

Copyright © 1982, 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.

LARGE SCALE INTEGRATION OF HYBRID-METHOD
DIGITAL SUBSCRIBER LOOPS

by

O. E. Agazzi

Memorandum No. UCB/ERL M82/41

20 May 1982

(cover)

LARGE SCALE INTEGRATION OF HYBRID-METHOD
DIGITAL SUBSCRIBER LOOPS

by
Oscar Ernesto Agazzi

Memorandum No. UCB/ERL M82/41

20 May 1982

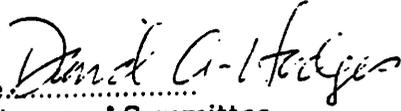
ELECTRONICS RESEARCH LABORATORY
College of Engineering
University of California, Berkeley
94720

ABSTRACT**LARGE SCALE INTEGRATION OF HYBRID-METHOD
DIGITAL SUBSCRIBER LOOPS**

by

Oscar Ernesto Agazzi

PhD.

Dept of Electrical
Engineering and
Computer SciencesSignature 
Chairman of Committee

This dissertation reports on studies concerning the feasibility of large-scale integrated realization of the circuits needed to provide hybrid-mode full-duplex digital transmission at 80 kb/s or higher rates over standard local telephone loops. Alternative means of achieving the required 60 dB or so of echo cancellation have been studied in detail. The conclusion is that a combination of analog and digital circuit techniques permit practical MOSLSI realization of the complete modem, including filters, echo canceller, timing recovery, and A/D and D/A converters, without need for external circuit elements, trimming, or adjustments.

The preferred system configuration has been evaluated by means of analysis, simulation, and laboratory and field measurements. A complete full-duplex system, including an experimental NMOS integrated circuit echo canceller, was built and tested. Measurements showed a bit error rate lower than 10^{-8} with line attenuation up to 40 dB, operating at 80 kb/s. We conclude that a fully-integrated MOSLSI circuit to implement all functions for a hybrid-mode digital local loop is entirely feasible.

ACKNOWLEDGEMENTS

I am very grateful to Professors David A. Hodges and David G. Messerschmitt for the opportunity to participate in their research program on integrated circuits for telecommunications. Their technical guidance and assistance are largely responsible for the success of this project. I am also very grateful to Prof. Paul R. Gray for many useful discussions and assistance. Talks with other graduate students at the Integrated Circuits group have always been highly beneficial.

I want to acknowledge the Argentine Research and Development Naval Service for awarding me the fellowship that made possible my PhD. program.

This research was supported by the National Science Foundation under Grant ENG78-11397, by a grant from TRW-Vidar, and by the University of California Microelectronics Innovation and Computer Research Opportunities Program (MICRO), with matching grants from Racal-Vadic, Fairchild, and GTE.

DEDICATION

I want to dedicate this thesis to my parents, whose inspiration and encouragement made this result possible.

TABLE OF CONTENTS

CHAPTER 1: INTRODUCTION	1
1.1. Comparison Between Burst-Mode and Hybrid Systems	12
CHAPTER 2: SYSTEM DESIGN	16
2.1. Design Parameters and Techniques	16
2.1.1. Error Rate Objectives	16
2.1.2. Echo Cancellation	20
2.1.2.1. Interpolating Echo Cancellers	24
2.1.3. Required Amount of Echo Cancellation	27
2.1.4. Channel Shaping	27
2.1.5. Equalization	35
2.1.6. Combining Echo Cancellation and Decision Feedback Equalization	42
2.1.7. Timing Recovery	46
2.1.8. Synchronization in Subscriber and Central Office Sets	48
2.2. System Description	48
2.2.1. Scrambler and Descrambler	50
2.2.2. Bipolar Coder	52
2.2.3. Transmit and Receive Filters	53
2.2.4. Equalizer	58
2.2.5. Echo Canceller	58
2.2.6. Detector	59
2.2.7. Reconstruction Filter	59
2.2.8. Full-Wave Rectifier	59
2.2.9. Bandpass Filter	59
2.2.10. Frequency Multiplier	60
CHAPTER 3: NONLINEAR ECHO CANCELLATION TECHNIQUES	61
3.1. A Binary Series Expansion of a Nonlinear Function	67
3.1.1. Expansion of Incompletely Specified Functions	72
3.1.2. Expansion in Terms of Other Binary Variables	75
3.1.3. Some Important Properties of the Nonlinear Expansion	75
3.2. Application to Echo Cancellation	79
3.2.1. Multilevel Transmitted Signals With Linear Canceller	80
3.2.2. Nonlinear Channel With Nonlinear Canceller	84
3.2.3. Line Codes With Memory	87
3.2.4. Adaptation Algorithm	90
3.2.5. Truncation of the Series Expansion	93
3.3. Cancellation MSE With MOS D/A Converter	95
3.4. Application to Timing Recovery	102
CHAPTER 4: TECHNIQUES FOR THE LARGE SCALE INTEGRATION OF THE SYSTEM	109
4.1. Alternative Implementations of the Echo Canceller	110
4.1.1. Analog Echo Canceller	110
4.1.2. Fully Digital Echo Canceller	115
4.1.3. Digital Echo Canceller With Analog Cancellation	115

4.1.4. Analog/Digital Echo Canceller	116
4.2. Resolution Requirements of the Data Converters	116
4.2.1. Resolution of the A/D Converter of Fig. 4.1a	118
4.2.2. Resolution of the A/D and D/A in Figs. 4.1b and 4.1c	118
4.3. Effect of D/A Nonlinearity	120
4.3.1. Nonlinear Distortion in Fig. 4.1c	122
4.3.2. Nonlinear Distortion in Fig. 4.1d	124
4.3.3. Nonlinear Distortion in Fig. 4.1b	127
4.4. Experimental Realizations	130
4.4.1. Analog Echo Canceller	130
4.4.1.1. Capacitor Matching	136
4.4.1.2. Offset, Low Gain, and Nonlinearity of the Summing Amplifier	137
4.4.1.3. Offset of the Integrators, Clock Feedthrough, and Leakage Currents	137
4.4.2. Digital Echo Canceller	141
4.4.3. Digital Echo Canceller With Analog Cancellation	146
4.4.4. Analog-Digital Echo Canceller	146
4.5. Experimental Results	154
CHAPTER 5: CONCLUSION	161
5.1. Digital Processor	161
5.1.1. Digital Processor Operation	163
5.1.1.1. Reset Cycle	163
5.1.1.2. Echo Canceller Convergence Cycle	163
5.1.1.3. DFE Convergence and Normal Operation	163
5.1.2. Processor Architecture	163
5.1.3. State and Timing Diagrams	165
5.2. Cancellation Block	168
5.3. Timing Recovery Block	171
5.4. Input and Output Filters	172
5.5. Concluding Remarks	172
APPENDIX A: MINIMAL INTERSYMBOL INTERFERENCE FILTERS	183
APPENDIX B: EFFECT OF PULSE ASYMMETRY ON THE SPECTRUM OF A BIPOLAR SIGNAL	188
APPENDIX C: NMOS SILICON GATE PROCESS	190
APPENDIX D: DIGITAL PROCESSOR DOCUMENTATION	193
REFERENCES	200

CHAPTER 1

INTRODUCTION

The telephone network has experienced a rapid technological evolution in recent years. The highlight of this progress has been the introduction of digital techniques first in the transmission and then in the switching areas. In the early 1960's digital communications proved to be a viable transmission system for short haul trunking. The development of a large stored program control machine, the No.4 ESS, initiated the era of digital toll and tandem switching. More recently, digital switching has started to find widespread application in the local area. This evolution has been spurred by the new developments in LSI and VLSI integrated circuits.

The next step towards a digital network will be the extension of the digital communications capability to the subscriber. This will provide an opportunity for a whole world of new services besides voice transmission. These new services will include access to massive amounts of information available in large data bases and widespread use of large computers, transmission of text and still video pictures or compressed graphical information, telemetry and remote control through the telephone network, etc. Furthermore, a far more sophisticated signaling system will supersede dial pulse and tone signaling methods. The ultimate goal of this digital trend is the so called Integrated Services Digital Network (ISDN).

Regarding the local telephone line, the first step towards the ISDN will require the provision of full duplex digital transmission over the conventional twisted pair. Use of four-wire facilities for the subscriber is not economic, since it would require massive deployment of new lines, possible only at the expense of

huge investments. Future advances in the ISDN will likely include the use of high capacity fiber optics links in the local area, but in the immediate future, use of the standard two-wire telephone line must be considered.

The minimum acceptable data rate for the new digital subscriber services is 64 Kb/s, in order to accommodate at least one pulse-code modulated voice channel. In addition, signaling information and possibly a data channel are required. A data rate of 80 Kb/s has been considered as probable candidate for standardization. Increasing the data rate to 144 Kb/s would allow simultaneous transmission of two voice channels, besides the above mentioned signaling and data information. Furthermore, some coding techniques generate a very low power spectral density at low frequencies, and so, they allow for an additional voice channel transmitted simultaneously with the data.

Two transmission systems have received attention to achieve the desired full-duplex two-wire transmission at the above mentioned data rates:

1) *Burst mode*, also called time compression multiplexing or ping-pong. In this system groups of bits (*bursts*) are alternatively sent in each direction over the twisted pair. If no delay were present in the line, this multiplexing would require doubling the instantaneous data rate. However, transmission delays in telephone lines are not negligible as compared to the baud period, and so a guard time must be provided between bursts to account for these transmission delays (Fig. 1.1), and less than 50% of the time is available for transmission in each direction. The net result is that the instantaneous data rate is higher than twice the average data rate. A typical figure is $f_{inst} = 2.25 f_b$.

2) *Hybrid-mode*. In this system data flows simultaneously in both directions over the pair of wires. A *hybrid transformer* is used to separate the transmitted from the received signal and perform the two-to-four wire conversion, as in voice transmission. Fig. 1.2 clarifies how this two-to-four wire conversion is achieved.

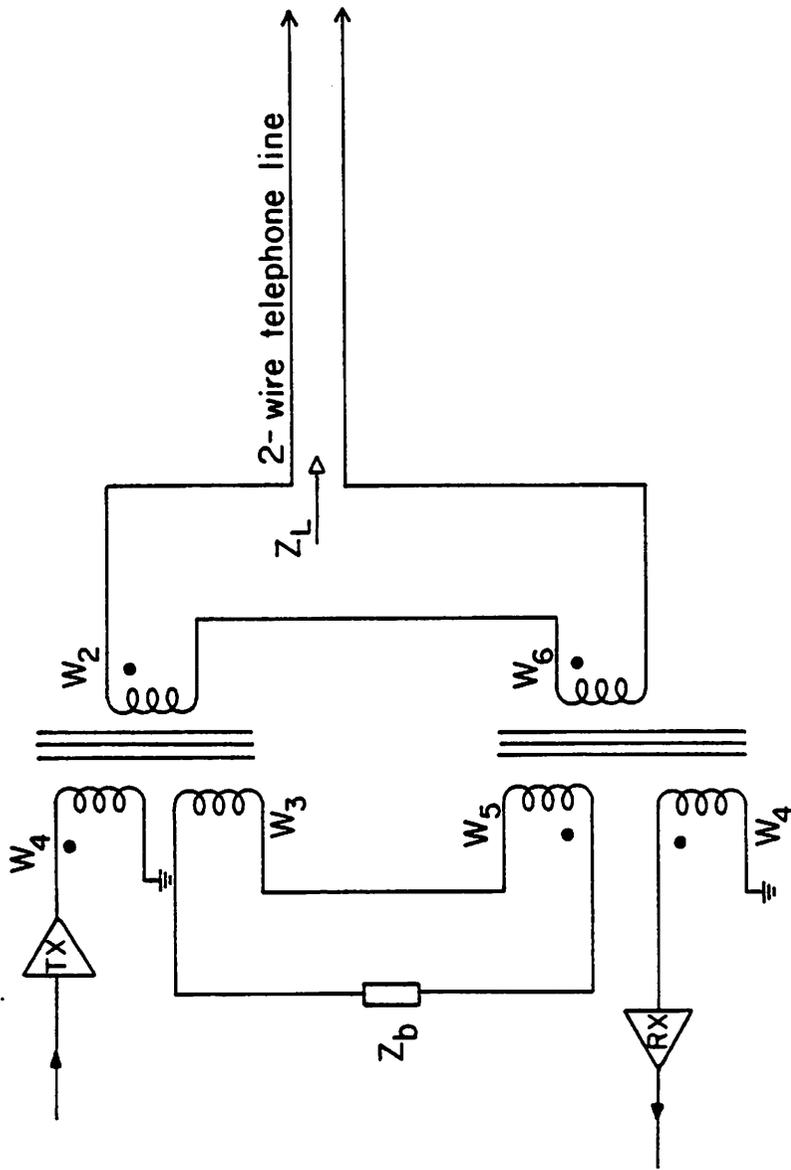


Fig. 1.2. Simplified diagram illustrating how the two-to-four wire conversion is achieved using a hybrid transformer.

Assume that the *balance impedance* Z_b is equal to the line impedance Z_L at all frequencies. Also assume that all the windings have equal number of turns. Then the voltages induced in windings W_2 and W_3 are equal, and since $Z_b = Z_L$, equal currents flow in both secondary circuits. Since windings W_5 and W_6 are connected in opposite directions, the voltage induced in W_4 is zero. Thus, ideally, the receiver is completely isolated from the transmitter. On the other hand, a signal coming from the line induces opposite voltages in W_3 and W_6 , and so this secondary circuit acts as an open circuit for signals coming from the line. Equal voltages are induced in W_1 and W_4 , so the received signal is also coupled to the transmitter, but has no effect on it since the transmitter has a low impedance output driver. Thus, in principle, a perfect isolation can be achieved between the transmitter and the receiver.

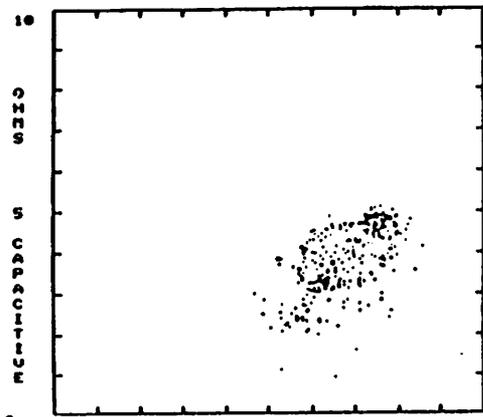
In practice, the balance impedance cannot match the line impedance at all frequencies, since the latter is distributed and the former is lumped. Furthermore, in a mass production environment, it is not desired to trim Z_b to closely match the line impedance Z_L . Z_L values can have large spreads because of the varied nature of the loops that can be found in practice. To give an idea, Fig. 1.3 shows a scatter plot of line input impedances at 32 KHz.

Because of the inevitable mismatch between Z_b and Z_L , the *transhybrid loss*, defined as:

$$THL = -20 \log \left(\frac{V_{receiver}}{V_{transmitter}} \right) \quad (1.1)$$

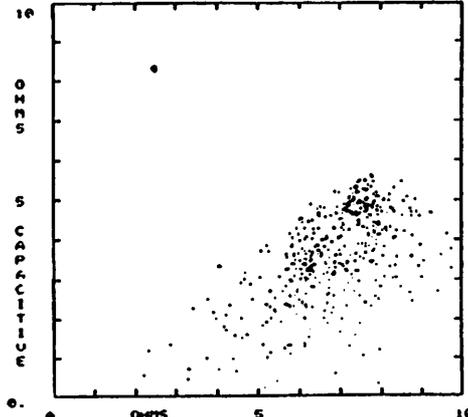
is not infinite in practice, but only 10 dB or less, for typical situations. In voice transmission this poor isolation between the transmitter and the receiver causes the effect called *sidetone* which means that the speaker can hear part of his own voice, as well as the voice coming from the far end. This effect is not necessarily undesirable. In fact, without sidetone speakers tend to speak too

FREQ. = 32.0 KHZ, MIN. REACT. AT 0.0, 0.0MS, MIN. REACT. = 0.0MS
 X SCALE = 20.0 OHMS/DIV., Y SCALE = 20.0 OHMS/DIV., AT 60. DEG F



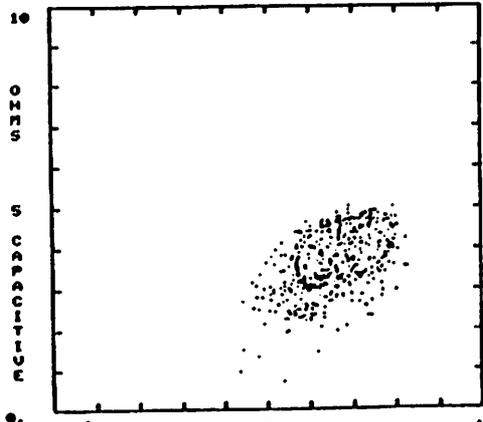
(a)
 LOOP REACTANCE (OHMS) VS RESISTANCE, C.O. SIDE, SIGN OMITTED

FREQ. = 32.0 KHZ, MIN. REACT. AT 0.0, 0.0MS, MIN. REACT. = 0.0MS
 X SCALE = 20.0 OHMS/DIV., Y SCALE = 20.0 OHMS/DIV., AT 60. DEG F



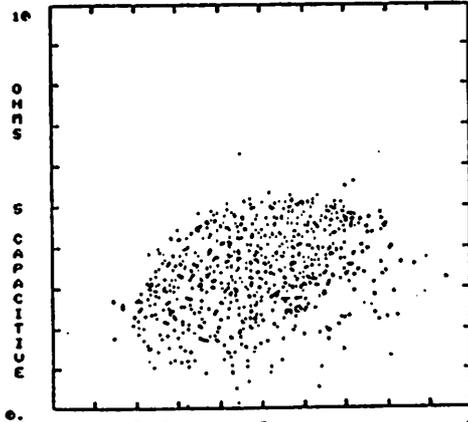
(b)
 LOOP REACTANCE (OHMS) VS RESISTANCE, C.O. SIDE, SIGN OMITTED

FREQ. = 32.0 KHZ, MIN. REACT. AT 0.0, 0.0MS, MIN. REACT. = 0.0MS
 X SCALE = 20.0 OHMS/DIV., Y SCALE = 20.0 OHMS/DIV., AT 60. DEG F



(c)
 LOOP REACTANCE (OHMS) VS RESISTANCE, SUBS. SIDE, SIGN OMITTED

FREQ. = 32.0 KHZ, MIN. REACT. AT 0.0, 0.0MS, MIN. REACT. = 0.0MS
 X SCALE = 20.0 OHMS/DIV., Y SCALE = 20.0 OHMS/DIV., AT 60. DEG F



(d)
 LOOP REACTANCE (OHMS) VS RESISTANCE, SUBS. SIDE, SIGN OMITTED

Fig. 1.3. Scatter plot of line impedances at 32 Khz. (a) Without bridged taps. (b) With bridged taps. (From reference [19])

loudly. The imbalance which causes low transhybrid loss also causes *echoes*, which are the result of the reflections of the transmitted signal in the hybrid at the far end of the line and in any impedance mismatch along the line, caused perhaps by gauge discontinuities, bridged taps, etc. The echo may or may not be annoying depending on its delay. For small delays, even relatively strong echoes can be tolerated, but when the echo comes after a long delay it becomes more and more annoying in direct proportion to its delay. Long delays appear in long distance communications, and particularly in satellite transmissions. Because of the long distance that the signals must travel, echoes may have a round trip delay of about 600 ms in satellite communications. With these delays, even relatively weak echoes are very annoying.

Since long ago, *echo suppressors* have been used to mitigate the effect of the echo. More recently *echo cancellers* have started to be introduced [6.13]. The echo suppressor is a nonlinear device whose operation is based in the momentary interruption of the echo return path. The echo canceller is a linear filter which generates a replica of the echo and subtracts it from the total signal (echo plus far-end component). Voice echo cancellers are similar in many respects to data echo cancellers, which are the subject of this research, however there are also important differences. Here we will not consider voice echo cancellers. For more details about them, the reader is referred to the literature.

A 10 *dB* transhybrid loss may be adequate for voice transmission, but not for data transmission, as will be discussed in detail later. About 60 *dB* or more is required for data transmission. In hybrid-mode data transmission, an adaptive echo canceller is used to improve the transhybrid loss to ≥ 60 *dB*, as shown in Fig. 1.4. The *echo* is defined as the signal leaking directly from the transmitter to the receiver through the hybrid, plus reflections from the far-end and in discontinuities along the line.

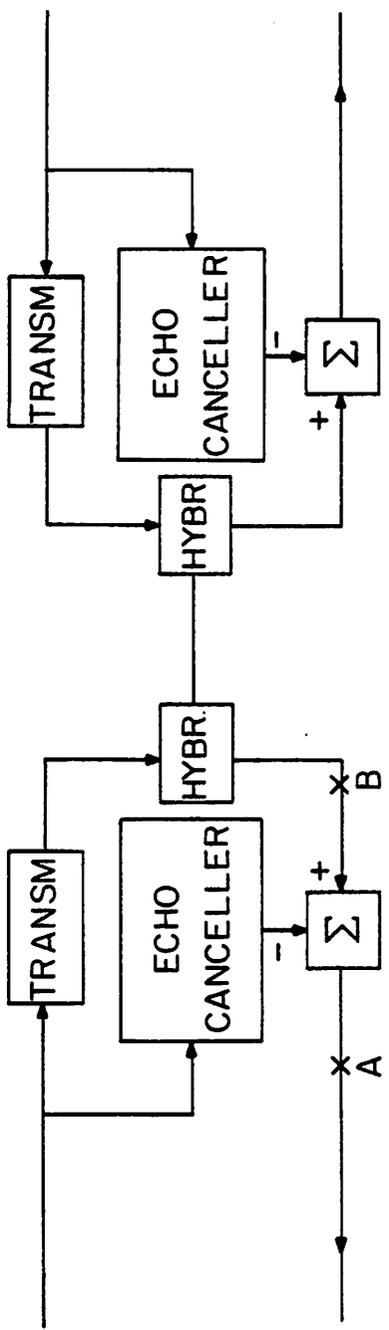


Fig. 1.4. Block diagram of a Hybrid-Method Digital Subscriber Loop, illustrating how an echo canceller is used to improve the isolation between the transmitter and the receiver.

The *echo canceller* is an adaptive transversal filter, the response of which is adjusted by a feedback loop to match the impulse response of the echo path. Since both the echo path and the echo canceller have the same impulse response and are driven by the same data signal, their outputs are also the same, and cancellation occurs by subtraction.

