

Copyright © 1988, by the author(s).
All rights reserved.

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

**ACOUSTIC ECHO CANCELLATION
FOR LOUDSPEAKER TELEPHONES**

by

Wen-Bin Hsu

Memorandum No. UCB/ERL M88/67

7 November 1988

COVER PAGE

**ACOUSTIC ECHO CANCELLATION
FOR LOUDSPEAKER TELEPHONES**

by

Wen-Bin Hsu

Memorandum No. UCB/ERL M88/67

7 November 1988

ELECTRONICS RESEARCH LABORATORY

College of Engineering
University of California, Berkeley
94720

TITLE PAGE

**ACOUSTIC ECHO CANCELLATION
FOR LOUDSPEAKER TELEPHONES**

by

Wen-Bin Hsu

Memorandum No. UCB/ERL M88/67

7 November 1988

ELECTRONICS RESEARCH LABORATORY


College of Engineering
University of California, Berkeley
94720

ACOUSTIC ECHO CANCELLATION FOR LOUDSPEAKER TELEPHONES

Ph.D.

Wen-Bin Hsu

Department of EECS


Chairman of Committee

ABSTRACT

The echo problems associated with loudspeaker telephones can be categorized as far-end talker echoes and near-end talker echoes. The operational difficulties due to these echoes are discussed and solutions are described. Among the various solutions are voice switching, echo cancellation, and multimicrophone reception. By comparing the operation and the limitations, the echo cancellation techniques appear to be a potential candidate for low-cost methods.

The requirements for eliminating far-end talker echoes in typical rooms were considered and an acoustic echo canceler is proposed. The acoustic echo canceler uses a 1000 tap transversal filter with floating point data representation (5 bit mantissa, 1 bit sign, and 3 bit exponent.) The trade-offs between performance and complexity were studied. The effects of the imperfections in A/D and D/A converters on the performance of the echo cancelers were examined. An implementation scheme that will fit in a single IC chip is described.

To show the feasibility of this scheme, a 1000 tap adaptive acoustic echo canceler occupying 28 mm^2 of die area in $3 \text{ }\mu\text{m}$ CMOS was designed and fabricated. The experimental results show 27 dB echo reduction being achieved after 1 second of convergence time.

ACKNOWLEDGEMENT

I would like to thank Professor David Hodges, my research adviser, for his support and guidance throughout my Ph.D. study. His advice is always excellent. I am grateful to Professor David Messerschmitt for his help and suggestions in this research work. Gratitude is also due to Professor Bob Brodersen for providing us the support of his speech lab to evaluate our schemes, to Professor Paul Gray for his constant encouragement, and to Professor Edward Lee for many interesting discussions.

The exchange of ideas with fellow students at UC Berkeley was particularly fruitful. Among them, Joey Doernberg, Bosco Leung, Steve Lewis, Yuh-Min Lin, Sehat Sutarja, Ho-Ping Tseng deserve special mention. The discussions with Takafumi Chujo of Fujitsu Laboratory during his one year visit in Berkeley were very constructive. Frank Chui helped lay out the data path of the chip. Ken Lutz provided me his assistance in the laboratory setup.

This research was supported by grants from Advanced Micro Devices, Bell Communications Research, Fairchild Semiconductor, Harris Corp., National Semiconductor, with matching grants from the University of California's MICRO program, and by the National Science Foundation under grant number MIP-8603430. Chips were fabricated through the MOSIS service. The working prototypes of the echo canceler chip have undergone the "micro-surgery" provided by Seiko Instrument to cut wires and deposit connections.

Finally, I thank my parents, Mr. and Mrs. Chau-Yang Hsu, who made my dream to study in the America come true, and my wife, Erlene, whose love makes my study in Berkeley all worthwhile.

Table of Contents

CHAPTER 1 - INTRODUCTION	1
1.1 Motivations	1
1.2 Thesis Organization	3
CHAPTER 2 - ECHO PROBLEMS AND SOLUTIONS IN LOUDSPEAKER TELEPHONES	4
2.1 Impairments in Loudspeaker Telephones	4
2.1.1 Singing and Stability Problem	6
2.1.2 Far-End Talker Echoes	8
2.1.3 Reduced Signal-to-Noise (S/N) Ratio	8
2.1.4 Near-End Talker Echoes	10
2.2 Voice-Switched Loudspeaker Telephones	13
2.2.1 Principles of Operation	13
2.2.2 Limitations	14
2.3 Echo Cancellation to Remove Far-End Talker Echoes	15
2.3.1 Principles of Operation	16
2.3.2 Implementation Difficulties	17
2.4 Equalization to Alleviate Near-End Talker Echoes	18
2.4.1 Previous Work	18
2.4.2 Adaptive Equalization	22
CHAPTER 3 - REQUIREMENTS OF ACOUSTIC ECHO CANCELERS	29
3.1 Room Acoustics of Sound Propagation	29
3.1.1 Image Method for Sound Reflections	30
3.1.2 Echo Response in the Average Sense	31
3.2 Reverberation Time and Process Window	33
3.3 Sound Pressure Level and Echo Reduction	35
CHAPTER 4 - DESIGN CONSIDERATIONS: A SYSTEM PERSPECTIVE	38
4.1 A Brute Force Approach	38
4.2 Quantization Effects	40
4.3 Adaptation Algorithm	43
4.4 Hardware Implementation	44
4.5 Computer Simulations	46
4.6 Detection of Near-End Talker Signals	48
4.7 Alternative Implementations of the Acoustic Echo Canceler	49

4.8	Effects of Imperfections in A/D and D/A Converters	52
4.8.1	Gain Error	52
4.8.2	DC Offset	54
4.8.3	Nonlinearity	59
CHAPTER 5 - PROTOTYPE ACOUSTIC ECHO CANCELER		63
5.1	Design Methodology and Design Environment	63
5.2	Architecture	66
5.2.1	Chip Interface	66
5.2.2	General Clock Scheme	68
5.3	Memory Allocation and Access Control	69
5.3.1	Interleaved Data Storage	69
5.3.2	Sharing the Same Address for all Memory Banks	72
5.3.3	Memory Access Control Circuits	74
5.4	Data Path of Adaptation and Convolution	74
5.4.1	The Adaptation Processor	75
5.4.2	The Convolution Processor	77
5.5	Layout and Fabrication	80
5.5.1	Floor Plan	80
5.5.2	Basic Cell Designs	81
5.5.3	Layout Verification and Simulation	83
5.5.4	Fabrication	86
5.6	Measurements and Discussions	87
5.6.1	Measurement Setup	87
5.6.2	Experimental Results	90
CHAPTER 6 - CONCLUSIONS		95
6.1	Summary of Research Results	95
6.2	Future Work	96
References		98

CHAPTER 1

INTRODUCTION

Voice communication through telephones has long been an important part of the social life because it offers audible and spontaneous interactions between the participating parties. However, telephone conversation using a handset forces the user to hold on to the handset. The loudspeaker telephone was designed to free the user from the constraints imposed by a handset telephone.

1.1 Motivations

The loudspeaker telephone has become an important piece of office equipment because it provides the user the convenience of hands-free telephone conversation. This feature is particularly useful in the upcoming age of the *Integrated Service Digital Network* (ISDN.) The ISDN technology is expected to provide its subscribers various communication media such as voice, image, and data. A subscriber can communicate with another through the voice channel while at the same time exchange personal data and images. For example, they can refer to the same graph or text in their voice communication. As a result, hands-free voice conversation is critical because the subscribers can better use their hands for typing on a keyboard to send data or pointing to a common figure.

As corporations continue to seek cost reduction and efficiency improvement, teleconferencing is becoming an attractive alternative to business travel. Perfecting the teleconference environment is a challenging task for today's communication experts. Because the teleconference is hands free by nature, it will benefit from the same technology that is

available in a loudspeaker telephone. The differences lie in the desired degree of perfection and the amount of money to spend on improvement.

Loudspeaker telephones are far from perfection. Common complaints include a talker hearing his own speech, half-duplex conversation (with voice switching), background noise chopping (also with voice switching), and barrel-like echoes. The acoustic coupling between the loudspeaker and the microphone is a major operating difficulty in a loudspeaker telephone. Although voice switching (which allows only one direction of transmission at any given time) presents a low-cost method to decouple the acoustic feedback, it also creates additional problems such as chopping of speech and background noise.

Echo cancellation techniques, which are very effective in other areas of application, are a potential candidate for improving the quality of loudspeaker telephones. However, the complexity of a direct implementation of an echo canceler to be used in an acoustic environment makes it unfeasible in a single IC chip even in today's technology.

More understanding of the problem is needed so that the engineering trade-offs can be established. By carefully balancing the performance and the complexity of an acoustic echo canceler, we hope to bring about a feasible and cost-effective solution. One of our objectives in this project is to demonstrate an appropriate integrated circuit design from a system perspective. As the level of system integration (into an IC chip) increases, conventional circuit optimization alone is not sufficient. One of the purposes in this project is to demonstrate how various levels in the design process can be combined to serve the same objective - a low-cost and effective method. In particular, the criteria for designing an acoustic echo canceler to remove the far-end talker echoes (defined in Chapter 2) are established. Techniques to realize a single-chip acoustic echo canceler are illustrated through the design of a prototype chip.

1.2 Thesis Organization

This thesis begins with a detailed examination of the echo problems associated with a loudspeaker telephone. Various techniques to cope with these problems are presented in Chapter 2, with special attention to their performance limitations and/or implementation difficulties. An adaptive acoustic echo canceler in a single IC chip is chosen as a low-cost solution. Its functional requirements are described in Chapter 3. The purpose is to determine the minimum performance requirements and to allow compromises between the performance and the complexity. These compromises are then exploited in the design considerations described in Chapter 4. Analysis and computer simulations are used to examine the various design issues. Chapter 5 describes the design of a single chip acoustic echo canceler. A structured design process and a coordinated design environment are the keys to a successful implementation. Architecture and circuit design techniques are also addressed in Chapter 5. Experimental results are presented. Conclusions from this project are summarized in Chapter 6, along with some suggestions on future work on the related topics.

CHAPTER 2

ECHO PROBLEMS AND SOLUTIONS IN LOUDSPEAKER TELEPHONES

Echo problems have long been associated with loudspeaker phones [1,2]. A detailed study shows that the echo problems can be divided into far-end talker echoes, near-end talker echoes, and distant talking (Section 2.1.) One solution (the one in use nowadays) to the problem of far-end talker echoes is to use voice switching. Its operation and limitations will be discussed in Section 2.2. Section 2.3 presents the application of acoustic echo cancelers as a potential alternative. The reductions in near-end talker echoes are the subject of Section 2.4. The differences in applying echo cancellation techniques to the far-end talker echoes and the near-end talker echoes are identified. An inverse filtering technique will be proposed to remove the near-end talker echoes. One common requirement is the need of an adaptive transversal filter with large number of taps.

2.1 Impairments In Loudspeaker Telephones

Figure 2.1 is a simplified diagram of a loudspeaker telephone connection, which contains a "two-wire" segment at the left side and a "four-wire" segment including the loudspeaker and the microphone on the right. In the two-wire portion of the connection, both directions of transmission are carried on the same wire pair. Joining the loudspeaker, the microphone, and the two-wire link is a hybrid, H. The hybrid performs a conversion from four-wire to two-wire. It is a nonreciprocal device that sends the transmitted signals from the microphone to the two-wire telephone line and directs the received signals from the telephone

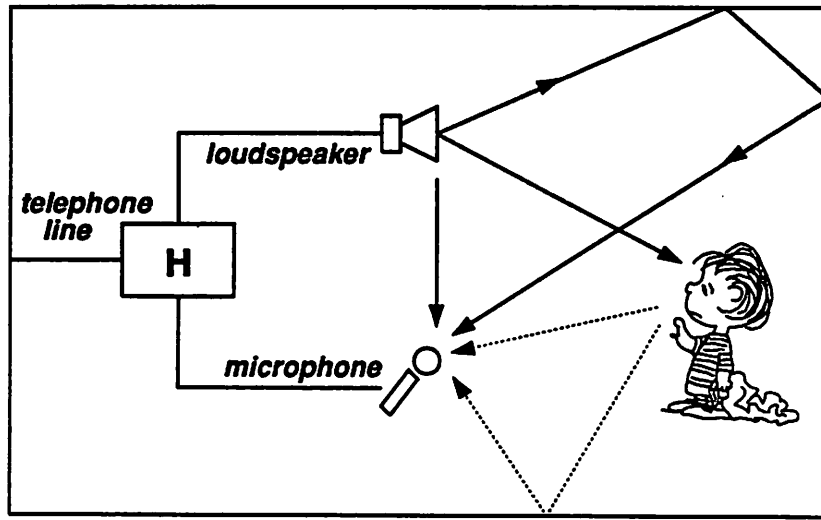


Figure 2.1 - Block diagram of a loudspeaker telephone connection

line to the loudspeaker. An ideal hybrid allows the microphone signal to be transmitted to the two-wire line without being fed to the loudspeaker and the received signal to be routed to the loudspeaker without any loss. An electronic hybrid will be described in the next section. For reference purposes, the person using the loudspeaker telephone is the near-end talker and the person at the other end of the connection is the far-end talker.

The speech of the far-end talker is acoustically radiated by the loudspeaker. It can bounce back and forth between the walls of the room and the furniture and can be picked up by the microphone and be transmitted back to the far-end talker as a far-end talker echo. This far-end talker echo can be very annoying because it causes the far-end talker to hear a delayed version of his or her own speech. On the other hand, the microphone not only picks up the direct sound of the near-end talker but also reverberant sound reflections (the dashed lines in Figure 2.1.) These sound reflections are near-end talker echoes. The energy of the direct sound decreases proportionally to the square of the separation distance. Higher gain (which

also boosts the ambient noise) is required with increasing separation resulting in reduced signal-to-noise (S/N) ratio.

2.1.1 Singing and Stability Problem

The loudspeaker telephone connection of Figure 2.1 is illustrated in further detail in Figure 2.2.

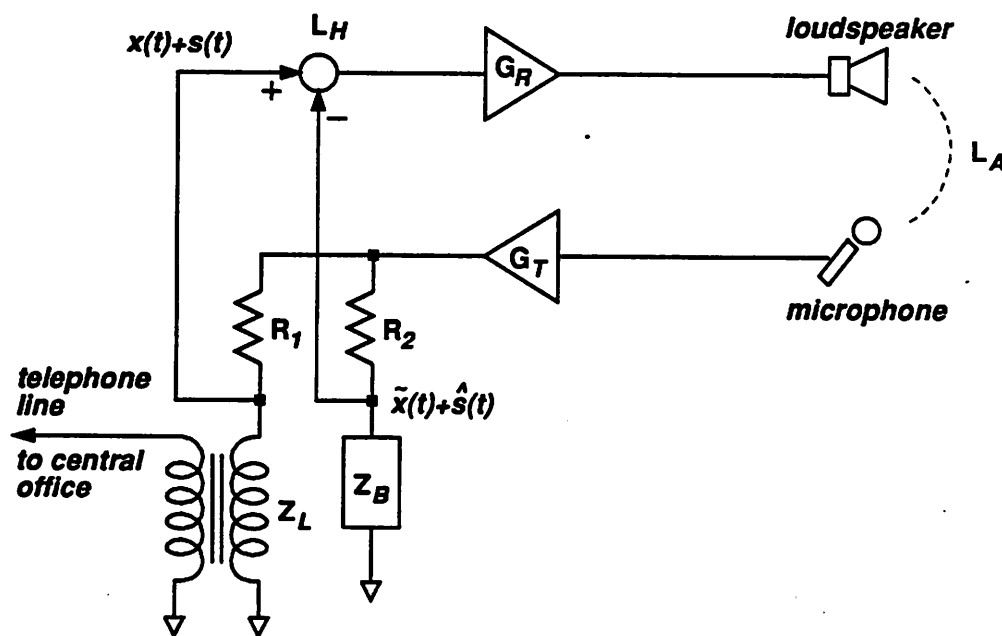


Figure 2.2 - Functional diagram of a loudspeaker telephone connection

An electronic hybrid is implemented by R_1 , R_2 , Z_B , Z_L , and a subtractor. The transmitting amplifier (G_T) is designed so that the signal level delivered to the central office is compatible to that delivered by a handset telephone. The transmitting gain G_T is usually fixed for a given nominal talking distance [1]. The receiving amplifier (G_R) with a variable gain for volume control produces the needed signal level to drive the loudspeaker. Although the receiving amplifier and the subtractor are shown separately in Figure 2.2, they can be implemented with

a single differential amplifier. The transformer isolates the loudspeaker telephone from the telephone line.

The near-end talker signal (picked up by the microphone and amplified by G_T) passes through a voltage divider with resistance R_1 in series with the impedance Z_L . The voltage at the center of this divider is the far-end talker signal $x(t)$ plus the near-end talker signal $s(t)$. To prevent this near-end talker signal from leaking to the loudspeaker, a second voltage divider with resistance R_2 and a balancing impedance Z_B generates a replica $\hat{s}(t)$. This replica is then subtracted from the signal across the transformer ($x(t) + s(t)$.) The purpose of Z_B is to match the transfer functions of the two voltage dividers, which is given by

$$\frac{R_1}{R_1 + Z_L} \approx \frac{R_2}{R_2 + Z_B} \quad (2.1)$$

If equation (2.1) is an exact equality (for example $R_1 = R_2$ and $Z_L = Z_B$), the replica $\hat{s}(t)$ will be the same as $s(t)$ and no component of the near-end talker signal will appear on the loudspeaker. However, the impedance Z_L depends on the detailed characteristics of the subscriber loop (such as line gauge, length, bridge taps, and distant termination), which varies from one subscriber loop to another. The choice of the balancing impedance Z_B is, therefore, a compromise. Consequently, the attenuation (L_H) of the feedthrough near-end talker signal is about 6 dB to 10 dB. The air-path loss from the loudspeaker to the microphone (L_A) depends on the separation between the loudspeaker and the microphone, their directivities, and the acoustics of the ambient objects.

Singing will occur if

$$G_T + G_R > L_H + L_A \quad (2.2)$$

Because there are electrical signal and acoustic signal in the loop, the gains G_T and G_R in equation (2.2) take into account the sensitivities of the microphone and the loudspeaker [3]. Solutions to prevent singing include reducing G_T or G_R (voice switching in Section 2.2) and increasing L_A and L_H (echo cancellation in Section 2.3.)

2.1.2 Far-End Talker Echoes

Because of the acoustic coupling between the loudspeaker and the microphone, the far-end talker will receive a delayed (hopefully attenuated) replica of his own speech as far-end talker echoes. Since they have been delayed by a round trip (as much as 1 second if via satellite), they are very annoying. In some cases, the far-end talker echoes can carry the far-end talker to the point of temporary stuttering. This phenomenon is analogous to one talking to oneself in a tunnel.

2.1.3 Reduced Signal-to-Noise (S/N) Ratio

In free space, the magnitude of the received sound pressure decreases proportionally with increasing separation between the receiver and the transmitter. For a loudspeaker telephone user with a talking distance of 18 to 21 inches (as opposed to the 1 inch talking distance of a handset user), an additional transmitter gain of 25 dB ($20 * \log 18 = 25$) is needed to produce a compatible signal level to that delivered by a handset telephone. This magnitude of insertion gain has been reported in [1]. The amplification not only increases the energy of the direct and the reverberant speech signal, but also raises the power of the ambient noise. As a result, the signal-to-noise ratio is reduced with increasing separation distance between the near-end talker and the microphone. With a handset, this is hardly a problem because the speech reaching the transmitter is much louder than the ambient noise reaching the transmitter, and the local noise at the receiving end is usually larger than the transmitted noise.

Lochner and Burger [4,5] have measured the syllable intelligibility of speech as a function of the speech level and the noise level. Their results are presented in Figure 2.3. The quantitative measure of the speech intelligibility was obtained by counting the number of discrete speech units correctly recognized by a listener. The procedure consisted of an

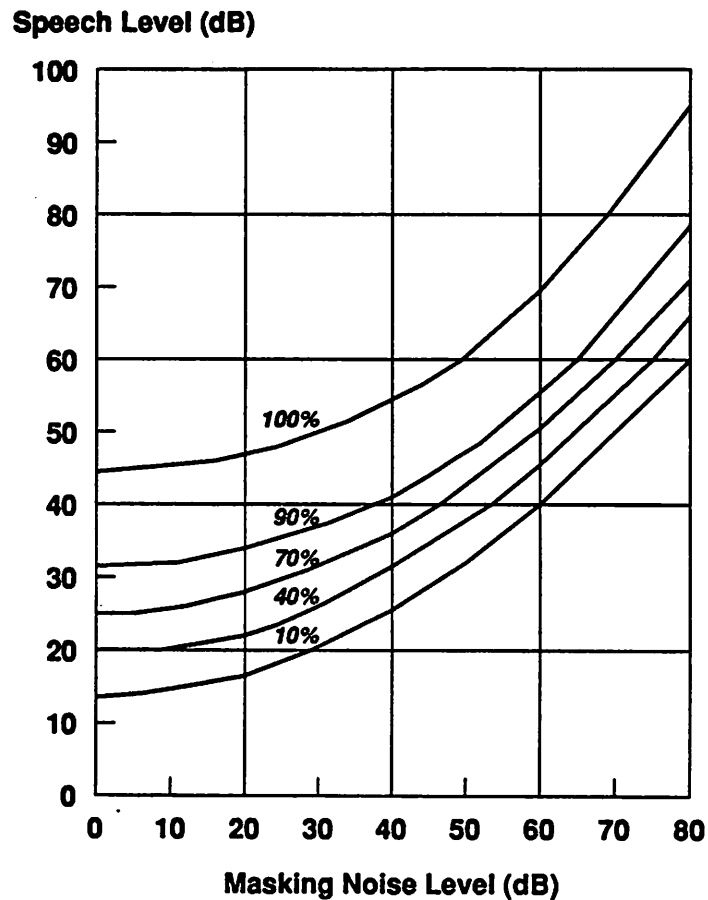


Figure 2.3 - Speech intelligibility as a function of speech levels and noise levels

announcer reading out lists of syllables, words, or sentences to one or more listeners. The percentage of items correctly recorded by these listeners was taken as a measure of the speech intelligibility. The masking noise was random noise filtered to give the same energy spectrum as that of the speech used in the tests. Each curve in Figure 2.3 corresponds to a constant intelligibility value. For a talking distance of 1.5 feet, the speech level is about 70 dB SPL* [2]. To obtain 100% intelligibility, the ambient noise should be no more than 60 dB SPL. Fortunately, the noise levels in the frequency range from 250 Hz to 4 kHz in a

* SPL is a relative sound pressure level in dB compared to a fixed standardized reference level (2×10^{-5} Newton/m²) which is roughly the hearing threshold of human ears at 1 kHz.

"moderately noisy" environment are less than 60 dB SPL [6]. However, for ambient noise above 60 dB SPL, the transmitted noise rapidly becomes very objectionable even when the talking distance is a mere 1.5 feet [1].

2.1.4 Near-End Talker Echoes

Because of the reverberant character of a room, the direct sound and the near-end talker echoes are mixed and transmitted to the far-end talker. Maximum talking distance with no noticeable reverberation to the far-end talker has been reported by Emling [7].

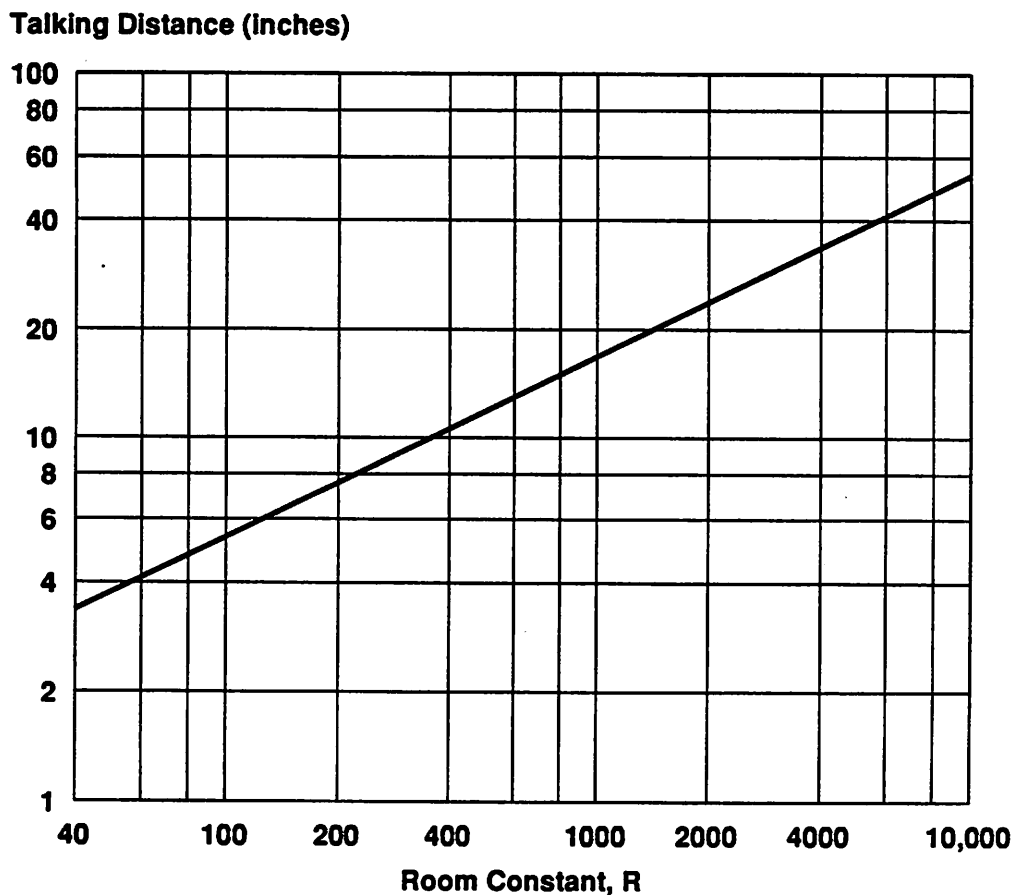


Figure 2.4 - Maximum talking distance without noticeable reverberation to the far-end talker

His result is shown in Figure 2.4, which describes the relation between the room constant (R)

and the talking distance at which reverberation becomes noticeable (the point at which the energy of the near-end talker echoes is 10 dB below the energy of the direct sound.) Room constant R is defined by

$$R = \frac{\bar{\alpha} S}{1 - \bar{\alpha}} \quad (2.3)$$

where $\bar{\alpha}$ is the average absorption coefficient of the reflecting surfaces and S is the total surface area in ft^2 . Therefore, the room constant reflects not only the acoustic properties of the room but also the size of the room. Rooms with smaller room constants are more reverberant. For most offices, the room constant ranges from 100 to 2000. For a perfect anechoic room, the room constant is ∞ .

The subjective perception of the transmitted reverberation depends on the relative delay between the direct sound and the echoes. For early echoes, the effect is predominantly spectral coloration (changing the speech spectrum) giving the speech a hollow quality (barrel effect.) The later echoes are typically due to multiple reflections from the walls, are weaker in magnitude, and are perceived as distinct echoes [8]. To illustrate the spectrum shaping due to the early echoes, consider a situation where there is only a single echo following the direct sound. Assume the sound source emits a signal, $s(t)$. The microphone picks up the direct sound and a reflection of the sound delayed by T seconds and attenuated by a factor of $e^{-\alpha T}$ (more attenuation loss for longer delays.) The received signal, $s_m(t)$, is given by

$$s_m(t) = s(t) + e^{-\alpha T} s(t-T) \quad (2.4)$$

The Fourier transform of the received signal is

$$S_m(\omega) = S(\omega) (1 + e^{-\alpha T} e^{-j\omega T}) \quad (2.5)$$

The spectrum of the source signal is distorted by the filter factor $1 + e^{-\alpha T} e^{-j\omega T}$. Figure 2.5(a) shows an average long-time power density spectrum for continuous speech by a group of male speakers [9]. The power density spectrums of the received signal (equation (2.5)) are shown in Figures 2.5(b) and 2.5(c) for two values of T . For smaller values of T (early echoes), the

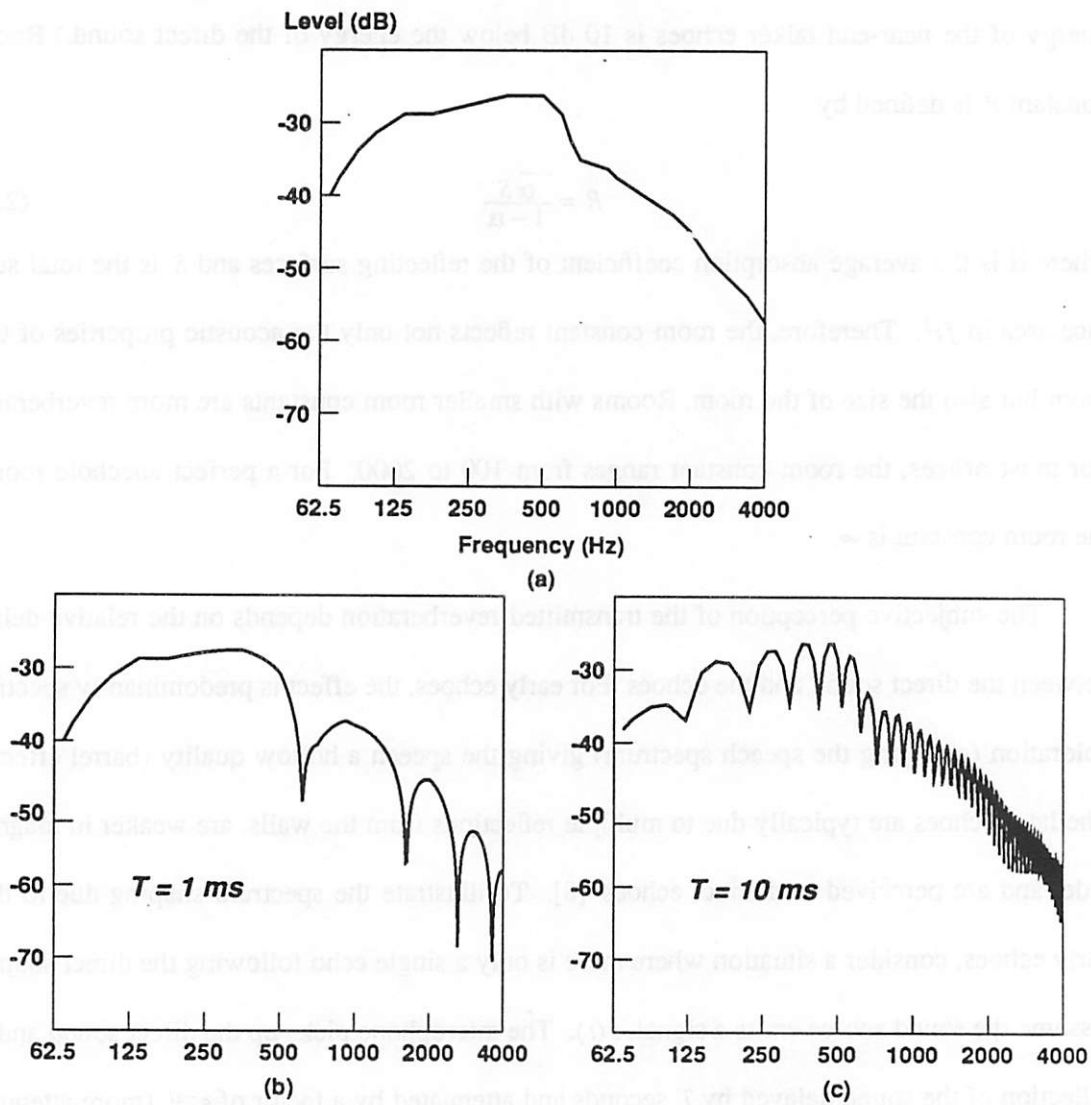


Figure 2.5 - Speech spectrum: (a) long-time average, (b) & (c) with single reflection

separation between peaks and valleys are wider, but the depths are deeper. For larger values of T (later echoes), the peaks and the valleys are denser and shallower. Because the later echoes result from multiple reflections, they usually cluster together. Therefore, the spectrum shaping due to later echoes tends to average out. It is the distinct early echoes that contribute more to the spectral coloration of the received signal. The perceptual effect of this distortion is to produce a hollow-sounding speech.

