives the desired common mode output and to periodically restore it to compensate for leakages. As shown in fig. 4.3.6 common mode feedback amplifier is composed of MC1, MC6. The gain of this amplifier will be about the same as the differential amplifier. The speed depends on $G_{m_{mc6}}$ and C_p . Normally MC6 is about one-tenth the size of MD6 because that means the size of the mismatch between the p-channel transistors (MD1, MD3) and n-channel transistors (ND5, MD7) that the common mode branch can take care of is one-tenth the size of MD5. The size in turn determines the relative speed of the common-mode feedback with respect to that of the differential mode feedback. The common-mode feedback can work slower than the differential mode feedback

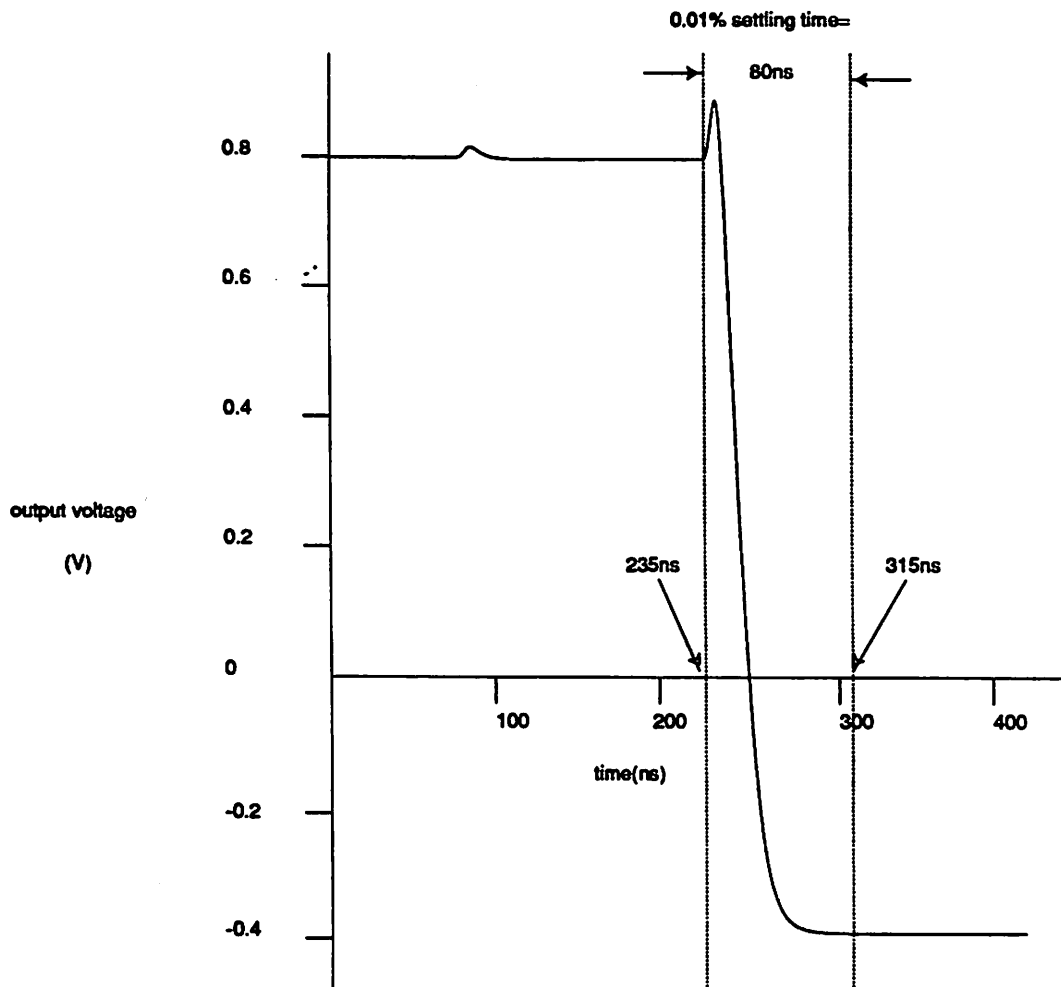


Fig. 4.3.8 Transient response

because the common mode to differential conversion due to mismatch can be small. Compared to the common-feedback circuit published before¹² the drain of MC6 is connected to the output node instead of to the cascode node. This is due to the fact that we want the opamp to have fast settling but only medium gain. Moving the $drain_{MC6}$ away from the cascode node reduces the parasitic capacitance at that point and increase speed. However putting it at the output node will shunt the output resistance and reduce the gain.

Capacitor structure

Since there are no two layers of poly the other capacitor that can be used is a metal-poly layer capacitor. The problem with this is two fold: one is the fact that the area will be large since metal-oxide thickness is 10 times that of poly1-poly2 layer in a typical process. One solution to that is the use of MOS

transistor as a capacitor for C_3 . The main advantage is the savings in area. Even though the MOS transistor has a nonlinear capacitance value with respect to the voltage across it, since the integrator is followed by a comparator, the nonlinear effect is cancelled out. Furthermore in an integrator C_3 is the largest capacitor of the system and such an approach will save a lot of area. However this approach is only good for C_3, C_4 . All the other capacitors still need to be linear capacitors. The solution that is being adopted here is to use 3 layers: metal2, metal1 and poly to make two capacitors and connect them in parallel.

The other problem is the parasitics that is sitting on the bottom plate of the capacitor is now on the same order of magnitude as the capacitor itself. This parasitic capacitor will normally be sitting at the output of the opamp and thus change the position of the dominant pole. With process variation the capacitor can change quite a bit and hence the opamp may become unstable. The above approach of a 3 layer structure helps to mitigate the problem. However extensive simulations are being run with capacitor variations of at least 20% on both sides of the nominal value to make sure there is no stability problem and that the settling time is still within 125ns.

All capacitors are being put into a well to prevent crosstalk. To save area common-mode feedback capacitor could have been implemented by the parasitic capacitor of C_3 . However that dictates that the C_7 to be around half of C_3 . With huge parasitics care has to be exercised to make sure the parasitic is on the output, not on the summing node so that the feedback factor is not further decreased.

Bias circuit

The bias circuit as shown in fig. 4.3.9 is similar to the one used in.¹³ Because of the requirement that the circuit operates on a single 5 volt supply, it is important that the cascodes be biased for optimum voltage swing. The generation of high-swing bias voltages has to pay special attention to the sensitivity of the body effect. This problem is avoided in this circuit which uses the triode region of device MBN7 to develop a well defined voltage drop in series with the source of MBN8. The W/L of MBN7 is adjusted so that it gives a V_{ds} equal to the V_{dsat} of MBN6 plus the necessary few hundred millivolts of $(V_{ds}-V_{dsat})$ desired in the biasing of the drain voltage of MD7. The body effect of MD5 is cancelled by that in MBN8, so that the $(V_{ds}-V_{dsat})$ of MD7 is first-order independent of γ .

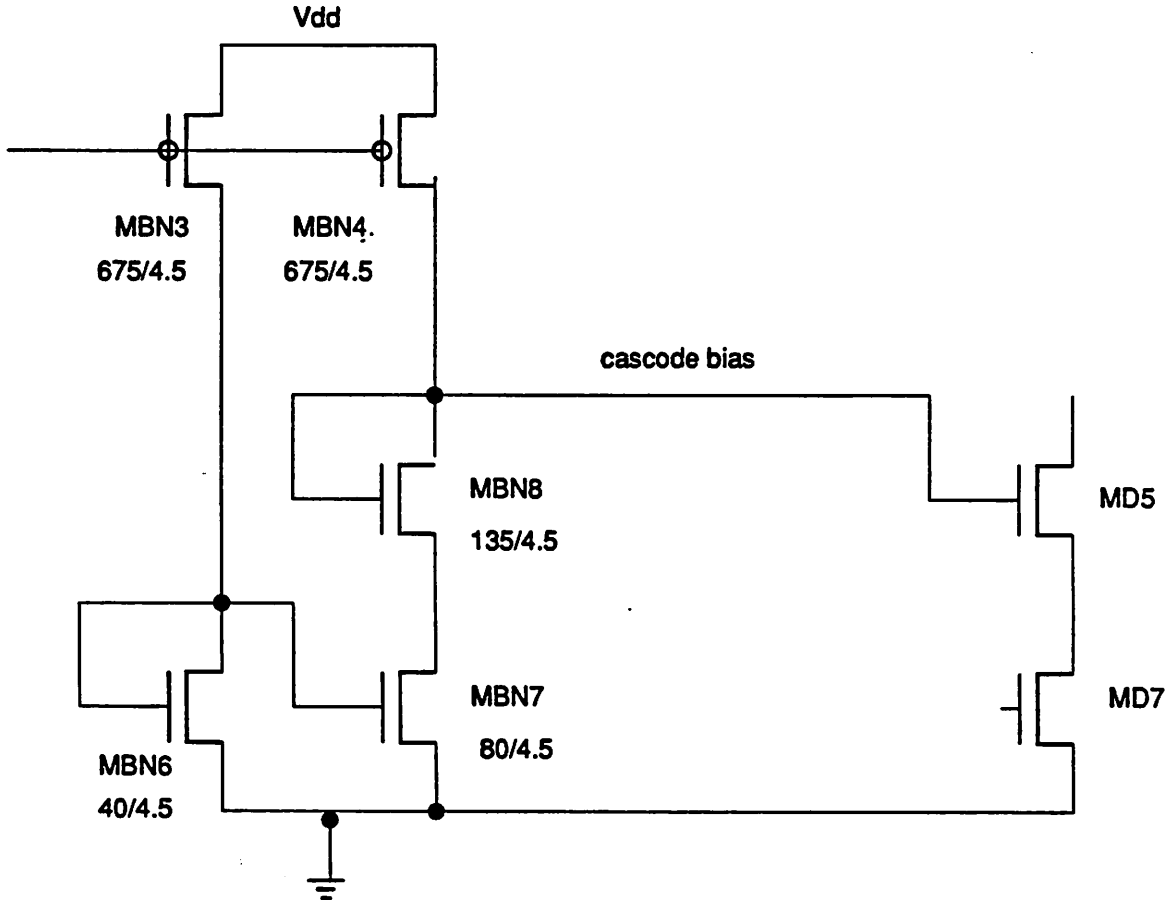


Fig. 4.3.9 Prototype high swing bias generator

4.3.5 Comparator Requirement

Offset

The comparator offset does not have any long term effect on the coder. All it does is to shift the comparison point up. That this is so is due to that fact we are having a degenerate case of a one bit A/D and so the precise position of the decision level is not important. A change to even a 2-bit A/D will totally invalidate this assumption.