Fig. 1.5a shows a mathematical model of one of the modems of Fig. 1.4. The echo signal can be represented as the response of a linear system which is driven by the sequence of transmitted pulses. A sampled-data notation will be used here and throughout this research. Thus, a linear system can be completely specified giving the sequence of samples of its impulse response $\{g(k)\}$, ($0 \leq k \leq N-1$). It will be assumed that the impulse response of the echo path is of finite length N . Similarly, the transmission path is another linear system specified by its impulse response $\{h(k)\}$, ($0 \leq k \leq M-1$). As a result, the signal coming from the far end of the line is

$$s_n = \sum_{k=0}^{M-1} h(k)y_{n-k} \quad (1.2)$$

and the echo is

$$e_n = \sum_{k=0}^{N-1} g(k)x_{n-k} \quad (1.3)$$

where $\{x_n\}$ is the sequence of near-end bits and $\{y_n\}$ is the sequence of far-end bits. If a noise signal with samples $\{n_n\}$ is also introduced in the channel, the received signal is

$$r_n = s_n + e_n + n_n \quad (1.4)$$

The only desired component here is s_n , whereas both e_n and n_n are unwanted components. In Fig. 1.5b, an *echo canceller* has been introduced to mitigate the effects of the echo. The echo canceller is another linear system, with impulse response $\{c(k)\}$, ($0 \leq k \leq N-1$), and its output, called the *echo replica* is

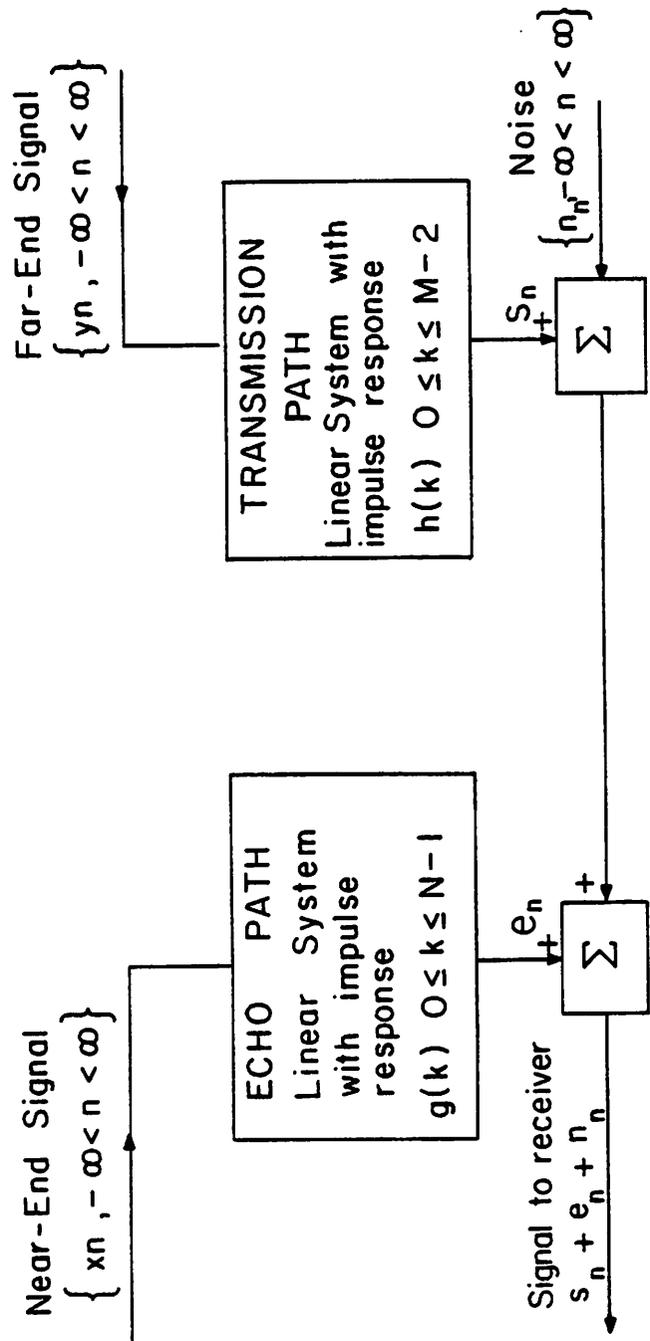


Fig. 1.5a. Mathematical model of the system before the echo canceller is included.

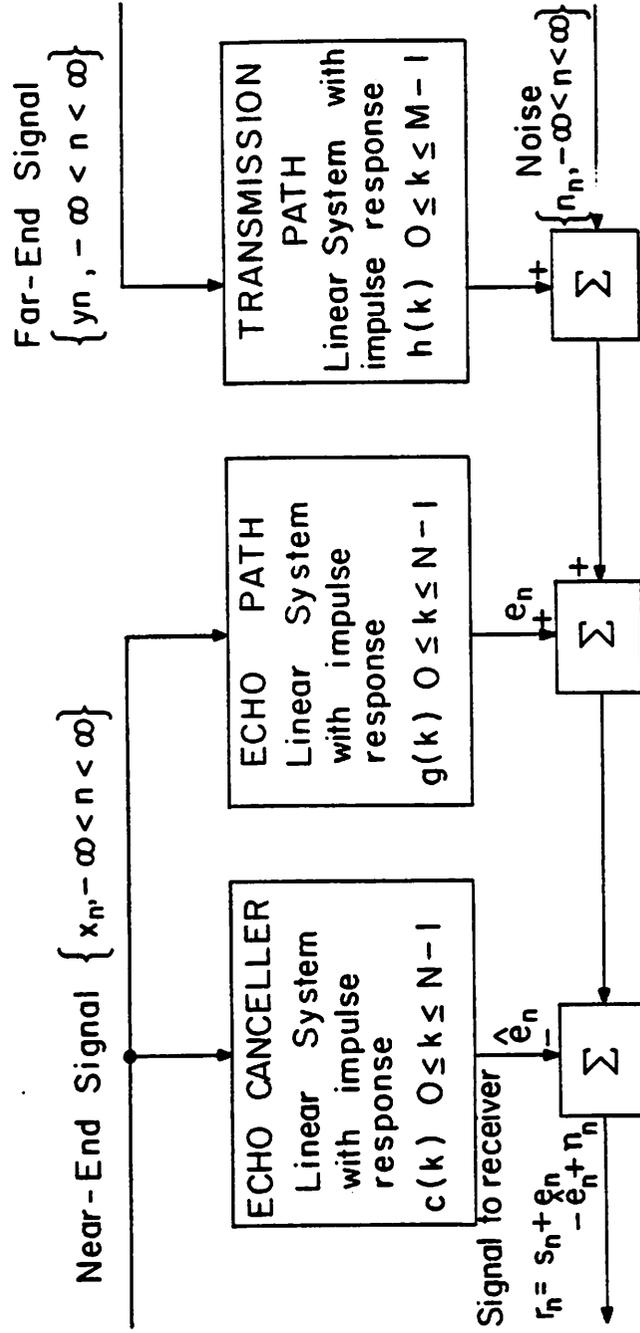


Fig. 1.5b. Mathematical model of the system including the echo canceller.

$$e_n = \sum_{k=0}^{N-1} c(k)x_{n-k} \quad (1.5)$$

The signal passed to the receiver is in this instance

$$r_n = s_n + e_n - e_n + r_n \quad (1.6)$$

If the condition

$$c(k) = g(k) \quad (0 \leq k \leq N-1) \quad (1.7)$$

holds, e_n is equal to e_n and perfect cancellation occurs.

The key issue in this discussion is how to implement a system whose impulse response is guaranteed to be equal to the impulse response of the echo path. Since the echo path is not known in general, and may even be time-dependent, a system with a fixed impulse response cannot be guaranteed to produce the desired results. The *transversal filter* of Fig. 1.6 has an impulse response $\{c(k)\}$, $(0 \leq k \leq N-1)$, where each $c(k)$ is an independently adjustable parameter, and consequently has enough degrees of freedom to generate the desired response. However, in order to guarantee that $c(k) = g(k)$ for all k , a control loop is required to adjust the coefficients $c(k)$ automatically. This control loop is not shown in Fig. 1.6, but will be analyzed in detail in Chapter 2.

Although in the above discussion both the echo path and the echo canceller have been modeled as purely linear systems, small amounts of nonlinear distortion may appear in them in practice. A modification of this basic model to account for nonlinearities will be presented in Chapter 3.

1.1. Comparison Between Burst-mode and Hybrid Systems.

In order to compare the performance of the two systems the following points must be considered:

1) *Line attenuations, range, data rates and crosstalk limitations.* In a burst-mode system line bit rates are higher than the data rate by a factor greater than 2, because of the need to provide a guard time between bursts. As a result,

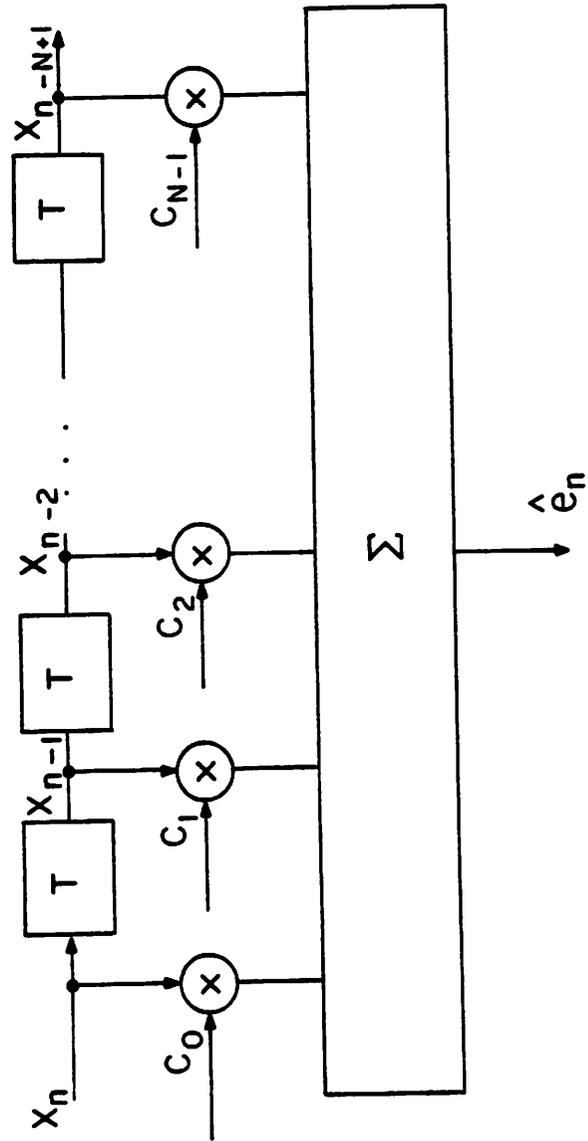


Fig. 1.6. Transversal filter with impulse response $\{c_k, 0 \leq k \leq N-1\}$.

line attenuation is larger than in the hybrid mode and since the maximum transmitted level is limited by compatibility with other services, and the minimum received level is limited by impulse noise and/or crosstalk from other services, maximum range is lower for burst-mode than for hybrid. Reference [19] gives results of very detailed analysis of crosstalk and impulse noise limitations for both systems. Fig. 1.7, reproduced from that reference, shows that the range of a hybrid system is typically about 7 dB larger than for burst mode. This range measured in dB can be converted to actual loop length for a particular gauge, using the attenuation characteristics of the cable. For a more thorough discussion of the two alternative systems the reader is referred to [19]. Other points to consider are:

- a) For a given design, maximum line delay has a fixed limit determined by the guard time between bursts.
- b) Delays are introduced in the signal by buffer registers.

2)Complexity. The implementation complexity of a hybrid mode system considerably exceeds that of a burst-mode. The extra complexity is associated with the echo canceller. A clearer idea of this complexity has come from the research reported in the following chapters. However, the conclusion is that LSI realization is well within reach of present techniques.

A central idea in this research is that the cost of the extra hardware is rapidly decreasing, and so, a larger weight must be given to performance considerations to make a final decision between the two systems. We conclude that an overall cost/performance evaluation favors the hybrid mode, provided that an intelligent use of the possibilities open by VLSI technology is made.

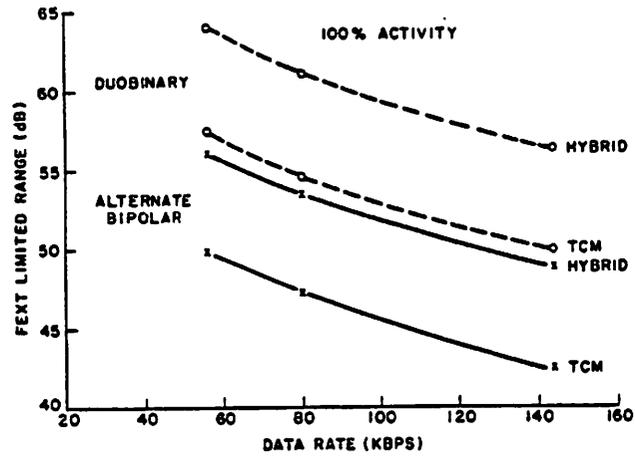


Fig. 19.

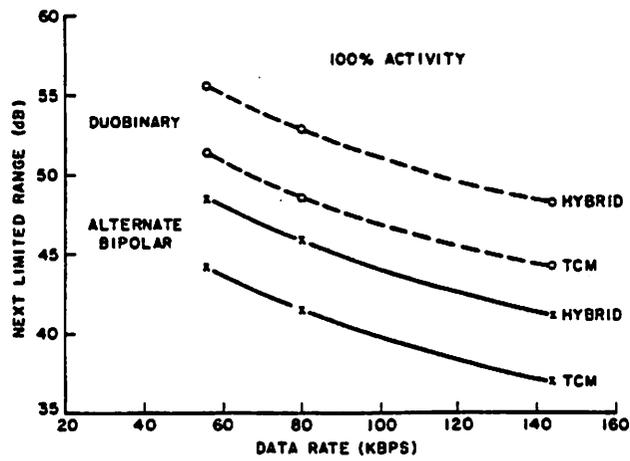


Fig. 1.7. Range of Burst and Hybrid systems. (From reference [19]).

CHAPTER 2

SYSTEM DESIGN

In Section 2.1 we analyze in detail several system parameters, as well as certain techniques that will be used later in the design of the full duplex modem. This design is specifically considered in Section 2.2.

2.1. DESIGN PARAMETERS AND TECHNIQUES

Basic techniques like echo cancellation, equalization and timing recovery will be introduced in this Section, and specifications required in the system design described in Section 2.2 will be given.

2.1.1. Error Rate Objectives

The required amount of echo cancellation can be derived establishing an objective for signal to noise ratio after cancellation, which in turn depends on the desired error rate. To compute the error rate for a given communications system it is required to know the statistical distribution of the noise. The main types of noise that must be considered in the context of a digital subscriber loop are echo, crosstalk and impulse noise. Thermal noise, on the other hand, is not an issue because it is much smaller than the other sources mentioned. When a large number of crosstalk interferers is present, crosstalk noise can be adequately modeled by a Gaussian distribution:

$$p(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{x^2}{2\sigma^2}} \quad (2.1)$$

where σ is the standard deviation of the noise. The echo has a discrete distribution over the set of values

$$e_n = \sum_{k=0}^{N-1} g(k)x_{n-k} \quad (2.2)$$

where the set of x_{n-k} runs over all possible 2^N values. Impulse noise is generally characterized in terms of the number of events per unit time, instead of its statistical distribution, and so it will not be considered in the following analysis. Assuming that the 2^N sequences of x_{n-k} are equally likely, the distribution of the sum of the echo and the crosstalk noise will be:

$$p(x) = \frac{1}{2^N \sqrt{2\pi}\sigma} \sum_{\{x_n\}} e^{-\frac{(x-e_n)^2}{2\sigma^2}} \quad (2.3)$$

where e_n depends on $\{x_n\}$ as shown in eq. (2.2). In many practical cases the distribution of the echo can be approximated by a uniform distribution over an interval $[-a, a]$. In those cases the distribution of the sum of the echo and the Gaussian noise can be modeled as:

$$p(x) = \frac{1}{2a\sigma\sqrt{2\pi}} \int_{(x-a)}^{(x+a)} e^{-\frac{y^2}{2\sigma^2}} dy \quad (2.4)$$

It is convenient to define

$$m = \frac{a}{\sigma} \quad (2.5)$$

The composite distribution of (2.4) can be evaluated numerically for various values of m and $\sigma=1$. Some results are shown in Fig. 2.1. It is clear that for small m the distribution approaches a Gaussian, whereas for large m it approaches a uniform.

Error rates as a function of the input signal to noise ratio for various echo to input noise ratios can be computed numerically for a bipolar coded signal using the distributions of Fig. 2.1. Results are plotted in Fig. 2.2. Here the input signal to noise ratio has been defined as

$$20 \log_{10} \left(\frac{V}{\sigma} \right) \quad (2.6)$$

where V is the rms value of the received signal.

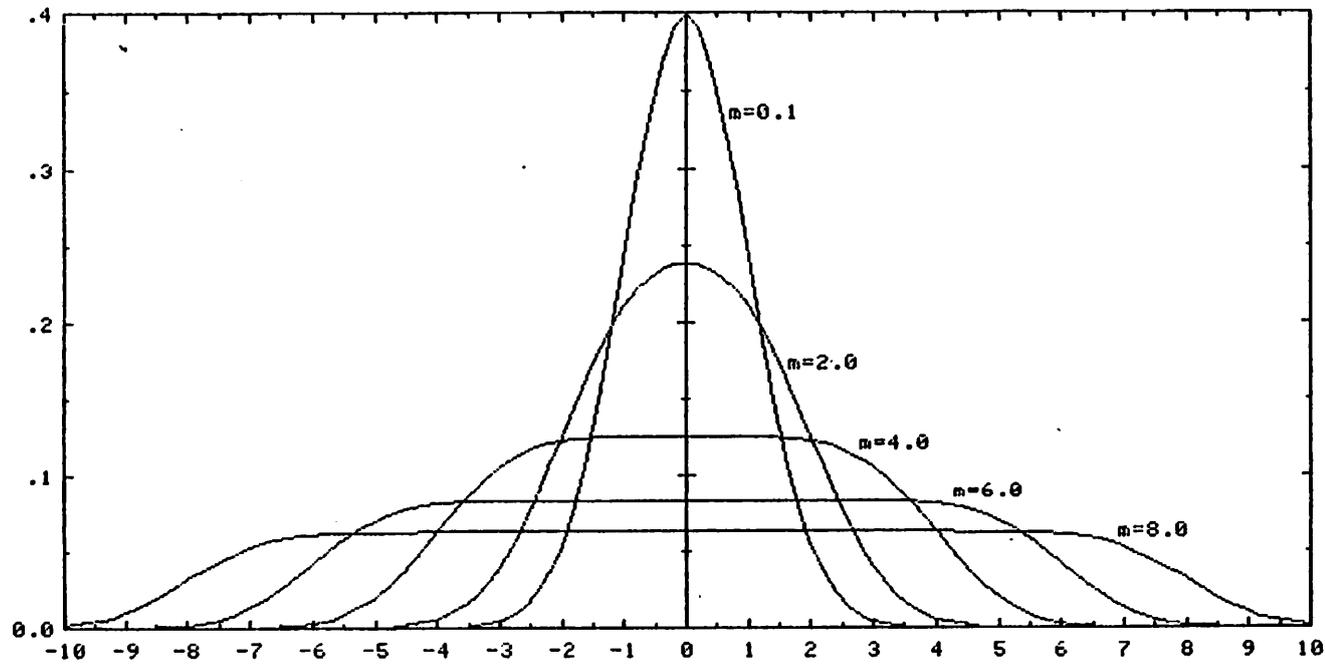


Fig. 2.1. Composite distribution of echo plus crosstalk noise.

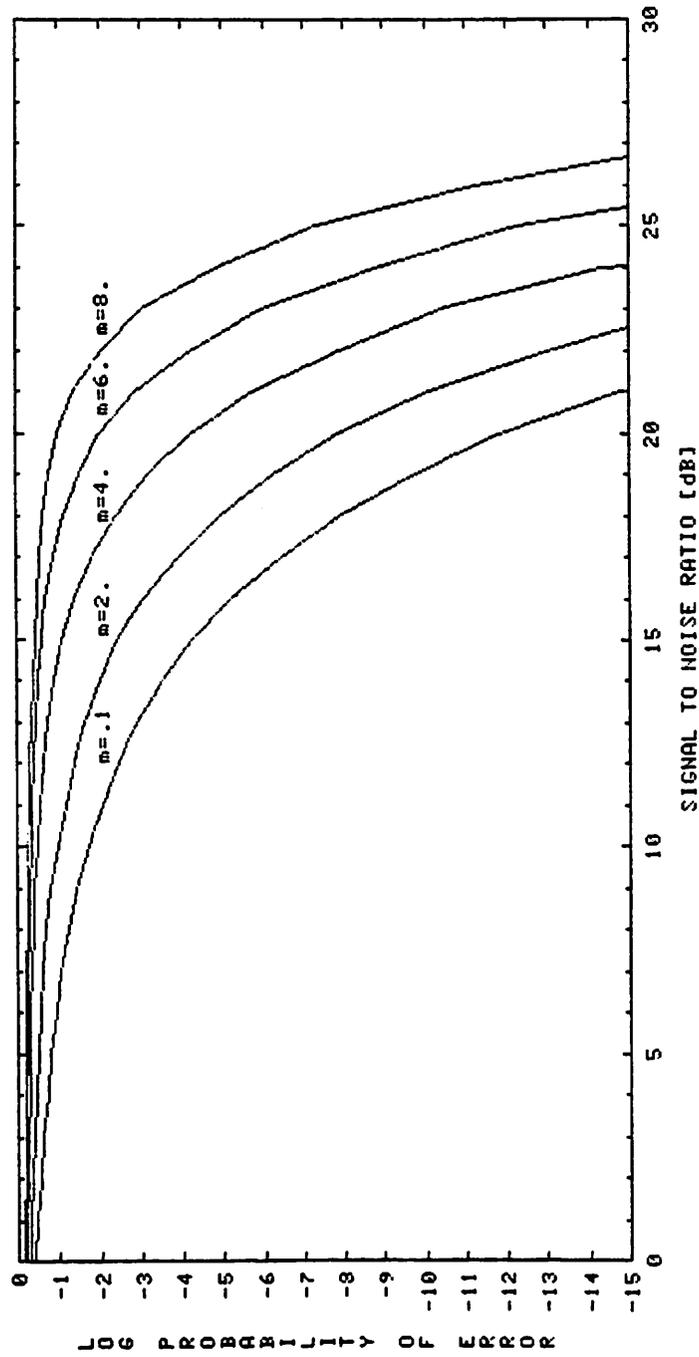


Fig. 2.2. Error rate statistics as a function of the signal to noise ratio for various values of m .

If the input noise is much larger than the echo the distribution is Gaussian and a threshold for low error rate exists in the neighborhood of 20 dB. When the echo is much larger than the input noise, the composite distribution approaches a uniform distribution, and signal to echo ratios smaller than 20 dB may be acceptable.

2.1.2. Echo Cancellation

The basic principle of echo cancellation was introduced in Chapter 1. Here we study techniques to adjust the coefficients of the transversal filter of Fig. 1.6 to force the impulse response of the echo canceller to match the impulse response of the echo path. It is desired to use an algorithm that can be computed in real time and lends itself to a simple hardware implementation.

Although other algorithms have been presented in the literature [6,13], the *least mean squares* (LMS) algorithm has gained widespread acceptance in adaptive transversal filters because of its simplicity and efficiency.

The LMS algorithm minimizes the *mean squared error* (MSE) after cancellation using a gradient technique. To facilitate the analysis let us introduce the following notation:

1) The local transmitted symbol vector

$$\mathbf{x}_n = (x_n, x_{n-1}, \dots, x_{n-N+1})^T \quad (2.7)$$

2) The far-end transmitted symbol vector

$$\mathbf{y}_n = (y_n, y_{n-1}, \dots, y_{n-M+1})^T \quad (2.8)$$

3) The echo-path impulse response vector

$$\mathbf{g} = (g(0), g(1), \dots, g(N-1))^T \quad (2.9)$$

4) The transmission path impulse response vector

$$\mathbf{h} = (h(0), h(1), \dots, h(M-1))^T \quad (2.10)$$

5) The echo canceller tap coefficient vector

$$\mathbf{c} = (c(0), c(1), \dots, c(N-1))^T \quad (2.11)$$

Now the echo signal e_n , the echo replica e_n , and the far-end signal s_n are:

$$e_n = \mathbf{x}_n^T \mathbf{g} \quad (2.12)$$

$$e_n = \mathbf{x}_n^T \mathbf{c} \quad (2.13)$$

$$s_n = \mathbf{y}_n^T \mathbf{h} \quad (2.14)$$

Then the signal at the input of the receiver in Fig. 1.5b is

$$r_n = \mathbf{x}_n^T (\mathbf{g} - \mathbf{c}) + u_n \quad (2.15)$$

where $u_n = s_n + n_n$ is an *uncancellable signal*.