2.2 Voice-Switched Loudspeaker Telephones

A voice-switched loudspeaker telephone continuously compares the transmitted and the received signals and produces higher gain in the channel (transmit or receive) with the stronger signal level [2, 10, 11]. Substantial insertion loss is introduced in the idle channel, which essentially provides two one-way telephone circuits with only one of them being activated at a time. Therefore, the far-end talker echoes are eliminated and the singing margin is increased. If a communication channel allows only one direction of transmission (although the direction can be reversed) at any given time, the communication is in the "half-duplex" mode. The operation and the limitations of voice switching will be discussed in the next two subsections.

2.2.1 Principles of Operation

A simplified diagram of a voice-switched loudspeaker telephone is illustrated in Figure 2.6. A transmit variolosseser, TVL, and a receive variolosseser, RVL, provide the needed insertion loss depending on whether the circuit is transmitting or receiving. The control circuit decides the amount of loss to be introduced based on the relative levels of V_{T1} , V_{T2} , V_{R1} , and V_{R2} (threshold detection.) In other words, the introduction of insertion loss is switched between the transmitting channel and the receiving channel. This switching must be done in a smooth manner without noticeable clipping of the speech syllables, line noise, and room noise. Therefore, the control circuit is carefully designed for both its transient response and its steady-state response.

To account for the different levels of room noise, automatic variation in the switching threshold is necessary to avoid blocking the received channel. A noise level detection circuit is designed to give very little response to fluctuating signals. (Unlike noise, the speech signal

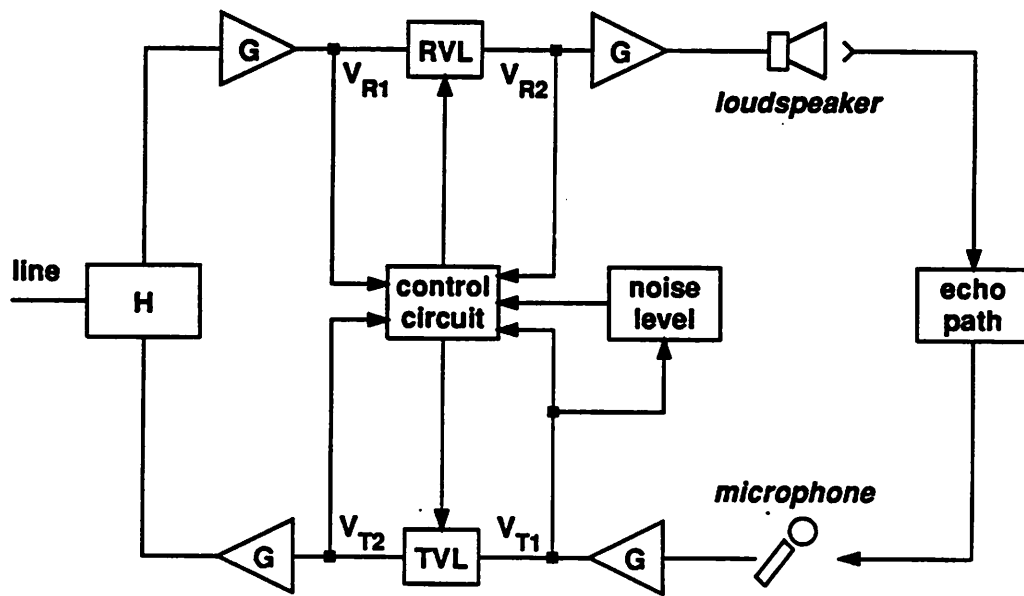


Figure 2.6 - Functional diagram of a voice-switched loudspeaker telephone

shows a rapidly fluctuating characteristics.)

2.2.2 Limitations

There are three key limitations in a voice-switched loudspeaker telephone: (a) half-duplex communication, (b) switching background noise, and (c) occasional incorrect switching.

The ability to talk and listen simultaneously is impaired because, in principle, voice-switched loudspeaker telephones are "half-duplex" systems, in which the line must be "turned around" each time signal goes the other way. Therefore, a loudspeaker telephone user is required to wait until the other party is completely finished before beginning to speak. Failure to wait usually results in cutting off the first part or the last part of what was said because the switching time cannot be made with perfect accuracy. Unfortunately, interruption is an integral part of a conversation. Brady [12] studied the on-off speech patterns in 16 experimen-

tal telephone conversations. Each conversation lasted about 7 minutes to 10 minutes. His results showed a 20% probability for one talker to interrupt the other. As a result, most people find this "half-duplex" conversation objectionable because of the frequent need of repeating.

Although the channel gains are switched in response to increases in speech energy, some duration of gain hangover after decreases in speech energy is needed. The speed of switching and the duration of hangover are usually designed to minimize clipping of the initial syllables, final syllables, and, sometimes, weaker syllables of a speech burst [2]. However, the switching is still objectionable because of the changes in background noise and room reverberation. The rise and fall of the transmitted background noise often results in the "swishing" effect.

If there is a sudden increase in the ambient noise level, the loudspeaker telephone may incorrectly switch direction causing an unintended interruption of the far-end talker. It is also possible that the room reverberation may incorrectly switch the direction of transmission whenever the far-end talker pauses, resulting in the far-end talker receiving a burst of room reverberation [11].

2.3 Echo Cancellation to Remove Far-End Talker Echoes

The operational impairments of singing and far-end talker echoes in loudspeaker telephones are all due to the acoustic coupling between the loudspeaker and the microphone. If we can quantitatively characterize the physical path between the loudspeaker and the microphone, we can electronically synthesize a replica of the far-end talker echo. The replica can then be subtracted from the signals picked up by the microphone and the acoustic coupling is effectively removed. The most common assumption in this technique is that the echo path from the loudspeaker to the microphone is linear and, therefore, completely specified by its impulse response. The validity of this assumption is supported by the fact that atmospheric pressure is roughly 194 dB SPL while the threshold of pain to human hearing is about 120 dB

SPL [13]. Consequently, at typical speech level, the nonlinearity of speech sound propagation through the air and into the microphone can be ignored.

2.3.1 Principles of Operation

Since discrete-time techniques are more suitable for integrated circuit implementation than continuous-time techniques, we only consider discrete-time echo cancelers here. The echo path of the far-end talker echoes is modeled as a linear system with sampled impulse response h_i . Given the speech samples, $x(n)$, the resultant far-end talker echo, $y(n)$, is

$$y(n) = \sum_{i=0}^{N-1} h_i x(n-i) \quad (2.6)$$

where the impulse response of the echo path is N samples long.

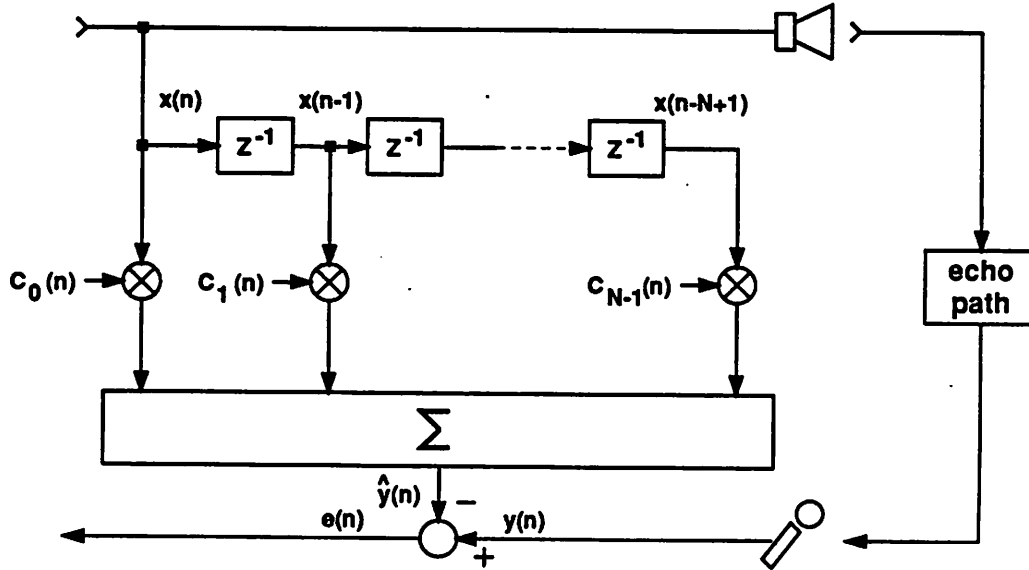


Figure 2.7 - Acoustic echo canceler to remove far-end talker echoes

An acoustic echo canceler is an adaptive transversal filter (Figure 2.7) that generates the replica of the far-end talker echo. The replica is then subtracted from the microphone output. If the coefficients of the transversal filter are the same as the sampled impulse response from

the loudspeaker to the microphone in a room, the generated replica will be the same as the far-end talker echo. The output of the transversal filter, which is the replica, $\hat{y}(n)$, is calculated from the following convolution sum:

$$\hat{y}(n) = \sum_{i=0}^{N-1} c_i x(n-i) \quad (2.7)$$

where c_i 's are the coefficients of the transversal filter. The coefficients are adapted by a feedback loop to match them with the sampled impulse response of the echo path. The residual far-end talker echo, $e(n)$, after cancellation is

$$e(n) = y(n) - \hat{y}(n) \quad (2.8)$$

2.3.2 Implementation Difficulties

The requirements of acoustic echo cancelers differ from those of other echo cancelers in that the acoustic echo impulse response is long (in the range of .2 to .3 seconds for typical rooms) and the dynamic range of input signals is large (40 dB or more).

A brute force implementation would require more than 1000 taps for typical office environment and 13 bits to encode data and coefficients. A straightforward design for a digital signal processor that would meet these specifications is too complex to be implemented in a single VLSI chip, at least in the near future.

In other parts of this thesis we will try to exploit the echo cancellation technique for removing the far-end talker echoes in loudspeaker telephones and to establish the engineering trade-offs between the performance and the complexity in the implementation of an acoustic echo canceler. Our goal is to show the feasibility of a single chip solution in implementing the acoustic echo canceler.

2.4 Equalization to Alleviate Near-End Talker Echoes

As pointed out in Section 2.1, there are two kinds of echoes in loudspeaker telephones: (a) the far-end talker echo and (b) the near-end talker echo. Although the generation mechanism is the same for both types of echoes, the echo cancellation technique used in the previous section (to reduce the far-end talker echoes) cannot be directly applied to the removal of the near-end talker echoes. For the far-end talker echoes, the source of origin is the far-end talker signal, which is available in a loudspeaker telephone unit. On the other hand, the near-end talker signal is already mixed up with the near-end talker echoes when it is picked up by the microphone. The echo cancellation technique described in the previous section, which needs the original signal to synthesize the echo replica, is not directly applicable to the elimination of near-end talker echoes because of the lack of a reference source (the direct sound of the near-end talker.)

Several methods to cope with the problem of near-end talker echoes will be reviewed in Section 2.4.1. These methods all require more than one input for reception (multimicrophones) and complex signal-processing algorithms. In Section 2.4.2, an inverse filtering scheme to equalize the near-end talker echoes is proposed. Some design issues are discussed but only the required long impulse response predictor (identical to the acoustic echo canceler) is realized in this research project.

2.4.1 Previous Work

It has been recognized that the perception of room echoes can be categorized into early echoes and later echoes [8]. The early echoes are generally due to single reflections from nearby objects surrounding the talker or the microphone and have the effect of shaping the speech spectrum (characterized by peaks and valleys) giving the speech hollow-sounding. The

later echoes usually experience multiple reflections from the walls of the room (therefore, weaker in sound level than the direct sound) and are perceived as distinct echoes. These characteristics in early echoes and later echoes are exploited by Flanagan et al. [14] and Allen et al. [8] to reduce the room echoes.

Multimicrophones with Polling

Flanagan's method requires two or more microphone inputs and is aiming at reducing early echoes. Because the early echoes cancel out some frequencies (corresponding to valleys in the received speech spectrum) depending on their relative delays, these specific frequencies which are canceled are different from one microphone input to another. Each microphone input is filtered by a filter bank occupying contiguous frequency ranges. (See Figure 2.8(a).) Within each frequency range, the microphone input that has the greatest energy is chosen for that frequency band. The microphone input that has the largest energy in a particular frequency range among all microphone inputs is least affected by the echoes that contribute to the nulls in that frequency band. Therefore, the combined output shows less coloration than any of the individual microphone input. (See Figure 2.8(b).) This method has been successful in situations that have only a few prominent echoes. However, its effectiveness is doubtful in a real room, which often contains later echoes. For an echo with 50 ms delay, the bandwidth for each filter in the filter bank would have to be less than 20 Hz, which would result in impractical number of filters for integrated circuit implementation.

Multimicrophones with Cross Correlation

An improved method was later presented by Allen et al. [8]. This method also requires two or more microphone inputs and each microphone input is again divided into several frequency bands using filter banks. Their assumptions are the early echoes received by the multimicrophones being correlated while the later echoes being uncorrelated. Within each frequency band, the microphone inputs are phase corrected (to account for the relative delay) and

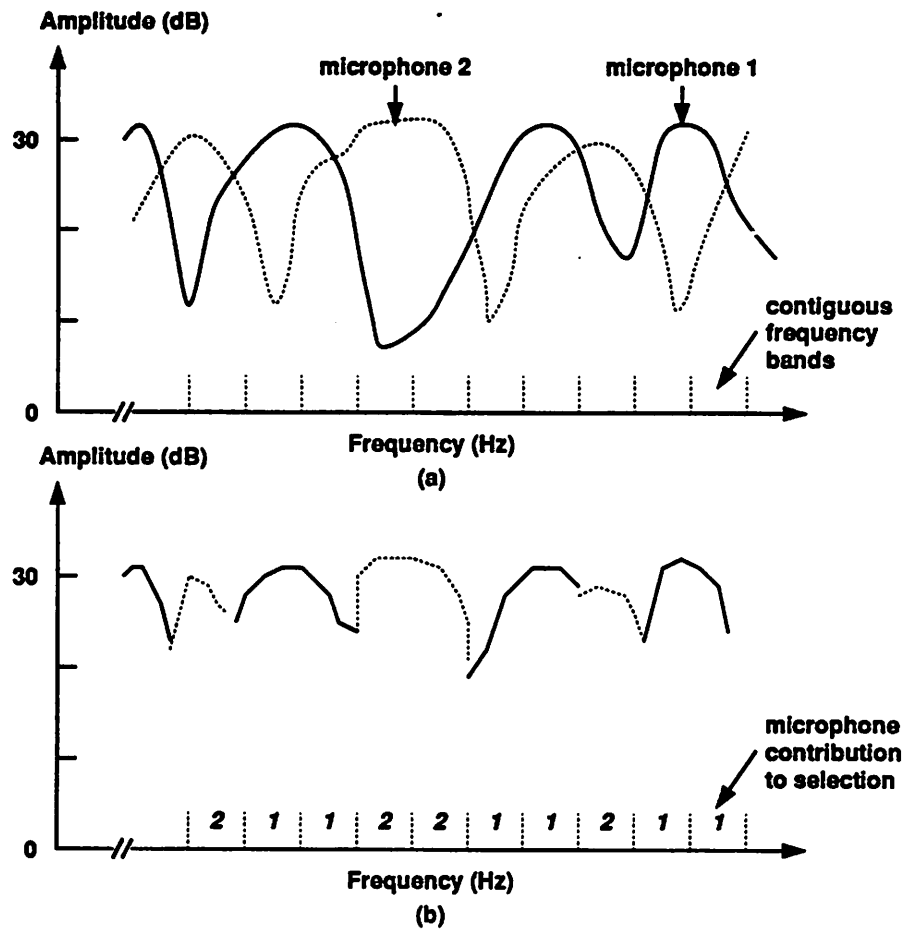


Figure 2.8 - Amplitude response of two microphones and the synthetic output

added to reduce the early echoes. This procedure is similar to the previous method (selecting the input with the greatest energy). After the "cophase and add", a gain adjustment is performed based on the cross correlation between the microphone inputs in that particular band. Therefore, the gain adjustment suppresses the later echoes (uncorrelated). Figure 2.9 outlines the process of this method. The results appear to be very effective, but the number of computations is quite substantial.

Microphone Array

Because the near-end talker echoes do not necessarily come from the same direction as

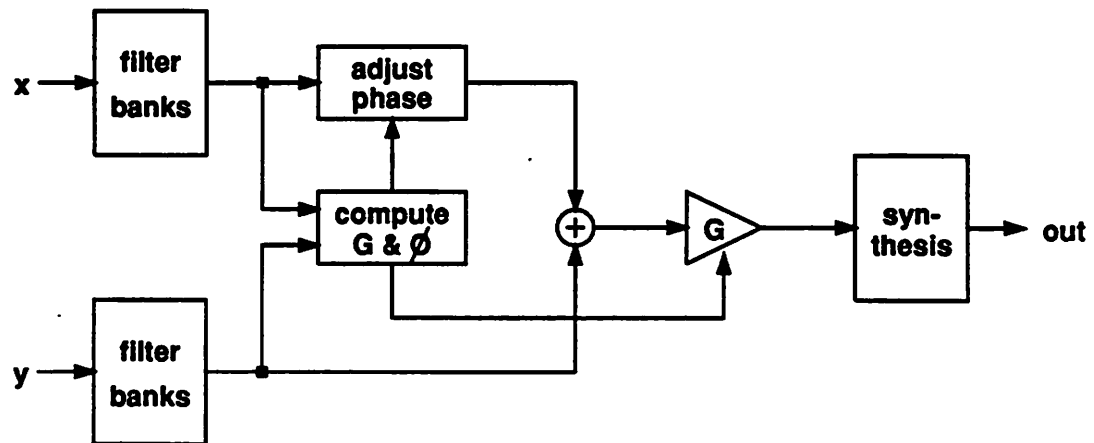


Figure 2.9 - Diagram of the signal processor (cross correlation to reduce echoes)

the direct sound, a steerable directional microphone can reduce the near-end talker echoes. A microphone array consisting of several microphones with variable delays provide a spatial discrimination on the reception of the incoming signal. (see Figure 2.10)

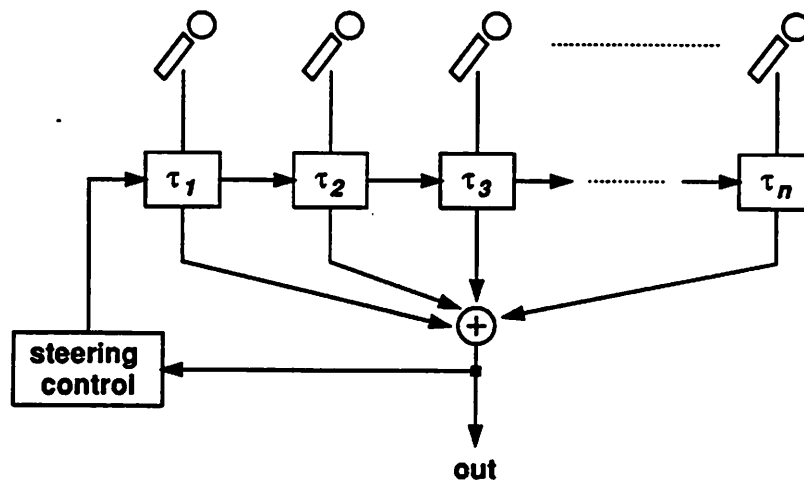


Figure 2.10 - Configuration of microphone array

Because signals arriving at the microphones may be in phase or out of phase depending on their relative delays, controlling the variable delay can steer the beam of reception to the

desired direction. Flanagan et al. have analyzed a linear microphone array for its beamwidth and usable bandwidth [15]. For signals in the frequency range of 300 Hz to 3300 Hz, eleven microphones with .22 ft separation are needed. Furthermore, a robust beam finder must be designed to capture the direction of the direct sound. The complexity of the microphone array system makes it more attractive for a high quality teleconference system than a low-cost loudspeaker telephone.

2.4.2 Adaptive Equalization

All the solutions cited in the previous section use two or more microphone inputs. For one microphone input, it takes some kinds of equalization to remove the near-end talker echoes. We believe that the use of training signals and the inverse filtering by exploiting the underlying model for speech production are two plausible approaches. In the following paragraphs, the concepts behind these two approaches will be discussed. Both of them rely on a common requirement - a long impulse response adaptive filter.

Use of Training Signals

The training signals can be transmitted by a small sound emitter. The loudspeaker telephone users are required to transmit the training signals when they first start the conversation or when they move to a new talking location. The transmitter (best positioned near the mouth) will send out a known training signal to the microphone. An adaptive equalizer will use this signal to adapt its coefficients. Figure 2.11 shows the arrangement of the adaptive equalization based on training. A pre-filter is needed to equalize the frequency response of the transmitter and the microphone. A desirable training signal is a binary pseudo-random sequence such as the modulo 2 division by the polynomial $1+X^3+X^{20}$ (Figure 2.12.) After the training period, the equalizer holds the coefficients until the next training takes place. The equalizer can be realized by an adaptive transversal filter.

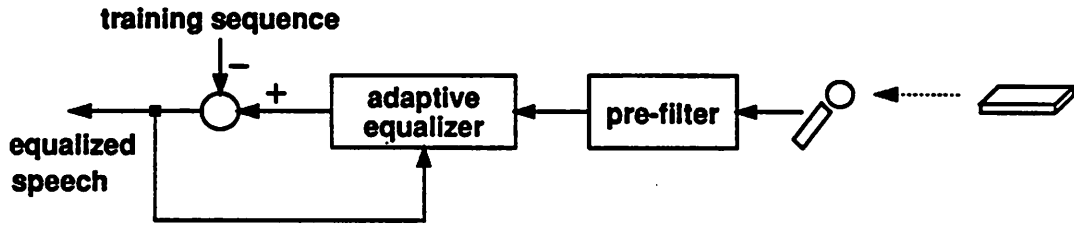


Figure 2.11 - Use of training sequence to adapt the equalizer

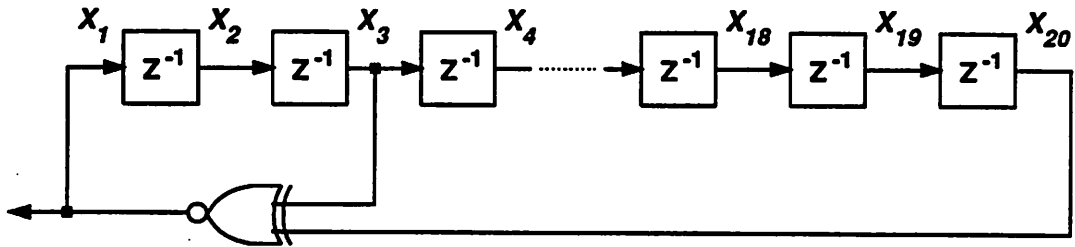


Figure 2.12 - Generation of pseudo-random sequence of length $2^{20} - 1$

Inverse Filtering Based on Speech Production

The purpose of equalization is to reverse the signals to the form before they are degraded by the transfer function between the talker and the microphone. However, the exact form of the anechoic signals is not known in advance because the direct sound has been mixed up with the echoes when they are picked up by the microphone. Fortunately, the model of speech production (Figure 2.13(a)) provides a good reference to the original form of the speech signals; namely, quasi-periodic pulses for the voiced sounds and white noise for the unvoiced sounds. The filter $H(z)$ models the function of the vocal tract. The desired information is contained in the gain G and the coefficients a_i , $1 \leq i \leq m$, where m is typically in the order of 10 to 15. The

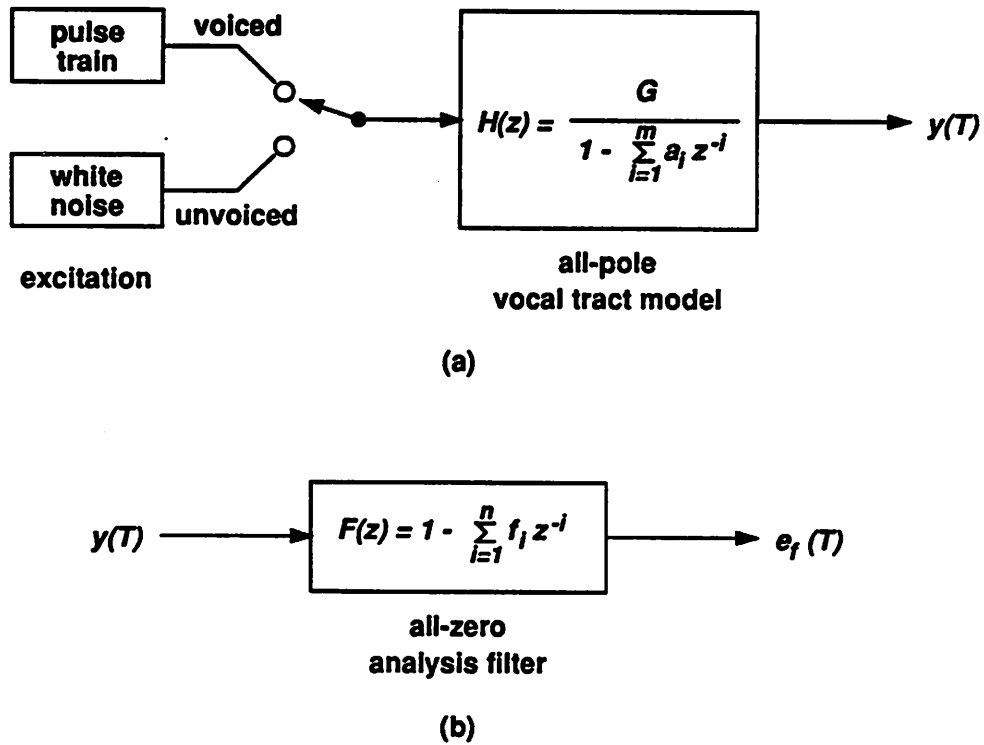


Figure 2.13 - LPC speech modeling and analysis

extraction of these coefficients (slowly varying with time) from the input speech waveform is the core of the linear predictive coding (LPC) analysis. Because the vocal tract filter is represented by an all-pole filter, the analysis filter is an all-zero filter (Figure 2.13(b)).