Speed

The comparator needs to latch in the time slot given a certain amount of overdrive. The overdrive is dependent on the uncertainty that the comparator can take due to hysteresis and other effects. Simulation has established that hysteresis on the order of 1/100th of the output swing of the integrator will begin to degrade the signal-to-noise ratio. Now given a certain output swing from the opamp, depending on the

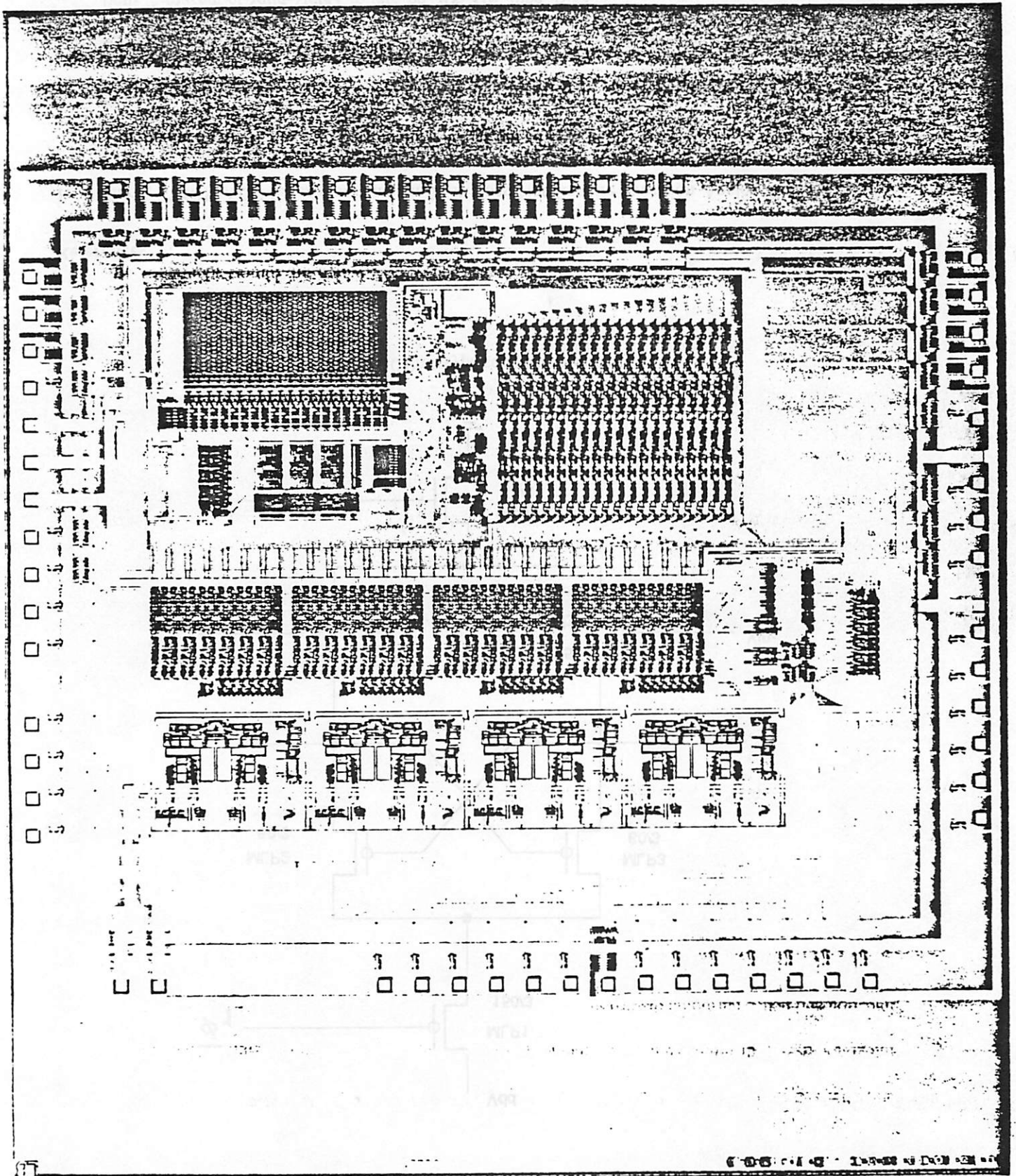
level of the input signal there is a certain minimum input level to the comparator. Since dither is applied the input signal will never be exactly at full scale. Assuming the input is at -3db down, the output of the integrator will be a sequence of 1,1,1,-1,1,1,-1... Since this means the output of the opamp is going up 3 times and then down once, that means the incremental step is around 1/4 of the maximum output swing. Any input signal that is smaller is going to have fewer number of ones in the sequence, consequently the incremental steps size will be larger. Therefore it can be expected that the smallest input of the comparator will be the maximum output swing of the opamp divided by four. This number will then be the minimum overdrive to the comparator and the comparator has to latch in the given amount of time. The final design is a trade-off of the swing of the opamp and the speed of the comparator for a given technology.

4.3.6 Comparator Circuit

Since offset is not important the comparator can be a simple latch as shown in fig. 4.3.10. This latch is a cross-coupled latch that is biased by MLN6. Initially when ϕ_1 is on MLN1 is off and the V_{out+}, V_{out-} is at a voltage close to ground. When ϕ_{1b} comes along, the output voltage V_{out+}, V_{out-} will both try to go up. Eventually because of the difference in input voltage one will go up faster than the other and start the regenerative action through positive feedback. Then one output will go towards Vdd and the other output will come down. The speed in which the comparator can latch depends on the G_m of MLP2 and MLP3, as well as the parasitic capacitances at the output node. Care has to be taken in the layout of the comparator to make sure the circuit is perfectly symmetric to make sure that the same amount of load is present at both V_{out+}, V_{out-} . With this comparator the latching can be achieved in 20ns with an overdrive of 10mv.

4.3.7 Layout

A photomicrograph of the complete chip is shown in fig. 4.3.11



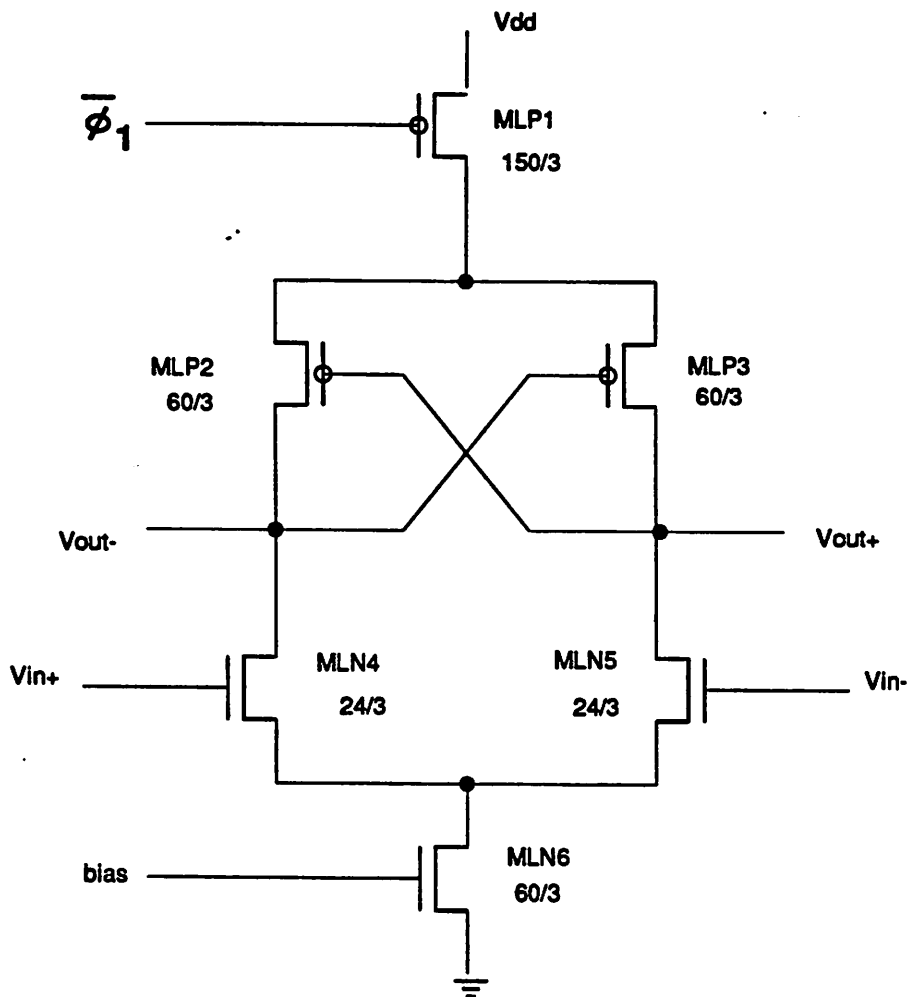


Fig. 4.3.10 Prototype comparator circuit

References

1. Hurst, "MOS ADC-Filter Combination That Does Not Require Precision Analog Components," *ISSCC Digest of Technical Papers*, pp. 80-81, 1985.
2. Candy, *Berkeley I.C. Seminar*, 1986.
3. DeMan, "Compact NMOS building blocks and a methodology for dedicated digital filter applications," *Journal of Solid-State Circuits*, pp. 306-316, June, 1983.
4. Pope, "Automatic generation of signal processing integrated circuits," *M85/11 ERL Memo, Berkeley*.

5. Raebaey, "Lager: an automated layout generating system for digital signal processing circuits," *M85/5 ERL Memo, Berkeley.*
6. Neff, *Master Report, Berkeley, 1987.*
7. Roettcher, "A compatible CMOS-JFET pulse density modulator for interpolative high-resolution A/D conversion," *Journal of Solid-State Circuits*, pp. 446-452, June, 1986.
8. Misawa, "Single-chip per channel codec with filters utilizing delta-sigma modulation," *Journal of Solid-State Circuits*, pp. 333-341, 1983.
9. Lee, "Low-distortion switched-capacitor filter design techniques," *Journal of Solid-State Circuits*, pp. 1103-1113, Decemeber, 1985.
10. Hauser, "Circuit and technology considerations for MOS delta-sigma A/D converters ," *Proceedings of ISCAS*, pp. 1310-1316, 1986.
11. Kaneshiro, "Circuit and technology considerations for high frequency switched-capacitor filters," *M83/42 ERL Memo, U.C. Berkeley.*
12. Castello, "Low-voltage low-power MOS switched-capacitor signal-processing techniques," *M84/67 ERL Memo U.C. Berkeley.*
13. Laber, "A high-performance 3u CMOS analog standard cell library," *Microlinear Corp..*

CHAPTER 5

EXPERIMENTAL RESULTS

INTRODUCTION

This chapter is a summary of the test results on the test chips that have been fabricated. A total of three runs have been submitted to MOSIS for fabrication in the 3-micron double metal single poly process. The first run contains logic problem and the chip does not function properly. The second run fixes this problem but runs slow. The chip can only run to 90% of the speed at 5V. It takes at least 6.5V to make some of the chips to function at full speed. But at that point since the process is only a 5V process the background noise starts to increase. The third run fixes up the speed problem and the chip can run at full speed at 5V.

5.1 Test Setup

The chip is being tested by the Spuds board and the tester board as shown in fig. 5.1.1. The operator will start by downloading a program called upload.com from the VAX into the program section of the Spuds board. The Spuds board uPC(microprocessor) actually runs a loop stored in the PROM and transfers the program from the VAX into the program section. Upon finished loading it will jump to the beginning of the program section and starts executing the program upload.com. This program does a number of things. First of all it resets the tester board and determines the amount of data that needs to be uploaded to the VAX eventually, according to the option specified by the operator. The operator can upload up to 15K of 32 bits data. Then it sets the TST bit on the tester board, goes into a loop and wait for the END signal from the tester board. The tester board, originally sitting idle and ignoring any data that is being sent from the chip, upon reset will start accepting the data. The control section takes a data valid signal from the chip and writes the data from the chip into the memory. The counter keeps counting until the memory is full. Then it sends an END signal to the Spuds board. The Spuds board then disables the TST bit, resets the counter on the tester board, and uploads a designated amount of data (as calculated previously in the upload.com program) to the Spuds board. While all these things are happening the VAX is sitting in a loop waiting for the

Spuds board to finish. When the Spuds board is done, it sends control characters to the VAX and the VAX then uploads the data.