Minimizing the mean squared cancellation error as a function of $\{c(k)\}$ is equivalent to minimizing the mean squared value of the residual signal r_n , since u_n is independent of $\{c(k)\}$. The mean squared residual signal is

$$\rho = E[r_n^2] = |\mathbf{g} - \mathbf{c}|^2 + U \quad (2.16)$$

where $U = E[u_n^2]$, and it has been assumed that $\{x_n\}$ are uncorrelated with one another and with $\{y_n\}$, and assume the values +1 and -1 with equal probability. To minimize ρ we use a gradient technique. The gradient of ρ is:

$$\text{grad } \rho = -2(\mathbf{g} - \mathbf{c}) \quad (2.17)$$

Since \mathbf{c} is not known, this expression is not directly useful to compute the gradient. However, by multiplying (2.17) by \mathbf{x}_n , taking expectation values and using the fact that the x_k are uncorrelated we obtain:

$$E[\mathbf{x}_n^T r_n] = \mathbf{g} - \mathbf{c} \quad (2.18)$$

Thus from (2.17) and (2.18):

$$\text{grad } \rho = -2E[\mathbf{x}_n^T r_n] \quad (2.19)$$

The LMS algorithm minimizes the error using a gradient technique. The coefficients are updated using the expression

$$\mathbf{c}^{(n+1)} = \mathbf{c}^{(n)} - \alpha \text{grad } \rho \quad (2.20)$$

Where a superscript (n) has been introduced to recognize that \mathbf{c} is a function of

time. In order to compute *grad* ρ , the expectation value of (2.19) is replaced by its noisy estimate $\mathbf{x}_n^T \mathbf{r}_n$, since the true expectation value is not available. Thus the algorithm becomes

$$\mathbf{c}^{(n+1)} = \mathbf{c}^{(n)} + 2\alpha \mathbf{x}_n^T \mathbf{r}_n \quad (2.21)$$

The dynamics of the convergence process can be studied using equation (2.21). It is shown in ref. [24] that, under some simplifying but very general assumptions, the evolution of the mean squared cancellation error

$$\varepsilon_n = E[(e_n - \bar{e}_n)^2] \quad (2.22)$$

is governed by the following difference equation

$$\varepsilon_n = (1 - 4\alpha + 4\alpha^2 N) \varepsilon_{n-1} + 4\alpha^2 NU \quad (2.23)$$

whose solution is

$$\frac{\varepsilon_n}{U} = (1 - 4\alpha + 4\alpha^2 N) \left(\frac{\varepsilon_0}{U} - \frac{\alpha N}{1 - \alpha N} \right) + \frac{\alpha N}{1 - \alpha N} \quad (2.24)$$

From here the following conclusions can be derived:

1) Convergence occurs only if:

$$|1 - 4\alpha + 4\alpha^2 N| < 1 \quad \text{or} \quad 0 < \alpha < \frac{1}{N} \quad (2.25)$$

2) After convergence the ratio of residual echo to uncancellable signal is:

$$10 \log_{10} \frac{\varepsilon_\infty}{U} = 10 \log_{10} \left(\frac{\alpha N}{1 - \alpha N} \right) \quad [dB] \quad (2.26)$$

3) The number of iterations required to reduce the residual error by 20 dB is

$$\nu_{20} = \frac{-2}{\log_{10}(1 - 4\alpha + 4\alpha^2 N)} \quad (2.27)$$

For $\alpha \ll \frac{1}{N}$ this becomes

$$\nu_{20} \approx \frac{1.15}{\alpha} \quad (2.28)$$

From Equation (2.26) we see that the parameter α determines the amount of echo cancellation that can be achieved. In order to achieve a large echo can-

cancellation, a small value of α must be used. This is so because in the adaptation algorithm of Eq. (2.21) r_n is used as an error signal. However r_n includes as a component the far-end signal and the noise, in addition to the cancellation error. There is no way to separate those additional components. Even if the cancellation error were exactly zero at some instant, the presence of the uncancellable signal in the adaptation algorithm would move the coefficients away from the optimal cancellation point, degrading the echo cancellation. Thus, because of the presence of this uncancellable signal, perfect echo cancellation cannot be achieved. The effect of the uncancellable signal on the coefficients can be decreased and the echo cancellation improved by decreasing α , as predicted by (2.26). In component k of the correction term of equation (2.21)

$$2\alpha r_n x_{n-k} = 2\alpha(e_n - e_n)x_{n-k} + 2\alpha u_n x_{n-k} \quad (2.29)$$

the second term averages to zero when the contributions of many correction cycles are added, because the uncancellable signal is uncorrelated to the transmitted sequence of bits. Since the cancellation error $(e_n - e_n)$ is correlated to that sequence, its average contribution does not vanish unless $e_n = e_n$. Although the average contribution of the term associated to the uncancellable signal is zero, its instantaneous contribution is not, and so it degrades the echo cancellation as explained before.

Using a very small value of α increases the convergence time and the dynamic range requirements of the adaptation circuitry as will be analyzed later.

The need for a small value of α is a consequence of the fact that data echo cancellers, unlike voice echo cancellers, must always adapt in "*double talk*" conditions, that is, in the presence of a far-end signal. In voice echo cancellers adaptation is generally performed only when the far-end signal is not present. The coefficients are "frozen" during double talk periods.

A different technique to eliminate the effect of the far-end signal is to use a decision-directed algorithm to subtract s_n from r_n . The term $\mathbf{y}_n^T \mathbf{h}$ is locally synthesized using the detector decisions as components of the vector \mathbf{y}_n and an adaptive transversal filter to generate \mathbf{h} . This is analyzed in more detail in Section 2.1.6.

2.1.2.1. Interpolating Echo Cancellers

One of the functions that has to be implemented in a digital subscriber loop is timing recovery. This function entails obtaining in the local receiver a clock that is locked in phase to the far-end transmitter. Since it is not desired to use a separate signal like, for example, a special timing tone to transmit this clock, it must be derived from the received signal. How this is done will be discussed in Section 2.1.7. One important condition to accomplish is to preserve the spectrum of the received signal without aliasing distortion. Aliasing would result from sampling the signal exactly at the data rate, since typically the spectrum of the data signal extends beyond $\frac{1}{2}f_b$, often reaching values close to f_b . Thus the minimum sampling rate is $2f_b$. Some coding techniques with spectra which extend even beyond f_b , require even higher sampling rates.

An oversampling factor of R can be implemented building a transversal filter with RN taps. However because of the fact that the input is a data signal at a rate f_b , a somewhat simplified structure shown in Fig. 2.3a results. This structure consists of R N -tap subfilters, each sampling at f_b and working in time-interleaved fashion, with a phase difference $\frac{T}{R}$. At the output the samples of the R filters are combined, and the output sample rate is Rf_b . Fig. 2.3b shows the samples of the echo path impulse response associated to each subfilter for the case $R=4$. For convenience we will sometimes in the future refer to each of the subfilters as a "phase" of the echo canceller.

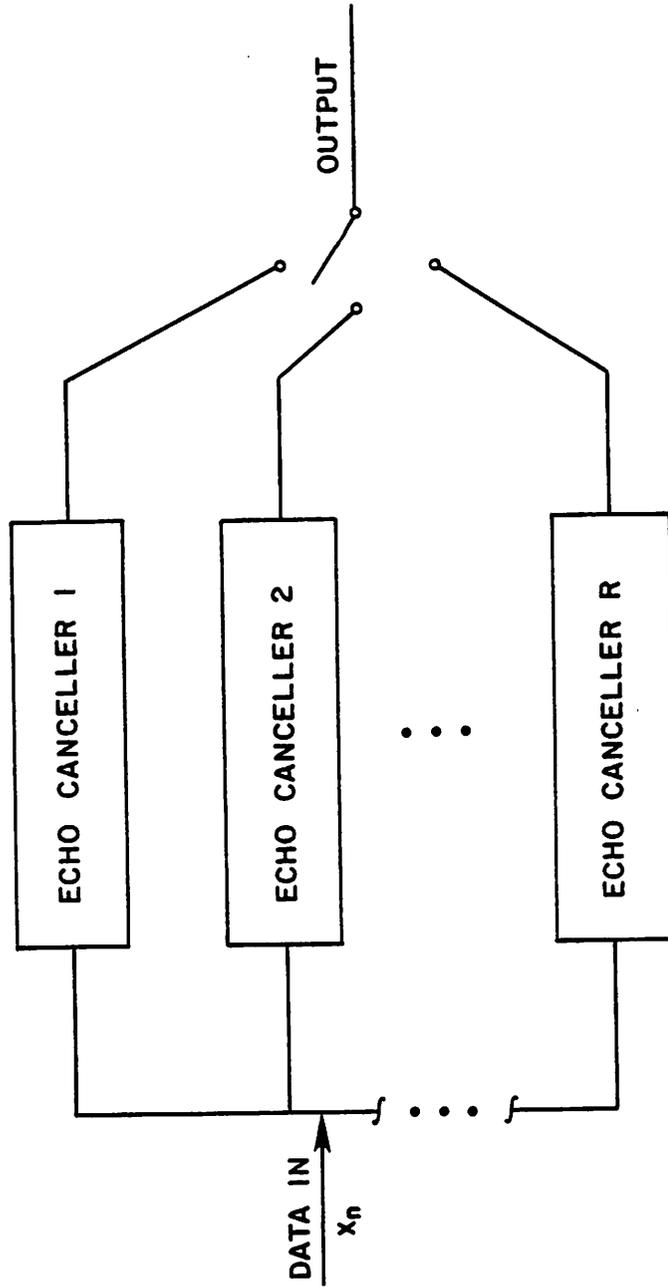


Fig. 2.3a. Structure of interpolating echo canceller.

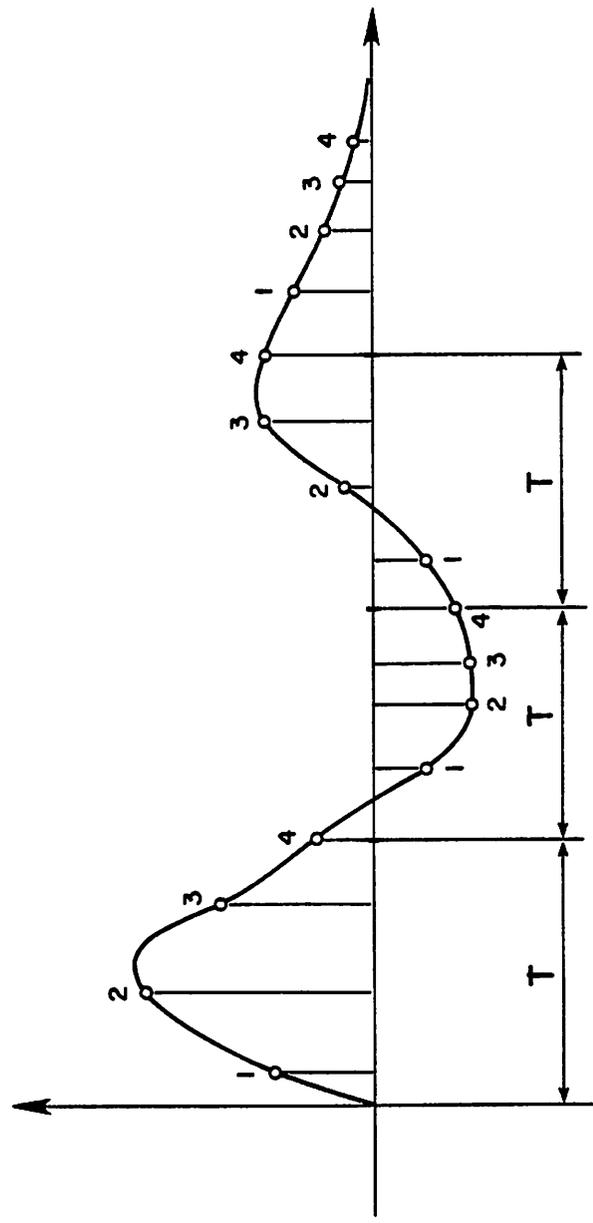


Fig. 2.3b. Diagram showing the multiphase sampling of the echo path impulse response.

2.1.3. Required Amount of Echo Cancellation

Echo cancellation E_c is defined as the ratio of the input to the output *rms* echo noise. When measured in *dB*, its expression is:

$$E_c = 20 \log_{10} \left(\frac{\text{Input Echo}}{\text{Output Echo}} \right) \quad (2.30)$$

If a transhybrid loss of 10 *dB* is assumed, the signal to echo noise ratio at the input of the echo canceller (point A in Fig. 1.4) is *negative* and equal to -30 *dB* for the case of 40 *dB* of line attenuation. To improve it to +20 *dB* (at point B in Fig. 1.4), an additional suppression of the undesired echo signal (or echo cancellation) of 50 *dB* is desired.

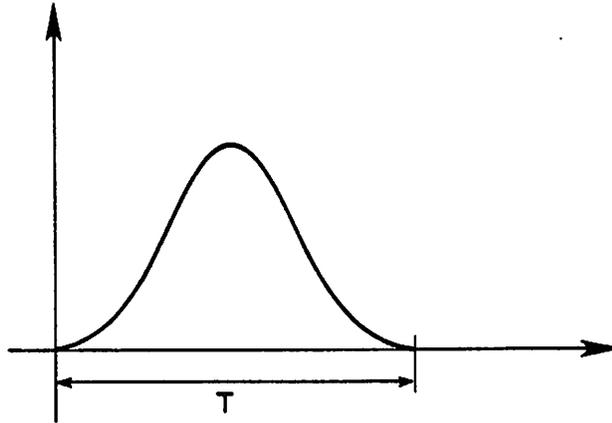
2.1.4. Channel Shaping

Because of the linear distortion introduced in the data signal by the frequency dependent attenuation and delay characteristics of the line, individual pulses are dispersed and interfere with each other. This effect is known as *inter-symbol interference (ISI)*. Fig. 2.4 shows a possible pulse shape before and after transmission through the line. Since at the receiver the signal is sampled at instants kT , where k is an integer and T is the data period, the samples of the impulse response should be zero for all values of $k \neq 0$. Mean square *inter-symbol interference* is defined as

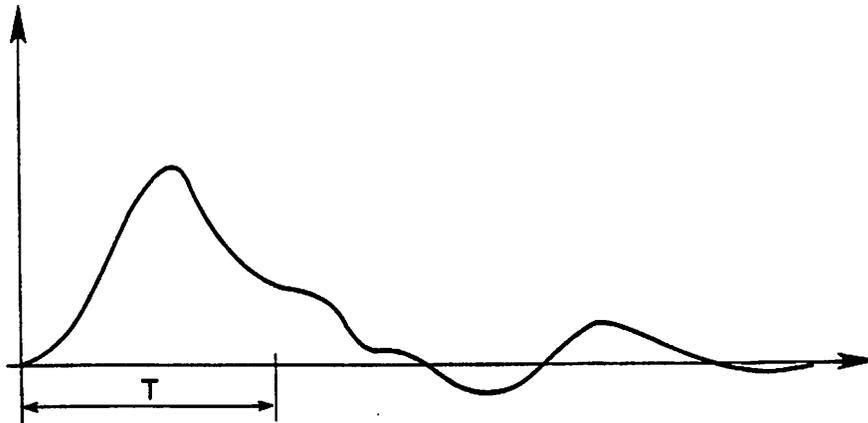
$$ISI = \frac{\sum_{k \neq 0} |h(kT)|^2}{|h(0)|^2} \quad (2.31)$$

where $h(t)$ is the channel impulse response. For the case of Fig. 2.4b, the value of ISI gives an indication as to what extent the detection of a given pulse is impaired by the interference from its neighbors.

A useful graphical representation of a data signal that allows immediate evaluation of the amount of ISI is the *eye diagram*. The eye diagram is the superposition of many pulses as would result for example, from observing the



(a) TRANSMITTED PULSE



(b) RECEIVED PULSE

Fig. 2.4. Effect of amplitude and phase distortion in the line.
(a) Transmitted pulse. (b) Received pulse.

data signal with an oscilloscope synchronously triggered at the data rate. In Fig. 2.5a we show the eye diagram of a two-level data signal for the case of zero intersymbol interference. In Fig. 2.5b a similar signal is shown, in this case with significant intersymbol interference introduced by a dispersive channel. Fig. 2.5c and 2.5d show eye diagrams for a three level data signal, as occurs in the case of bipolar coding (see Section 2.2.2). The *eye opening*, usually specified in %, is the ratio $\frac{b}{a}$ in Fig. 2.5b, and is another indication of ISI (although it is related to *peak* ISI instead of mean squared ISI as defined by (2.31)).

It is interesting to investigate the conditions under which zero intersymbol interference is achieved. Of course one such situation arises when the channel is so wideband that pulses can be kept confined to their own time slots. The impulse response must then be zero for $t \geq T$. This condition is not practical since most real channels are bandlimited. The Nyquist theorem on vestigial symmetry establishes that a sufficient condition for zero interference is that the channel frequency response be symmetrical around half the data rate. Fig. 2.6 illustrates this condition. Mathematically it can be stated as follows:

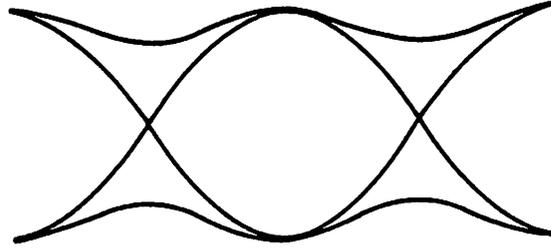
$$\sum_{-\infty}^{\infty} H(\omega + \frac{2\pi k}{T}) = \text{constant} \quad (2.32)$$

where $H(\omega)$ is the channel frequency response and T is the signalling period. For the case of a channel that is bandlimited to less than the data rate this condition implies that the rolloff of $H(\omega)$ is symmetrical around $\frac{1}{2}f_b$, as illustrated in Fig. 2.6.

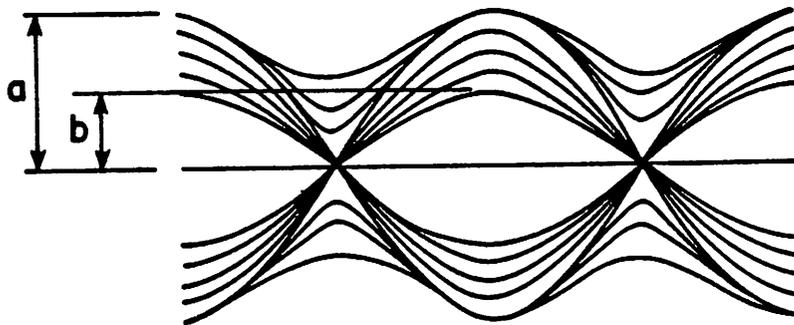
The Nyquist condition can be derived as follows. The condition for zero intersymbol interference is

$$s(t) \sum_{k=-\infty}^{\infty} \delta(t - kT) = s(0) \delta(t) \quad (2.33)$$

Taking Fourier transforms of both sides of Eq. (2.33), and recalling that the Fourier transform of $\delta(t)$ is a constant, and the Fourier transform of the left

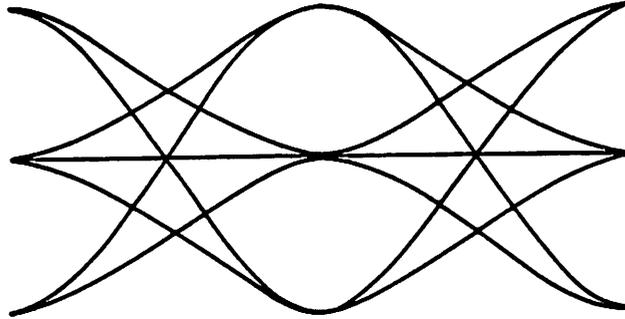


(a)

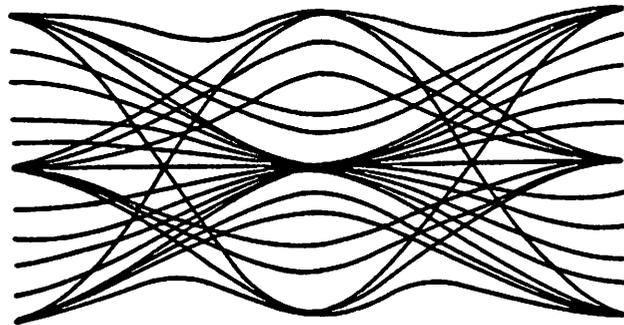


(b)

Fig. 2.5. (a) Two-level open eye. (b) Two-level closed eye.

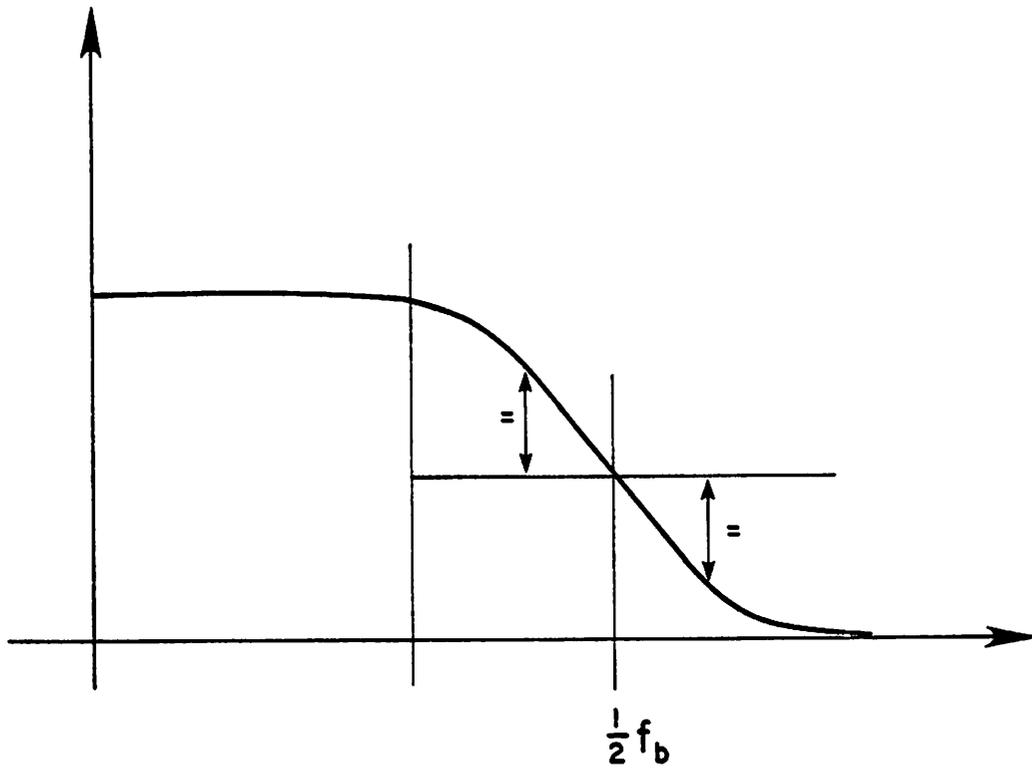


(c)



(d)

Fig. 2.5. (c) Three-level open eye. (d) Three-level closed eye.



(a)

Fig. 2.6a. Filter with symmetric rolloff satisfies Nyquist criterion.