Figure 2.14 shows two voiced speech waveforms. Waveform (a) is recorded with the microphone placed close to the speaker such that the echoes are much smaller than the direct sound. Waveform (b) is obtained by convolving waveform (a) with an impulse response representing the transfer function between the microphone and the speaker (separated by 2 feet) in a room of size 10x15x12.5 feet. The conjecture is that at the beginning of voiced speech (preceded by silence or unvoiced speech), the contribution of echoes from the prior sounds is minimal. Therefore, the pitch period and the LPC spectrum (that of the vocal tract

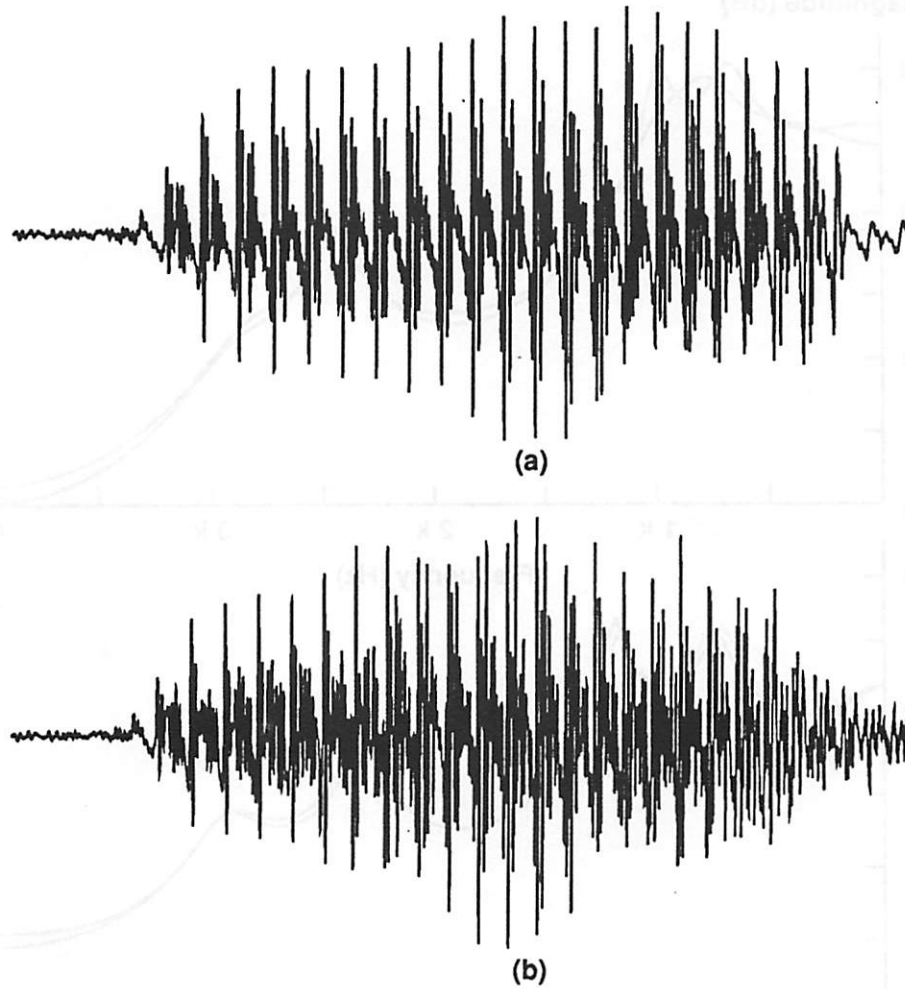


Figure 2.14 - Speech waveform: (a) anechoic, (b) reverberant

filter) can be accurately estimated. (See Figure 2.14(b) for distinguished pitch pulses in the beginning.) After that, although the speech waveform is "contaminated" by echoes, the LPC spectra remain similar between the anechoic speech and the reverberant speech. An LPC analysis on the speech waveforms shown in Figure 2.14 is performed at two locations: one at the mid-points of the waveforms and the other at approximately a quarter total length from the beginning. The size of the windows in the LPC analysis is 20 ms and the order of the filter is 12. The LPC spectra of the anechoic and the reverberant waveforms in these two windows are

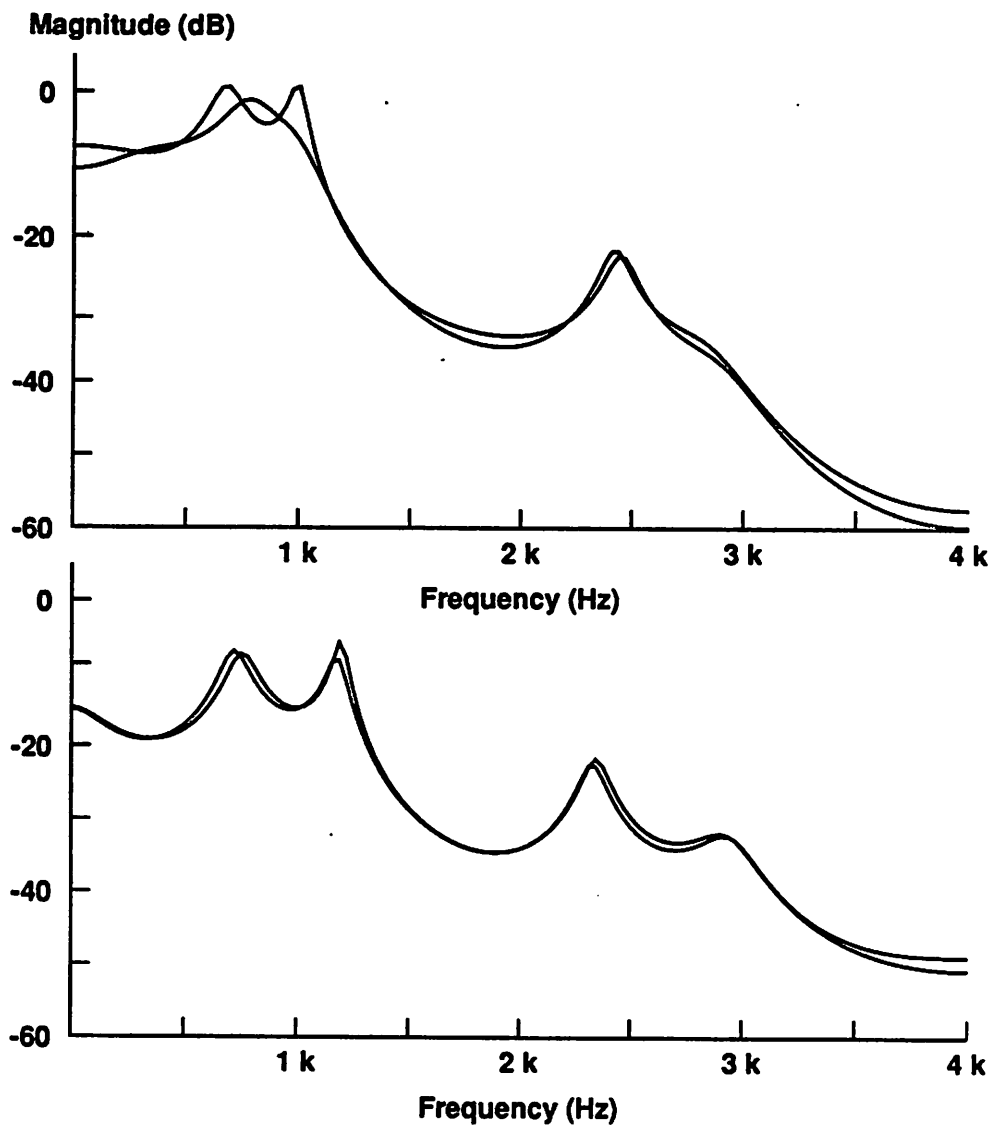


Figure 2.15 - Spectra of LPC analysis

shown in Figure 2.15. The close resemblance of these LPC spectra suggests that the vocal tract filter coefficients can still be extracted even in a reverberant environment.

Based on these conjectures, an inverse filtering to equalize the transfer function between the microphone and the speaker is plausible. An equalization system is proposed in Figure 2.16. The speech production is illustrated by an excitation source (pulse train for voiced

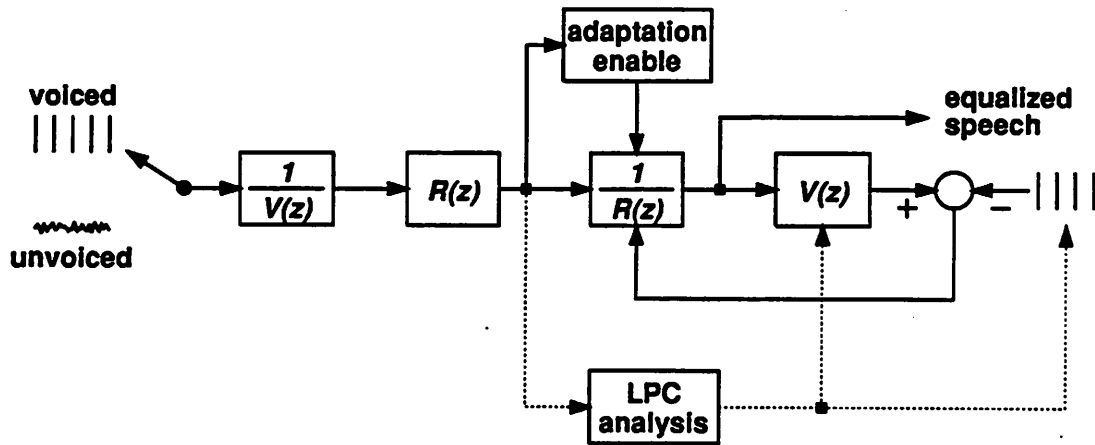


Figure 2.16 - Adaptive inverse filtering based on LPC analysis

speech and white noise for unvoiced speech) and an all-pole vocal tract filter. The transfer function between the speaker and the microphone is represented by a filter, $R(z)$. The LPC analysis extracts the filter coefficients and the pitch period. Because the early part of voiced speech is less affected by echoes, the pitch information estimated in this period is used in the complete voiced sound but no more than a certain limit to account for the change in pitch period due to a new sound. If the adaptive filter, $\frac{1}{R(z)}$, is exactly the inverse of the room filter, $R(z)$, the resultant signal after $V(z)$ will be either a pulse train or white noise. The inverse filter is implemented by an adaptive transversal filter to assure stability. The coefficients of the transversal filter are adapted by a feedback loop to minimize the error. Adaptation is allowed only when there is indisputable estimation on the pitch period. Best results are obtained at the beginning of a sentence or a word.

There are many questions to be answered before the system is usable. For example, a quantitative criterion for enabling the adaptation is needed. If, after further study, the above scheme is useful, a long impulse response adaptive transversal filter is required. Therefore, for

the moment, this research is concentrated on the acoustic echo canceler (a long response adaptive filter) to remove the far-end talker echoes.

CHAPTER 3

REQUIREMENTS OF ACOUSTIC ECHO CANCELERS

The requirements of acoustic echo cancelers differ from those of other echo cancelers in that the acoustic echo impulse response is long (in the range of .2 to .3 seconds for typical rooms) and the dynamic range of the input signals is large (40 dB or more). To efficiently design an acoustic echo canceler, a careful examination of major design parameters is necessary. Within this context, the length of the process window (related to number of taps) and the amount of echo reduction are two critical criteria. The process window is a time frame within which the echoes can be eliminated. It is determined by the reverberation time of the room where a loudspeaker telephone is used. As for the required echo reduction, it depends on the signal levels of both the near-end talker and the far-end talker. Section 3.1 briefly reviews the sound propagation in a room. The results are characterized by the reverberation time (T_{60}) and an exponentially decayed average impulse response. Section 3.2 uses these results to set up a criterion of choosing appropriate length for the process window. Section 3.3 examines both the objective and the subjective factors in the selection of the required echo reduction.

3.1 Room Acoustics of Sound Propagation

The sound propagation in a room follows the law of the wave equation (subject to certain boundary conditions):

$$\nabla^2 P = \frac{1}{c^2} \frac{\partial^2 P}{\partial t^2} \quad (3.1)$$

However, a solution to equation (3.1) is usually complicated and lends little help in establishing the requirement of the process window of an acoustic echo canceler. An image method

has been proposed and shown to be the same as an exact solution for a lossless rectangular room [16, 17].

3.1.1 Image Model for Sound Reflections

A single point source of sound in free space emits a sound wave whose pressure at any location is inversely proportional to the distance from the point source. For sound propagation in a room, because the normal velocity of a rigid wall is zero, the boundary condition can be satisfied if an image source (relative to the rigid wall) is placed symmetrically at the other side of the wall. Therefore, the sound reflection from a rigid wall is similar to the optical reflection from a mirror, where a symmetrical image is established. Like the optical reflection in a rectangular room with its six walls covered by mirrors, each image is itself imaged. The image method calculates the transfer impulse response from one point to another in a room by exciting the point source and all its images simultaneously.

Consider a room of size L_x , L_y , and L_z . The talker is at location (x, y, z) and the microphone (assumed to be an ideal omnidirectional point receiver) is at (x', y', z') . Each reflected sound wave is considered as emitted by an image source at:

$$R_{p,l,m,n} = (2l L_x \pm x, 2m L_y \pm y, 2n L_z \pm z)$$

where l , m , and n are integers such that $-\infty < l, m, n < \infty$ and p represents the eight possible permutations over \pm (see Figure 3.1.) The transfer impulse response from the talker to the microphone is:

$$h(t) = \sum_{p=1}^8 \sum_{l,m,n} \frac{\delta(t - \frac{d_{p,l,m,n}}{c})}{4\pi d_{p,l,m,n}} \quad (3.2)$$

where $d_{p,l,m,n}$ is the distance between the microphone and the image source $R_{p,l,m,n}$ and c is the speed of sound.

If the walls of a room are not rigid, only portions of the incident sound wave will be

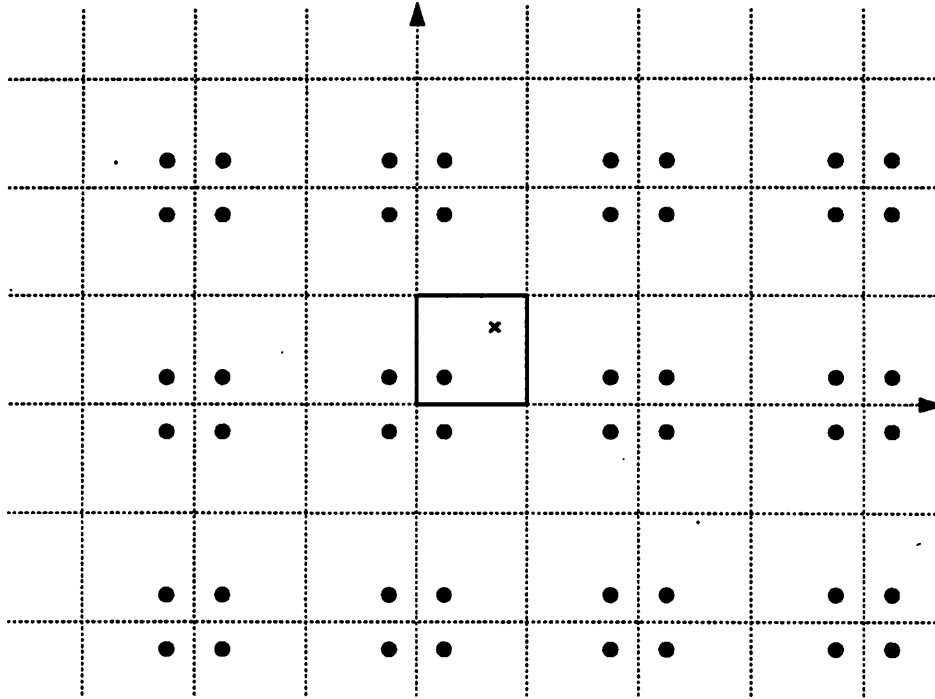


Figure 3.1 - Two-dimensional images expansion. The solid box is the original room.

reflected by the walls. A wall reflection coefficient β can be introduced to account for the loss in pressure magnitude after each sound reflection. Therefore, equation (3.2) becomes:

$$h(t) = \sum_{p=1}^8 \sum_{l,m,n} \beta_{x1}^{|l-r(p)|} \beta_{x2}^{|l|} \beta_{y1}^{|m-s(p)|} \beta_{y2}^{|m|} \beta_{z1}^{|n-t(p)|} \beta_{z2}^{|n|} \frac{\delta(t - \frac{d_{p,l,m,n}}{c})}{4\pi d_{p,l,m,n}} \quad (3.3)$$

where $r(p)$, $s(p)$, and $t(p)$ are either 0 or 1 depending on the permutation p . The β 's are the pressure reflection coefficients for the six walls with the index 1 referring to the walls adjacent to the coordinate origin and the index 2 to the opposing walls. The superscripts of the β 's denote the number of reflections by that particular wall.

3.1.2 Echo Response In the Average Sense

In geometrical room acoustics, the concept of a sound wave is replaced by the concept of a sound ray. Using the image model, the sound source and its images generate impulses of

equal strength at the same time. In the time interval from t to $t+dt$, the received sound reflections are generated by mirror images whose distances from the microphone are between ct and $c(t+dt)$. Therefore, the mirror images are located in a spherical shell with a radius ct and a thickness $c dt$ (see Figure 3.2.)

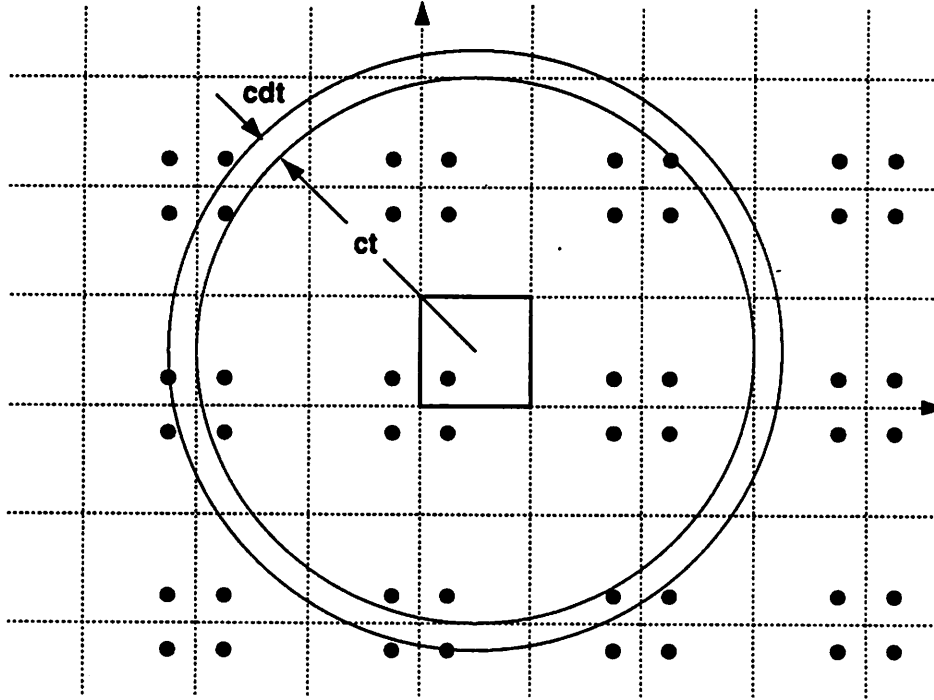


Figure 3.2 - Mirror sound sources for a rectangular room. The solid box is the original room.

Because there is one mirror image per room volume (V), the number of mirror images that contribute to the received sound reflections between time t and time $t+dt$ is

$$dN_r = \frac{4\pi (ct)^2 c dt}{V} \quad (3.4)$$

If the average number of wall reflections per second is \bar{n} and the absorption coefficient of sound intensity is α ($\alpha = 1 - \beta^2$), these reflections will attenuate by

$$(1-\alpha)^{\bar{n}} = e^{\bar{n} \ln(1-\alpha)} \quad (3.5)$$

Because the sound pressure decreases proportionally to the distance (equation (3.3)), the

sound intensity (energy) decreases proportionally as $(ct)^{-2}$. The received sound intensity between time t and time $t+dt$ is (neglecting the attenuation by absorption in air)

$$dE(t) = dh^2(t) = \frac{dN_r}{(4\pi ct)^2} = \frac{cdt}{4\pi V} e^{-\bar{n} \ln(1-\alpha)} \quad (3.6)$$

Therefore, the sound intensity at time t is

$$E(t) = E_0 e^{-\bar{n} \ln(1-\alpha)} \quad (3.7)$$

Equation (3.7) demonstrates that the average trajectory of sound intensity following an impulsive sound source is an exponentially decayed curve. The average number of wall reflections per second, \bar{n} , is shown to be [16]

$$\bar{n} = \frac{c}{2} \left(\frac{1}{L_x} + \frac{1}{L_y} + \frac{1}{L_z} \right) = \frac{cS}{4V} \quad (3.8)$$

where S is the surface area of a rectangular room.

3.2 Reverberation Time and Process Window

Reverberation time is a parameter commonly used in the acoustic design of rooms. Figure 3.3 shows a typical sound pressure level in dB after the sound source is turned off. The nearly straight line decay is due to the approximately constant loss in energy after each sound reflection. Reverberation time, T_{60} , is the time interval in which the reverberation level drops down by 60 dB. It is a function of the room size and the materials inside the room.

The relationship between the reverberation time and the room dimensions can be derived from equations (3.7) and (3.8).

$$\frac{E(T_{60})}{E_0} = 10^{-6} = e^{-\frac{cS}{4V} T_{60} \ln(1-\alpha)} \quad (3.9)$$

$$T_{60} = \frac{-24V \ln 10}{cS \ln(1-\alpha)} = \frac{49V}{-S \ln(1-\alpha)} \quad (3.10)$$

where T_{60} is the reverberation time in ms , V is the volume of a room in ft^3 , S is the total surface area in ft^2 , α is the absorption coefficient, and 49 is a lumped constant with unit in $\frac{ms}{ft}$. If the absorption coefficients are different for different walls, an average absorption coefficient,

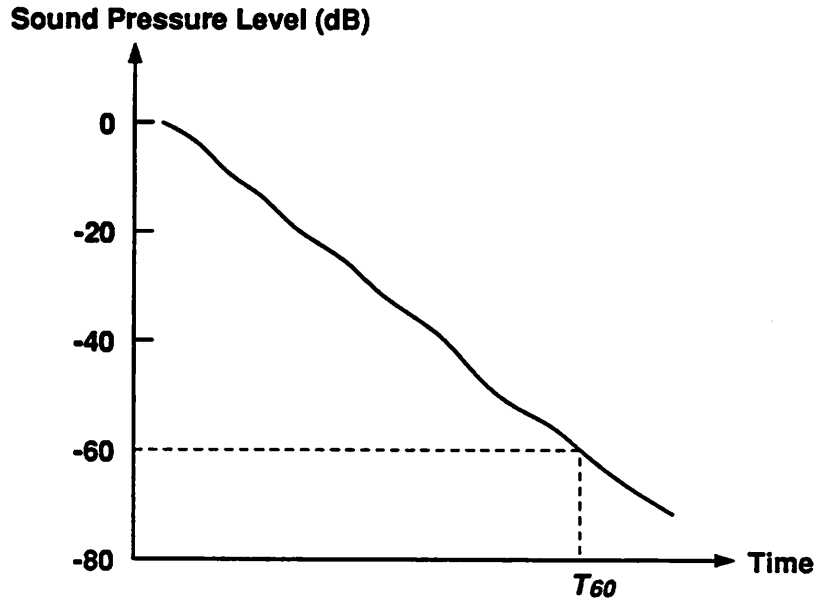


Figure 3.3 - Sound pressure level and reverberation time

$\bar{\alpha}$, should substitute α in equation (3.10).

The linear dimension of a typical office is about 10 to 15 feet and the average absorption coefficient is over 0.35 (examples of absorption coefficients for typical materials shown in Table 3.1) [18]. Therefore, the reverberation time is about 200 to 300 ms. For an office with lots of furniture, the surface area will be larger and the reverberation time shorter. The exponential law of sound reflections (equation (3.7)) can be simplified to

$$E(t) = E_0 e^{-at} \quad (3.7a)$$

The fraction of echo energy from time 0 to time τ is

$$\frac{\int_0^{\tau} E(t) dt}{\int_0^{\infty} E(t) dt} = 1 - e^{-a\tau} \quad (3.11)$$

To get a 30 dB echo reduction (the total energy of the tail echoes is no more than 30dB), $e^{-a\tau} = 10^{-3}$. By definition, $e^{-aT_{60}} = 10^{-6}$. Consequently, we need a process window about half of

Table 3.1 Absorption Coefficients for Typical Materials

Materials	Absorption Coefficients			
	.5 kHz	1 kHz	2 kHz	4 kHz
plate glass	0.04	0.03	0.02	0.02
heavy carpet on concrete	0.14	0.37	0.60	0.65
medium weight velour drape	0.49	0.75	0.70	0.60
ceiling tile mounted to hard surface	0.56	0.70	0.68	0.50
ceiling tile hung on suspension system (16" air space)	0.65	0.75	0.72	0.55
moderately upholstered chairs (0.90m x 0.55m)	0.67	0.74	0.83	0.87
mineral wool blanket (2" thick) mounted with 1" air space	0.85	0.86	0.87	0.87
fiber glass (4" thick) mounted to hard surface	0.98	0.97	0.93	0.88

the reverberation time ($\tau = \frac{T_{60}}{2}$), that is about 100 to 150 ms, to reduce the echoes by 30 dB.

A process window of 125 ms (assuming the surface area is 20% larger than that of an empty office) would require a 1000 tap transversal filter at 8 kHz sampling rate.

3.3 Sound Pressure Levels and Echo Reduction

Figure 3.4 shows the typical sound pressure levels in a comfortable telephone conversation using a loudspeaker telephone [2]. The received far-end talker signals are approximately 74 dB SPL. SPL is a relative sound pressure level in dB compared to a fixed standardized reference level which is roughly the hearing threshold of human ears at 1000 Hz (2×10^{-5} Newton per square meter.) The returned far-end talker echoes are about 70 dB while the near-end talker signals are from 55 dB to 70 dB depending on the distance between the talker and the

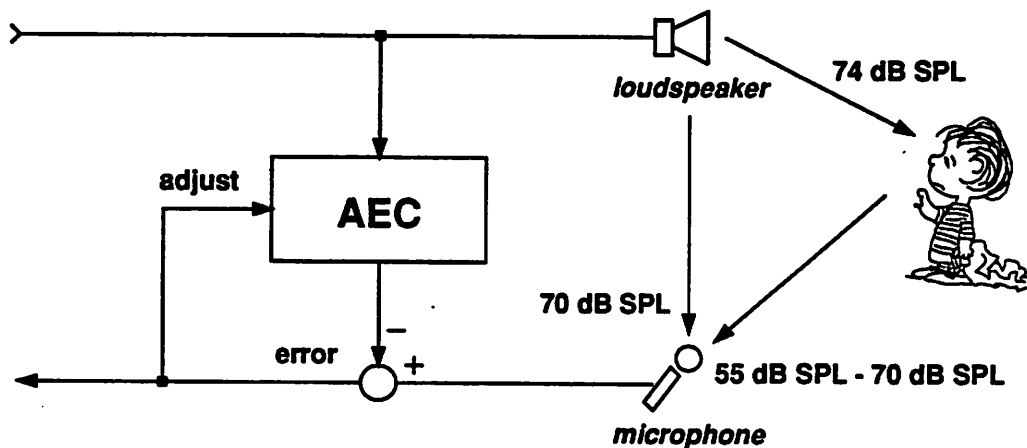


Figure 3.4 - Sound pressure levels in a loudspeaker telephone connection

microphone. The 70 dB level corresponds to a talking distance about 1.5 feet; 55 dB corresponds to 9 feet. In other words, the returned far-end talker echoes may be 15 dB higher than the near-end talker signal.

Experiments to find the critical level of echo reflections were carried out by Haas [16] using continuous speech as a sound signal. This signal was broadcast by two loudspeakers. One of them deliberately attenuated and delayed the speech signal (to simulate sound reflections.) Figure 3.5 shows the percentage of observers who felt disturbed by an echo of given relative level versus the delay time between the undelayed speech and the delayed one [16]. The numbers next to the curves indicate the relative level (in dB) of the delayed speech. The rate of speech was 5.3 syllables per second. At the relative level of -10 dB, the percentage of annoyance is less than 2% regardless of the delay time. Assume the far-end talker speaks at the same sound pressure level as the received near-end talker signal. The near-end talker signal level must be at least 10 dB higher than the returned far-end talker echoes to avoid any annoying effects. Therefore, the echo canceler must be able to reduce the far-end talker echoes by 25 dB.

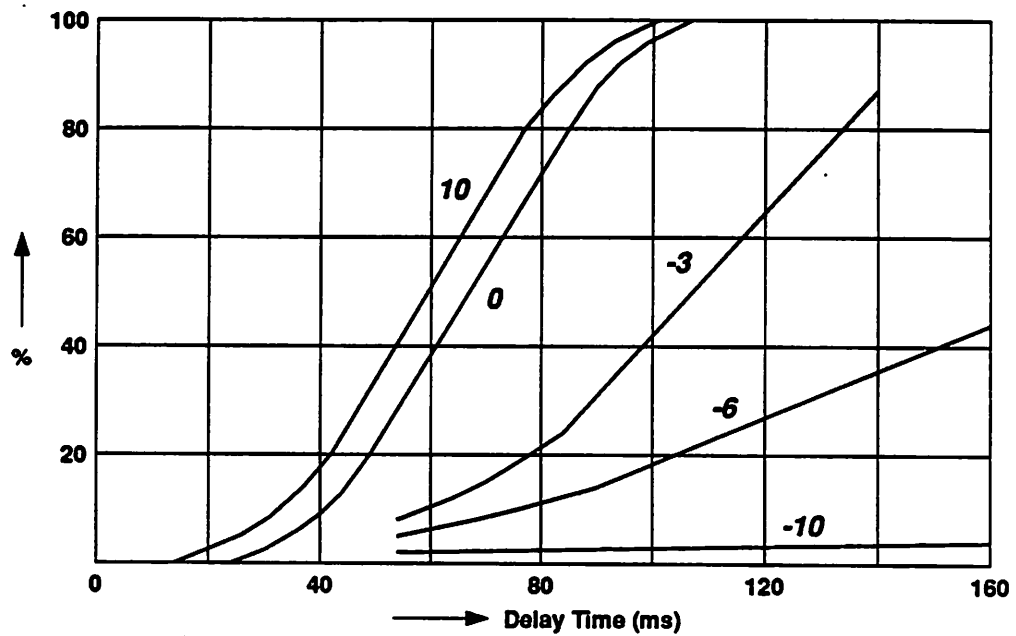


Figure 3.5 - Percentage of listeners disturbed by the delayed speech signal

Table 3.2 summarizes the specifications of the acoustic echo canceler. The *echo return loss enhancement* (ERLE) measures the reduction in echo energy.

Table 3.2
Specifications of Acoustic Echo Cancelers

Design Specifications	
Process Window	125 ms
ERLE	25 dB
Dynamic Range	40 dB

CHAPTER 4

DESIGN CONSIDERATIONS: A SYSTEM PERSPECTIVE

A straight forward design of an acoustic echo canceler that would meet the specifications outlined in the previous chapter is too complex to be implemented in a VLSI chip in the near future. The requirements of long echo impulse response and wide dynamic range of input signals result in large memory for data storage and wide word length for computations. These two demands have greatly increased the complexity of the implementation. Fortunately, the required echo reduction is only 25 dB. As a result, trade-offs can be exploited between the performance and the complexity. In Section 4.1, the computation and the storage requirements of a direct realization will be examined first. Quantization effects, particularly floating point coding, will be discussed in Section 4.2. A modified LMS adaptation algorithm from an implementation point of view will be presented in Section 4.3. A scheme with balanced design trade-offs is proposed in Section 4.4. Computer simulations and results of this scheme are discussed in Section 4.5. The presence of a near-end talker is detected based on the expected ERLE (Section 4.6.) Alternative implementations are considered in Section 4.7. The effects of imperfections in A/D (or D/A) converters are analyzed in Section 4.8.

4.1 A Brute Force Approach

Figure 4.1 shows an acoustic echo canceler using a transversal filter. The far-end talker echo originates from the loudspeaker. A transversal filter with a tap delay line generates the replica of the far-end talker echo. If the coefficients of the transversal filter are the same as the

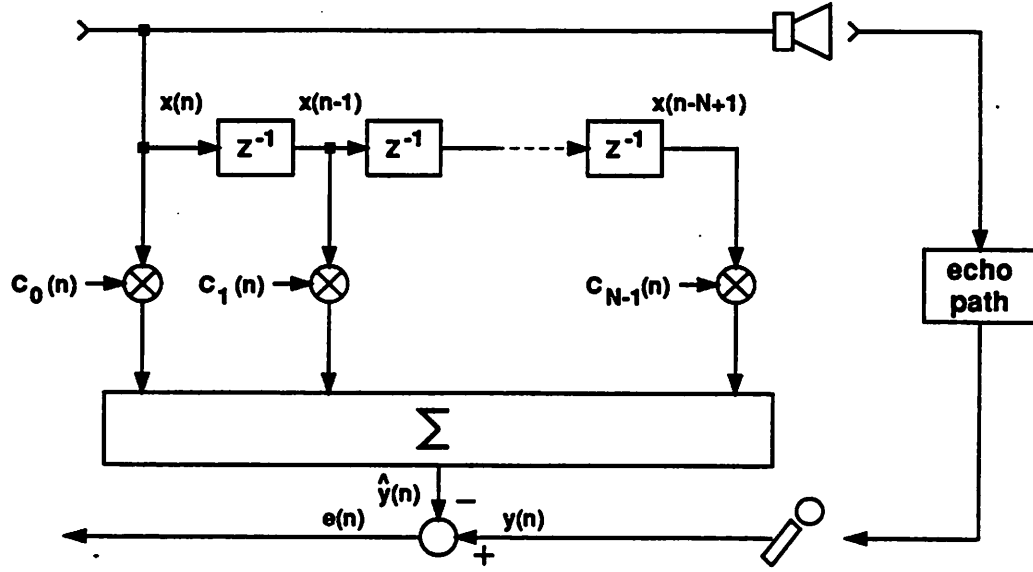


Figure 4.1 - Acoustic echo canceler to remove far-end talker echoes

sampled impulse response from the loudspeaker to the microphone in a room, the generated replica will be the same as the far-end talker echo. Because the adaptation is more easily realized in the digital domain, a digital echo canceler is chosen here.