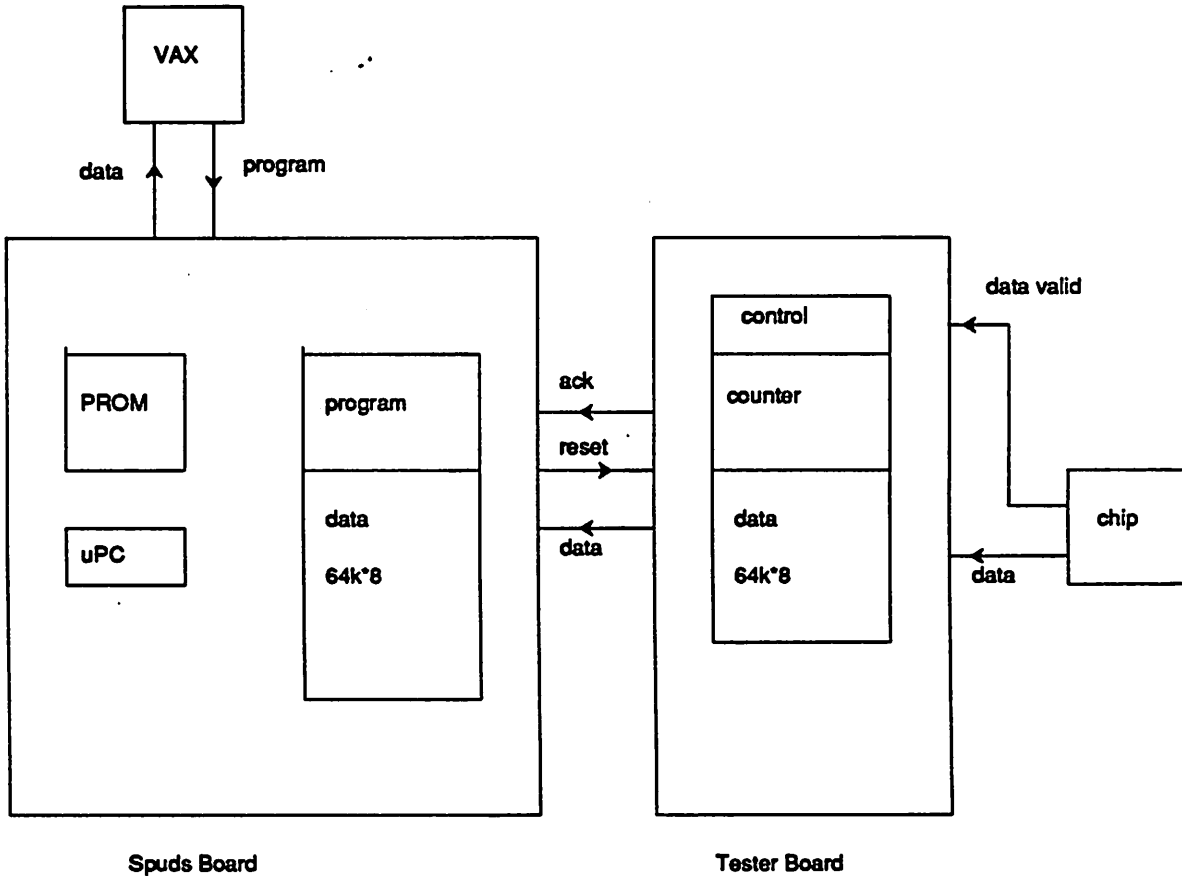


fig. 5.1.1 Test setup

5.2 Experimental Results

5.2.1 Experience from chip1

There are logic errors found in chip1. Separate chips have been fabricated for the the individual analog, FIR and IIR parts to see where the errors come from. Both the analog front end and the FIR are found to function correctly logically. It was found out that the logic error comes from the IIR filter. It has to be stressed that the IIR filter has been through logic simulation by the program KESIM. The fact that the errors escape the logic simulation is because the logic verification is not complete. ¹ As it turns out there are

three errors in the IIR filter, the first one is a GROUND that is not connected. The second one is a logic error in the control circuitry of the AUJO. This error will make the adder add the absolute value instead of the true value of the inputs. The third error is a timing error in the barrel shifter that causes intermittent problems. These problems are fixed and chip2 is sent out.

Chip2 has the following unique features: The first feature is the use of the transmission gate adder as shown in fig.4.12, the reason being that this gate takes up less area but it also is slower with a simulated per stage delay of 5.5ns. Using a 14 bit adder and other logic delay the total time it takes to traverse the critical path is around 100ns. The allowable time slot is 125ns subtract the necessary non-overlapping between the clock phases. The necessary non-overlapping was thought to be on the order of 5ns for each phase, making the total critical path to be 110ns, giving a safety margin of around 10% only. Also chip2 uses one common power and separate grounds for the digital and analog sections.

5.2.2 Experience from chip2

After testings have been done it was found out that chip2 works only to 90% of the designed speed. In addition separate chips have been fabricated for the the individual analog, FIR and IIR parts to see the limitations. It was found out that the adder used in the FIR of chip2 is the limiting factor. The reason for the limitation is the fact that the required non-overlapping between phi2 to phi1 clock is a lot more than 5ns, more like 15 to 20ns. The nonoverlapping is necessary for the control signal latch.phi2 in fig. 4.10 to be derived from other control signals. If not enough non-overlapping is allowed, the next data will come along at the rising edge of phi1 before the latch.phi2 signal goes down, thus overwriting the original data. *This kind of long non-overlapping is typical of a system that involves decimation, or a change of clock rate.*

5.2.3 Experience from chip3

First of all chip3 uses the adder as shown in fig.4.13 which is faster but occupies more area. From simulations this adder can run with a ripple carry delay of only 3ns. Secondly it uses 3 separate power and ground buses, one for analog, one for digital and one for clock driver and output buffers. This arrangement is selected to minimise switching transients between the digital and the analog portions. Thirdly the guard ring around each separate analog front end is increased from a width of 6 microns to 30 microns to help minimise crosstalk. Fourthly the buffers switch only at the downsampled frequency and the operational

amplifier settles at the clock phase when output buffers are not switching. Finally the switched capacitor integrator uses a different switching scheme as shown in fig.4.3.5 to minimise signal dependent charge injection and hence harmonic distortion.

5.2.4 Test results and methods

Chip3 is tested and it works to full speed while satisfying the idle channel and signal to noise ratio requirements. The frequency response also satisfies the D3 specs. Below is a summary of all the test results. A die photograph of the converter is shown in Figure 5.5. The signal to noise ratio of the converter was evaluated by putting a sine wave of 1 KHz into the converter, uploading the digital output to a computer and then running a 1024 point FFT to compute the ratio of the signal power to the noise power. The crosstalk was measured by putting a full scale sine wave at 1 KHz at the input of channel one, a zero input to channel two, and then measured the 1 KHz component at the output of channel two. Power supply rejection ratio at 1 KHz was measured by applying a 20mV peak to peak sine wave at 1 KHz at the Vdd of the chip, zero input to the chip and measured the 1 KHz component at the output. Measurements were repeated for different frequencies from 500 Hz to 20 KHz. The idle channel noise was measured by putting zero input at the channel input, summed up the noise at the output and take the ratio between the measured noise with the input signal level at the overload point. Also the noise spectral density is C-message weighted. The observed performance is summarized in Table 1. In Figure 5.1, the signal to noise ratio is plotted against input amplitude for a power supply of 5V and 4.75V. In fig. 5.2 the frequency response of the overall coder is shown for a power supply of 5V and 4.75V, which are almost identical. Anti-alias requirement is also tested at higher frequency at 13 KHz and 29KHz to measure how effective the filter FIR1 and FIR2 are in removing frequency components around 16 KHz and 32 KHz respectively. At 13 KHz the signal is attenuated by 42db and at 29 KHz the signal is attenuated by 39db , both attenuations larger than the 33db requirement. In fig.5.3 the gain tracking characteristics of the coder is shown. In fig.5.4 the FFT output of a 1KHz sine wave input is shown. The output is taken at a downsampling frequency of 16 KHz. Note that the noise is not white but is filtered so that when the output is decimated to 8 KHz, the noise aliased will not increase the noise in the passband by much. Also note that the second and the third harmonics are more than 76db down.

Table1-Typical Performance: 5V power supply and 25 deg C

Area	8250 sq. mils per channel
Dynamic range	79db
Idle channel noise	11.6dbrnC0
Crosstalk	-83db
Power supply rejection ratio	43db (1 KHz) 60db (10 KHz)
Harmonic distortion	-76db (2nd) -82db (3rd)
Power	50mW per channel

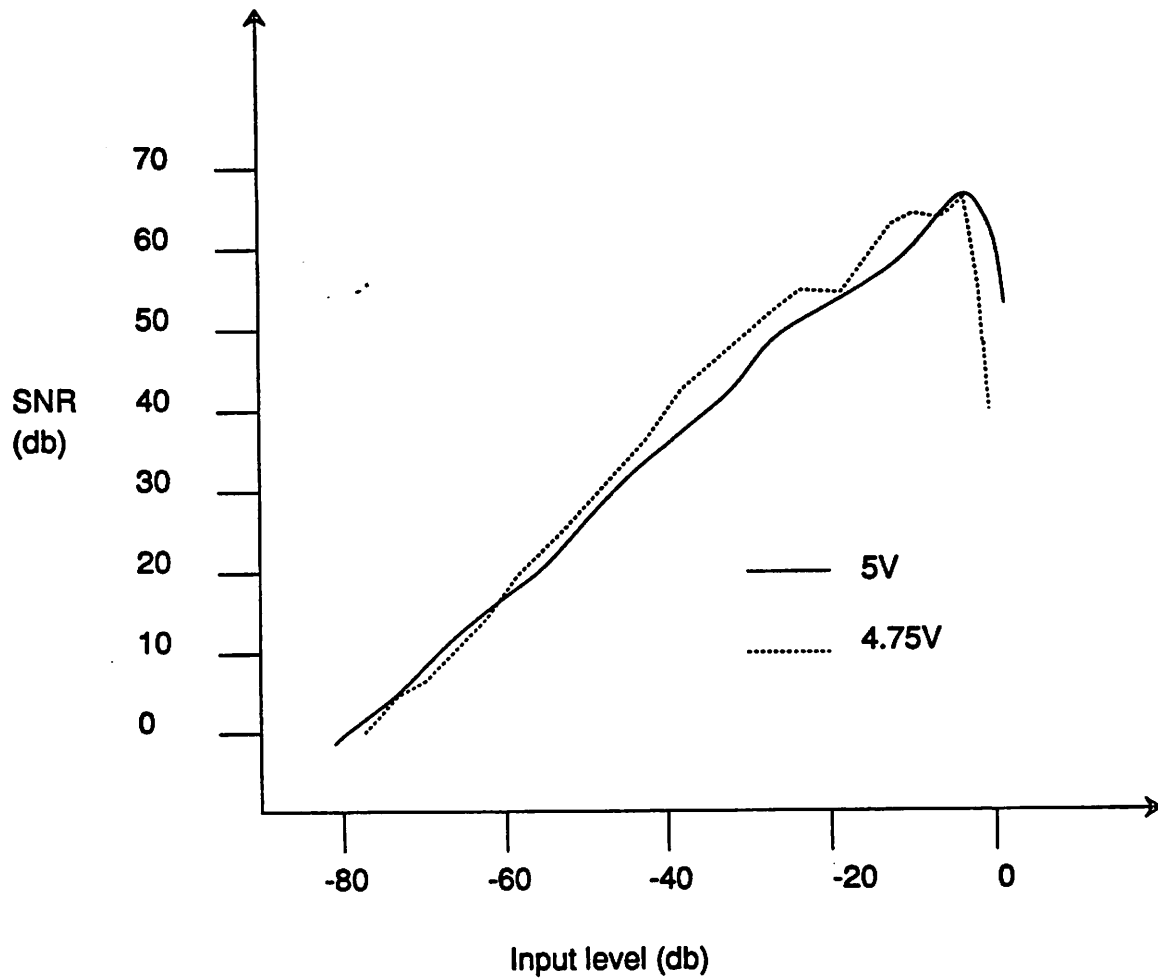


fig.5.1 SNR curve (5V, 4.75V)

The signal to noise ratio curve is obtained by applying a sine wave of 1 KHz at the input, upload 1024 points of the digital output, applying a 4th order Hanning window and a 1024 points FFT, then sum up all the energy at 1 KHz to determine the signal energy, and sum up all the rest to determine the noise energy (which include the harmonic distortion components) and calculate their ratio. The input amplitude is varied every 3db around the peak input amplitude and around the idle channel condition, and every 5db in between.

