AN MOS-LSI ADAPTIVE LINEAR PREDICTION FILTER FOR SPEECH PROCESSING

by

R. D. Fellman

Memorandum No. UCB/ERL M82/82

8 November 1982

AN MOS-LSI ADAPTIVE LINEAR PREDICTION FILTER FOR SPEECH PROCESSING

by

Ronald David Fellman

Memorandum No. UCB/ERL M82/82

8 November 1982

ELECTRONICS RESEARCH LABORATORY

College of Engineering
University of California, Berkeley
94720

An MOS-LSI Adaptive Linear Prediction Filter for Speech Processing

Ronald David Fellman

Ph. D. EECS

Robert W. Brodersen
Chairman of Committee

ABSTRACT

An approach to integrating an adaptive linear prediction filter on a single integrated circuit will be described. Implementation of the Lattice Linear Predictive Coding (LPC) Algorithm (Itakura & Saiko,Kang) for speech processing is first investigated. Then the tradeoffs between analog and digital techniques are discussed and an optimal configuration is presented. In order to fabricate a low cost, single chip realization of the adaptive filter, an analog/digital compatible CMOS process is presented. The design of an integrated circuit which implements the linear prediction speech analysis filter is then described. Experimental results are presented on the integrated circuit fabricated at U.C.Berkeley which show adequate performance for speech processing applications.

Research sponsored by
Defense Advanced Research Projects Agency (DoD)
ARPA Order No. 3793.

Under Contract No. MDA903-79-C-0429 issued by
Department of Army, Defense Supply Service - Washington,
Washington, DC 20310.

TABLE OF CONTENTS

CHAPTER 1

Introduction

1.1. Speech Processing and Applications

A signal processing system that can compress speech by removing unnecessary and redundant information is an essential component in many applications which involve the human voice. With condensed speech information readily available, a myriad of voice operated products becomes feasible and existing products can become more efficient. A telephone line can transmit many times more conversations concurrently because a processed speech signal uses less bandwidth than a raw speech signal. Electronic storage of speech becomes more efficient since processed speech uses less memory space. More words can be stored and each word can be accessed more rapidly. Automatic word recognition systems can handle more words and operate faster since non-essential speech information is discarded and not stored or processed. Speaker identification systems also become more efficient since the essential information has already been extracted.

Automatic word recognition systems and vocoders for low bit-rate voice transmission have not been widely used due to the high cost of the speech processing hardware. Currently, most speech processing systems employ either a high-speed digital signal processing system using bipolar bit slice arithmetic logic units and costly parallel multiplier circuitry, or a channel bank filter using up to 19 analog band-pass filters. Both techniques require many costly electronic components and a significant amount of labor to construct.

The object of this research has been to investigate techniques which would allow implementation of a complete speech processing system on a single inexpensive MOS (metal-oxide-semiconductor) integrated circuit. Such a system would function to remove any redundant information in the speech waveform producing a condensed representation of the original speech. With the Lattice LPC (linear predictive coding) technique described in chapter 2, the bit-rate required to represent human speech can be reduced by a factor of about 30 over conventional digital sampling methods while retaining the fidelity of the original speech.

With this integrated circuit, many new products using speech as input or output could be produced inexpensively. Inexpensive word recognition for consumer applications, secure speech transmission, and speaker identification systems are but a few of the products that could become available at low cost.

There are many approaches to processing speech to obtain a low bit-rate representation. This research focused on the Lattice method of Linear Predictive Coding. There are several advantages to this algorithm which lend themselves to monolithic implementation. These advantages, as well as the disadvantages, will be discussed at length in chapter 2.

1.2. A Speech Model

The speech processor condenses the speech signal by modeling the human voice process with an excitation generator that stimulates a time varying filter (figure 1.1). The excitation generator models the action of the larynx while the filter represents the response of the vocal tract to this stimulation. Air flowing through the larynx can cause the glottis, loose flaps of skin on either side of a narrow opening, to vibrate producing bursts of pressure at regular intervals. The period between these bursts is called the pitch and it can vary from word to word but stays relatively constant during a particular utterance. This vibrating air

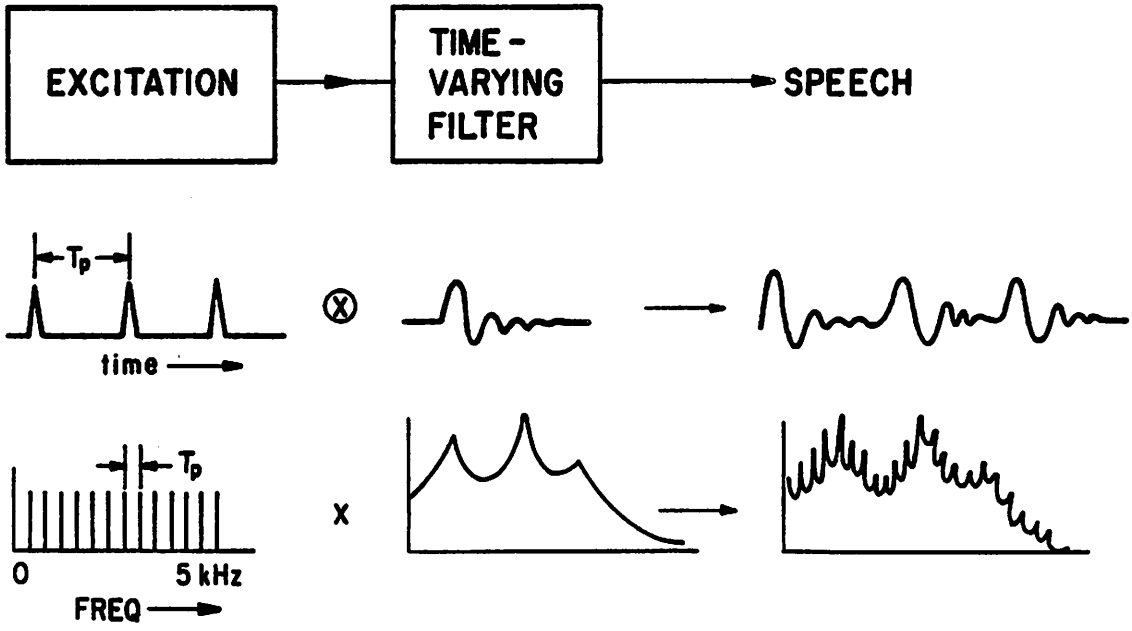


Figure 1.1. Basic speech model

then travels through the vocal tract stimulating resonances as it travels through the throat, past the nasal cavity and out the mouth (figure 1.2). The vocal tract acts as a non-uniform resonant cavity with reflections and losses occurring throughout. Its response can be modeled by the zero and pole locations of a slowly time-varying filter. Like pitch, these filter parameters stay relatively constant during the production of a particular sound. Sounds modeled by exciting this filter with periodic bursts of pressure are called voiced sounds.

Other sounds in human speech are better modeled using a noise source to excite the vocal tract filter. Sounds such as /t/ and /s/ are produced by a rush of air through the mouth. They are called unvoiced sounds. For simplicity we use the same vocal tract model as before but replace the excitation source with a noise generator.

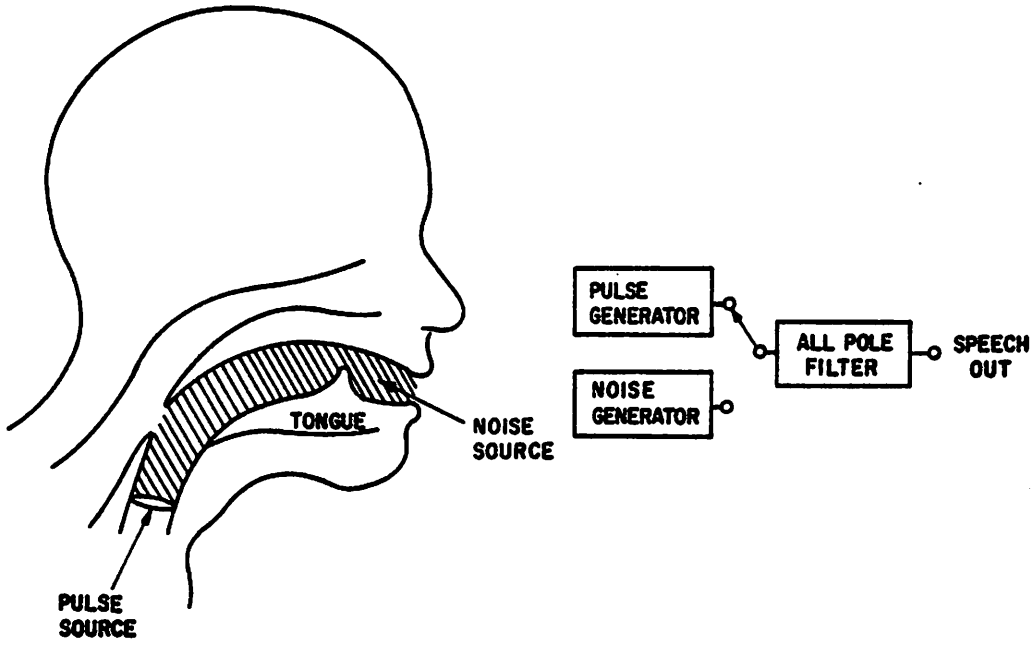


Figure 1.2. Physical model of the human speech process

In the most general case, the vocal tract filter model is most accurately implemented using both poles and zeros. But because of the difficulty in analyzing speech with arbitrary pole and zero locations, some simplifications need to be made. Usually, zeros in the filter response are due to the interaction of the nasal cavity with the vocal tract as with the sounds /m/ and /n/. Empirically, it has been determined that an acceptable filter model can be produced by using only poles since the zeros can be adequately approximated by a series expansion of poles.

Each resonance, or peak, in the spectrum of a segment of speech is called a formant. A formant can be equivalently represented by a complex pole-pair. Listening tests on synthesized speech [12][41] have demonstrated that the first three formants are the most important. For an average vocal tract length of 17cm. the first three formant frequencies are contained in a frequency interval of

about 250 Hertz to 2800 Hertz [31]. Since the speed of sound is roughly 34cm/sec, there is a delay of .5msec for an impulse originating at the glottis to travel to the lips. After an excitation pulse the glottis remains closed while the lips are open, thus, a standing wave in the vocal tract has a quarter wavelength of .5msec with a fundamental frequency of about 500 Hertz. The first formant is usually near this frequency with successive formants, if they exist, nominally separated by 1 KHz., due to the 1msec delay from the glottis to the lips and back. A minimum of 2 poles are therefore needed for each kilohertz of speech bandwidth, but usually two additional poles are used when modeling the vocal tract to help model zeros, compensate for distortion in the antialias filter, and to compensate for lip radiation and glottal effects. So for a 4000 Hertz speech bandwidth a ten pole vocal tract model is used.

The effects of the glottis and lips on the vocal tract model are slight. The lips are usually modeled with a low frequency zero while for voiced speech the glottis is typically represented by two low frequency poles. Their effects partially cancel to allow representation as a single low frequency pole at a fixed frequency. To compensate for this, the speech to be analyzed can first be processed in a high pass filter with a fixed zero near 100 Hertz. This pre-conditioning of the speech signal before being analyzed is called pre-emphasis and it can increase computational accuracy and efficiency.

CHAPTER 2

The Lattice LPC Algorithm

2.1. Introduction

In chapter one we have seen what a suitable model for human speech should be composed of, but the method by which speech can be analyzed for determining the parameters for this model has not yet been described. Any such algorithm should be simple enough to perform the parameter extraction in "real time" (as the speaker is talking) with no noticeable delay. It must be amenable to implementation using only a few low cost parts. And it must give an accurate enough representation of the speech so that the speech can be resynthesized with no significant degradation in fidelity.

There are many algorithms in the literature [39] that satisfy the last requirement. Among them are : the Belgard (Channel Bank) Vocoder, the Autocorrelation (Levinson-Durbin) LPC Algorithm, and the Covariance Matrix LPC Algorithm. This thesis will confine itself to the Lattice LPC Algorithm because of its many desirable characteristics which, as we shall see, allow it to satisfy all of the above requirements.

Lattice Linear Predictive Coding (Lattice LPC) models the vocal tract resonances by fitting an all-pole transfer function to the vocal tract transfer function. The model has the following form (in z-transform notation) :

$$H(z) = \frac{1}{1 - \sum_{i=1}^{p} a_i z^{-i}} . \qquad (2.1)$$

The model parameters (LPC coefficients) are the a_i's. The number of poles (p) used in the model is typically nine to twelve for bandwidths from 3500 Hertz to 5000 Hertz. More poles improve the accuracy, but a minimum number of poles is desired to obtain the maximum efficiency of representation. Typically, two poles are used for each kilohertz in bandwidth with two additional poles used to improve accuracy from second order effects. Thus, $p = Bandwidth / 500Hz + 2$.

There are p previous samples of speech which are used in a weighted linear combination with the a_i's to predict the next sample of speech. This predicted speech sample, added to the excitation source of the speech model produces the next sample of speech. The LPC coefficients and excitation parameters are updated periodically as the vocal tract changes to produce words, usually between 10 and 20 msec. A block diagram of this predictor is shown in figure 2.1.

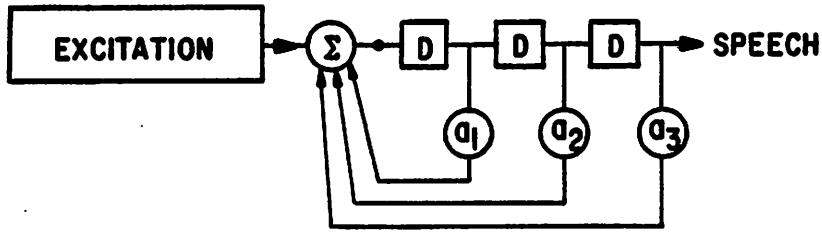


Figure 2.1. All-pole speech model using a linear predictor

The inverse of this filter can be used for analysis of a speech waveform in order to compute the LPC model parameters. The analysis filter is an all-zero filter with the following transfer function:

$$H(z) = 1 - \sum_{i=1}^{p} a_i z^{-i} \quad . \tag{2.2}$$

Using the speech waveform as the input, this filter produces an output corresponding to the excitation of the all-pole synthesis filter. This signal is termed the residual error. It can be used directly in a high quality vocoder scheme as in a Baseband Vocoder [31], or it can be the basis for extracting pitch parameters in a low bit-rate vocoder.

The Lattice LPC Algorithm, as first described by Itakura and Saito [23] and later modified by Burg [9], has been shown to produce a very accurate model of human speech [25]. The algorithm has some very close analogies to the actual physics of the traveling sound pressure waves in the vocal tract. For simplicity, the human vocal tract is modeled as a cascade of equal length lossless acoustic tubes of different diameters each varying independently with time. Since we are modeling the action of the muscles in controlling the shape of the vocal tract to produce the sound, and not the sound waveform itself, the rate at which these diameters change is proportional to the muscle response time which is several orders of magnitude slower than changes in the actual speech waveform. This is the basis for the reduction in the data rate required to represent speech.

In order to use this model to analyze speech we start with the sound produced at the lips and work our way back through the vocal cavity, modeling the reflections at discontinuities in such a way as to end up at the larynx with the impulse response or white noise source that models the excitation source. The reflection coefficients thus obtained are directly analogous to the filter pole locations. While it is true that the actual excitation source in speech is not a true

impulse or noise source, the parameters extracted by this method include this deviation from reality in the vocal tract filter response.

In the Lattice LPC Algorithm, the LPC coefficients are not calculated explicitly. But, the reflection coefficients which are produced are related to the LPC coefficients by the following simple recursive relation known as the Levinson-Durbin Recursion [11].

$$a_i^{(i)} = k_i \tag{2.3a}$$

$$a_j^{(i)} = a_j^{(i-1)} + k_i a_{i-j}^{(i-1)} \tag{2.3b}$$

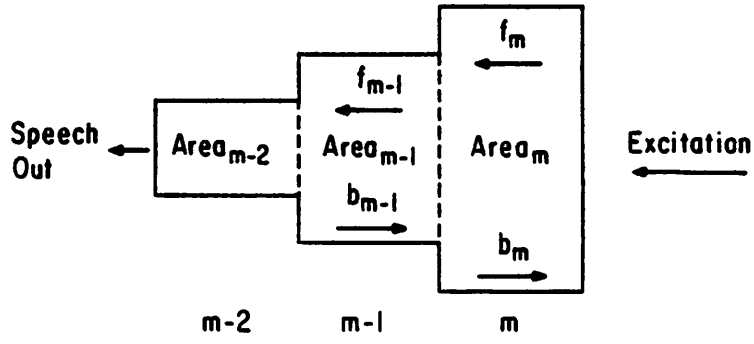
for $1 \leq j \leq i-1$.

These equations are solved recursively for i = 1,2,....,p with the final solution being $a_i = a_i^{(p)}$ for $1 \leq j \leq p$.

2.2. Derivation of the Lattice LPC Algorithm

The Lattice LPC algorithm can be derived directly from physical considerations by modeling the vocal tract as a cascade of equal length lossless acoustic tubes and using conservation and continuity constraints at the juncture between adjacent tubes [3][31]. The cross-sectional area of each tube is assumed to vary very slowly with time compared to the traveling sound waves so that the diameters can be treated as constants. We will number the tubes m=0,....,p where p is the total number of tubes in our vocal tract model and m=0 is the tube nearest the lips. The area for tube m will be A_m, the length of each tube will be l, and v_s will be the velocity of sound. A diagram of this acoustic tube model is shown in figure 2.2.

A traveling sound wave is comprised of two components: a traveling pressure wave which is the differential pressure change induced by the sound wave as it travels over a constant background pressure, and the particle velocity which is a measure of the speed with which the air particles are compressed or relaxed by



$$\tau/2 = \frac{\ell_{tube}}{v_{sound}} \qquad r_m = \frac{Area_{m-1} - Area_m}{Area_{m-1} + Area_m}$$

$$f_{m-1}(t+\frac{\tau}{2}) = (1-r_m) f_m(t) - r_m b_{m-1}(t-\frac{\tau}{2})$$

$$b_m(t) = r_m f_m(t) + (1+r_m) b_{m-1}(t-\frac{\tau}{2})$$

Figure 2.2. Acoustic tube model of the human vocal tract

the sound wave.[19]

In our speech model a forward traveling sound wave in tube m, $f_m(x-v_s t)$, is created by the excitation source. This, in turn, creates a backward traveling sound wave, $b_m(x+v_s t)$, as it encounters the discontinuity between tubes m and m-1 at x_m in our model. The transmitted fraction of wave $f_m(x-v_s t)$ travels in tube m-1 as $f_{m-1}(x-v_s t)$. Continuity of pressure in figure 2.2 requires that the total pressure on either side of this discontinuity be equal while conservation of mass requires that the flow of air out of tube m equal the flow of air into tube m-1. Matching continuity and conservation constraints at x_m gives us the following two equations.

$$f_m^p(x_m - v_s t) + b_m^p(x_m + v_s t) = f_{m-1}^p(x_m - v_s t) \tag{2.4a}$$

$$\rho_0 A_m \left[f_m^v(x_m - v_s t) + b_m^v(x_m + v_s t) \right] = \rho_0 A_{m-1} f_{m-1}^v(x_m - v_s t) \tag{2.4b}$$

The superscript of p or v denotes the pressure or particle velocity component of

the traveling sound wave, respectively. The factor ρ_0 is the density of air.

From Newton's Second Law, *Force = mass · acceleration*, we can derive the following relationships between the pressure and particle velocity for the traveling wave:

$$f_m^v(t) = f_m^p(t)/\rho_0 v_s \text{ and } b_m^v(t) = -b_m^p(t)/\rho_0 v_s \tag{2.5}$$

Combining equations 2.4 with equations 2.5 yields the reflection coefficient for the pressure wave at this juncture as $r_m = \frac{b_m^p}{f_m^p} = \frac{A_m - A_{m-1}}{A_m + A_{m-1}}$. Dividing equation 2.4 by $f_m^p(x_m - v_s t)$ the transmission coefficient is seen to be $1+r_m$. The backward traveling wave has a reflection coefficient of $-r_m$ and a transmission coefficient of $1-r_m$ at x_m since it encounters the discontinuity from the opposite direction.

Referring to figure 2.2 we note the following relationships for $f_{m-1}^p(x_m - v_s t)$ and $b_m^p(x_m + v_s t)$.

$$f_{m-1}^p(x_m - v_s t) = (1+r_m) f_m^p(x_m - v_s t) - r_m b_{m-1}^p(x_m + v_s t) \tag{2.6a}$$

$$b_m^p(x_m + v_s t) = (1-r_m) b_{m-1}^p(x_m + v_s t) + r_m f_m^p(x_m - v_s t) \tag{2.6b}$$

Rearranging terms and substituting allows us to rewrite these equations for the wave functions in tube m in terms of the wave functions in tube m-1.

$$(1+r_m) f_m^p(x_m - v_s t) = f_{m-1}^p(x_m - v_s t) + r_m b_{m-1}^p(x_m + v_s t) \tag{2.7a}$$

$$(1+r_m) b_m^p(x_m + v_s t) = b_{m-1}^p(x_m + v_s t) + r_m f_{m-1}^p(x_m - v_s t) \tag{2.7b}$$

We will now define τ as the time for a sound wave to travel across one section of tube and back so that $\tau/2 = l/v_s$. Noting that $x_m = x_{m-1} + l$ we can rewrite the x-coordinate of the m-1 terms as x_{m-1} and regroup l in terms of $\tau/2$.

$$(1+r_m) f_m^p(x_m - v_s t) = f_{m-1}^p(x_{m-1} - v_s(t + \tfrac{\tau}{2})) + r_m b_{m-1}^p(x_{m-1} + v_s(t + \tfrac{\tau}{2})) \tag{2.8a}$$

$$(1+r_m) b_m^p(x_m + v_s t) = b_{m-1}^p(x_{m-1} + v_s(t + \tfrac{\tau}{2})) + r_m f_{m-1}^p(x_{m-1} - v_s(t + \tfrac{\tau}{2})) \tag{2.8b}$$

Now we can simplify our notation to only include the time dependence if we fix the x coordinate at the right hand side of a tube. Shifting the time coordinate by $-\tau/2$ gives:

$$(1+r_m)f_m^p(t-\frac{\tau}{2})=f_{m-1}^p(t)+r_m b_{m-1}^p(t-\tau) \tag{2.9a}$$

$$(1+r_m)b_m^p(t-\frac{\tau}{2})=b_{m-1}^p(t-\tau)+r_m f_{m-1}^p(t). \tag{2.9b}$$

The delay of $\tau/2$ on the left side of equations 2.9 just represents the propagation delay for the sound wave to travel from one tube to the next and can be ignored. The factor $(1+r_m)$ represents a normalizing factor at each stage of the model and can be dropped if we allow for non-unity overall filter gain. The resulting vocal tract filter power gain will be derived in section 2.4. The resulting algorithm in terms of traveling sound pressure waves in z-transform notation becomes:

$$F_m^p(z)=F_{m-1}^p(z)+r_m z^{-1}B_{m-1}^p(z) \tag{2.10a}$$

$$B_m^p(z)=z^{-1}B_{m-1}^p(z)+r_m F_{m-1}^p(z). \tag{2.10b}$$

At the lips the vocal tract is open to the atmosphere so the pressure there is held nearly constant and the sum of the forward and backward traveling sound pressure waves equals zero (ignoring losses) - although the total particle velocity is at a maximum [16]. The boundary condition at the lips is: $B_0^p(z) = -F_0^p(z)$. By rewriting the algorithm in terms of particle velocity traveling waves using equations 2.5 we get a simpler boundary condition, $F_0^v(z) = B_0^v(z)$.

To complete the derivation we identify $k_m = -r_m$ as the PARCOR (partial correlation) coefficients of Itakura and Saito [23] to obtain the Lattice LPC analysis algorithm as follows:

$$F_m(z) = F_{m-1}(z)+k_m z^{-1}B_{m-1}(z) \tag{2.11a}$$

$$B_m(z) = z^{-1}B_{m-1}(z)+k_m F_{m-1}(z) \tag{2.11b}$$

(The superscripts denoting particle velocity have been dropped.) A block diagram of this algorithm is shown in figure 2.3. In computing the z-transform of equations 2.9 we implicitly set the sampling rate equal to $1/\tau$ so that the delay element of figure 2.3 (z^{-1}) is a delay of one sample.

For a model of p poles (equation 2.1), p stages of the form of figure 2.3 would be cascaded. By constraining the left side of figure 2.3 to be equal to the speech waveform which is being analyzed, the right side of figure 2.3 should become the excitation for the vocal tract filter. This estimate of the excitation is the residual error of the lattice filter in equation 2.2.

2.3. Linearity of the Lattice algorithm

The Lattice algorithm derived in the previous section is linear with respect to the forward and backward residuals. We will prove this by setting the residual

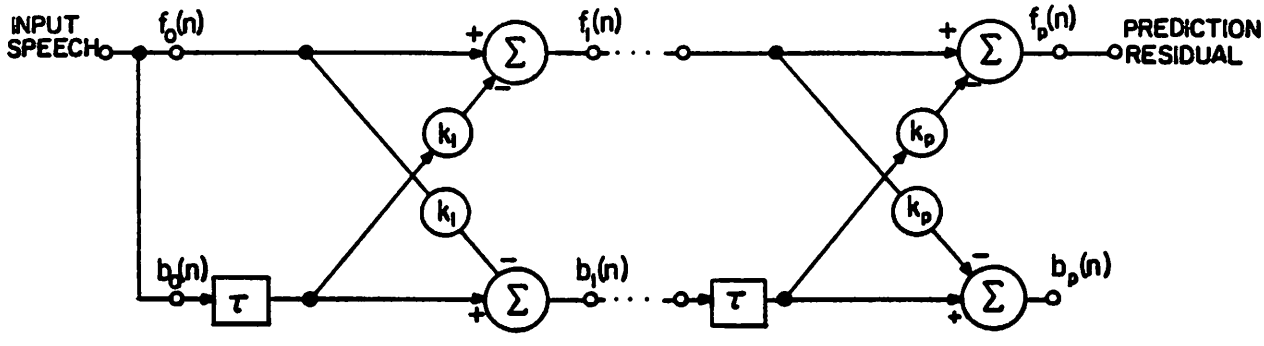


Figure 2.3. Block diagram of a multi-stage Lattice LPC analysis filter

inputs to linear combinations of two signals, each multiplied by a scalar (α and β). A function is then linear if its output can be expressed by the same linear combination as the input. For inputs $f_{m-1}(n)$ and $b_{m-1}(n)$ to the Lattice algorithm we will substitute the following linear combinations.

$$f_{m-1}(n) = \alpha \cdot {}^1 f_{m-1}(n) + \beta \cdot {}^2 f_{m-1}(n)$$
$$b_{m-1}(n) = \alpha \cdot {}^1 b_{m-1}(n) + \beta \cdot {}^2 b_{m-1}(n)$$

This yields:

$$\begin{aligned} f_m(n) &= \left[\alpha \cdot {}^1 f_{m-1}(n) + \beta \cdot {}^2 f_{m-1}(n)\right] + k_m \left[\alpha \cdot {}^1 b_{m-1}(n-1) + \beta \cdot {}^2 b_{m-1}(n-1)\right] \\ &= \alpha\left[{}^1 f_{m-1}(n) + k_m {}^1 b_{m-1}(n-1)\right] + \beta\left[{}^2 f_{m-1}(n) + k_m {}^2 b_{m-1}(n-1)\right] \end{aligned}$$

This algebra can be similarly done for $b_m(n)$. Rewriting in terms of $f_m(n)$ and $b_m(n)$ proves the assertion.

$$f_m(n) = \alpha \cdot {}^1 f_m(n) + \beta \cdot {}^2 f_m(n)$$
$$b_m(n) = \alpha \cdot {}^1 b_m(n) + \beta \cdot {}^2 b_m(n)$$

Q.E.D.

One property of linearity is that any scaling, positive or negative, of residual signals within a multi-stage Lattice filter can be lumped together as a single gain factor for the filter. This is true provided that both the forward and backward signal paths are scaled by the same amount at each stage. This fact was used in the Lattice derivation when the factor $(1+r_m)$ was dropped. The resulting power gain for the Lattice analysis filter will be derived later in this chapter. In chapter 4 this property will be cited when a Lattice filter is implemented with a scale factor of -1 at each stage. The negative gain will be shown to reduce the effects of parasitics in a practical implementation.

2.4. The PARCOR coefficients

For the vocal tract analysis filter, the reflection coefficients (or PARCOR coefficients) are unknowns. Although we are given the resulting speech waveform, the actual shape of the vocal tract which produced it is what we wish to

determine. There are many methods of choosing these filter parameters [29], but the most commonly used method is to minimize the sum of the time-averaged forward and backward mean square error residuals at each stage of the lattice filter [9]. This method has the advantage of insuring a stable all-pole vocal tract filter for synthesis with $|k_m| \leq 1$ while being computationally simpler than the partial correlation used by Itakura.

Itakura computed k_m as the negative of the correlation between the backward residual delayed one sample and the forward residual error. The correlation insures that $|k_m| \leq 1$ but requires computing a square root and it does not directly minimize any error criteria. In the sampled time domain these PARCOR coefficients are computed from:

$$k_m^I = -\frac{E[f_{m-1}(n)b_{m-1}(n-1)]}{\sqrt{E[f_{m-1}^2(n)]E[b_{m-1}^2(n-1)]}} \tag{2.12}$$

where $E[\cdot]$ is the statistical expectation function with respect to time (n) [21].

Starting with the Lattice LPC analysis algorithm:

$$f_m(n) = f_{m-1}(n)+k_m b_{m-1}(n-1) \tag{2.13a}$$

$$b_m(n) = b_{m-1}(n-1)+k_m f_{m-1}(n) \tag{2.13b}$$

we wish to minimize the error criterion given by

$$E[f_m^2(n)+b_m^2(n)]\,. \tag{2.14}$$

After first substituting equations 2.13 into into 2.14 we can take the partial derivative with respect to k_m. By setting the result equal to zero and then solving for k_m we can obtain the following expression for k_m.

$$k_m = -\frac{2E[f_{m-1}(n)b_{m-1}(n-1)]}{E[f_{m-1}^2(n)]+E[b_{m-1}^2(n-1)]} \tag{2.15}$$

To prove that k_m computed from equation 2.15 is bounded in magnitude by unity we note that $(f_{m-1}(n)-b_{m-1}(n-1))^2$ is always greater than or equal to zero. For clarity, the time dependence will be dropped so $b_{m-1}(n-1)$ will be replaced by

bd_{m-1} and $f_{m-1}(n)$ will be replaced by f_{m-1}.

$$E[(f_{m-1}-bd_{m-1})^2] = E[f_{m-1}^2]+E[bd_{m-1}^2]-2E[f_{m-1}bd_{m-1}] \geq 0$$

so that $|k_m| \leq 1$.

We can show that $|k_m| \leq |k_m^I| \leq 1$ by expanding $(E[f_{m-1}^2]-E[bd_{m-1}^2])^2 \geq 0$.

$$E^2[f_{m-1}^2]+E^2[bd_{m-1}^2] \geq 2E[f_{m-1}^2]E[bd_{m-1}^2]$$

$$E^2[f_{m-1}^2]+2E[f_{m-1}^2]E[bd_{m-1}^2]+E^2[bd_{m-1}^2] \geq 4E[f_{m-1}^2]E[bd_{m-1}^2]$$

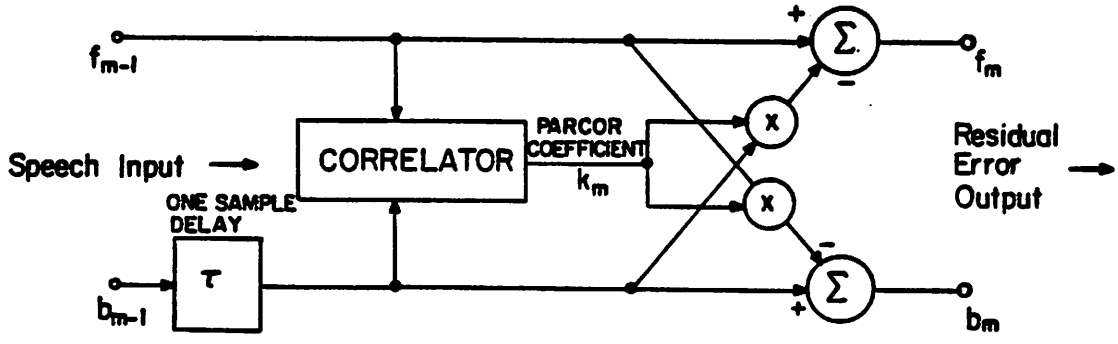
$$\frac{1}{2}\left\{E[f_{m-1}^2]+E[bd_{m-1}^2]\right\} \geq \sqrt{E[f_{m-1}^2]E[bd_{m-1}^2]}$$

The functional block that computes the reflection coefficients is called the correlator, since historically Itakura first computed these coefficients as partial correlations. A block diagram of the Lattice analysis filter structure which includes the correlator is shown in figure 2.4.

2.5. Correlator interfacing

Once the reflection coefficients are found they must be multiplied back into the Lattice filter of figure 2.4. To insure the most accurate possible modeling of the input speech, the reflection coefficients should be computed and used in the Lattice filter during the same sample period. The reflection coefficients must be computed each sample period to accurately adapt to changes in the input speech waveform and to provide the proper input sample rate to the expectation functions in the correlator. This requires computation of the Lattice algorithm of equation 2.11 after computation of the PARCOR equation of Burg (equation 2.15). Since both operations are performed within one sample period, there is less time to perform each operation. For most practical implementations, this amount of time is not sufficient.