The output of the transversal filter is computed by a convolution sum:

$$\hat{y}(n) = \sum_{i=0}^{N-1} c_i x(n-i) \quad (4.1)$$

where c_i 's are the coefficients of the transversal filter. The residual far-end talker echo, $e(n)$, after cancellation is

$$e(n) = y(n) - \hat{y}(n) \quad (4.2)$$

The coefficients are adapted by a feedback loop to match them with the sampled impulse response of the echo path.

The most commonly used adaptation algorithm is the *least mean square (LMS)* gradient algorithm because of its simplicity in hardware implementations. The coefficients are updated by the following equation [19]:

$$c_i(n+1) = c_i(n) + \frac{\beta}{K + \sum_{k=0}^{N-1} x^2(n-k)} e(n) x(n-i) \quad (4.3)$$

β controls the amount of adjustment allowed in each cycle. The summation estimates the total input signal power, and β is normalized to the estimated power. Without the normalization, when the input signal power grows too large, the adaptation may become unstable due to the large amount of adjustment in each cycle. Of course, this can also be accomplished by reducing β , but that will slow down the adaptation. K is a constant added to prevent the effective β from growing too large when the input signal power is small.

A brute force implementation of the echo canceler would require 13 bits to encode the data and the coefficients since the dynamic range of the input signal is 40 dB. The number of taps needed is 1000 to cover the 125 ms process window. A direct realization using the LMS adaptation algorithm would require 4 multiplications of 13 x 13 for each data sample. Because the sample rate is 8 kHz and 1000 cycles are needed for the computation of 1000 taps, the minimum clock rate is 8 MHz. In other words, either a multiplier of 13 x 13 operating at 32 MHz rate or 4 multipliers at 8 MHz rate are required in the direct realization. Neither of these is feasible in a single chip in the near future. As a result, trade-offs must be exploited if a single chip solution is to be sought. The Texas Instruments TMS32020 digital signal processor has been used to implement a 128 tap echo canceler with a small margin [20].

4.2 Quantization Effects

An obvious implication of the requirements of 40 dB dynamic range and 25 dB echo reduction is the choice of an optimum word length for the data and the coefficients. As pointed out, if the coefficients of the transversal filter are the same as the impulse response of the echo path, the generated echo replica will be the same as the echo. However, in the actual implementation, only finite number of bits are available to represent the data and the coefficients; quantization effects set an ultimate limit on performance. Assume the coefficients

are the same as the sampled impulse response but are subject to quantization. The data are also properly quantized by an *analog-to-digital* (A/D) converter. The resultant echo replica, $\hat{y}(n)$, is

$$\begin{aligned}\hat{y}(n) &= \sum_{i=0}^{N-1} (h_i + \Delta h_i) \left[x(n-i) + \varepsilon_x(n-i) \right] + \varepsilon_{da}(n) \\ &= y(n) + e(n)\end{aligned}\quad (4.4)$$

where Δh_i , ε_x , and ε_{da} are the quantization errors of the coefficients, the data, and the *digital-to-analog* (D/A) converter respectively. Therefore, the residual echo, $e(n)$, which is the difference between the echo and the echo replica is

$$e(n) \approx \sum_{i=0}^{N-1} h_i \varepsilon_x(n-i) + \sum_{i=0}^{N-1} \Delta h_i x(n-i) + \varepsilon_{da}(n) \quad (4.5)$$

Because of the requirement of 40 dB dynamic range, floating point format for the data and the coefficients can reduce the size of the memory and the complexity of the multiplier. The quantization error in a floating point representation is approximately equal to the product of the exact value and the quantization error in the mantissa. As a result, the variance of the residual echo is

$$\text{Var}[e(n)] = \sum_{i=0}^{N-1} h_i^2 \sigma_x^2 \frac{\delta_x^2}{3} + \sum_{i=0}^{N-1} h_i^2 \frac{\delta_h^2}{3} \sigma_x^2 + \sigma_{\varepsilon_{da}}^2 \quad (4.6)$$

where δ_x and δ_h are the step sizes of the mantissa of the data and the coefficients, σ_x^2 and $\sigma_{\varepsilon_{da}}^2$ are the variance of the data and the variance of the quantization noise in the D/A converter. Truncation quantization is assumed here. However, as we will see later, the adaptation of the coefficients eventually forces the resultant coefficients to be rounded (in which case the factor 3 will be replaced by 12). Because the mean of the residual echo is zero, the power of the residual echo is equal to its variance. The echo power is simply the product of the power of the data samples and the power of the impulse response. Therefore, the ERLE is (assuming the worst case that $\sum h_i^2 = 1$)

$$\text{ERLE} = \frac{\sum_{i=0}^{N-1} h_i^2 \sigma_x^2}{\text{Var}[e(n)]} = \frac{3}{\delta_x^2} + \frac{3}{\delta_h^2} + \frac{\sigma_x^2}{\sigma_{\varepsilon_{da}}^2} \quad (4.7)$$

Using this equation, the performance limit versus various numbers of mantissa bits for the data and the coefficients is plotted in Figure 4.2.

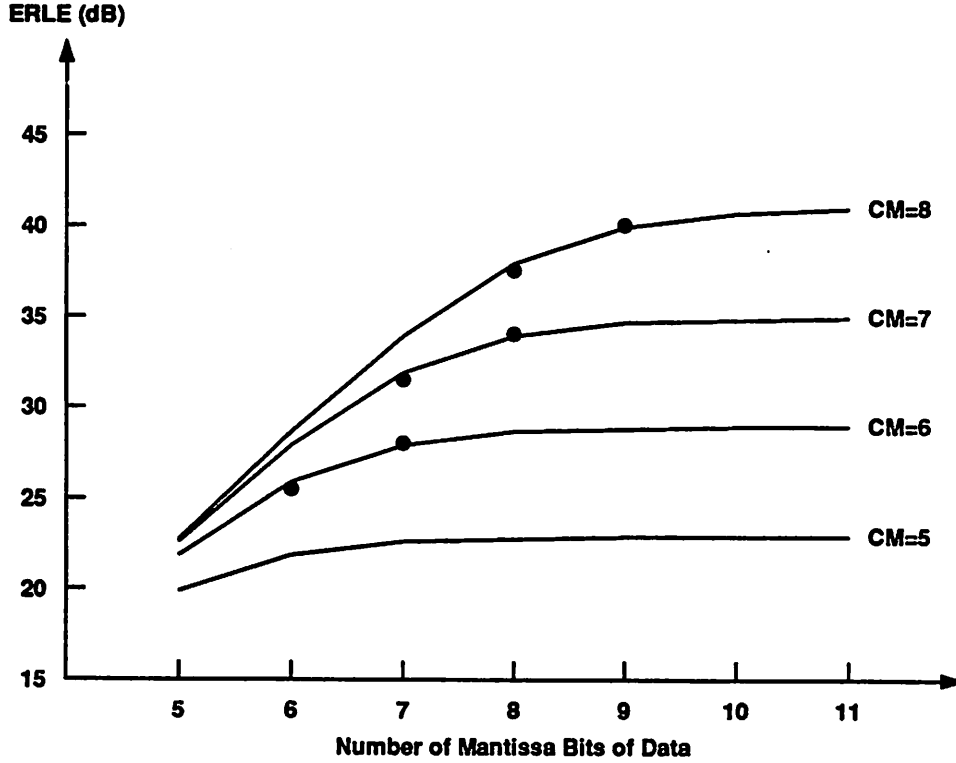


Figure 4.2 - Quantization effects on ERLE

The D/A converter is assumed to have the same dynamic range as the A/D converter, which overloads when the input signal is more than $4\sigma_x$ from the mean. For example, if a 6 bit mantissa (including a sign bit) and a 3 bit exponent are adopted, a 13 bit D/A is needed (including the sign bit.) As a result, $\delta_x = 2^{-5}$, $\delta_h = 2^{-5}$, and $\sigma_{\epsilon_x}^2 = \frac{(4\sigma_x)^2}{3}(2^{-12})^2$. The different curves are for different number of mantissa bits in the coefficients. The dots are obtained from computer simulations. The inputs in the computer simulations are Gaussian distributed. To get a 25 dB echo return loss, a 6 bit mantissa for the data and the coefficients is needed (sign bit included). A 6 bit mantissa is readily obtainable from μ -law quantization [21], which

will be further described in Section 4.4. A 3 bit exponent is chosen to meet the requirement of 40 dB dynamic range. Again, this is also available from μ -law quantization.

4.3 Adaptation Algorithm

After choosing the floating point representation for the data and the coefficients, its impact on the adaptation will be examined in this section. As pointed out, LMS gradient algorithm is widely used because of its simplicity in hardware implementations. The coefficients are updated by equation (4.3) and repeated here:

$$c_i(n+1) = c_i(n) + \frac{\beta}{K + \sum_{k=0}^{N-1} x^2(n-k)} e(n) x(n-i) \quad (4.3)$$

The total input signal power estimation (shown by the summation) can be rearranged as:

$$\begin{aligned} L(n) &= \sum_{k=0}^{N-1} x^2(n-k) \\ &= L(n-1) - x^2(n-N) + x^2(n) \end{aligned} \quad (4.8)$$

Therefore, the total input power estimation is calculated by deleting the oldest sample from the previous estimation and adding the newest sample. Because the data are in floating point format, the square can be approximated by multiplying the exponent by two, which is just a left shift [22]. Similarly, the computation of coefficient adjustment is also carried out by the power-of-two multiplications to simplify the hardware, which has been proven to have no significant effect in slowing down the adaptation [23]. β was chosen to be 1 so that the exponent of that term is zero.

Although each coefficient needs only 6 mantissa bits, internally it has 8 additional buffer bits. The purpose of the buffer bits, which are not used in the convolution computation, is to filter the noise resulting from the adaptation. Because each coefficient is in a floating point representation and the amount of adjustment is in the power-of-two format, the coefficient update is described by

$$\begin{aligned} c_i(n+1) &= c_i(n) \pm \Delta c_i(n) \\ &= m \cdot 2^e \pm 2^x \\ &= (m \pm 2^{x-e}) \cdot 2^e \end{aligned} \tag{4.9}$$

If the amount of adjustment is very small such that $x-e$ is less than 0, the adjustment is forced to be at least one LSB (2^1). On the other hand, if the adjustment is too large such that $x-e$ exceeds 12 (1 bit smaller than the internal coefficient bits), only a maximum adjustment of 1 MSB (2^{12}) is allowed. In other words, when adding the two floating point numbers, instead of lining up the smaller number with the larger number, the adjustment is always lined up with the coefficient by clipping the adjustment itself. The purpose is to avoid the need of converting from floating point to fixed point and then back to floating point during the coefficients update.

4.4 Hardware Implementation

Figure 4.3 is a simplified functional schematic of the acoustic echo canceler. The part inside the shaded area does the total input signal power estimation. The left shift (SHL) operating on the exponent squares the value of the data sample. When the newest sample comes in, the oldest sample is deleted at the time it is removed from the data memory. The block shown by Σ is an accumulator. The F2L performs a floating point to fixed point conversion to obtain the approximated power of an input signal. This is done by a barrel shifter. The L2F transforms the total input signal power estimation (in the fixed point format) into a two's power representation, which can be carried out by a leading one detector.

The interface is compatible with a μ -law quantizer. A segmented μ -255 encoder is chosen because it is a standard PCM coder and a conversion to a floating point format is easy to implement. The encoding formats of the segmented μ -255 and the floating point are shown in Figure 4.4. A μ -255 coded data is an 8-bit word divided into three fields. The first bit is a sign bit, being 1 for positive numbers and 0 for negative numbers. There are 8 segments for

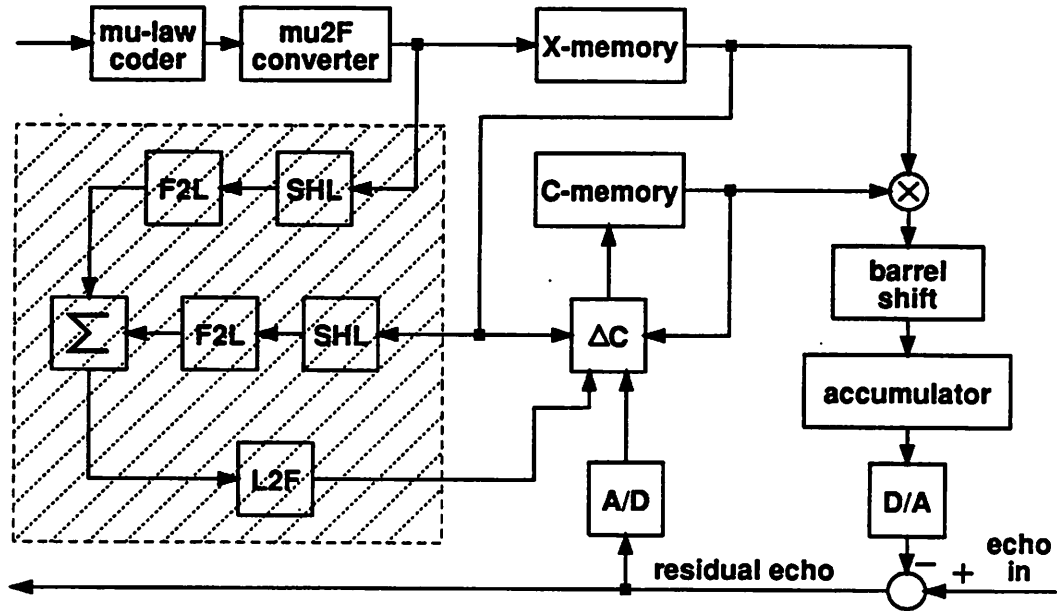


Figure 4.3 - Functional diagram of the acoustic echo canceler

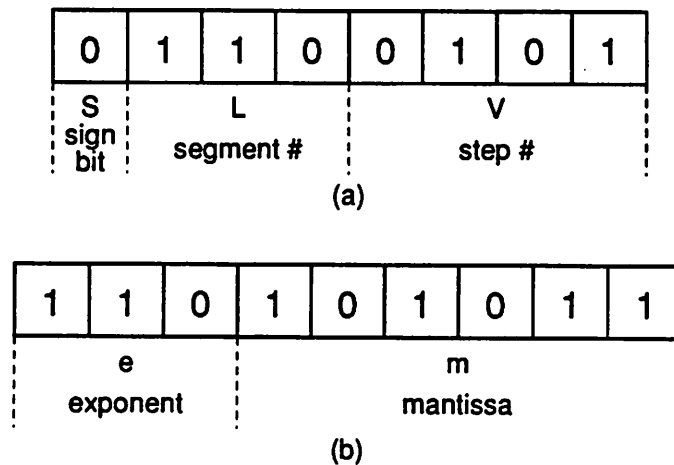


Figure 4.4 - Data formats of μ -255 and floating point

each polarity represented by the 3-bit segment field. Within each segment, there are 16 steps indicated by the 4-bit step field. The step size in a segment doubles as it moves from the lower

segment to the next higher segment. A floating point coded data is a 9-bit word with a 3-bit exponent and a 6-bit mantissa. The mantissa uses 2's-complement representation to facilitate the accumulation computation of convolution and the addition/subtraction of coefficients update. The equivalent linear code (L) of a μ -255 code can be computed by:

$$L = 2^L (V + 16.5) - 16.5 \quad (4.10)$$

Therefore, a μ -255 to floating point conversion can be implemented by the following equations:

$$|m| = V + 16.5 - \frac{16.5}{2^L} \quad (4.11)$$

$$e = L \quad (4.12)$$

To compute the convolution sum, a barrel shifter is needed to convert a floating point output into a fixed point format for the subsequent accumulation. The barrel shifter is also time-shared to compute the input signal power estimation. The size of the multiplier is 6x6 bits (including a sign bit).

Table 4.1 compares the hardware requirements of the proposed method and a direct approach. In the direct approach, a fixed point data representation is used and all the required multiplications are performed.

4.5 Computer Simulations

In the computer simulations, an exponentially decayed impulse response for the echo path was assumed. To avoid generating the impulse response of a single pole filter, the exponential sequence was multiplied by a binary pseudo-random sequence, $r(i)$.

$$h_i = r(i) \frac{K}{(-1.00346)^i} \quad 0 \leq i < 1000 \quad (4.13)$$

K is a constant such that $\sum h_i^2 = 1$. Both Gaussian inputs and speech inputs were used to simulate the proposed acoustic echo canceler. In the case of speech inputs, a 1.5 seconds binary pseudo-random training sequence was used to speed up the initial convergence. This training sequence can be conveniently incorporated during the ringing time of each telephone call.

Table 4.1
Comparisons of Hardware Requirements between
the Proposed Method and a Direct Approach

	Direct	Proposed
Memory		
data	13 bits	9 bits
coefficient	21 bits	17 bits
total	34K	26K
Convolution		
multiplier	13x13 bits	6x6 bits
adder	No	4 bits
barrel shifter	No	11 bits 15 positions
accumulator	32 bits	32 bits
Adaptation		
multiplier	13x8 bits	No
adder(m)	21 bits	14 bits
adder(e)	No	3 bits
R/L shifter	No	14 bits
divider	Yes	No
$\sum X^2$		
multiplier	13x13 bits	No
left shifter	No	4 bits
barrel shifter	No	11 bits 15 positions
accumulator	32 bits	32 bits
L2F	No	Yes

The ERLE is computed by averaging the ratios of short term mean echo power over instantaneous residual echo for 500 samples.

Figure 4.5 shows the ERLE of the acoustic echo canceler with Gaussian inputs. The various curves correspond to different number of mantissa bits in the data and the coefficients. For 6 bit mantissa, after convergence, the ERLE is about 29dB. The performance is better than the previous assessment based on the quantization effects. This is because the adaptation process forces the resultant coefficients to be rounded rather than truncated. For reference purpose, a direct approach with all the multiplications intact and using the usual precision of a VAX machine is also included. The result indicates that the proposed method doesn't slow down the convergence speed much.

Figure 4.6 shows the ERLE of the acoustic echo canceler with speech inputs. A 1.5 second binary pseudo-random training sequence was used to speed up the initial adaptation.

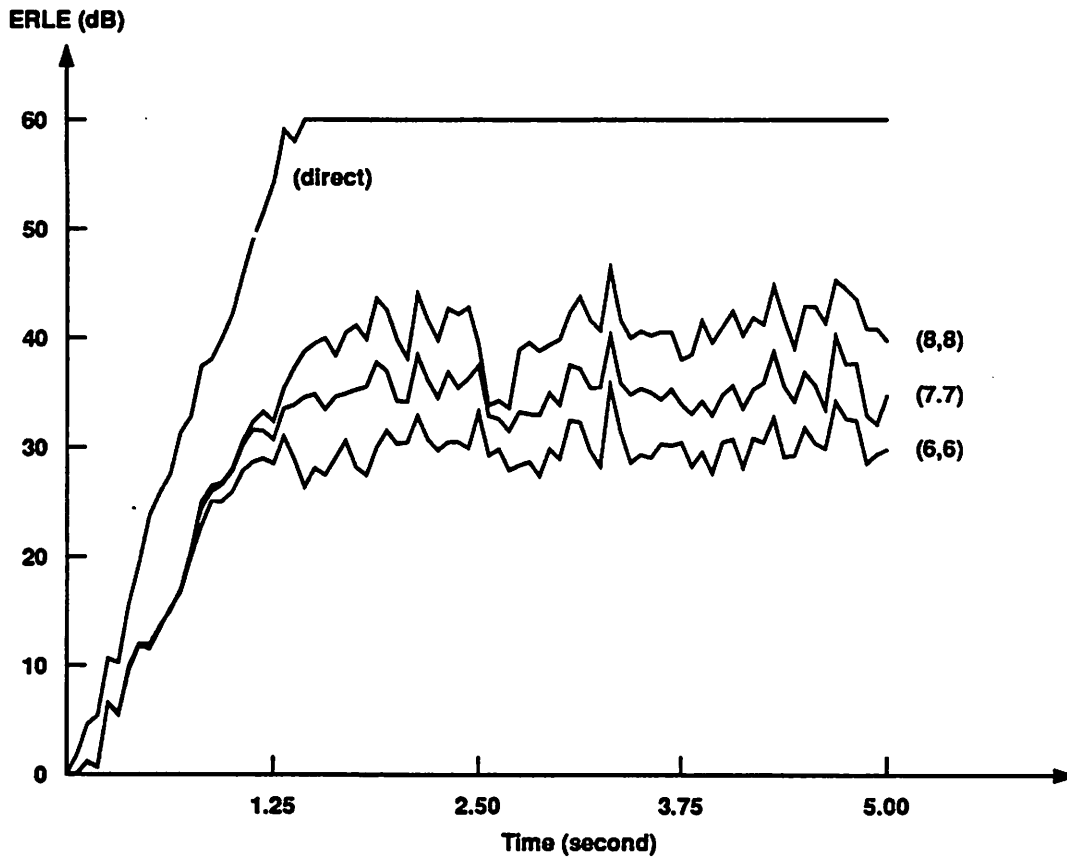


Figure 4.5 - Computer simulation of ERLE with Gaussian inputs

The result indicates that the proposed canceler also works well with speech inputs.

4.6 Detection of Near-End Talker Signal

As with any other echo canceler, the adaptation must be stopped if the near-end talker is speaking. A simple detection of near-end talker signal is based on the expected ERLE as shown in Figure 4.7. To calculate the ERLE, the microphone output and the residual error are both low pass filtered. The purpose is to obtain a short term averaged ERLE to avoid the fluctuation in the input signal.

$$ERLE(n) = 10 \log_{10} \frac{\sigma_y^2(n)}{\sigma_e^2(n)} \quad (4.14)$$

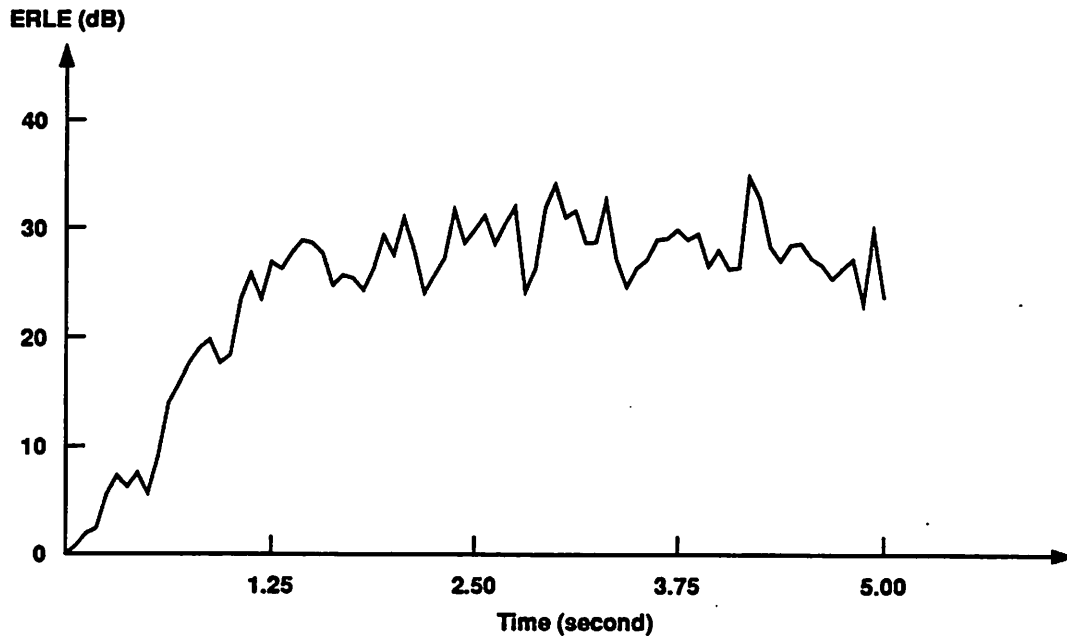


Figure 4.6 - Computer simulation of ERLE with speech inputs

If the ERLE is worse than a predetermined threshold, the presence of a near-end talker is declared and the adaptation is disabled. The choice of the threshold is a compromise between the possible change in the environment and the level of the near-end talker signal. Making the threshold too high will cause the adaptation to stop when a change in the impulse response occurs. A degradation of 6 dB in ERLE due to movements of the near-end talker has been reported [24]. On the other hand, a too small threshold will not stop the adaptation even though the near-end talker signal is at a moderate level. A threshold of 12 dB (approximately half of the expected ERLE) appears to be a good choice.

4.7 Alternative Implementations of the Acoustic Echo Canceler

Because the adaptive filter for the acoustic echo cancellation requires long impulse response and the adaptation is much easily done in the digital domain, an adaptive digital

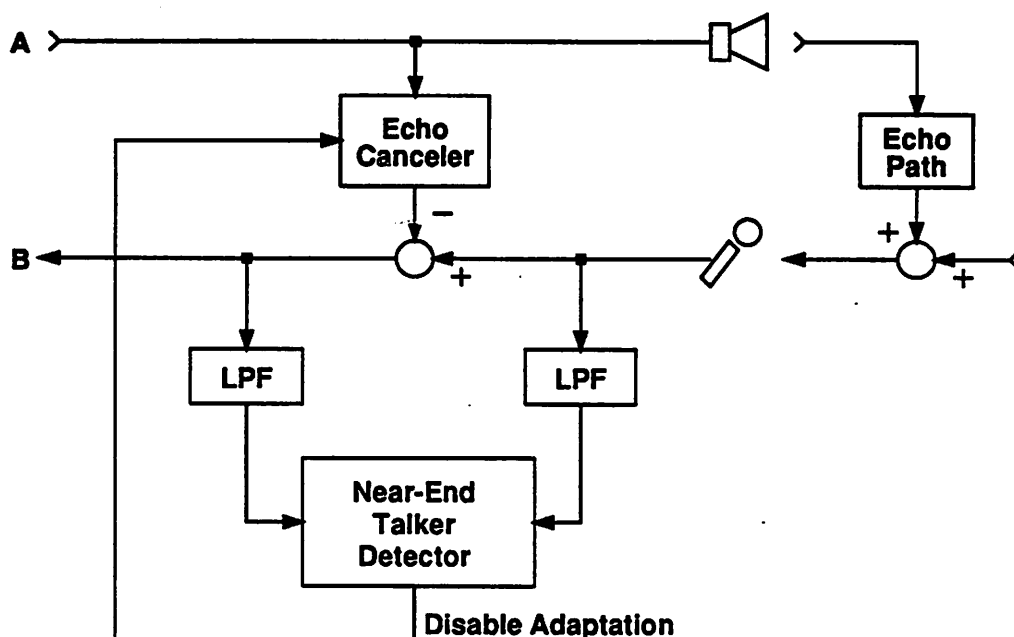


Figure 4.7 - Detection of near-end talker signal

transversal filter is chosen as the core component of the acoustic echo canceler. As a result, analog-to-digital (A/D) converters and digital-to-analog (D/A) converters are needed to interface with the speech signals (which are analog by nature.) Two configurations of alternative implementations will be discussed here: (1) analog cancellation and (2) digital cancellation.

Analog Cancellation

Figure 4.8(a) shows a digital echo canceler with analog cancellation. The far-end talker signals are quantized by an A/D converter. The required range of this A/D converter is 13 bits while the integral linearity is 6 bits (discussed in the next section), implying that this converter can be realized by a segmented μ -255 quantizer. The digital output of the echo replica is converted to an analog signal by a D/A converter with 13 bit range and 6 bit integral linearity (also discussed in the next section.) As a result, these two converters can be implemented by a typical PCM voice codec such as [25]. The requirements of the feedback A/D converter

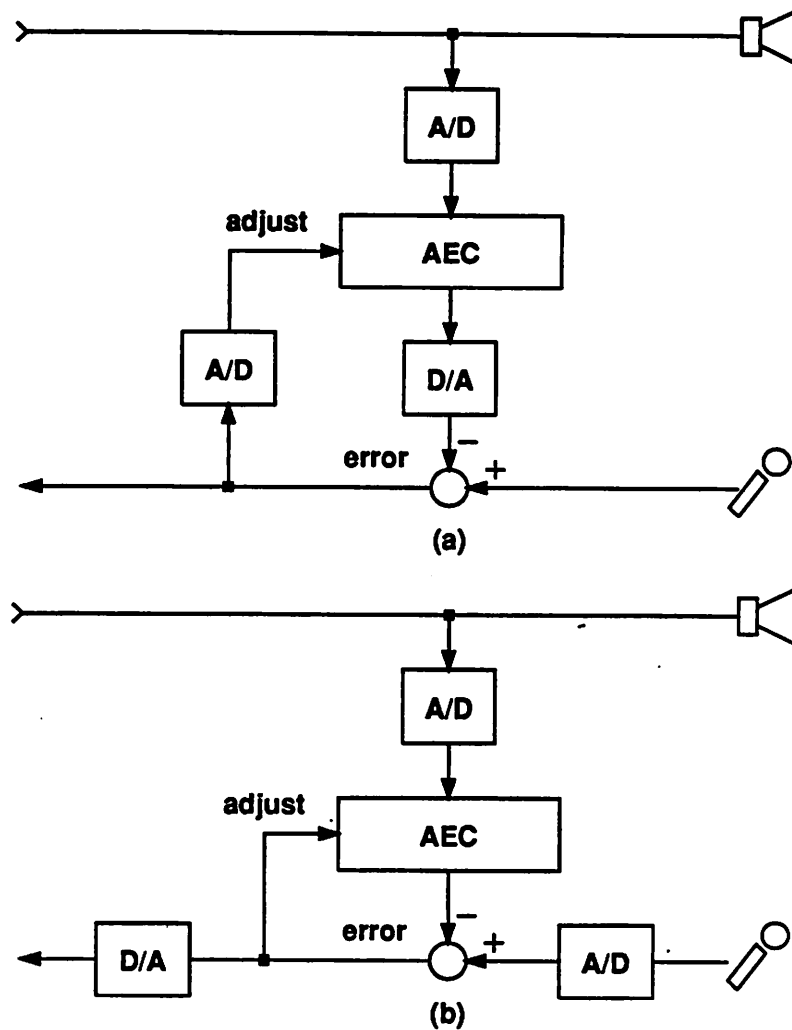


Figure 4.8 - Acoustic echo canceler with (a) analog cancellation, (b) digital cancellation

are much more relaxed because only the 2's power of the error signal is used in the adaptation computation. The main criterion is that it should be monotonic [26]. The range of the 2's power of 13 bit data is 12, which can be encoded by a 4 bit word. Therefore, a nonlinear A/D converter with 5 bit (including the sign bit) range is sufficient.