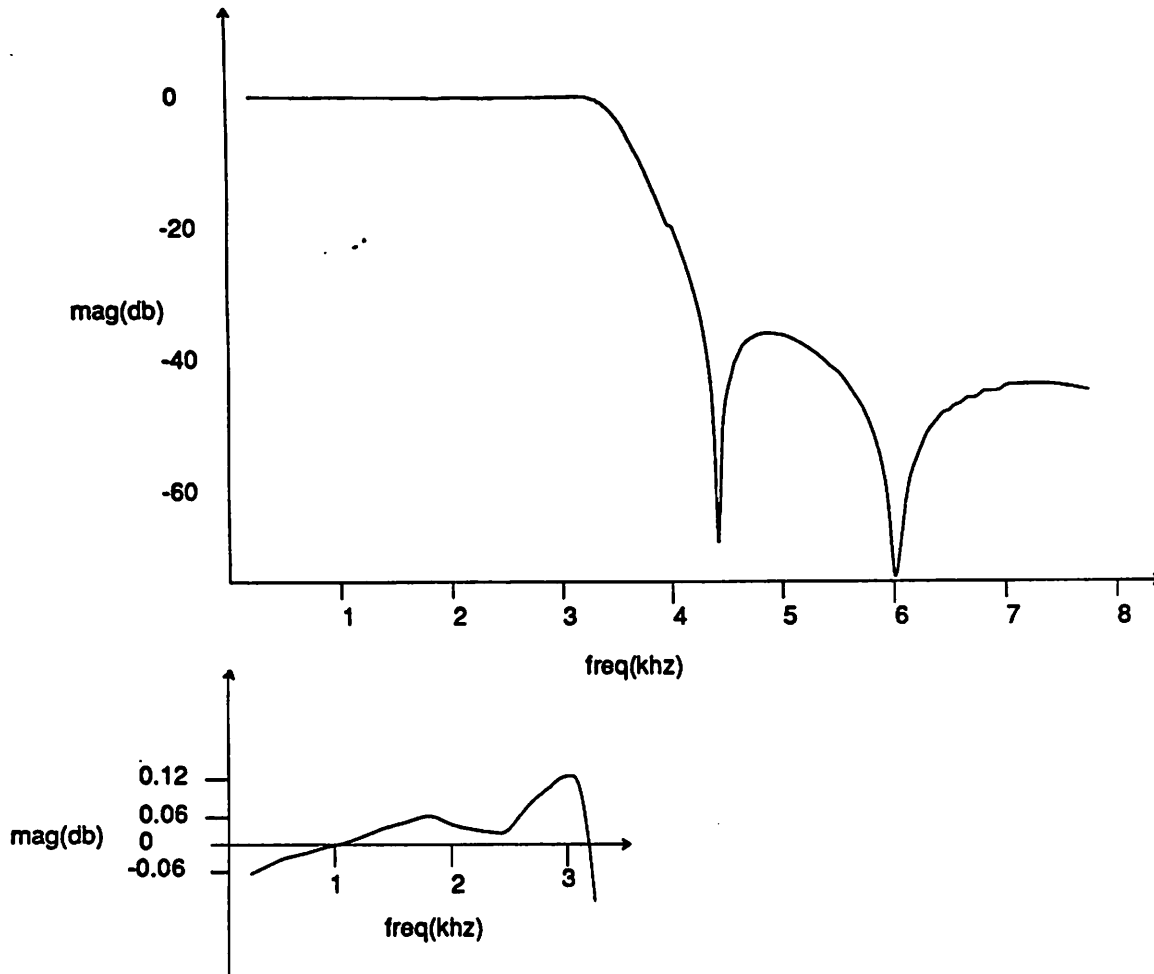


fig.5.2. frequency response(5V, 4.75V)

The frequency response is obtained by applying a sine wave at the input, upload 1024 points of digital output, applying a 4th order Hanning window and a 1024 points FFT, then sum up the energy at the input frequency. The frequency is varied every 50 Hz in the passband and in the stopband and every 20 Hz in the transition band.

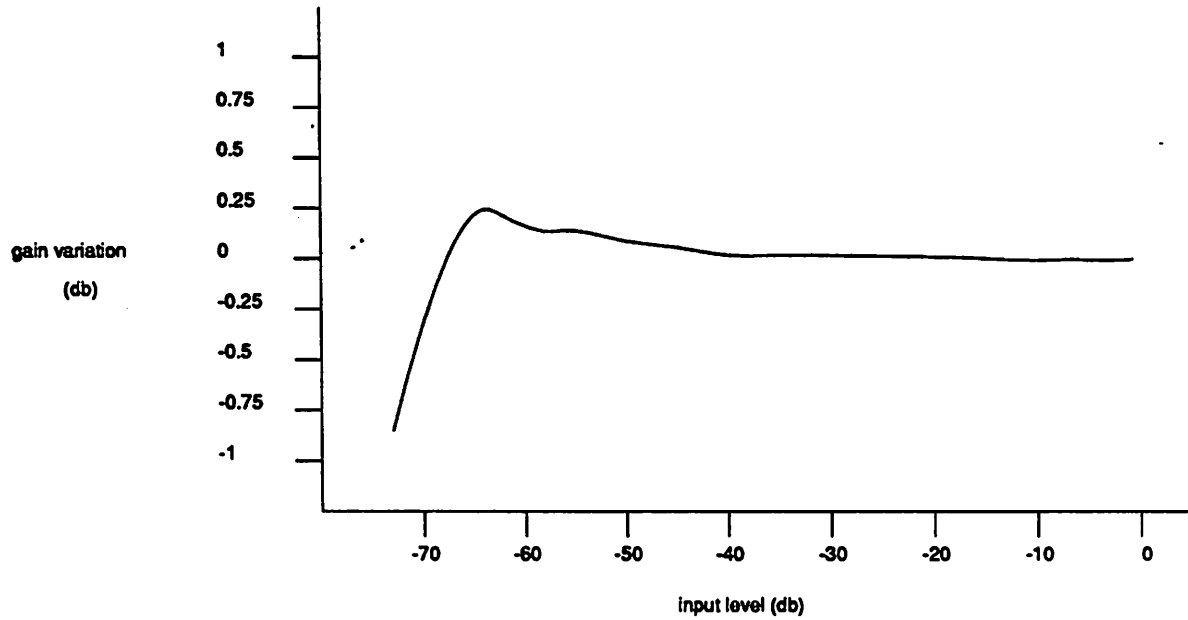


fig.5.3. gain tracking

The gain tracking is obtained by applying a sine wave of 1 KHz at the input, upload 1024 points of digital output, applying a 4th order Hanning window and a 1024 points FFT, then sum up the energy at 1 KHz to calculate the output. The input is measured by a ac voltmeter and the ratio is determined. The input amplitude is varied every 5db from peak to idle channel condition.

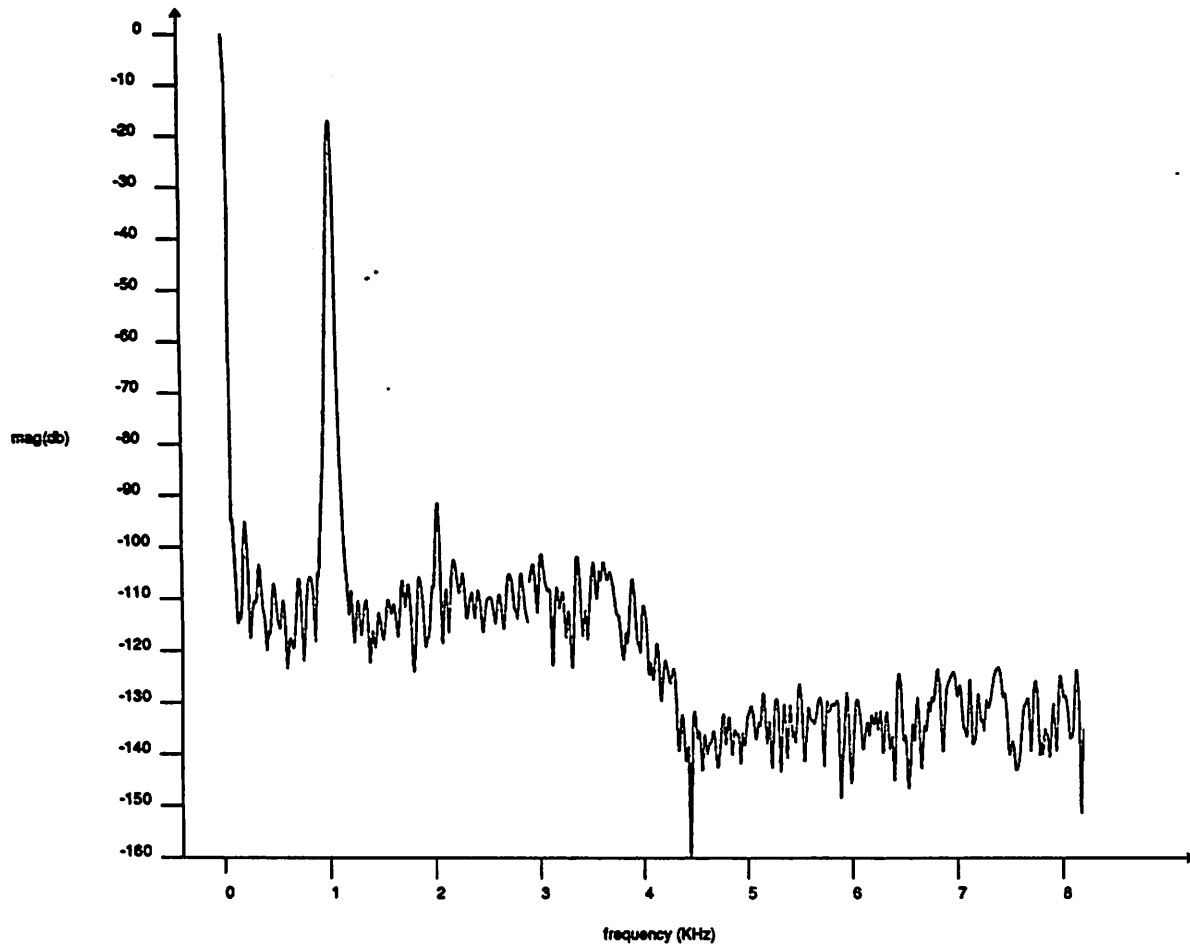


fig. 5.4 FFT of the output from the coder with a 1 KHz sine wave input

Note that the harmonics are 76db down. Also note that there is a large dc input that is due to the FIR filter.