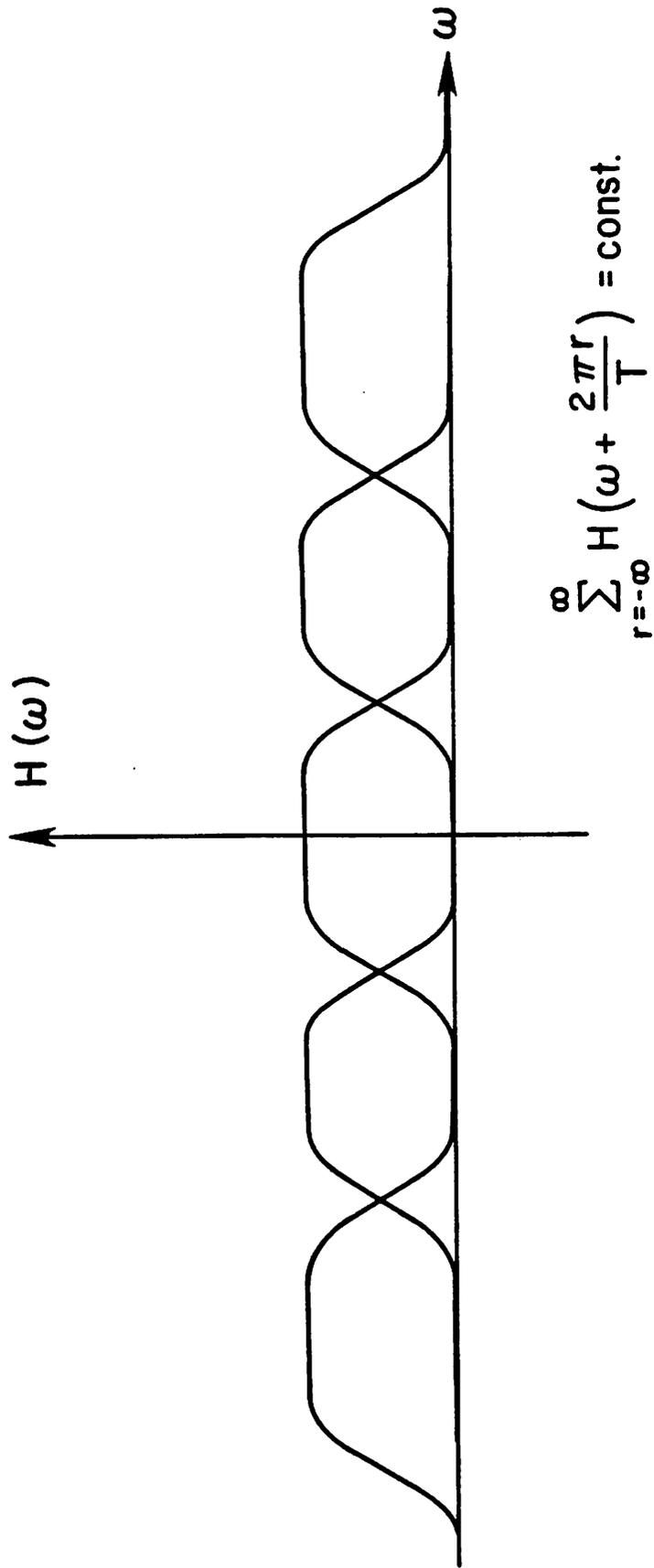


Fig. 2.6b. The sum of the aliases of the transfer function is a constant.

hand side is:

$$\sum_{-\infty}^{\infty} H(\omega + \frac{2\pi k}{T}) \quad (2.34)$$

the result of (2.32) follows. We have neglected a constant additive delay τ in the sampling of Equation (2.33). This would introduce a linear phase characteristic in $H(\omega)$.

If the transmission line has a flat amplitude response, condition (2.32) can be met by designing pulse shaping filters with a symmetrical amplitude rolloff, as in Fig. 2.6, and a linear phase. One such filter commonly used in practice is the *raised-cosine* filter whose amplitude response is

$$H(\omega) = \begin{cases} 1 & |\omega| \leq \frac{1}{2}(1-\alpha)\omega_b \\ \frac{1}{2} - \frac{1}{2} \cos\left[\frac{\pi(\omega-\omega_b)}{\alpha\omega_b}\right] & \frac{1}{2}(1-\alpha)\omega_b \leq |\omega| \leq \frac{1}{2}(1+\alpha)\omega_b \\ 0 & |\omega| \geq \frac{1}{2}(1+\alpha)\omega_b \end{cases} \quad (2.35)$$

This characteristic can be approximated to an arbitrary accuracy by finite impulse response filters. Usually this filtering function is split in equal parts between the transmitter and the receiver. This results in *matched filters* which optimize the signal to noise ratio. Often, other characteristics different from those described by (2.35) are used for ease of implementation.

One interesting approach to filter design consists in finding a response that minimizes (2.31) directly, instead of attempting to approximate as closely as possible a given, relatively arbitrary characteristic like (2.35). Such approach has been pursued by Lind and Nader [49,52] and when applied to an all pole transfer function results in a set of nonlinear equations for the pole locations of the filter. These equations are repeated in Appendix A for easy reference, and a Fortran program is given to solve them.

2.1.5. Equalization

Generally the input and output filters of a data transmission system are designed either assuming a flat frequency response in the line, or a particular compromise response, which attempts to approximate as closely as possible the responses that are going to be found in practice. However the actual line response will generally depart from the one used in the design, because of the varied nature of the lines with which the system will be required to operate. This line distortion will generate intersymbol interference.

Equalizers are linear or nonlinear networks designed to correct intersymbol interference arising because of the poorly controlled distortion characteristic of the line. They may be *manually adjustable* or *automatic*, depending on the way their impulse response is tailored to the individual characteristics of the line with which they are required to operate. Examples of manually adjustable equalizers are the LBO (line build out) networks used in pulse-code modulation (PCM) repeaters. Here the characteristics of the line are relatively well controlled and transmission lengths are fixed, so that a finite set of correction networks can be used to equalize different line lengths. Once the network has been installed, no changes are expected in the line characteristics and consequently the equalizer does not need adjustments either.

In order to avoid manual selection of the equalization network, and also to take into account changes in the line characteristics, possibly due to aging or environmental changes, automatic line build out (ALBO) networks have been introduced in PCM repeaters. They interpolate line lengths between a small number of fixed buildout networks, thereby avoiding a large inventory of different LBO's. These ALBO networks are an example of automatic equalizers.

A more advanced example of automatic equalizers are the *adaptive equalizers* used in voiceband data sets. Adaptive equalizers use an adaptive transversal

filter to generate a frequency response which equalizes the channel distortion. To understand how a transversal equalizer works, consider first the infinite equalizer of Fig. 2.7. Since a delay T generates a frequency response $e^{-j\omega T}$ the transversal equalizer of Fig. 2.7 will have a transfer function

$$F(\omega) = \sum_{k=-\infty}^{\infty} c_k e^{-j\omega k T} \quad (2.36)$$

This is the Fourier expansion of function $F(\omega)$ which is periodic in ω with period $\frac{2\pi}{T}$. Thus, provided that the transfer function $H(\omega)$ that we want to equalize is bandlimited to $|\omega| \leq \frac{2\pi}{T}$, it is always possible to find a set of coefficients c_k such that $F(\omega)H(\omega)$ is approximately constant in the interval $|\omega| \leq \frac{2\pi}{T}$.

In practice a finite transversal filter must be used, so that the Fourier expansion of (2.36) is truncated and $H(\omega)$ is only approximately equalized. To make the equalizer adaptive the coefficients can be computed using the LMS algorithm as in the echo canceller of Section 2.1.2. The adaptation error is determined by comparison of the output of the equalizer with a set of fixed references, corresponding to the expected discrete values of the signal samples. Consider for example the equalizer of Fig. 2.8. If a binary data signal is being handled, the output samples y_n are expected to be either +1 or -1; thus the error is

$$\varepsilon_n = \begin{cases} y_n - 1 & (y_n > 0) \\ y_n + 1 & (y_n < 0) \end{cases} \quad (2.37)$$

This error is used to adapt the coefficients using the LMS algorithm in the same way as in the echo canceller of Section 2.1.2. An important difference with the echo canceller is that the signal is not binary and multipliers are required in a hardware implementation of the equalizer.

Adaptive transversal equalizers have found widespread application in voiceband modems. However digital subscriber loops do not justify the

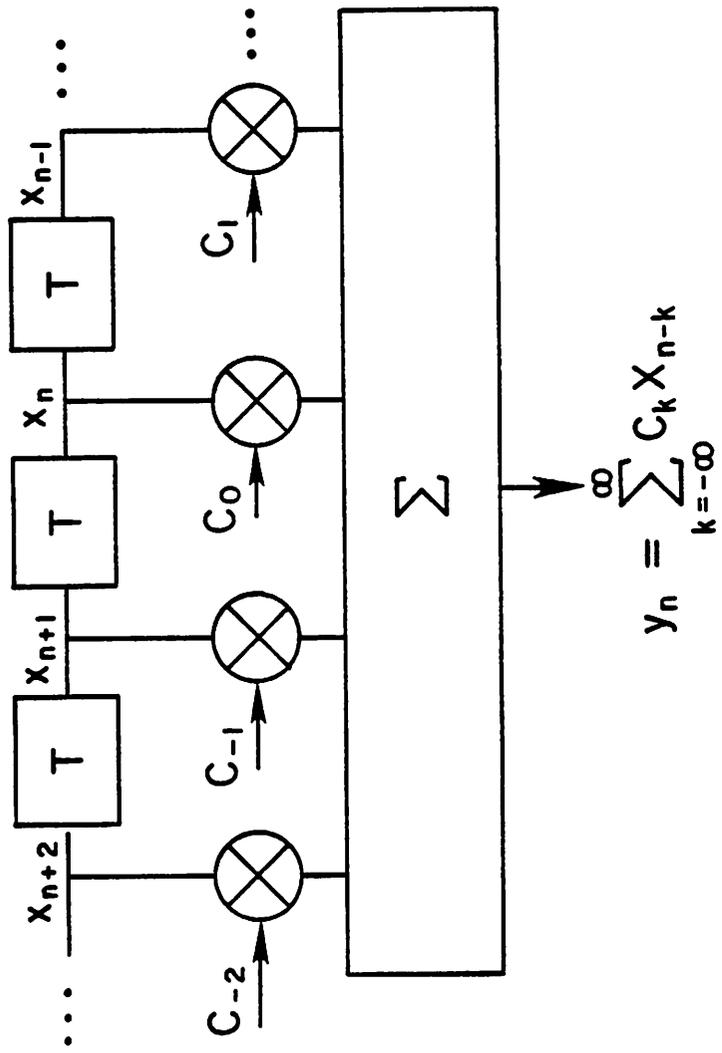


Fig. 2.7. Infinite Equalizer.

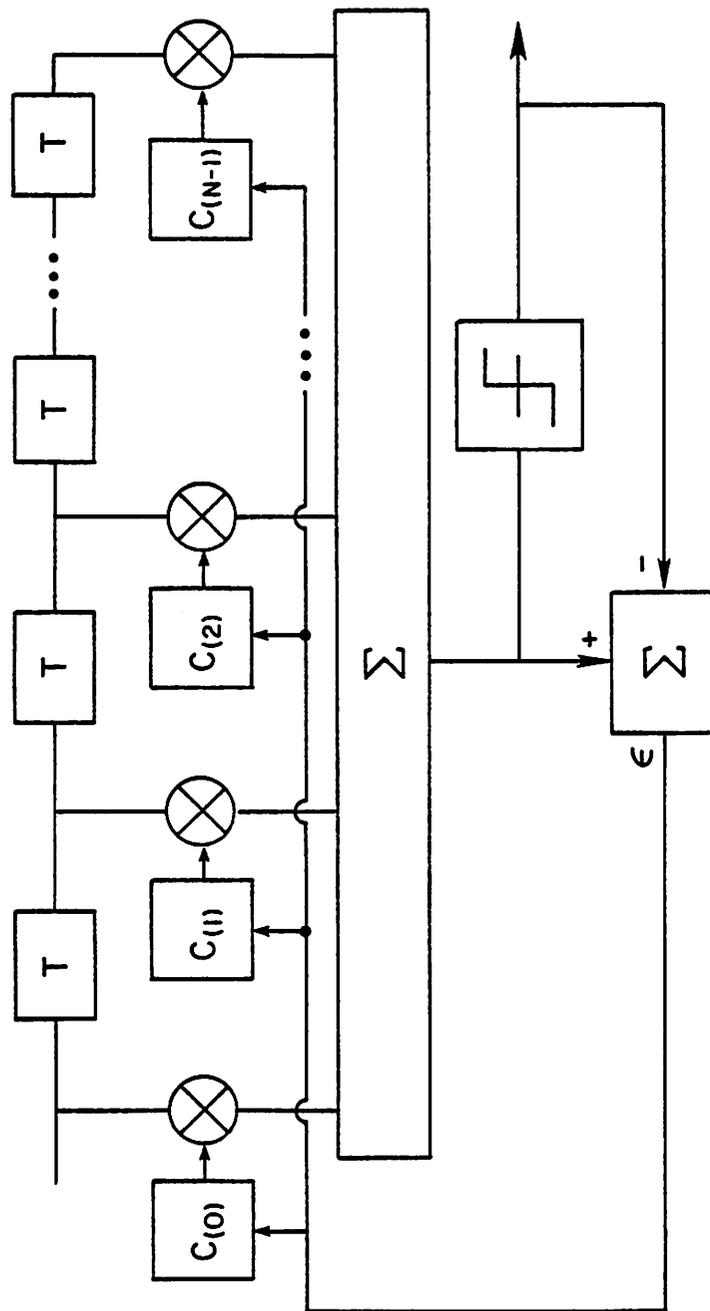


Fig. 2.8. Linear Equalizer.

complexity of a transversal equalizer. This complexity is much higher than that of the echo canceller due to the need for multipliers.

Another kind of adaptive equalizers are *decision feedback equalizers* (DFE), which are an example of nonlinear equalizers. DFE's are useful only to correct *postcursor interference*, which occurs when the impulse response of the channel is as in Fig. 2.9. Here the largest sample of the impulse response is the first, $h(0)$, and intersymbol interference is generated by the postcursors, i.e. $h(k)$, $k > 0$. The received signal at sample time n is:

$$s_n = \sum_{k=0}^{M-1} h(k)y_{n-k} = h(0)y_n + \sum_{k=1}^{M-1} h(k)y_{n-k} \quad (2.38)$$

The second term, which represents the intersymbol interference, is generated by an adaptive transversal filter in the DFE structure of Fig. 2.10, using past decisions \hat{y}_k . For low error rates these decisions are almost always equal to y_k . The intersymbol interference component

$$\sum_{k=1}^{M-1} h(k)y_{n-k} \quad (2.39)$$

is subtracted from the received signal and the remaining component $h(0)y_n$ is used to make the decisions. Since the transversal filter is in the feedback path, instead of being in the forward path as in Fig. 2.10, the name "decision feedback equalizer" is justified.

It is shown in the literature [43,45] that the DFE will converge in spite of some initial decision errors, and that the structure can recover relatively easily of occasional errors made during normal operation, although it does have error propagation properties.

Decision feedback equalizers are attractive in digital subscriber loops because of their hardware simplicity. Similarly to echo cancellers, they do not require multipliers. The fact that they can correct only postcursor interference is not considered a serious limitation here because in general local telephone

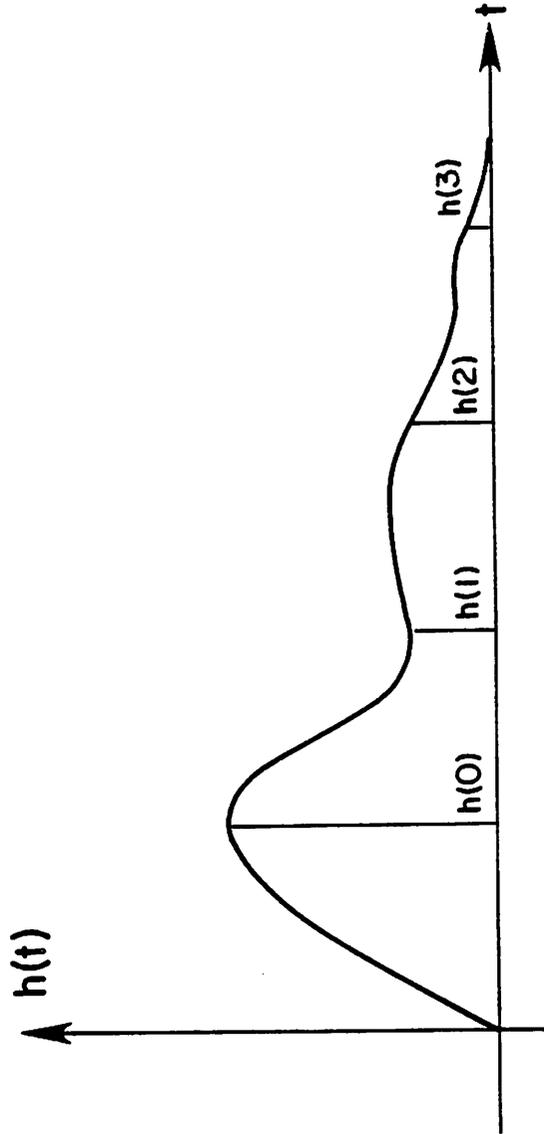


Fig. 2.9. Channel impulse response showing postcursor intersymbol interference.

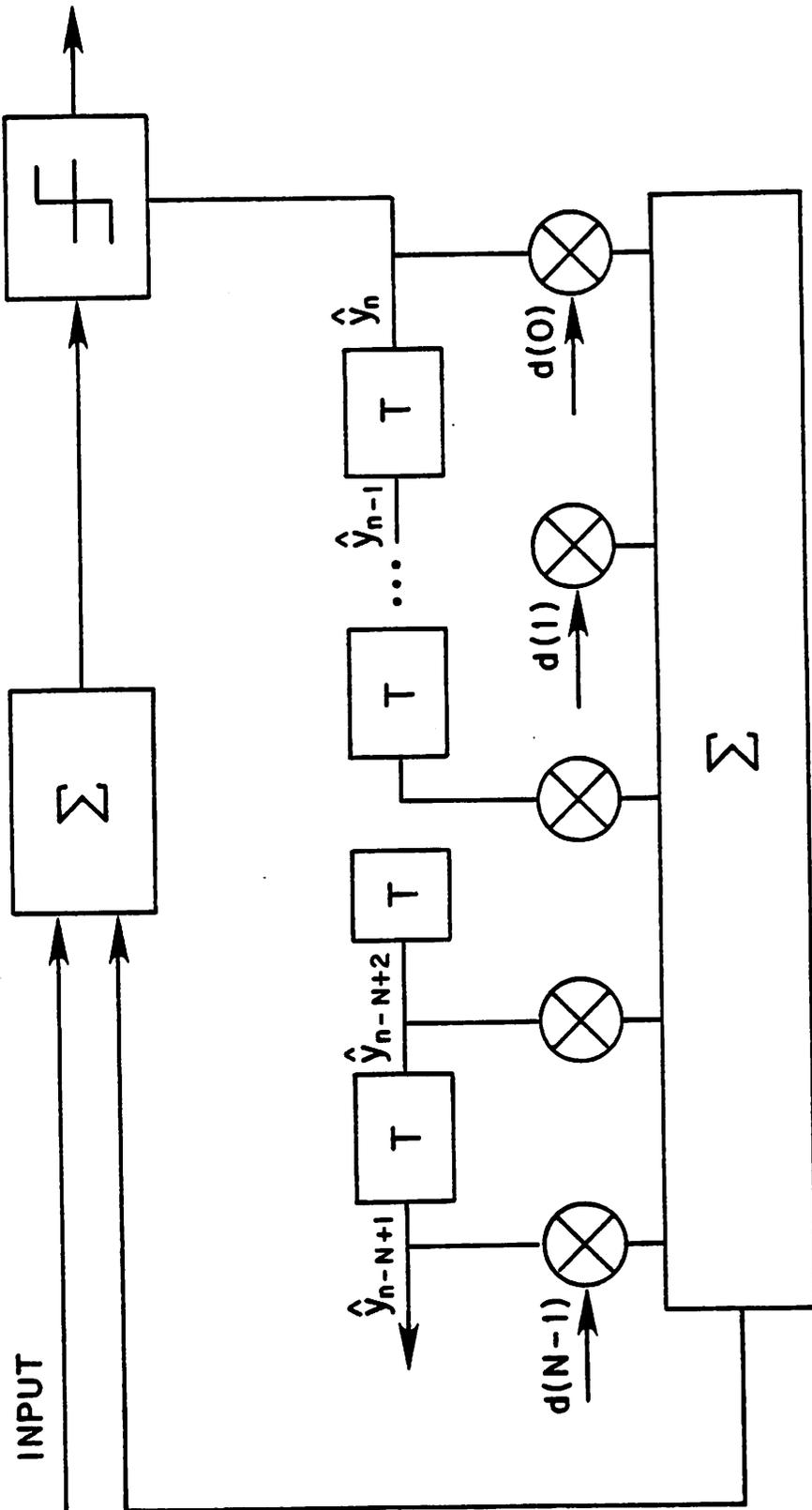


Fig. 2.10. Decision Feedback Equalizer.

lines do not exhibit precursor interference. Another advantage of DFE's as compared to linear transversal equalizers is that they do not generate noise enhancement as do the latter, and their convergence is independent of the characteristics of the channel. Noise enhancement results when the linear equalizer attempts to correct a channel characteristic that has a spectral zero or a very small amplitude at some point within the frequency band of interest. The linear equalizer will generate a very large gain at that frequency, which will greatly amplify the noise. Because of these spectral nulls, linear equalizers may also occasionally show poor convergence. Spectral zeroes may result, among other reasons, because of the aliasing of the frequency components beyond $\frac{1}{T}$ inside the band $0 \leq f \leq \frac{1}{T}$, as shown by Lyon [39]. This problem is avoided with *fractional tap spacing* equalizers [40,41].

2.1.6. Combining Echo Cancellation and Decision Feedback Equalization

We discussed in Section 2.1.3 how the presence of a far-end signal which is undistinguishable from the cancellation error (except for the fact that, unlike the cancellation error, it is uncorrelated to the transmitted data) causes random fluctuations of the echo canceller coefficients and degrades cancellation. To mitigate this problem, a very small adaptation gain is used in the LMS algorithm. Although this improves the cancellation, it also makes convergence slower.

It has been proposed by Mueller to combine in a single structure the echo canceller of Fig. 1.6 and the decision feedback equalizer of Fig. 2.10. The resulting structure is shown in Fig. 2.11. The echo canceller transversal filter generates an echo replica signal given by:

$$e_n = \sum_{k=0}^{N-1} g(k)x_{n-k} \quad (2.40)$$

and the decision feedback equalizer uses past decisions \hat{g}_n to generate a replica

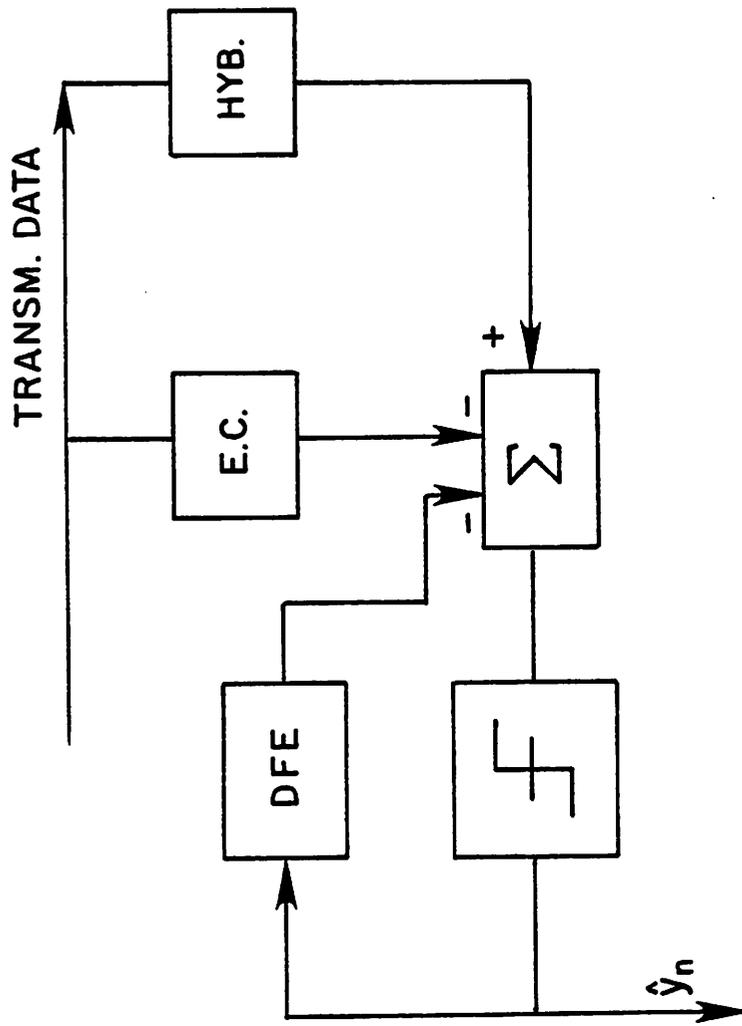


Fig. 2.11. Combined echo canceller and decision feedback equalizer.

of the far-end signal

$$s_n = \sum_{k=0}^{M-1} d(k)g_{n-k} \quad (2.41)$$

The residual signal, after echo cancellation and simultaneous decision feedback equalization becomes:

$$\tau(n) = e_n - e_n + s_n - s_n + n_n \quad (2.42)$$

Except for the noise term n_n , this represents a combination of the echo cancellation and equalization errors. In the absence of noise, this error can be eventually reduced to zero, after convergence of the adaptive filters. It should be noted that (2.41) involves a term with $k=0$ which corresponds to the present received bit g_n . The operation of the system must then be performed in two steps. During the first step the echo replica and the component of the far-end signal associated to previous received bits are subtracted, and so the quantity

$$e_n - e_n + h_0 y_n + \sum_{k=1}^{M-1} (h(k)y_{n-k} - d(k)g_{n-k}) + n_n \quad (2.43)$$

is computed. If $e_n = e_n$, $g_k = y_k$ and $d(k) = h(k)$, this quantity is $h_0 y_n + n_n$, and a decision can be made with minimum probability of error concerning the value of y_n , which yields g_n . During the second step $d(0)g_n$ is also subtracted and the quantity $\tau(n)$ (2.42) is finally computed. $d(0)$ is known as an *AGC (automatic gain control) tap*, because of the role it plays in the detection [23].