Digital Cancellation

Figure 4.8(b) illustrates a digital echo canceler with digital cancellation. The near-talker

signals and the returned far-end talker echoes are quantized by an A/D converter. The range of this converter is 13 bits but the integral linearity is 9 bits (3 bits more compared to the previous A/D converter) because the far-end talker echoes might be 15 dB higher than the near-end talker signals (Chapter 3.) The segmented μ -255 quantizer used in the previous configuration cannot be applied here since a reduction in signal-to-noise ratio by 15 dB will not be perceptively acceptable. Therefore, a linear 13 bit A/D converter must be used. The requirements of the A/D converter that digitizes the far-end talker signals remain the same as the one used in the previous configuration. Without further processing the digital output after the cancellation, a 13 bit D/A converter with 9 bit integral linearity is needed.

Comparing the requirements between the analog cancellation and the digital cancellation, it appears that the analog cancellation is more favorable. The major reason behind this conclusion is because the near-end talker signals might be contaminated by much larger far-end talker echoes (by 15 dB) causing the front-end quantization less favorable. As a result, the analog cancellation shown in Figure 4.8(a) is chosen for our acoustic echo canceler.

4.8 Effects of Imperfections in A/D and D/A Converters

The ideal transfer curves of the A/D and the D/A converters are shown in Figure 4.9. Typical imperfections in the A/D and the D/A converters include: (1) gain error (the slope of the transfer curve being not 1), (2) dc offset (the transfer curve not passing the origin, and (3) nonlinearity (the transfer curve being not a straight line). The effects of these imperfections on the performance of the acoustic echo canceler will be discussed in the next 3 subsections.

4.8.1 Gain Error

The far-end talker echo, $y(n)$, is generated by (we consider only the discrete time case for simplicity)

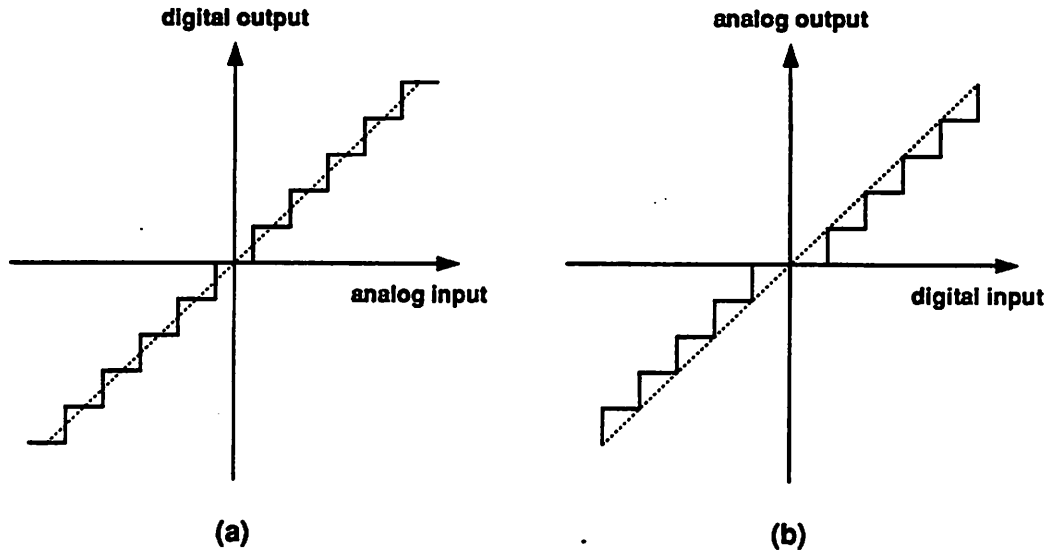


Figure 4.9 - Ideal transfer curves for (a) A/D (b) D/A converters

$$y(n) = \sum_{i=0}^{N-1} h_i x(n-i) \quad (4.15)$$

where h_i 's are the impulse response (assume time limited) of the echo path and $x(n-i)$'s are the far-end talker signals. The $x(n-i)$'s in this subsection and the following subsections are assumed to be independently and identically distributed (therefore they are uncorrelated and have the same statistics.) The echo replica, $\hat{y}(n)$, (after the D/A conversion) is computed from

$$\hat{y}(n) = \sum_{i=0}^{N-1} c_i x(n-i) \quad (4.1)$$

The residual error, $e(n)$, after cancellation is

$$e(n) = y(n) - \hat{y}(n) \quad (4.2)$$

If the A/D converter has a gain factor α other than 1, the generated echo replica will become

$$\hat{y}(n) = \sum_{i=0}^{N-1} c_i \alpha x(n-i) \quad (4.16)$$

Similarly, if a gain factor α is used in the D/A conversion, the echo replica will become

$$\hat{y}(n) = \alpha \sum_{i=0}^{N-1} c_i x(n-i) \quad (4.17)$$

Because equations (4.16) and (4.17) are identical, the effects of the gain errors in the A/D and the D/A converters will be treated in the same context.

Intuitively, the gain errors in the A/D and the D/A converters will be compensated by the feedback adaptation because the gain errors are linear. This is better illustrated by computing the variance of the residual error (which has zero mean):

$$E(e^2(n)) = \sum_{i=0}^{N-1} (h_i - \alpha c_i)^2 \sigma_x^2 \quad (4.18)$$

Minimizing $E(e^2(n))$ results in $h_i = \alpha c_i$ or $c_i = \frac{h_i}{\alpha}$. Therefore, the effect of gain error simply scales the values of the resultant coefficients provided they are still within the range of their word length. Because the coefficients are encoded in floating point, the quantization noise is also scaled by α giving the quantity αc_i the same quantization noise as those obtained with no gain errors in the A/D and the D/A converters. Consequently, the variance of the residual error after convergence remains the same (no scaling.)

The coefficients are updated by the LMS gradient algorithm:

$$c_i(n+1) = c_i(n) + \frac{\beta}{K + \sum_{k=0}^{N-1} x^2(n-k)} e(n) x(n-i) \quad (4.3)$$

Because the total power estimation is scaled by α^2 , the data, $x(n-i)$ is scaled by α , and the residual error, $e(n)$ is unscaled, the adjustments to the coefficients are scaled by α . In other words, the percentage of coefficient adjustment remains the same. As a result, the convergence speed stays the same even with gain errors in the A/D and the D/A converters. Figure 4.10 plots the effect of gain error on ERLE. As expected, the residual error and the convergence speed do not vary with different gain factors.

4.8.2 DC Offset

The dc offset in data converters can be modeled by adding a dc value to the resultant data after the conversion. For the offset in the A/D converter, this means the echo replica is

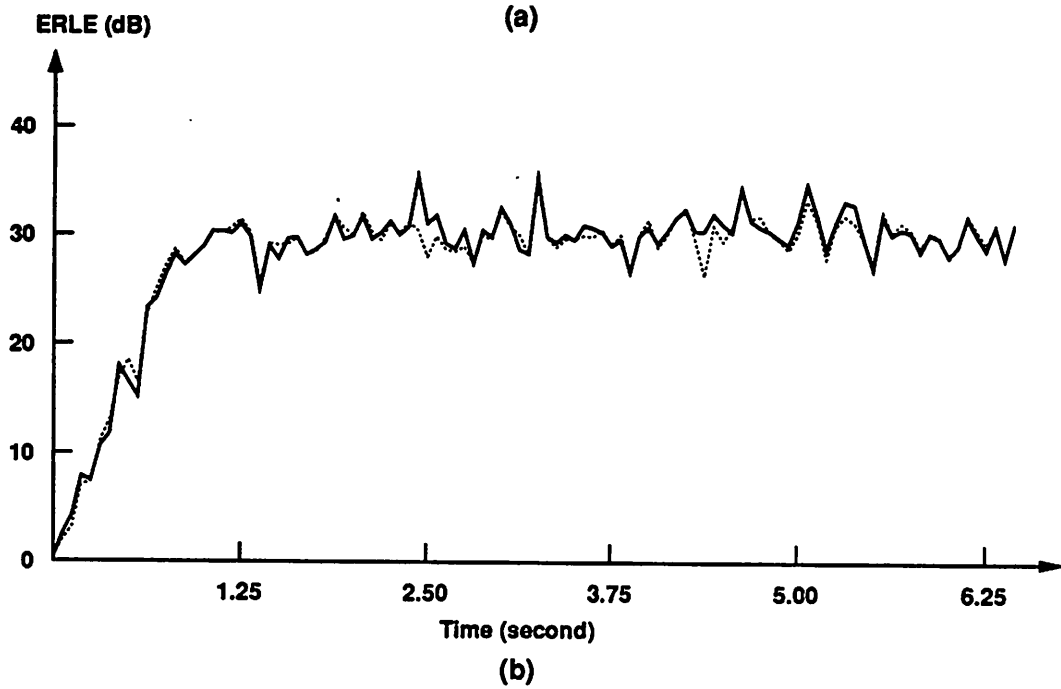
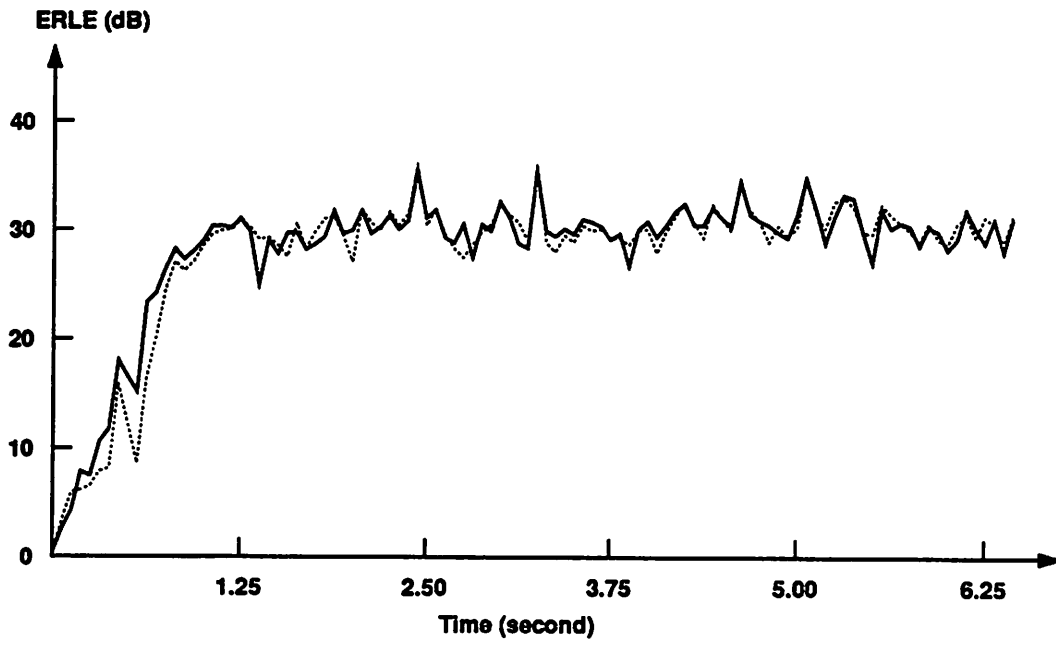


Figure 4.10 - Effects of gain errors ($\alpha = 0.8$) on ERLE: (a) A/D (b) D/A

represented by

$$\hat{y}(n) = \sum_{i=0}^{N-1} c_i (x(n-i) + \alpha), \quad (4.19)$$

whereas the offset in the D/A converter results in

$$\hat{y}(n) = \sum_{i=0}^{N-1} c_i x(n-i) + \alpha. \quad (4.20)$$

These offsets are not linear and, therefore, can not be compensated by a linear echo canceler.

Their effects are studied in the following paragraphs.

Offset In A/D Converters

From equation (4.19), the residual error with offset in the A/D converter is

$$e(n) = \sum_{i=0}^{N-1} (h_i - c_i) x(n-i) - \sum_{i=0}^{N-1} c_i \alpha \quad (4.21)$$

The mean of the input data, $x(n-i)$, is zero. The power of the residual error is

$$E[e^2(n)] = \sum_{i=0}^{N-1} (h_i - c_i)^2 \sigma_x^2 + \alpha^2 \left(\sum_{i=0}^{N-1} c_i \right)^2 \quad (4.22)$$

where the second term on the right hand side is an additional power of the residual error. To minimize the error power, take derivatives on both sides of equation (4.22) with respect to c_i 's and set the derivatives to zeroes. It can be shown that

$$c_i = h_i - \frac{\alpha^2}{\sigma_x^2} \sum_{i=0}^{N-1} c_i \quad (4.23)$$

For a 13 bit A/D converter (overloaded at 4σ) with 2 LSB offset,

$$\frac{\alpha^2}{\sigma_x^2} \sum_{i=0}^{N-1} c_i \leq \frac{\alpha^2}{\sigma_x^2} N = \frac{2^2}{2^{20}} 2^{10} = \frac{1}{2^8} \ll 1 \quad (4.24)$$

Therefore, $c_i \approx h_i$ will minimize the error power. To prove the convergence will actually take place, consider the coefficient update:

$$\begin{aligned} c_i(n+1) &= c_i(n) + \beta' e(n) x(n-i) \\ &= c_i(n) + \beta' \left[\sum_{k=0}^{N-1} (h_k - c_k(n)) x(n-k) - \sum_{k=0}^{N-1} c_k(n) \alpha \right] x(n-i) \end{aligned} \quad (4.25)$$

Taking the expectation values on both sides of equation (4.24) yields:

$$E[c_i(n+1)] = (1 - \beta \sigma_x^2) E[c_i(n)] + \beta' h_i \sigma_x^2 \quad (4.26)$$

If we define the misalignment error between the coefficient and the impulse response as

$$\varepsilon_i(n+1) = E[c_i(n+1)] - h_i \quad (4.27)$$

Substitute (4.27) into (4.26) results in

$$\begin{aligned}\varepsilon_i(n+1) &= (1 - \beta\sigma_x^2) \varepsilon_i(n) \\ &= (1 - \beta\sigma_x^2)^{n+1} \varepsilon_i(0)\end{aligned}\tag{4.28}$$

Therefore, for $|1 - \beta\sigma_x^2| < 1$, the misalignment error will become smaller and smaller (converging to h_i .)

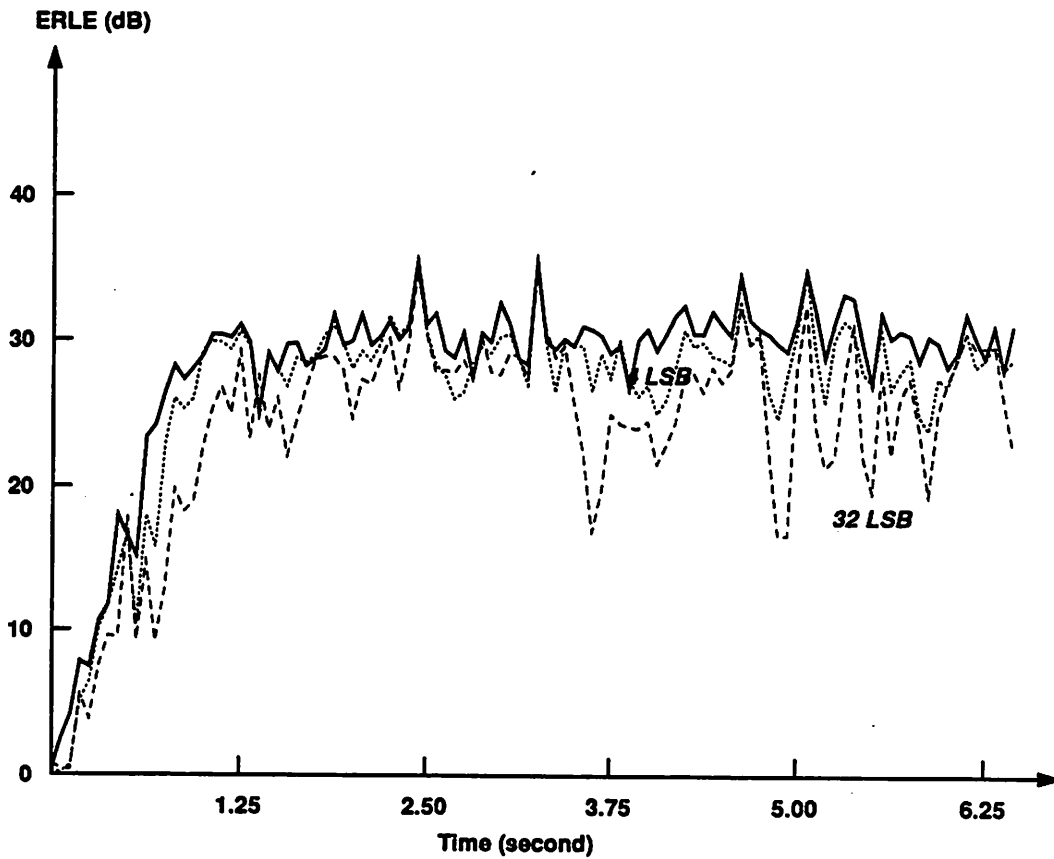


Figure 4.11 - Effects of the A/D offset on ERLE

The effect on the convergence speed introduced by the A/D offset is illustrated by computer simulations. The A/D converter has 13 bits and the dc offsets are 4 LSB and 32 LSB. The results are shown in Figure 4.11. For 4 LSB dc offset, the degradation on ERLE is about 2 dB and the convergence speed is not much affected.

Offset In D/A Converters

From equation (4.20), the residual error and the error power are:

$$e(n) = \sum_{i=0}^{N-1} (h_i - c_i) x(n-i) - \alpha \quad (4.29)$$

$$E[e^2(n)] = \sum_{i=0}^{N-1} (h_i - c_i)^2 \sigma_x^2 + \alpha^2 \quad (4.30)$$

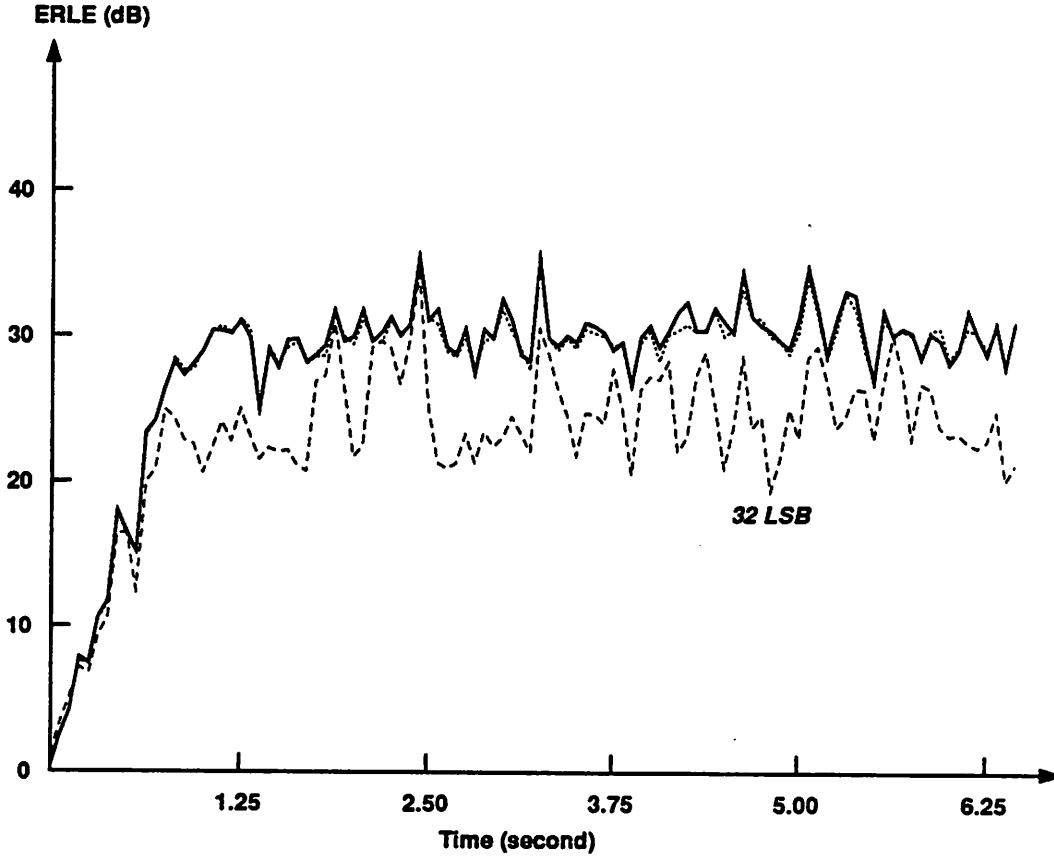


Figure 4.12 - Effects of the D/A offset on ERLE

Therefore, an additional error power of α^2 is introduced by the D/A offset. The error power is minimized by $c_i = h_i$ (which will be accomplished by the gradient algorithm.) To show that the LMS gradient algorithm will lead to the convergence of the coefficients, equations similar to (4.25) through (4.28) can be derived and the misalignment error will converge to zero provided $|1 - \beta\sigma_x^2| < 1$ (as also concluded in the A/D offset.) This shows that the average

trajectory of the coefficient will converge but, of course, with greater error power. Figure 4.12 illustrates the effect of the D/A offset on ERLE through computer simulations. The D/A converter has 13 bits and two offset values, 4 LSB and 32 LSB, were chosen in the simulations.

4.8.3 Nonlinearity

The nature of the distortion introduced by the D/A (or A/D) nonlinearity in an echo canceler for full-duplex data transmission has been extensively investigated by Agazzi et al. in [27], where the nonlinearity is represented by an expansion similar to a Volterra series expansion in continuous amplitude signals. The approach there can also be applied to data samples (instead of data symbols in [27].)

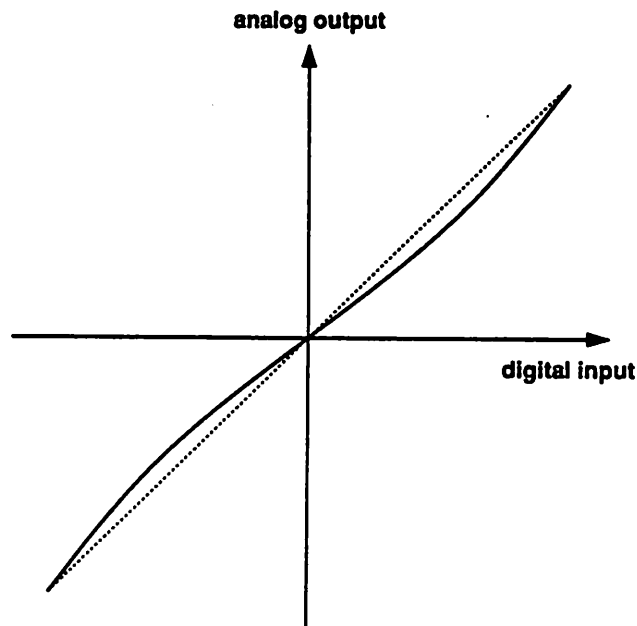


Figure 4.13 - Nonlinear D/A transfer function

However, to compute the power of the residual error, it is more convenient to model the D/A (or A/D) nonlinearity with a known function. Figure 4.13 shows a typical transfer curve of a

D/A converter (similarly for an A/D) normalized to its maximum value. This transfer curve can be modeled as

$$f(x) = \left[a \left(\frac{x}{x_{\max}} \right) + b \left(\frac{x}{x_{\max}} \right)^3 \right] x_{\max} \quad (4.31)$$

where $a + b = 1$ such that the end points are fixed (usually by the reference supplies.) For the A/D nonlinearity, the echo replica is represented by:

$$\hat{y}(n) = \sum_{i=0}^{N-1} c_i f \left[x(n-i) \right] \quad (4.32)$$

For the D/A nonlinearity, the resultant echo replica is:

$$\hat{y}(n) = f \left[\sum_{i=0}^{N-1} c_i x(n-i) \right] \quad (4.33)$$

In other words, the nonlinearity in an A/D (or D/A) converter is modeled as a nonlinear transformation followed by an ideal A/D (or D/A) converter.

The equivalent integral linearity of the nonlinear transfer curve shown in Figure 4.13 can be obtained by computing the maximum deviation of the transfer curve away from the ideal curve (the dashed line.) It is much easier to work on the normalized transfer functions where the transformations for an ideal A/D (or D/A) converter and an nonlinear one is represented by:

$$g(x) = x \quad (4.34)$$

$$f(x) = ax + bx^3 \quad -1 < x < 1, \quad a+b = 1 \quad (4.35)$$

The deviation (Δ) due to the nonlinearity is

$$\Delta = g(x) - f(x) = (1-a)x - bx^3 \quad (4.36)$$

The maximum deviation (Δ_{\max}) occurs at a point where the first derivative of equation (4.36) is zero. It can be shown that

$$\Delta_{\max} = \frac{2b}{3\sqrt{3}} \quad (4.37)$$

For 6-bit integral linearity, $\Delta_x = 2^{-5}$ and $b = 0.08$.

The effects of the A/D (or D/A) nonlinearity on the ERLE are demonstrated by computer

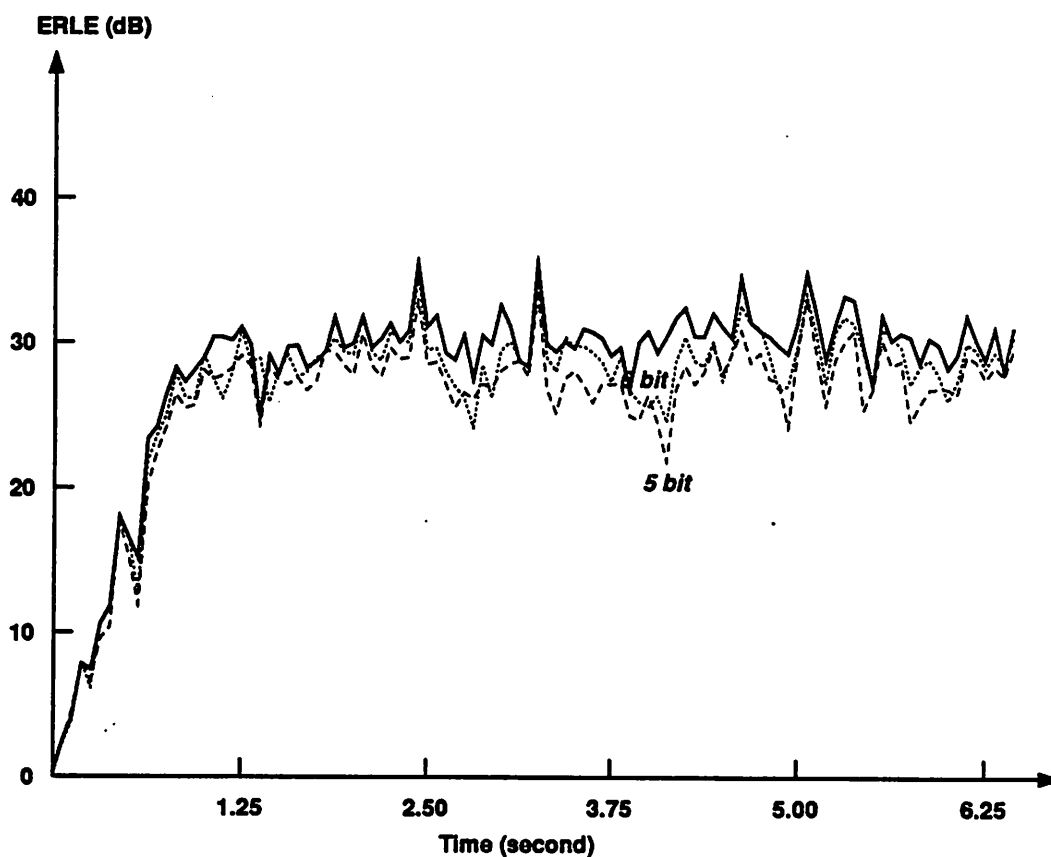


Figure 4.14 - Effects of the A/D nonlinearity on ERLE

simulations based on equations (4.32) and (4.33). Figure 4.14 illustrates the convergence and the ERLE of the echo canceler due to the A/D nonlinearity. For 6 bit integral linearity, the degradation is quite acceptable. Figure 4.15 presents the results with D/A nonlinearity. Again the degradation is not critical for 6 bit integral linearity.

Table 4.2 lists the effects of the imperfections in the A/D (or D/A) on the performance of the acoustic echo canceler and the required limitations on these imperfections.