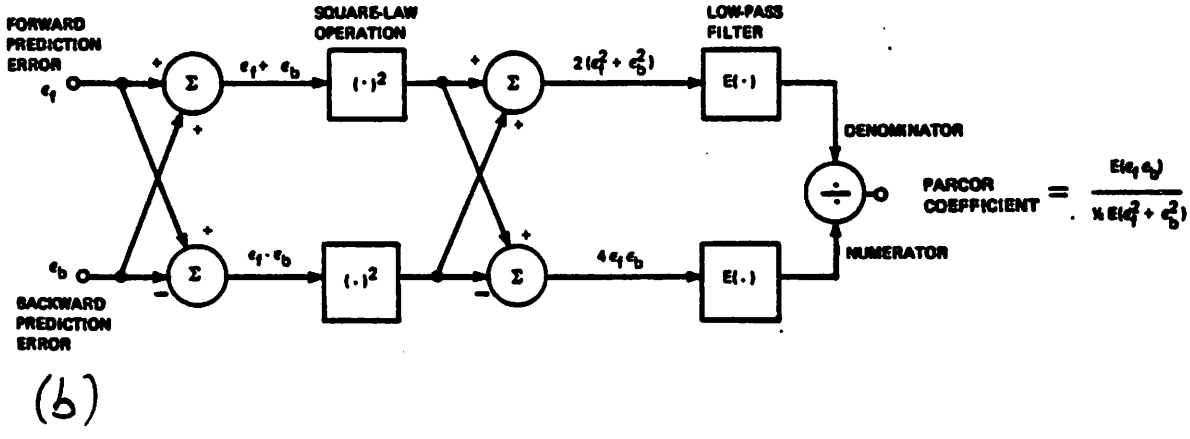
Alternatively, the reflection coefficients from the previous sample period can be used in the lattice filter. This minor modification allows parallel processing of



$$f_m(t) = f_{m-1}(t) - k_m\, b_{m-1}(t-\tau)$$

$$b_m(t) = b_{m-1}(t-\tau) - k_m\, f_{m-1}(t)$$

(a)



(b)

Figure 2.4. (a) Block diagram showing one stage of the Lattice LPC analysis algorithm with correlator attached.
(b) Block diagram of basic correlator algorithm.

both operations. Each operation can then be given a full sample period for computation, which is adequate for practical implementations. The expectation function in the PARCOR computation restricts the reflection coefficients to only slow time variations, and thereby justifies the use of reflection coefficients that are delayed by one sample period. Although, a minor degradation in the accuracy of the modeled speech can be measured.

Transmission of the reflection coefficients as part of the vocoder output should be done at a low bit rate. This will then take advantage of the processing done on the speech signal. The speech model that has been presented makes use of the fact that vocal tract dimensions change very slowly relative to changes in the actual speech waveform. Transmission frame rates of 10 to 20 msec are typically used to model these vocal tract changes. The expectation function in the reflection coefficient computation helps to smooth the time variations in these coefficients to conform to the rate of change of the actual vocal tract parameters that we are modeling. The simplest method of selecting the reflection coefficients for transmission is just to select the last set of coefficients in a frame. However, during the start of a pitch pulse, the sudden change in amplitude of the input speech waveform will render the reflection coefficients at this time inaccurate.

The most accurate method for reflection coefficient selection involves monitoring the power in the forward residual error of the entire multi-stage analysis filter [25]. The most accurate speech model is obtained when the power in the final residual output is at a minimum within a frame. The reflection coefficients should be selected at this point in the frame for transmission.

The residual power becomes maximum at the start of a pitch pulse and decreases during the pitch period as the algorithm adapts to the waveform. By the end of a pitch period, the analysis filter should have adapted to the vocal

tract resonances leaving low-energy noise at the residual output. More complicated vocoder schemes have variable frame lengths which coincide with the end of pitch periods, when the reflection coefficients are transmitted. This is called pitch-synchronous analysis. Variable frame rates can reduce the output bit rate by taking advantage of redundancies in the speech waveform.

Both of these methods require additional circuitry to measure the power in the residual and keep track of the reflection coefficients when this power is at a minimum. Furthermore, pitch-synchronous analysis requires even more complicated circuitry because of the variable frame rate. A good compromise which greatly improves the quality of speech modeling over the simple end-of-frame transmission approach, yet requires much simpler circuitry to implement than the minimum residual methods, is to average the reflection coefficients within a frame. This approach simply requires that a running total of each of the reflection coefficients within a frame are kept and that the resulting sum be divided by the number of samples in a frame (division by a constant). Listening tests have demonstrated that there is little, if any, discernible difference between the accuracy of this method and the minimum residual method.

2.6. Analysis filter gain

We can define a mean-squared gain for the analysis filter as the ratio of the mean-squared values of the residual output to the input speech. For an M-stage Lattice analysis filter, the mean-squared gain is:

$$G^2_{filter} = \frac{E[f_M^2(n)]}{E[f_0^2(n)]} .$$

The gain for the M-stage filter can be written as the product of the gains of each of the individual stages.

$$G^2_{filter} = \frac{E[f^2_M(n)]}{E[f^2_{M-1}(n)]} \cdot \frac{E[f^2_{M-1}(n)]}{E[f^2_{M-2}(n)]} \cdot \cdots \cdot \frac{E[f^2_1(n)]}{E[f^2_0(n)]}$$
$$= \prod_{m=1}^{M} G^2_m \qquad (2.16)$$

where

$$G^2_m = \frac{E[f^2_m(n)]}{E[f^2_{m-1}(n)]} .$$

Substituting equation 2.13a for $f_m(n)$ and using equation 2.15 for k_m. G^2_m can be written in terms of the reflection coefficients for stage m.

$$G^2_m = \frac{E[(f_{m-1}+k_m bd_{m-1})^2]}{E(f^2_{m-1})}$$
$$= \frac{E(f^2_{m-1})+k_m \cdot 2E(f_{m-1}bd_{m-1})+k^2_m E(bd^2_{m-1})}{E(f^2_{m-1})}$$
$$= \frac{E(f^2_{m-1})-k^2_m[E(f^2_{m-1}+E(bd^2_{m-1})]+k^2_m E(bd^2_{m-1})}{E(f^2_{m-1})}$$

$$G^2_m = 1-k^2_m \quad . \qquad (2.17)$$

Equation 2.17 can now be substituted into equation 2.16 to yield the expression for mean-squared gain of the analysis filter.

$$G^2_{filter} = \prod_{m=1}^{M} (1-k^2_m) \quad .$$

CHAPTER 3

Implementation of Algorithm

3.1. Introduction

Until recently, implementations of the LPC algorithm have used microcomputers employing special purpose arithmetic processors. In 1979, Texas Instruments introduced a speech synthesizer using the Lattice LPC technique designed around a set of integrated circuits. The synthesizer used a chip set comprised of a read-only memory to store the speech, a calculator chip as a controller, and a pipelined digital multiplier for computing the algorithm. Due to the extensive use of MOS-LSI technology, this was the first complete speech synthesizer that could be manufactured inexpensively and it even sold as a toy. This was quickly followed by the introduction of a number of speech synthesis integrated circuits using an all-digital approach. A number of efforts are now nearing completion which use the analog technique of switched capacitor filtering [1]: one using two-pole sections to model the formants, and the other using the Lattice LPC technique.

Due to the increased complexity, speech analysis devices have been slower in coming. Presently, most speech analyzers for word recognition pre-processing use channel bank filters, some of which have been fabricated as single integrated circuits using either CCD technology (Texas Instruments) or a switched capacitor approach (Reticon). The most recent implementations of the Lattice LPC algorithm are built around either a bit-slice microprocessor (Lincoln Labs), a number of specialized digital signal processing (DSP) integrated circuits (Motorola), or an array processor (Chi Systems). To date, there have been no attempts at using

switched capacitor technology to implement a Lattice LPC analysis filter aside from the research presented in this paper. Part of the reason for this is because at least a portion of the analysis algorithm is best suited to a digital approach, and traditionally integrated circuits have been either solely analog or digital with some recent analog-to-digital converters being the rare exception. In particular, the multiplications and divisions involved in computing the reflection coefficients are difficult to accurately implement in analog circuitry.

The development of switched capacitor techniques for general signal processing was formalized in 1980. At that time, McCharles [32] developed analog building blocks for signal processing using switched capacitor techniques which he tested on an A/D converter. Continuing in this vein, the research presented here focuses on implementing the Lattice LPC algorithm as a speech analysis filter using an optimal combination of analog and digital circuitry for the greatest efficiency of implementation as an integrated circuit [14]. Signal processing is done by a combination of digital circuitry and analog switched capacitor techniques according to the advantages of each method. This chapter will describe the tradeoffs between analog and digital circuitry, the necessary building blocks to implement an adaptive filter for speech analysis, and compare the Hybrid Analog/Digital approach with an all-digital approach using special purpose microprogrammed circuitry. In chapter 4 the requirements unique to integrated circuit design are discussed.

3.2. Hybrid Analog/Digital Approach

3.2.1. Comparison of Analog and Digital Techniques

The Lattice LPC algorithm can be divided into a number of functions, all of which are common to adaptive filtering in general. Basic to the analysis filter sec-

tion of figure 2.4a are three components: a summer, to add or subtract a number of signals; a multiplier, in this case both the multiplier (k_m) and the multiplicand are functions of time; and a storage element, to provide a unit sample of delay. The correlator which computes the reflection coefficients requires functions to perform division and expectations aside from most of the other building blocks already mentioned.

In any technology, multiplication and division with both inputs as variables are the most difficult to accurately compute of all the previously mentioned functions. Purely analog multipliers and dividers with high accuracy have substantial settling times and at best, less than one percent accuracy without costly laser or external trimming, especially when using MOS integrated circuit technology [42]. Fully parallel digital multipliers require large amounts of chip area [46] and serial digital multipliers, although much smaller, are nearly an order of magnitude slower than parallel. But for high accuracy applications, digital is the only choice.

An advantage of analog over digital approaches is the simplicity in routing signals. Analog signals can be run over a single wire. Digital, however, requires either large bus structures of between 16 to 32 lines or else complicated clocking schemes and serial-to-parallel and parallel-to-serial converters to route data. This often leads to wasted area in an integrated circuit layout. Also, there are many times more active elements in digital circuitry than in analog, requiring a large increase in power dissipation. But even with these advantages, both analog and digital approaches seem to require roughly the same amount of chip area to implement [37] mainly due to the large capacitors required for good dynamic range in switched capacitor circuitry.

One clear-cut advantage that digital circuitry has is in the use of read-only memories (ROMs) and programmable logic arrays (PLAs) to implement compli-

cated functions with look-up tables.

Digital circuitry can accommodate practically any function at any desired accuracy given enough chip area and clock cycles, but therein lies its present limitation. For a given function, a digital approach may require too much chip area or require too many clock cycles in a particular technology to be competitive with analog. Switched capacitor analog circuitry is inherently simpler than digital since only one wire is required to transmit a signal and because the complete signal is represented as charge on a capacitor. So with the aid of an operational amplifier (op-amp), addition, subtraction, multiplication by a constant, and storage is made very simple. Often not more than two clock pulses are required for any of these operations, though the clock rate is often slower for analog circuitry.

Yet hybrid analog/digital circuitry has some unique advantages over either analog or digital circuitry separately. A/D and D/A converters can be adapted to perform a number of nonlinear functions with reasonably high accuracy. In particular, a D/A converter implemented in switched capacitor circuitry can perform a number of high speed, high precision multiplies with minimal chip area between an analog signal and a digital signal to produce an analog output. Some logarithmic and exponential functions can be implemented by slightly modifying these building blocks. The major restriction to using A/D and D/A converters is the necessity for converting the information between analog and digital forms.

The lattice analysis algorithm itself suggests a natural division between analog and digital implementations allowing the advantageous use of hybrid analog/digital techniques. The division function in the correlator restricts at least part of the correlator implementation to a digital form. This has the advantage of producing reflection coefficients in a digital form which simplifies their

quantization and makes the interface to other vocoders or other word recognition systems simple.

If digital reflection coefficients are available, the lattice filter structure of figure 2.4a becomes ideally suited to an analog implementation. A multiplying digital to analog converter (MDAC) can be used to multiply the reflection coefficients into the analog lattice filter structure. The rest of the required elements, the summers and the delay element, are easily implemented as analog building blocks. Detailed descriptions of the implementation of the digital correlator and the analog filter structure are presented in the following sections.

3.2.2. The Correlator

The algorithm that was derived in chapter 2 for computing the reflection coefficients is:

$$k_m = -\frac{2E[f_{m-1}bd_{m-1}]}{E[f_{m-1}^2]+E[bd_{m-1}^2]} . \tag{3.1}$$

As it stands, computation of k_m requires one multiplication, two squaring operations, three expectations, a division, and scaling by a constant. Equation 3.1 can be rewritten as follows to reduce the number of required multiplications.

$$k_m = \frac{E[(f_{m-1}-bd_{m-1})^2]-E[(f_{m-1}+bd_{m-1})^2]}{E[(f_{m-1}-bd_{m-1})^2]+E[(f_{m-1}+bd_{m-1})^2]} \tag{3.2}$$

Equation 3.2 has no scale factor and requires one less multiplication and one less expectation at the expense of three more summations. If we are given the sum and difference of f_{m-1} and bd_{m-1}, then just two squaring operations, two expectations, a sum and difference, and one division are necessary. Even this can be simplified if instead of computing k_m, we compute the area ratio:

$$\frac{Area_{m-1}}{Area_m} = \frac{1+k_m}{1-k_m} = \frac{E[(f_{m-1}-bd_{m-1})^2]}{E[(f_{m-1}+bd_{m-1})^2]} . \tag{3.3}$$

Since the division must be done digitally, a small ROM look-up table can be used

to convert the area ratios into reflection coefficients.

Quantizing the area ratio is desirable in order to reduce the bit-rate for transmission of the filter parameters. A non-linear MDAC can be used in the lattice filter section to make use of the area ratios directly.

The forward and backward residuals from the analog filter section are themselves analog, so computing their sum and difference is easily done using analog circuitry. Computing the area ratio from equation 3.3 then just reduces to two squaring operations, two expectations, and a division. The hardware to perform the squaring and expectation can be multiplexed between the numerator and denominator. Figure 3.1 shows a block diagram of the correlator.

A major drawback to building an analog squaring circuit is that the output must span twice the dynamic range (in decibels) as the input. The maximum dynamic range in analog circuitry is limited by noise and supply voltage. These

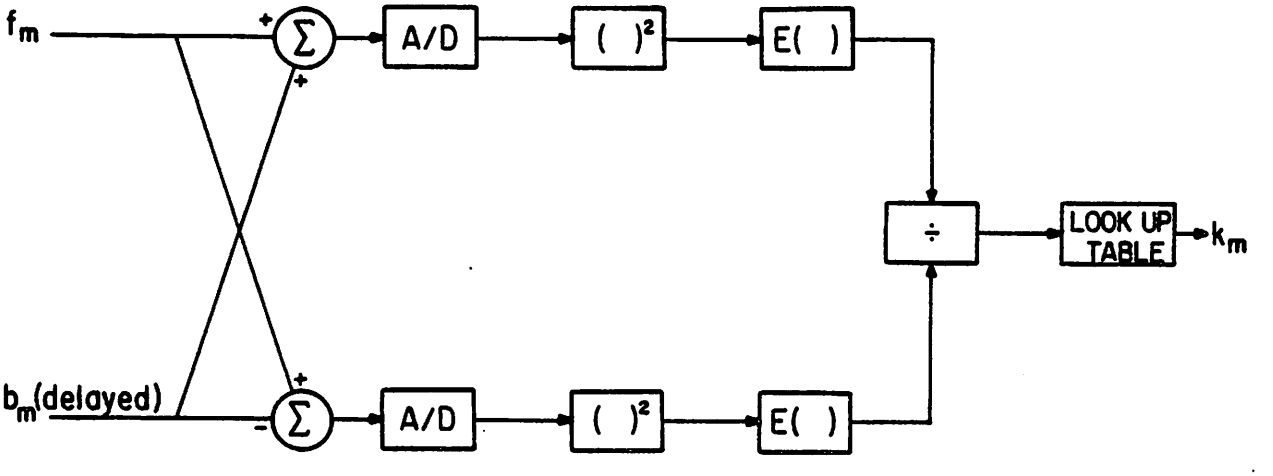


Figure 3.1. Block diagram of optimized correlation algorithm of equation 3.3

constraints make it extremely difficult to design an analog squaring circuit with sufficient dynamic range on the output without reducing the input dynamic range. The squaring function should therefore be implemented by digital logic. An analog to digital conversion will be necessary between the analog filter section and the digital squaring circuit. However, by using a logarithmic A/D converter the squaring circuit can be greatly simplified. The squaring function would then require a multiplication by two (just a simple shift left of one bit) and an antilog conversion requiring simple shifting logic and, possibly, a ROM.

A true logarithmic A/D converter is difficult to build but μ-Law companding (compressing-expanding) A/D have already been fabricated in CMOS technology with switched capacitor circuitry [27]. The μ-255 Law, as specified by Bell Telephone, is defined by the equation

$$F(x) = Sign(x)\frac{ln(1+\mu|x|)}{ln(1+\mu)}$$

where μ equals 255. Actual μ-255 Law A/D converters use a piecewise linear approximation. A typical eight bit μ-255 A/D converter has one sign bit, three chord bits, and four step bits, in that order - from most significant bit (MSB) to least significant bit (LSB). Excluding the sign bit, the three chord bits provide eight quantization ranges, each range differing by a factor of two from the next. Within each chord are sixteen evenly spaced steps represented by the four step bits. The resulting digital word can be thought of as being in floating point notation to the base two. The chord bits would represent the exponent and the step bits plus one would represent the mantissa. A more accurate conversion from μ-255 Law to floating point will be found in the appendix.

Computer simulations helped to determine the optimum number of bits to use for the μ-law quantization. The results of these simulations (figure 3.2) show that an eight bit representation with 3 chord bits and 4 step bits, as described

above, were optimum for this application.

With a companding μ-255 Law A/D converter, the squaring function is accomplished using a small lookup table to square the mantissa and shifting logic to shift the output of this ROM by twice the chord value. The result is a fixed point digital number.

The next step in computing the area ratio is to calculate the expectation of these squares as in equation 3.3. The expectation function must process a digital input and form a digital output for use in the division circuitry. The expectation can be adequately approximated by a one pole digital low pass filter with a pole located near 20 Hertz [25], although in a recent paper Kang has suggested that a two pole low pass filter with poles near 85 Hertz produces slightly clearer speech [26]. The z-transform equation for a one pole low pass filter at 20.2 Hertz with an

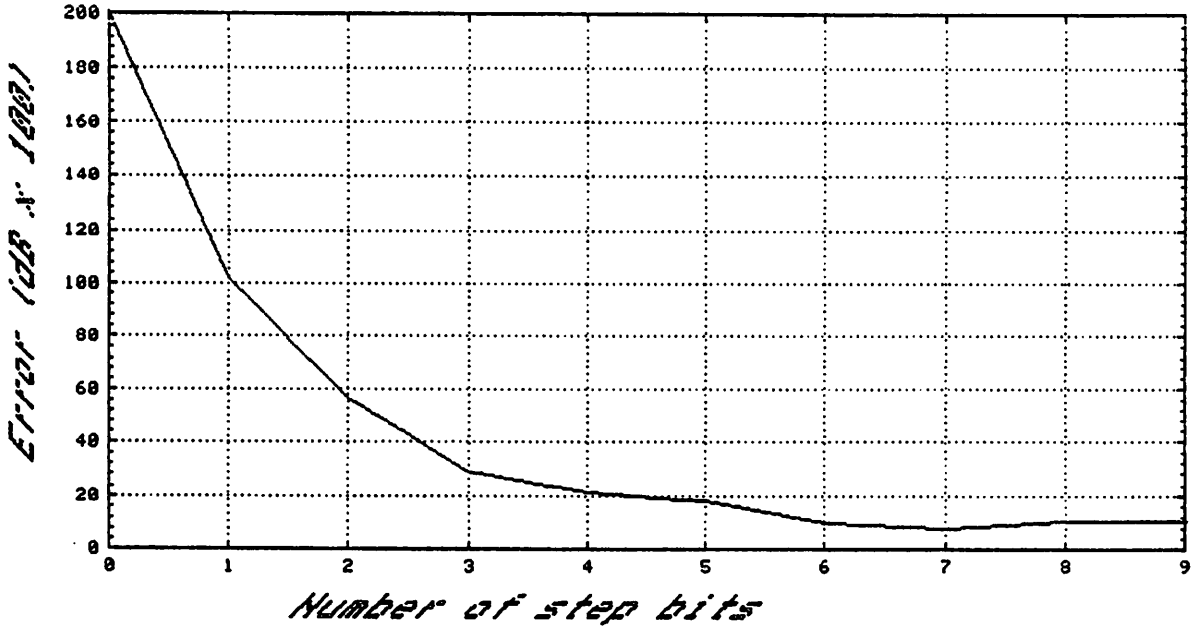


Figure 3.2. Computer simulations to determine optimal μ-law quantization for the logarithmic squaring input to correlator

8 KHz. sampling rate is

$$V_{out}(z) = E[V_{in}(z)] = z^{-1}\frac{63}{64}V_{out}(z) + \frac{1}{64}V_{in}(z) \quad . \tag{3.4}$$

This equation is normalized for unity gain. Implementation of equation 3.4 can be simplified by rewriting the multiplication by 63/64 as 1−1/64.

$$V_{out}(z) = z^{-1}[V_{out}(z) - V_{out}(z)/64] + V_{in}(z)/64 \tag{3.5}$$

Figure 3.3 shows a block diagram of a circuit to implement this one-pole expectation function. The division by 64 is easily accomplished by shifting the bus wiring six bits resulting in a shift right operation. The two-pole expectation filter could also be implemented with only a slight increase in circuitry.

A two to one multiplexing circuit selects between $V_{in}(z)/64$ and the ones complement of $V_{out}(z)/64$. By placing the hardwired divide operations before the multiplexer, a smaller multiplexer can be used because there are six fewer bits

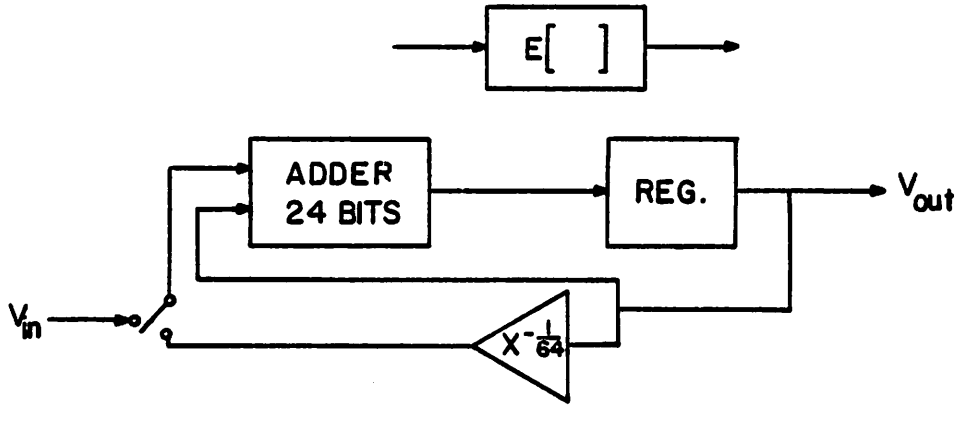


Figure 3.3. One-pole low pass filter for approximating expectation function

required to represent the input signals. The additive inverse of $V_{out}(z)/64$ is produced by using inverters to form the ones complement and applying a logic one to the carry input of the adder to turn the ones complement number into a twos complement number so that subtraction can be performed.

The adder together with an edge triggered master/slave latch form an accumulator which can evaluate equation 3.5 in two clock cycles. First the twos complement of $V_{out}(z)/64$ is added to $V_{out}(z)$ to produce $(63/63)V_{out}(z)$. Then the result is stored in the latch and added to $V_{in}(z)/64$ to complete the cycle.

Dividing the two expectations is the last step in calculating the area ratio. The division can be implemented in a logarithmic fashion similar to what was used for the squaring circuit. By first converting the dividend and divisor to floating point, a small lookup table can be used to divide the mantissas while the exponents can be subtracted in a small adder. The resulting floating point quotient is the area ratio. Another lookup ROM can then be used to convert the area ratio into quantized reflection (or PARCOR) coefficients.

Computer simulations show that while 24 bits are necessary to compute the expectation, the floating point inputs to the divider require only 8 bits: 4 bits for the exponent and 4 bits for the mantissa. The area ratio can produce five bit vocal tract parameters and still reproduce good fidelity speech [44], yet the implementation which is presented here uses 8 bit linearly quantized reflection coefficients for simplicity.

A circuit to convert a fixed point number into a floating point number must find the location of the first non-zero MSB for the exponent and select the next several bits for the mantissa. Such circuits are not difficult to design as combinational logic, yet they are fast and make efficient use of chip area. The use of floating point multipliers and dividers takes advantage of the robustness of the Lattice

LPC algorithm in its insensitivity to quantization of the reflection coefficients thereby making implementation more efficient.

3.2.3. Implementation of Correlator

The correlator was implemented using standard digital parts compatible with monolithic MOS integration. It was designed for time-multiplexed use with a ten stage lattice filter using delayed reflection coefficients. For an 8 kHz sample rate, there is 12.5 μsec of time available for the computation of k_m. From figure 3.1, we see that the computation of k_m requires: an analog sum and difference, two 7 bit (minus sign bit) companded A/D conversions, a square look-up, two float to fixed conversions, two 24 bit digital additions for computing the expectations, a fixed to float conversion, a division look-up, and an area ratio to reflection coefficient look-up.

The two A/D conversions are done simultaneously with two converters, so together they require nine clock cycles. One cycle is required for each of the seven bits plus one for sign bit and one for reset in the successive approximation conversion. Since the expectations are computed using the one pole low pass filter algorithm of equation 3.5, two addition operations are required. Two expectations therefore require a minimum of four clock cycles. However, since the 2 MHz clock rate is too fast for the MOS logic used, these four clock cycles are computed at a 1 MHz rate which is equivalent to eight 2 MHz clock cycles. The fixed to float, float to fixed, and ROM table look-ups can all be done using combinational logic, and therefore do not use additional clock cycles. The total amount of time used by the correlator is 17 clock cycles at 2 MHz or 8.5 μsec. The excess 4 μsec is can be used for time-multiplexing the A/D conversions so that only one μ-law converter would be necessary in a monolithic implementation. In either case, the A/D converter uses 0.5 μsec per bit. This requires implementation of an MOS

comparator which settles within this amount of time. A 0.5 μsec MOS comparator is well within the limitations of most MOS technologies.

3.2.4. The Analog Lattice Filter Section

As previously mentioned, all of the blocks for the Lattice Filter Section of figure 2.4a can be easily implemented using switched capacitor techniques [32]. The Lattice Filter structure requires the following fundamental blocks: an adder, a subtractor, an MDAC, and a delay element. Switched capacitor circuitry to implement each of these functions will be discussed in this section.

In analog switched capacitor technology, a signal is stored as charge on a capacitor in the sampled time domain. A signal is periodically sampled with the closing of an MOS transistor switch which allows charge to flow into a capacitor. The switch must be closed long enough to allow the capacitor to fully charge (or discharge) to the required signal level. This must occur frequently enough so that the full frequency bandwidth of the signal can be represented. By the sampling theorem, the maximum signal bandwidth that can be represented is one half the sampling frequency. This is known as the Nyquist Criterion. All higher frequencies must be eliminated with a low pass filter (anti-alias filter) to avoid distortion due to the translation of higher frequencies down to lower frequencies, a process known as aliasing. The signal bandwidth therefore determines the maximum sampling time, and the capacitance and switch resistance must be kept low enough to allow sufficient signal transfer within this sampling time.

Transferring all of the signal charge from one capacitor to another is best done with the aid of an op-amp. Figure 3.4 shows a simple circuit to sample a signal with an input capacitor (C_i) and then transfer the charge to a feedback capacitor (C_f). This circuit requires two non-overlapping clock phases to complete the operation. During the first phase (φ_1), the feedback capacitor is discharged

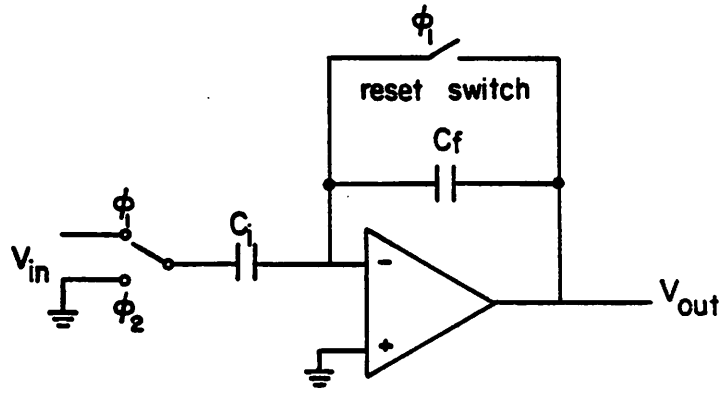


Figure 3.4. A simple switched capacitor circuit illustrating charge transfer techniques

while the input capacitor is connected to the input signal. Since negative feedback forces the inverting input of the op-amp to remain at ground potential, C_i acquires a charge of $Q_0 = V_{in} C_i$. After φ_1 and before φ_2, when all the switches are open, C_i remains charged with the voltage V_{in} across it while C_f remains discharged with zero volts across it. When C_i is switched to ground during φ_2, the feedback around the op-amp causes all the charge from C_i to be transferred to C_f since both terminals of C_i must settle to zero volts, discharging it. The output voltage (V_{out}) changes to allow C_f to acquire the charge Q_0 so that $Q_0 = V_{out} C_f$. Since ideally no charge is lost, all the charge from C_i has been transferred to C_f and $V_{in} C_i = V_{out} C_f$ so that

$$\frac{V_{out}}{V_{in}} = \frac{C_i}{C_f} .$$

This circuit therefore performs a non-inverting multiplication by C_i / C_f. It can

also act as a storage element since even with all the switches open, the charge remains stored (assuming no leakage) and the output voltage remains constant.

It is important to note that only the relative voltage change on C_i is important. If instead of grounding C_i on φ_2, C_i is switched to V_{in2}, the transfer function for this circuit would become:

$$V_{out}=(V_{in1}-V_{in2})C_i / C_f$$

allowing two signals to be subtracted. (V_{in} has been replaced by V_{in1} for clarity.)

If a second input capacitor is added to the circuit with its own set of input switches connected to φ_1 and φ_2, as shown in figure 3.5, the transfer function would be:

$$V_{out} = (V_1-V_2)\frac{C_{i1}}{C_f}+(V_3-V_4)\frac{C_{i2}}{C_f} \qquad (3.6)$$

An adder can be formed by setting $V_2 = V_4 = 0$ and $C_{i1} = C_{i2} = C_f = 0$ to yield

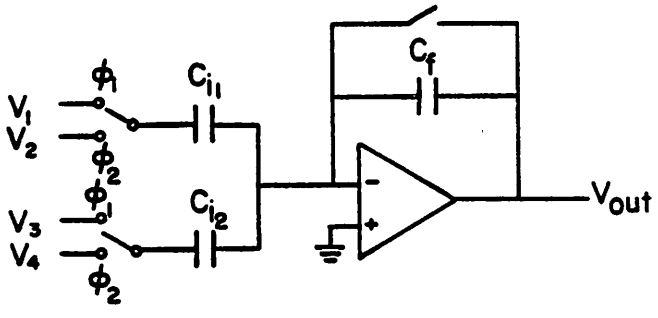


Figure 3.5. Basic switched capacitor circuit with two input capacitors

$V_1+V_3 = V_{out}$. This basic circuit element illustrates the versatility of switched capacitor circuitry.

As we shall see, both analog delay lines and MDACs have similar circuit topologies since multiple capacitors are necessary for multiple storage elements (or delays) and multiple gain values.

The simple switched capacitor circuit of figure 3.4 suggests two approaches to implementing a delay line or MDAC. Either C_f or C_i can be replicated with appropriate switching elements to provide the desired function. However, switching in and out feedback capacitors can introduce much undesirable clock feedthrough and crosstalk. These effects will be fully discussed in chapter 4 when parasitics and integrated circuit implementation are described.

The most prudent approach to constructing an MDAC or delay element would be to switch in and out various input capacitors. Figure 3.6 shows several input capacitors ($C_1, C_2, \ldots, C_N$) each connected by a series switch ($Sw_1, Sw_2, \ldots, Sw_N$) to a common node where V_{in} or ground can be applied using φ_1 or φ_2. A single feedback capacitor C_f is connected directly across the op-amp to provide negative feedback. A reset switch is connected across C_f and is controlled by $\varphi 1$.

If the switches, Sw_1 through Sw_N, are restricted to changing state just before φ_1, then the circuit behaves just as before except that the gain is determined by which capacitors are switched in during that $\varphi_1 \rightarrow \varphi_2$ clock cycle. An N-bit MDAC function is implemented by giving the capacitors C_1 through C_2 a binary weighting of powers of two (ie. $C_1 = C_0, C_2 = 2C_0, \ldots, C_N = 2^{N-1}C_0$). A binary (digital) number controlling the switches Sw_1 through Sw_N, from LSB to MSB respectively, would control the gain. An analog multiplicand and a digital multiplier would then form an analog product. Although the analog input and analog output can be either positive or negative, for this design the digital input is restricted to

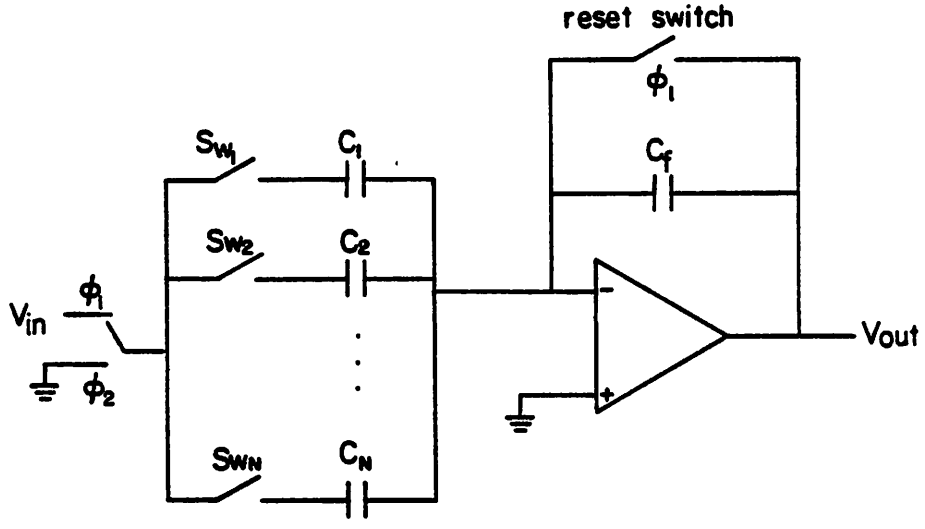


Figure 3.6. Switched capacitor circuit with multiple input capacitors for implementing an MDAC or delay line.

only positive values resulting in a two quadrant multiplier.

A four quadrant MDAC can be built by adding another input capacitor (C_x) equal to $2^{N-1}C_0$, the same size as C_N, and setting $C_f = 2^{N-1}C_0$, also the same size as C_N. C_x would always switch between ground on φ_1 and the input signal (V_{in}) on φ_2, having the opposite phasing of the other capacitors. The gain of V_{in} through C_x would be -1 while the gain of V_{in} through C_N is $+1$. Thus, any charge introduced by C_N when it is switched in would be canceled by charge from C_x. With Sw_N closed (C_N switched in), the gain of the MDAC is always a positive fraction -from zero to nearly unity- of the input signal. With Sw_N open, C_x will not be canceled by C_N but will contribute a gain of -1 to the positive gain from the other capacitors C_1 through C_{N-1}. The overall gain of the MDAC will then be a negative fraction from -1 to nearly zero. This is known as an "offset binary" four quadrant MDAC.