If transversal filter tap coefficient vectors \mathbf{c} and \mathbf{d} are defined for the echo canceller and decision feedback equalizer, in analogy with Section 2.1.2, as well as data vectors \mathbf{x}_n , \mathbf{y}_n , \mathbf{g}_n , the adaptation algorithm can be written:

$$\begin{aligned} \mathbf{c}^{(n+1)} &= \mathbf{c}^{(n)} + 2\alpha r(n) \mathbf{x}_n \\ \mathbf{d}^{(n+1)} &= \mathbf{d}^{(n)} + 2\alpha r(n) \mathbf{g}_n \end{aligned} \quad (2.44)$$

It is shown in [22] that this algorithm converges, and that the residual signal to noise ratio after cancellation and equalization is:

$$10 \log_{10} \left(\frac{\varepsilon_{\infty}}{U} \right) = -10 \log_{10} \left(\frac{\alpha(N+M)}{1-\alpha(N+M)} \right) \quad (2.45)$$

Convergence is described by the equation:

$$\varepsilon_k = \varepsilon_{\infty} + [1 - 4\alpha + 4\alpha^2(N+M)]^k (\varepsilon_0 - \varepsilon_{\infty}) \quad (2.46)$$

which is equivalent to (2.24).

Fastest convergence occurs for

$$\alpha = \frac{1}{2(N+M)} \quad (2.47)$$

in which case (2.46) becomes:

$$\varepsilon_k = \varepsilon_{\infty} + \left(1 - \frac{1}{J+L} \right)^k (\varepsilon_0 - \varepsilon_{\infty}) \quad (2.48)$$

In the case when the noise is zero, ε_{∞} can be reduced to exactly zero, according to (2.45). If the noise is not zero, use of the optimal value of α will result in $\varepsilon_{\infty} = U = \sigma^2$ (where σ is the standard deviation of the noise), which implies a noise enhancement of 3 dB. Lower noise enhancements can be achieved at the expense of a decrease in convergence speed.

An advantage of the larger values of α that can be used in this structure is the reduced dynamic range requirements in the coefficient registers and in the adder that computes the adaptations of Eq. (2.44).

Unfortunately this structure requires that the local transmitter be synchronized to the incoming data rate. This requirement is difficult to meet in practice because in most data communications systems timing is derived from the received signal (no separate timing tones or clock lines are allowed). However this timing recovery cannot be performed before echo cancellation because the signal is greatly corrupted by the echo, and it cannot be derived after echo cancellation either, because the signal has been sampled only once per cycle at this point and thus a significant aliasing distortion has been introduced. Thus this technique cannot be used in conjunction with standard timing recovery techniques.

Although other timing recovery techniques compatible with this approach can be imagined they do not seem to be reliable and thus render the above structure impractical for the moment, although it retains its theoretical interest and may become practical if some better timing recovery technique is found.

2.1.7. Timing Recovery

A basic system requirement is to synchronize the *receiver* to the far-end transmitter. This is necessary to sample the received signal at the instant of maximum eye opening and so detect the original data with minimum probability of error.

A distinct requirement is to synchronize as well the *local transmitter* to the received signal. Depending on the design of the echo canceller, this can be a requirement of the echo canceller or not. However it is always a requirement of the switch with which the system must interface.

Whether the echo canceller is designed for synchronous operation or not, the transmitter *must* be synchronized to the received signal in order to ensure that the digital switch with which the modem located at the central office interfaces, transmits and receives data at the same rate. Thus, synchronization is always a requirement of the environment in which the system is to be used, even if it is not a requirement for its proper internal operation.

The technique chosen in this design to perform timing recovery has been extensively used in PCM repeaters, where it has proved to be very reliable. By means of a nonlinear operation (which in practice is generally rectification) spectral components not originally present in the received signal are created. More specifically it will be shown in Chapter 3 that a discrete spectral line at the

data rate is created. This line is recovered from other undesired spectral components using a high Q resonant circuit and the resultant sinusoidal signal used for synchronization.

Use of this technique in a hybrid modem requires that timing recovery be done after echo cancellation. Since at this point we have a sampled-data signal, a continuous-time signal must be obtained prior to the rectification to avoid interference from the aliases of the signal, as will be shown in more detail later. It also requires that sampling be done at least at twice the data rate, in order to avoid aliasing distortion of the received signal, that would destroy the desired timing information.

The spectrum of the received signal is bandlimited by the transmit and receive filters to a value between half the data rate and the data rate. It can be shown that the timing information is contained mainly in the spectral components around half the data rate. This sampling rate requirement has a profound impact on the design of the echo canceller, since it is a transversal filter and as such the required number of taps increases proportionally to the sampling rate. Furthermore, timing recovery is done after echo cancellation, and this implies that the echo canceller cannot rely on synchronous operation for its proper function. Thus the second of the two alternative approaches for the echo canceller design must be chosen. As a matter of fact, no proven techniques exist to perform timing recovery in the synchronous sampling scheme, although a number can be easily imagined, most of them based on the use of some training or initialization sequence, during which timing is acquired, and some control loop used later to correct small departures from the correct sampling phase during normal operation. However, use of these techniques was considered potentially unreliable and so the timing recovery scheme based on oversampling was adopted.

A detailed mathematical analysis of the timing recovery technique adopted for our experimental system is delayed until nonlinear echo cancellation techniques are introduced (Chapter 3). In Section 3.4 we carry out the analysis of timing recovery in sampled-data systems, and find expressions for the intensity of the timing tone and the jitter. Similar results for a continuous time system are given in reference [27].

2.1.8. Synchronization in Subscriber and Central Office Sets

As shown in Fig. 2.12, the Central Office transmitter receives its clock from the switch. The subscriber receiver synchronizes to the received signal and thus is also slaved to the switch clock. The subscriber transmitter is also synchronized to the recovered clock, except at the beginning of the transmission when the echo canceller has not converged yet, and so a timing signal is not available. During this time the subscriber transmitter clock is free-running. Finally, the Central Office receiver recovers its timing from the signal received from the subscriber. This is so because, although the correct frequency is available from the switch, the correct sampling phase depends on the delay of the line, and consequently it must be determined from the received signal. Thus, the timing information follows an open loop: from Central Office transmitter to subscriber receiver, from here to subscriber transmitter, and finally to Central Office receiver (Fig. 2.12).

2.2. SYSTEM DESCRIPTION

In this Section all the functions to be implemented in the system are described and the motivations that justify all the design decisions are given. Since this description is mainly based on an breadboard implementation, details

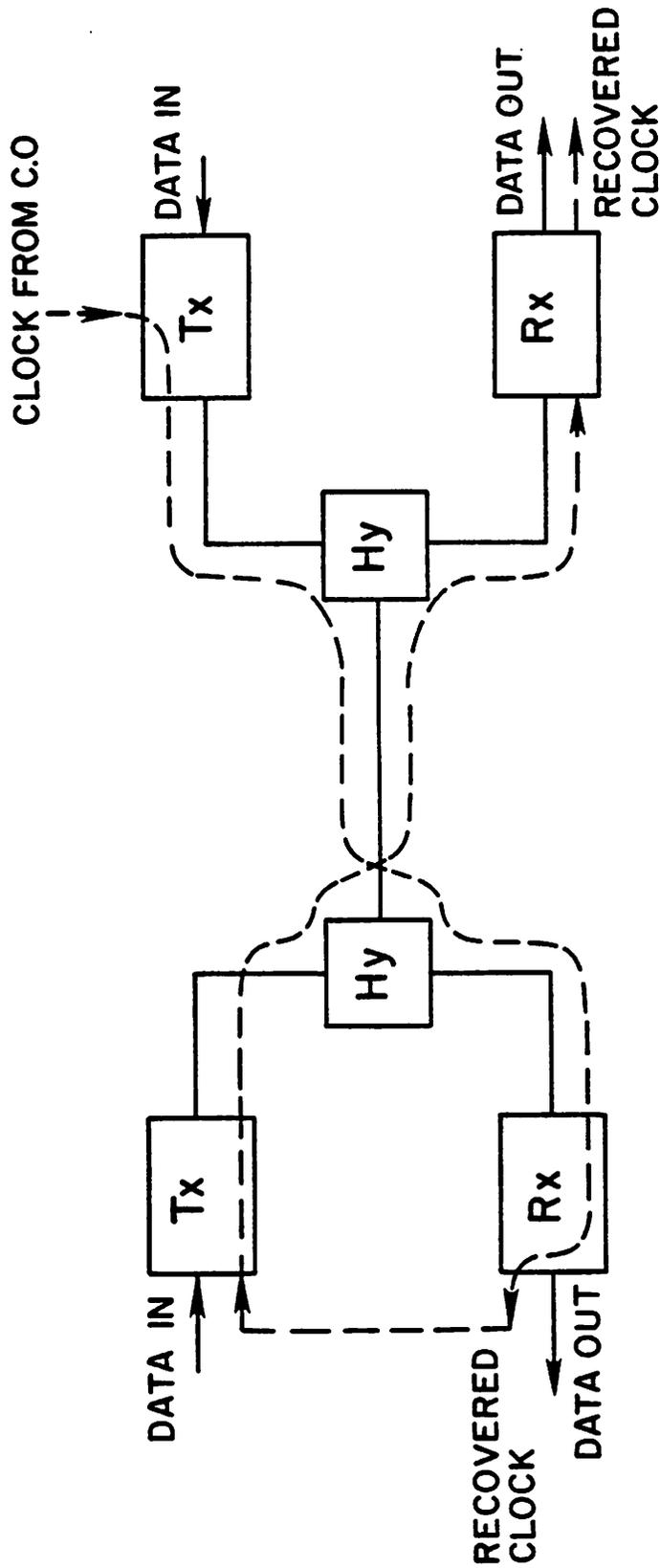


Fig. 2.12. Timing information flow in the digital subscriber loop. Timing originates at the central office, is recovered at the subscriber receiver and used by the subscriber transmitter. The central office receiver must lock to the subscriber transmitter because correct sampling phase depends on the line delay.

related to the actual circuit implementation will be given only occasionally, when it is considered that they help to clarify either system-level concepts or solutions which are valid also for an LSI implementation. Details specifically related to the technology chosen in this particular implementation, and not considered applicable for an LSI implementation, will be omitted. However, circuit schematics, state and timing diagrams and ROM tables are given in Appendix D.

Fig. (1.4) shows a block diagram of the system under consideration. Two baseband modems communicate at 80 kb/s on a single pair of wires. One of them is located at the central office, the other at the subscriber set. Typical specifications include a transhybrid loss of 10 dB, a line attenuation of up to 40-45 dB for a 5 km subscriber loop (measured at half the data rate, where the spectral density peaks for the bipolar coded signal), and a required echo cancellation of 50-55 dB for 20 dB signal to noise ratio after cancellation. Even greater cancellation of the echo would be desirable if it could be achieved.

The system configuration which was chosen is shown in greater detail in Fig. (2.13). The following subsections discuss the design choices which were made.

2.2.1. Scrambler and Descrambler

Scrambling the incoming data ensures that the transmitted sequence is random (or pseudo-random) even during idle or repetitive data patterns [28,31,32]. A random data sequence is required for the convergence of the echo canceller, to avoid placing discrete spectral components on the line (that would cause RFI and crosstalk interference), as well as to aid timing recovery in the receiver. The particular scrambler chosen is recommended by CCITT and performs a modulo 2 division by the polynomial $1+x^3+x^{20}$ [32], generating a pseudo-random sequence of length $2^{20}-1$. The descrambler, which is self-synchronizing, performs a modulo 2 multiplication by the same polynomial. If a constant binary value is placed at the scrambler input, a periodic sequence

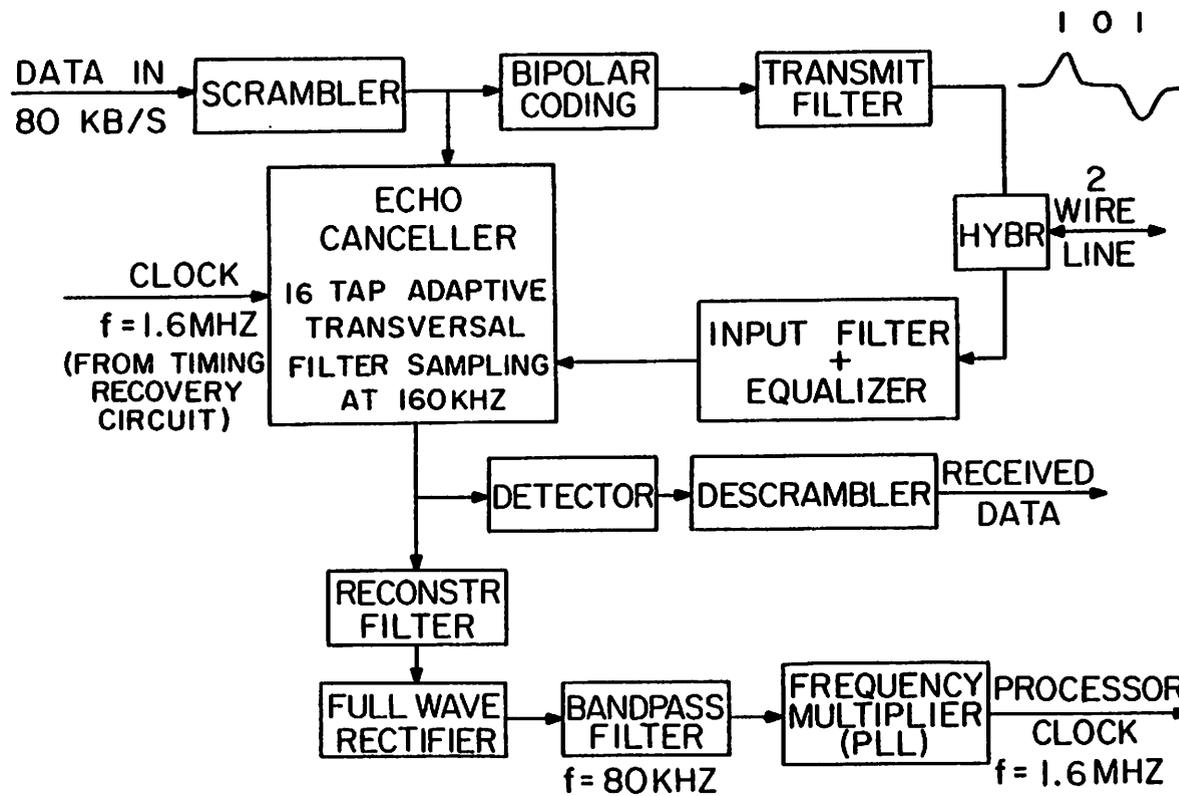


Fig. 2.13. Detailed block diagram of the transceiver.

whose period is $2^{20}-1$ will be generated. The spectrum of this periodic sequence consists of a set of discrete lines with a spacing $\frac{f_b}{(2^{20}-1)}$. Because of the close spacing between lines, the spectrum can be considered continuous for all practical purposes. In the case in which the input sequence is not constant, even longer sequences with even closer line spacing are generated. The envelope of this quasi-continuous spectrum is determined by the pulse shape. For ideal *delta* pulses the envelope would be a constant, and the spectrum would be similar to a white noise spectrum. For square pulses, the envelope is a

$$\left| \frac{\sin\left(\frac{\pi f}{f_b}\right)}{\left(\frac{\pi f}{f_b}\right)} \right|$$

curve. Further spectral shaping is introduced by the line coder (which in our implementation is a linear filter, as described in Section (2.2.2)), and the output transmitter filter.

2.2.2. Bipolar Coder

Alternative line codes have recently received attention in the literature [33], and will not be discussed here. "Bipolar" or "alternate mark inversion (AMI)" coding has been chosen in this design in view of its simplicity and robustness; however, most of the techniques used here are readily adaptable to other codes. The bipolar transmitted signal is three-level, and it was desired to avoid building a canceller which accepted three-level data, not only because of its additional hardware complexity, but also because bipolar coding introduces correlation between bits which degrades the performance of the adaptation algorithm (in the derivation of the adaptation algorithm it is explicitly assumed that the input data bits are uncorrelated). One possibility is to input the same data to the line coder and the echo canceller. However, since the coder is then located in the echo path, that echo path would then be nonlinear if true bipolar

coding were utilized. A modified technique is therefore used here, which consists simply of differentiating the input binary signal using an analog differentiator with a time constant much shorter than the baud period. A three-level signal is thus created which obeys the same constraint as the standard bipolar signal; namely, that the transmitted signal has no *dc* content. In addition, since the line coding is obtained through a linear operation, it is consistent with linear echo cancellation. In practice the analog differentiator is incorporated in the output filter making it into a *bandpass* instead of a *lowpass* filter. Thus the block named "bipolar coder" has conceptual rather than physical existence in the hardware implementation.

It should also be noted that the coding could be made into true bipolar by performing a modulo 2 running sum of the data sequence before entering it into the canceller and the transmit filter with differentiator, as suggested in [34] and shown in Fig. (2.14). This additional complexity was not included, however.

2.2.3. Transmit and Receive Filters

Minimal intersymbol interference filters [52] were used at the transmitter output and the receiver input.

One of the alternatives considered was to use minimal intersymbol interference *matched* filters [50]. However use of matched filters significantly complicates the implementation because of the requirement of an all-pass phase correction network in the receiver filter in order to achieve overall linear phase response. The additional complication was not considered justified in view of the relatively small improvement in performance.

Furthermore, low intersymbol interference is not the only consideration in the design of these filters. Intersymbol interference is defined by Equation (2.31). Thus, intersymbol interference is only related to the samples of the impulse response at multiples of the symbol period. Samples of the impulse

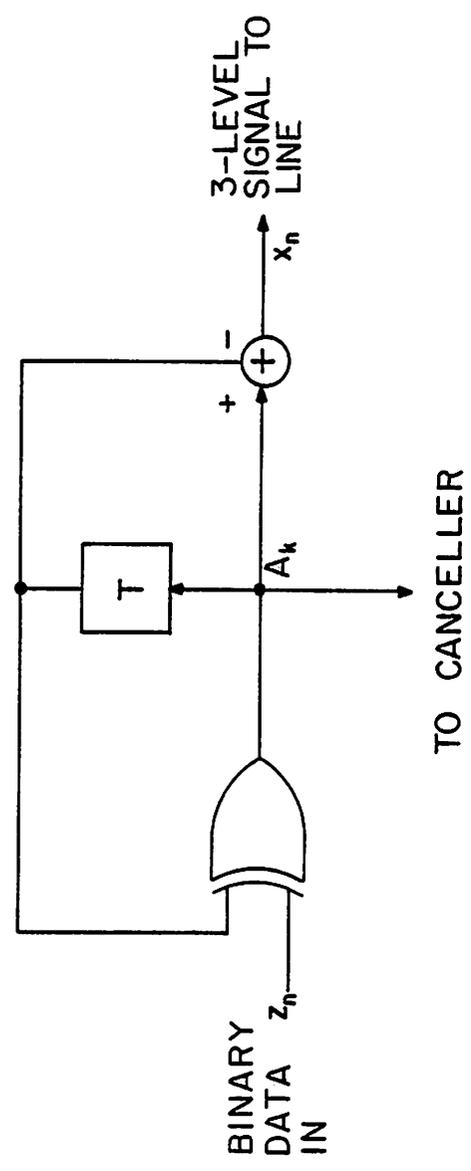


Fig. 2.14. Bipolar coder.

response taken at other values of t can be relatively large even for large values of t , in the case of a filter with a long impulse response. However, it is not desirable to build a filter with a long impulse response, because this will require an increase in the number of echo canceller taps, since both the transmit and the receive filters are located inside the echo path. The echo canceller samples at twice the data rate, and so samples are taken also at points other than the zero crossings of an ideal zero intersymbol interference impulse response. The optimization criterion used in the design of minimal ISI filters does not take into account the impulse response length. Thus judicious design requires the analysis of the filter impulse response and some iteration until the desired results are achieved.

The transmit filter shapes the pulse placed on the line to minimize high frequency components that would unnecessarily increase crosstalk and radio frequency interference. Care must be exercised in its design to keep nonlinear distortion components (which a linear echo canceller cannot compensate) lower than approximately -60 dB relative to the transmitted signal level. This also requires that the input pulses be symmetrical to a similar degree. Perfectly square pulses would be desirable, but unfortunately the output of a logic gate departs significantly from that ideal due to rising and falling transients that are not equal in general. Satisfactory pulses have been obtained by clamping a CMOS gate with diodes as shown in Fig. 2.15.

Because of the high degree of symmetry between positive and negative pulses which is required, oscilloscope measurements in the time domain are generally not sufficient to evaluate the quality of the pulse generating circuitry. A better evaluation can be made by measuring the output signal with a spectrum analyzer. For a bipolar signal, spectral zeros are to be expected at multiples of the data rate. It is shown in Appendix B that in the presence of nonlinear distortion the departure from zero of the spectral density at multiples of the

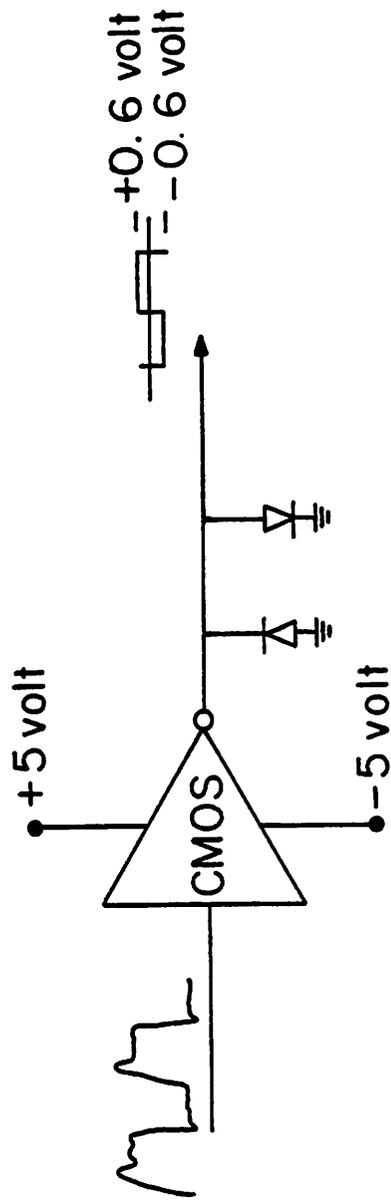


Fig. 2.16. CMOS gate clamped to generate highly symmetric pulses.

resolution characteristic which represents the limitation on the minimum achievable mean squared error due to the quantization. This characteristic takes into account the beneficial dithering effect, as said before. The finite precision characteristic shows how for very small values of uncancellable signal the mean squared error is determined by the finite precision characteristic. This is clear, since for low values of U little or no dithering effect occurs and quantization error limits the cancellation. For large values of U , however, significant dithering occurs and the infinite precision characteristic determines the mean squared error.