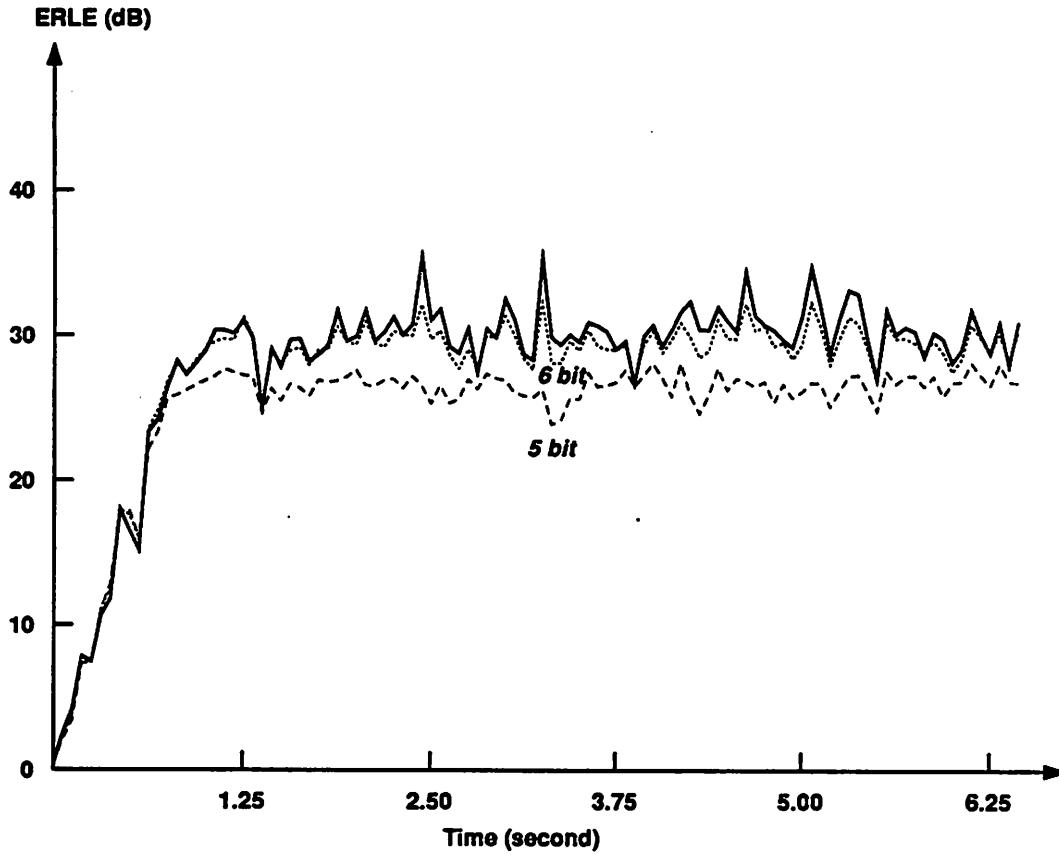


Figure 4.15 - Effects of the D/A nonlinearity on ERLE

Table 4.2 Effects of the A/D and D/A Imperfections and Their Requirements

Imperfections		Effects on the ERLE	Requirements
gain error		no	must not overflow coefficients
dc offset (α)	A/D	mild degradation	$< 4 \text{ LSB}$
	D/A	mild degradation	$< 4 \text{ LSB}$
nonlinearity	A/D	slight degradation	$> 6 \text{ bit integral linearity}$
	D/A	slight degradation	$> 6 \text{ bit integral linearity}$

CHAPTER 5

PROTOTYPE ACOUSTIC ECHO CANCELER

The goals in the design of this prototype were to demonstrate the possibility of a 1000 tap, single chip acoustic echo canceler and examine the schemes outlined in the previous chapters. The prototype was designed to operate at 8 MHz clock rate such that 1000 cycles would be available for the adaptation and the convolution computations. To meet this speed requirement and to minimize the design effort, parallelism and pipelining were exploited at several levels in the design. Although the memory was not included in this prototype, all the address and read/write signals needed for memory access were generated on the chip. The core area of the chip is 5.3 mm by 5.3 mm using a 3 μ m double metal CMOS process and can be scaled down for a 2 μ m process. The chip was designed to have a maximum of 1024 taps but its number of taps can be programmed so that residual errors can be reduced if fewer taps are sufficient. A test mode was incorporated in the chip for this purpose.

5.1 Design Methodology and Design Environment

In a complicated chip design such as this one, a highly structured design process and a well-coordinated design environment including the CAD tools are essential to a successful implementation. To realize an acoustic echo canceler, a description of the high level behavior or architecture of the processor is transformed into a collection of geometric layouts. As the echo canceler goes from a behavioral description and system specifications to the final form on the silicon, it passes through many representations and their corresponding design levels. Interactions among various design levels are necessary to obtain best results. Figure 5.1 sum-

marizes the design process adopted for this project, in which we divide the design levels into system, architecture, logic, and layout.

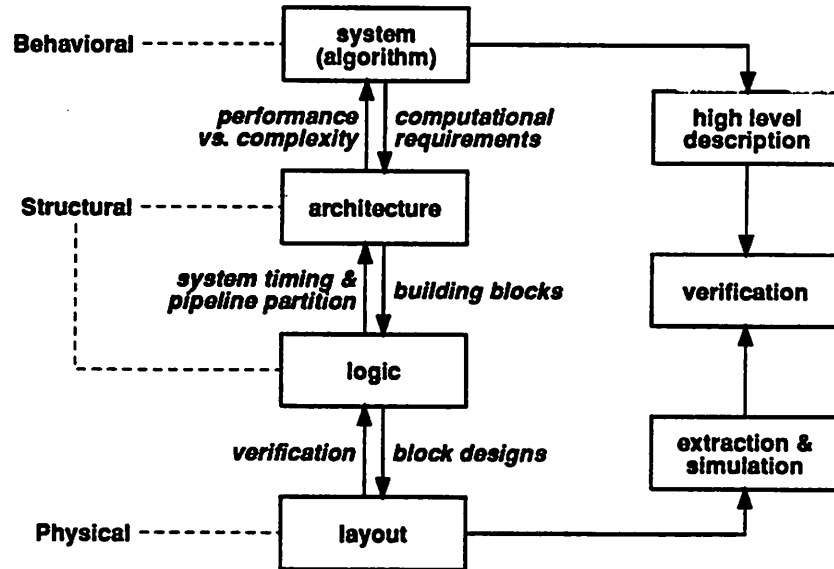


Figure 5.1 - Design process of the chip

In the system level, an echo cancellation algorithm is first transformed into a set of computational requirements such as additions and multiplications. We then examine the implementation of these requirements. For example, a direct realization of the least mean square (LMS) gradient algorithm would require one multiplier of size 13x13 operating at 32 MHz rate or four multipliers at 8 MHz rate. Neither of them is feasible on a single chip. As a result, system specifications were reexamined and design trade-offs were considered. The multiplier was subsequently reduced in size to 6x6 bits. A C language program representing the complete data path including the finite word length, the power-of-two multiplication, and the floating point addition was written to verify this new architecture and its hardware. System and circuit operation were examined through extensive computer simulations as discussed in the previous chapter.

After the architecture was decided, a hierarchical division of the functional blocks was performed. An initial assessment of speed requirements resulted in the system timing specification and the pipeline partition. Each block was implemented and verified using various CAD tools. After the complete chip was laid out, we extracted the electrical circuit that was implemented by the layout. Logic and circuit simulations were performed. The results were compared with those generated by the high level simulation program.

The complete physical design and verification of this chip was conducted on a μ Vax workstation with a color graphic monitor. Figure 5.2 shows the CAD environment set up for this design.

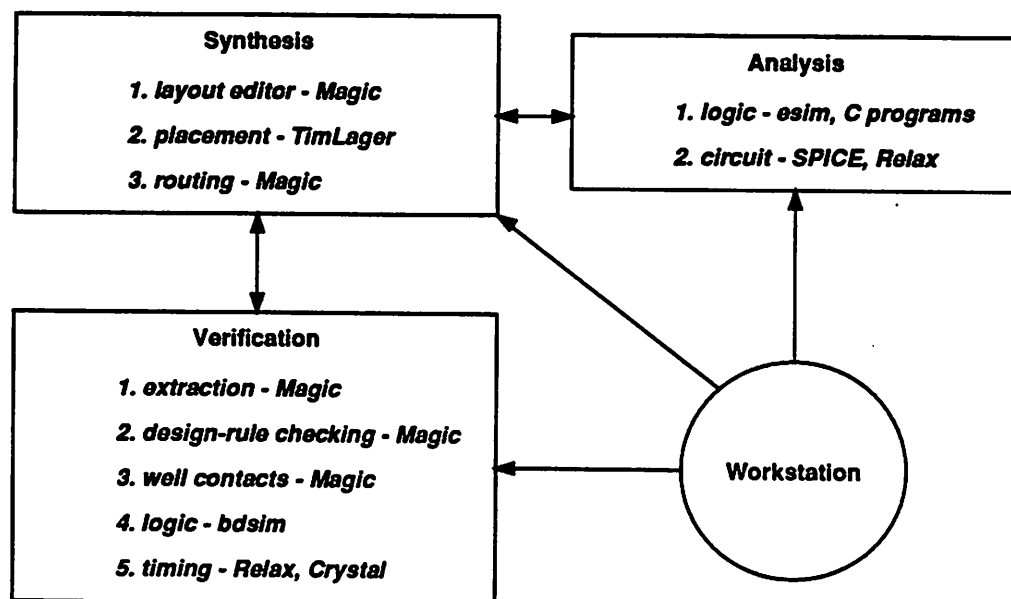


Figure 5.2 - Design environment of the chip

Logic simulation was done by esim and C language programs. SPICE and Relax were used for critical circuit simulation. Relax uses a waveform relaxation technique, which runs much faster than SPICE for a large circuit and is good for catching any racing or glitch problems in the control circuits. Magic is our major layout editor. Its built-in array tiling and router are good

for generating random logic by using standard cells and channel routing. TimLager (*Tiler for module Layout generation*) is helpful for the placement of I/O pad cells. Design rule violations are continuously checked and flagged by Magic. Mis-placed well contacts or floating wells can be detected by taking advantage of the capacitance among various layers in Magic. The extracted circuit was simulated by using bdsim, Relax, or Crystal. Basim is an event-driven switch-level simulator that allows the user to specify weak transistors and the direction of current flow. These features are necessary for the simulation of pass-gate exclusive-OR circuit and some circuits with feedback. Timing verification is performed by a direct circuit simulation using Relax on the control circuits and a critical path analysis using Crystal.

5.2 Architecture

The basic chip architecture of the acoustic echo canceler is shown in Figure 5.3. Since each coefficient is updated in every sample period, we need to do one read and one write per clock cycle for the coefficients. To reduce the bandwidth requirement of the memory, the coefficients are divided into two banks. The data and the coefficients are pipelined through the adaptation computation. The updated coefficients can then be both written back to the memory and pipelined through the convolution processor. The new data register holds the data before it is written to the data memory. Normalization circuitry controls the amount of adjustment to the coefficients in accordance with the input signal power. A double talk flag disables the adaptation in the presence of a near-end talker signal.

5.2.1 Chip Interface

A chip interface to the external memory, the A/D, and the D/A is shown in Figure 5.4. As it will be discussed in Section 5.3, a common address bus is shared by all memory banks. The data memory is 9 bits wide while the coefficient memory is 17 bits wide. In the test mode,

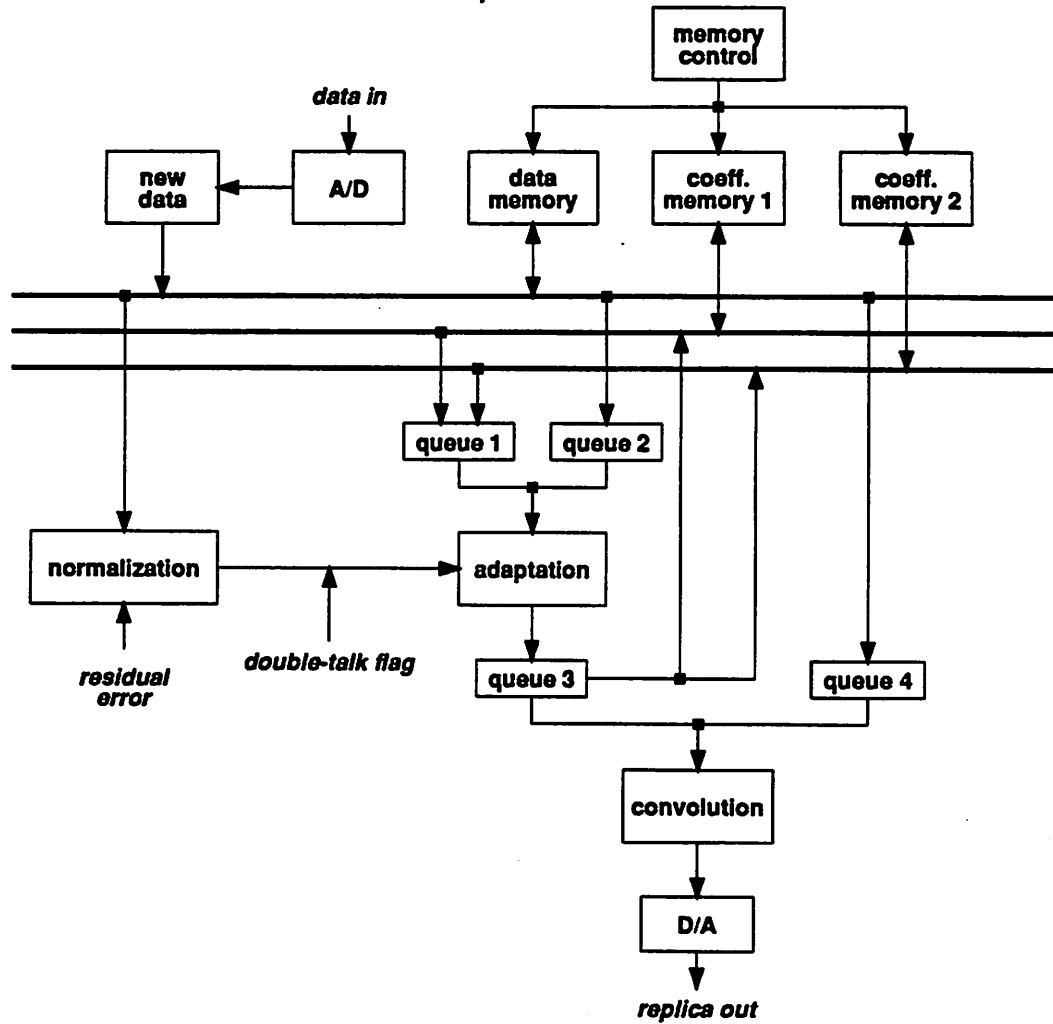


Figure 5.3 - Architecture of the acoustic echo canceler

the pins shown by the dashed lines set the parameter registers and program the number of taps in the echo canceler. To identify the functions of these parameters, the equation for the coefficient update is repeated below:

$$c_i(n+1) = c_i(n) + \frac{\beta}{K + \sum_{k=0}^{N-1} x^2(n-k)} e(n) x(n-i) \quad (5.1)$$

where β and K are shown as beta and Vth in Figure 5.4. The A/D converter is a μ -255 coder. Because the outputs to a D/A converter is time-multiplexed to save the I/O lines, "DA_latch"

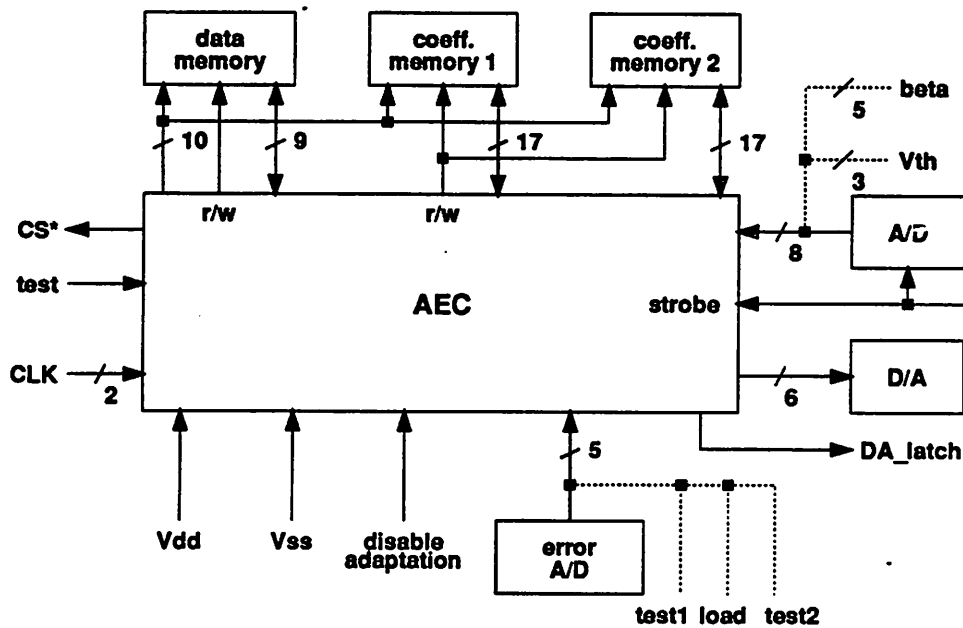


Figure 5.4 - Interface to the chip

is set when the upper 6 bits are available and reset for the lower 6 bits. The error A/D converter can be implemented by a 12 bit A/D converter in series with a 11 input priority decoder or a 5 bit nonlinear A/D converter. Two phase non-overlapping clocks are used in this chip.

5.2.2 General Clock Scheme

We use non-overlapping two phase clocks to isolate the pipeline stages (Figure 5.5.) To simplify the timing design, an input is latched into a pipeline stage at $\Phi 1$ (ph1) and its output released at $\Phi 2$ (ph2) for all stages of the data path. For memory access control, the address is set up at $\Phi 2$, a fetch request is issued at $\Phi 1$, and the data latched at the following $\Phi 2$. The clock period is 125 ns with a $\Phi 1 - \Phi 2$ non-overlap interval of 20 ns. As a result, the worst delay in any pipeline stage must be no more than 85 ns. The rule-of-thumb is to limit the number of gate delays within any pipeline stage to no more than 10.

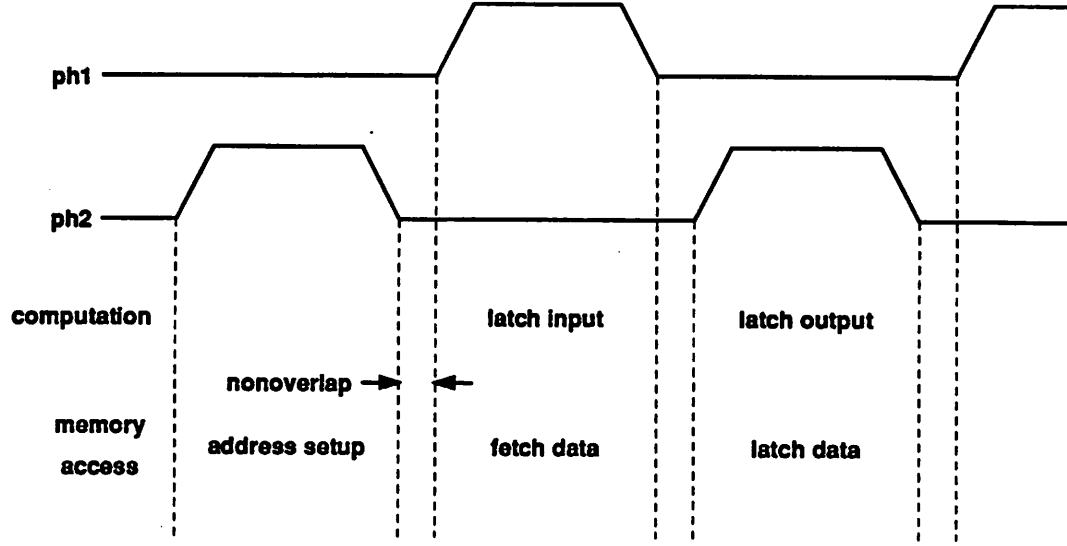


Figure 5.5 - General clock scheme

5.3 Memory Allocation and Access Control

Because of the large quantity of data and large number of coefficients, memory fetch and storage deserves some attention to reduce the bus traffic. In this chip, we interleave data and coefficients in the memory and share the same address for all memory banks to reduce the bandwidth requirement and the complexity of address generation. Parallelism and pipelining are exploited to achieve high throughput.

5.3.1 Interleaved Data Storage

The data storage and processing is designed to facilitate the parallelism and the pipelining of adaptation and convolution. The coefficients update at sample time n based on the LMS gradient algorithm is described by (lumping all constants in the same sample cycle):

$$C_i(n+1) = C_i(n) + \beta e(n) x(n-i) \quad (5.2)$$

To compute C_7 , for example:

$$C_7(n+1) = f(C_7(n), x(n-7))$$

On the other hand, the convolution computation at sample time $n+1$ is represented by:

$$\hat{y}(n+1) = \sum C_i(n+1) x(n+1-i) = \dots + C_7(n+1) x(n-6) + \dots \quad (5.3)$$

Therefore, at sample time n , $C_7(n)$ and $x(n-7)$ are used to update the coefficient $C_7(n+1)$. At sample time $n+1$, the updated $C_7(n+1)$ is multiplied by $x(n-6)$ as part of the convolution. These two operations are carried out in the same sample cycle in this chip. The updated coefficients are first generated and then multiplied by the appropriate data to obtain the convolution sum. To achieve in-place and in-time adaptation and convolution, data are interleaved to account for the clock delays in a pipeline design. Figure 5.6 illustrates this scheme in further detail.

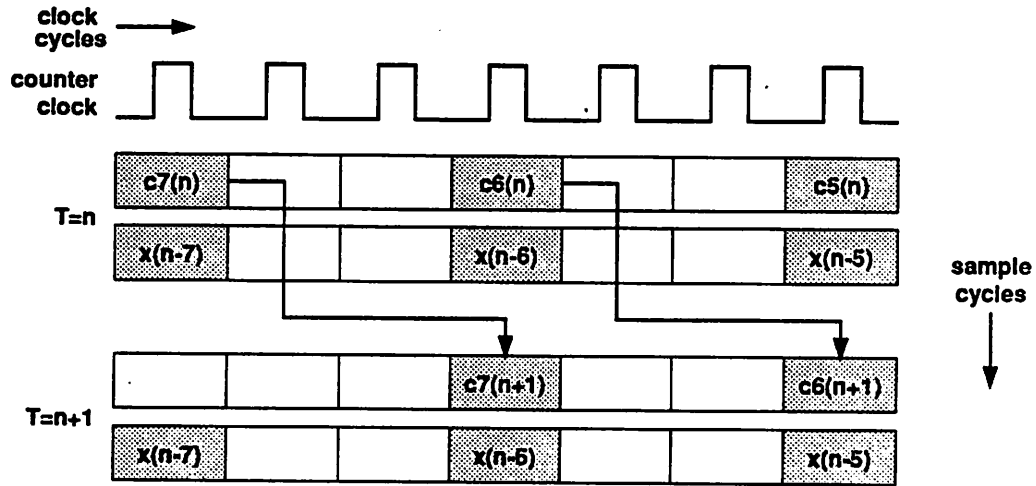


Figure 5.6 - Interleaved data storage

In Figure 5.6, the clock cycle time is shown to move from left to right and the sample cycle time is moving from top to bottom. The clock rate is 8 MHz and the sample rate is 8 kHz. As the clock cycles, a 10 bit counter generates the address for the memory. The memory location represented by the address is shown by a box and its content by the corresponding

label. As a result, the coefficient memory and the data memory are represented by two rows of boxes at any given sample time. Since the adaptation computation is pipelined to meet the 8 MHz speed requirement, the updated $C_7(n+1)$ won't be available until 3 clock cycles later. If we interleave the data $x(n-6)$ from $x(n-7)$ by 3 positions, when the updated $C_7(n+1)$ is ready to be written back to the memory, it is also in the position to be multiplied by $x(n-6)$. Therefore, instead of shifting the data through a shift register, we rotate the address through a 10 bit counter. However, as shown in Figure 5.6, the coefficient $C_6(n)$ has to be read before $C_7(n+1)$ can be written to that location. As a result, the required cycle time of memory access would be less than 50 ns and the internal clock rate would be more than 20 MHz.

In general, if there are p pipeline stages in the adaptation process, the adjacent data (in the time domain) are interleaved by $p-1$ positions. A sample cycle begins at a memory position where the oldest data resides. The newest data is also separated from the oldest data by $p-1$ positions to complete a circular storage. Figure 5.7 depicts an example of such an arrangement.

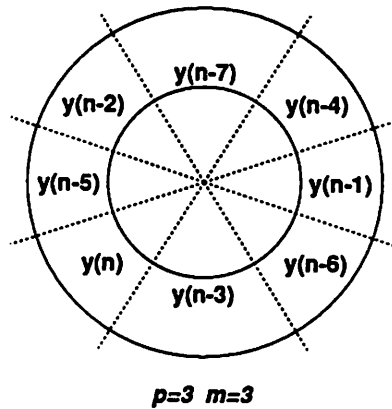


Figure 5.7 - Circular memory with interleaved data storage

The data are stored in m groups with the i -th data of the j -th group being adjacent (in the time domain) to that of the $(j+1)$ -th group. Since the adjacent data are separated by $p-1$ positions,

there are p items of data in each group. The oldest data is at both the first position of the first group and the last position of the last group because of the circular nature of this arrangement. Consequently, there are $pm-1$ data with the oldest data belongs to two groups. This presents a minor restriction that the number of taps has to be in the form of $pm-1$. In this chip, $p=9$ and $m=113$. Therefore, the maximum number of taps of the echo canceler is 1016.

5.3.2 Sharing the Same Address for all Memory Banks

As pointed out in the previous section, one coefficient needs to be read and another has to be written in every clock cycle. In this chip, coefficients are divided into two memory banks to reduce the bandwidth requirement of the memory access. Both coefficient banks are being read at one clock cycle and written at another. On each cycle, the same address is used for the data memory and the two coefficient memory banks to reduce the width of the address bus and the complexity of address generation. Figure 5.8 illustrates this design in further detail.

The clock cycle time moves from left to right and the sample cycle time moves from top to bottom in Figure 5.8. The address of the data memory is generated by a 10 bit counter. The coefficient memory is divided into two banks with their 9 bit address also generated by the same 10 bit counter by stripping off the least significant bit. As a result, the 10 bit counter points to the same coefficient memory location for 2 clock cycles. We use the first clock cycle to read in the coefficients and the second clock cycle to write out the updated coefficients. As an example, C_7 is updated and is in the right position to multiply $x(n-6)$ when $x(n-6)$ is read in. Similarly, C_4 with the same address as C_7 but in a different bank goes into the adaptation pipeline a clock cycle later and produces the updated C_4 also in the right position to multiply $x(n-3)$. Both updated C_4 and C_7 are written back to the coefficient memory in the second clock cycle. The content of the data memory remains the same except at the end of a sample cycle, where the oldest data is replaced by the newest data. Notice that $x(n-3)$ and $x(n-6)$ are

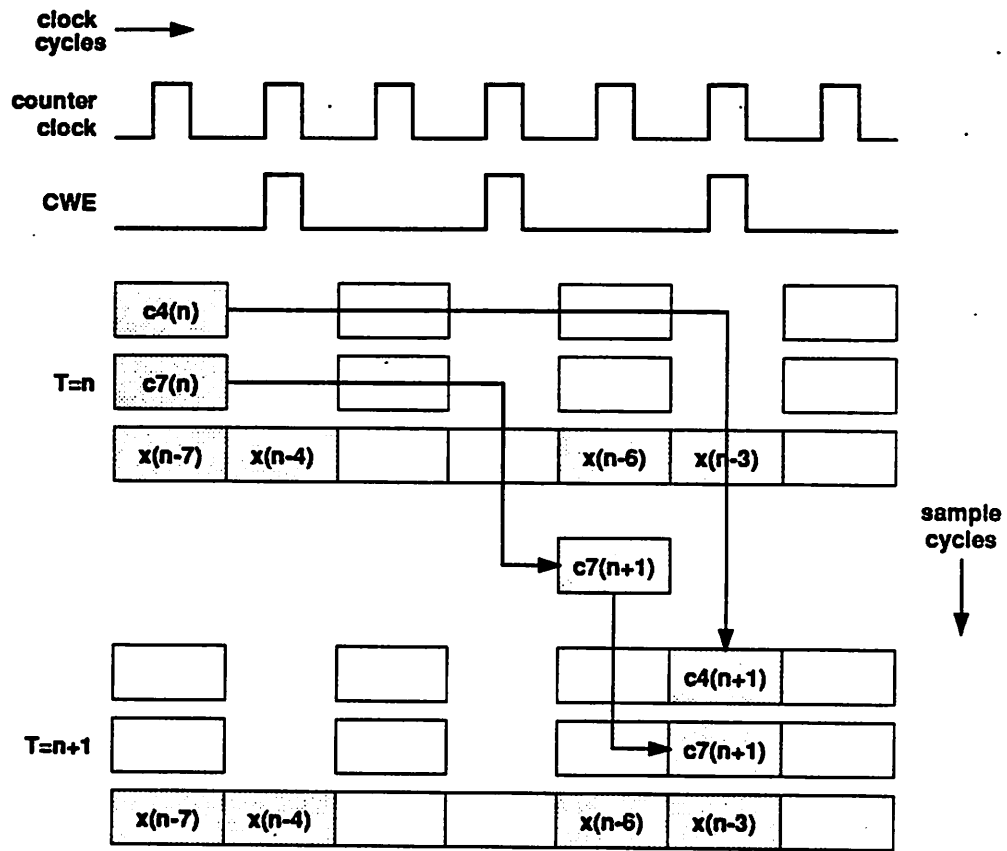


Figure 5.8 - Multi-bank memory and memory access

separated from $x(n-4)$ and $x(n-7)$ by p positions to account for the clock cycles needed for the coefficient update. Both the coefficient memory banks and the data memory share the same address generated by the 10 bit counter.

In summary, interleaved data storage is used to achieve in-place convolution by rotating the address in circle. To reduce the bandwidth requirement, two coefficient memory banks sharing the same address with the data memory are used. This reduces the width of the address bus and simplifies address generation. In fact, the single 10 bit counter generates the address for both the data memory and the coefficient memory banks.

5.3.3 Memory Access Control Circuits

A simplified memory access control is shown in Figure 5.9.

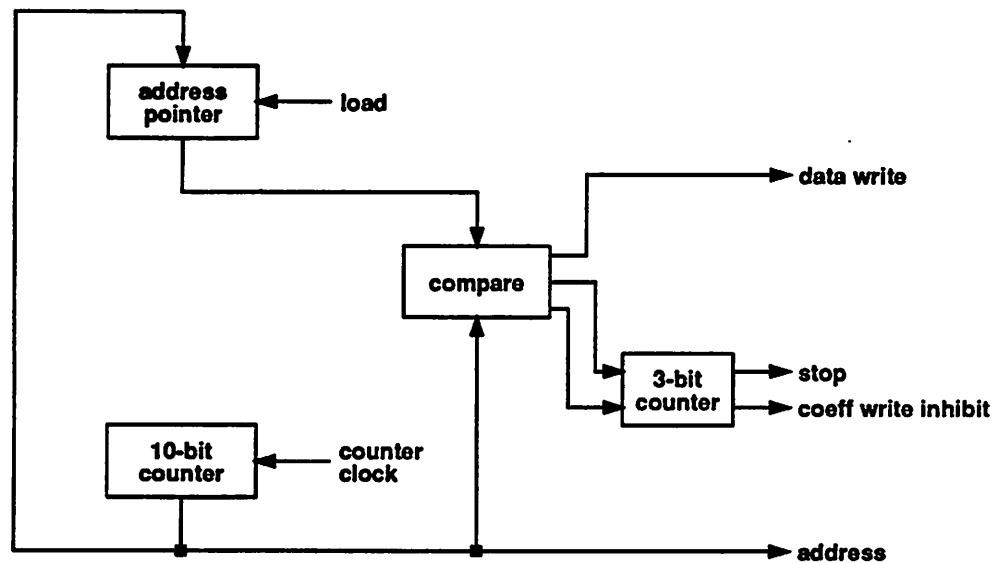


Figure 5.9 - Simplified memory access control

The address is generated by a 10 bit counter. The counter clock is enabled when a new data comes in. The address pointer holds the address of the oldest data, which is also the starting position of a sample cycle. When the content of the 10 bit counter passes its starting position, the oldest data is replaced by the newest data. At the same time, a 3 bit counter is enabled to determine the end of the sample cycle. This is determined by the number of pipeline stages. Therefore, the 10 bit counter serves as an address generator and a program counter.