References

1. **Neff, *Master Report, Berkeley, 1987.***

CHAPTER 6

Conclusions

6.1 Introduction

The overall objectives of this thesis are to explore new circuit and system concepts that will allow multi-channel oversampled PCM voiceband coder realized in CMOS technology to be a viable alternative to the conventional analog approach, such as that of the successive approximation. The major goal is to minimize the total chip area. Below is a discussion of the comparisons between the multi-channel oversampled approach, the single channel oversampled approach and the conventional analog approach. The impact of scaling on the comparisons will also be addressed.

6.2 Specific Functions Of The Coder

The problem that has to be tackled is to find the architecture that will result in a minimum cost multi-channel CMOS voiceband coder. This interface has to accomplish the following functions:

- 1) Perform the analog to digital conversion that has an idle channel noise that is 80dB down from full scale. This is because in the telephone communication, a human ear is very sensitive to such idle channel noise.
- 2) Incorporates the anti-alias filter. This filter has to meet the D3 specifications.
- 3) Achieves a channel to channel crosstalk of less than 75dB. This crosstalk will result in interference between two pairs of speakers using the phones.

The choice of the architecture will be based on minimum cost, which in a CMOS technology will mean taking up the least amount of area. In the following discussions the area taken up by different approaches will be explored, and the relationship with the feature size of the process established.

6.3 Area Comparisons Between Different Architectures

6.3.1 Conventional Approach

A conventional coder in MOS technology typically is consisted of a continuous time anti-alias filter, a 5th order elliptic switched capacitor filter and an ADC using the charge redistribution technique as shown in fig. 6.1. The area of the coder is consisted of these three components. From published data an example of a PCM voiceband coder at 3.5u CMOS technology has been estimated to take an area of 7K sq. mils. To scale this to 1.5u using constant voltage scaling let us go through an example. From it has been shown that a switched capacitor filter at 3u technology occupies 1000 sq. mils. This chip has been scaled to 1.5u technology. It has been shown that the area of the switched capacitor filter can shrink by a factor of two. The major reason for a factor of two, rather than a factor of four reduction as would be the case in ordinary digital circuits, is the fact that the capacitor cannot shrink as the feature size. This is due to the kT/C noise inherent in the sampling process in a switched capacitor filter. For a constant power supply, like 5V, the value of the noise has to be kept below a certain value for a given dynamic range. This means the capacitance value C cannot be reduced below a certain value. Assuming constant voltage scaling, the capacitance per unit area goes up by a factor of $\sqrt{2}$ every time the feature size shrinks by half. Assuming the total capacitance stays constant, then the area for the capacitance is reduced by $\sqrt{2}$. Since about half of the original chip area comes from the capacitor, or 500 sq. mils, then after scaling by 0.707 it is estimated to be 380 sq. mils. The rest of the circuit takes up 500 sq. mils and is estimated to shrink by a factor of four to 125 sq. mils. Therefore the final chip size for the 1.5u process is estimated to be 500 sq. mils. This estimate is close to the actual chip that has been fabricated. This method of estimating the area for scaled circuits is then applied to the whole coder mentioned above. Assuming again half of the area of the circuit comes from capacitors, or 3500 sq. mils, this will scale to 2500 sq.mils. The rest of the circuits takes up 3500 sq. mils and will scale by a factor of 4 to 875 sq. mils. The final circuits at 1.5u is then 3375 sq. mils. Fig.6.2 shows how the conventional coder is estimated to shrink down in size as a function of scaling as calculated above. It can be seen as one scales beyond 1.5u a large fraction of the chip is occupied by the capacitor and the whole circuit does not scale down as fast as digital circuits.

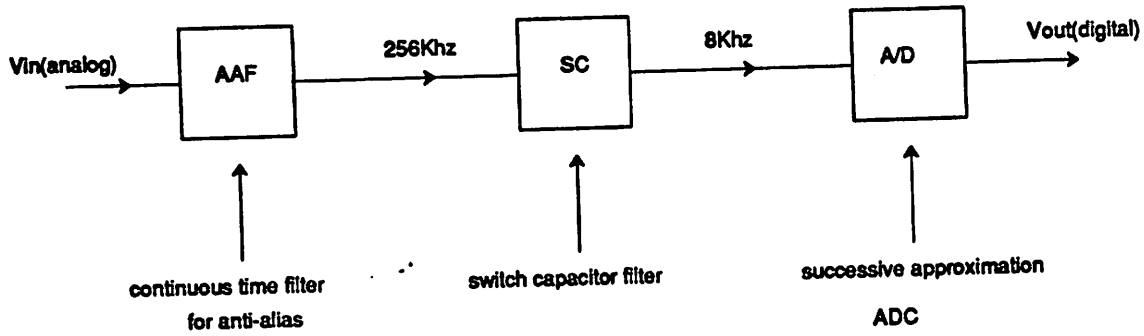


Fig. 6.1 Conventional approach

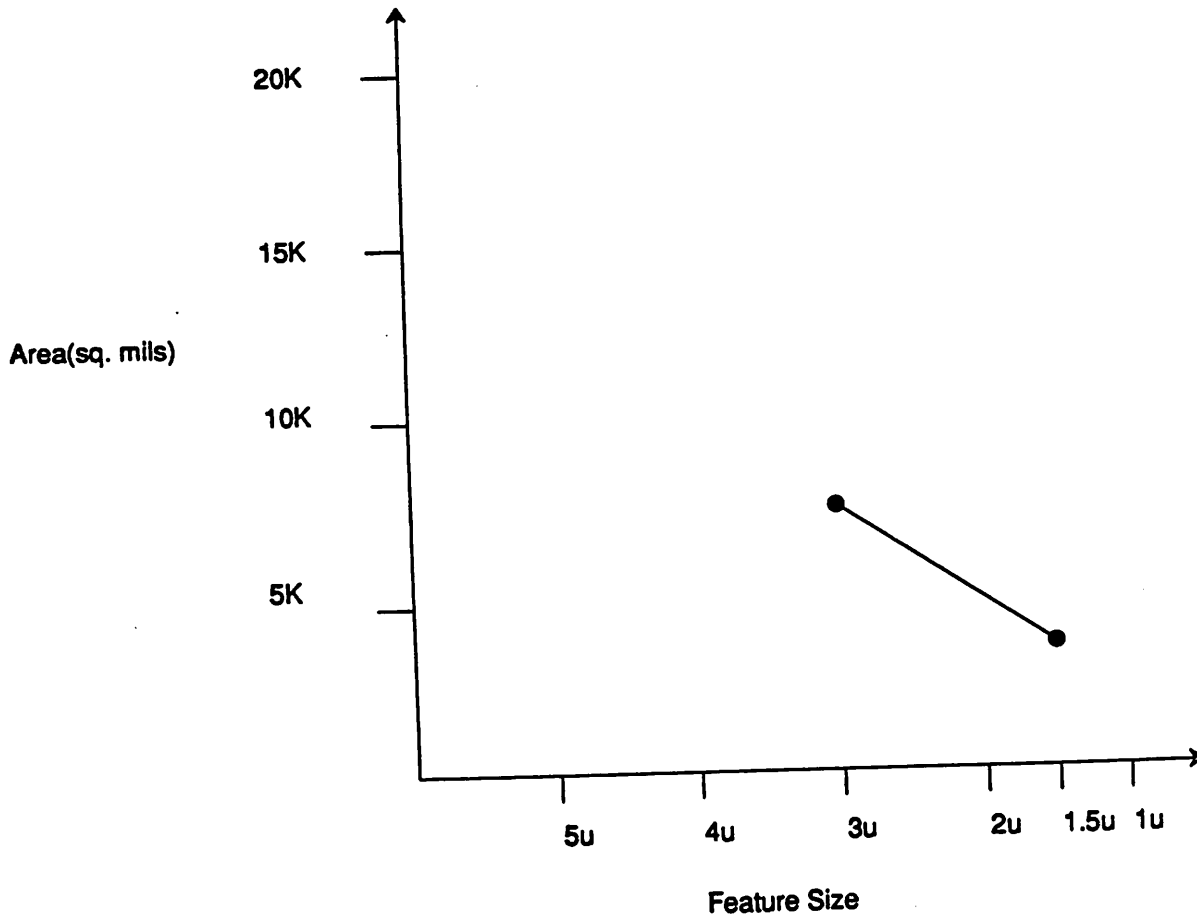


Fig. 6.2 Area versus feature size for conventional approach

6.3.2 Single Channel Sigma-Delta Coder

In this scheme the the sigma-delta coder incorporates both the anti-alias function and the out-of-band quantization noise suppression in the same filter. The major distinction is whether we use a first order versus a second order coder. The second order coder gets a higher quantization noise reduction per octave

increase in the oversampling frequency. Hence the first order coder needs to oversample at 4MHz whereas the second order coder needs to oversample only at 1MHz. The filter is broken up into two parts, the first part being a custom built FIR decimation filter and the second stage is a bit-serial IIR filter. For a first order coder the FIR filter has a triangle window response and for a second order coder the FIR filter has a sinc^3 window response as discussed in chapter 3. The reason for choosing a bit-serial approach over a microprogrammed approach is discussed in chapter 4.2. There it is shown that for storing state variables the microprogrammed approach is using RAM cells while the bit-serial approach is using shift registers. Therefore for storage the area efficiency is better in a microprogrammed approach. On the other hand the computation unit of a microprogrammed has to be heavily pipelined to improve the speed while a bit-serial approach uses only simple one-bit adders. Therefore the computation unit of a bit-serial approach is more area efficient. Now in a single channel the computation unit takes up more space because only the state-variables of one channel is needed. Hence the bit-serial approach should be adopted. In a multi-channel case since the computation unit is time-shared but the storage is not, the storage area dominates and so a microprogrammed approach is favored. This conclusion applies to both the first and second order coder.

The overall system is as shown in fig.6.3. To get an estimate of the area taken the coder is broken up into three parts: the analog front end, the FIR filter, the IIR filter. For a first order coder, the analog front end takes up around 2K sq. mils, the FIR filter around 7.2K sq. mils. The FIR filter's ROM takes around 4.2K sq. mils and the adder 3K sq. mils. and the IIR filter around 6K sq. mils giving a total of 15.2K sq. mils. For a second order coder, the analog front end takes up 3K sq. mils because two integrators are used instead of one. The FIR filter takes up around 6.8K sq. mils. The FIR filter's ROM takes 3K sq. mils and the adder takes 3.8K sq. mils. It is noted that the ROM is less than the first order's ROM because of smaller L(though more bits). However the adder is larger because of more bits and larger L/D ratio. The IIR filter is the same and takes up 6K sq. mils. Therefore the total chip area estimated for the second order coder is 15.8K sq. mils.

Fig.6.4 shows the area trend as one scales from 3 micron to 1.5 micron. The estimate of the area taken by the first and second order coder will be separated into the analog and the digital portion. For the analog part, assume that the sampling capacitor is small enough so that the kT/C noise is dominant. About

50% of the analog front end is taken up by the capacitors. Using the same methodology as outlined in the section on the conventional approach, the capacitor area scale down by 0.7 and the rest to 0.25. Using this the estimate for the first order coder at 1.5u is 4.25K sq. mils and for the second order coder is 4.6K sq. mils.