Also considered in reference [24] is the quantization error introduced by the D/A. This has two effects. One of them is to introduce quantization noise at the output of the echo canceller, which adds to the uncancellable signal and increases the mean squared error after cancellation. The other effect, more subtle, is to degrade the dithering effect in the A/D. This effect was studied by computer simulation in ref. [24], which reports the results shown in dotted lines in Fig. 4.3. The conclusion of this study is that at least 12 bits of resolution are required in the D/A, whereas 8 bits are required in the A/D.

4.3. EFFECT OF D/A NONLINEARITY

Linearity requirements of the data converters in the configurations of Figs. 4.1b, c, and d have not been reported in the literature, perhaps because previous designs have been oriented towards discrete implementations or monolithic implementations with off-chip data converters. If the constraint to use a process-compatible monolithic A/D and D/A implementation does not exist, one is free to choose high performance, absolute accurate converters, and no need arises to consider the effect of nonlinear distortion. However when a monolithic MOS implementation of the complete digital subscriber loop is considered, performance limitations imposed by the data converters must be accounted for.

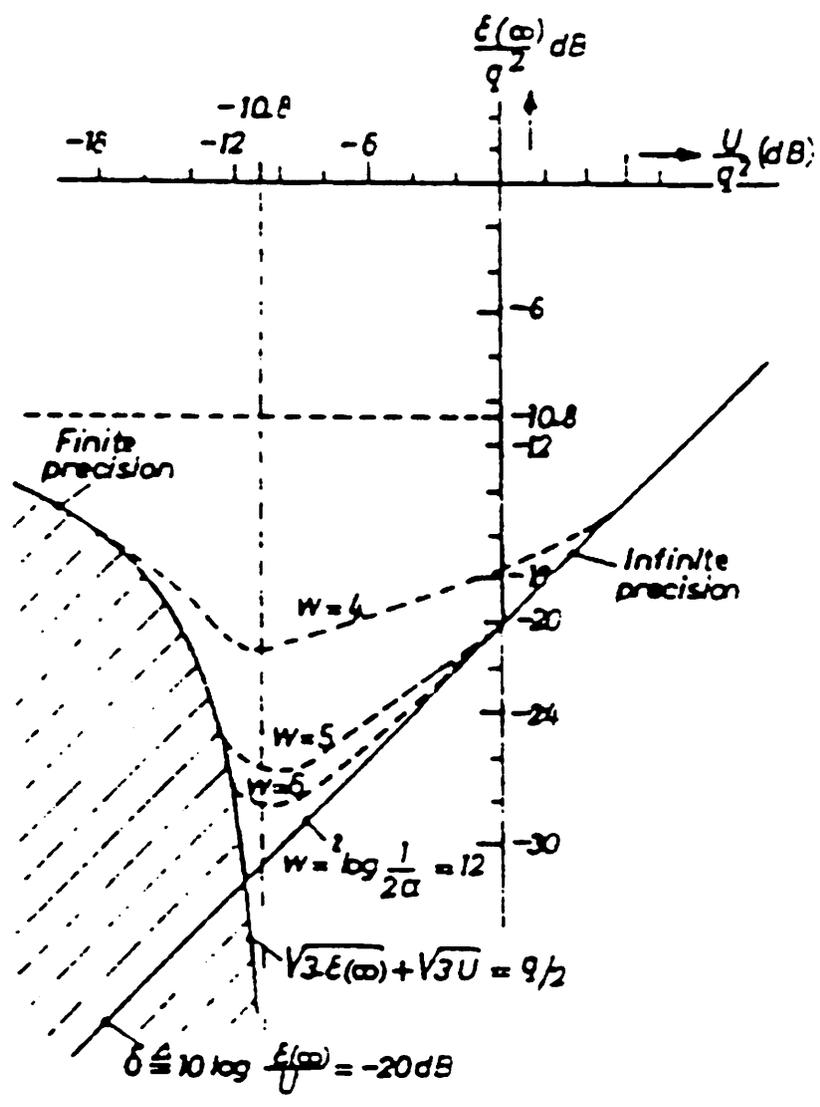


Fig. 4.3. MSE as a function of the quantization step.

4.2.1. Resolution of the A/D converter of Fig. 4.1a

The input A/D converter must have a large dynamic range in order to be able to handle the worst case situation of a relatively small signal superimposed on a large echo. The A/D must be able to resolve the small signal with low quantization error and still not clip the large signal. Assuming that the input signal is dominated by the echo and its level is adjusted to use all the available dynamic range of the A/D, the quantization error should be much smaller than the desired residual echo after cancellation, which is in the order of 50 to 60 dB. If q is the quantization error and S_{\max} is the peak value of the input signal, the required A/D resolution $W_{A/D}$ is determined from the relation

$$-(2^{W_{A/D}-1}-1) \leq \frac{S_{\max}}{q} \leq (2^{W_{A/D}-1}-1) \quad (4.1)$$

which yields

$$W_{A/D} = \text{int} \left[\log_2 \left(\frac{S_{\max}}{q} + 1 \right) \right] + 1 \quad (4.2)$$

where "int" stands for *integer part of*. For a signal to quantization noise larger than 60 dB this equation gives $W_{A/D} = 12$ bits. Even more resolution would be desirable in order to achieve larger signal to noise ratios.

4.2.2. Resolution of the A/D and D/A in Figs. 4.1b and 4.1c

These implementations have similar requirements. Both the A/D and the D/A must be considered. This problem has been studied in detail by Verhoeckx *et al.* They show how the dithering effect that the uncancellable component of the signal generates on the cancellation error greatly relaxes the resolution requirement of the A/D. Fig 4.3, reproduced from that reference, shows the MSE as a function of the quantization step q of the A/D. Shown in the Figure are both the infinite resolution characteristic, which represents the MSE as a function of the uncancellable signal as given by Eq. (2.26) (here the factors affecting the MSE are the number of taps N and the gain factor α) and the finite

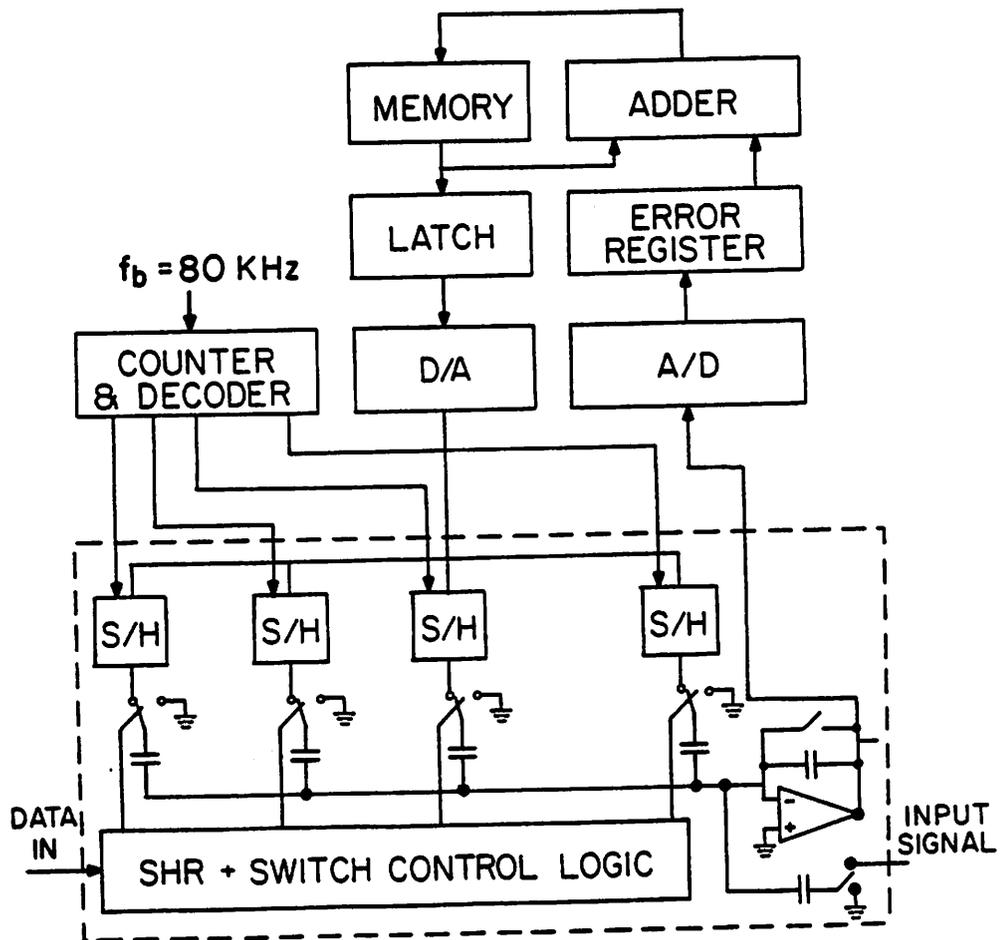


Fig. 4.2. Analog/digital echo canceller in more detail.

4.1.4. Analog/Digital Echo Canceller

An architecture in which the convolution sum and the echo cancellation are performed with analog techniques is shown in Fig. 4.1d. The adaptation algorithm is performed digitally, and the tap weights are converted to analog using a single time-shared D/A converter. The multiplications and additions are then performed in the charge domain, where the nonlinear distortion can be kept very small without special circuit design effort.

Since a large dynamic range and long time-constant are still required in the adaptation algorithm, it cannot be implemented by analog means. Thus the use of a digital processor to perform the adaptation, while a switched capacitor analog transversal filter computes the convolution sum and cancels the echo in the received signal. The coefficients of the transversal filter are stored in sample and hold (S/H) capacitors (Fig. 4.2) and refreshed periodically by the digital adaptation processor. In order to reduce the required speed of the D/A converter, only one of the S/H capacitors is refreshed each pulse period; however, all are updated in the digital processor. The consequences of this for echo canceller convergence are shown in Section 4.3 to be negligible. The possibility of using a slow D/A converter in this approach represents a significant advantage, since slow operation can be traded off for increased resolution and reduced die area and power consumption.

4.2. RESOLUTION REQUIREMENTS OF THE DATA CONVERTERS

The organization shown in Fig. 4.1a obviously does not require any data converters, and so it will not be considered here. The organizations of Figs. 4.1c and 4.1d have completely similar resolution requirements, thus they will be addressed together. In short, only two cases must be considered, namely, that of Fig. 4.1a and those of Figs. 4.1c and d.

4.1.2. Fully Digital Echo Canceller

In this approach, shown in Fig. 4.1b, the canceller is implemented entirely with digital logic, and the received signal is sampled and A/D converted prior to the cancellation. This technique was used in the first breadboard version of the echo canceller described in Section 2.2.5. A monolithic implementation of this technique is impractical at present because A/D converters of adequate resolution and linearity have not been demonstrated in MOSLSI. Advances in monolithic data converters may change this situation in the future. For this reason, we will further analyze the effect of nonlinearities in the A/D converter in Section 4.3.

4.1.3. Digital Echo Canceller with Analog Cancellation

This organization, shown in Fig. 4.1c, uses an all-digital implementation of the canceller and converts the echo replica to analog using a D/A converter prior to cancellation in the analog domain. MOS monolithic D/A converters of the required speed and resolution have been demonstrated [56]; however, the integral linearity of these converters cannot be guaranteed to be better than 7 or 8 bits unless laser trimming is used. Nonlinear distortion in the D/A greatly degrades the echo cancellation, as shown in Section 4.3. The requirement for a low cost implementation of the system precludes the use of laser trimming, making this alternative not directly applicable to the monolithic implementation of the echo canceller. Nonlinear echo cancellation techniques as described in Chapter 3 could be used to precompensate the D/A distortion. Alternatively, a self-correcting D/A can be used.

Since the A/D is used only to generate a feedback signal for the adaptation algorithm, it must be monotonic, but not necessarily linear.

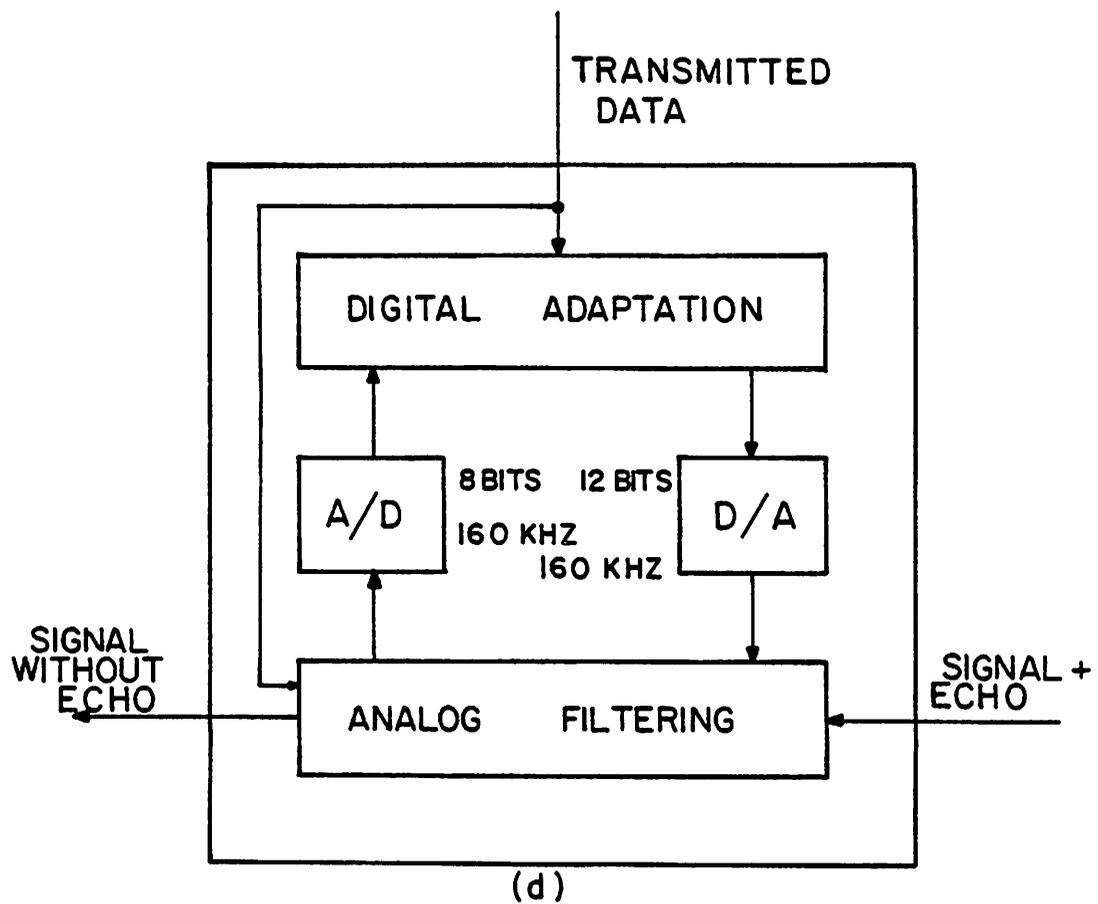


Fig. 4.1d. Analog/digital echo canceller.

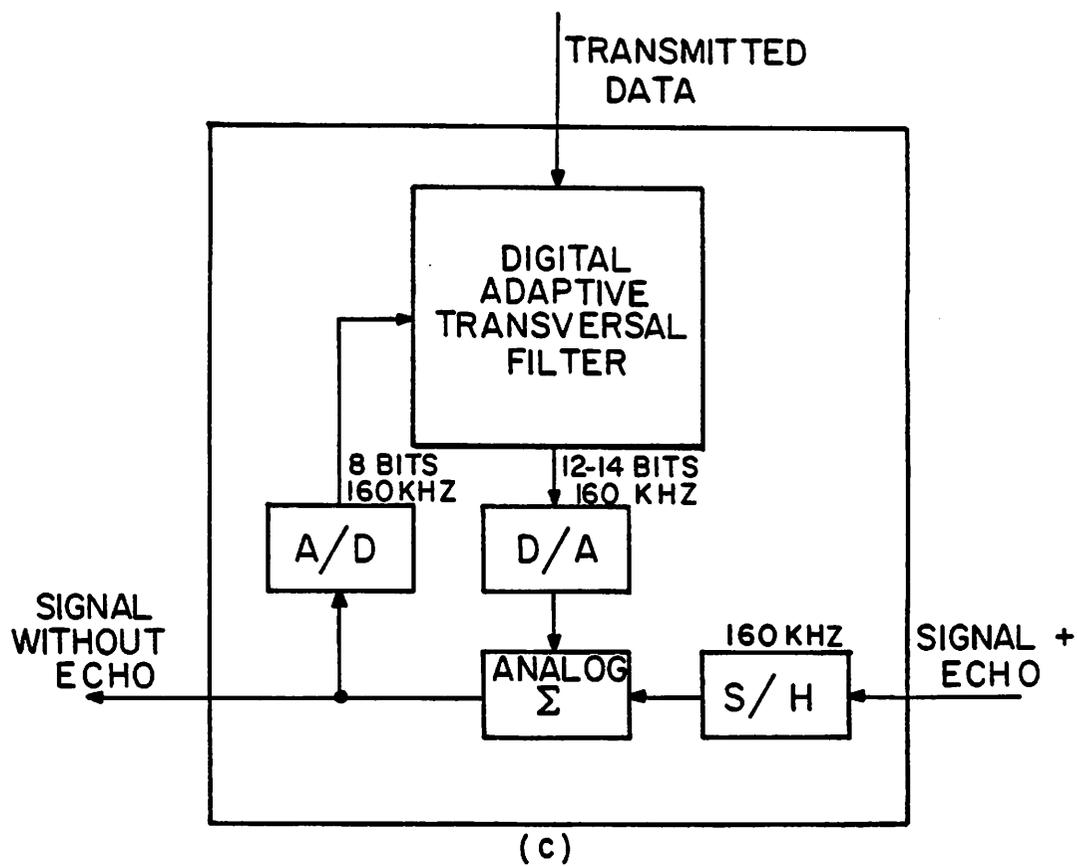


Fig. 4.1c. Digital transversal filter with analog cancellation.

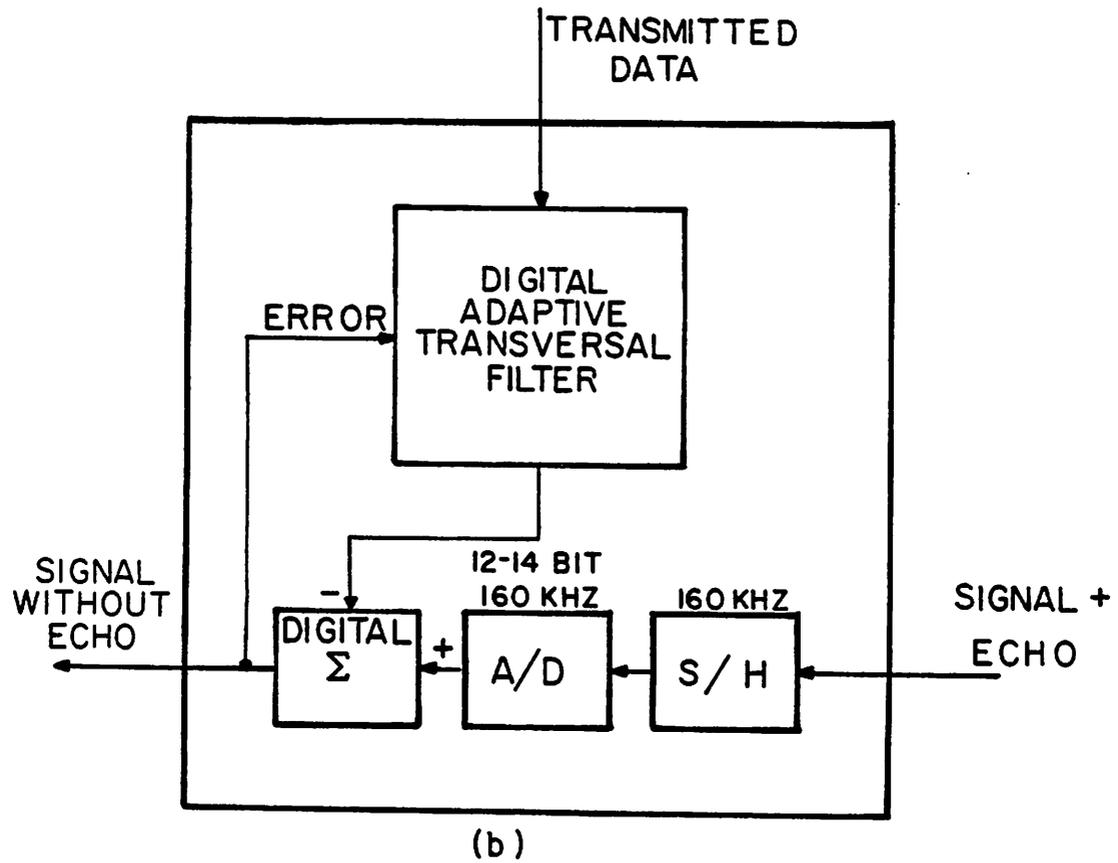


Fig. 4.1b. Digital echo canceller.

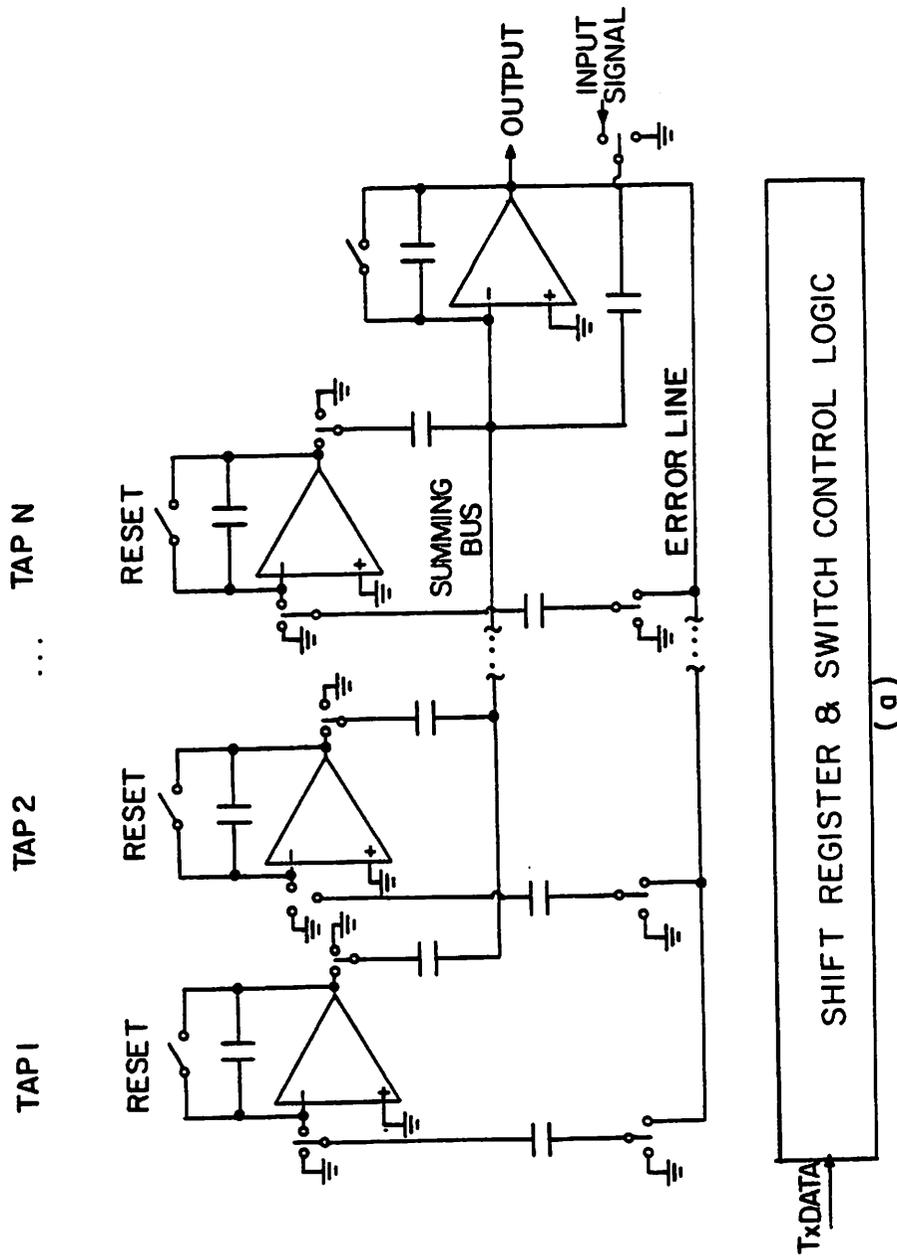


Fig. 4.1a. Analog echo canceller.

ments and a monolithic echo canceller chip that implements one of the alternatives of Section 4.1.