5.4 Data Paths of Adaptation and Convolution

The data paths of adaptation and convolution are designed with emphasis on regularity and modularity. Because the multiplications in the adaptation process are carried out as power-of-two multiplications, no multipliers are required. The convolution process, on the

other hand, needs a 6x6 multiplier and a 12 bit by 14 position barrel shifter to convert the floating point output into a fixed point format for the subsequent accumulation. In order that the total input signal power can be calculated in every sample cycle, a few clock cycles are allocated so that this computation can be done by time-multiplexing the same hardware used for the convolution. Both the adaptation and the convolution data paths are pipelined to achieve a 8 MHz processing speed. Non-overlapping two phase clocks are used to isolate the pipeline stages.

5.4.1 The Adaptation Processor

The coefficient update based on the LMS gradient algorithm is described by:

$$c_i(n+1) = c_i(n) + \frac{\beta}{K + \sum_{k=0}^{N-1} x^2(n-k)} e(n) x(n-i) \quad (5.4)$$

If power-of-two multiplications are adopted, equation (5.4) becomes:

$$\begin{aligned} c_i(n+1) &= c_i(n) \pm \frac{2^\alpha}{2^L} 2^{Pe} 2^{Px} \\ &= c_i(n) \pm 2^{\alpha-L+Pe+Px} \end{aligned} \quad (5.4a)$$

where $L = \log_2 \left[K + \sum_{k=0}^{N-1} x^2(n-k) \right]$ The numerical ranges of $\beta, K + \sum x^2(n-k), e(n), x(n-i)$ and, therefore, α, L, Pe, Px are listed in Table 5.1.

Table 5.1 Range of Parameters

variable	range	variable	range
β	(.125, 1)	α	(-3, 0)
$K + \sum x^2(n-k)$	($2^{10}, 2^{30}$)	L	(10, 30)
$e(n)$	(0, 2^{11})	Pe	(0, 11)
$x(n-i)$	(0, 2^{12})	Px	(0, 12)

Because $\alpha - L + Pe$ needs to be computed only once in every sample period, the result is

designated as η and used in the coefficient update throughout the sample period.

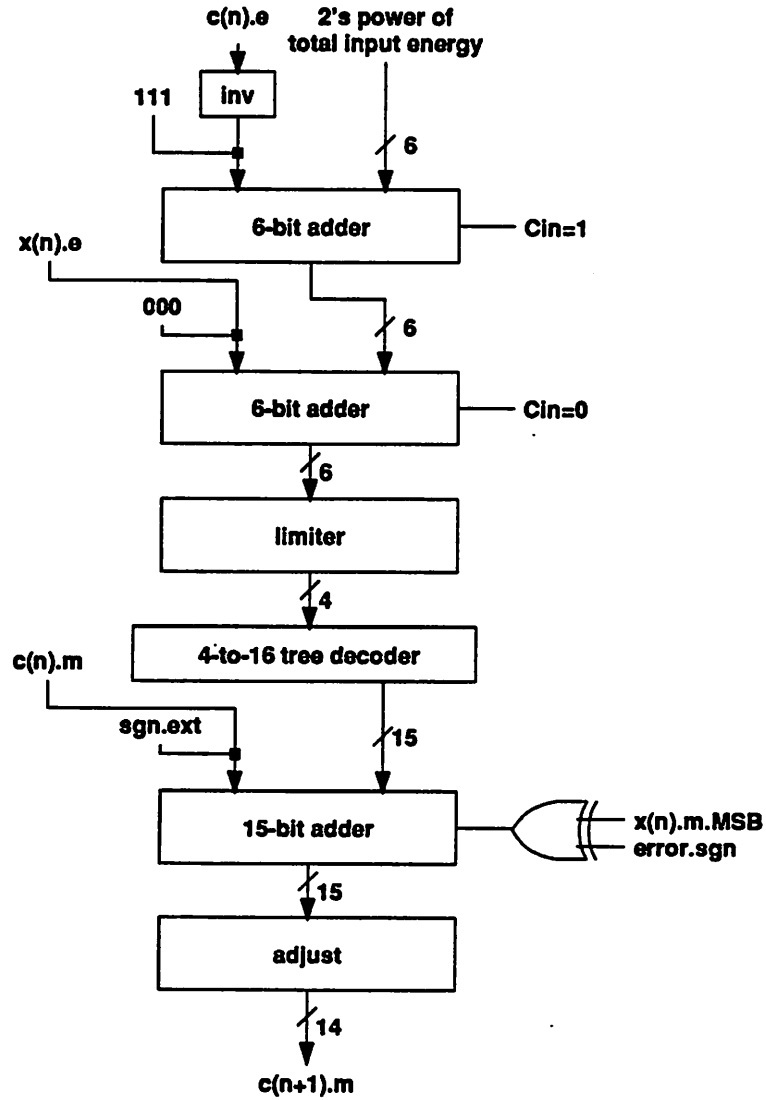


Figure 5.10 - Data path for adaptation

As pointed out in the previous chapter, the coefficients are encoded in floating point format, equation (5.4a) can be described as:

$$\begin{aligned}
 c_i(n+1) &= c_i(n) \pm 2^{\eta+Px} \\
 &= m \cdot 2^e \pm 2^{\eta+Px} \\
 &= (m \pm 2^{\eta-e+Px}) \cdot 2^e
 \end{aligned} \tag{5.4b}$$

The adaptation process outlined in Chapter 4, therefore, consists of the following

computations:

- (1) a subtractor and an adder to compute $\eta - e + Px$;
- (2) a magnitude limiter and a decoder to generate $2^{\eta - e + Px}$;
- (3) an adder to obtain $m \pm 2^{\eta - e + Px}$;
- (4) a final adjustment to keep $m \pm 2^{\eta - e + Px}$ in the proper range.

To filter out the noise generated in the adaptation process, an additional 8 buffer bits are added to the mantissa of the coefficient. However, these buffer bits are not used in the convolution computation. The complete adaptation data path is shown in Figure 5.10.

5.4.2 The Convolution Processor

The computation convolution performs a function of multiply-and-accumulate. Because the data and the coefficients are encoded in floating point format, a barrel shifter is needed to convert the floating point output into a fixed point format for the accumulation. The key elements in the convolution processor are a 6x6 multiplier, a 12-bit by 14-position barrel shifter, a 4 bit adder, and a 24 bit accumulator. Because the convolution processor is also used for computing the total input signal power, time-sharing is done through a set of multiplexers with appropriate control signals.

The functional diagram of the convolution processor is shown in Figure 5.11. To compute the total input signal power, the oldest sample is deleted at the same time the newest sample is added. The multiplexing is controlled by C0. The left shift (SHL) operating on the exponents squares the values of the data samples. C1 and C2 decide whether the computation is for convolution or total input power estimation. C3 and C4 reset the accumulator before the convolution takes place. In the case of total input power estimation, an old estimate has to be loaded into the accumulator. This is done by C5, which connects the configuration of the 24

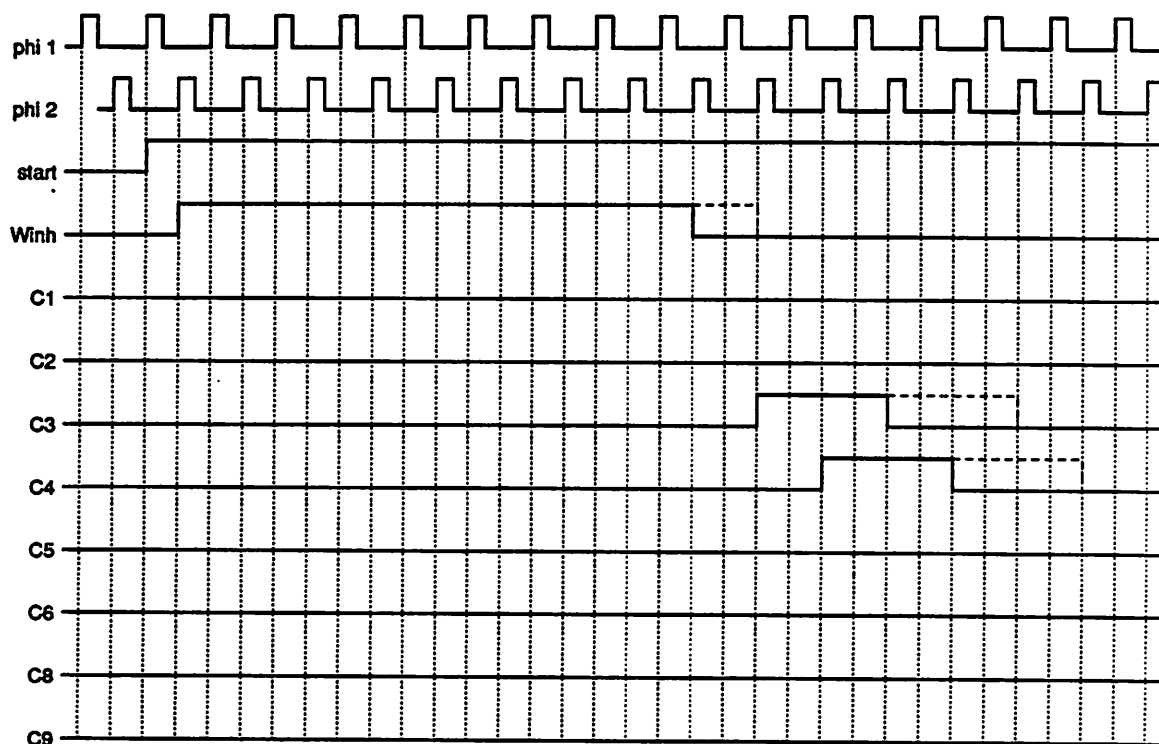


Figure 5.12 - Timing diagram at the beginning of convolution

bit adders. This is because the regularity of the convolution doesn't require any intermediate accumulation result. Consequently, at the end of the convolution, 3 more clock cycles are needed to propagate the carries.

The 6x6 bit multiplier is a Booth decoded array multiplier. To achieve the 8 MHz speed requirement, carry select is used in the last stage of the multiplier to minimize the carry delay. The core of the barrel shifter is an array of NMOS transistors to save area. Precharge is used to speed up the logic evaluation.

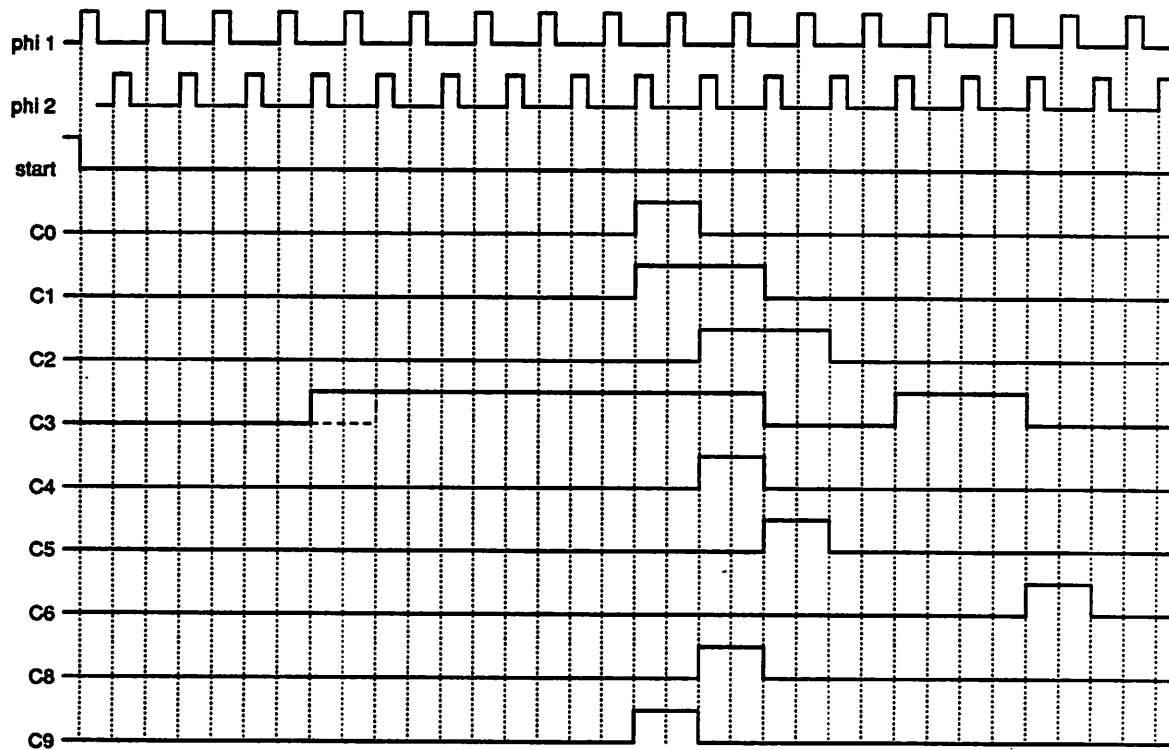


Figure 5.13 - Timing diagram at the end of convolution

5.5 Layout and Fabrication

5.5.1 Floor Plan

A floor plan (Figure 5.14) was designed in the beginning and continuously modified as more functional blocks were being laid out. Signals run from left to right and controls from bottom to top. Because the convolution processor is also used for computing total input signal power, a time-sharing control unit (to generate C0 - C9) is needed. The convolution processor is placed between the adaptation processor and the control units because the adaptation processor doesn't require any additional controls other than the two phase clocks for pipelining. A bus running on top allows the chip to execute the adaptation and the convolution simultane-

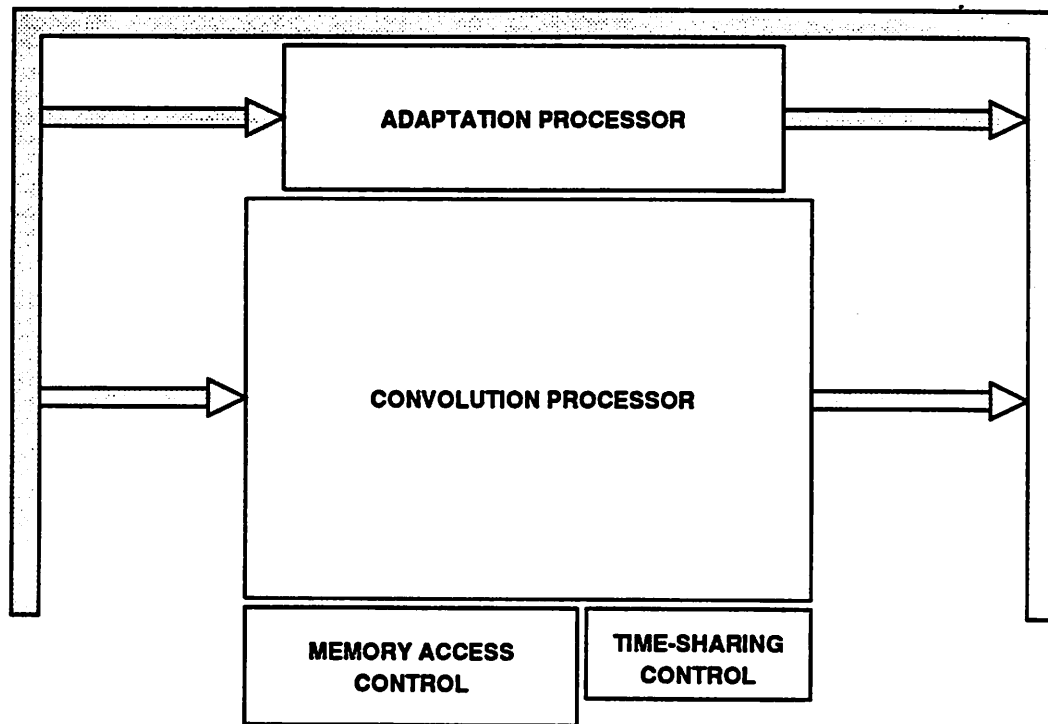


Figure 5.14 - Floor plan of the chip

ously. Memory is not included in this chip but the address and the read/write signals for memory access are generated on the chip.

5.5.2 Basic Cell Designs

In the cell designs, we emphasize modularity and regularity. As an example, in a 1 bit full adder (Figure 5.15), the signals are running from left to right and the carries from top to bottom. The and-or-invert logic gate generates the complementary carry out signal. The purpose of generating this complementary carry-out is to limit the carry-in to carry-out delay to only one gate delay. In a typical 3 μm process, this delay is about 3 ns to 5 ns. As a result, for an 8 bit adder, a simple cascade of eight 1 bit adders is good enough without resorting to the more complex but less regular carry look-ahead scheme. Because of the complementary carry

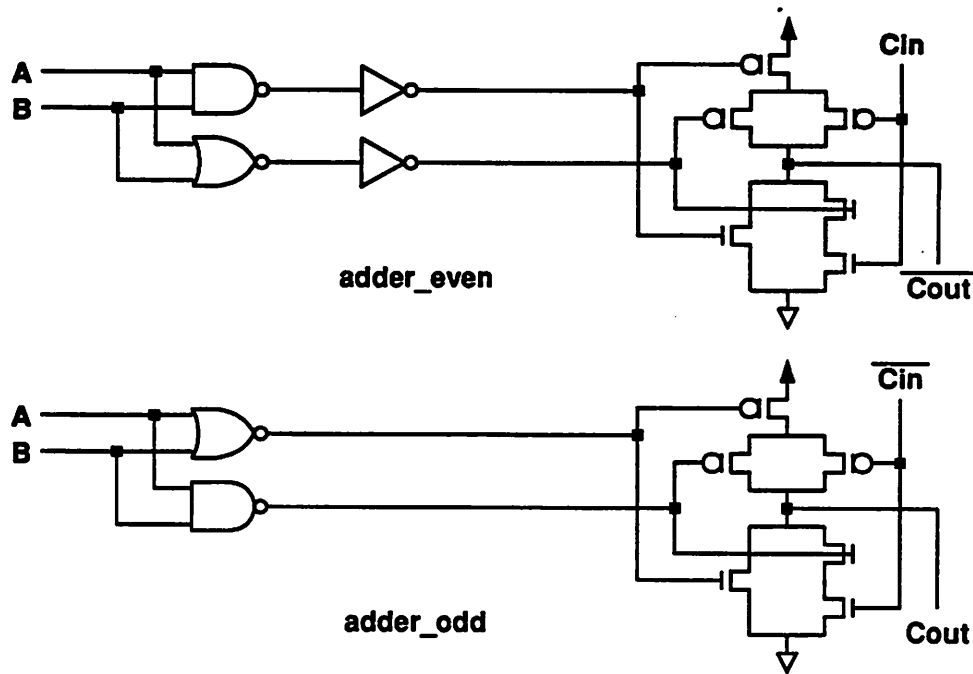


Figure 5.15 - Adder cell with 1 gate delay for carry chain

signals, two types of adders are needed with each complementary to the other. However, the design effort is minimal since their circuitry is quite similar.

Another example is the cell design of a synchronous counter (Figure 5.16). The exclusive-OR circuit serves as a half adder. When the carry-in is high, the output will change state in the following clock cycle. If both the carry-in and the output are high, a complementary carry-out is generated. Again, this is to reduce the carry-in to carry-out delay to one gate delay. Therefore, two types of counter cells are needed. However, the symmetry of these two cells means that very little effort is needed to generate one from the other. The clock, reset, and count signals are running vertically so that several cells can abut against one another. Because the carry-in to carry-out is only one gate delay, a 10 bit counter can be constructed without requiring more complex and less regular schemes to meet the timing specs. It can be seen from these two examples that the demand for modularity and regularity by the higher

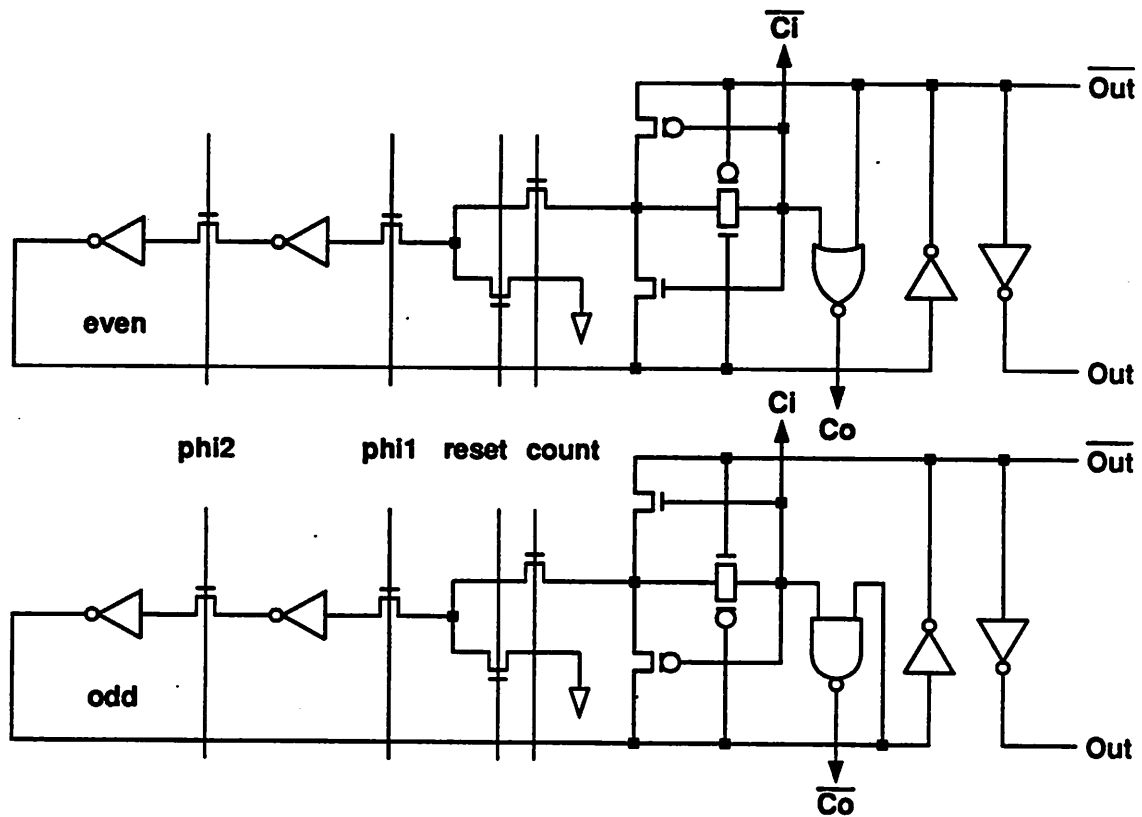


Figure 5.16 - Counter cell with 1 gate delay for carry chain

hierarchy has greatly shaped the basic cell design.

5.5.3 Layout Verification and Simulation

After the layout of the basic cells, the electrical circuit being implemented by the layout is extracted, which includes the stray capacitance associated with each node. Circuit simulations (e.g., using SPICE) to characterize the speed of these basic cells are performed. The results are used throughout the complete chip implementation.

The prototype is divided into 6 major modules. They are the adaptation processor, the convolution processor, the memory interface unit, the time-multiplexing control unit, the μ -255 interface, and the pipeline queuing circuit. The pipeline queuing circuit (Figure 5.17)

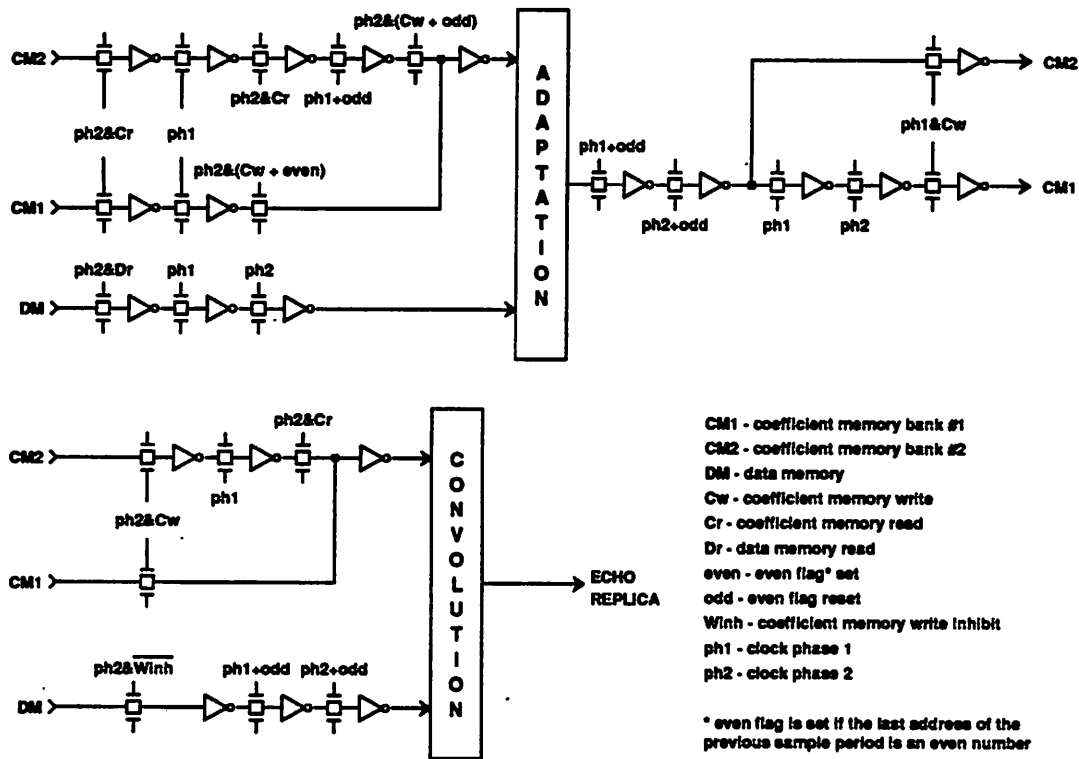


Figure 5.17 - Pipeline queue circuits to the adaptation and the convolution processors

provides the queuing path to guide two coefficients into the pipelined processors and generate two updated coefficients in parallel.

After each module is implemented and extracted, switch level logic simulation is done by esim or bdsim. Because the memory interface unit and the time-multiplexing control unit use complicated circuit schemes to generate timing sequence, circuit simulations (in this case, using RELAX) are conducted. These simulations revealed some race conditions that were overlooked in the original design and identified some glitch signals.

A final routing of interconnections was made after all 6 modules had been completed and simulated. The complete circuit was extracted from the final layout. To logically simulate the chip as an echo canceler, a subtractor circuit to compute the residual error between the echo replica and the echo was loosely laid out with its terminals and cell names properly

labeled. These labels allowed the extracted circuit description file to be concatenated directly with that of the chip. Because of the repetitiveness of the static RAMs, a C language program was written to generate the circuit description file of the memory. This is compatible (in terms of labels) with that generated by the extractor. Caution was taken to ensure that the unique designation of the node names such that the memory description file can be concatenated with the circuit description file of the chip. The combined file was then used as input to simulator bdsim. The results after each sample cycle were compared with those generated by the C language program that describes the hardware implementation discussed in Chapter 4. The simulation showed a complete agreement between these two and the implemented circuit indeed behaved like an echo canceler.

Some layout errors surrounding the wells, the substrate contacts, and the well contacts cannot be detected by the Magic extractor and the switch level simulator. Examples include a floating well and a P-well contact connected to Vdd. The extract section of the Magic technology file provides a convenient way to detect these errors. The general idea is to force all capacitance values zero except the one under examination and then extract the circuit. As an example, if all the capacitance values except the P-well/well-contact (they can be disconnected forcibly in the technology file of Magic) are zeroes and the extracted circuit shows a capacitance between a P-well and Vdd (instead of GND), that particular P-well has a well contact connected to Vdd. Because Magic will automatically generate wells when converting to CIF format, it is necessary to check these errors on the final layout in the CIF format. The CIF format can be converted back to the Magic format, which is then extracted and checked for layout errors involving wells and well contacts. This process is time consuming because a complete layout has to be extracted for each type of error checked.

Because Magic considers two labeled Vdd lines as physically connected, any unconnected Vdd lines can't be detected in the simulation. Although Magic gives warnings to

nodes with the same label inside a cell, it does so for every repeatedly called cell regardless of whether the nodes are connected outside the cell. To check all these warnings would be like visually checking all the Vdd connections. Two unconnected Vdd lines were later found in the fabricated chip (they were still connected to the Vdd through substrate and substrate contacts.) A procedure was later developed to cope with this problem. After the layout and the extracted circuit had successfully passed the simulation, we modified all the Magic files such that the 2nd Vdd labels inside a cell was renamed to Vdd2 and the 3rd one to Vdd3, etc. Similarly, GND labels were also renamed. This was done automatically by a C program. The modified Magic files were then extracted again. The extracted hierarchical representation can be converted to a flattened representation using the program ext2sim. At the same time, a list of nodes is also created. If all the Vdd lines inside any cell are connected to the global Vdd bus, there will be only one Vdd node on the list. Otherwise, a pattern such as "Vddx" (for example, Vdd2) will appear on the list with a prefix clearly indicates which cell it belongs to and, therefore, that particular Vdd line is not connected to the global Vdd bus. Using this scheme, the two unconnected Vdd lines found in the fabricated chip were detected.

5.5.4 Fabrication

After successfully passing logic simulation, the chip layout implemented by Magic was converted to the CIF format for fabrication. The layout is done in a λ -based design rules system. As a result, with the exception of the pads, the layout can be fabricated by MOSIS using scalable CMOS technology. We chose to have it fabricated by a 3μ , double-metal, P-well process. Because the layout generically includes P-well and N-well, an N-well process can also be used. The choice of a particular technology is specified in the conversion from the Magic representation to the CIF format.