The circuit of figure 3.6 can be used as a delay line if all the capacitors are of equal capacitance and the switches Sw_1 through Sw_N are selected in succession, one at a time, and change state just before the beginning of φ_2. The capacitors are thus selected in a cyclic fashion. On φ_1 the most recent value of V_{in} is stored on capacitor C_n and C_f is reset. Then on φ_2, C_{n+1} is selected in place of C_n and the delayed signal is transferred to C_f and output as V_{out}. At a time N cycles later, the signal that was stored on C_n becomes the output, and the cycle is repeated.

3.2.5. The Complete Lattice LPC Analysis Filter

All of the building blocks necessary to complete a section of the Lattice LPC analysis filter have been described. Generally, ten or more such sections must be cascaded in series to construct a complete speech analysis filter. Instead of building many identical sections, a single section can be time multiplexed to efficiently implement a complete multi-stage filter. The main disadvantage to multiplexing is that a single time multiplexed section must operate much faster than for a non-multiplexed implementation. For a ten stage filter, a single section must perform ten times as many operations in a single sample period. Fortunately, with a little care in the integrated circuit design, this is not a serious obstacle for most analog integrated circuit technologies.

A simplified schematic for the time multiplexed analog filter section of figure 2.4a is shown in figure 3.7. This circuit is meant to illustrate the use of the fundamental building blocks discussed in this chapter. A more practical clocking scheme for integrated circuit implementation will be presented in chapter 4 after various parasitic considerations have been explored. This circuit requires seven op-amps labeled Op_1 through Op_7. Four op-amps (Op_1 through Op_4) compute the basic lattice algorithm while two op-amps (Op_5 and Op_6) provide the sum and

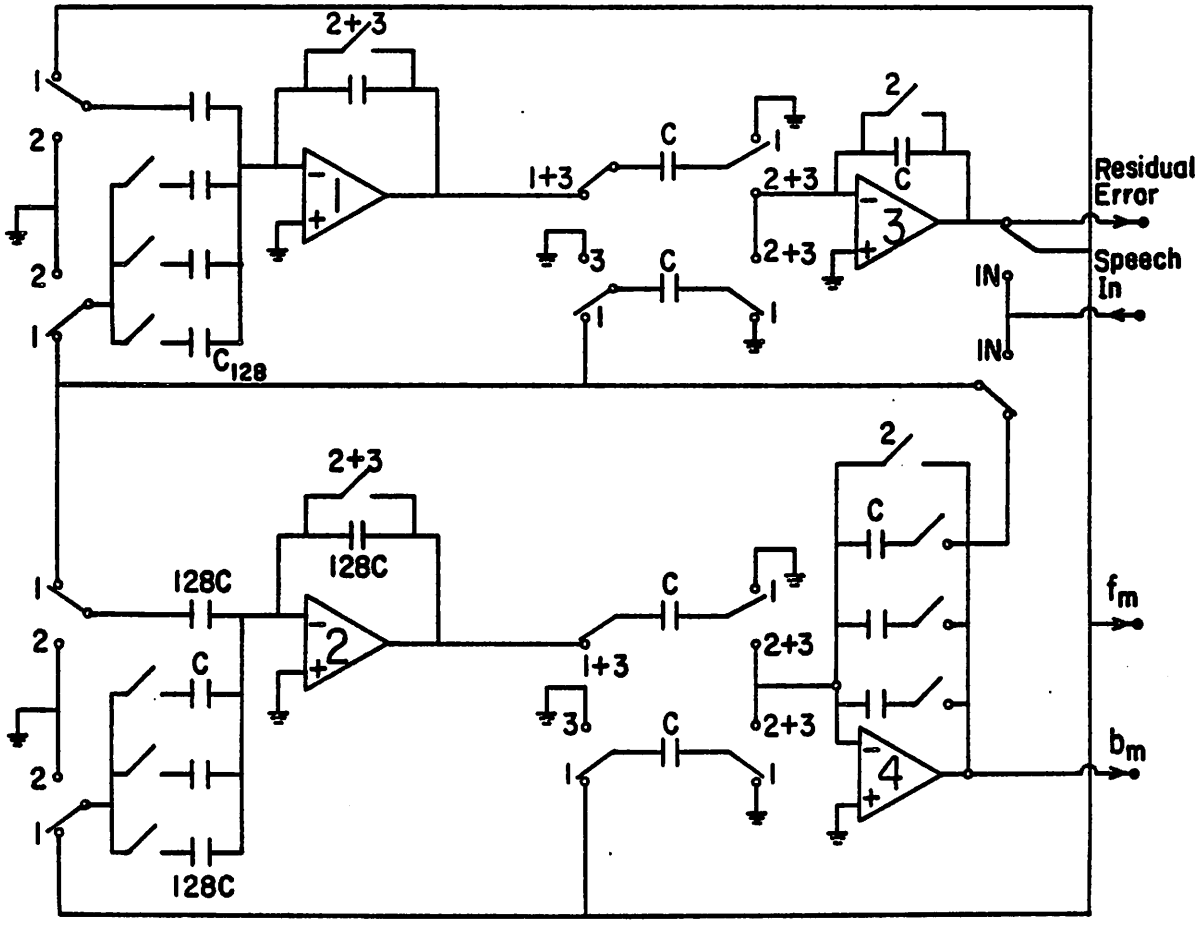


Figure 3.7. A simplified schematic for a time multiplexed lattice analysis filter

difference of the residuals for use in the correlator. A seventh op-amp (Op_7) is included to sample and hold the final forward residual error as the output of the all-zero analysis filter. There are five non-overlapping clock signals in this circuit labeled φ_1 through φ_5. A switch is normally open except during the clock phase that drives it.

To compute the Lattice algorithm, Op_1 and Op_2 are each arranged as both a four quadrant MDAC and a summer, providing most of the computation. On φ_1 their feedback capacitors are reset, the proper input capacitors are switched in according to the reflection coefficient for that stage of the computation, setting the gain, and the signals which are to be added are switched to charge input capacitors. On φ_2 the proper input capacitors are either switched to ground or to the signals to be subtracted to complete the calculation.

Op-amps Op_3 and Op_4 both provide sample and hold operations to enable the overall filter to be multiplexed. The signals computed by a previous "stage" on φ_2 are stored on the input capacitors on clock phase φ_3 and transferred to the output on φ_4 to be used as inputs to the next "stage" of computation. Op-amp Op_4 has additional capacitors to provide the delay of the backward residuals required by the algorithm. For a ten stage filter, there would be eleven input storage capacitors comprising an eleven bit delay line. These storage capacitors could be successively selected by an eleven stage ring counter clocked just prior to φ_4. During a single sample period the ten stages of backward residuals are stored and progress through the delay line. The eleventh delay element produces an overall delay of one sample period for the backward residuals.

One way to understand this is to consider that one capacitor is necessary to store a backward residual throughout an entire sample period for each of the ten stages, in order to implement a delay. While the other ten capacitors are storing

signals to be delayed, the eleventh capacitor in the "delay line" holds the charge to be input to the next stage of the computation. The eleventh capacitor is performing the sample and hold operation of Op_3. Acting as a delay line, all of the capacitors are eventually used to store charge for all of the stages. As a simplified example, consider a one stage filter where the forward and backward residuals are to be sampled and held for further processing. Here the delay circuit would require two capacitors: one to hold the latest computed value for the backward residual, and the other for holding the delayed value of the backward residual to be outputted.

Op-amp Op_5 computes the difference between the forward and backward residuals for use in the correlator. A single input capacitor switches between f_m on φ_4 and bd_m on φ_5 to yield $f_m - bd_m$. Op-amp Op_6 computes the sum of f_m and bd_m with two input capacitors, each switching between one of the signals on φ_4 and ground on φ_5. Op-amp Op_7 is a sample and hold similar to Op_3 except its switches are clocked only once each sample period to sample the forward residual of the last stage to provide an output for the analysis filter.

At the beginning of each sample period, the input speech (V_{in}) is stored in both op-amps Op_3 and Op_4 since the boundary condition at stage zero requires: $V_{in}(n) = f_0(n) = b_0(n)$.

3.3. All-digital Approach

For comparison with the analog lattice filter, a lattice analysis filter was constructed using only digital logic. The digital version uses an approach that is compatible with monolithic MOS integration. For versatility, a microprogrammed architecture was adopted.

In this microprogrammed architecture, a versatile signal processing element, called the processing unit, is controlled by a ROM, called the controller.

Microprogramming allows for a more trouble-free and condensed implementation since it eliminates the need for large amounts of random logic. As an additional advantage, a minor modification in the microcode can change the speech analysis filter into a speech synthesis filter. Thus, the same digital filter circuitry has the capacity to perform both the analysis and synthesis functions of a vocoder. Although optimized for the lattice algorithm, the signal processing hardware proved flexible enough to implement a variety of signal processing algorithms by just changing the ROM programming. A digital monolithic implementation of a complete lattice vocoder IC was designed based on this microprogrammed signal processing unit [37]. The same signal processing hardware was used the analysis, synthesis, pre-emphasis, and de-emphasis filtering. The signal processing unit was replicated elsewhere on the chip to implement the correlator.

The resulting hardware provides a very reliable and parasitic free real-time implementation of the analysis filter structure. The digital filter uses a 12 bit linear A/D converter on the input and a 12 bit linear D/A converter on the output. These converters are only required to operate at an 8 kHz sample rate, although internally the digital filter is multiplexed at ten times that rate to implement a ten stage lattice filter. The 125 μsec, 12 bit A/D conversion is easily implemented with standard off-the-shelf parts. The digital filter was designed to be compatible with the analog version so that it could be used with the same correlator.

Although the digital hardware does not suffer from parasitic offsets and charge injection that analog approaches face, potential problems exist due to truncation, quantization, and overflow errors. The amount of digital circuitry required increases linearly with the number of bits used. An optimal number of bits for the lattice filter computation was found to be 16. This allows two bits of head-room to minimize overflow and two bits to minimize underflow and trunca-

tion error.

In the case of overflow, analog circuitry has the advantage that it saturates to an upper limit upon overflow. Unless saturating adders are used, digital adders reverse sign on overflow which causes severe distortion. Since the 16 bit hardware has 12 dB of head-room, it did not exhibit much overflow sensitivity. The 12 dB of underflow proved satisfactory in providing sufficient accuracy when the fractional reflection coefficients were multiplied.

3.3.1. Signal Processing Unit

A block diagram for the signal processing unit is shown in figure 3.8. The signal processing unit is built around a 16 bit accumulator which is accessed via a single bi-directional bus. The accumulator consists of an adder in series with an edge triggered output latch with the latch outputs fed back to the adder inputs. The accumulator input is routed to the other adder input through a shift register and some combinational logic. The shift register and combinational logic form a simple arithmetic logic unit (ALU) and edge triggered input latch. The ALU provides a one bit shift right with sign extend operation, twos-complement logic, and add-inhibit logic. This is all the logic necessary to perform serial-parallel multiplies. Connected to the bus are an eleven word memory, the input A/D converter, and the accumulator input and output. The memory provides the sample and hold and delay line storage for time-multiplexed operation.

Combinational logic in series with a latch is used to perform the fixed to floating point conversion for output to the correlator. To reduce the overall number of computation steps, this output conversion logic is connected directly to the adder output rather than to the bus. A schematic block digram of this fixed to float converter is presented in figure 3.9.

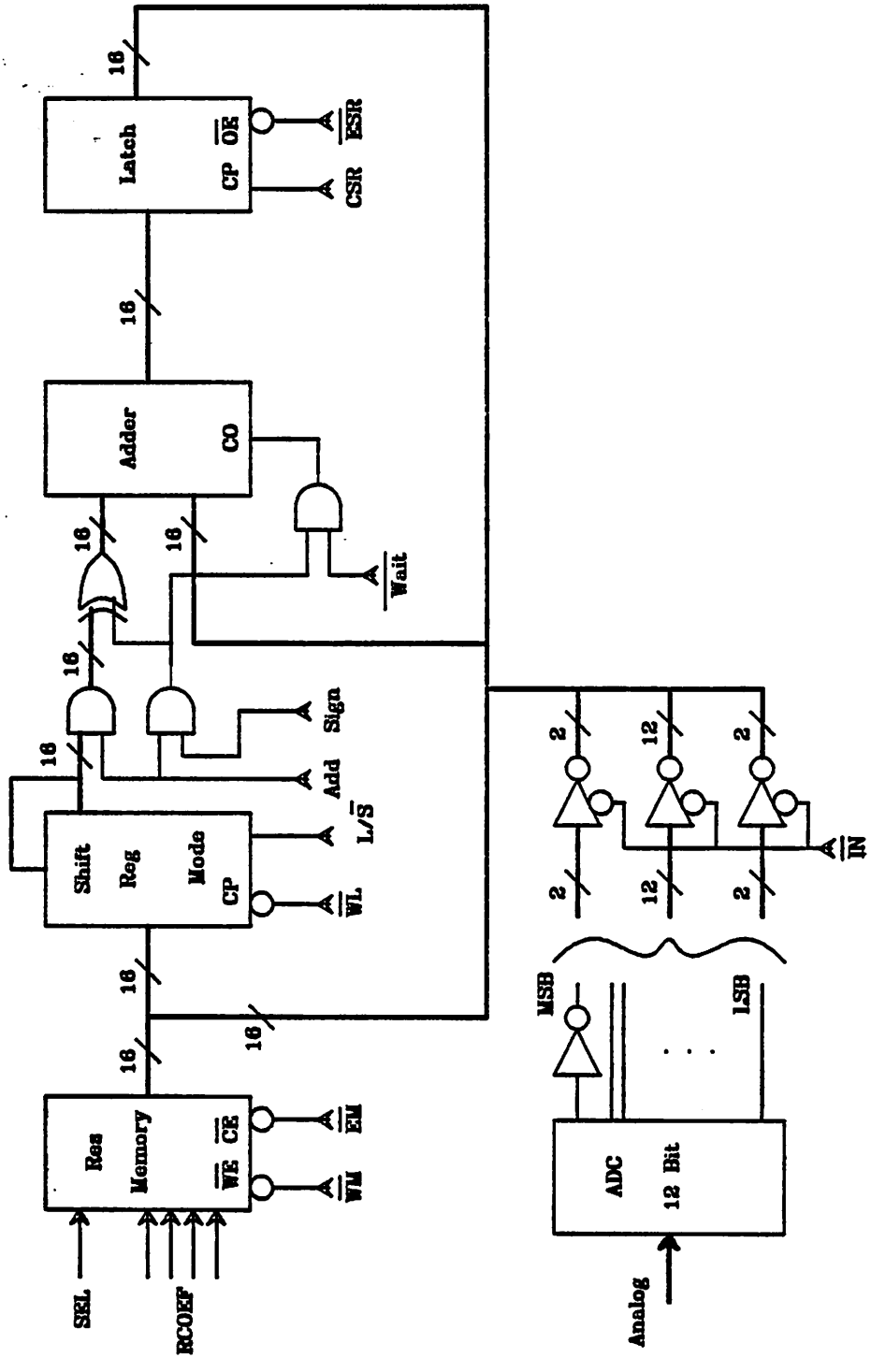


Figure 3.8. Block diagram for the signal processing unit

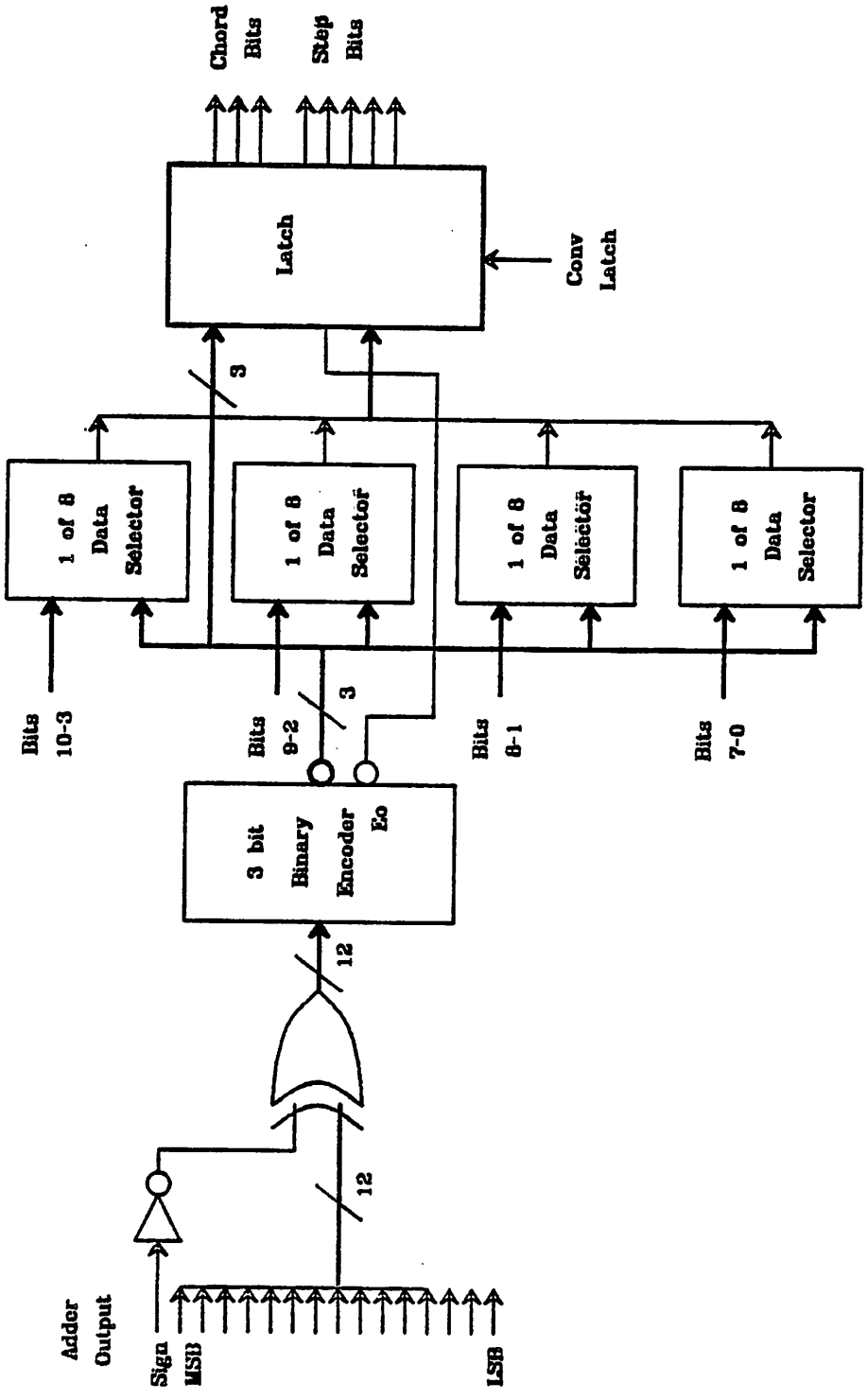


Figure 3.9. Block diagram of a fixed-point to floating-point converter for interfacing from digital lattice filter to correlator

3.3.2. Controller

Figure 3.10 is the block diagram of the controller. The heart of the controller is a ROM which must provide eleven control lines and enough clock states to implement the computations. For the lattice analysis algorithm, nineteen clock states are required. For each clock cycle there are two periods, a low state and a high state. For ease of design, the convention was adopted that input data and bus lines would be set up during the low period and all clocking or latching of output data would be done on the low to high transition. This divided the clock cycle evenly between input and output data settling times. During operation, a modulo 19 counter cycles through the ROM microcode. There is a halt instruction which can stop the instruction counter and signal the correlator to start. The counter can then be restarted upon completion of the correlation computation. So that varying propagation delays in the ROM outputs cannot lead to race conditions, all of the output clock control lines are synchronously clocked by the logical ANDing of the master instruction counter clock with these lines. Bus control outputs are usually brought directly out to the appropriate bus control line.

Tables 3.1a and 3.1b show the actual microprogramming code for the two controller PROMs used for controlling the eleven control lines when computing the lattice analysis algorithm (without the reflection coefficient computation). An X represents a "don't care" condition in the code. The storage register for computing the forward path is enabled for SEL=1. The backward delay registers are selected when SEL=0. The majority of the first seventeen instruction cycles control the two shift-and-add serial-parallel multiplies by the 8 bit reflection coefficients since one clock cycle is required for each shift-and-add operation. The remaining two instructions perform the sum and difference operations for the correlator.

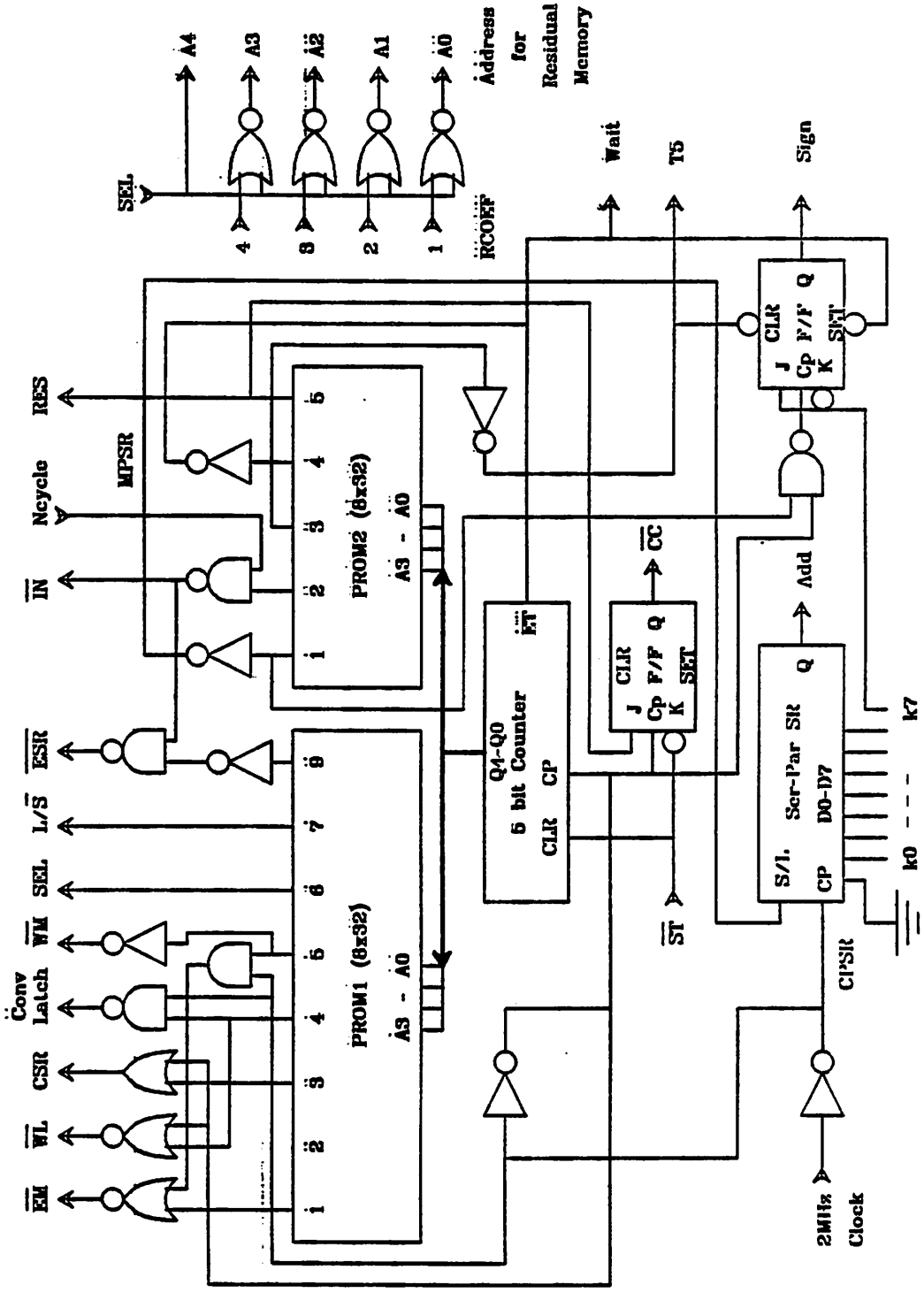


Figure 3.10. Block diagram of the controller for the digital lattice filter

PROM Data 1								
Cycle	Clocks				Buses			
	EM	*WL*	*$\overline{CSR}$*	*CONV. LATCH*	*WM*	*SEL*	*L/$\overline{S}$*	*$\overline{ESR}$*
(pin)	*(1)*	*(2)*	*(3)*	*(4)*	*(5)*	*(6)*	*(7)*	*(9)*
0	1	0	0	0	0	0	0	1
1	0	0	0	0	0	X	0	0
2	0	0	0	0	0	X	0	0
3	0	0	0	0	0	X	0	0
4	0	0	0	0	0	X	0	0
5	0	0	0	0	0	X	0	0
6	0	X	0	0	0	X	x	0
7	1	0	1	0	0	0	1	1
8	0	1	1	0	1	0	X	0
9	1	0	0	0	0	1	0	1
10	0	0	0	0	0	X	0	0
11	0	0	0	0	0	X	0	0
12	0	0	0	0	0	X	0	0
13	0	0	0	0	0	X	0	0
14	0	0	0	0	0	X	0	0
15	0	X	0	0	0	X	x	0
16	0	0	1	0	1	1	1	0
17	1	1	X	1	0	0	X	1
18	1	1	X	1	0	0	X	1

Table 3.1a. Microprogram Control Sequence - PROM 1

PROM Data 2					
Cycle	External Control				
	MPSR	*IN*	*CC*	*WAIT*	*RES*
(pin)	*(1)*	*(2)*	*(3)*	*(4)*	*(5)*
0	1	0	0	0	0
1	0	0	0	0	0
2	0	0	0	0	0
3	0	0	0	0	0
4	0	0	0	0	0
5	0	0	0	0	0
6	0	0	0	0	0
7	X	0	0	0	0
8	X	1	0	0	0
9	1	0	0	0	0
10	0	0	0	0	0
11	0	0	0	0	0
12	0	0	0	0	0
13	0	0	0	0	0
14	0	0	0	0	0
15	0	0	0	0	1
16	0	1	1	0	0
17	0	0	0	0	0
18	0	0	0	1	0

Table 3.1b. Microprogram Control Sequence - PROM 2

3.3.3. Interface to Correlator

Because the data is exclusively in digital form, the A/D converters of the correlator were bypassed. The sum and difference of the forward and backward residuals are computed in the processing unit and fed to a fixed-to-float conversion unit which uses combinational logic. The controller is then placed in a wait state while the correlator hardware is started. Although independent of each other, both the digital lattice filter and the correlator operate at 2 MHz clock

rates. As was mentioned, nineteen clock cycles are required to compute the lattice analysis algorithm and eight cycles are used for the correlator without the A/Ds. By overlapping two of the filter cycles with the eight correlator clock cycles, 25 clock cycles total can be used to compute one stage of the complete adaptive filtering algorithm. The total computation time for these 25 cycles is 12.5 μsec which allows for ten stages of adaptive filtering at an 8 kHz sample rate.

CHAPTER 4

Integrated Circuit Implementation

4.1. Introduction

As shown in figure 3.7, the analog switched capacitor circuit derived in chapter three for the Lattice analysis filter cannot be directly implemented in integrated circuit form. The components which are available on integrated circuits are far from ideal and their associated parasitics and imperfections must be taken into consideration in order for an analog integrated circuit to perform as specified. The particular integrated circuit process used and the way in which a circuit is laid out determines the kinds of parasitics which are important and the severity of the parasitics upon the functionality of the overall integrated circuit. Proper circuit design and layout can reduce the effects of the serious parasitics to an acceptable level. In this chapter, particular attention is paid to the implementation of the op-amp, being the basic building block in the design of the filter, and to the clocking scheme, which can help reduce the effects of parasitics. The relationship between circuit layout and design on the performance of the op-amp will be closely examined. The discussion of the dominant integrated circuit parasitics and their effect on circuit performance will be made with respect to an analog metal-gate CMOS process that was specifically designed for fabricating the analog portion of the Lattice LPC speech analysis filter described in chapter three.

The most important implementation concern for the digital portion of the analysis filter, the correlator, is the number of bits to be used in different portions of the circuitry. For efficient implementation, it is prudent to use the

minimum number of bits possible without introducing undue distortion or loss in dynamic range. In chapter 3, the results of computer simulations were presented which address this concern.

4.2. Component Parasitics

The basic components used in a switched capacitor filter are: the switches, the capacitors, the interconnects, and the op-amps. Each of these components can have certain undesirable characteristics which make integrated circuit design and layout more complex.

4.2.1. The MOS transistor as a switch

Ideally, a switch should have either zero impedance or infinite impedance depending upon its state. It should be able to change state instantaneously and the switching signal should effect only the state of the switch but have no other effects upon the signal itself. An MOS transistor switch comes close to fulfilling some of these requirements, but has certain drawbacks which must be considered. The controlling gate electrode of an MOS transistor presents a purely capacitive load which can be driven by simple on-chip circuitry. The transistor impedance in its "off" state is nearly infinite while the impedance in the "on" state is low enough for most analog applications. The switching speed can be entirely adequate for signal frequencies up to the low megahertz range. The major drawback is the parasitic capacitance between the gate, where the switching signal is applied, and the two terminals of the switch, the source and the drain. This gate "overlap" capacitance can cause significant coupling into the signal path of the large voltage swing at the gate that is required to change the state of the transistor. The switching signal at the gate normally must swing completely between the power supply rails to insure that the transistor will stay completely "on" or "off"

for any analog signal voltage on its source or drain. Unfortunately, as switches are made smaller to reduce this overlap capacitance, the "on" resistance of a closed switch increases. A typical minimum size MOS switch can have about 10000 ohms of resistance in the "on" state, leading to a 100 nanosecond time constant when used in conjunction with a 10 picofarad capacitor. Compared to most op-amp speeds, this is still negligible for most audio applications. Thus, the analog switches are usually made minimum size and symmetric to reduce parasitic capacitances.

For analog applications, only enhancement mode MOS transistors can be used as analog switches to insure that the switch can be turned off. For typical metal-gate MOS processes the threshold voltage is near +1 volt for n-channel enhancement transistors and -2 volts for p-channel enhancement transistors. For both NMOS and PMOS enhancement devices, a voltage greater in magnitude than $|V_{th}|$ must be maintained between the gate and the source for the transistor to remain on. Thus, even if the gate voltage swings between the two supply voltages ($\pm V_{supply}$), the analog signal can have a maximum range of only $2V_{supply} - V_{th}$ when only one polarity of MOS transistor is used. Although in practice the signal range is limited by the op-amp, the output swing of an op-amp can also depend upon the transistor thresholds. Therefore, V_{th} should be minimized to allow maximum dynamic range. Because threshold voltages cannot be accurately controlled, enhancement thresholds are usually greater then .5 volts.

When two transistors of different polarities are combined in parallel, the analog voltage swing is no longer dependent upon the switch thresholds but becomes limited only by the op-amp output range. Also, with this CMOS switch the "on" resistance is reduced and the clock feedthrough through the overlap capacitance can be partially canceled since the polarity of the switching voltages are opposite.

Complete clock feedthrough cancellation cannot usually be realized with CMOS switches because of problems in matching the amount of lateral diffusion between the p-type and n-type source/drain diffusions. These three advantages obtained with CMOS switches are not usually significant enough to justify the increased complexity in layout and design.

Another charge cancellation scheme for reducing clock feedthrough uses a "dummy" switch of the same polarity but half the size of the actual MOS transistor switch. The "dummy" switch is added with both terminals connected together to the critical node of the actual MOS switch so that its total parasitic gate overlap capacitance matches the gate overlap capacitance of the actual switch at a critical node in the circuit (such as an op-amp input). The "dummy" transistor is controlled by a clock of the opposite phase from the clock for the actual switch. This is often a more reliable method of canceling the unwanted charge since transistors of like polarity can be made to match very closely. Also layout area is reduced compared to the CMOS approach because transistors of like polarity can be laid out closer together. In either case, extra inverter circuitry would be needed to provide the additional clock signal of the opposite phase.

The problem of clock feedthrough by the gate overlap capacitance is exacerbated when the signal voltage varies at the switch terminals since the gate voltage required to turn the transistor "on" depends upon these terminal voltages. When a switch is connected to a high impedance point such as a capacitor, the amount of clock feedthrough when the switch turns off will therefore depend upon the signal level. For an n-channel transistor a more positive voltage on the capacitor will allow a greater amount of charge to be injected as the switch is disconnected. After the switch turns off the gate voltage continues to change by $V_{signal} + V_{th} - V_{supply}$ injecting charge through the gate overlap capacitance. Thus

the clock feedthrough can be signal dependent and has the effect of changing the effective value of the storage capacitance. In addition, the threshold voltage changes in a nonlinear fashion as the source voltage changes due to body effect. This threshold modulation is more pronounced in heavily doped substrates [45].

$$V_{th}(V_{ss}) = V_{th0}+\gamma\left[\sqrt{|V_{ss}+2\varphi_f|}-\sqrt{|2\varphi_f|}\right]$$

and

$$\gamma = \frac{\sqrt{2q\,\varepsilon_{si}\,\varepsilon_0 N_{bulk}}}{C_{ox}},$$

where:

$|\varphi_f| = kT\ln N_{bulk}/n_i$ = the Fermi level,

C_{ox} = the gate to substrate capacitance,

N_{bulk} = the substrate doping concentration,

V_{ss} = the substrate voltage with respect to the source,

(T = Temperature in degrees Kelvin.)

The physical constants are:

q = the charge on an electron.

k = Boltzmann's constant, and

ε_{si} = the relative dielectric constant for silicon,

ε_{ox} = the relative dielectric constant for silicon dioxide,

ε_0 = the dielectric constant for a vacuum,

n_i = the intrinsic silicon carrier density.

The Fermi level is positive for p-type substrates (n-channel transistors) and negative for n-type substrates (p-channel transistors). For typical MOS processes, φ_f = 0.3 volts and γ = 0.25 $\sqrt{\text{volts}}$.

Channel charge redistribution and charge pumping are both clock waveform dependent error mechanisms in MOS switches. Their effects can be described by

what happens to the channel charge as the transistor turns off. As the clock fall time is decreased, their effects become more pronounced but are limited by the amount of channel capacitance under the gate. For short fall times and minimum geometry switching transistors, the effects are minimized and the channel charge becomes evenly distributed between the source and drain when the transistor is turned off. As the fall time is decreased, more of the channel charge goes to the side of the switch with the lower impedance (assuming an impedance mismatch across the switch). If the fall time is fast enough, the channel can become forward biased with respect to the substrate before the charge has a chance to flow out of the channel. Under this condition charge pumping occurs whereby the channel charge flows into the substrate and is lost. For a metal-gate process, both of these effects can be ignored since they are usually insignificant compared to clock injection through the gate overlap capacitance.