4.1. ALTERNATIVE IMPLEMENTATIONS OF THE ECHO CANCELLER

Four alternative echo canceller configurations will be considered in this Section:

1. *Analog implementation (Figure 4.1a).*
1. *A fully digital echo canceller with digital cancellation and A/D conversion of the input signal (Figure 4.1b).*
3. *A digital echo canceller with analog cancellation (Figure 4.1c).*
4. *Analog-digital echo canceller with analog cancellation (Figure 4.1d).*

Each of these alternatives will be introduced in the following four sub-sections, and then a more thorough description will be given in Section 4.4.

4.1.1. Analog Echo Canceller

Figure 4.1a shows the structure of a completely analog echo canceller using switched-capacitor techniques. The coefficients of the transversal filter are stored in integrators, and the binary weighting is done using switched capacitors. A summing amplifier performs the convolution sum. The adaptation algorithm is also implemented by a switched capacitor circuit that adds a correction term to the coefficients stored in the integrators. The desired correction term is obtained by multiplying the cancellation error coming from the output of the summing amplifier by either +1 or -1, corresponding to the data bit at the corresponding tap.

CHAPTER 4

TECHNIQUES FOR THE LARGE SCALE INTEGRATION OF THE SYSTEM

The implementation complexity of hybrid mode digital subscriber loops would result in a high manufacturing cost for any realization using off-the-shelf components. Thus from an economic point of view the hybrid mode technique cannot compete with the burst mode technique, in spite of the recognized technical advantages of the former [19]. The high cost is mainly associated with two sections of the echo canceller, namely

1. *The digital processor implementing the canceller adaptation and/or transversal filtering, and*
2. *The analog-to-digital and/or digital-to-analog converters.*

Large scale integration techniques have proven very effective in decreasing the cost of digital circuits. Thus, a monolithic implementation of this system will clearly overcome the cost factor associated with 1. However, the requirement of on-chip high performance data converters poses additional problems that have to be solved before the system can be fully integrated. This Chapter will address the problems associated with the data conversion in an integrated hybrid-method digital subscriber loop. Switched-capacitor filters are now a proven technique that can be used for the implementation of the various filters described in Chapter 2, and will not be considered here. In Section 4.1 we analyze available alternatives for the implementation of the echo canceller, in view of the problems associated with the data conversion. Resolution requirements of the data converters in each implementation are considered in Section 4.2. Linearity requirements are analyzed in Section 4.3, and a detailed description of each alternative is given in Section 4.4, including breadboard experi-

tabulated.

It is interesting to observe that the amplitude of the timing tones is determined exclusively by the *dc* component of the nonlinear impulse response vector $\mathbf{F}[\mathbf{h}^{(r)}]$. The continuous spectrum is responsible for the *jitter* in the timing waveform. In order to minimize the jitter, the nonlinear function $f(\cdot)$ could be chosen so as to maximize the *dc* component of $\mathbf{F}[\mathbf{h}^{(r)}]$, and minimize all the other components. Jitter is clearly a consequence of the pulse overlap, as can be seen from the fact that it vanishes when $N=1$.

$A_{00}(m)$ is 1 for all m . The other elements of $A(m)$ are either 0 or 1 and all vanish for $|m| \geq N$. Thus we can express

$$A(m) = B + C(m) \quad (3.64)$$

where

$$B = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & \cdots & 0 \\ & & & \ddots & \\ 0 & 0 & 0 & \cdots & 0 \end{bmatrix} \quad (3.65)$$

and $C(m)$ is identically 0 for $|m| \geq N$ but is different from 0 for $|m| \leq N$. It is interesting to note also that $C(m)$ depends only on N , and so a universal table of $C(m)$ matrices could be computed as a function of N .

Now the power spectrum is:

$$\begin{aligned} X(\omega) &= \sum_{m,r} \mathbf{F}[\mathbf{h}^{(0)}]^T (B + C(m)) \mathbf{F}[\mathbf{h}^{(r)}] e^{-j\omega T \left[m + \frac{r}{R} \right]} \\ &= \sum_{r=0}^{R-1} \mathbf{F}[\mathbf{h}^{(0)}]^T \left\{ \sum_{m=-\infty}^{\infty} (B + C(m)) e^{-jumT} \right\} \mathbf{F}[\mathbf{h}^{(r)}] e^{-j\frac{\omega T r}{R}} \\ &= \sum_{r=0}^{R-1} \mathbf{F}[\mathbf{h}^{(0)}]^T \left\{ \frac{2\pi}{T} \mathbf{B} \sum_{k=-\infty}^{\infty} \delta\left(\omega - \frac{2\pi k}{T}\right) + \sum_{m=-\infty}^{\infty} C(m) e^{-jumT} \right\} \mathbf{F}[\mathbf{h}^{(r)}] e^{-j\frac{\omega T r}{R}} \end{aligned} \quad (3.66)$$

Considering the form of B , we see that:

$$\begin{aligned} X(\omega) &= \left[\sum_{r=0}^{R-1} \left([\mathbf{F}[\mathbf{h}^{(0)}]]_0 [\mathbf{F}[\mathbf{h}^{(r)}]]_0 \right) e^{-j\frac{\omega T r}{R}} \right] \frac{2\pi}{T} \left[\sum_{k=-\infty}^{\infty} \delta\left(\omega - \frac{2\pi k}{T}\right) \right] + \\ &+ \sum_{r=0}^{R-1} \mathbf{F}[\mathbf{h}^{(0)}]^T \left[\sum_{m=-N+1}^{N-1} C(m) e^{-jumT} \right] \mathbf{F}[\mathbf{h}^{(r)}] e^{-j\frac{\omega T r}{R}} \end{aligned} \quad (3.67)$$

The first term generates an infinite set of discrete lines at all multiples of the data rate, whose amplitude is given by:

$$Q(\omega) = \frac{2\pi}{T} \sum_{r=0}^{R-1} \left([\mathbf{F}[\mathbf{h}^{(0)}]]_0 [\mathbf{F}[\mathbf{h}^{(r)}]]_0 \right) e^{-j\frac{\omega T r}{R}} \quad (3.68)$$

The second term generates a continuous spectrum, which can also be written as:

$$\sum_{r=0}^{R-1} \mathbf{F}[\mathbf{h}^{(0)}]^T \mathbf{M}(\omega) \mathbf{F}[\mathbf{h}^{(r)}] e^{-j\frac{\omega T r}{R}} \quad (3.69)$$

where $\mathbf{M}(\omega)$ is a universal matrix which depends only on N (the dimension of $\mathbf{M}(\omega)$ is $2^N \times 2^N$). This matrix could be computed once and forever and

generated by the timing recovery circuit. To analyze this in detail, consider samples of a data signal

$$\mathbf{s}_n^{(r)} = \sum_{k=0}^{N-1} h^{(r)}(k) x_{n-k} \quad (3.59)$$

where x_n is some sequence of received bits and $h^{(r)}(k)$ is the channel impulse response (including any shaping filters in the transmitter and receiver). The superscript (r) identifies each one of the R samples that are taken in each period T of the data signal. Thus R is the *oversampling factor*, and runs from 0 to $R-1$. This will be assumed in all the equations that follow even if it is not explicitly indicated. Using the generalized nonlinear notation of the previous sections, the signal can be written as

$$\mathbf{s}_n^{(r)} = \mathbf{x}_n^T \cdot \mathbf{h}^{(r)} \quad (3.60)$$

This notation allows for nonlinearities in the channel. Although these nonlinearities do not affect the validity of the results that will be obtained here, we prefer to consider that the channel is linear. To perform timing recovery, we generate some nonlinear function of these samples. Typically, such a nonlinear function $f(\cdot)$ is a full-wave rectification. Using identity (3.20), we can write:

$$f(\mathbf{s}_n^{(r)}) = \mathbf{x}_n \cdot \mathbf{F}[\mathbf{h}^{(r)}] \quad (3.61)$$

where $\mathbf{F}[\mathbf{h}^{(r)}]$ is a 2^N -dimensional nonlinear transformation induced by $f(\cdot)$ on the vector $\mathbf{h}^{(r)}$. We want to compute the spectrum of the sequence (3.61). The power spectrum of this sequence is the Fourier transform of its autocorrelation function. The autocorrelation function is:

$$r\left(m + \frac{T}{R}\right) = \mathbf{F}[\mathbf{h}^{(0)}]^T E\{\mathbf{x}_n \mathbf{x}_{n+m}^T\} \mathbf{F}[\mathbf{h}^{(r)}] \quad (3.62)$$

where E stands for expectation value.

Now consider the matrix

$$\mathbf{A}(m) = E\{\mathbf{x}_n \mathbf{x}_{n+m}^T\} \quad (3.63)$$

This is a $2^N \times 2^N$ matrix. Because component 0 of vector \mathbf{x}_n is always 1, element

TABLE II							
TAP NUMBER	PRODUCT TERM	ALWAYS > 0.005	ALWAYS > 0.001	ALWAYS > 0.0005	ALWAYS > 0.0001	ALWAYS > .00005	ALWAYS > .00001
0	C0	1	1	1	1	1	1
1	C1	1	1	1	1	1	1
2	C2	1	1	1	1	1	1
3	C3	1	1	1	1	1	1
4	C4	1	1	1	1	1	1
5	C5	1	1	1	1	1	1
6	C6	0	1	1	1	1	1
7	C7	0	1	1	1	1	1
8	C8	0	0	1	1	1	1
9	C9	0	0	0	1	1	1
10	1	0	1	1	1	1	1
11	C0C1	0	1	1	1	1	1
12	C0C2	0	0	1	1	1	1
13	C0C3	0	0	0	1	1	1
14	C0C4	0	0	0	0	1	1
15	C1C2	0	0	1	1	1	1
16	C1C3	0	0	0	1	1	1
17	C1C4	0	0	0	1	1	1
18	C2C3	0	0	0	1	1	1
19	C2C4	0	0	0	1	1	1
20	C3C4	0	0	0	0	1	1
21	C0C1C2	0	1	1	1	1	1
22	C0C1C3	0	0	1	1	1	1
23	C1C2C3	0	0	0	1	1	1
24	C0C2C3	0	0	0	1	1	1
25	C0C1C2C3	0	0	0	1	1	1

Table II. Maximum values of taps over the 100 runs of Fig. 3.12.

TABLE Ia					
$f(x)=1.01333x-0.01333x^2$					
TAP NUMBER	PRODUCT TERM	NUMERICAL VALUE(*)	TAP NUMBER	PRODUCT TERM	NUMERICAL VALUE(*)
0	C0	0.4454855025	13	C10C03	0.0000619562
1	C1	0.2010037839	14	C002C04	0.0000592478
2	C2	0.0904013440	15	C001C05	0.0000569880
3	C3	0.0403234402	16	C100C04	0.0000282178
4	C4	0.0182554312	17	C003C04	0.0000270387
5	C5	0.0082028458	18	C002C05	0.0000269134
6	C6	0.0036656938	19	C001C06	0.0000267841
7	C7	0.0016559253	20	C100C05	0.0000125894
8	C0C1C2	0.0009801916	21	C100C04	0.0000124928
9	C8	0.0007441015	22	C000C03	0.0000121263
10	C0C1C3	0.0002908728	23	C001C07	0.0000117915
11	C0C2C3	0.0001319754	24	C000C05	0.0000116928
12	C0C1C4	0.0001306451	25	C000C06	-0.0000000888

TABLE Ib					
$f(x)=x-0.005x^2$					
TAP NUMBER	PRODUCT TERM	NUMERICAL VALUE(*)	TAP NUMBER	PRODUCT TERM	NUMERICAL VALUE(*)
0	C0	0.4493283908	9	C001	0.0010097334
1	C1	0.2018965483	10	C00	0.0007485757
2	C2	0.0907173627	11	C002	0.0004535968
3	C3	0.0407625360	12	C00	0.0003355653
4	C4	0.0183155797	13	C000	0.0002040363
5	C5	0.0082295339	14	C004	0.0000915387
6	C6	0.0036978516	15	C005	0.0000415029
7	C1	0.0022468689	16	C103	0.0000001015
8	C7	0.0016816218			

(*)Numerical values shown are for the infinite resolution case.

TABLE I. Order in which taps were included in the simulations and their numerical value. (a)Case of Fig. 3.11a. (b)Case of Fig. 3.11b.

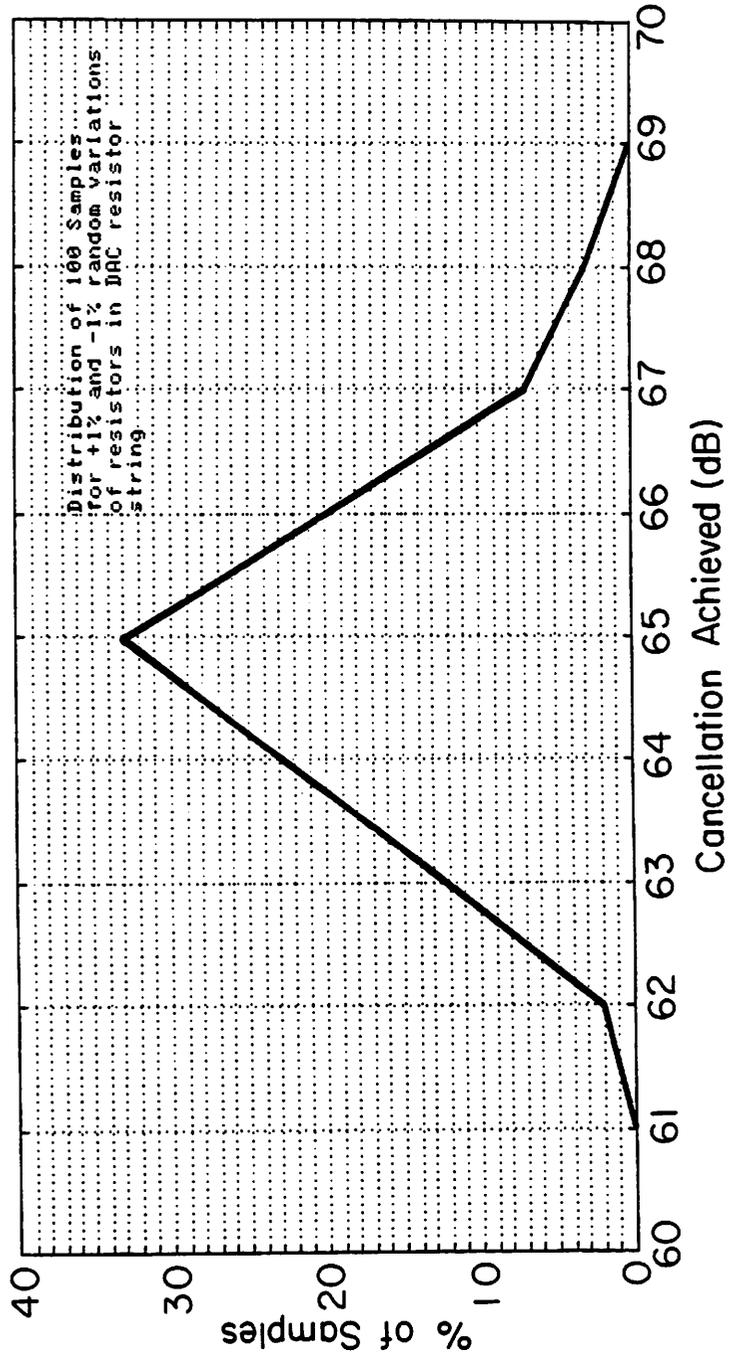


Fig. 3.12. Histogram showing the results of random +1% and -1% perturbations in resistor of the 12-bit DAC of Fig. 3.9, for 100 samples.

cases many higher order taps are more important than the linear taps beyond the tenth. The importance of the nonlinear taps depends on the number of bits of quantization. With 10 bits there is no point to using nonlinear taps in Figure 3.11a, whereas in Figure 3.11b the nonlinear taps give about a 10 dB reduction in asymptotic residual error. For 13 bits of resolution, with a modest number of nonlinear taps a 20 to 30 dB improvement can be obtained. In both cases, the number of taps is dramatically smaller than would be required in the table look-up method (1024 for a ten data bit cancellation).

The effect of random photolithographic mismatches in resistors was also simulated. Individual mismatches of either +1% or -1% (chosen randomly) were added to each of the 16 resistors of a string initially designed to implement a 12-bit D/A with the characteristic of (3.53). This level of mismatch is typical of what would be expected from an MOS process, although it is extremely unlikely that all the resistors would be simultaneously mismatched to this degree. This type of mismatch leads to a continuous piecewise linear characteristic in the D/A. In each simulation the same set of 26 taps shown in Table II were used (although many of them were very small). A histogram of the residual error in 100 randomly chosen mismatches is presented in Fig. 3.12. There is a considerable spread of 6 dB in the residual error, due to the choice of the same taps in each case (changing which taps were implemented for each random mismatch would presumably narrow this range and improve the cancellation). Table II shows the distribution of the maximum absolute values of the taps. Here a 0 in the intersection of column .001 and row $x_0x_1x_2x_3$ means that the corresponding tap was smaller than .001 (in absolute value) for all the 100 samples.

3.4. APPLICATION TO TIMING RECOVERY

The nonlinear expansion of Eq. (3.1) can also be used to obtain useful expressions for the intensity of the spectral line and the continuous spectrum

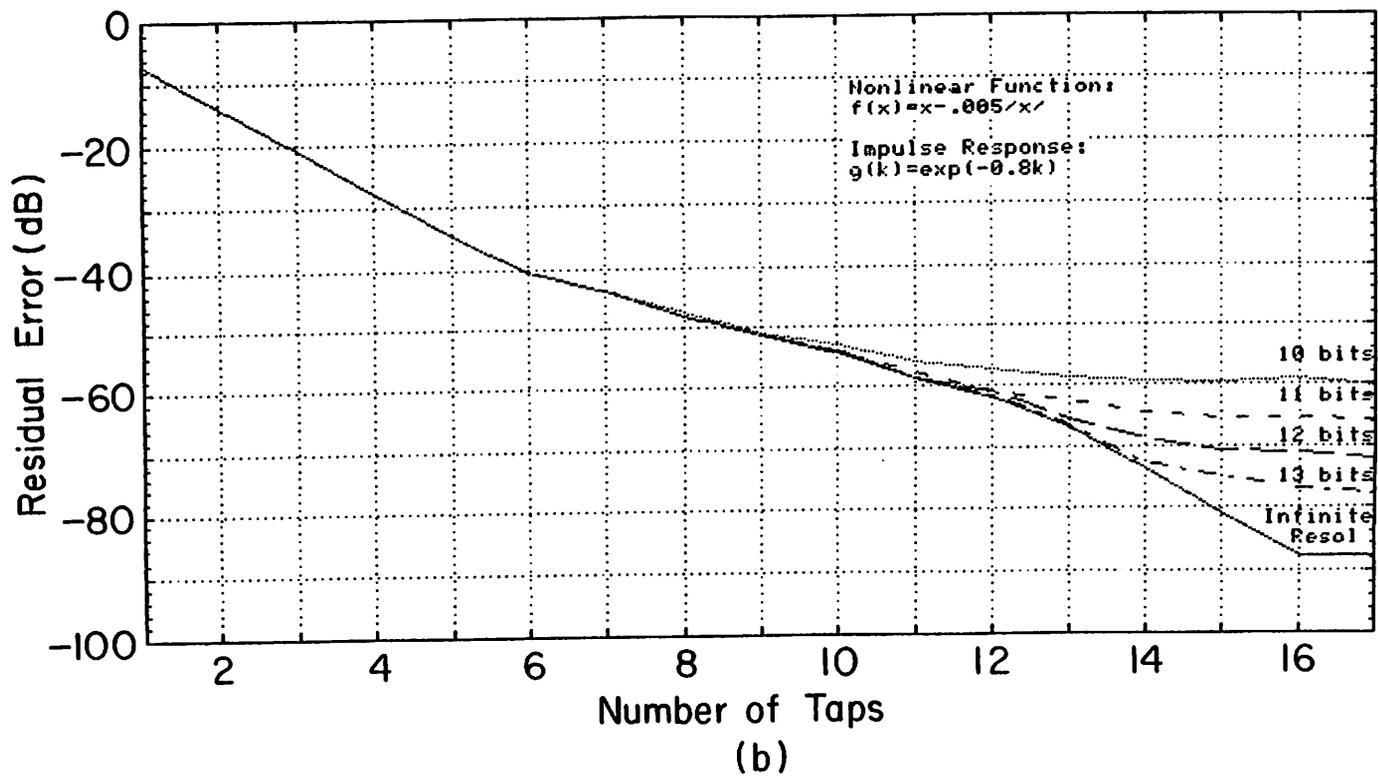


Fig. 3.11b. Residual error as a function of the number of taps for the non-linearity of Fig. 3.10b. The order in which taps are introduced is shown in table Ib.

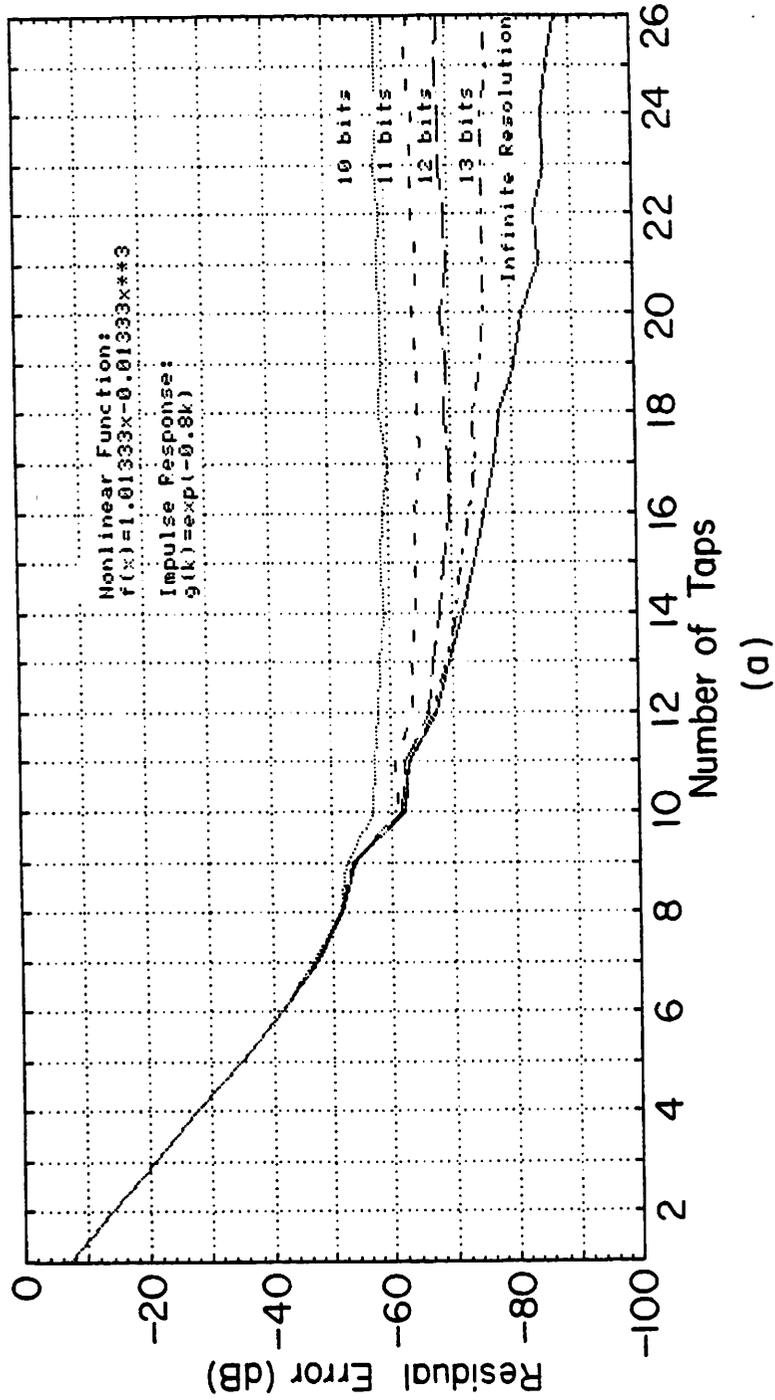


Fig. 3.11a. Residual error as a function of the number of taps for the non-linearity of Fig. 3.10a. The order in which taps are introduced is shown in table Ia.

where $b = -0.005$. For the characteristic of (4.1), the nature of the inverse can be determined by finding a power series expansion for $d^{-1}(\cdot)$. Defining this power series as

$$d^{-1}(y) = \sum_{n=0}^{\infty} b_n y^n \quad (3.55)$$

then we obtain

$$\begin{aligned} d^{-1}(d(x)) &= x \\ &= \sum_{n=0}^{\infty} b_n (ax + bx^3)^n \end{aligned} \quad (3.56)$$

Equating coefficients in (3.55) and solving for the b_n 's,

$$d^{-1}(x) = \frac{1}{a}x - \frac{b}{a^4}x^3 + \frac{3b^2}{a^7}x^5 - \dots \quad (3.57)$$

and we see that the third-order nonlinearity predominates in $d^{-1}(\cdot)$ as it does in $d(\cdot)$. The even harmonics are missing since the characteristic of (3.53) has odd symmetry about the origin. Thus, we expect that the important terms in the Volterra series expansion will be the first and third order terms. For the characteristic of (3.54), the easiest method is to find $d^{-1}(\cdot)$ directly and approximate it by a polynomial. Since the nonlinear portion of this nonlinearity has even symmetry, the odd powers will be missing, and the important terms in the Volterra series will be the first and second order terms. These conclusions are confirmed by the simulations which follow.