5.6 Measurements and Discussions

Because of the complexity of this chip, it is desirable that each functional module can be tested individually. The purpose is to test the chip and debug the test board simultaneously. Several self-test programs were embedded in the test board such that any new changes to the test board may be compared with prior results. The complete chip was evaluated as an echo canceler until every module was checked up.

5.6.1 Measurement Setup

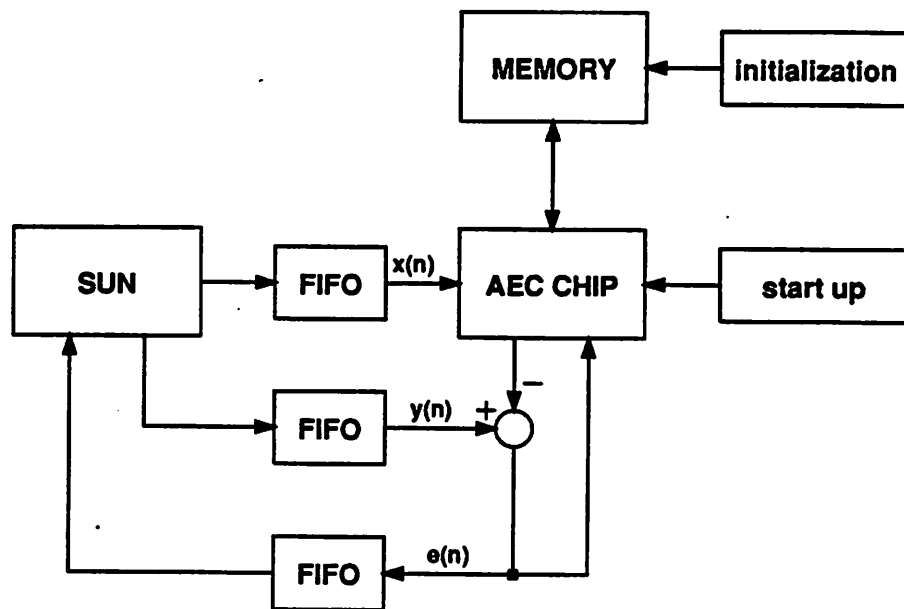


Figure 5.18 - Configuration of the test setup

The measurement of the prototype AEC chip is carried out on a Sun workstation with a multi-bus compatible parallel I/O board, which allows data issuance and acquisition. Figure 5.18 illustrates the test setup. The first-in-first-out's (FIFO's) serve as data buffers between the AEC chip and the Sun because they are running from two independent clocks. The Sun

sends the far-end signal ($x(n)$) and the returned far-end talker echo ($y(n)$) to the AEC chip. The residual error, $e(n)$, between the AEC generated echo replica, $\hat{y}(n)$, and the returned far-end talker echo, $y(n)$, is fed back to the AEC chip and recorded by the Sun. Three banks of static RAMs are connected to the bus of the AEC chip. Among them are a 9 bit wide memory bank for the far-end data and two 17 bit wide memory banks for the coefficients. The memory is initialized through an initialization circuit, that sets zero to all bits in all memory locations. Because the number of taps of the AEC chip is programmable, a start up circuitry properly programs the AEC chip to the desired number of taps.

The AEC chip has a test pin, which can set the chip in the test mode. The test signals that initialize the chip and select the number of taps are sent to the chip through the error pins (e0 - e2) in the test mode. The timing diagram of the initialization and the tap selection signals are shown in Figure 5.19 where n is the number of taps subject to the restrictions that n being an even number and $n+1$ being a multiple of 9. Among these test signals, "load" presets the starting pointer of the program counter; "test1" down loads the number of taps to the tap register; and "test2" resets the 10 bit counter after the power is turned on. At the end of the test period, "strobe" is issued to start the normal operation by fetching the far-end signal and the returned far-end talker echo into the AEC chip. This strobe signal is activated thereafter with a period equal to 1024 basic clock cycles and a 50% duty cycle.

The test board and the AEC chip use non-overlapping two-phase clocks. The generation of these two clock phases is shown in Figure 5.20. The flip-flop serves as a divide-by-2 counter and ensures that the active periods for both $\text{ph1 } (\Phi1)$ and $\text{ph2 } (\Phi2)$ are about the same in length. After the power-on-reset (POR), the following sequence of events occurs:

(I) (1) $A=0$ and $B=1$ (2) $C=1$ (3) $\text{ph1}=0$ (4) $D=0$ (5) $\text{ph2}=1$

When a master clock pulse comes in, the following sequence of events occurs:

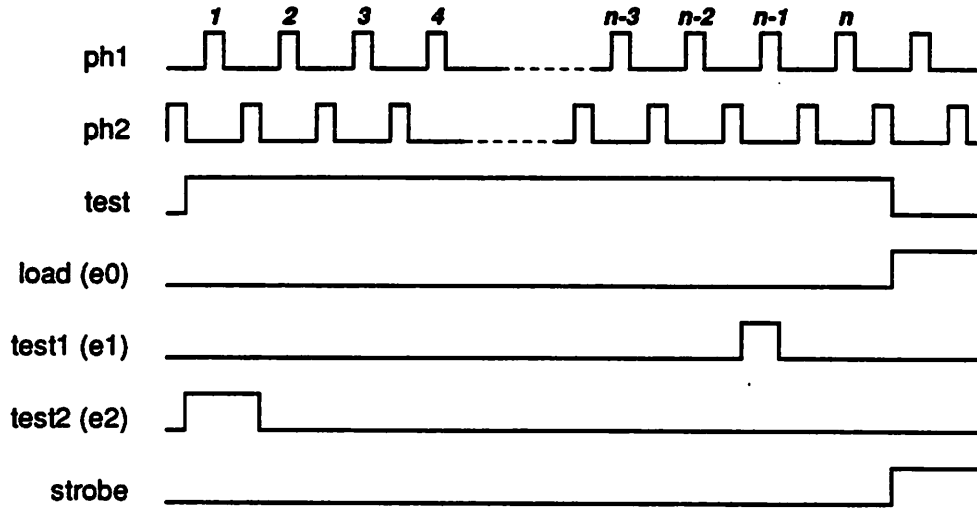


Figure 5.19 - Timing diagram of the initialization signals

(II) (1) A=1 and B=0 (2) D=1 (3) ph2=0 (4) C=0 (5) ph1=1

Thereafter, sequences (I) and (II) appear alternatingly in synchronization to the phase change of the master clock. The resultant timing diagram is also shown in Figure 5.20. The separation (non-overlapping) between the active periods of ph1 and ph2 is determined by the time delay through the gates and the delay modules. The delay module is a passive LC delay line with 10 equally spaced taps made by Data Delay Devices, Inc.

Because the 12 bit echo replica generated by the AEC chip is time-multiplexed to save the I/O pins, a demultiplexer shown in Figure 5.21 properly aligns the data. The "DA_latch" flag generated by the AEC chip is set when the upper 6 bits of the echo replica are available and reset for the lower 6 bits. Two latch signals, A and B, are derived from "DA_latch" (after it has settled) to avoid the relative delays between the time-multiplexed outputs of the echo replica and the "DA_latch" flag, and, therefore, prevent the data from being latched incorrectly.

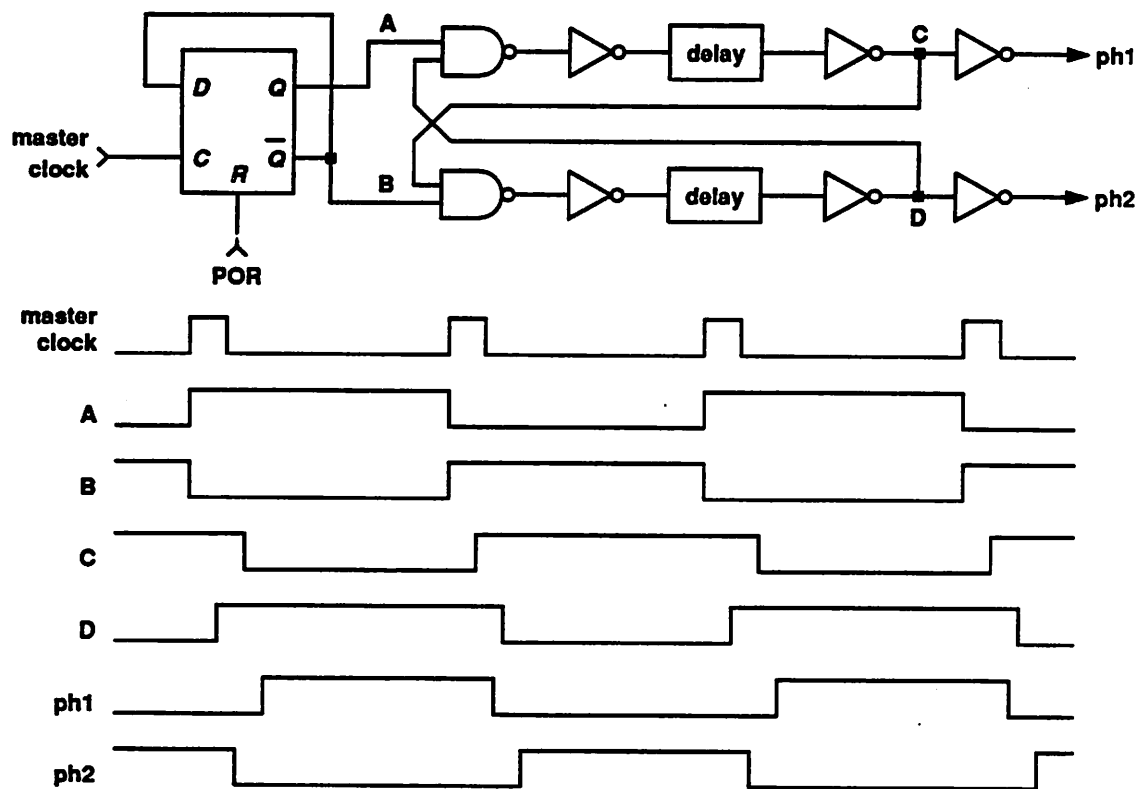


Figure 5.20 - Generation of non-overlapping clocks

5.6.2 Experimental Results

In the functional test, we are interested in the initial convergence and the steady state echo return loss enhancement (ERLE) of the AEC chip. The contents of the external memory banks are initially reset to zero. The talker signal is Gaussian distributed. The mean of the talker signal is zero and the standard deviation (σ) is set to a value such that 4σ will saturate a 13 bit register (a μ -255 code is equivalent to a 13 bit linear code.) Because the full-scale magnitude of a 13 bit linear code (including the sign bit) is 4096 least significant bits (LSBs), the standard deviation of the test signal is 1024 LSBs. 50,000 such data samples are generated.

An exponentially decayed impulse response for the echo path is assumed. To avoid generating the impulse response of a single pole filter, the exponential sequence is multiplied by a

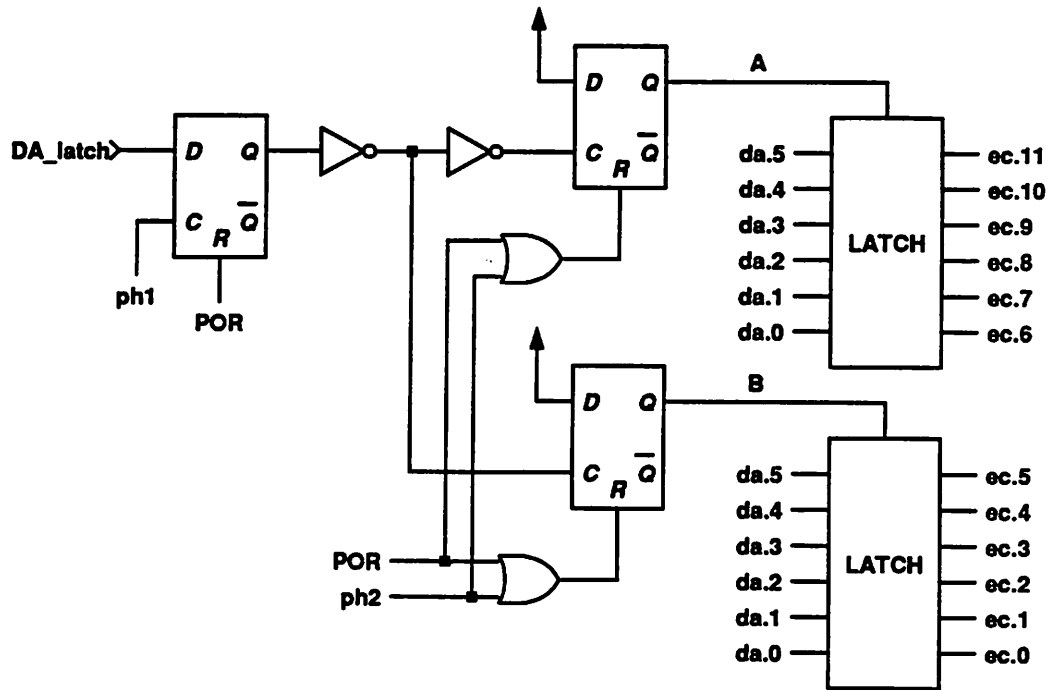


Figure 5.21 - Decoding the time-multiplexed echo replica outputs

binary pseudo-random sequence, $r(i)$.

$$h_i = r(i) \frac{K}{(-1.00346)^i} \quad 0 \leq i < 1000 \quad (5.5)$$

where K is a constant such that $\sum h_i^2 = 1$. This sequence is equivalent to an echo impulse response with a reverberation time of 250 ms. The condition that $\sum h_i^2 = 1$ represents a worst scenario in which the total reflection energy received by the microphone is equal to the total energy transmitted by the loudspeaker.

The echo signal is calculated by convolving the aforementioned data samples with the impulse response. The resultant echo signal is then coded into a 13 bit linear format. Similarly, the data samples are coded by a μ -255 coder. The coded data samples are sent to the AEC chip and converted into a floating point format. The coded echoes are compared with the generated echo replica. The errors are recorded and also fed back to the AEC chip to update

the coefficients.

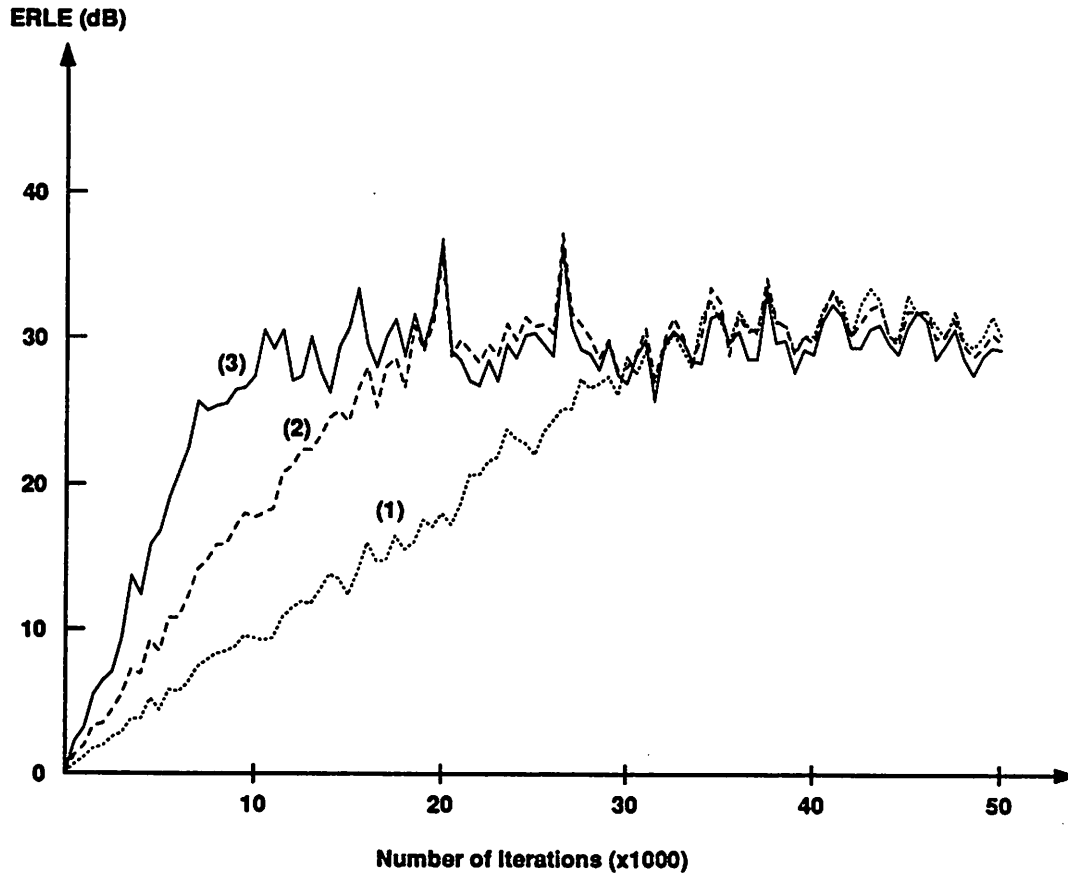


Figure 5.22 - Measurement of ERLE and convergence time: (1) $\beta=.25$ (2) $\beta=.5$ (3) $\beta=1$

Figure 5.22 shows the experimental results of the ERLE measurement with three different values of β . For β equal to 1, the convergence time is about 1 second and the steady state ERLE is no less than 27 dB. The convergence time for $\beta=.5$ and $\beta=.25$ are 2 seconds and 4 seconds respectively. This result agrees with our analysis in Chapter 4, which predicts that the convergence speed is proportional to β . The steady state ERLE with $\beta=.25$ is better than that with $\beta=.5$ and $\beta=1$. However, the improvement is not substantial (in spite of the reduction in step size with smaller β) because the errors are dominated by the quantizations due to the power-of-two multiplications. Table 5.2 summarizes the chip performance.

Table 5.2 Summary of the Chip Performance

Item	Specification
power supply	5 V
power consumption	30 mw
internal clock	4 MHz
number of taps	1000
ERLE	> 27 dB
initial convergence time	< 1 sec.
signal interface	μ -law compatible

The die photo of the chip is shown in Figure 5.23. The chip was fabricated by MOSIS using 3 μ m, double-metal, P-well, CMOS technology. The core area of the chip is 5.3mm by 5.3 mm and can be scaled down for a 2 μ m process. The estimated area of the 26k bits memory using 3 μ m technology is about 6.3mm by 6.3mm. Therefore, a complete acoustic echo canceler including the memory in a single IC chip is possible.

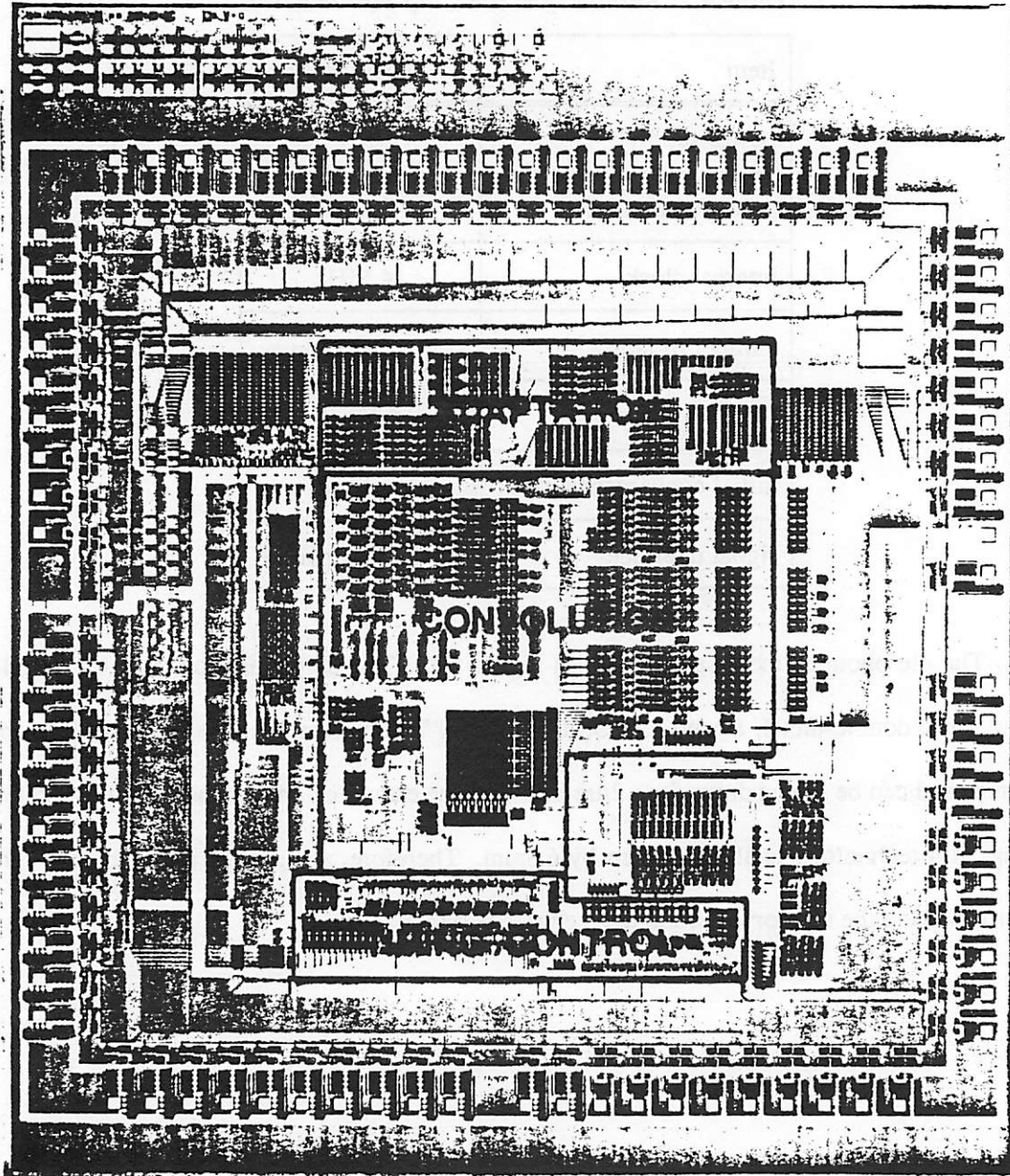


Figure 5.23 - Die photo

CHAPTER 6

CONCLUSIONS

This research shows that echo cancellation is of potential interest for removing far-end talker echoes in a loudspeaker telephone. A single-chip acoustic echo canceler in 3 μm CMOS technology has been designed to show the feasibility of a low-cost solution to the problem of far-end talker echoes. The summary of research results is presented in Section 6.1. Suggestions on further work in the related areas are contemplated in Section 6.2.

6.1 Summary of Research Results

The echo problems associated with loudspeaker telephones can be characterized by far-end talker echoes and near-end talker echoes. The far-end talker echoes cause the far-end talker to hear a delayed version of his own speech and pose potential instability or singing. The near-end talker echoes reduce the signal-to-noise ratio and result in transmitted reverberation.

The conventional voice switching increases the stability margin, but the inherent half-duplex transmission casts an insurmountable barrier in quality improvement. Echo cancellation, on the other hand, offers the promising prospect to cope with the problem of far-end talker echoes while maintaining the full-duplex transmission capability.

A multimicrophone approach has been the effective means of solving the problem of near-end talker echoes. However, the heavy computational requirements and the need of a minimum separation distance among the microphones prevent it from becoming a low-cost option. The use of training signals and the inverse filtering (though yet to be proven useful)

offer plausible alternatives. These two methods also make use of the acoustic echo canceler structure designed for the removal of the far-end talker echoes.

An acoustic echo canceler needs a 125 ms process window, an echo return loss enhancement over 25dB, and a dynamic range more than 40dB to reduce the far-end talker echoes in typical rooms. Balancing performance and complexity, use of a floating point data representation and power-of-two multiplication in coefficient update can achieve 27dB echo return loss enhancement with reasonable hardware requirements. The proposed method saves memory and chip area compared to a direct approach.

A CMOS test chip has been designed and fabricated to show the feasibility of this method. A highly structured design process and a well coordinated design environment are developed in this project, which makes the fast design of this chip possible. The core area of the chip is about 5.3 mm by 5.3 mm using MOSIS 3 μ m double metal process and will be 3.5 mm by 3.5 mm if 2 μ m technology is used. (The chip has been laid out using scalable design rules.) Although the memory is not on the chip, a complete echo canceler including the memory in a single IC chip is possible.

Experimental results show a 27 dB echo reduction being achieved after 1 second of convergence time, which is consistent with the computer simulations.

6.2 Future Work

An immediate extension of the present work is to evaluate the subjective performance of the acoustic echo canceler. Although a near-end talker detection scheme was proposed in this research, its effectiveness was not tested. Near-end talker detection is particularly critical if the loudspeaker telephone is used in a noisy environment.

Although we were not successful in finding filter structures other than the transversal filter in realizing the adaptive echo canceler, continuing efforts to reduce the filter complexity

by exploiting different structures are needed. In our initial attempts, we used an echo canceler to remove the early echoes and a low rate echo canceler to reduce the later echoes (assume the acoustic environment has a low pass characteristics.) This attempt met its doom when we discovered that some materials (for example, a glass window) had demonstrated high pass behavior. However, the physical origin of sound reflections (room acoustics) may still hold the key to the discovery of a new filter structure that will have much reduced complexity.

Of course the speech itself contains unique information that might be used to its advantage. One conjecture is that the LPC parameters can be correctly estimated despite the echoes (Chapter 2, Section 2.4.2.) Based on this assumption, an inverse filtering is plausible. If this method should be successful, not only the near-end talker echoes can be alleviated but the complexity of the echo canceler to remove the far-end talker echoes can be greatly reduced.

Other potential research topics include frequency domain filtering using sub-band coding (so that subjective weighting can be applied) and auditory modeling on perception.

References

1. Mark B. Gardner, "A Study of Talking Distance and Related Parameters in Hands-Free Telephony," *The Bell System Technical Journal*, pp. 1529-1551, November, 1960.
2. W. F. Clemency and W. D. Goodale, Jr., "Functional Design of a Voice-Switched Speakerphone," *The Bell System Technical Journal*, vol. XL, pp. 649-668, May, 1961.
3. Juro Ohga and Shokichiro Yoshikawa, "Study of Howling in Speaker Phone Telephony," *Review of the Electrical Communication Laboratories*, vol. 20, pp. 690-697, July-August, 1972.
4. J. P. A. Lochner and J. F. Burger, "The Intelligibility of Speech Under Reverberant Conditions," *Acustica*, vol. 11, pp. 195-200, 1961.
5. J. P. A. Lochner and J. F. Burger, "The Influence of Reflections on Auditorium Acoustics," *Journal of Sound & Vibration*, vol. 1(4), pp. 426-454, 1964.
6. Robert E. McFarlane and Robert J. Nissen, "Room Design and Engineering for Two-Way Teleconferencing," *The Teleconferencing Resource Book: A Guide to Application and Planning*, pp. 155-170, Elsevier Science Publishers B.V. (North-Holland).
7. J. W. Emling, "General Aspects of Hands-Free Telephony," *A.I.E.E. Trans.*, vol. 76, Part I, pp. 201-205, May, 1957.
8. J. B. Allen, D. A. Berkley, and J. Blauert, "Multimicrophone Signal-Processing Technique to Remove Room Reverberation from Speech Signals," *Journal of the Acoustical Society of America*, vol. 62 (4), pp. 912-915, October, 1979.
9. L. R. Rabiner and R. W. Schafer, *Digital Processing of Speech Signals*, p. 177, Prentice-Hall, 1978.
10. A. Busala, "Fundamental Considerations in the Design of a Voice-Switched Speakerphone," *The Bell System Technical Journal*, vol. XXXIX, pp. 265-294, March, 1960.

11. B. Copping and R. G. Fidler, "Designing a Voice-Switched Loudspeaking Telephone - Loudspeaking Telephone No. 4," *P.O.E.E.J.*, vol. 60, pp. 65-71, 1967.
12. P. T. Brady, "A Statistical Analysis of On-Off Patterns in Sixteen Conversations," *Bell System Technical Journal*, vol. 47, pp. 73-92, 1968.
13. N. H. Morgan, "Room Acoustics Simulation with Discrete-Time Hardware," *Ph.D. Thesis*, University of California, Berkeley, Berkeley, CA 94720, December, 1980.
14. J. L. Flanagan and R. C. Lummis, "Signal Processing to Reduce Multipath Distortion in Small Rooms," *Journal of the Acoustical Society of America*, vol. 47, pp. 1475-1481, June, 1970.
15. J. L. Flanagan, "Beamwidth and Usable Bandwidth of Delay-Steered Microphone Arrays," *AT&T Technical Journal*, vol. 64, pp. 983-995, April, 1985.
16. Heinrich Kuttruff, *Room Acoustics*, Applied Science Publishers Ltd., London, UK, 2nd edition, 1973.
17. Jont B. Allen and David A. Berkley, "Image Method for Efficiently Simulating Small-Room Acoustics," *Journal of the Acoustical Society of America*, vol. 65(4), pp. 943-949, April 1979.
18. L. A. Parker and C. H. Olgren, *The Teleconferencing Resource Book: A Guide to Applications and Planning*, North Holland, 1984.
19. Michael L. Honig and David G. Messerschmitt, *Adaptive Filters: Structures, Algorithms, and Applications*, pp. 49-56, Kluwer Academic Publishers, Boston, 1984.
20. Private communication, *Digital Voice Echo Canceller on a TMS32020*, Teknekron Communications Systems.
21. Bell Laboratories Staff, *Transmission Systems for Communications*, p. 624, Bell Telephone Laboratories, Inc., 5th edition, 1982.

22. Donald L. Duttweiler, "A Twelve-Channel Echo Canceler," *IEEE Transactions on Communications*, vol. COM-26, No. 5, pp. 647-653, May, 1978.
23. Donald L. Duttweiler, "Adaptive Filter Performance with Nonlinearities in the Correlation Multiplier," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. ASSP-30, No. 4, pp. 578-586, August, 1982.
24. T. Kawasaki and S. Minami, "Experimental Results of Improved Acoustic Coupling Using Echo Canceler," *IECE Japan, National Convention Record*, vol. No. 2344, Spring, 1984.
25. 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, pp. 504-510, August, 1978.
26. O. Agazzi, D. A. Hodges, and D. G. Messerschmitt, "Large-Scale Integration of Hybrid-Method Digital Subscriber Loops," *IEEE Transactions on Communications*, vol. COM-30, pp. 2095-2108, September, 1982.
27. O. Agazzi, D. G. Messerschmitt, and D. A. Hodges, "Nonlinear Echo Cancellation of Data Signals," *IEEE Transactions on Communications*, vol. COM-30, pp. 2421-2433, November, 1982.