4.2.2. Capacitor parasitics

The capacitors used in a metal-gate MOS process have a metal top plate, a thin silicon dioxide dielectric about 1000 angstroms thick, and a diffused region of heavily doped silicon for the bottom plate. This is very similar in construction to an MOS transistor except that the metal top electrode lies over a heavily doped region. The bottom plate forms a diode junction with the silicon substrate which is reversed biased for isolation. Associated with the diode junction between the capacitor's bottom plate and the substrate is a depletion region with both a nonlinear capacitance and a small leakage current. The size of this depletion capacitance is roughly one-fifth the size of the oxide capacitor and is inversely proportional to the square root of the bottom plate voltage (V_{cap}).

$$C_{depletion} = A\sqrt{\frac{q\,\varepsilon_{si}\,\varepsilon_0 N_{bulk}}{2\,|\,V_{cap} - V_{ss} + \varphi_{bi}\,|}}$$

Where:

$\varphi_{bi} = kT \ln(N_D N_A / n_i^2)$,

$N_D = n^+$ doping concentration,

$N_A = p^+$ doping concentration,

A = the surface area of the bottom plate.

With a metal-gate process, only metal-over-n^+ capacitors should be used to minimize pinholes in the silicon dioxide dielectric. From this equation, it can be seen that the parasitic depletion capacitance can be minimized by using an N-well CMOS process since N_{bulk} is smaller for metal-over-n^+ capacitors not in the well.

The reverse-bias junction leakage current is mainly caused by metallic impurities and crystalline defects which disturb the silicon crystal lattice. The leakage current is therefore very process dependent. It is also voltage and temperature dependent. Although a weak function of voltage, the reverse-bias junction leakage current can increase by a factor of 1000 for the temperature range from 25 degrees C to 125 degrees C. A typical value of leakage current at room temperature (25 degrees C) for 10 volts of reverse bias is 10^{-14} amps/μm^2 [10].

Besides depletion capacitance and leakage current, diffusions have a distributed resistance of typically 20 - 100 Ω/square. (The number of squares in a wire is the length divided by the width.) This parasitic resistance can slightly increase the charging and discharging time constant. Its effect is usually insignificant unless the capacitor is made very long and narrow.

A major advantage to the use of capacitors in analog integrated circuits is the precision to which capacitor ratios can be controlled [33]. Careful capacitor layout and circuit design that takes advantage of capacitor matching can realize gains and filter coefficients that are controlled to within 0.1%. Likewise, MOS transistors of the same polarity can be carefully matched in order to control

clock feedthrough. An example of this is the use of a dummy transistor for canceling charge injection from the clock.

4.2.3. Interconnects

There are only two types of interconnections used to route signals in a metal-gate process: metal lines, and diffusions. Metal lines are nearly ideal for routing signals because their parasitics are so small. Aluminum wires have a very small resistance of 0.03 Ω/square. The capacitance to the substrate is determined by the thickness of the field oxide which usually is about 10,000 angstroms, ten times thicker than the oxide capacitor's dielectric. Aluminum can handle a few milliamps per μm of line width [34] which is more than sufficient for carrying most signals.

Diffusion lines have all of the parasitics of a capacitor because they are just reversed biased diodes to the substrate. The resistance of diffusion wires can be significant. This combination of parasitics makes diffusion wires undesirable for most interconnects. Thus, metal wires should be used as much as possible except where two signals must cross, making the use of diffusion wires unavoidable.

4.2.4. Op-amp limitations

The op-amp is a critical part of any switched capacitor filter. Besides providing gain and buffering, an op-amp is necessary for complete charge transfer from one capacitor to another throughout a circuit. Realizable integrated circuit op-amps have limitations. Even when the length of the sample period is unrestricted, the accuracy of an op-amp in transferring charge is limited. Finite gain will cause incomplete transfer of charge in a circuit such as figure 3.7. A gain of a few thousand usually provides sufficient accuracy for most applications. If a CMOS process is available then high gain op-amps are not difficult to realize.

Actual integrated circuit op-amps usually require a small voltage at the input to produce zero volts at the output. This voltage is called the input offset voltage (V_{off}). In typical MOS op-amps, V_{off} can range from a few millivolts to a hundred millivolts, depending upon the process and circuit design.

If the effects of finite voltage gain (A_v) and input offset voltage are included, then the circuit in figure 3.7 will have the following transfer function.

$$V_{out} = \left[\frac{A_v}{A_v - 1}\right]\left[\frac{C_i}{C_f}V_{in} + V_{off}\right]$$

The amount of time required for an op-amp to respond to a change in its inputs and settle to an equilibrium value depends to a certain extent upon the amount of capacitance it must drive. An MOS op-amp can be made to settle to 0.1% of the final equilibrium output voltage in about 500 nanoseconds although the typical settling time is closer to a microsecond. The speed of an op-amp is many times slower than the time constant for charging a capacitor through a single MOS switch and is usually the dominant factor in setting the signal bandwidth of a switched capacitor filter.

Because of the size and number of MOS transistors in an op-amp, the op-amp often contributes the most noise to a filter and limits the dynamic range. A typical equivalent input noise voltage for an MOS op-amp is $100nV/\sqrt{Hz}$. The input noise current is almost zero since ideally no current flows through the gate of the MOS input transistors. Op-amp noise can be modeled by a noise voltage source from the non-inverting input to ground. For a typical switched capacitor circuit as in figure 4.1, this noise is amplified by the op-amp with a gain of $(C_i + C_{stray})/C_f$ where C_{stray} is the sum of all parasitic capacitance to ground. Since the signal gain is only C_i/C_f, C_{stray} should be minimized in order to maximize the dynamic range. Thus the top plates of both capacitors (C_i and C_f) should be connected to the inverting op-amp input to minimize the contribution of parasitic junction

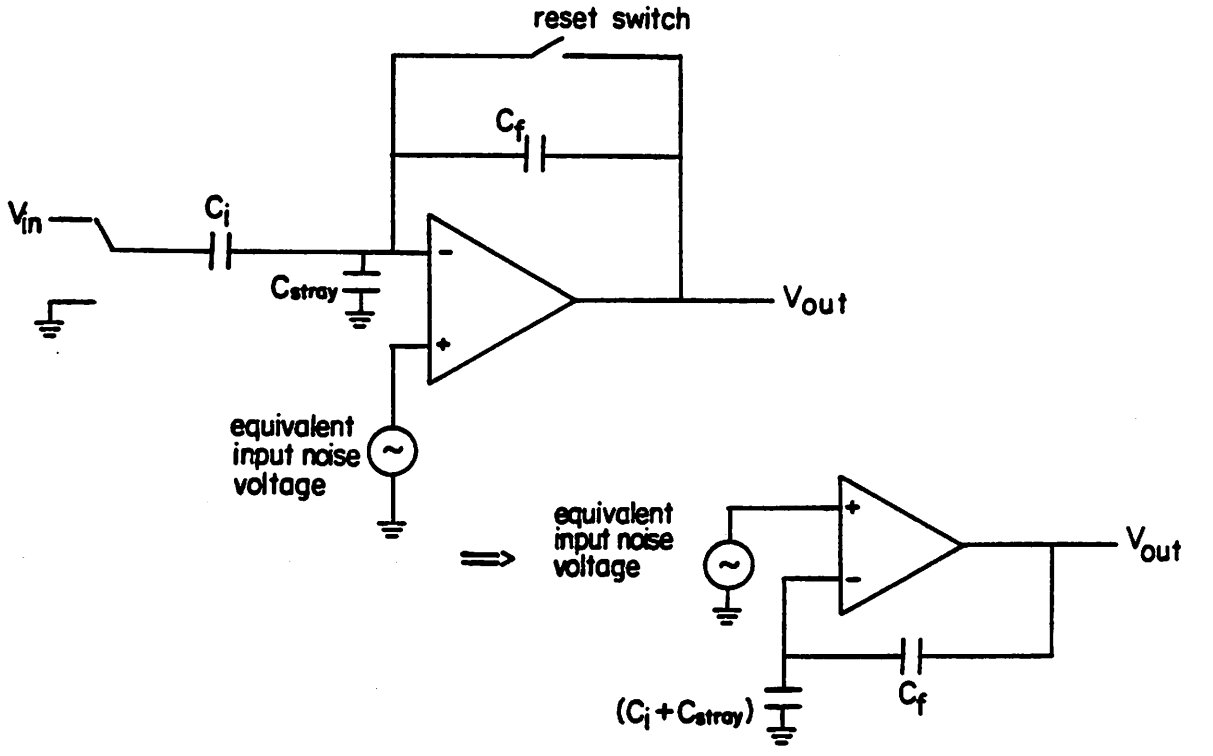


Figure 4.1. Noise model for basic switched capacitor circuit

capacitance to C_{stray}.

However in certain circuits, such as the delay line described in chapter three, mismatches in capacitor sizes and clock feedthroughs of the MOS switches can be the dominant source of noise. A subsequent section in this chapter will analyze the noise sources for the analog analysis filter implementation.

4.3. Circuit considerations

With the preceding discussion as background, the optimal use of switched capacitor circuit elements in the implementation of the analog analysis filter will now be presented.

4.3.1. Assumptions on parasitics

The integrated circuit process that was designed for this switched capacitor filter is an inverted CMOS process that uses a p^--type substrate and an n^- well.

This minimizes the depletion capacitance of the metal-over-n^+ capacitors and the body effect of the n-channel transistors while allowing larger breakdown voltages and higher NMOS transconductance [4]. Thus compared to conventional CMOS processes, larger dynamic range and signal to noise ratios are possible.

To simplify the error analysis of the switched capacitor circuitry the following assumptions about the integrated circuit parasitics will be made. For the MOS capacitors, only the depletion capacitance (C_d) and leakage current (i_d) will be considered. Where ever possible, interconnects will use metal lines so that distributed resistance and capacitance can be ignored. Only in complicated circuits such as the delay line will the depletion capacitance from cross-unders be considered.

The analog switches consist of single minimum-size n-channel MOS transistors. All analog switches are symmetric and identical to all other analog switches so that their parasitics can match. The gate overlap capacitance (C_p) is assumed to be the dominant transistor parasitic. The transit time across the channel is assumed to be small compared to the clock transition time so that quasi-static equilibrium conditions can be assumed. Thus, charge pumping and other channel charge effects will be ignored.

Although threshold modulation can contribute to signal distortion, this will also be ignored since it is insignificant compared to other types of signal dependent error in the filter circuit. Therefore, V_{th} is assumed to be constant. For most circuits the source and drain junction parasitics will be ignored or lumped in with the storage capacitor bottom plate parasitics unless a number of transistors are wired together (as in the delay line).

The op-amp gain is assumed to be large enough so that it will not contribute much error. However, the op-amp offset voltage (V_{off}) is usually significant and

will be the only source of op-amp error considered.

The clock period is assumed to be long enough for sufficient settling of the op-amp and so that the finite on resistance of the switches can be ignored.

4.3.2. Error analysis of basic circuits

In this section, several different switched capacitor circuits will be presented which can be used as building blocks in signal processing systems. Error analysis will be performed and the relative merits of each circuit discussed so that an optimal circuit for use in the Lattice LPC analysis filter can be chosen. These circuits are configured as sample and hold blocks that can implement the Lattice filter functions described in chapter 3 so that their performance characteristics in this application can be compared.

4.3.2.1. Basic switched capacitor integrator

Figure 4.2 is a schematic diagram of a basic switched capacitor integrator showing the important integrated circuit parasitics. On φ_1 the input capacitor (C_i) charges to the input signal while the feedback capacitor (C_f) is discharged by the reset switch. On φ_2 the signal charge on C_i is transferred to C_f where it is stored for outputting. The resulting op-amp output voltage after φ_2 turns off is described to first order by equation 4.1.

$$V_{out} = -V_{in}\frac{C_i}{C_f}\left[1+2\frac{C_p}{C_i}\right]+(V_{ss}+V_{th}-V_{off})\frac{C_p}{C_f}+V_{off}\left[1+\frac{C_i}{C_f}(1+2\frac{C_p}{C_f})\right] \qquad (4.1)$$

Here $\pm V_{ss}$ is the supply voltage and V_{off} is the op-amp input offset voltage. The switch parasitics (C_p) add to the input capacitor (C_i) to cause a gain error of ($1+2C_p/C_i$). Additional signal dependent charge injection introduced by the input switch Q_1 as it turns off is approximately canceled when switch Q_2 turns on. Charge injection from Q_2 and the op-amp offset voltage contribute a constant output offset error when φ_2 is turned on. When φ_2 turns off, additional charge is

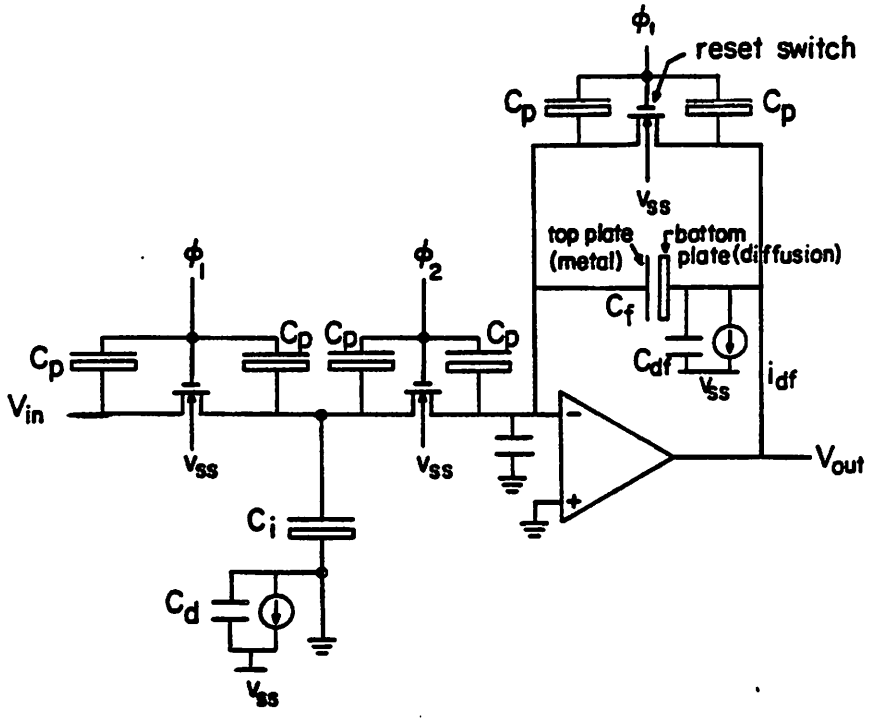


Figure 4.2. Basic switched capacitor integrator circuit

injected by Q_2 which results in a constant error term of half the magnitude but opposite in sign from the contribution of Q_2 when φ_2 was on. These constant error offsets are represented by the last two terms in equation 4.1.

4.3.2.2. 'Parasitic free' integrator

In figure 4.3 the schematic diagram for a "parasitic free" integrator is shown. This circuit is commonly referred to as "parasitic free" in the literature [8] because the gate overlap capacitance of the analog switches does not directly contribute to gain error as it does in the basic integrator circuit of figure 4.2. The input side of C_i switches between two low impedance points so that any parasitic capacitance not in parallel with C_i will not affect the total signal charge when one of the input switches, Q_{1b} or Q_{1b}, are on. However, there is some signal dependent charge injection introduced when the switches change state. The op-amp side of C_i switches between actual ground and the virtual ground of the

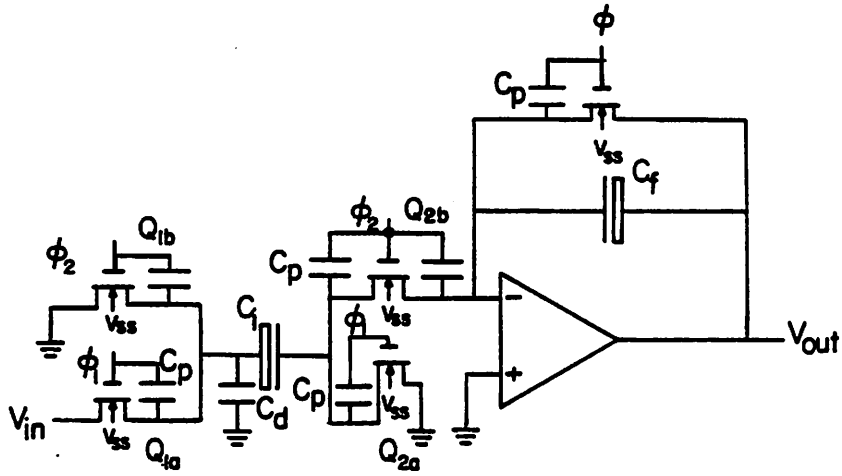


Figure 4.3. Schematic diagram for improved "parasitic free" integrator

inverting input of the op-amp so parasitic capacitance at that node cannot contribute charge to C_f.

On φ_1, the input capacitor (C_i) is connected between the input signal and ground through switches Q_{1a} and Q_{2a} while the feedback capacitor (C_f) is discharged by the reset switch. The charge that was stored on C_i during φ_1 is transferred to C_f on φ_2 when switches Q_{1b} and Q_{2b} close. This connects the input capacitor between actual ground and virtual ground which transfers the charge from C_i to C_f as C_i discharges. Since virtual "ground" of the inverting input of the op-amp is actually the voltage V_{off}, when the op-amp side of C_i switches from ground a constant offset error proportional to V_{off} is introduced. The op-amp output voltage after φ_2 turns off is given to first order by equation 4.2.

$$V_{out} = V_{in}\frac{C_i}{C_f}\left[1-\frac{C_p}{C_i(2+C_d/2C_p)+C_d+2C_p}\right]-V_{off}\left[1+\frac{C_i}{C_f}\right]+2(V_{ss}+V_{th}-V_{off})\frac{C_p}{C_f}$$

(4.2)

Signal dependent clock feedthrough through C_p of Q_{1a} gives rise to a gain error as shown in the first term of equation 4.2. This error is due to the signal dependence of the "on" voltage of Q_{1a} which is caused by the source terminal being connected to V_{in}. In the absence of all parasitic capacitance to ground, the gain error for this "parasitic free" integrator is roughly four times better than for the basic integrator of figure 4.2. However, parasitic capacitance to ground (such as C_d) on the input side of C_i will absorb some of the clock feedthrough that would be injected onto C_i and further reduce the gain error.

Like the previous circuit, the op-amp offset error voltage is increased by $V_{off}\, C_i / C_f$ when φ_2 turns on. However, this circuit has twice the charge injection as shown by the last term of equation 4.2.

4.3.2.3. Switched capacitor differencer

When the "parasitic free" integrator is used as a sample and hold, the reset switch across the feedback capacitor eliminates the need for switches Q_{2a} and Q_{2b} of figure 4.3. The function of these switches is to charge and discharge C_i independent of C_f. But in a sample and hold circuit, C_f is reset on φ_1 while C_i is sampling the input so C_i can be connected directly to the op-amp input without affecting the discharging of C_f. Thus the use of switches Q_{2a} and Q_{2b} becomes redundant. If these switches are removed, the resulting circuit is a switched capacitor differencer which is shown in figure 4.4. This decrease in circuitry not only simplifies construction, but also eliminates some of the sources of error found in the "parasitic free" integrator.

Although the basic operation of this circuit was described in section 2.3 of chapter 3, a basic error analysis for the switched capacitor differencer is presented here. Equation 4.3 shows the effects of the significant error

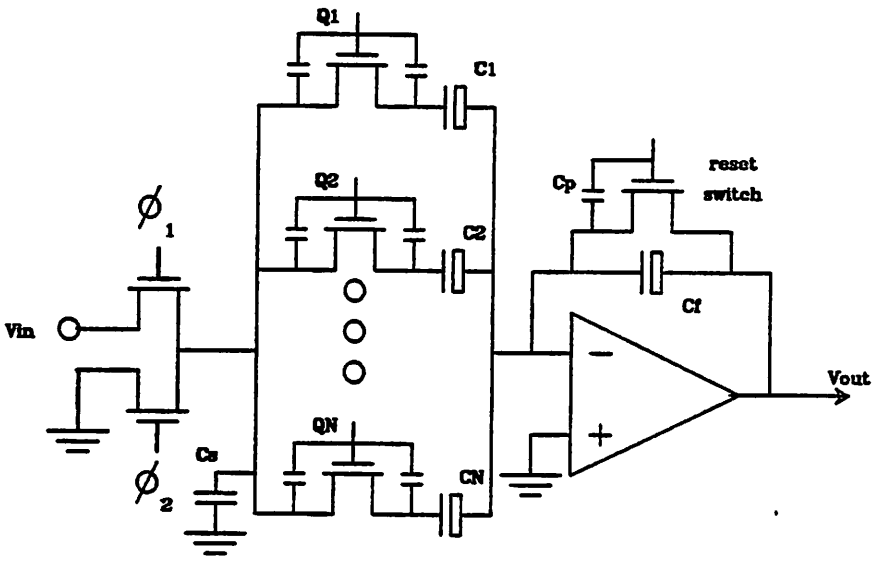


Figure 4.4. Switched capacitor differencer circuit

mechanisms on the switched capacitor differencer implemented as the sample and hold of figure 4.4.

$$V_{out} = V_{in}\frac{C_i}{C_f} + (V_{ss}+V_{th}-V_{off})\frac{C_p}{C_f}\left[1+\frac{C_i}{2C_p+C_i+C_d}\right]-V_{off} \qquad (4.3)$$

Equation 4.3 exhibits several interesting characteristics. First, there is no gain error. This happens because signal dependent feedthrough is eliminated by the reset switch during φ_1 and the input capacitor is switched to ground on φ_2. If instead of being grounded on φ_2, C_i was switched to a second signal voltage (V_{in2}) to form a differencing circuit, then the charge injection when φ_2 turns off would become dependent upon this second input voltage. The error signal would be roughly equal to $V_{in2}\frac{C_p}{C_f}\left(\frac{C_i}{C_i+C_d+2C_p}\right)$. However, if the output is taken during φ_2, then this error signal would not be seen.

A second advantage of this switched capacitor differencer over the two integrator circuits is that the error due to the op-amp input offset voltage is not amplified. In fact, the term $-V_{off}$ is constant regardless of the clock phase, a detail that can help to cancel it.

Also, the parasitic junction capacitance of the input capacitor (C_d) can help reduce clock injection. The second component in equation 4.3 has a component $\left[1+\frac{C_i}{C_i+C_d+2C_p}\right]$. If C_d is increased, then the voltage of this term decreases. Even if C_d is equal to zero, this source of error will still be less than the equivalent error in the circuit of figure 4.3.

4.3.3. Error cancellation schemes

To first order, both of the error terms in equation 4.3 for the switched capacitor differencer can be canceled. The effect of the op-amp offset voltage (V_{off}) on successive stages of signal processing can be canceled using differential circuitry. This is done by feeding the output of the differencer to another stage where the input depends upon the difference between the differencer outputs at φ_1 and φ_2. The V_{off} error term during φ_2 is the equal to the differencer output during φ_1 so that the input to the next stage will not depend on V_{off}. Although this second stage might have an offset voltage of its own, the error from V_{off} will not build up as the signal propagates. Regardless of the number of stages, the maximum op-amp offset error contributed to the output after any number of signal processing stages is just the op-amp offset of the last stage.

Most of the charge injection due to the clock feedthrough of the switches can be canceled by cascading two differencer stages. In this scheme, the differencer for the second stage is implemented as a unity gain inverter. The polarity of the clock feedthrough error from the first stage is inverted by the second stage, yet

the magnitude remains the same. The clock feedthrough error generated by the second stage will have the opposite polarity and cancel the clock feedthrough error from the first stage.

A circuit that implements both of these error cancellation schemes is shown in figure 4.5. Here both op-amps and all switches are identical. Although the gain of the second stage is unity, the gain of the first stage can vary by changing C_i. C_f is the same for both stages.

Because the reset switches and feedback capacitors are identical, the clock feedthrough error after φ_1 turns off will be the same for the output of both op-amps. Thus if the output signal is sampled during φ_{2b} there will be no clock feedthrough error present for any values of C_i and C_f.

If φ_{2b} of the second stage turns off after φ_{2a} of the first stage, then charge injection from Q_2 will propagate to the second stage. After φ_{2b} goes low, turning

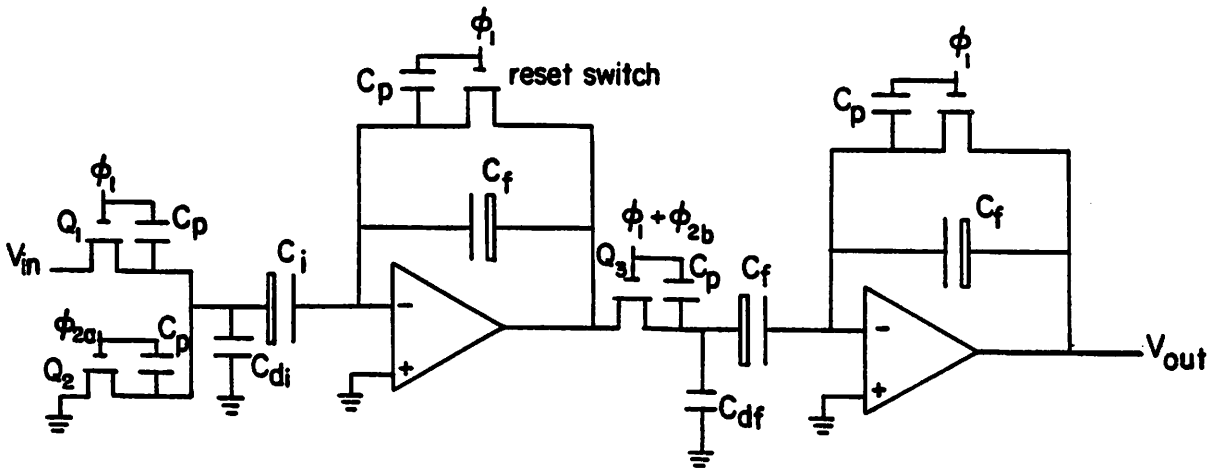


Figure 4.5. Switched capacitor differencer with error cancellation

off Q_3, the total clock feedthrough error at the output will be:

$$(V_{ss}+V_{th}-V_{off})\frac{C_p}{C_f}\left[\frac{C_f}{C_f+2C_p+C_{df}}-\frac{C_i}{C_i+2C_p+C_{di}}\right].$$

If $C_i = C_f$ or both C_i and C_f are much larger than the parasitic capacitances, then nearly all of the clock feedthrough error will be canceled.

4.3.4. Delay line analysis

The delay element is an essential part of the lattice algorithm yet it is also the most sensitive to parasitics. As the storage capacitors are switched to recall delayed signals, charge injection from the switches will interfere with the stored signal. Signal dependent feedthrough can lead to crosstalk under certain conditions. Mismatches in the parasitics will cause an error signal to be superimposed on the delay line output. Proper design can help to minimize these errors. In this section, circuit configurations and significant error mechanisms are analyzed in order to produce an optimal delay line design. Although this circuit was specifically designed for use in a lattice filter, it can be expanded as a general purpose delay line for audio applications such as echo units and comb filters.

4.3.4.1. Multiple storage elements-configurations

There are two possible configurations for the delay line. The multiple storage capacitors can be wired either as an array of input capacitors or feedback capacitors. A configuration using multiple feedback capacitors [6] is shown in figure 4.6.

The feedback capacitors can be switched either at the input or output node. If switched at the output node, signal charge on a feedback capacitor can cause signal dependent feedthrough to accumulate on the parasitic op-amp input capacitor (C_{stray}). This charge would then be transferred onto the next storage capacitor to be switch in, causing crosstalk. If the capacitors are switched at the input node, signal dependent error would be eliminated since the op-amp input

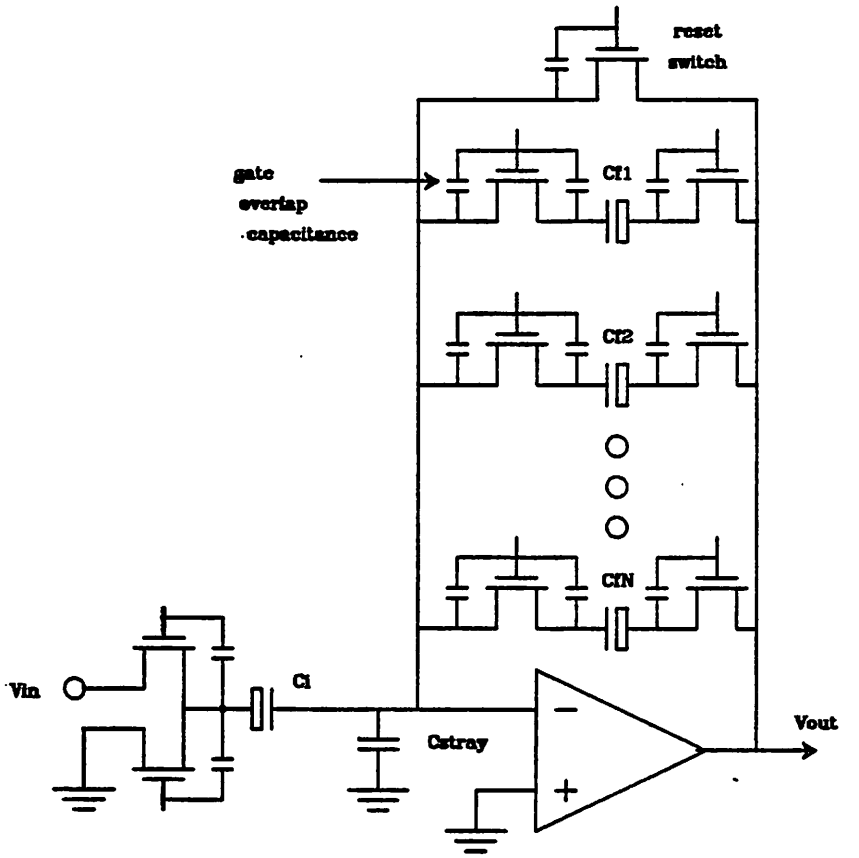


Figure 4.6. Delay line configuration using multiple feedback capacitors

sits at a virtual ground. However, another problem may occur. If the op-amp output swings negative, the input side of the feedback capacitor can be bootstrapped below the negative supply for a stored positive signal voltage on that capacitor. This would cause the switch to turn on, thereby discharging the storage capacitor.

The only obvious solution would be to include both input and output switches on the feedback capacitors as in figure 4.6. First, the input switch (Q_{1a}) would be opened, causing a constant offset to appear on C_{stray}. Then, the output switch (Q_{1b}) would be opened to prevent bootstrapping. To eliminate any offset appearing across C_{stray}, the reset switch can be closed at this point, provided that an extra stage of delay is available to allow the op-amp to settle. After the reset switch opens, the next storage capacitor can be switched in by closing output switch Q_{2a}. Finally, the input switch (Q_{2a}) would close, thereby closing the feed-

back loop.

This method of storing multiple signals requires precise and complicated timing. Additional time is necessary whenever the feedback loop is closed in order to give the op-amp time to settle. This amounts to two clock cycles more for this circuit than if input capacitors were switched, one for the reset switch to discharge offsets, and one for the next storage capacitor to be switched in. The only advantage to switching feedback capacitors is that it allows non-destructive recall of a stored signal.

The other alternative, using multiple input capacitors, leads to much simpler timing. As described in chapter 3, only one switch in series with the input capacitor is necessary. This is the configuration that will be used to implement the delay function of the lattice filter. An analysis of parasitics and error mechanisms for this circuit will be presented in the next two sections.

4.3.4.2. Fixed pattern noise

Even with zero signal stored on the multiple input storage capacitors, a signal can be produced at the delay line output because of mismatches in the parasitic feedthroughs. This repetitive noise signal is called fixed pattern noise. It has a period which is as long as the number of stages in the delay line, eleven for this application of a ten stage lattice filter. In a typical IC design, threshold and overlap parasitics will contribute fixed pattern noise about 80db below the maximum signal level. However, layout irregularities can contribute mismatches significantly greater than this. In particular, clock lines running near a storage capacitor can couple additional offset to that capacitor. Also, to reduce op-amp noise, the op-amp input capacitance is minimized by connecting the metal top plate directly to the op-amp input. Mismatches in the leakage current of the diffusion bottom plate will therefore contribute to fixed pattern noise since the

leakage current affects the side of the capacitor which stores the signal.

4.3.4.3. Parasitic Error Analysis

An error analysis is presented here based upon the delay line schematic of figure 4.7. The significant parasitics are the overlap capacitances (C_p) of the series input switches (Q_1 through Q_N) and stray capacitance (C_s) at the node where these switches are connected together. Because the feedback loop in this circuit is never broken, op-amp input capacitance cannot introduce error since it always is at ground potential.

When the sample switch turns on, the reset switch is closed and a storage capacitor (C_x) charges through Q_x. Then, both the sample switch and the reset switch turn off letting the input node of C_x float. Charge injection on C_x is the same as for a simple switched capacitor differentiator.

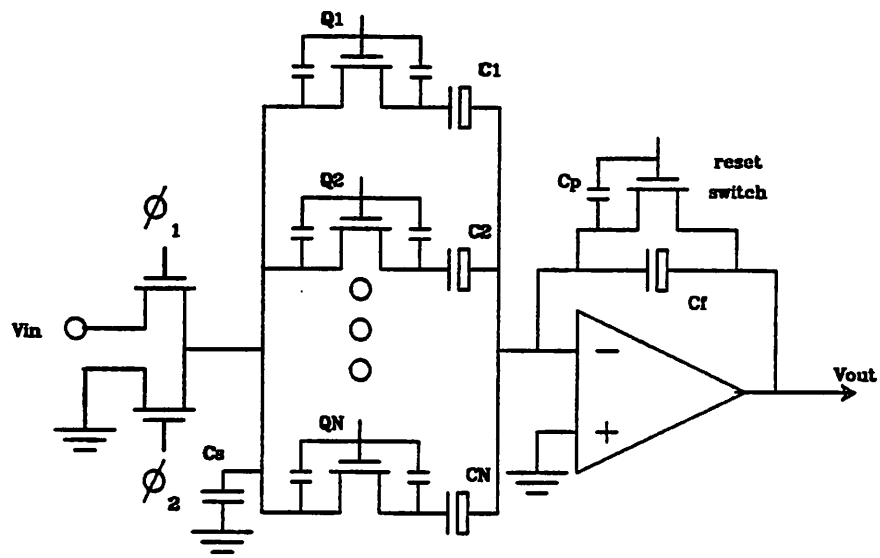


Figure 4.7. Delay line schematic with significant parasitics included

Next, Q_x turns off and Q_{x+1} turns on, accessing the stored signal on C_{x+1}. Charge injected onto C_x from Q_x turning off also appears on the feedback capacitor (C_f). As in equation 4.3, depletion capacitance of C_x helps to reduce the amount of injected charge by charge-sharing this feedthrough. Injected charge onto C_{x+1} when Q_{x+1} turns on is also accumulated on C_f as before but with the opposite polarity. Thus, constant feedthrough is canceled leaving only signal dependent error components on C_f. The error voltage on C_f is proportional to the voltage on C_x and adds to the voltage on C_{x+1} when Q_{read} closes. This source of error therefore contributes to crosstalk. The magnitude of the crosstalk is a complicated function of the clock waveform, the transistor switching speed, and the parasitic capacitances. The larger the capacitance C_s, the smaller the crosstalk, since the voltage at that node stays more constant so that more of the signal dependent feedthrough can be canceled.