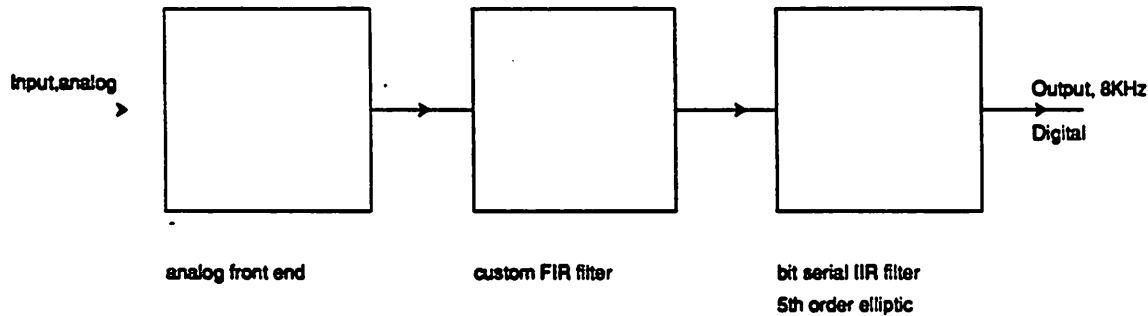


Fig. 6.3 Single channel sigma-delta coder

6.3.3 Multi-channel Sigma-Delta Coder

Shown in fig. 6.5 is the schematic of a multi-channel sigma-delta coder for both the first and second order coder. Separate analog front ends are used because we want to keep the crosstalk between the channels to be small. The area penalty at 3 micron is small since the analog front end takes up a small percentage (around 20%) of the total chip area. At 3 micron the first and second order coder is consisted of a first stage FIR filter followed by a time shared IIR filter. As shown in chapter 4, the IIR filter should be a microprogrammed type filter rather than a bit serial architecture. As shown in the final layout in chapter 4 the area of a multi-channel first order coder comes from the analog front end (2K sq. mils) plus the first stage FIR filter (3K sq. mils) and the second stage time shared microprogrammed IIR filter(3.5K sq. mils) giving it a total of 8.5K sq. mils. It can then be estimated for the second order coder the area comes from the analog front end(3K sq. mils) the first stage FIR filter (6K sq. mils) and the second stage IIR filter (3.5K sq. mils), giving it a total of 12.5K sq. mils. In the estimate for the second order coder the analog front end takes up more area because of the added integrators. The FIR filter also takes up more area because it needs to suppress more out of band quantization noise as discussed in chapter 3. As can be seen the first order

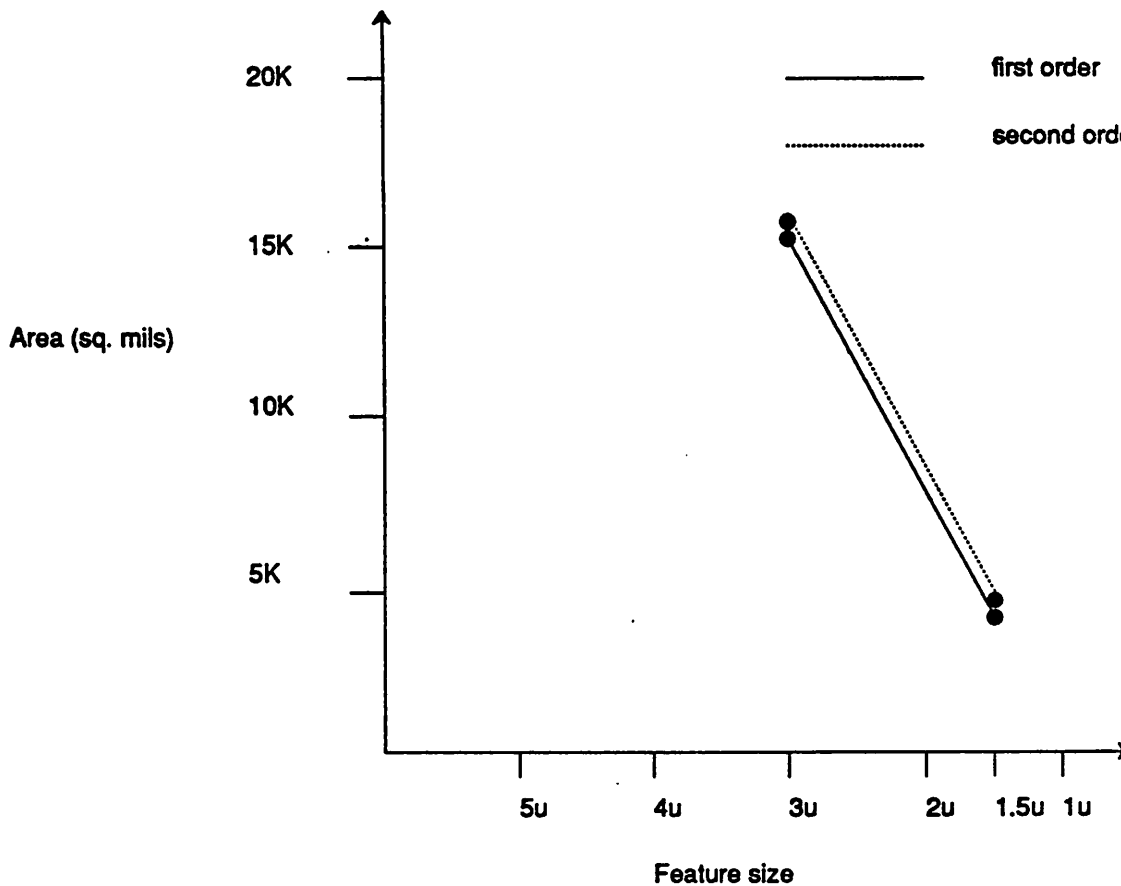


Fig. 6.4 Area versus feature size for single channel coder

coder is better than the second order coder in terms of area efficiency.

Fig. 6.6 shows the area trend as we scale from 3 micron to 1.5 micron. If as assumed as above, that the capacitor area can only be scaled by 0.7 and the rest of circuits can be scaled by 0.25 then first order coder takes 2.6k sq. mils and the second order coder takes 3.75K sq. mils.

The overall area comparisons are summarised in fig.6.7. In the above comparisons two facts stand out. One is the fact that in going from a 3u technology to a 1.5u technology the multi-channel first order coder scales faster than the conventional approach. It is expected as the technology is further scaled down the multi-channel coder will perform even better in terms of area efficiency. The second point is that in both the 3u and 1.5u technology the multi-channel approach uses less area than the single channel approach.

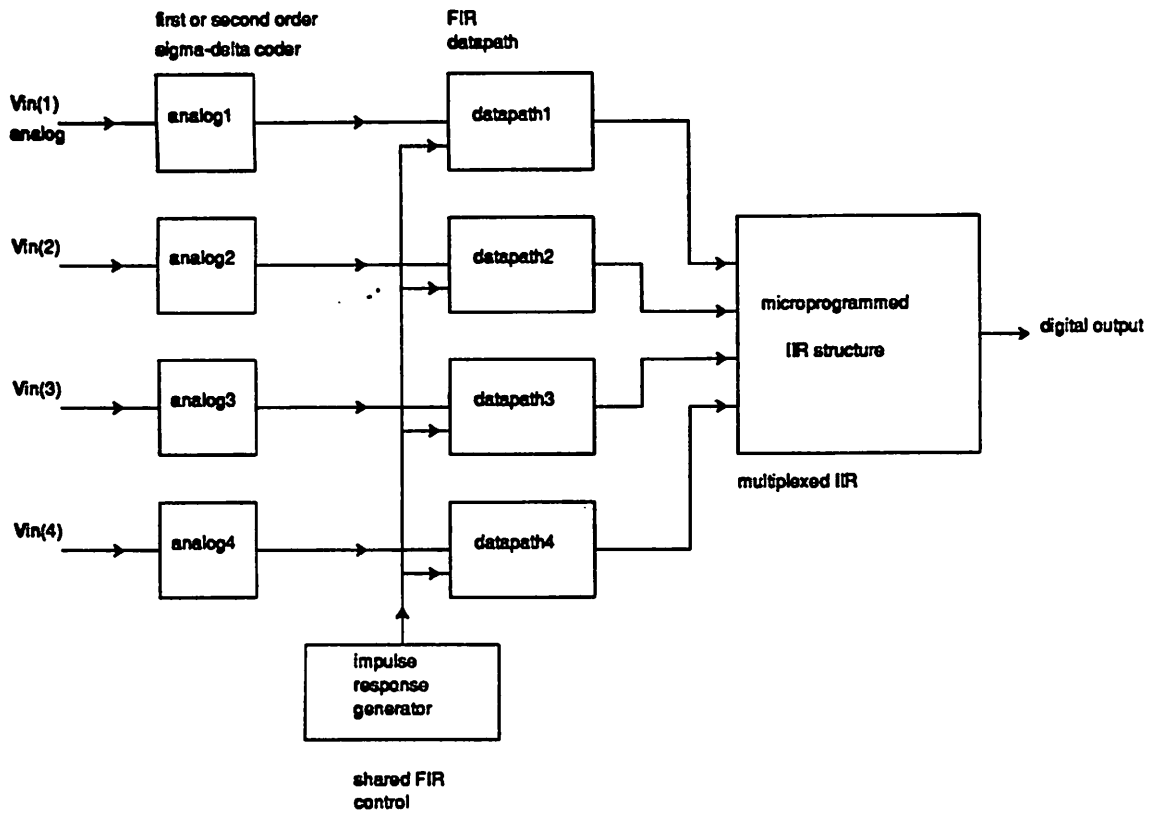


Fig. 6.5 Multi-channel sigma-delta coder

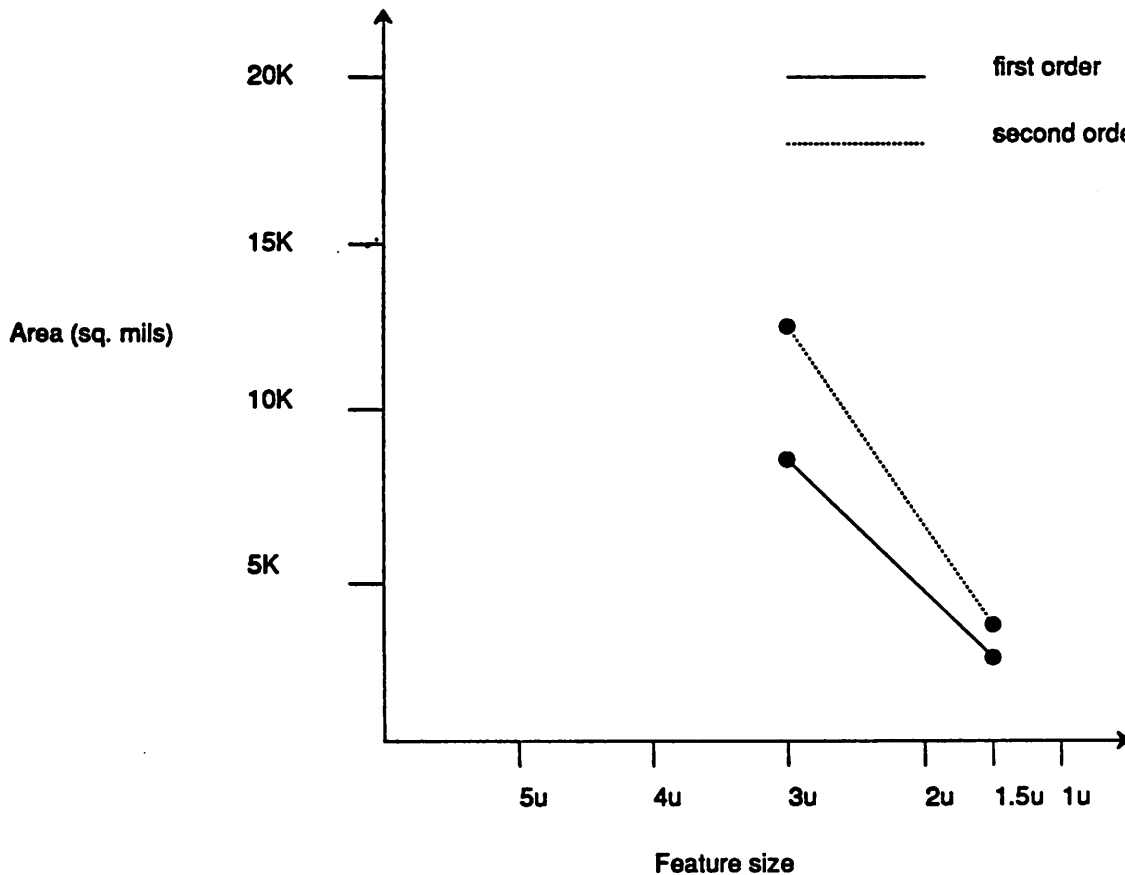


Fig. 6.6 Area versus feature size for multi-channel coder

6.4 Major Research Results

The major research results can now be summarised:

- 1) In terms of area efficiency, a multi-channel first order sigma-delta coder is going to be the optimal solution as technology scales. The crossover point with the conventional approach is around 1.5u.
- 2) In a multi-channel coder, it is the L/D ratio for the decimation filter that determines the complexity. This is different than from a single channel coder case where the parameter L is the deciding factor.
- 3) In a multi-channel coder, a fully differential architecture proves to be effective in minimising the channel to channel crosstalk and improving the power supply rejection.
- 4) For a chip as complex as a multi-channel coder, extensive use of CAD tools such as the silicon compiler system Lager and other simulation tools is essential in producing an error free chip.

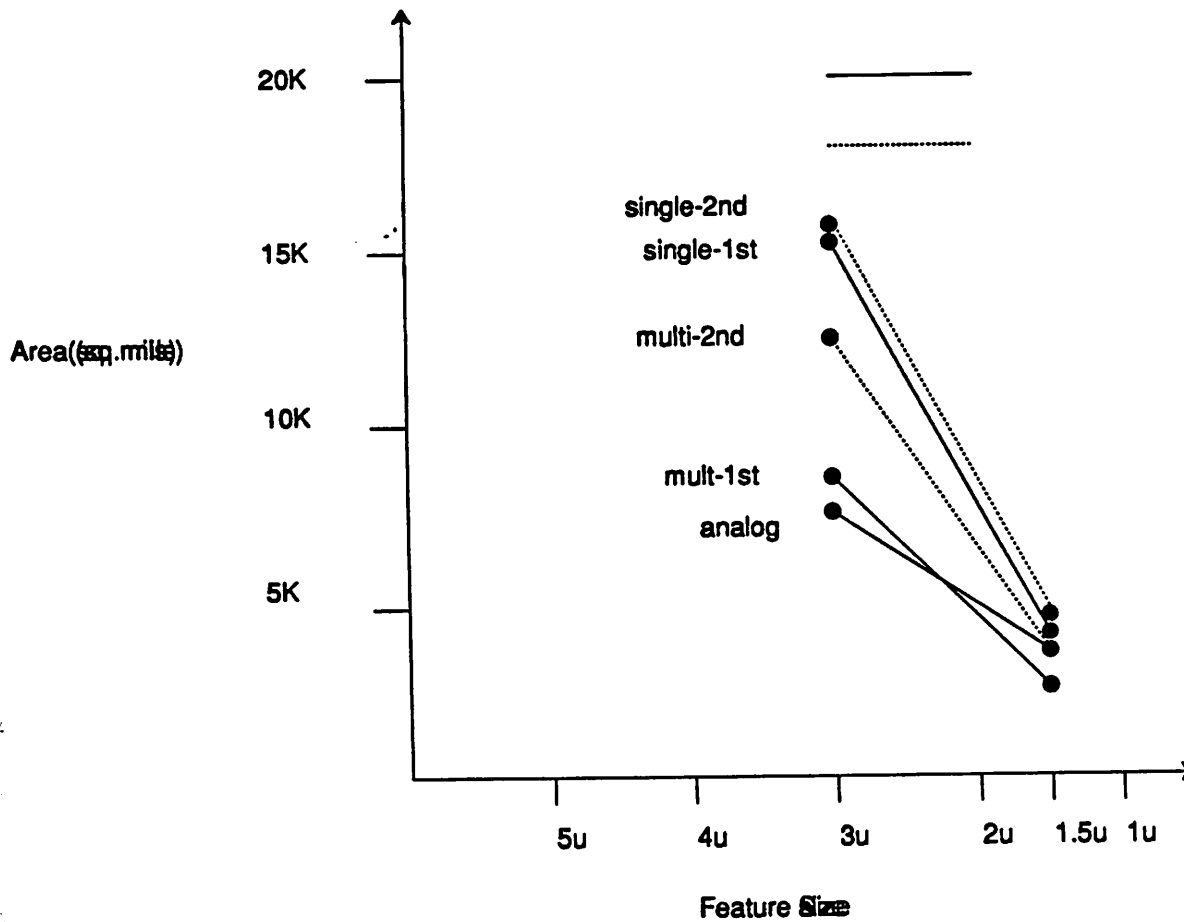


Fig. 6.7 Area versus feature size for all the approaches

6.5 Future Directions

Taking advantage of the speed advantage of scaled technologies, multi-channel coder can be applied in different areas that require higher precision or higher speed, or both. One such application is applying the concept to a multi-channel instrumentation A/D converter where the speed requirement is even lower but the precision needed is higher. A fundamentally different set of requirements will be placed on the analog front end, making more stringent demands on the gain and offset requirements of the operational amplifier. Other areas of interest are speech processing, adaptive networks where the needs of large amount of digital signal processing makes it even more attractive to use oversampled coders, where compatibility with digital processes is assured. Ultimately, a successful design involves an in-depth understanding of the system issues, the analog environment as well as the digital signal processing aspects which are unique to the oversampled A/D converter under consideration.

APPENDIX

CAD TOOLS FOR THE DESIGN OF A MULTI-CHANNEL PCM CODER

Introduction

In this appendix the various cad(computer aided design) tools that are being used in our design process will be discussed. Since the system involves a lot of design tradeoffs, cad tools are particularly useful in the various design phases of the project. In general cad tools are used in the system design phase to come up with system parameters, in the circuit design phase for designing circuits and in the layout phase for automating the layout process as well as extracting and verifying the correctness of the layout.

A.1 Sigma-delta simulator loop

Since the sigma-delta loop is a highly non-linear loop due to its one bit quantizer, a closed form solution for the circuit is difficult to obtain. If one assume a white noise model both the first order and the second order loop can be solved exactly. However because of the one-bit quantizer, such an assumption is not really valid. The problem lies in the second order effects coming from non-idealities of the circuits, such as the opamp finite-gain effect and the capacitor non-linear voltage dependency. Exact solution without assuming that the noise power is white does exist for the first order sigma-delta modulator, and is used to explain and justify the use of the dither signal. This solution can be used to predict the noise power spectrum, and hence noise power, as a function of the input signal and frequency. However even this model becomes exceptionally complicated when we add all the non-idealities outlined above. For higher order loop no closed form solution for the spectrum has been reported. Therefore if one is to investigate the effects of circuit parameters on the noise and distortion behaviors of the circuit, computer simulation is the next best thing. Such a package has been written at Berkeley¹ and modified for the present design. It consists basically of 2 programs: mod2.c and fil1.c. mod2.c accepts a frequency and amplitude input for a sinewave and generates the one-bit code output for both the first order and the second order loop. It also contains the options for checking the effects of non-idealities such as opamp-finite gain, opamp linear settling, comparator hysteresis. Fil1.c is just the program to simulate the FIR decimation filter. Together they

are used to check the overall system design.

The input is a command line:

```
mod2 -q 3200000 -A -3 -k 1000 -f 0.06375 -n 256000 | fil1 -d -r 256 | fil3 -d -r 2 | pcm1p51 | snr1 > snr.out
```

The output is a file called snr.out:

SNR computation using 400 samples:

Total energy 6.92643e+17; mean energy 1.73161e+15; RMS amplitude 4.16126e+07

Energy at frequency 0.1275 is 6.92643e+17; remainder is 3.89193e+09

Signal-to-noise ratio at specified frequency is 82.503441 dB

A.2 Filter Design

Extensive cad tools have been used for the design of both the FIR and the IIR filter.

A.2.1 Fwfir

For the FIR filter a program Fwnew which is a modified version of Fwfir² is first used to generate various windows as inputs to the fil1.c to check and make sure that other window functions (such as hanning, kasier) do not outperform the triangle window. Fwfir is a program that can be used to design FIR digital filters using the window method. The program is capable of designing lowpass, bandpass, bandstop, and highpass filters for both even and odd values of N, (the impulse response duration in samples), using either a rectangular, a triangular, a Hamming, a Hanning, a Chebyshev, or a Kaiser window.

Algorithm

The program uses the well known method of window design for FIR digital filters. If we denote the N-point window as $w(n)$, for $0 < n < N-1$, and we denote the impulse response of the ideal digital filter (obtained as the inverse Fourier transform of the ideal frequency response of the filter) as $h(n)$, $-\infty < n < \infty$, then the windowed digital filter is given as

$$\hat{h}(n) = \begin{cases} w(n)h(n) & 0 < n < N-1 \\ =0 & \text{otherwise} \end{cases} \quad (1)$$

In the discussions above it is assumed that $h(n)$ incorporates an ideal delay of $(N-1)/2$ samples, and that $w(n)$ is symmetric around the point $(N-1)/2$. In order to design windowed digital filters requires specifications of

1. type of ideal filter ie. lowpass, highpass, bandpass or bandstop filter
2. filter cutoff frequencies
3. type of window ie. e, rectangular, triangular, Hamming, Hanning, Chebyshev, Kaiser.
4. Window duration in samples.

From the filter specifications the sequences $h(n)$ and $w(n)$ of eq (1) are computed, and the windowed filter is obtained as the final output.

A.2.2 Filsyn

The design of the IIR window starts with Filsyn which generates a 4th order elliptic filter to meet the filter specs (the D3 specs). Filsyn is a general filter synthesis program that is used to generate filter of different kinds (FIR, IIR..etc) of various forms (cascade, ladder, lattice). It also generates Fortran program to simulate the filter that has been generated. It has some analysis capabilities as far as pole, zero pairing and noise analysis. However in our case the IIR filter needed to compensate for the inband droop of the FIR filter. The droop is on the order of 1db. Filsyn does not have the appropriate feature for generating such a filter. So we start from the prototype design from Filsyn and use it as an input to another program, Optiir.

A.2.3 Optiir

The purpose of this program ² is given the cascade realization of a digital filter transfer function, this program searches for the proper coefficients until the performance meets the arbitrary frequency- domain specifications on the magnitude. The response is calculated as insertion loss in dB.

Algorithm

The method solves the design problem by working directly with the coefficients of the transfer function

$$H(z^{-1}; \alpha) = \alpha_0 \prod \frac{\alpha_{1i} z^{-2} + \alpha_{2i} z^{-1} + 1}{\alpha_{3i} z^{-2} + \alpha_{4i} z^{-1} + 1}$$

where $\alpha = [\alpha_0, \alpha_{11}, \alpha_{21}, \dots, \alpha_{4n}]^t$, $z^{-1} = e^{-sT}$ and T is the sampling period. Note that $N=4n$ is the total number of coefficients in n second-order sections. For a given sampling rate, the system is completely defined if n and α are known. This initial design is obtained by Filysn.