There are two possible clock modifications which can reduce this crosstalk. By keeping the op-amp reset when Q_x turns off, charge injection to C_f is eliminated. After turning off the reset switch, charge injected when Q_{x+1} turns on will flow back out of C_f when the read switch is shorted to ground. The other possible clock change would be to leave the sample switch connected to V_x while Q_x and Q_{x+1} change state, but not leave the reset switch on. Since the input node voltage cannot vary, signal dependent feedthrough will be completely canceled. The charge, $(V_x - V_{x+1})C_i$, which flows onto C_f will flow back out when the read switch turns on and connects C_{x+1} to ground.

4.3.5. Analog LPC analyzer circuitry

A circuit schematic for the analog portion of the Lattice LPC analyzer of figure 2.4a is shown in figure 4.8. This circuit has been optimized for integrated circuit implementation and uses the parasitic cancellation schemes of section

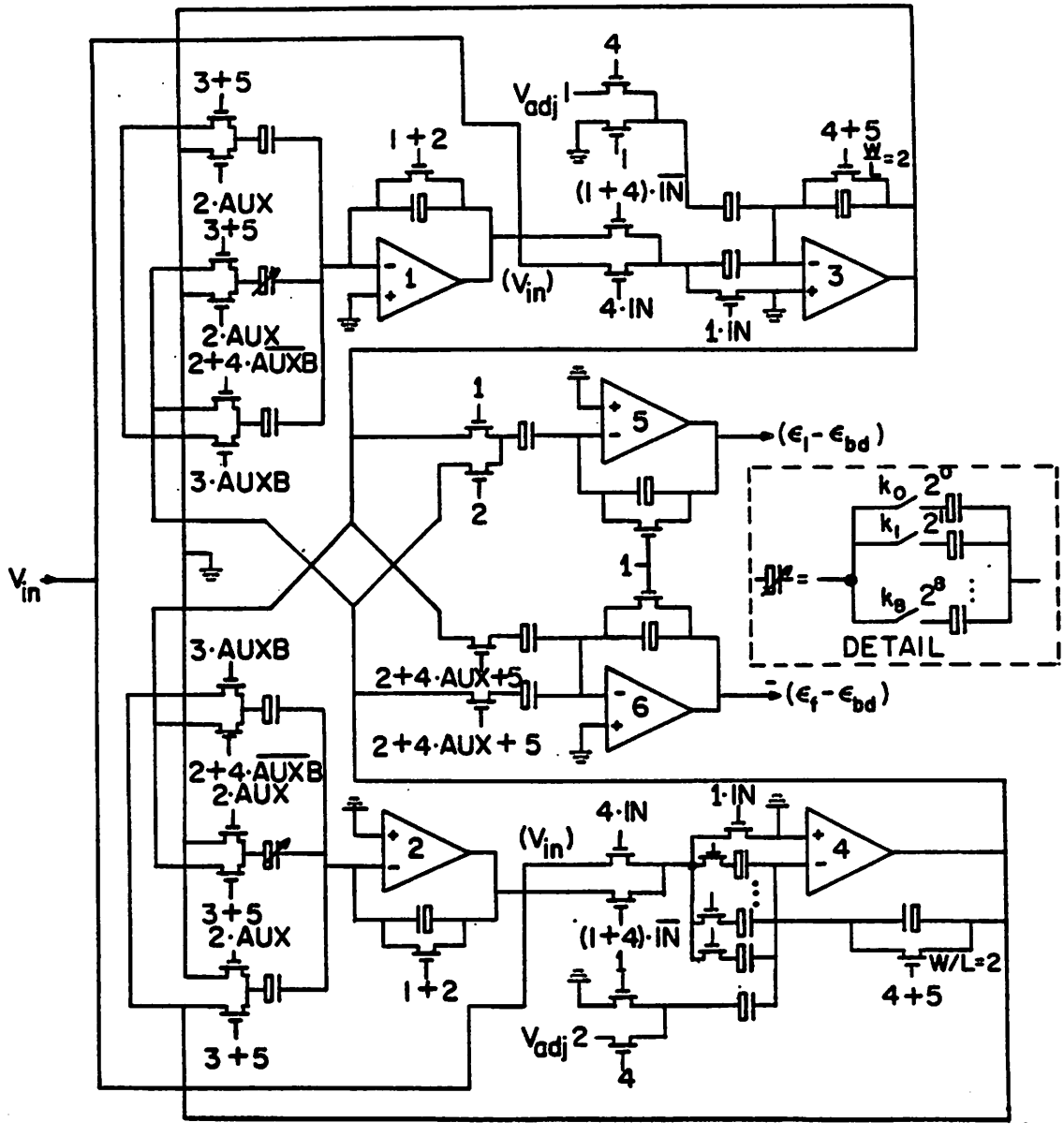


Figure 4.8. IC implementation for switched capacitor lattice analysis filter

4.3.3 wherever possible. Op-amps Op_1 and Op_2 implement the MDAC, addition, and subtraction functions while Op_3 and Op_4 provide the sample and hold operations that enable time multiplexing of this circuit. Op-amps Op_1 and Op_3 process the forward residual signal and op-amps Op_2 and Op_4 process the backward residual. Op-amp Op_4 also implements the delay function in the backward path. Op-amps Op_5 and Op_6 produce the sum and difference residual outputs for use with the correlator circuitry.

Excluding the MDACs, the op-amp pairs Op_1, Op_3 and Op_2, Op_4 are each inverting unity gain sample and hold circuits similar to the circuit of figure 4.5. As with that circuit, the effects of the major integrated circuit parasitics are canceled. The outputs of each op-amp pair are feed back to their respective inputs to begin another stage of computation in the Lattice algorithm. The signal inversion does not adversely affect the implementation of the Lattice algorithm since the algorithm is linear with respect to the forward and backward signal paths so multiplication of the residual inputs by the same scalar merely carries through to the output. Computation of the reflection coefficients is not affected since the sum and difference correlator inputs are immediately squared making their sign irrelevant. If an even number of stages of the Lattice filter are implemented, then the final residual error output of the filter is of the correct polarity.

Because the input to each op-amp depends upon the difference between V_{off} of the previous op-amp the signal plus this offset, the effect of the op-amp input offset voltage is canceled within the filter. However, the sum and difference outputs to the correlator include the offsets of op-amps Op_5 and Op_6 which could not be canceled within the circuit. The same outputs also include the errors from clock injection through the switches. These errors can be taken care of in several ways. One approach would be to include a "dummy" differencer on the chip which

would be clocked without a signal to produce identical constant offset errors (charge injection and V_{off}) and match the errors of the output op-amps. This error output could be used as a "ground" reference for the companding A/D converter in the correlator. Another technique would be to integrate a switched capacitor companding A/D converter on the same chip which uses the previously mentioned error cancellation schemes.

The major disadvantage in designing this type of circuit is the complexity of the clocking. This circuit requires a five phase non-overlapping clock and various other clock pulses to control the multiplexing. Many of the switches require combinations of the various clock pulses. When two successive clock phases are combined to drive a switch, additional circuitry is required to remove the "non-overlap" period between the pulses so that a single continuous pulse can be formed. The greatest problem is not in generating the clocks, which can be done with ROM look-up tables, but for the designer- to keep track of the circuit operation. Parasitic cancellation techniques tend to complicate the circuit operation, making the timing of the circuit blocks more interdependent.

4.3.5.1. Operation

On φ_4, the sample and hold op-amps (Op_3 and Op_4) are reset and the input is sampled on their input capacitors. During φ_1 these op-amps remain reset while the input capacitors for the backward path (Op_4) are switched to recall the delayed input from the previous sample period. Op_4 has eleven input capacitors to provide the delay storage for a ten stage multiplexed filter. These capacitors are successively switched on φ_5 in a cyclic fashion. φ_1 then goes high placing the sample and hold inputs at V_{off} and transferring the signal to the sample and hold outputs.

Because the MDACs of Op_1 and Op_2 are four quadrant, there exist both inverting and non-inverting signal paths to provide for the offset binary operation. The MDACs use a variable capacitor which is implemented as eight switchable binary weighted capacitors equal in value to the feedback capacitor. This variable capacitor is clocked with a negative gain. The capacitor that provides the offset for four quadrant operation is therefore clocked with positive gain. The MDAC capacitors are connected together to the opposite filter signal path according to the Lattice structure- the MDAC for the forward path gets its input from the backward residual and vice versa. An additional capacitor provides the unity gain inverting path from the same type of filter signal path as required by the algorithm - the capacitor in the forward path gets its input from the forward residual and likewise for the capacitor in the backward path.

The inverting signal paths through Op_1 and Op_2 have their input capacitors reset to V_{off} on φ_5. This phase is a "don't care" state for the output of Op_1 and Op_2. Charge transferred to the feedback capacitors of these op-amps when the input capacitors are reset is eliminated during the next two clock phases (φ_1 and φ_2) when the op-amps are reset. On φ_3 the inverting input capacitors are allowed to charge to the input signal, transferring the negative of their signal to the output of Op_1 and Op_2 in the process.

The input capacitors of Op_1 and Op_2 that are clocked with positive gain store their input signal on φ_2 while these op-amps are reset. Their charge is added to the feedback capacitor by changing the input voltage to V_{off} on φ_4 when Op_3 and Op_4 are being reset. This completes one cycle for one stage of computation in the Lattice filter algorithm.

Op-amps Op_5 and Op_6 are reset on φ_1. Op_5 subtracts the two residuals by charging its input capacitor to the output of Op_3 on φ_1 and then to Op_4 on φ_2

resulting in $(f_m(n)-b_m(n-1))$ at the output. Op_6 adds the two residuals together by discharging its input capacitors on φ_5 to V_{off}, discharging its feedback capacitor on φ_1, and then charging the input capacitors to the outputs of Op_3 and Op_4 on φ_2. This produces $-(f_m(n)+b_m(n-1))$ at the output. The outputs of these two op-amps are held constant from φ_2 through φ_4 for processing in the correlator. With a ten stage filter, a five phase clock, and an 8khz sample rate, there is about 7.5μsec available for the companding A/D converter to digitize the signal, which is sufficient time for an eight bit conversion (including sign bit).

The variable capacitors of the MDACs are first accessed during the φ_5 prior to φ_3 when they are charged to the forward and backward residuals of the previous filter cycle. Since these residual input signals are not available until φ_2, it would be very difficult to provide the current reflection coefficients in time for their use by the MDACs. However, since no signal on the switchable capacitors is transferred to the op-amp output until φ_3, all of these switchable capacitors in the MDACs can be reset to V_{off} on φ_5 instead of just the capacitors that would be selected by k_m. In this case, the output of the correlator would not be required until φ_3. Yet, this would still leave only φ_2 for digitizing and computing the reflection coefficients. Extra time can be gained if we use reflection coefficients that are delayed by one sample period. The use of delayed reflection coefficients has very little effect on the performance of the analysis filter since the coefficients are low pass filtered as part of their computation. A delay of one sample period makes very little difference in the value of the reflection coefficients. Using delayed coefficients, an extra filter cycle of typically 12.5μsec is available for processing the residual signals in the correlator.

4.3.5.2. Error analysis

Most of the constant offset error can be eliminated from the analog lattice filter implementation by using the techniques described in previous sections. However, for the filter implementation of figure 4.8, signal dependent error components are introduced by the MDACs. In each of the two signal paths (forward and backward) there is a constant offset error due to charge injection produced by the non-inverting sample and holds (Op_3 and Op_4). For this circuit, these offsets cannot be canceled until they propagate through a successive stage with a gain of -1. A problem arises because the MDACs cross couple this error with variable gain. The error introduced is directly proportional to the reflection coefficients and is accumulated into the successive filter stages. The effect of this error cannot be seen until the second stage of filtering after k_1 modulates the offset through the MDACs. Thus the reflection coefficient, k_1, is itself not affected. Equation 4.4 shows the effects of this error on one stage of the analysis filter. Equation 4.5 shows how this error affects the value of the reflection coefficients and distorts the analysis filter spectral response.

$$F_m(z) = F_{m-1}(z)+k_m z^{-1}B_{m-1}(z)+k_m\alpha \tag{4.4a}$$

$$B_m(z) = z^{-1}B_{m-1}(z)+k_m F_m(z)+k_m\alpha \tag{4.4b}$$

$$k_m \approx \frac{E[(f_{m-1}(n)-b_{m-1}(n-1))^2]}{E[(f_{m-1}(n)+b_{m-1}(n-1)+2k_{m-1}\alpha)^2]} \tag{4.5}$$

The constant offset error at the output of Op_3 and Op_4 is represented by α. Here it is assumed that all constant offset errors to the correlator are canceled by either matched "dummy" stages integrated on chip or external offset adjustments of the A/D ground reference.

This error causes a shift in center frequency and Q of the formants in the spectral response. Because we assume that parasitic errors are matched and that the reflection coefficients vary slowly with time (due to the expectation

function), error terms in the numerator of equation 4.5 cancel while the error terms in the denominator add. The effect of this error is most pronounced when $(f_m + b d_m)$ is small. This sum can become small not only for small amplitude amplitude signals, but also near certain signal frequencies where the delay through the backward path causes a 180 degree phase shift.

$$f_{180} = f_s / 2(m+1) \tag{4.6}$$

However for speech, the reflection coefficients of successive stages tend to decrease in magnitude considerably as the amplitude of the residual errors decrease. As m increases, k_m has less of an effect on the spectral response and the first few reflection coefficients become the most important in determining the formant locations. Therefore, distortion from offset modulation is greatest at high signal frequencies as seen from equation 4.6 when m is small.

The choice of an analog circuit with optimal error properties to implement the analysis filter is very limited. The delay stage requires non-inverting clock phasing since a stored signal on one of the input capacitors is recalled by grounding the appropriate capacitor. Therefore, the MDAC stage must be the inverting stage to implement the charge cancellation. Yet, the offset from this delay stage is what is modulated by the MDAC to produce a significant error in the filter circuit.

An inverting delay stage can be made by placing the storage capacitors in the feedback path of the op-amp. Since the clock injection offset at the output of this stage would be canceled, the MDAC modulation error would be greatly reduced. However, there are severe crosstalk and switch feedthrough problems associated with switching feedback capacitors. Also an extra clock cycle would be required to recall the stored signal, making the circuit even slower.

4.3.5.3. Noise analysis

The major source of noise in the analysis filter is fixed pattern noise of the delay line. Mismatches in the switches and capacitors of a modulo eleven delay line in a ten stage filter can cause a noise pattern that repeats every 110 sample periods. Because the order of the eleven storage elements in the delay line precess with respect to the ten stage filter every sample period, there are effectively eleven different noise patterns. These noise patterns are convolved into each of the ten stages to produce the 110 sample noise output. In practice, this error limits the dynamic range of the analysis filter. Noise frequency components are produced with a fundamental frequency of $f_s/110$ for a ten stage lattice filter. The harmonic content depends upon the distribution of the mismatches and varies from chip to chip of the analog integrated circuit.

4.4. Process Technology and Layout Considerations

A metal-gate CMOS integrated circuit process was chosen. A metal-gate process has the advantage of ease of fabrication. CMOS has the advantage of complementary polarities of MOS devices, n-channel and p-channel, to design with. Fast, high gain op-amps are easier to design because of the increased flexibility in level shifts and active loads. Higher power supply rejection can also be obtained.

In particular, a high voltage inverted CMOS process was chosen to implement the lattice filter. In analog applications, high breakdown voltage provides higher dynamic range, but with some sacrifice in chip area. This inverted CMOS process is based on a previous inverted CMOS process developed at U.C. Berkeley [4]. Major modifications were made to enhance circuit speed by decreasing gate overlap capacitance. Junction depths were reduced and techniques were used to self-align the transistor channels with the metal gates. A complete process schedule for this process is provided in appendix C.

4.4.1. Advantages of Inverted CMOS

Conventional CMOS processes are based on metal-gate PMOS processes. P-channel devices are fabricated on the substrate an n-channel devices are put in a heavily doped well to provide positive thresholds. Inverted CMOS uses substrate n-channel devices and has n-type wells for the p-channel devices. An implant is used to control the threshold voltage.

Substrate NMOS transistors have the advantage of higher transconductance (g_m) because mobility is greater in lightly doped channels. Breakdown voltage is also higher because of the lightly doped channel and because the relatively high avalanche coefficient of electrons makes it difficult to control NMOS channel avalanching in a well [43].

PMOS devices make good current sources because they are in a well. The higher channel doping increases the output impedance. P-channel transistors also exhibit lower noise [7]. The non-uniform well doping causes sub-surface (buried channel) channel conduction which reduces the effect of surface defects on noise performance.

4.4.2. Layout

Basic layout constraints were as follows. The minimum oxide cut was 8 μm. The alignment tolerance was originally set at 1 μm, although this was later relaxed to 2 μm due to equipment problems during fabrication. The minimum metal linewidths and spacings were 10 μm.

Channel stops of p^+ diffusions were used around all n-channel transistors and n^+ diffusion lines. P-channel transistors did not require channel stops because of the increased channel doping of the well. Only metal over n^+ capacitors were used in order to reduce yield problems associated with oxide pin-holes over p^+

regions.

To achieve a 30 volt breakdown voltage, 2 μm junction depths were used with a lightly doped 5 to 7 ohm-cm substrate ($\approx 7x\,10^{15}/\,cm^3$). The p^--well doping was approximately $10^{16}/cm^3$. Channel lengths were 8 μm after diffusion. Diffusion spacing was kept at 13 μm on the mask.

A complete table of layout rules designed for this process are given in appendix D. Appendix D also discusses general analog switched capacitor layout guidelines.

4.4.3. Self Aligned Gates

Two techniques were used to decrease gate overlap capacitances, self-alignment implants and differential oxide growth rates. PMOS transistor gates were made narrower than their channel length on the mask. With thin oxide over the channel overlapping the source and drain diffusions, a boron implant, using the metal gate as the mask, extended the source and drain regions while minimizing the gate overlap area. The NMOS overlap capacitance was reduced by taking advantage of the difference in oxide growth rates over heavily and lightly doped regions [20]. Oxide grows faster over regions of heavy phosphorus (n^+) doping. This effect is enhanced by using long oxidation times at low temperatures in a steam ambient. Using the oxidation parameters presented in appendix C, a two to one ratio in oxide thickness was observed over n^+-diffusion and p^--substrate (channel) regions. The combination of these two techniques resulted in op-amps being fabricated with 500 nsec. settling times (to 0.1%). In the final runs, this technique had to be discarded since it was based on an 800 Å gate oxide, yet the fabrication facilities were not sufficiently clean to produce an acceptable yield with oxide this thin.

4.5. Op-Amp Design

The op-amp is a crucial element in determining the performance of the lattice filter. It must be fast enough for multiplexed operation of the filter and have sufficient gain to provide for accurate computations. Noise and layout area must also be optimized.

A transconductance op-amp was chosen since in a switched capacitor circuit, the op-amp generally must source or sink current to high impedance capacitive nodes. A low impedance output buffer would be unnecessary and would reduce op-amp speed. The design that was chosen is similar in operation to the RCA transconductance op-amp CA3080 [40]. A schematic diagram of the op-amp is shown in figure 4.9.

4.5.1. Operation

Transistors Q_1 and Q_2 form the input differential pair. Because they are sub-surface p-channel devices and have a large W/L, op-amp noise is minimized. Equivalent input noise voltage is reduced since the large g_m helps to isolate noise generated by the active loads when referred to the input.

Transistor triplets Q_3-Q_5 and Q_6-Q_8 both form Wilson current mirrors with a gain of 2.5. Q_3 and Q_8 are minimum size to minimize noise since they are directly connected to the differential pair drains. Transistors Q_4, Q_5, Q_7, and Q_8 each have W/L ratios of 2.5 to increase op-amp gain, provide additional noise isolation, and to allow increased operating current in the output to increase slew rate and capacitive loading capability. For the CMOS process available, computer simulations determined this value of gain to be optimal. Higher Wilson current mirror gain would increase settling time by lowering the frequency of second order poles. Lower gain would force higher input differential pair operating currents and increase the power dissipation to achieve the same output slew rate at the same

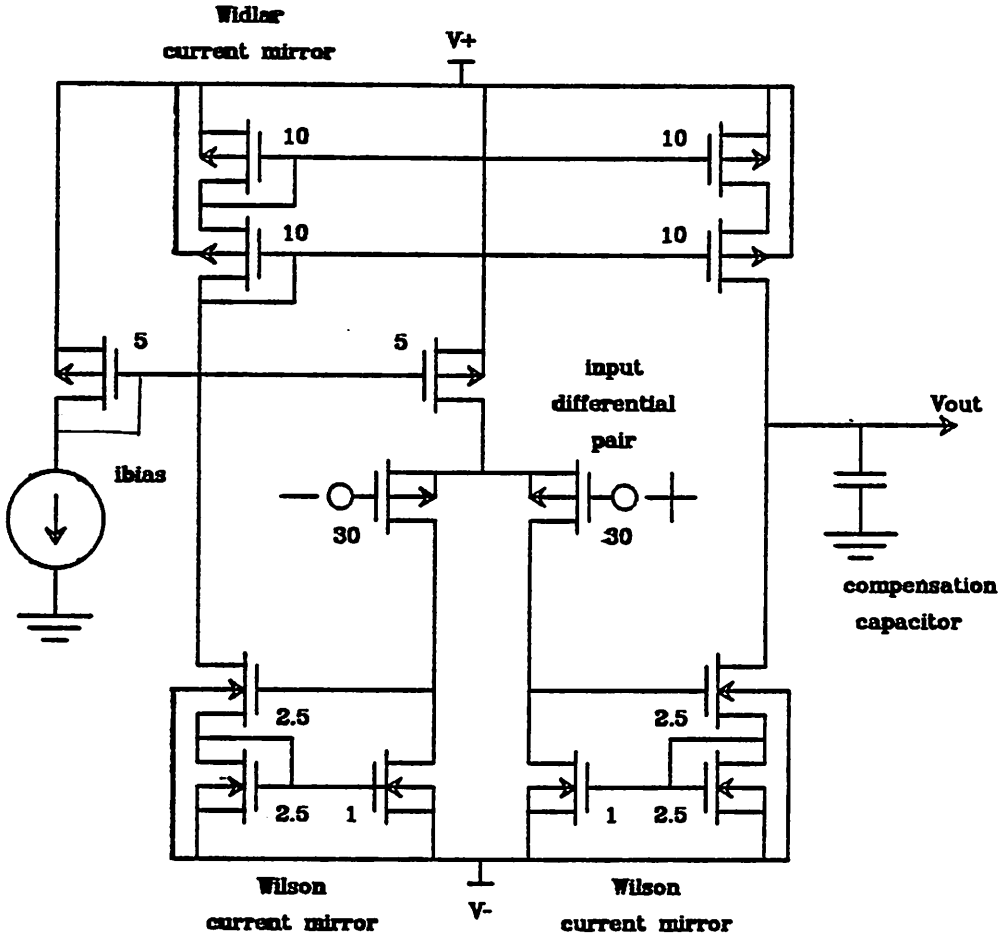


Figure 4.9. Circuit schematic of transconductance op-amp

loading capacity.

Current mirror Q_6-Q_8 reflects the drain current of Q_2 directly to the output. The other Wilson mirror reflects an equal amount of current to the op-amp output via a PMOS unity gain current mirror Q_9-Q_{12}. This is a cascaded Widlar current mirror. It provides source current at the output while the output Wilson mirror provides sink current. The op-amp therefore has a true push-pull operation that is completely balanced. There is ideally zero input offset voltage in this design since all bias current is equally balanced between source and sink current. Input offset voltage is further minimized by the use of the cascaded Widlar mirror since it is more accurate than a Wilson, although slower. As long as the two Wilson mirrors are matched, their gain errors cancel while their speed is used to advantage in their location in the circuit. The primary source of input offset voltage is in mismatches in the threshold voltages of the differential pair and its loads, Q_3 and Q_6. This source of offset is minimized by symmetric layout.

The gate of Q_2 is the non-inverting input of the op-amp. An increase in its gate voltage will reduce the current at the drain thereby reducing the sink current at the output. Simultaneously, the source current is increased since additional current is provided by Q_1. As a result, the op-amp output voltage increases. Conversely, the gate of Q_1 functions as the inverting input.

Transistors Q_{13} and Q_{14} form a simple Widlar current mirror to provide bias current. Although the op-amp was designed for a 100 μamp bias, the op-amp performance characteristics can be varied by changing this current. In particular, settling time and power dissipation versus gain can be varied by changing the operating bias. The op-amp will operate satisfactorily over a wide range of operating currents and voltages. Its symmetric design provides a high degree of power supply rejection.

4.5.2. Circuit Layout

A plot of the op-amp layout is included as figure 4.10. Common centroid design was used wherever possible. The input differential pairs were each made as two semi-circular transistor pairs. The transistors are symmetric with respect to a point in the center of each pair and to a point between Q_1 and Q_2. The rounded edges made with 45 degree lines help to increase breakdown voltage by reducing the electric field density at corners in the transistors. By wiring the source on the inside and the drain on the outside, breakdown voltage for the op-amp was further increased, since higher voltage drops appear at the drain. The input differential pair was placed in its own well to minimize body effect thereby increasing gain. The two Wilson mirrors were placed as close together as possible and in a symmetric fashion for optimal matching.

4.5.3. Compensation, Speed, and Gain

This op-amp design essentially has one gain stage since all current is mirrored to the output directly. The voltage gain can be calculated from the differential pair transconductance times the Wilson current gain times the output resistance. For this circuit as shown in figure 4.9, the voltage gain would be:

$$A_v = g_{m1,2}2.5[r_{o8}(1+g_{m7}r_{o7}) \,||\, r_{o12}(1+g_{m11}r_{o11})] \tag{4.6}$$

where g_m can be written as:

$$g_m = \sqrt{2\mu C_{ox}\frac{W}{L}I_D}\ .$$

Using typical process parameters, $A_v \approx 10{,}000$.

The circuit of figure 4.9 is inherently stable. There is only one dominant pole which is at the output node of the op-amp. The drains of the differential pair contribute much higher frequency poles because of the relatively low input impedance of the Wilson current mirrors. Another high order pole exists between

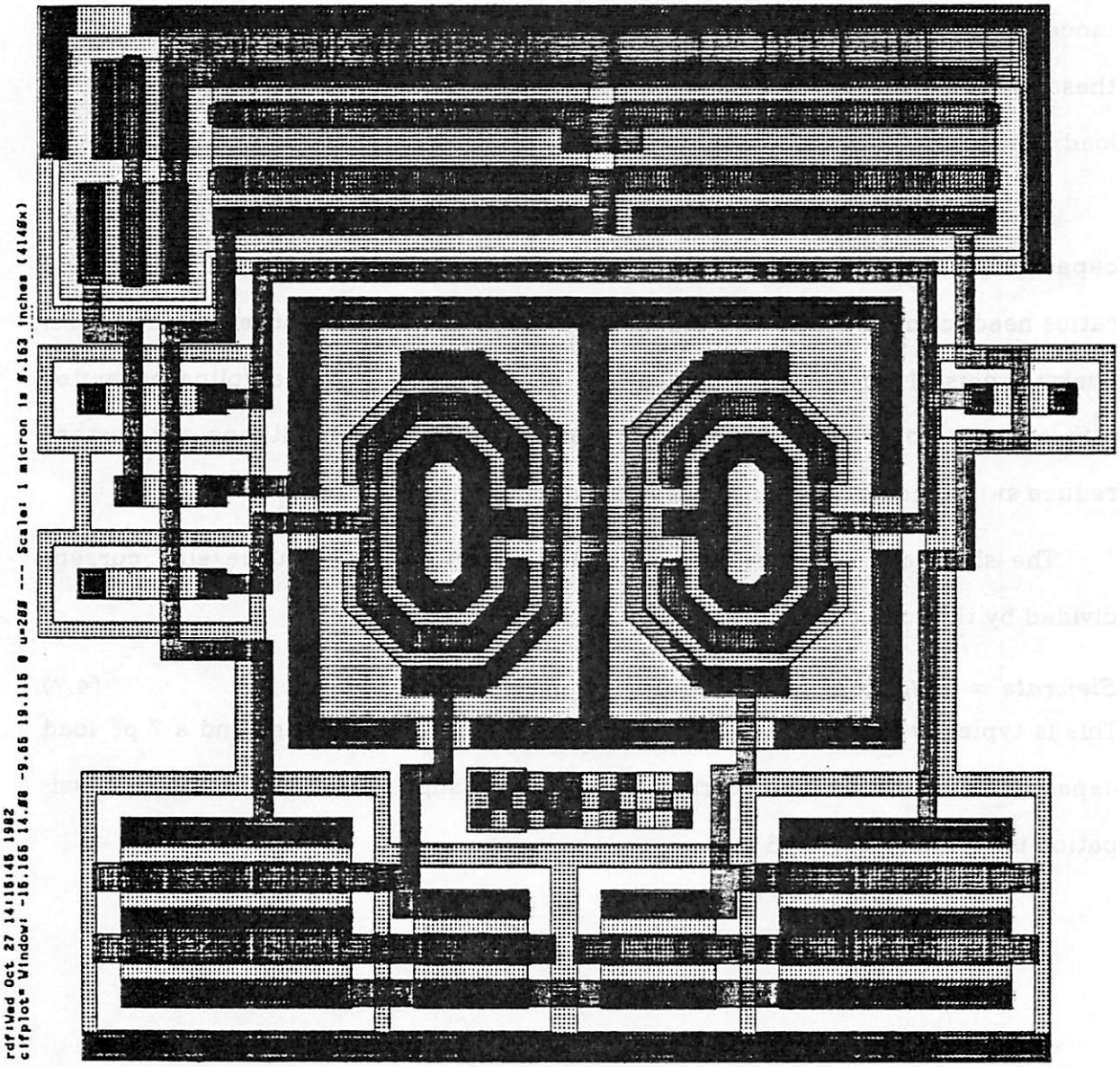


Figure 4.10. Layout plot of op-amp, scale: 1μm is 0.022 inches (559x)

Q_5 and Q_9, the cascaded Widlar input. However, a fairly low impedance exists at this point also. To compensate the op-amp for these higher order poles, the op-amp must simply have a sufficient amount of load capacitance. This load capacitance must lower the dominant pole frequency at the output to where the gain of these second order poles is reduced below unity. From SPICE simulations, a 7 pF load capacitance was calculated.

This capacitance value was used as the basic size for input and feedback capacitors. This amount of capacitance is large enough to allow for the 8 bit ratios needed in the MDAC and still have the LSB about 12 dB above the parasitics. Thermal noise from the MOS switches is bandlimited by the sampling capacitor with a mean squared value of kT/C. Larger values of capacitance will further reduce switch noise but also reduce circuit speed.

The slew rate is determined by the maximum output source/sink current divided by the load capacitance:

$$Slewrate = 2.5 I_{bias} / C_{load} \quad . \tag{4.7}$$

This is typically equal to 34 Volts/μsec for a 100 μA bias current and a 7 pF load capacitance. For this bias current and a ±15volt supply, the op-amp power dissipation is typically: P_D = 18 mW.

CHAPTER 5

Experimental Results

An integrated circuit that implements the analog portion of the Lattice analysis filter was fabricated. The integrated circuit uses the circuitry and CMOS process described in chapter 4. The correlator described in chapter 3 was breadboarded with standard off-the-shelf digital ICs. The correlator and analog filter chip were fully functional and together formed a complete adaptive analysis filter for speech.

Various speech samples, both sentences and vowel sounds, were digitized and stored in a computer. These speech samples were played back as analog speech into the adaptive filter. The resulting reflection coefficients were simultaneously stored in the same computer. Log power spectra and resynthesized speech were produced by the computer from these reflection coefficients. This was then compared with computer simulations of the adaptive filter and with the original digitized speech. Several different sets of reflection coefficients were also fed to the analog filter chip without the correlator attached. The resulting spectra of the all-zero filter responses were compared with computer simulations in order to test the analog filter chip separately.

This chapter will describe in detail the results of these tests and the measured performance of each of the functional blocks if the integrated circuit filter.

5.1. Op-amp

The op-amp performed as expected, except for its speed. Because of problems with the mask aligner and adhesion of the positive photoresist to the alumi-

num, the alignment tolerance had to be increased from 1 μm to a very conservative 3 μm. The metal gate-electrodes had to be extended over both the source and drain regions by an additional 2 μm. This along with other process variations resulted in a two-fold increase in the gate overlap capacitance. The op-amp showed a corresponding decrease in speed. Earlier runs with the original 1 μm alignment tolerance had 0.1% settling times of approximately 700nsec. The final run produced a settling time which was more than twice as slow. This necessitated testing the filter chip at half the speed, a 4kHz sample rate. The earlier runs indicated that in a commercial production environment op-amps of the necessary speed could be produced to allow operation of the analysis filter chip at the full 8kHz sample rate. Measurements describing the functional performance of the op-amp are listed in table 5.1. These results are for the final run with the 3 μm alignment tolerance. Because this is a transconductance op-amp the output impedance is very high, on the order of 10 Megohms. The op-amp characteristics had to be measured using a standard bifet op-amp as a high impedance voltage follower. These measurements take into account the effects of this second op-amp.

5.2. Delay line

A breadboard of the delay line indicated a potential for serious noise and crosstalk problems. Mismatches in the hand-selected discrete capacitors and wiring capacitance produced a fixed pattern noise voltage of 15 mV. Breadboard parasitics contributed significant crosstalk between adjacent storage elements.

The integrated circuit version of the delay line exhibited little of the problems which appeared in the breadboard. Because of the high precision to which IC components can be matched, fixed pattern noise in the integrated circuit was significantly reduced.

Measured Op-amp Characteristics	
Gain	8,000 V/V
Unity Gain Bandwidth	4 MHz
Slew Rate	>30 V/μsec
Settling time (to 0.1%)	1.5 μsec (0.7 μsec*)
Maximum output swing (V_{supply} = ±7.5 volts)	±3 volts
Noise	100 nV/$\sqrt{Hz}$
Dynamic Range (4kHz BW)	80 dB
CMRR	60 dB
PSRR- V+	80 dB
V-	60 dB
Power Dissipation	10 mW
Number of transistors	14
Size	450 mil^2

Table 5.1. (*Measurement from runs with 1 μm alignment tolerance.)

Although this error was measured on 7 different chips with a mean peak value of 11mV, improved layout can possibly decrease this value to less than 2 mV. Of the eleven delay elements, the nine elements that are surrounded by other identical stages had a root-mean-squared mismatch error averaged over the 7 working chips of 1.4 mV. The peak mismatch occurred at the end of the delay line. This peak noise value can be reduced if coupling from adjacent clock lines at the ends are reduced with improved layout.

Figure 5.1 is an oscilloscope photograph of the delay line showing the fixed pattern noise. A small DC signal has been introduced to shift the output so that

switching transients would not obscure this error signal.

Charge injection from the switches produced a constant offset voltage of 100 mV. Large gate overlaps due to alignment problems in the IC fabrication amplified this error. The offset error can be reduced by a factor of 5 with a self-aligned silicon gate process.

Depletion leakage currents were negligible at the clock frequencies used for the multiplexed filter. Leakage became significant for switch off times longer than 500 μsec.

5.3. Analog analysis filter

A die photograph and layout plot of the analog analysis filter are shown in figures 5.2 and 5.3. The total area of the analog IC is 18,000mil^2. The integrated

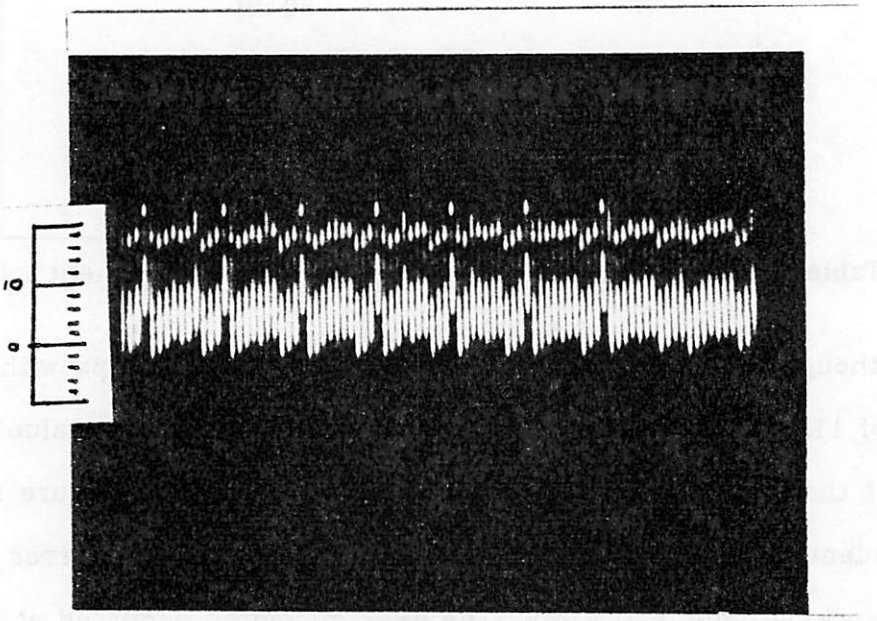


Figure 5.1. Fixed pattern noise in the delay line. The vertical scale is 10mV/div. and the horizontal scale is 10μsec/div. The center of the display is set to the op-amp offset voltage. The op-amps are reset to this voltage every fifth clock cycle.

circuit consumes $\approx 100mW$ of power with a ±7.5volt supply. (The lower supply voltage made interfacing easier).

The analog switched-capacitor integrated circuit was tested independent of the correlator as a programmable all-zero filter. The 10 zero locations were converted to reflection coefficients and fed to the MDACs on the chip. The filter response was sampled at the forward residual output of the filter after ten stages of multiplexed operation. A computer simulation of the response was also produced for comparison.

Noise and dynamic range were measured by setting all of the coefficients (k_0, k_1, $\cdots$, k_{10}) equal to zero to form an all-pass filter. The measured filter response and noise spectrum is shown in figure 5.4. From this spectrum analyzer photograph, the dynamic range is measured at over 45 dB. The major noise

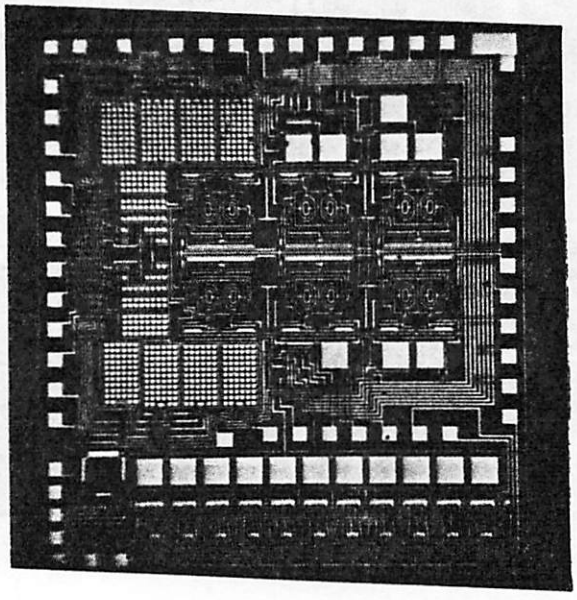


Figure 5.2. Die Photograph of CMOS switched capacitor filter IC

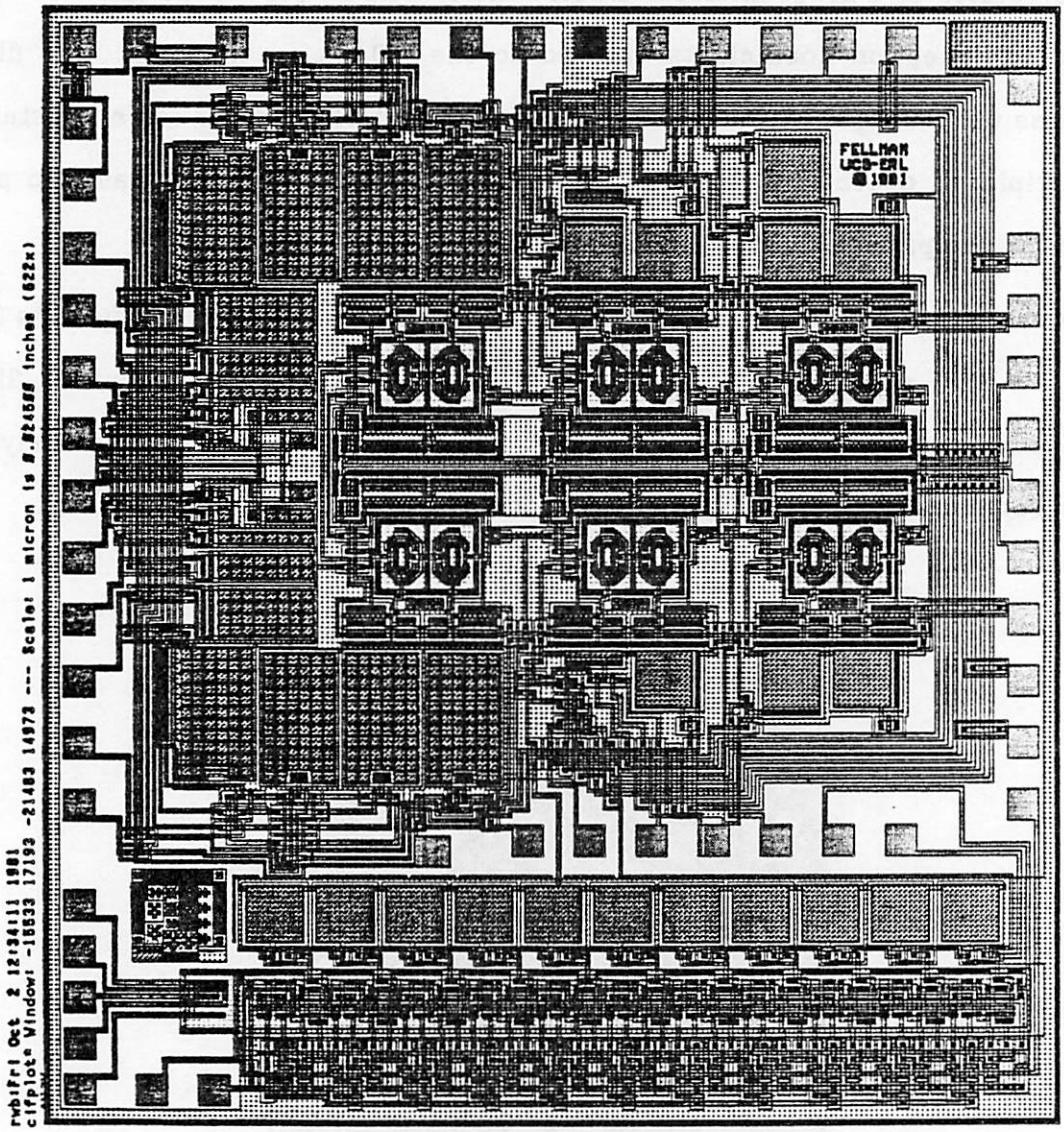


Figure 5.3. Layout Plot of Analog Analysis Filter IC

components that limit the dynamic range are at 36 Hz and harmonics of this frequency. This noise is produced by mismatches in the delay line as described in the last section. As was mentioned, with improved layout the delay line noise can be significantly reduced. With 2 mV of delay line mismatch, the dynamic range can be increased to over 60 dB. A fully differential configuration [22] would increase the dynamic range and reduce sensitivity to parasitics even further.

Graphs comparing the filter response to computer simulations for various sets of reflection coefficients are presented in figures 5.5 through 5.9. Figure 5.5 is the programmed spectral response derived from the vowel sound /ah/. Figures 5.6 and 5.8 have single and multiple notch filter responses which test the limits of the accuracy of the IC.

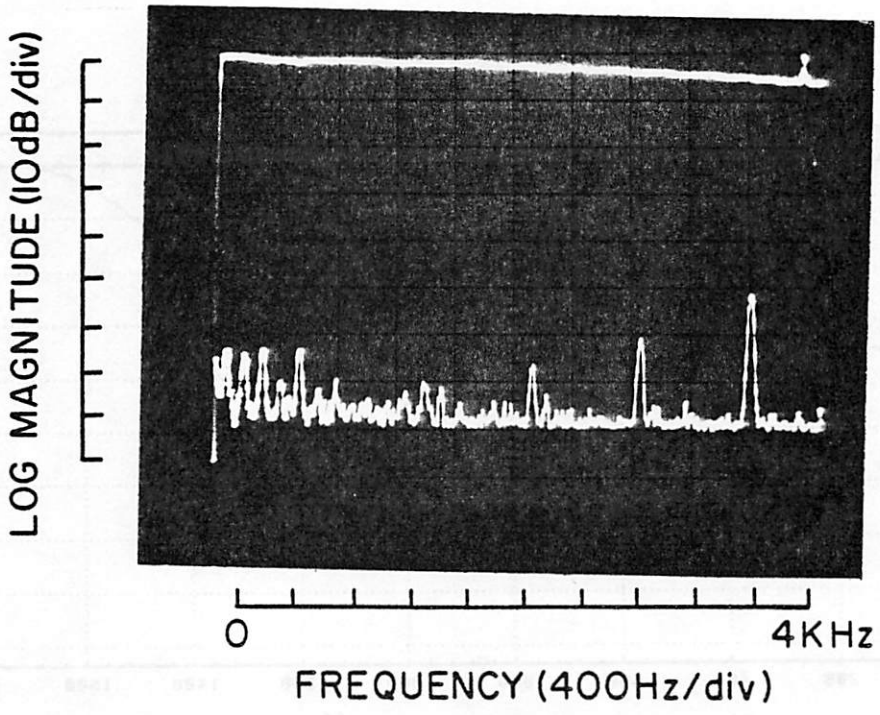
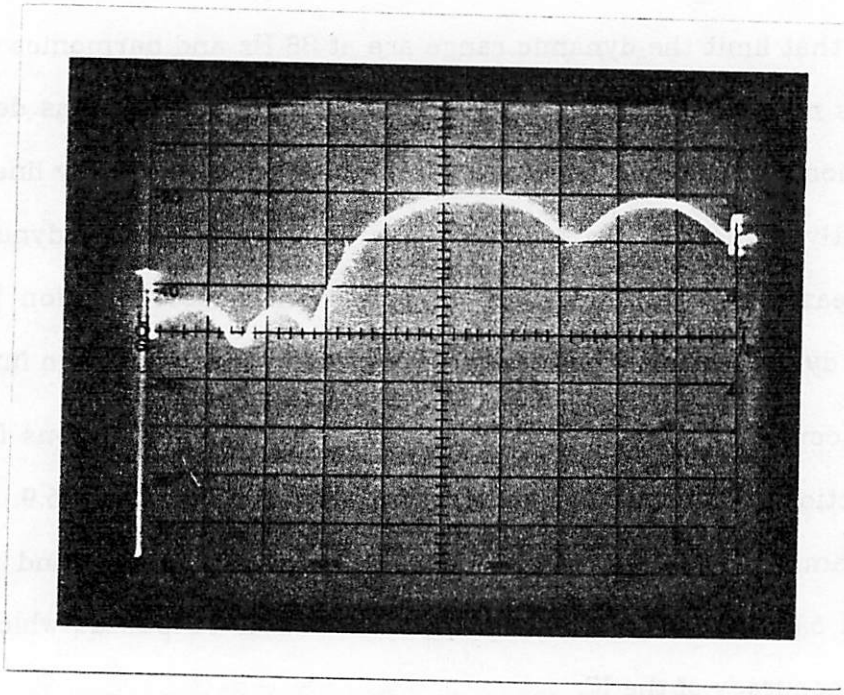


Figure 5.4. All-pass filter response and noise spectrum for lattice filter IC



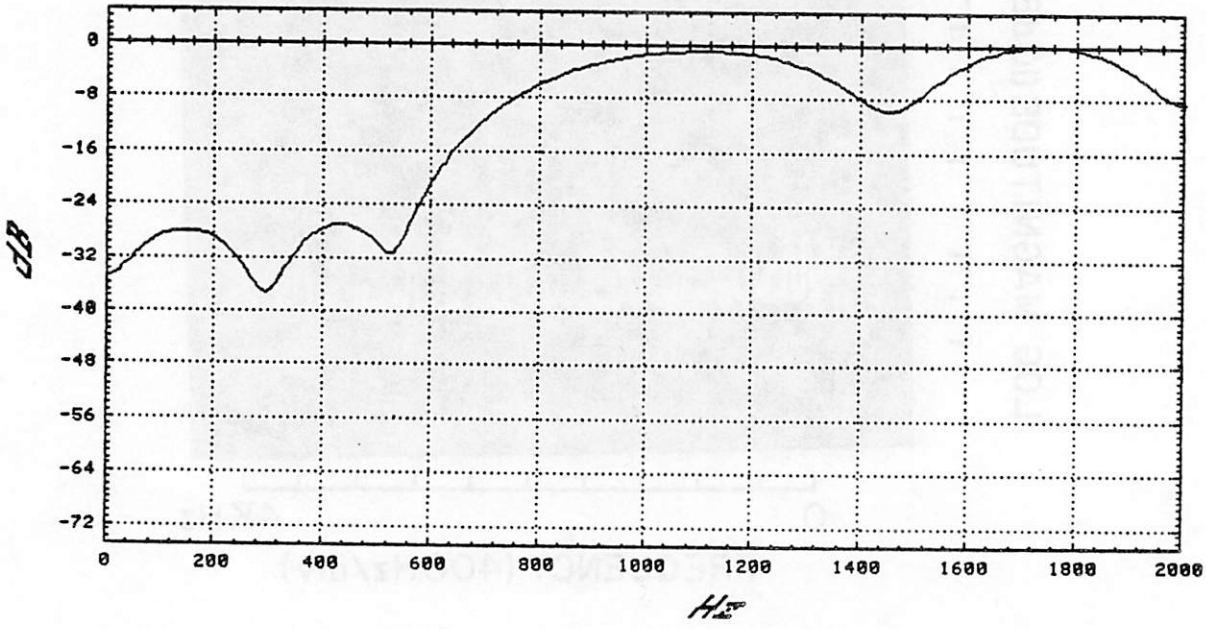
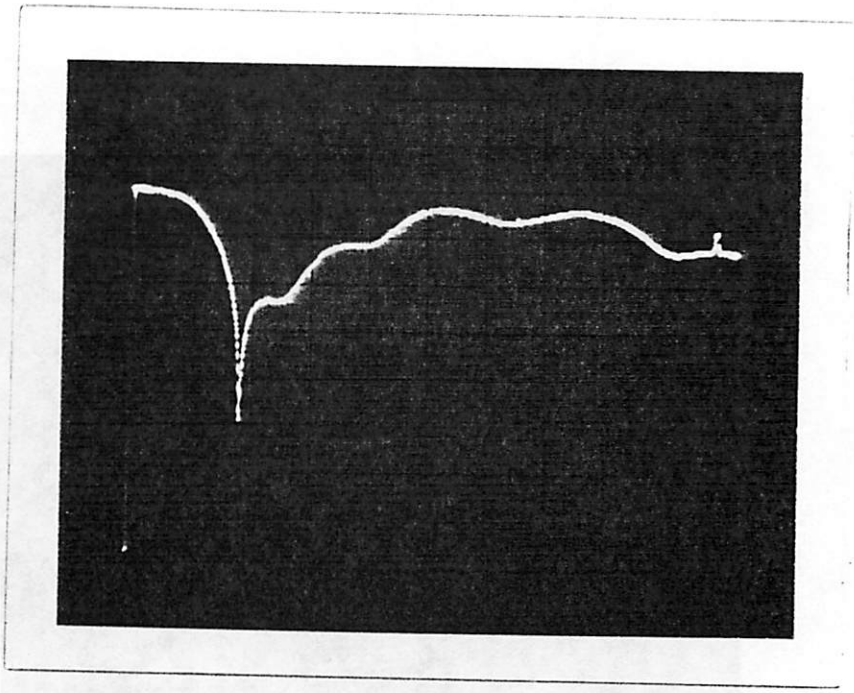


Figure 5.5. Lattice filter programmed response derived from LPC analysis of the vowel sound /ah/.
(a) IC data
(b) Computer simulation



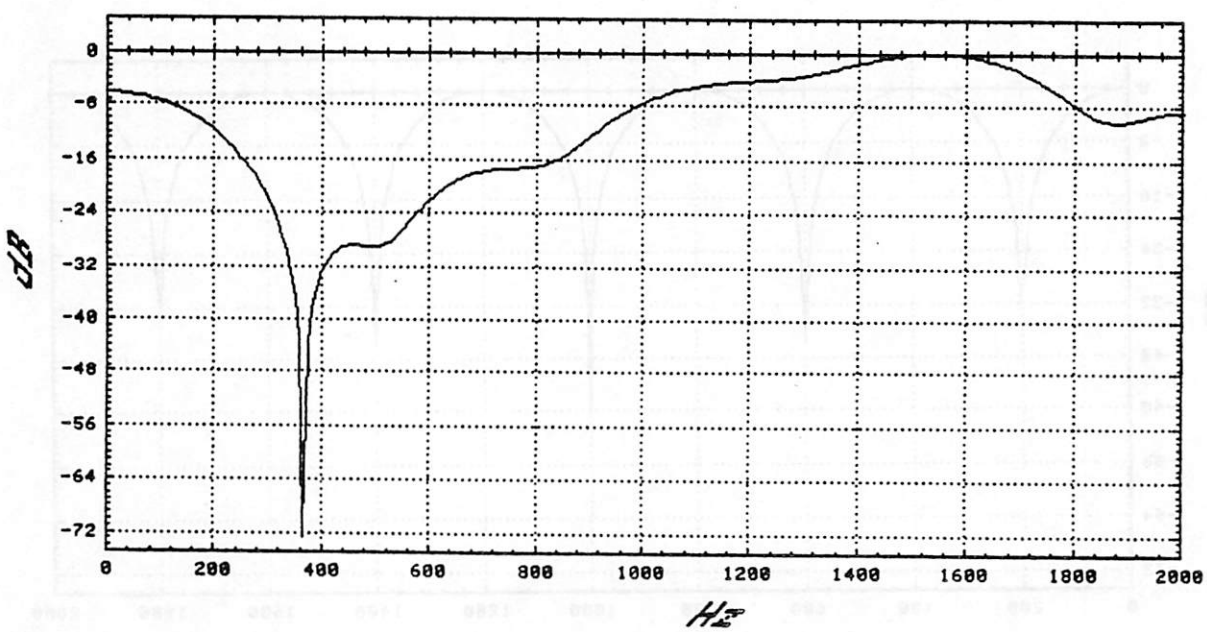
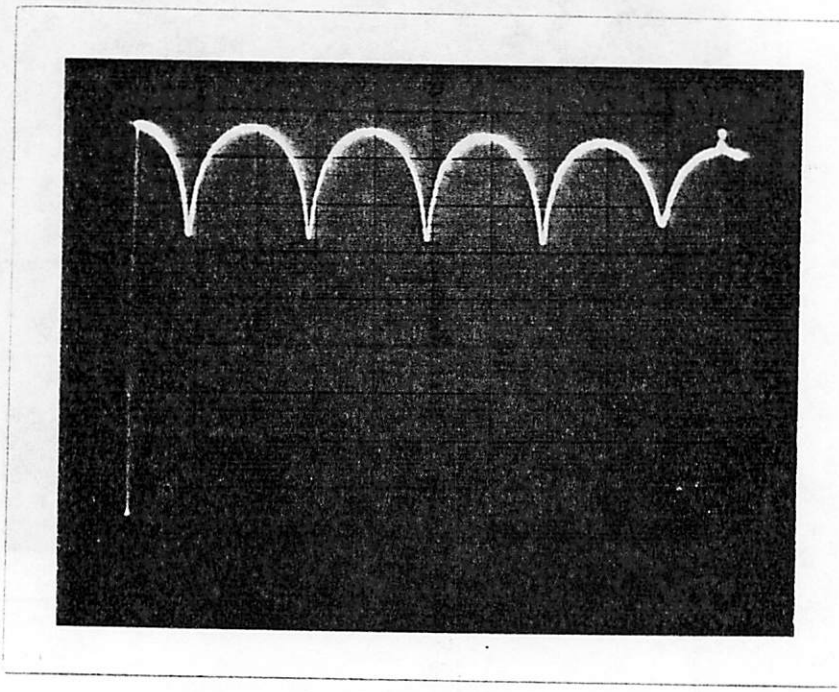


Figure 5.6. Lattice filter programmed response for a 10-zero notch filter derived from LPC analysis of a 720Hz sinewave.
(a) IC data
(b) Computer simulation



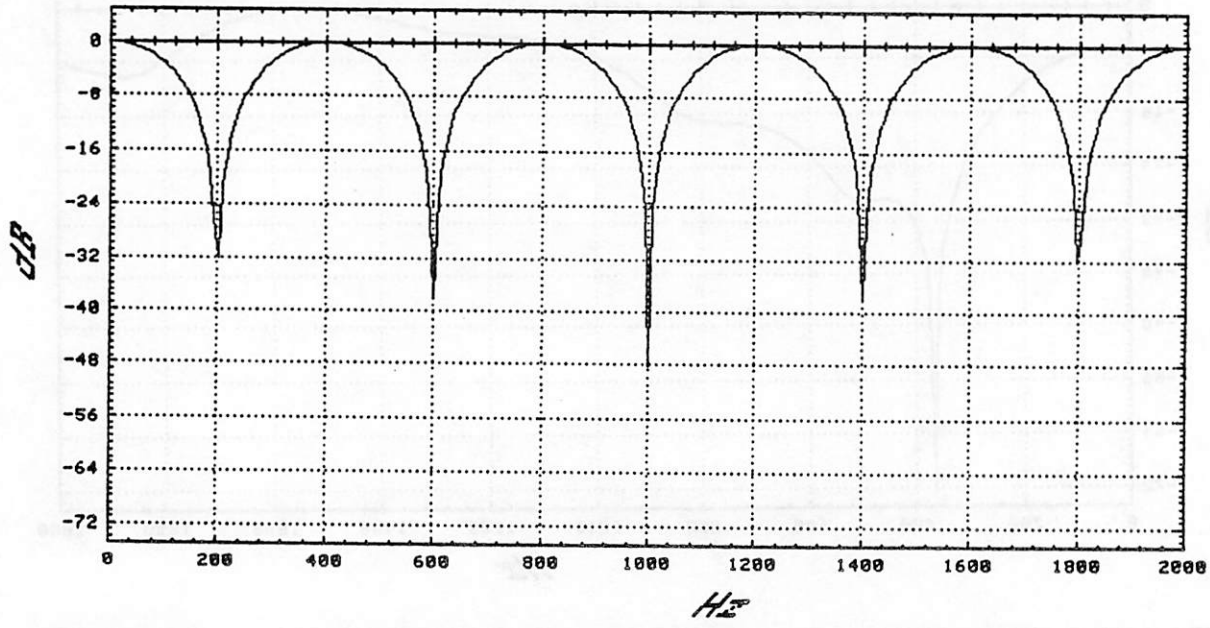


Figure 5.8. Lattice filter programmed response for a multiple 2-zero notch filter.
(a) IC data
(b) Computer simulation

5.4. Correlator

The correlator was constructed using a 7 bit (excluding sign bit) Companding A/D converter, a 24 bit expectation function, a 10 bit floating point division, and an 8 bit reflection coefficient output. Although the hardware did not use saturating adders for the expectation, overflow was generally not a problem. Some spectral distortion was introduced in the division and conversion from area ratios to linear reflection coefficients but this error was not significant and could not be detected in listening tests. In figures 5.11 through 5.15 are computer simulations showing the extent of this error for several different vowel sounds. In these Log spectra plots, distortion of the formant peaks are most noticed by the human ear. Distortion in the valleys between peaks are imperceptible.

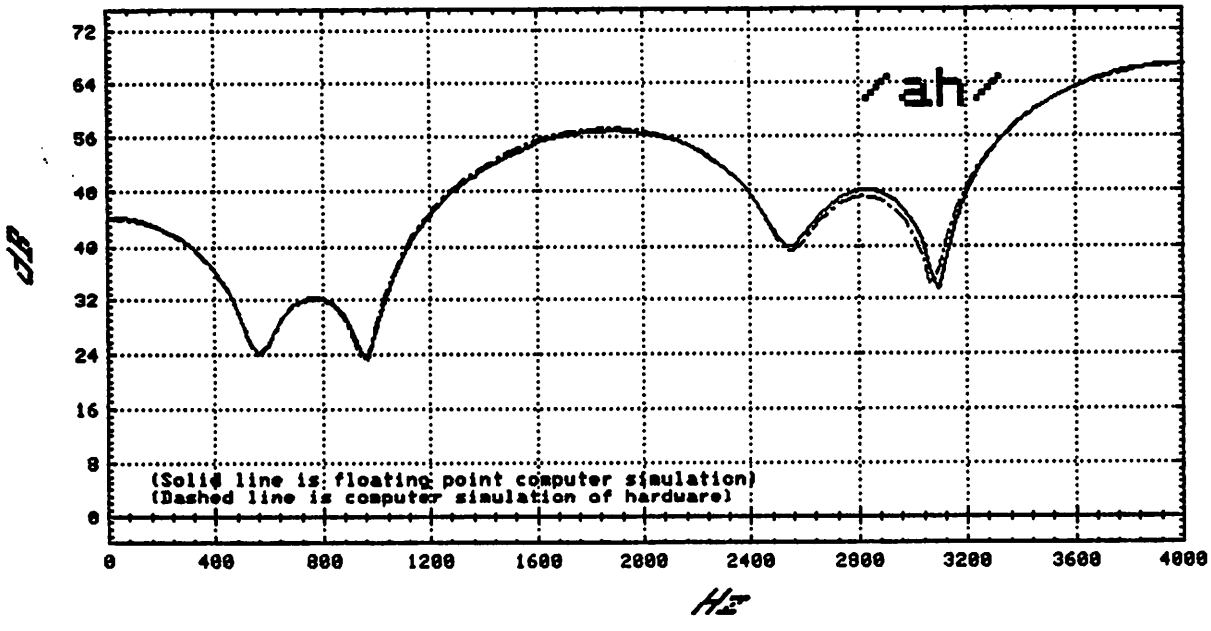


Figure 5.11. Simulation of Correlator Distortion for Vowel /ah/
(Solid line is actual smoothed vowel spectrum, dashed line is filter spectrum with correlator distortion.)

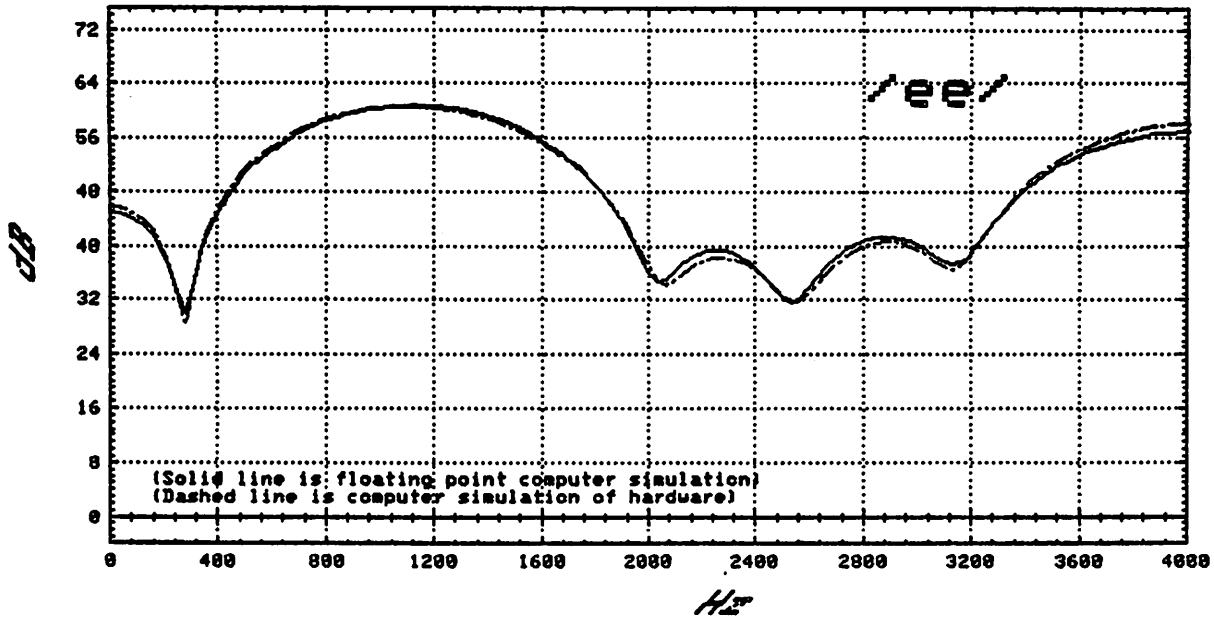


Figure 5.12. Simulation of Correlator Distortion for Vowel /ee/
(Solid line is actual smoothed vowel spectrum, dashed line is filter spectrum with correlator distortion.)

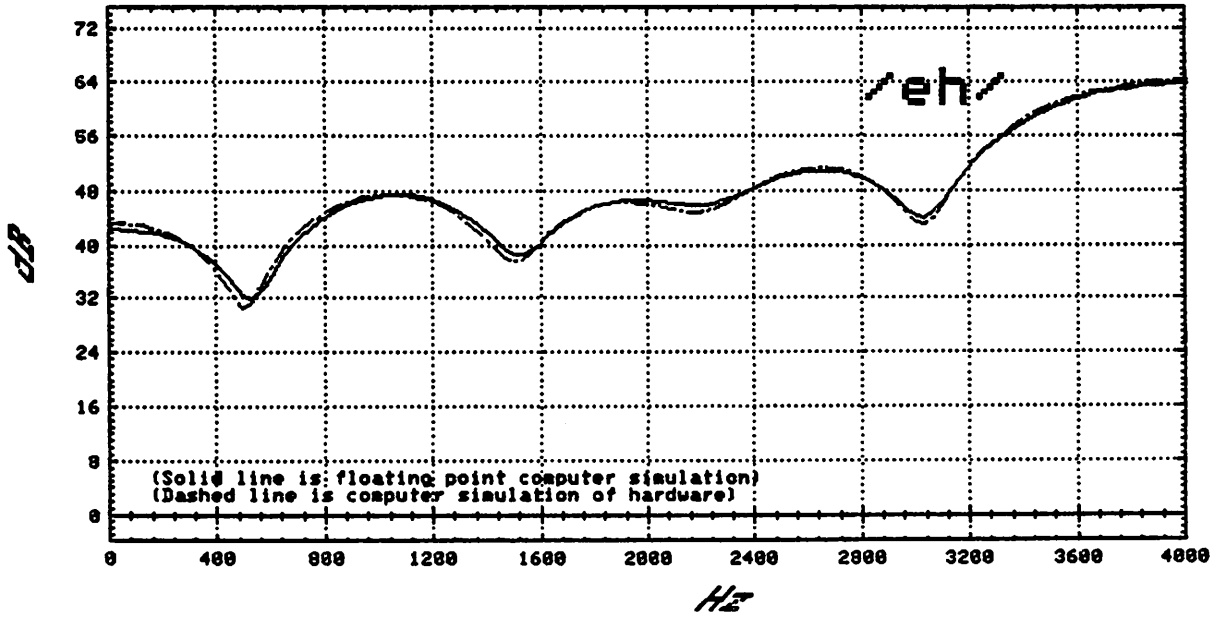


Figure 5.13. Simulation of Correlator Distortion for Vowel /eh/
(Solid line is actual smoothed vowel spectrum, dashed line is filter spectrum with correlator distortion.)