Let $L(\omega; \alpha)$ be a function of the filter on which insertion loss specifications are imposed and $l_1(\omega), l_2(\omega)$ be two given real-valued functions of ω defined, respectively on Ω_1 and Ω_2 , two not necessarily disjoint intervals on the frequency axis. Then, all the common filter or equalizer specifications can be expressed by requiring the following inequalities to be satisfied:

$$f_1(\omega; \alpha) \equiv L(\omega; \alpha) - l_1(\omega) \geq 0, \omega \text{ is a subset of } \Omega_1$$

$$f_2(\omega; \alpha) \equiv l_2(\omega; \alpha) - L(\omega) \geq 0, \omega \text{ is a subset of } \Omega_2$$

(2)

A vector α of a given dimension which satisfies the inequalities belong to the specification set S ,

$$S \equiv \alpha | f_1(\omega; \alpha) \geq 0, \omega \text{ is a subset of } \Omega_1; f_2(\omega; \alpha) \geq 0, \omega \text{ is a subset of } \Omega_2 \quad (3)$$

A stable digital filter for which α is a subset of S will be called acceptable, the philosophy being that a solution (not necessarily a global optimum) satisfies the specifications. There are no constraints on stability. Since we are dealing with insertion loss only, if a final design is unstable, it is made stable by inversion of the poles which lie inside the unit circle in the z^{-1} plane.

The specification set S is defined in (3) through an infinite number of constraints. Because of the discrete nature of the computation, the user must select a finite number of discrete frequencies ω_k in Ω_1 and Ω_2 . The program works with the discretized specification set

$$S_d \equiv \alpha | f_1(\omega_k; \alpha) \equiv L(\omega_k; \alpha) - l_1(\omega_k) \geq 0$$

$$f_2(\omega_k; \alpha) \equiv l_2(\omega_k) - L(\omega_k; \alpha) \geq 0 \quad (4)$$

The optimality criterion adopted is of the following min-max character: If a filter of a given order is acceptable, then maximize the minimum amount by which it exceeds the specifications; if it is not acceptable, then minimize the maximum amount by which it fails.

The input for filter IIR4 in chapter 4 is given below. The explanation of the parameter will be given in the references.²

6 0.62500000e-04 0.48525318e-01
0.10000000e+01 1.53996212e+00 2.10645563e-01 -5.42127072e-01
0.10000000e+01 4.53717123e-01 7.45440818e-01 -3.29742556e-01
0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00

6

1 5 9 10 11 12

1 16

0.10000000e-02 0.50000000e+03 0.10000000e+04 0.15000000e+04
0.20000000e+04 0.25000000e+04 0.30000000e+04 0.32500000e+04
0.33906250e+04 0.40000000e+04 0.46000000e+04 0.50430803e+04
0.54222495e+04 0.61329350e+04 0.67815157e+04 0.73711900e+04

16

-0.00000000e+00 -0.05400000e+00 -0.19600000e+00 -0.44100000e+00
-0.78500000e+00 -1.23100000e+00 -1.77900000e+00 -2.02400000e+00
-1.96000000e+00 0.13000000e+02 0.29500000e+02 0.29500000e+02
0.30500000e+02 0.30500000e+02 0.30500000e+02 0.30500000e+02
0.10000000e+01 0.10000000e+01 0.10000000e+01 0.10000000e+01
0.10000000e+01 0.10000000e+01 0.10000000e+01 0.10000000e+01
0.10000000e+01 0.10000000e+01 0.10000000e+01 0.10000000e+01
0.10000000e+01 0.10000000e+01 0.10000000e+01 0.10000000e+01

1 9

0.17500000e+00 0.12100000e+00 -0.02100000e+00 -0.26000000e+00
-0.61000000e+00 -1.05600000e+00 -1.60400000e+00 -1.84900000e+00
-0.96000000e+00
0.10000000e+01 0.10000000e+01 0.10000000e+01 0.10000000e+01

0.10000000e+01 0.10000000e+01 0.10000000e+01 0.10000000e+01

0.10000000e+01

0

Below is the output that is generated. Note the number of iterations that it takes to converge. Also on each iteration the parameter ALPHA decreases in value, indicating a successful convergence.

0INITIAL COEFFICIENTS FOR R= 0.75529886e-06

A2	A1	B2	B1
----	----	----	----

0.10000000e+01	0.15011772e+01	0.21473292e+00	-0.37911814e+00
----------------	----------------	----------------	-----------------

0.10000000e+01	0.39352062e+00	0.74247873e+00	-0.33545986e+00
----------------	----------------	----------------	-----------------

0.	e+00	0.	e+00	0.	e+00	0.	e+00
----	------	----	------	----	------	----	------

GAIN= 0.13837500e+00 ALPHA=-0.47539841e-01 FUNCTION=-0.46717606e-01

0FINAL COEFFICIENTS FOR R= 0.75529886e-06

A2	A1	B2	B1
----	----	----	----

0.10000000e+01	0.15011230e+01	0.21452147e+00	-0.37727240e+00
----------------	----------------	----------------	-----------------

0.10000000e+01	0.39339924e+00	0.74198270e+00	-0.33497337e+00
----------------	----------------	----------------	-----------------

0.	e+00	0.	e+00	0.	e+00	0.	e+00
----	------	----	------	----	------	----	------

GAIN= 0.13864194e+00 ALPHA=-0.52171029e-01 FUNCTION=-0.49907714e-01

0MODIFIED COEFFICIENTS

0.10000000e+01	0.15011230e+01	0.21452147e+00	-0.37727240e+00
----------------	----------------	----------------	-----------------

0.10000000e+01	0.39339924e+00	0.74198270e+00	-0.33497337e+00
----------------	----------------	----------------	-----------------

0.	e+00	0.	e+00	0.	e+00	0.	e+00
----	------	----	------	----	------	----	------

MODIFIED GAIN= 0.13864194e+00

0GRADIENTS

0.	e+00	-0.19083640e-01	0.10682638e-01	0.36683697e-01
----	------	-----------------	----------------	----------------

0.	e+00	-0.51106930e-01	-0.63481979e-01	0.91556862e-01
----	------	-----------------	-----------------	----------------

0.	e+00	0.	e+00	0.	e+00	0.	e+00
----	------	----	------	----	------	----	------

96 FUNCTION EVALUATIONS 11 GRADIENT EVALUATIONS

gain function with these coefficient sets and display the feasible coefficient vectors.

Multivariate Search

This extends the univariate search by extending the search space to higher dimension (rather than just a_1 , it does a_1, a_2 together) We concentrated mainly on the univariate search which takes less time. It is found out the multi-variate search takes considerable more time and does not yield any significantly better result.

The following is our input for IIR4 to Candi

MDL DESCRIPTION OF PCM LOWPASS FILTER PCMLP4.I

;pcmlp with peaking for fir: triangular 0..127127..0 fir3 14641

;coefficient taken from output50, specs from input50 of optiir

;x33 or coefficient is giving problems so try to fix it

; with scaling of second one to -9.7db or around 0.316

; basically make x11 and x21 0.316 times smaller

;TOPOLOGY

;FIRST SECTION

ADDER N11 IN=U0 X31 X41 WL=16

ADDER U02 IN=N11 X11 X21 WL=16

DELAY N21 IN=N11 CL=CL NDEL=1 WL=16

DELAY N31 IN=N21 CL=CL NDEL=1 WL=16

MULT X11 IN=N21 VAL=0.0474354E+01 OPT

MULT X21 IN=N31 VAL=0.0316000E+01 OPT

MULT X31 IN=N21 VAL=0.3772724E+00 OPT

MULT X41 IN=N31 VAL=-0.21452147E+00 OPT

;

;SECOND SECTION

ADDER N12 IN=U02 X32 X42 WL=16

ADDER YOUT IN=N12 X12 X22 WL=16

```
;  
DELAY N22 IN=N12 CL=CL NDEL=1 WL=16  
DELAY N32 IN=N22 CL=CL NDEL=1 WL=16  
;  
MULT X12 IN=N22 VAL=-0.39339924E+00 OPT  
MULT X22 IN=N32 VAL=-0.1000000E+01 OPT  
MULT X32 IN=N22 VAL=-0.33497337E+00 OPT  
MULT X42 IN=N32 VAL=-0.7419827E+00 OPT  
;  
;INPUT EXCITATION FOR FREQUENCY ANALYSIS  
VIN UIN 0 ACMAG=1  
ADC U0 IN=UIN VAL=1.0 CL=CL  
;OUTPUT NODE OF ADC IS ASSUMED TO BE FILTER INPUT  
;  
;DEFINE FILTER OUTPUT NODE USING 'PRINT' COMMAND  
PRINT YOUT  
;  
;DEFINE CLOCK INPUT SOURCE FOR DELAYS AND ADC  
INPUT CL CLOCK=0.0125 0.025 CYC  
;  
;SPECIFICATION OF FREQUENCY CONSTRAINTS FOR CANDI  
;TOTAL FREQUENCY POINTS 7  
;PASS-BAND 7 POINTS  
;TRANSITION BAND 2 POINTS  
;STOP-BAND 7 POINTS
```

A.2.5 Digest

Since the coder needs to have an idle channel noise that is 80db down from the full scale, the noise and the limit cycle contribution from the IIR filter cannot exceed this specification. To check this Digest³, is used to analyze the limit cycle and noise contribution from the IIR filter. For the FIR filter, since there is no truncation of the coefficients there is no roundoff noise.

A.2.6 Demon

Finally we simulate the IIR filter behavior as implemented in the hardware directly. Demon is the hardware description language that simulates the microcode on the hardware. We combine Demon with Mod2.c and Fil1.c and simulate the overall system behavior by doing the following tests:

a) calculate the SNR versus amplitude for sine wave of different frequencies. b) Frequency response of the whole system.

A.3 Automatic layout tools

The IIR filter is generated by the automatic layout tool Lager.⁴ First of all TimLager then comes up with all the basic macrocells and the necessary interconnections.

A.3.1 Flint

Flint is then used to allow the designer to manually place the individual macrocells and do the automatic routing. The FIR filter which is generated using a custom layout is treated as a macrocell and the designer makes use of the automatic routing mechanism to route the I/O between the IIR and the FIR. Similarly the analog front end is manually placed and automatically routed to the FIR.

A.4 Layout Check

The checking on the whole chip is separated into the digital and analog portions. The digital part is mainly checked for logic and the program Esim is used. The analog part is checked using Spice and the AC and Transient response of the circuits from the extracted file are compared to the original simulation.

1) Spice checking:

It turns out that the extracted gain is roughly the same as obtained in the original Spice simulation. The extracted settling time is around 120ns rather than the 75ns as obtained in the original Spice

simulation. This is due to the extraction model. The extraction tool takes lumped capacitance rather than voltage dependent capacitance as in Spice. Therefore in general the capacitance is larger and the settling time is longer.

2) esim checking:

The digital circuits are checked by Esim and the correct functional behavior is verified.

References

1. Hauser, *Simulation package for sigma-delta coders.*
2. IEEE ASAP society, *Programs for digital signal processing.*
3. DeMan, "Custom Design of a VLSI PCM-FDM transmultiplier," *Journal of Solid-State Circuits*, pp. 73-86, February, 1986.
4. Neff, *Master Report, Berkeley, 1987.*