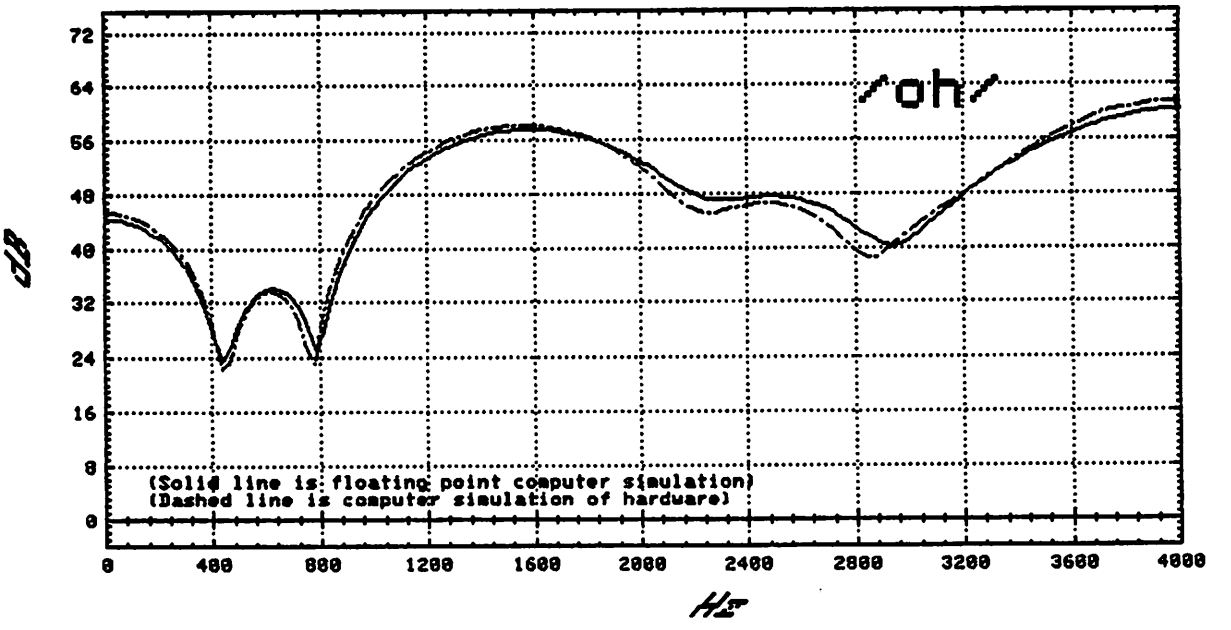


Figure 5.14. Simulation of Correlator Distortion for Vowel /oh/
(Solid line is actual smoothed vowel spectrum, dashed line is filter spectrum with correlator distortion.)

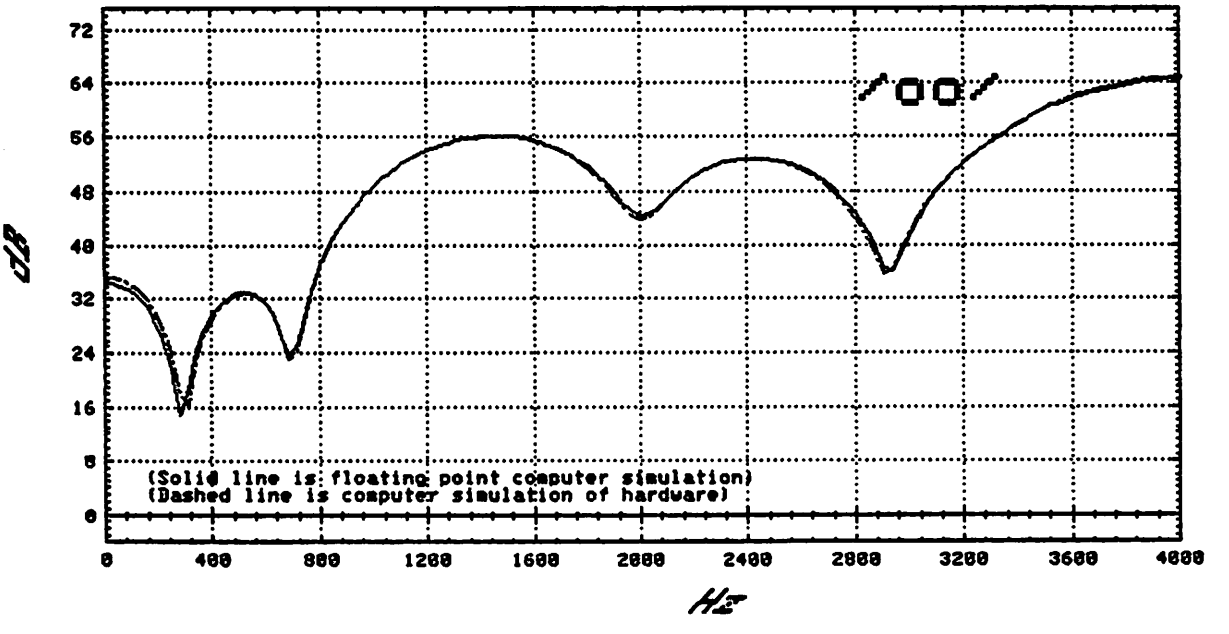
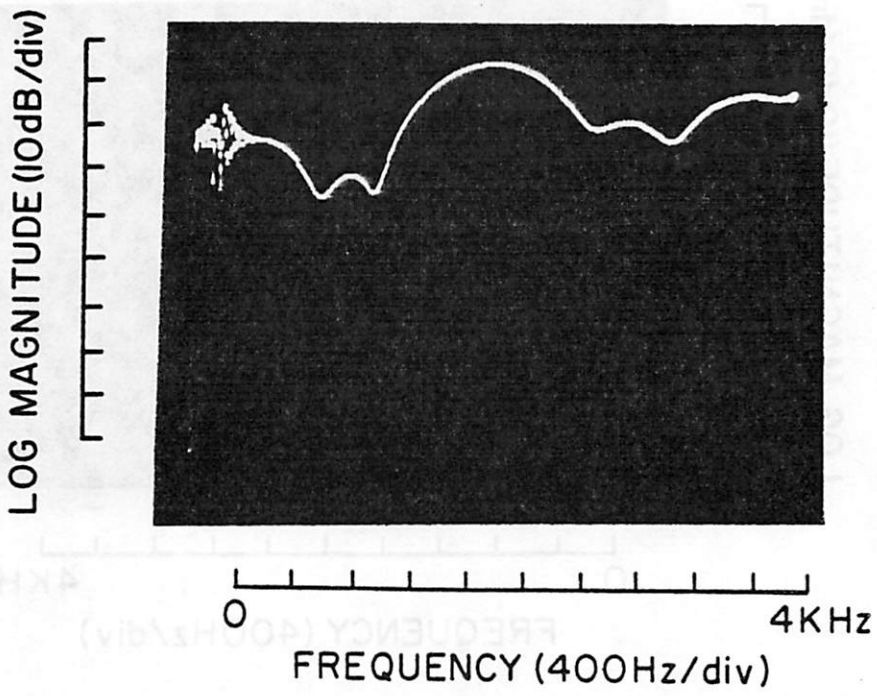


Figure 5.15. Simulation of Correlator Distortion for Vowel /oo/
(Solid line is actual smoothed vowel spectrum, dashed line is filter spectrum with correlator distortion.)

5.5. Adaptive filter

The correlator and analysis filter chip were connected together to form an adaptive filter. Analog speech sounds were output to the adaptive filter while the reflection coefficients were simultaneously sampled. The reflection coefficients produced by the filter were compared to computer simulations of the hardware by computing the spectral responses for these two sets of reflection coefficients. Comparisons were also made by resynthesizing the reflection coefficients into speech using constant voiced excitation so that listening tests could be performed.

Graphs comparing the spectral responses of the hardware and the computer simulations are presented in figures 5.16 through 5.20 for five different vowel sounds, /eh/, /ee/, /ah/, /oh/, and /oo/. Figure 21 compares the adaptation of the filter at different amplitudes.



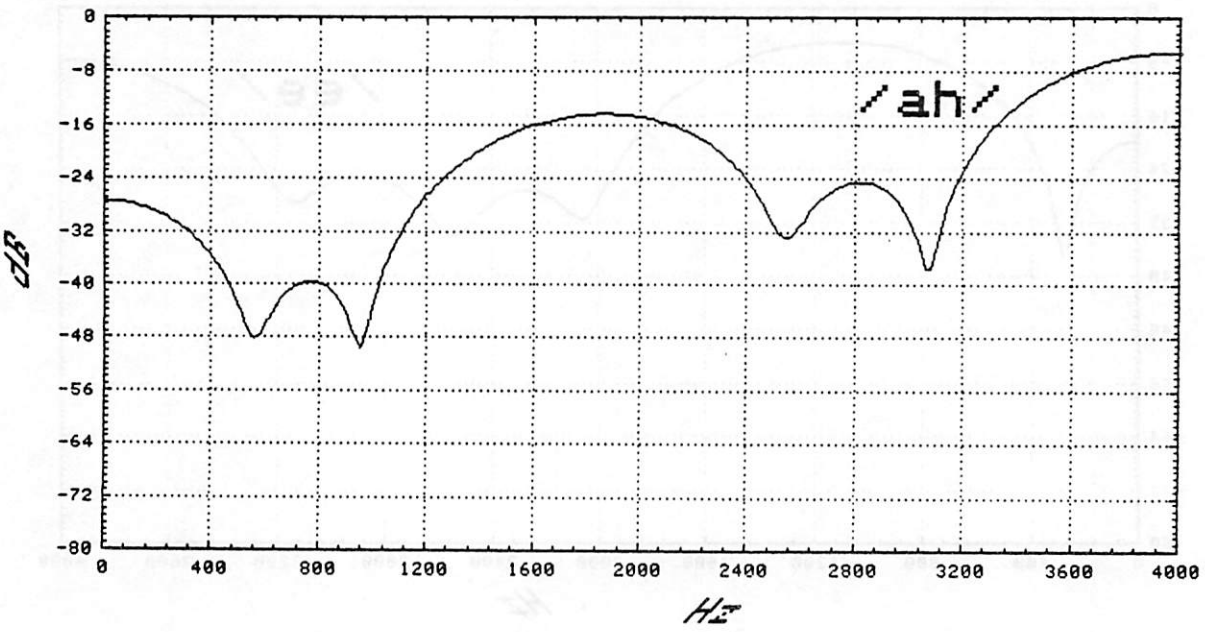
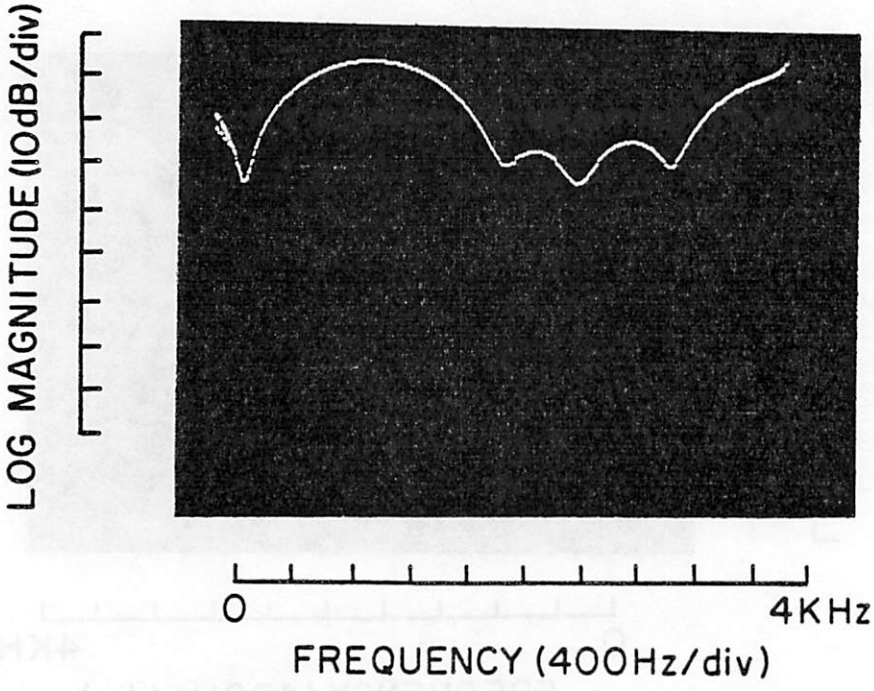


Figure 5.16. Adaptive Lattice Filter response adapting to vowel sound /ah/.
(a) Adaptive filter hardware data
(b) Computer simulation



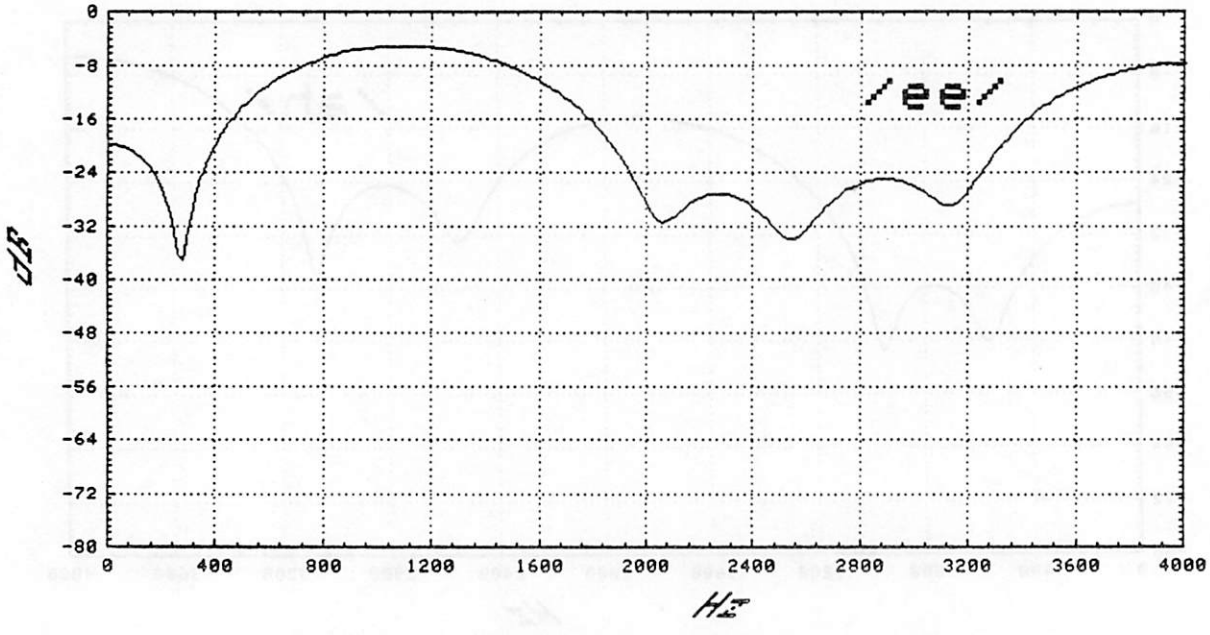
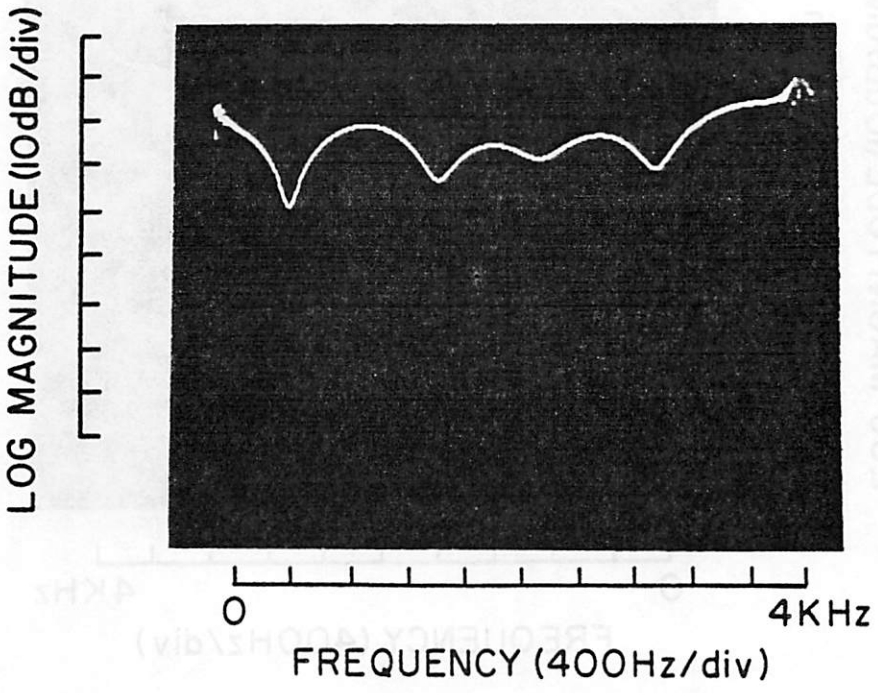


Figure 5.17. Adaptive Lattice Filter response adapting to vowel sound /ee/.
(a) Adaptive filter hardware data
(b) Computer simulation



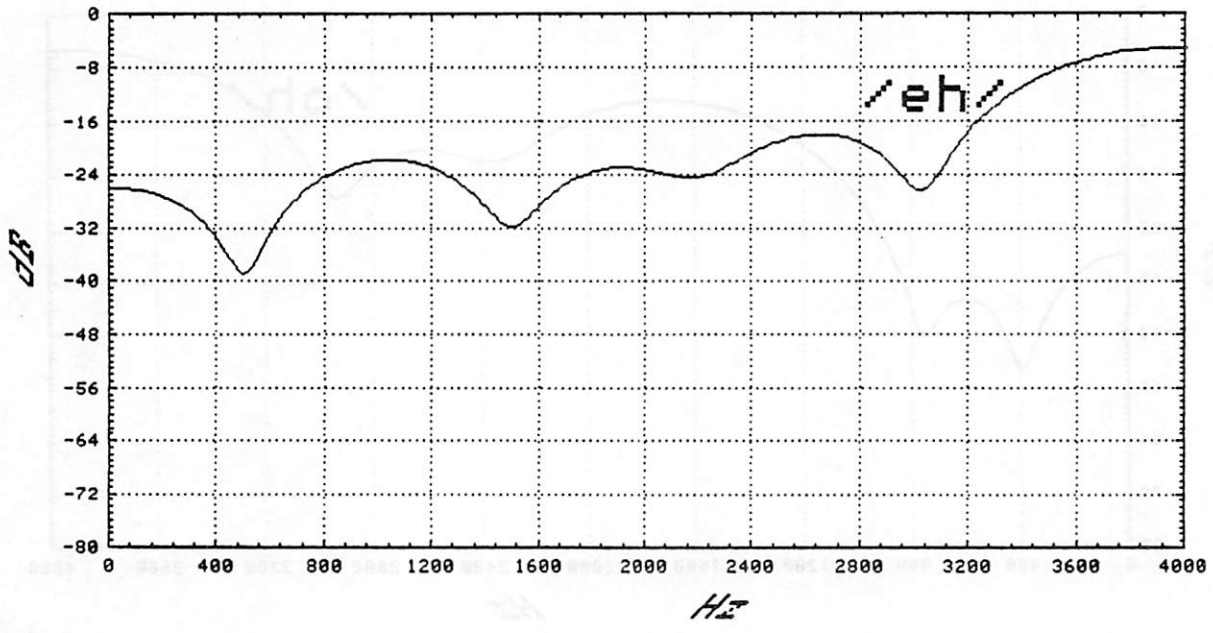
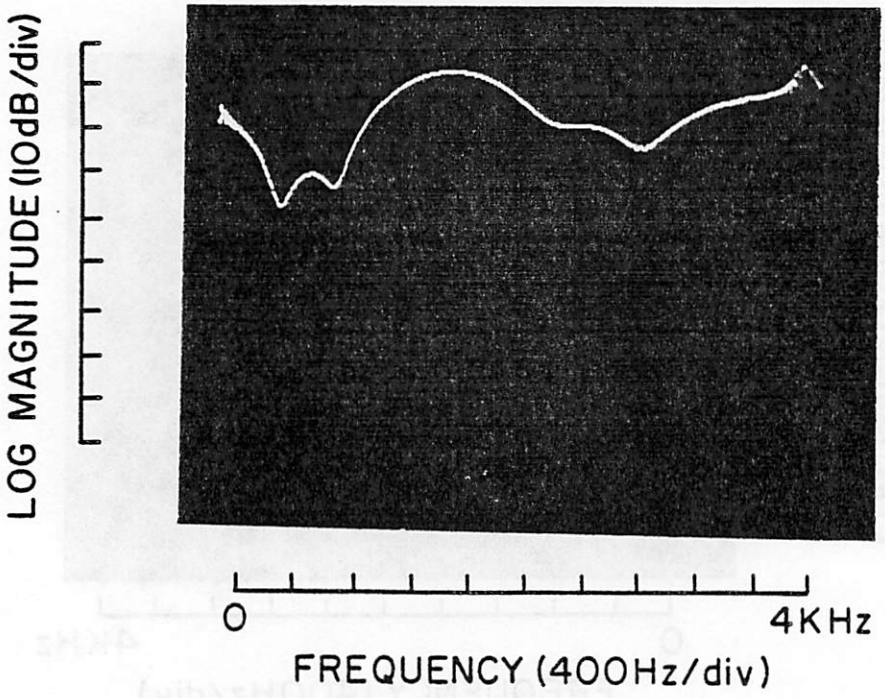


Figure 5.18. Adaptive Lattice Filter response adapting to vowel sound /eh/.
(a) Adaptive filter hardware data
(b) Computer simulation



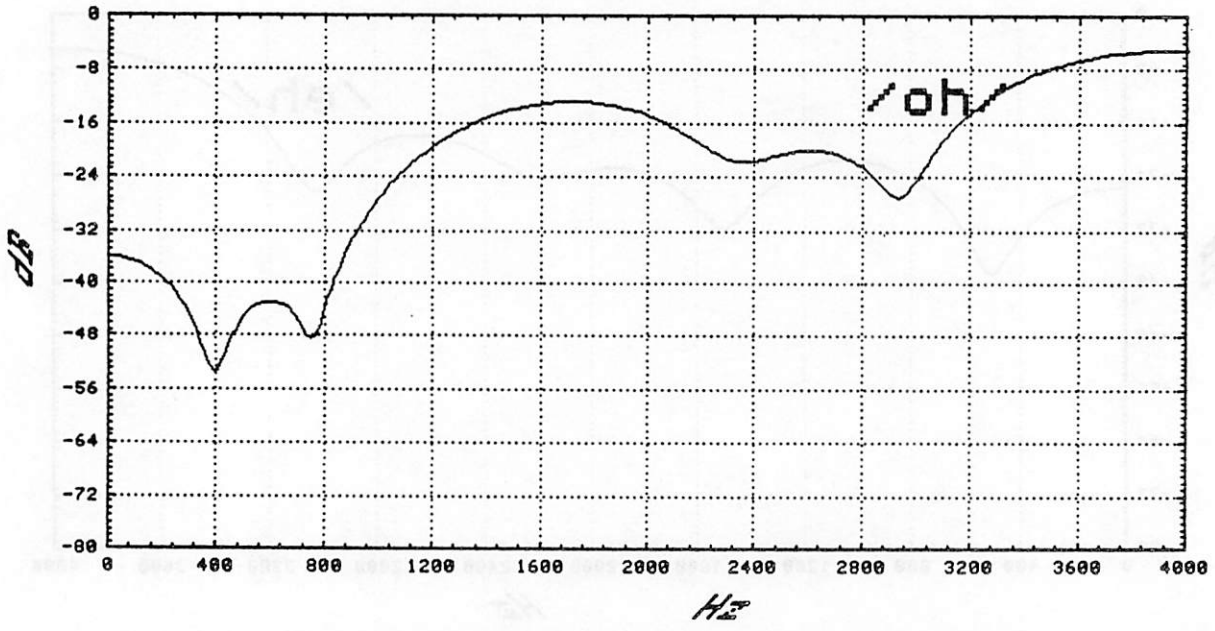
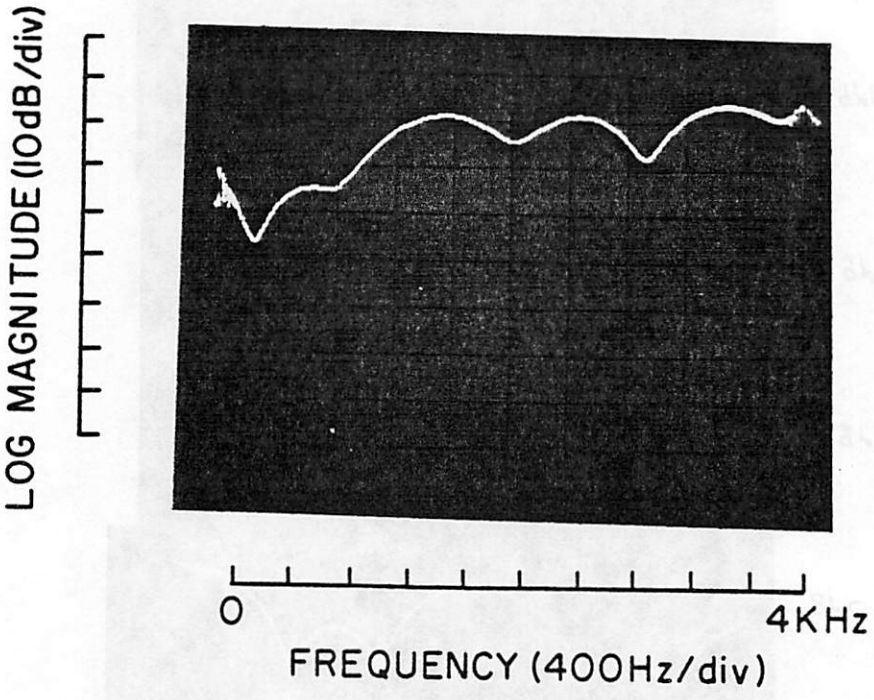


Figure 5.19. Adaptive Lattice Filter response adapting to vowel sound /oh/.
(a) Adaptive filter hardware data
(b) Computer simulation



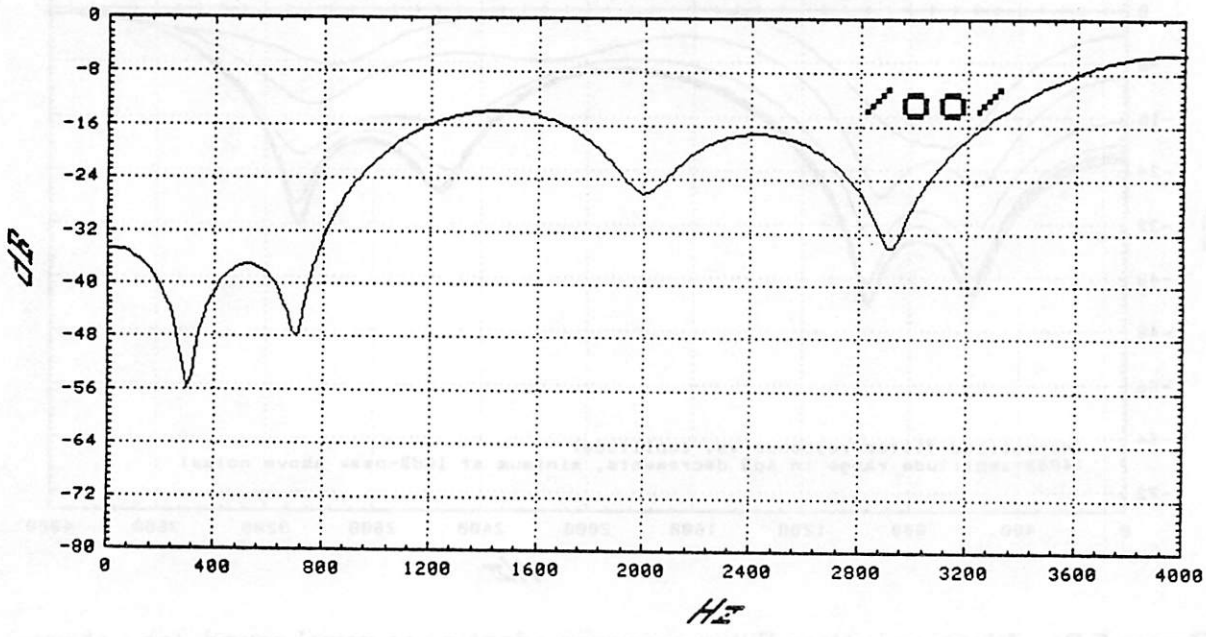


Figure 5.20. Adaptive Lattice Filter response adapting to vowel sound /oo/.
(a) Adaptive filter hardware data
(b) Computer simulation

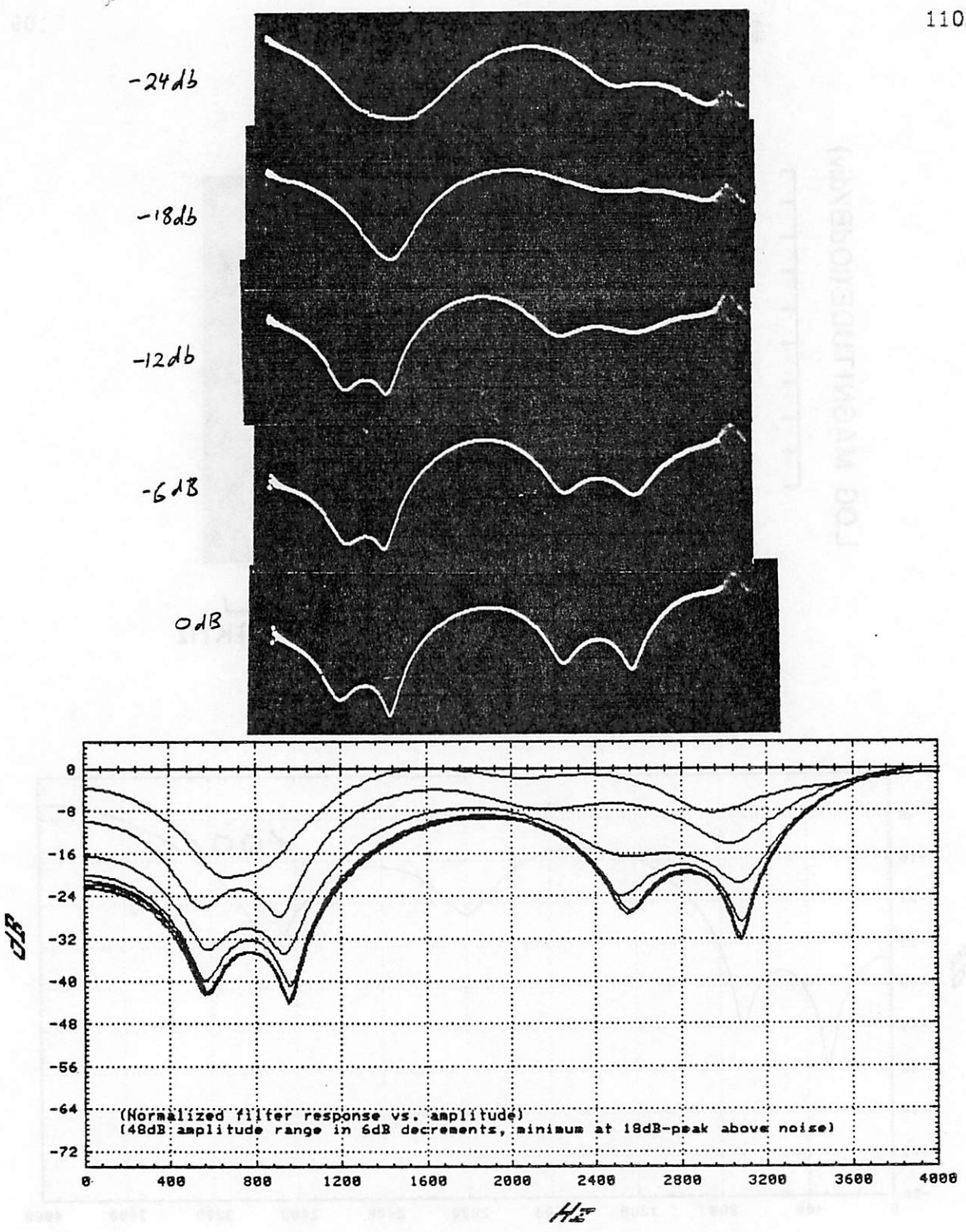


Figure 5.21. Adaptive Lattice Filter response adapting to vowel sound /ah/ at amplitudes decreasing by 6dB steps.
(a) Adaptive filter hardware data
(b) Computer simulation

5.6. Comparison of Analog and Digital Filter Implementations

Although only digital implementations of the correlator were discussed, both analog and digital implementations of the lattice analysis filter were presented. The lattice analysis filter was fabricated as an analog CMOS IC in order to determine the effect and magnitude of the various IC parasitics. Since digital circuitry is much less affected by these parasitics, only the functionality of the digital approach was tested using a breadboard. However, as part of the on-going research, the digital version was laid out and simulated for speed and power consumption. These parameters were computed based upon a 2.5 μm silicon-gate NMOS process.

Functionally, both the analog and digital implementations adequately computed the lattice algorithm and were in close agreement with computer simulations. Besides being immune to many of the parasitics that affect an analog approach, quantization error in the digital implementation did not present a significant problem. Using 16 bit arithmetic, the digital filter had comparable dynamic range to the analog chip. Computation speed was also comparable if allowance is made for metal-gate versus silicon-gate technologies.

There were several drawbacks to the digital approach. Although the silicon-gate NMOS layout required $14{,}900 mil^2$ compared to $18{,}000 mil^2$ for the high voltage metal-gate CMOS design, the analog layout could be reduced by more than a factor of two using a silicon-gate process. For the digital layout, power dissipation was computed to be $\approx 350 mW$ compared with $\approx 100 mW$ for the analog chip. The digital approach also resulted in a more complicated circuit design. In a switched capacitor design, an addition/subtraction, multiplication by a constant, and storage function requires only two clock pulses and the following components: 3 switches, 3 capacitors, and 1 op-amp. The same function implemented

using digital logic would require 10 clock cycles and 13 logic gates.

The analog approach had several disadvantages also. Layout was far more complex because parasitics had to be taken into consideration. Circuit design had to be modified to reduce sensitivities to certain parasitics, as was discussed in chapter 4. The external μ-law A/D converter must also be considered. Both approaches require an A/D conversion at some point. The μ-law converter requires only 7 bits (if an absolute value circuit is used) compared to the 12 bit converter used in the digital implementation. However, the μ-law converter is required to operate at a ten times higher rate.

In balance, both approaches can produce acceptable monolithic implementations of an adaptive lattice filter for speech processing having comparable performance. Because of the high degree of automation of digital design and the availability of reliable digital processes, an all-digital approach using specialized circuitry would probably yield the most efficient implementation of a monolithic adaptive filter for speech processing. In particular, circuit design and IC layout are more amenable to automation when a digital approach is used.

APPENDIX A

μ255-Law Compensation Circuitry

The logarithmic μ255-law, defined by the following equation, is usually implemented by a piece-wise linear approximation.

$$F(x) = Sign(x)\frac{ln(1+\mu|x|)}{ln(1+\mu)}$$

The actual decode function is given by equation A.1 [35]. This is slightly different from the encode function where a 1/2 LSB bias is used to reduce noise.

$$x = 2^C(S+16.5)-16.5 \qquad (A.1)$$

Here C is the chord value and S is the value of the step bits.

For the case where C is large, equation A.1 closely resembles a floating point or scientific notation with C as the exponent and $(S+16.5)$ as the mantissa. This approximation can introduce a significant error when C is small and the term (-16.5) cannot be neglected. Rewritting this as equation A.2 suggests a technique for correcting this error.

$$x = 2^C[S+16.5-2^C16.5] \qquad (A.2)$$

By making the mantissa a function of the exponent, equation A.2 can be expressed in floating point notation without error. A simple circuit for computing the mantissa: $[S+16.5-2^C16.5]$ is shown in figure A.1. A three to eight line binary decoder divides the factor 16.5 by powers of two and a 6 bit adder performs a twos-complement subtraction to complete the calculation.

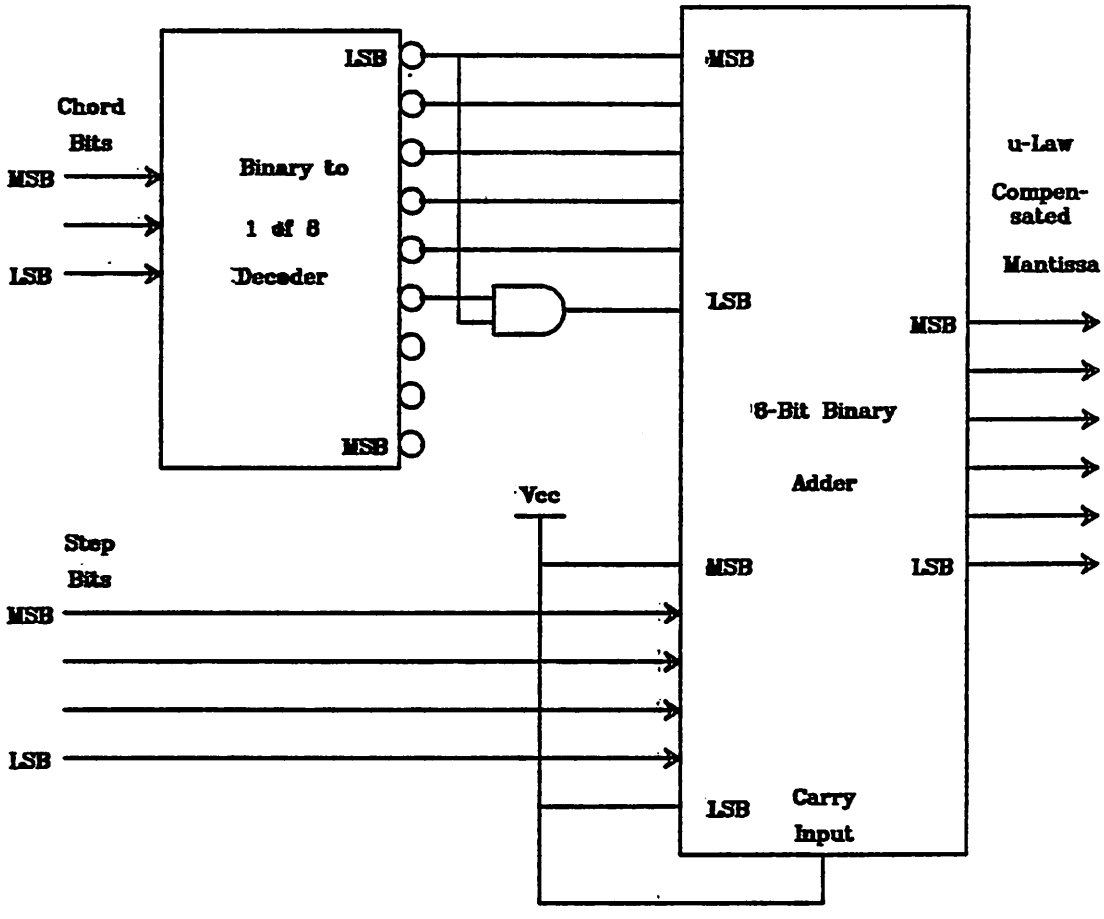


Figure A.1. μ255-law to floating point conversion circuit

APPENDIX B

Computer Simulation Programs

All programs were written in the programming language C for Vax750/780 computer systems. The computer systems ran under the Unix operating system. The LPC simulation programs closely model the actual implementation of the analog and digital portions of the hybrid Analog/Digital filter. Operations are done in floating point arithmetic to simulate analog circuitry. Appropriate word length arithmetic is used to model the digital portions of the algorithm.

B.1. Lattice LPC Analysis Program

Below is the analysis filter program. It takes raw or pre-emphasized speech and produces two files. One file has the 8-bit reflection coefficients that form a ten pole model of the speech. The ten reflection coefficients are outputted every 100 samples. This is equivalent to a 12.5 msec frame length. The other file contains the residual error for the ten stage filter. There is one output word for each input sample.

```
/*
                Lattice LPC Analysis Program

                Hardware Simulation:
                        delayed reflection coefficients
                        u255-law companding A/D
                        floating point analog
                        average reflection coefficients per frame
*/
#include <stdio.h>
#define BUFSZE 259
#define STAGES 10
#define SCALE 128
#define MINNUM .0001
```

```
#define FRAMERATE 100
#define LAST 10
float ef[STAGES+1],eb[STAGES+1],ebd[STAGES];
int c1[STAGES],c2[STAGES];
FILE *fp,*fpr,*fpk;
main(argc,argv)
  int argc;
  char *argv[];
{
short ksh,kavg[STAGES],i,j,res;
int eod,ca,c[STAGES+1];
float x;
if (argc<2){
        fprintf(stderr,"USAGE: lpcdelay file {amount_of_data} {skip}\n");
        exit(1);}
fp=stdin;
if (argv[1][0]!='-')
        if ((fp=fopen(argv[1],"r"))==0){
                fprintf(stderr,"lpcdelay: can't open %s\n",argv[1]);
                exit(1);};
if ((fpr=fopen("lpc.r","w"))==0){
        fprintf(stderr,"lpcdelay: can't open lpc.r\n");
        exit(1);}
if ((fpk=fopen("lpc.k","w"))==0){
        fprintf(stderr,"lpcdelay: can't open lpc.k\n");
        exit(1);}
if (argc==4)
        fseek(fp,2L*atoi(argv[3]),0);
if (argc>2)
        eod=0-(atoi(argv[2])+1);
else eod=0;
while(++eod) {
        ebd[0]=ef[0];
        for(j=1;j<STAGES;j++)
                ebd[j]=eb[j];
        ef[0]=getsh(fp);
        if (feof(fp)) {
                fflush(stdout);
                fprintf(stderr,"EOF encountered: %d numbers read\n",eod-1);
                exit(0);}
        /* Analysis Filter Loop */
        for(j=1;j<=STAGES;j++){
                ef[j]=ef[j-1]-(c[j]*ebd[j-1])/SCALE;/* Forward residual */
                eb[j]=ebd[j-1]-(c[j]*ef[j-1])/SCALE;/* Backward residual*/
                c[j]=k(j);
                /* Average reflection coefficients within each frame */
                kavg[j-1]=kavg[j-1]+c[j];
                if((eod+1)%FRAMERATE==0){
                                ksh=kavg[j-1];
                                kavg[j-1]=0;
                                putsh(ksh,fpk);}
                if(-eod<=LAST&&argc>2)
                        printf("k%d=%-6.4f ",j,-(x=c[j])/SCALE);}
```

```
        res=(*(ef+STAGES));
        putsh(res,fpr);
        if(-eod<=LAST&&argc>2)
                printf("res=%d\n",ca=(*(ef+STAGES)));}
exit(0);
}
/* Compute reflection coefficients */
k(n)
int n;
{
int a,b;
float z1,z2;
n=n-1;
a=ef[n]+ebd[n];
a=quant(a);
a=a*a;
b=ef[n]-ebd[n];
b=quant(b);
b=b*b;
c1[n]=(c1[n]-c1[n]/64+a);
c2[n]=(c2[n]-c2[n]/64+b);
z1=c2[n];
z2=c1[n];
if (z2==0) z2=MINNUM;
if (z1==0) z1=MINNUM;
return((1-z1/z2)*SCALE/(1+z1/z2));
}
/* u255-law simulation */
quant(x)
int x;
{
int y=0;
x=(x<0?-x:x)+16;
while(x>31){y+=1;x=x/2;}
return((x<<y)-16);}
```

B.2. Lattice LPC Synthesis Program

This is the program that resynthesizes the reflection coefficients into intelligible speech. It requires two input files, reflection coefficients and the input excitation, plus an optional power file. The reflection coefficients and power are read in at the 100 sample frame rate. One excitation sample is read for each output sample. The excitation can be either the actual residual from the analyzer or synthesized from pitch data. For most of the testing, the hardware and software

were compared using a constant pitch excitation with an 8 msec period. This greatly simplified the testing procedure. It also allowed comparisons to be made that were not influenced by pitch or voiced/unvoiced decision errors. file.

All of the arithmetic was done in floating point to minimize the errors contributed by the speech resynthesis.

```
/*
                Lattice LPC Synthesis Program
*/
#include <stdio.h>
#define FRAMERATE 100
#define STAGES 10
#define PNORM 63.0
#define SCALE 128.0
main(argc,argv)
 int argc;
 char *argv[];
{
/* These are automatically init. to 0 by cc */
short i,j,k[STAGES],n,out,powerflag,rflag=0,temp;
float ef[STAGES+1],eb[STAGES+1],ebd[STAGES+1];
float maxpower=PNORM,P,power,sqrt();
FILE *fpr,*fpp,*fpk,*fpo,*fopen();
if((fpo=fopen("lpc.o","w"))==0){
        fprintf(stderr,"lpcsynth: can't create lpc.o\n");
        exit(1);}
if((fpr=fopen("lpc.r","r"))==0){
        fprintf(stderr,"lpcsynth: can't open lpc.r\n");
        exit(1);}
if((fpp=fopen("lpc.p","r"))==0)powerflag=0;
else powerflag=1;
if((fpk=fopen("lpc.k","r"))==0){
        fprintf(stderr,"lpcsynth: can't open lpc.k\n");
        exit(1);}
for (i=1;i<argc;i++) {
        if (argv[i][0]!='-') {
                fprintf(stderr,"lpcsynth: bad parameter\n\n");
                fprintf(stderr,"       valid options are:\n");
                fprintf(stderr,"       -r  when lpc.r is actual residual\n");
                fprintf(stderr,"       -n  to normalize power in lpc.p\n");
                exit(1);}
        switch(argv[i][1]) {
                case 'r' :
                        rflag=1;
                        break;
```

```
                case 'n' :
                        maxpower=1;
                        while(!feof(fpp)) {
                                if (maxpower<(temp=getsh(fpp))) maxpower=temp;}
                        rewind(fpp);
                        printf("Maximum power in lpc.p is %d\n",temp=maxpower);
                        fflush(stdout);
                        break;
                default :
                        fprintf(stderr,"lpcsynth: unknown option %s\n",argv[i]);
                        exit(1);}}
while(++j){
        ef[STAGES]=getsh(fpr);
        if(feof(fpr)) {
                fprintf(stderr,"EOF on lpc.r\n");
                exit(1);}
if ((j-1+rflag)%FRAMERATE==0){
        for(i=0;i<STAGES;i++){
                k[i]=getsh(fpk);
                if(feof(fpk)) {
                        fprintf(stderr,"EOF on lpc.k\n");
                        exit(1);}}
        for(i=0,P=1;i<STAGES;i++) P=P*(1.0-(k[i]/SCALE)*(k[i]/SCALE));
        P=sqrt(P);
        if (powerflag) {
                power=getsh(fpp)/maxpower;
                if(feof(fpp)) {
                        fprintf(stderr,"EOF on lpc.p\n");
                        exit(1);}}}
ef[STAGES]=rflag?ef[STAGES]:ef[STAGES]*P;
ef[STAGES]=powerflag?ef[STAGES]*power:ef[STAGES];

/* Lattice Loop */

for(n=STAGES;n>0;n--){
        ef[n-1]=ef[n]+k[n-1]*ebd[n-1]/SCALE;
        eb[n]=ebd[n-1]-k[n-1]*ef[n-1]/SCALE;}
out=eb[0]=ef[0];
for(i=0;i<STAGES;i++) ebd[i]=eb[i];
putsh(out,fpo);}}
```

B.3. Reflection Coefficient to LPC Parameter Conversion

This program will take any number of 8-bit reflection coefficients and produce linear prediction coefficients suitable for digital FFT input. This program was used to compute the Log spectrum of the analysis filter response. Since the LPC parameters are just the impulse response for the all-zero analysis filter, the

FFT of these coefficients provide the actual frequency response of the filter. Various reflection coefficients could be fed through this program in conjunction with an FFT program to simulate the ideal filter response of the analog chip alone (without the correlator attached). The program uses the Levinson-Durbin recursion of equation 2.3.

```
/*
                Reflection coefficient to LPC conversion
*/
#include <stdio.h>
#define MAXSTAGES 100
#define ASCALE 4095
#define KSCALE 128.0
main(argc,argv)
int argc;
char *argv[];
{
int i,j,ij,halfnum,n;
int numstages=0;
short out;
float a[MAXSTAGES],k[MAXSTAGES];
float atemp,q;
FILE *fp;
fp=stdin;
if (argc>1)
if ((fp=fopen(argv[1],"r"))==0){
        fprintf(stderr,"ktoa: can't open %s\\n",argv[1]);
        exit(1);};
a[0]=1;
for(i=1;i<MAXSTAGES;i++){
        a[i]=k[i]= -getsh(fp)/KSCALE;
        if(feof(fp)) break;}
numstages=i-1;
for (i=2;i<=numstages;i++){
        halfnum=i/2;
        q=k[i];
        for (j=1;j<=halfnum;j++){
                ij=i-j;
                atemp=a[j]+q*a[ij];
                a[ij]=a[ij]+q*a[j];
                a[j]=atemp;}}
for (i=0;i<=numstages;i++){
        out=a[i]*ASCALE;
        putsh(out,stdout);}
}
```

APPENDIX C

Process Schedule

For reference, the actual process schedule for the High Voltage metal-gate CMOS process is presented here. This process was extensively modified from that of Black [4] to improve speed. Junction depths were reduced and the bipolar components eliminated. The simulation program SUPREM [2] was used to determine the diffusion and implant parameters.

This process uses positive photo-resist throughout. Alignment was done with a projection aligner to ~1 μm precision. Differential gate oxide growth provided some degree of self-alignment for the n-channel devices [20]. The p-channel devices were self aligned by an implant over the metal. The aluminum gates were used as the mask and the sintering step activated the implant. Because of the large oxide steps in CMOS processes, the metal thickness was set near 10,000Å to prevent step coverage problems.

Berkeley Metal Gate CMOS Process
November 18, 1981 (rev. March 5 1982)

1. Initial wafer cleaning - p-type, 5-7 ohm-cm, <100>
   a. TCE clean, 60 °C, 10 minutes (Degrease) (Watch temperature : Boiling TCE will shatter wafers)
   b. Dip in acetone
   c. Dip in methyl alcohol
   d. Rinse in De-ionized (DI) water
   e. Piranha etch (H_2SO_4:H_2O_2 - 5:1) for 10 min.
   f. Rinse thoroughly in running DI water
   g. Dip in HF:H_2O (1:10) for 20 seconds
   h. Rinse in DI water
   i. Blow dry
   j. Inspect under collimated light for dust. (If dust remains, repeat from h.; if that doesn't work, repeat from a.)

2. Initial oxidation - 4000A, Initial Oxide Furnace
   a. Wet O_2, 1100 °C, 27 min. (.4μm) (temperature setting=1095 °C)
   b. Dry N_2 (anneal), 900 °C, 20 min.

3. Positive Photoresist (PR) Step, Mask #1 : N- well mask.
   a. Prebake, 85-90 °C, 15 min. (if wafer just came from furnace then skip prebake)
   b. HMDS treatment
      1. N_2 purge, 10 min.
      2. Load wafers
      3. Bubble HMDS, 3 min.
      4. N_2 purge, 5 min.
   c. Spin AZ 1450J, 6000 rpm, 30 seconds. (PR thickness = 1.4μm)
   d. Softbake, 85-90 °C, 15-20 min.
   e. Align and expose (Blow off dust on top and bottom)
   f. Develop, AZ 351 developer (DI H_2O:Developer - 5:1), 1 min. (or regular AZ developer at 1:1 dilution)
   g. Rinse in DI water and blow dry
   h. Inspect
   i. Hardbake, 110-120 °C, 20 min.
   j. Oxide etch, buffered HF (NH_4F:HF - 5:1) etch rate .12μm/min. (.145μm/min if solution is just made) (etch time= 3.5-4 min. with 15% over etch) (prepare at least 4 hours before use)
   k. PR strip, acetone, 15 min.
   l. Piranha clean, H_2SO_4:H_2O_2 - 5:1, 15 min.

4. Implant step - Initial Oxide Furnace
   a. Grow implant oxide - 290Å
      1. Wet O_2, 900 °C, 3 min., flowmeter setting=4 cm
      2. Dry N_2, 900 °C, 5 min., flowmeter setting=4 cm
   b. Implant Phosphorus (N-), Dose=$1x10^{13}$, 200keV (xj(well)=.462μm, Rsh=1.29kohm/square)

5. Piranha Clean, H_2SO_4:H_2O_2 - 5:1, 5 min.

6. Well oxidation - 7229Å, Initial Oxide Furnace
   a. Wet O_2, 1150 °C, 65 min., (.76/.92μm) (setting=1144)
   b. Dry N_2, 900 °C, 30 min. (xj(well)=1.978μm, Rsh=943 ohms/square)

7. Well drive-in, Arsenic Furnace
   a. Dry N_2, 1200 °C, 24 hours
   b. Dry N_2 (anneal), 900 °C, 60 min. (xj(well)=9.055μm, Rsh=827 ohms/square)

8. Pos. PR Step, Mask #2 : N+ (N channel S/D) mask.
   a. Apply photoresist, expose, develop (follow 3.a-i)
   b. Oxide etch, buffered HF (NH_4F:HF - 5:1) etch rate .115μm/min. (etch time=7.5 min.)
   c. PR strip
   d. Piranha clean, 15 min.

9. Implant step - Initial Oxide Furnace
   a. Grow implant oxide - 290Å
      1. Wet O_2, 900 °C, 3 min.
      2. Dry N_2, 900 °C, 5 min.
   b. Implant Phosphorus (N+), Dose=$5.84x10^{15}$, 160keV

10. Piranha Clean, 5 min.

11. Oxide growth over N+ - 3983Å, N+ Drive-in Furnace
    a. Wet O_2, 950 °C, 60 min.
    b. Dry N_2, 900 °C, 30 min. (xj(N channel S/D)=0.476μm, Rsh=27.88 ohms/square)

12. Pos. PR Step, Mask #3 : P+ (P channel S/D) mask.
    a. Apply photoresist, expose, develop (follow 3.a-i)
    b. Oxide etch, buffered HF (NH_4F:HF - 5:1) etch rate .115μm/min. (etch time=7.5 min.)
    c. PR strip

d. Piranha clean, 15 min.

13. Implant step - P+ Drive-in Furnace
    a. Grow implant oxide - 290Å
        1. Wet O_2, 900 °C, 3 min. (.29/.40μm)
        2. Dry N_2, 900 °C, 5 min.
    b. Implant Boron (P+), Dose=5.50x10^{15}, 32keV (xj(P channel S/D)=.598μm, Rsh=22.28 ohms/square, Rsh(well)=958 ohms/square

14. Gettering step :
    Implant Boron Difluoride into back side of wafer
    Dose=5x10^{15}, 200 keV

15. Piranha Clean, 5 min.

16. Oxide growth over P+ and N+/P+ drive-in - 3200Å, P+ drive-in Furnace
    a. Wet O_2, 1100 °C, 15 min. (setting=1110) (Calibrate furnace temp. first)
    b. Dry N_2, 1100 °C, 28 min. (27 min. then 2 min. pulling out) (xj(N channel S/D)=1.654μm, Rsh=18.39 ohms/square) (xj(P channel S/D)=1.589μm, Rsh=40.9 ohms/square)

17. Pos. PR Step, Mask #4 : Thin Oxide (P channel and N channel Gates) mask.
    a. Apply photoresist, expose, develop (follow 3.a-i)
    b. Oxide etch, buffered HF (NH_4F:HF - 5:1) etch rate .115μm/min. (etch through ~12000 Å oxide) (etch time=7.5 min.)
    c. PR strip
    d. Piranha clean, 15 min.

18. Gate Oxide Growth - 840Å/P+(well), 824Å/P+(substrate), 1622Å/N+ (or 1100Å/gate, 1200Å/N+) P+ drive-in furnace
    a. TCE clean P+ drive-in furnace at least 12 hours before use
    b. Wet O_2, 840 °C, 75 min. (Calibrate temp. first, check water after 1.5 hour) (setting=840) (or Dry O_2, 1000 °C, 145 min., init. ox. furnace)
    c. Dry N_2 (anneal), 840 °C, 15 min. (or Dry N_2 (anneal),1000 °C, 20 min. for Dry O_2 growth)

19. Threshold shift implant - Boron, Dose=5x10^{11}, 55 keV (75 keV alt.)

20. Piranha clean, 5 min.

21. Threshold shift anneal - P+ drive-in furnace Dry N_2, 1000 °C, 15 min. (load and unload wafers in SLOWLY)

22. Pos. PR Step, Mask #5 : Contact cut mask.
    a. Apply photoresist, expose, develop (follow 3.a-i) (spin on at 5000 rpm) (check very carefully for gaps)

2. O_2, .76 torr, 10 watts, 5 min.

c. Oxide etch, buffered *HF* (NH_4F:*HF* - 5:1) etch rate .115μm/min. (etch time=1.5 min.) (etch through ~1700A of oxide)

d. PR strip

e. Piranha clean, 15 min.

23. Metal deposition - 8000A Aluminum

a. *HF* dip (*HF*:water - 1:20), 10 sec. (just until back of wafer repels water)

b. Bake under IR lamp, 15 min.

c. Use following settings: 25μA,5.2 remote,1 staple,~$5x10^{-6}$ Torr,80 sec.,25 toward

24. Pos. PR Step, Mask #6 : Metal mask.
Apply photoresist, expose, develop (follow 3.a-i) (spin on at 6200 rpm) (VERY CRITICAL ALIGNMENT to N+)

25. Metal etch - Al etchant type A, 45-50 °C, (Do not overetch) (~3.5 min.)

26. PR strip, acetone, 10 min.

27. Rinse in DI water

28. Pos. PR Step, Mask #1 : Self align implant. Apply photoresist, expose, develop (follow 3.a-i) (spin on at 6000 rpm)

29. Implant (self aligned PMOS) Boron, Dose=$3.81x10^{13}$, 45 keV (dose=40 μcoulombs)

30. PR strip

a. Acetone, 30 min. (1 min. ultrasonic every 15 min.)

b. DI rinse and blow dry

c. Plasma etch, 150 watts, 1 torr, 5 min. or until done

31. Sinter Aluminum - 450 °C, 15 min. with forming gas,14 cm.

32. Spin on PR, 15 min. prebake(100 °C), 5000 rpm, 30 sec., 20min. softbake(90°C)

33. Dice - dicing saw, speed=2(hi), 4.0 mils not sawed, channel 1 = 4.250,2.500, channel 2= 4.630,1.750, distance to edge of chip= .250 mm.

APPENDIX D

Layout Rules

The layout and design rules that were used in making the analog switched capacitor integrated circuit are presented here. The design rules apply to the design of any analog switched capacitor integrated circuit. The layout rules are specific to the metal-gate CMOS process used in this research. The layout rules were designed to provide good reliability and yield and provide high breakdown voltage (30 volts). Linewidths and spacings were minimized in order to reduce speed and chip size. The channel widths are designed to be 8-10 μms after diffusion, a factor of two smaller than previous metal gate processes used at U.C. Berkeley.

D.1. Layout Rules

Table D.1 lists the basic minimum geometry layout rules. Wherever possible, lines are made about 50% larger. For example, long metal lines without tight space constraints are usually 14 μms. Diffusion lines are usually made 16 μms to reduce parasitic resistance except in circuit locations that are sensitive to junction parasitics. Contact cuts are surrounded by thin oxide cuts extending 2 μm on every side in order to reduce step coverage problems and decrease oxide etch time. Possible problems due to pinholes in the positive contact mask are reduced by the shortened etch time.

D.2. Design rules

The following are electrical design rules designed to minimize parasitic effects such as coupling from the clocks into signal paths. Electrical signal paths

Minimum Geometry Layout Rules				
Layer (or feature)	To Layer (or feature)	Minimum Linewidth (μm)	Minimum Spacing (μm)	Minimum Overlap (μm)
n^--well	n^--well	28	43	
	p^+-diffusion			9
	n^+-diffusion (guard ring)		4	
p^+ or n^+-diffusion	p^+-diffusion	8	13	
	n^+-diffusion		13	
thin oxide	thin oxide	9	10	
	contact cut			2
	channel			1
contact cut	contact cut	8	21	
	metal			3
metal	metal	10	10	
	NMOS channel			2
	PMOS channel			-1
	thin oxide			4
bonding pad	anything	100	100	

Table D.1. The spacings for the MOS channel regions are as drawn on the mask (before lateral diffusion).

are categorized into one of four types:

(1) power and ground lines,

(2) clock lines,

(3) signal lines (moderate to low impedance),

(4) high impedance signal points.

An example of a low impedance signal line would be an op-amp output. Some examples of high impedance points are op-amp inputs and capacitive storage nodes. Component parasitic effects are also considered in the following list.

(1) Keep clock and signal lines spaced as far apart as practical (ie. give 50-100% more space than minimum between adjacent lines). Also, use crossunders between low impedance signal and clock lines sparingly.

(2) Wherever possible, run power or ground lines between clock and signal lines.

(3) Shield high impedance points by running ground lines around them. Avoid the use of crossunders between high impedance points and clocks.

(4) Avoid using diffusion crossunders for power lines, because the parasitic resistance will cause a voltage drop.

(5) Crossunders can cause problems for clock lines if there is heavy capacitive loading.

(6) Parasitic capacitance should be minimized at op-amp inputs to minimize noise. The top plate of diffusion capacitors should therefore go to op-amp inputs unless bottom plate leakage current will become a problem.

(7) Minimum geometry MOS switches should be used to reduce charge injection. Under some conditions the W/Ls may have to be increased if speed becomes important.

(8) Matched capacitors should be made square to maximize precision. The metal top plates should be surrounded by equally spaced metal for better

control over the etch rate. If necessary, dummy metal lines should be used.

(9) Ratioed capacitors should be made of unit blocks. These minimum size capacitors are then strung together to form the required value.

REFERENCES

[1] D. J. Allstot, R. W. Brodersen, and P. R. Gray, "MOS Switched-Capacitor Ladder Filters," *IEEE Journal of Solid-State Circuits*, Vol. SC-13, No. 6, pp. 806-814, December 1978.

[2] D. A. Antoniadis, S. E. Hansen, and R. W. Dutton, "SUPREM II - A Program for IC Process Modeling and Simulation," Tech. Rep. SEL 78-020, Stanford Electronics Labs., Stanford University, June 1978.

[3] B. S. Atal and S. L. Hanauer, "Speech Analysis and Synthesis by Linear Prediction of the Speech Wave," *Journal of the Acoust. Society of America*, Vol. 50, No. 2 (Part 2), pp. 637-655, August 1971.

[4] W. C. Black, Jr., "High Voltage Metal Gate CMOS Process for Large Scale Analog Circuit Applications," M.S. thesis, University of California, Berkeley, June 1977.

[5] W. C. Black, Jr., "High Speed CMOS A/D Conversion Techniques," Ph.D. dissertation, University of California, Berkeley, Memorandum No. UCB/ERL M80/54, November 1980.

[6] P. W. Bosshart, "A Multiplexed Switched Capacitor Filter Bank," *IEEE Journal of Solid-State Circuis*, Vol. SC-15, No. 6, pp. 939-945, December 1980.

[7] R. W. Brodersen and S. P. Emmons, "Noise in Buried Channel Charge-Coupled Devices," *IEEE Journal of Solid-State Circuits*,

Vol. SC-11, No. 1, pp. 147-155, February 1976.

[8] R. W. Brodersen, P. R. Gray, and D. A. Hodges, "MOS Switched-Capacitor Filters," *Proceedings of the IEEE*, Vol. 67, pp. 61-75, January 1979.

[9] J. Burg, "Maximum Entropy Spectral Analysis," Ph.D. dissertation, Stanford University, Stanford, CA, May 1975.

[10] W. N. Carr and J. P. Mize, **MOS/LSI Design and Application**. McGraw-Hill: New York, 1972. pp. 21-58.

[11] J. Durbin, "Efficient Estimation of Parameters in Moving-Average Models," *Biometrika*, Vol. 46, Parts 1 and 2, pp. 306-316, 1959.

[12] C. G. M. Fant, **Acoustic Theory of Speech Production**. Mouton and Co.: 's-Gravenhage, The Netherlands, 1960.

[13] R. D. Fellman, "A 30-Stage Charge Transfer Device Transversal Filter with Absolute Value Input," M.S. Thesis, University of California, Berkeley, CA 94720, June 1977. pp. 6-8 and pp. 14-15.

[14] R. D. Fellman, P. J. Hurst, and R. W. Brodersen, "MOS-LSI Analog/Digital Techniques for Linear Prediction," *International Communications Conference*, Seatle, Washington, June 1980.

[15] B. Fette, "Correlation Windowing in Flow Form Formulation of lattice Filter Linear Prediction of Speech," Ph.D. dissertation, Arizona State University, June 1981.

[16] A. P. French, **Vibrations and Waves**. W. W. Norton & Co.: New York, 1971. pp. 174-5.

[17] P. R. Gray and R. G. Meyer, **Analysis and Design of Analog Integrated Circuits**. John Wiley & Sons: New York, 1977.

[18] D. J. Hamilton and W. G. Howard, **Basic Integrated Circuit Engineering**. McGraw-Hill: New York, 1975.

[19] J. Heiserman and I. Rudnick, **Notes for Physics 7C: Vibration, Wave Motion, Sound, Heat, and Kinetic Theory**, University of California, Los Angeles, September 1973. pp. 32-45 and pp. 74-100.

[20] C. P.-H. Ho, "Silicon Oxide Studies: Physical Modeling and Device Applications of Thermal Oxidation of Heavily Doped Silicon," Technical Report No. 4970-2, Stanford Electronics Labs., Stanford University, Stanford, CA 94305, November 1978. pp. 37-55 and pp. 127-147.

[21] P. G. Hoel, S. C. Port, and C. J. Stone, **Introduction to Probability Theory**. Houghton Mifflin Co.: Boston, 1971. pp. 82-100.

[22] K.-C. Hsieh, P. G. Gray, D. Senderowitz, and D. G. Messerschmitt, "A Low-Noise Chopper-Stabilized Differential Switched-Capacitor Filtering Technique," *IEEE Journal of Solid-State Circuits*, Vol. SC-16, No. 6, pp. 708-715, December 1981.

[23] F. Itakura and S. Saito, "Speech Analysis/Synthesis by Partial Correlation Coefficients," *Proceedings of the Fourth Electrical Engineers Society Joint Conference*, Paper 2835, 1970 (in

Japanese).

[24] R. Kaneshiro, Private communications.

[25] G. S. Kang, "Application of Linear Prediction Encoding to a Narrowband Voice Digitizer," NRL Report 7774 , Naval Research Laboratory, Washington, D.C. 20375, 1974.

[26] G. S. Kang, L. J. Fransen, E. L. Kline, "Flow-Form Implementation of a Linear Predictive Coderfor Speech Communication," *SPIE: Real-Time Signal Processing II*, Vol. 180, pp. 15-24, 1979.

[27] G. Smarandoiu, D. A. Hodges, P. R. Gray, and G. F. Landsburg, "CMOS Pulse-Code-Modulation Voice Codec," *IEEE Journal of Solid-State Circuits*, Vol. SC-13, No. 4, pp. 504-510, August 1978.

[28] J. Makhoul, "Linear Prediction: A Tutorial Review," *Proceedings of the IEEE*, Vol. 63, No. 4, pp. 561-580, April 1975.

[29] J. Makhoul, "Stable and Efficient Lattice Methods for Linear Prediction," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, Vol. ASSP-25, No. 5, pp. 423-428, October 1977.

[30] J. D. Markel and A. H. Gray, Jr., "A Linear Prediction Vocoder Simulation Based upon the Autocorrelation Method," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, Vol. ASSP-22, pp. 124-134, April 1974.

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

[32] R. H. McCharles and D. A. Hodges, "Charge Circuits for Analog LSI," *IEEE Transactions on Circuits and Systems*, Vol. CAS-25, No. 7, pp. 490-497, July 1978.

[33] J. L. McCreary and P. R. Gray, "All-MOS Charge-redistribution A/D Conversion Techniques," *IEEE Journal of Solid-State Circuits*, Vol. SC-10, pp. 371-9, December 1975.

[34] C. Mead, l. Conway, **Introduction to VLSI systems.** Addison-Wesley: Reading, Mass., 1980. pp. 51-3.

[35] T. J. Mroz, "Designing with Codecs: Know Your A's and μ's," **Siliconix- Telecommumications Data Book**, TA78-1, pp. 1-41 - 1-46, Siliconix Inc.: Santa Clara, CA 95054, July 1978.

[36] A. V. Oppenheim and R. W. Shafer, **Digital Signal Processing.** Prentice-Hall: Englewood Cliffs, New Jersey 07632, 1975.

[37] S. Pope, B. Solberg, R. D. Fellman, and R. W. Brodersen, "A Monolithic MOS-LSI Vocoder System," *To be published.*

[38] L. R. Rabiner and B. Gold, **Theory and Application of Digital Signal Processing.** Prentice-Hall: Englewood Cliffs, New Jersey 07632, 1975.

[39] L. R. Rabiner and R. W. Shafer, **Digital Processing of Speech Signals**. Prentice-Hall: Englewood Cliffs, New Jersey 07632, 1978.

[40] **RCA Linear Integrated Circuits**. Data Book, RCA Solid State, RCA Corporation, 1978. pp. 132-7 and pp. 146-151.

[41] R. W. Schafer and L. R. Rabiner, "System for Automatic Formant Analysis of Voiced Speech," *Journal of the Acoust. Society of America*, Vol. 47, pp. 634-678, 1970.

[42] D. Soo and R. G. Meyer, "A 4-Quadrant Analog MOS Multiplier," *Proc. of the IEEE Int'l. Solid-State Circuits Conference*, Vol. 29, pp. 36-37, February 1982.

[43] S. M. Sze, **Physics of Semiconductor Devices**. John Wiley & Sons: New York, 1969. pp. 56-65.

[44] R. Viswanathan and J. Makhoul, "Quantization Properties of Transmission Parameters in Linear Predictive" Systems," *IEEE Trans. on Acoustics, Speech, and Signal Proc.*, Vol. ASSP-23, No. 3, pp. 309-321, June 1975.

[45] A. Vladimirescu and S. Liu, "The Simulation of MOS Integrated Circuits Using Spice2," Memorandum No. UCB/ERL M80/7, Electronics Research Lab., University of California, Berkeley, CA 94720, February 1980. pp. 36-37.

[46] S. Wasser, "Survey of VLSI for Digital Signal Processing." *IEEE Int'l Conf. on Acoust., Speech, and Sig. Proc.*, Vol. ICASSP-80, pp. 376-

379, April 1980.