The simple echo path impulse response assumed in all cases was

$$g(k) = e^{-0.8k}, \quad k \geq 0 \quad (3.58)$$

The effect of quantization was also included. Runs with 10, 11, 12 and 13 bits and with infinite resolution were done, varying the number of taps from 1 to 26 in the case of Fig. 3.10a and from 1 to 17 for Fig. 3.10b. As previously mentioned, the taps were added in the order of decreasing absolute value as determined by another program. The resulting order is shown in Tables Ia and Ib together with the residual cancellation error in Figures 3.11a and 3.11b. Observe that in both

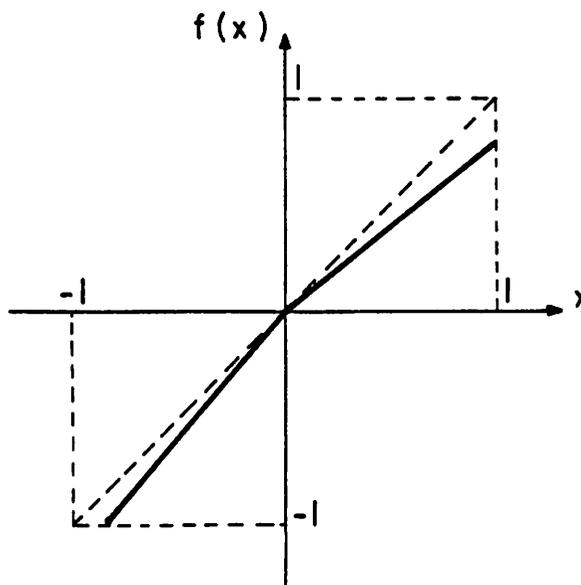
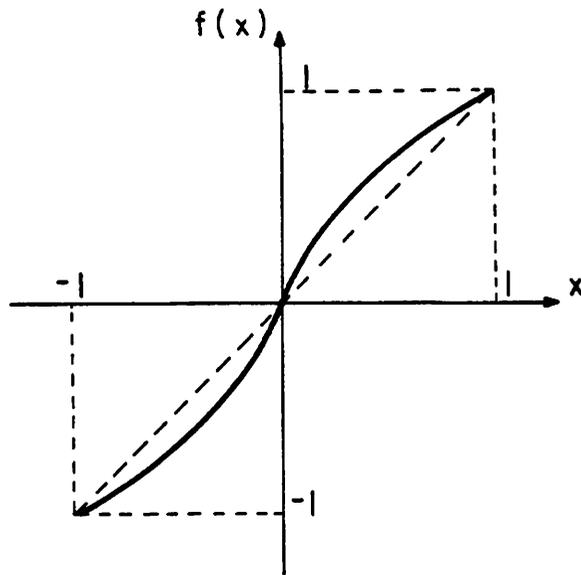


Fig. 3.10. Typical DAC transfer functions. (a) $f(x) = 1.01333x - 0.01333x^3$ (b) $f(x) = x - 0.005/x$

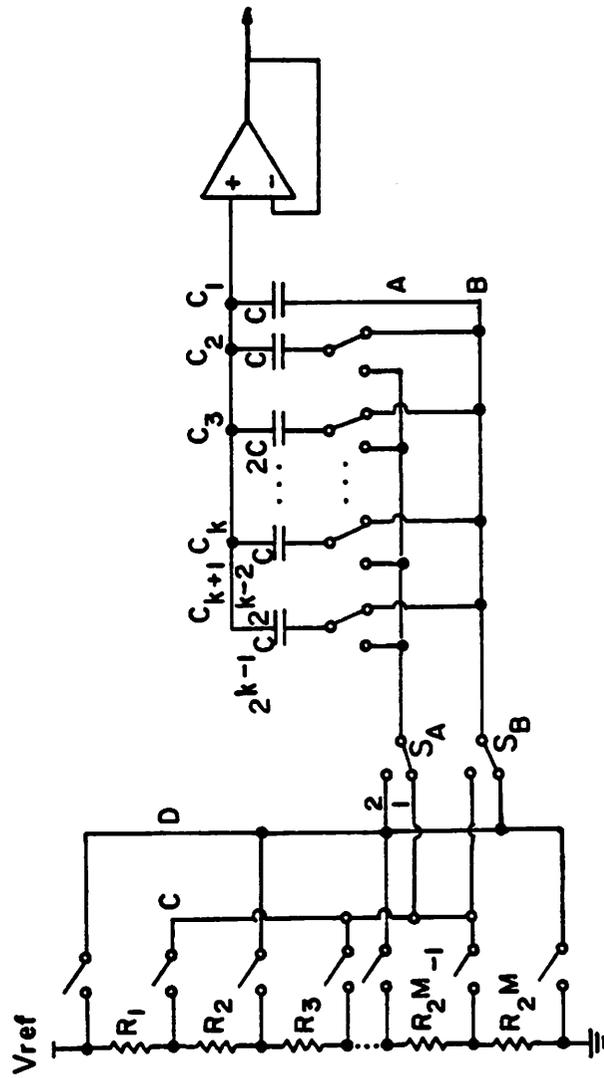


Fig. 3.9. DAC using resistor string and capacitor array. (Switch control is omitted for simplicity).

adaptive echo canceller derived in Section 3.2.5 was simulated in the presence of certain nonlinearities inherent in MOS D/A converters. The purpose of simulating the canceller, rather than using the procedure described in Section 3.1.3 for finding the coefficients of the expansion, was to establish that the adaptive algorithm does indeed work properly in the presence of nonlinearities. The asymptotic mean-square echo cancellation residual error was noted as a function of the total number of taps implemented in the canceller. The particular taps which were implemented, and the order in which they were added, was determined by first running a program which calculates all the coefficients of the expansion in accordance with the procedure of Section 3.1.3 for the particular nonlinearity being studied. The taps were then added in the order of decreasing absolute value.

In order to make the numerical examples realistic, assume the D/A converter is to be implemented in MOS technology using the technique shown in Fig. 3.9. The 4 most significant bits are provided by a string of 16 diffused resistors and the remaining bits (from 6 to 9 in our simulations) by a binary weighted capacitor array. Because of diffusion concentration gradients, voltage coefficient, and photolithographic mismatches the resistors cannot be guaranteed to be equal to within 1 LSB unless laser trimming is used. Thus, in the absence of trimming, a nonlinear transfer characteristic results. This nonlinearity can have a systematic component due to concentration gradients, and a random component due to photolithographic mismatches.

Two of the most common kinds of systematic nonlinearity are shown in Fig. 3.10. We model the transfer characteristic of Fig. 3.10a by

$$d(x) = ax + bx^3 \quad (3.53)$$

where $a=1.01333$ and $b=-0.01333$, and the one in Fig. 3.10b by

$$d(x) = x + b|x| \quad (3.54)$$

$$\left[\sum_{j=0}^{N-1} h(j)x_{n-j} \right]^2 = \sum_{j_1=0}^{N-2} \sum_{j_2=j_1+1}^{N-1} 2h(j_1)h(j_2)x_{n-j_1}x_{n-j_2} + a \sum_{j=0}^{N-1} h(j)^2 + b \sum_{j=0}^{N-1} h(j)^2 x_{n-j} \quad (3.52)$$

From this relationship note that this square term contributes primarily to the second order terms in (3.1), but also to the first order term. Also note that the important terms will generally be those for which h_{j_1} and h_{j_2} are both large.

From this one can conclude more generally that large n -th order terms in $q(\cdot)$ will contribute most heavily to n -th order terms in the expansion of (3.1), and that generally the large nonlinear taps will be those containing x_{n-j} corresponding to the larger $h(j)$.

When the D/A is nonlinear, and $d^{-1}(\cdot)$ must be incorporated, it can be expanded in a Taylor series and a similar analysis can be applied to (3.49) to determine which taps in c are important. There are at least two methods for obtaining the Taylor series expansion for $d^{-1}(\cdot)$. One method is to first find an analytical expression for $d^{-1}(\cdot)$, and then expand it in the Taylor series. The second method is to do a Taylor series expansion of $d(\cdot)$, and then use the method described in [4, p. 362] to directly find the Taylor series of its inverse. This latter procedure will be illustrated in Section 3.3.

Note that the validity of the expansion of (3.1) to echo cancellation does not depend on the existence of a Taylor series expansion of $d(\cdot)$ or $d^{-1}(\cdot)$, as can be seen from (3.39). When the function $d(\cdot)$ is not analytic and a Taylor series does not exist (as for example when the function is piecewise linear), then it can be approximated to any desired accuracy by a polynomial (which is a truncated Taylor series) and we can proceed as before.

3.3. CANCELLATION MSE WITH MOS D/A CONVERTER

In this section we study using computer simulations the performance of a nonlinear echo canceller in the configuration of Fig. 3.1b. The operation of the

rapidly decaying echo path impulse response, most of the coefficients of vector \mathbf{c} are negligible and can be ignored. This will be established in Section 3.3 by simulation for typical nonlinearities encountered in MOS D/A converters. However, it is important to develop a methodology by which the non-negligible taps can be predicted, in order to develop insight and to avoid an inordinate number of simulations.

If only L taps are used, it is apparent that the L taps which are largest in absolute value should be chosen. This is confirmed in (3.41), for when $D(\mathbf{c})$ is constrained to have only L non-zero elements, ρ will be minimized by choosing those elements for which g is largest in absolute value. For small deviations from linearity this will be the same as choosing the same L elements of \mathbf{c} to be non-zero.

If the characteristics of the echo channel nonlinearity and D/A nonlinearity are known and fairly reproducible, then the taps which are important can be predicted. This will be illustrated by example. Suppose the echo channel can be modeled by an FIR filter followed by a memoryless nonlinearity $q(\cdot)$. Then (3.23) becomes instead

$$a_n = q\left(\sum_{j=0}^{N-1} h(j)x_{n-j}\right) . \quad (3.49)$$

Then the function $q(\cdot)$ can be expanded in or at least approximated by a Taylor series expansion. Consider for example the square term in this expansion, which becomes

$$\left(\sum_{j=0}^{N-1} h(j)x_{n-j}\right)^2 = \sum_{j_1=0}^{N-1} \sum_{j_2=0}^{N-1} h(j_1)h(j_2)x_{n-j_1}x_{n-j_2} \quad (3.50)$$

which can be simplified by eliminating the duplicated terms and noting from Section 3.1.2 that since x_{n-j}^2 is a binary function it can be represented as

$$x_n^2 = a + bx_n \quad (3.51)$$

for some constants a and b . Then (3.50) becomes

gence analysis of [24], yielding a ratio of asymptotic residual echo to uncancellable signal of

$$\frac{E[e_n - \hat{e}_n]^2}{U} = \frac{\alpha L}{1 - \alpha L} \quad (3.47)$$

$$\approx \alpha L .$$

In (3.47) the number of elements in the augmented transmitted signal vector has been assumed to be L in anticipation of using only a small number L of the 2^N taps. Hence, in general $L \ll 2^N$ will be chosen, and the asymptotic error will be correspondingly smaller than for the table look-up canceller, where $L = 2^N$. Similarly, the speed of convergence can be measured by the time constant of convergence, which is [24]

$$\tau = - \frac{1}{\log_e(1 - 4\alpha + 4\alpha^2 L)} \quad (3.48)$$

$$\approx \frac{1}{4\alpha} .$$

The approximations in (3.47) and (3.48) are valid for practical values of α , which are very small. We can compare the convergence of the linear canceller, the nonlinear canceller proposed here, and the table look-up canceller by setting the asymptotic residual errors of (3.47) equal for the three cases and then comparing the time constants of (3.48). The result is that the time constant of the nonlinear canceller proposed here is $\frac{L}{N}$ times as great as for the linear canceller, while it is $\frac{2^N}{N}$ times as great for the table look-up canceller. Thus, we pay a convergence time penalty for the extra nonlinear taps (about a factor of two for the numerical examples of Section 3.3), but a much larger penalty for the table look-up canceller.

3.2.5. Truncation of the Series Expansion

In the preceding analysis, the full 2^N -tap echo canceller has been considered. We expect that under the conditions of a) small nonlinearity, and b)

usual, the parameter α is adjusted to obtain the desired tradeoff between convergence rate and asymptotic excess mean-square error. This is the same adaptation algorithm used in [24], and it is interesting that the presence of the nonlinearities in the channel has not affected the adaptation algorithm at all aside from the augmentation of the transmitted signal vector with nonlinear taps.

If there is a nonlinearity $d(\cdot)$ in the echo replica path, the gradient of (3.43) becomes

$$\mathbf{grad} \rho = -2 \left[\frac{\partial \mathbf{D}(\mathbf{c})}{\partial \mathbf{c}} \right] (\mathbf{g} - \mathbf{D}[\mathbf{c}]) \quad (3.45)$$

If the Jacobian matrix $\frac{\partial \mathbf{D}(\mathbf{c})}{\partial \mathbf{c}}$ is nonsingular, the unique minimum of ρ is obtained for

$$\mathbf{D}[\mathbf{c}] = \mathbf{g} \quad (3.46)$$

as in (3.38). In this case the gradient technique will also apply, since there will be no local minima in which the algorithm can get lost. The condition of nonsingularity of the matrix $\frac{\partial \mathbf{D}(\mathbf{c})}{\partial \mathbf{c}}$ is a very mild one, since for the case of small nonlinearity (the case of interest here), $\frac{\partial \mathbf{D}(\mathbf{c})}{\partial \mathbf{c}}$ differs from the identity matrix by only a small perturbation, and the small size of the perturbation ensures that the matrix is nonsingular. Further, since the function $\mathbf{D}(\mathbf{c})$ is not known by the adaptation algorithm, it is necessary that $\frac{\partial \mathbf{D}(\mathbf{c})}{\partial \mathbf{c}}$ be replaced by the identity matrix. This is again justified for small perturbations from linearity, and results again in the standard LMS gradient algorithm of (3.44).

The speed of convergence and asymptotic residual echo can be predicted from the analysis of [24]. Although the elements of the vector \mathbf{x}_k are not statistically independent, they are uncorrelated and zero-mean for the case where the data digits x_n assume the values +1 and -1 with equal probability and are statistically independent. This condition is sufficient for the validity of the conver-

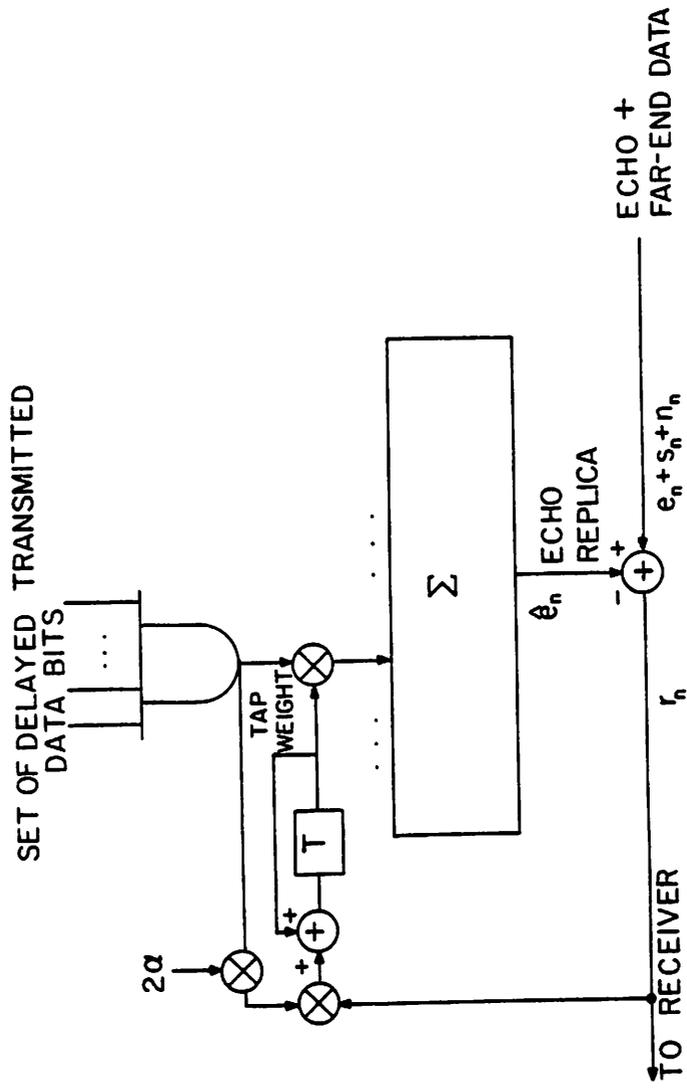


Fig. 3.8. Adaptation of tap weight coefficients is the same as for the linear canceller.

3.2.4. Adaptation Algorithm

In this Section we show how the usual LMS adaptation algorithm can also be used in the presence of nonlinear distortion. From Figure 3.8, the residual signal after echo cancellation is

$$r_n = e_n + s_n - e_n + n_n \quad (3.40)$$

where s_n is the data signal coming from the remote transmitter and n_n a the noise term, both assumed to be uncorrelated with the echo. If the data digits x_n are assumed to be uncorrelated and assume the values $+1$ and -1 with equal probability, it is easily verified that the elements of vector \mathbf{x}_n are uncorrelated (although not independent). Then by an analysis similar to that in [24], the mean-squared residual can be calculated to be

$$\begin{aligned} \rho &= E(r_n^2) \\ &= (\mathbf{g} - \mathbf{D}(\mathbf{c}))^T \cdot (\mathbf{g} - \mathbf{D}(\mathbf{c})) + U \end{aligned} \quad (3.41)$$

where

$$U = E(s_n^2) + E(n_n^2) \quad (3.42)$$

is the total power of the remote data signal and noise.

Assume initially that the canceller does not have a nonlinear D/A, so that $d(\cdot)$ is the identity function. Then ρ of (3.41) is quadratic, and there is a unique global minimum which can be determined by setting the gradient of ρ with respect to the tap vector \mathbf{c} to zero. As in [24] this becomes

$$\begin{aligned} \mathbf{grad} \rho &= -2(\mathbf{g} - \mathbf{c}) \\ &= -2E(r_n \mathbf{x}_n) \end{aligned} \quad (3.43)$$

and the minimum ρ occurs as expected for equal augmented echo path vector and canceller tap vector, $\mathbf{c} = \mathbf{g}$. To find this minimum adaptively, let the canceller tap vector \mathbf{c} be a function of time $\mathbf{c}^{(n)}$ and use the standard gradient algorithm,

$$\mathbf{c}^{(n+1)} = \mathbf{c}^{(n)} + 2\alpha r_n \mathbf{x}_n \quad (3.44)$$

This algorithm is illustrated in Figure 3.8 for just one tap of the canceller. As

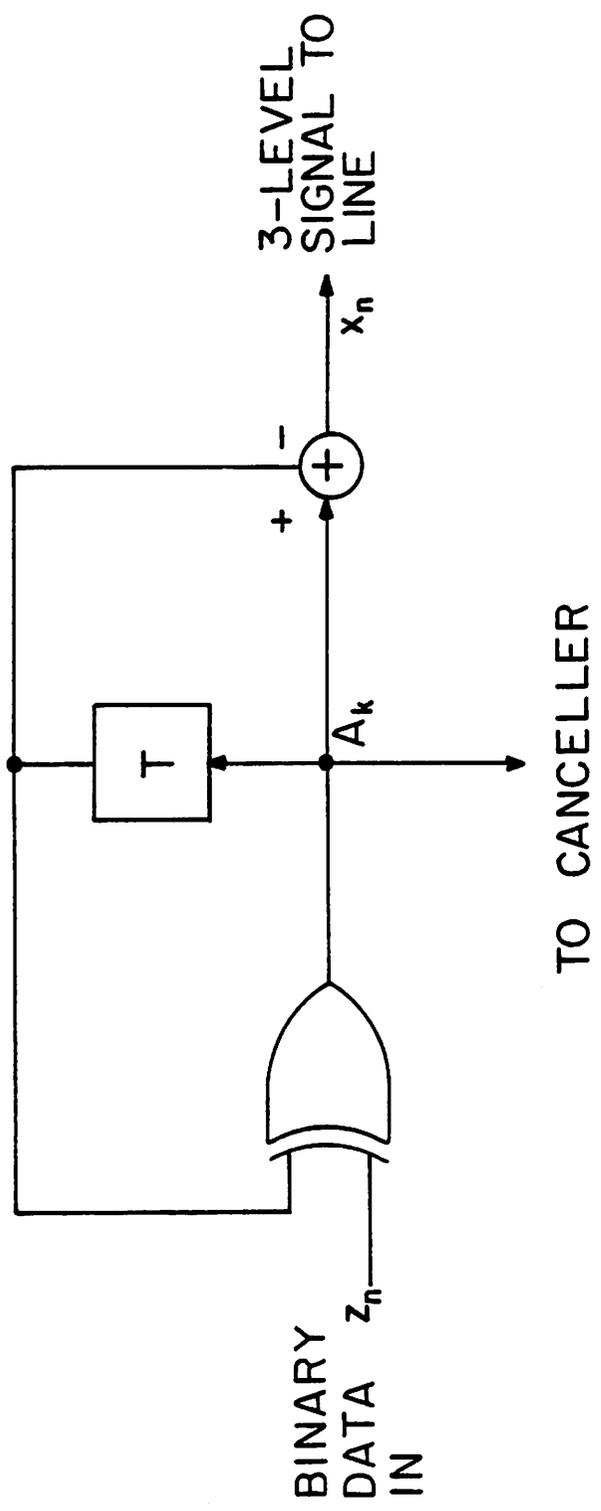


Fig. 3.7. Bipolar encoder.

Section 3.2.1 showed how the three-level x_n could be represented by two bits $z_{1,n}$ and $z_{2,n}$, and the transmitted level could be represented as in (3.26). In the presence of a nonlinear echo channel, the received echo signal of (3.31) can be rewritten as a function of $2N$ bits, and an expansion with 2^{2N} terms results. However, due to the fact that the signal is three-level, and due to the redundancy in the line code, many of these terms are unnecessary. For example, since $z_{1,n-l} = z_{2,n-l} = 1$ can never occur in the transmitted signal, all terms in the expansion containing the product $z_{1,n-l}z_{2,n-l}$, $0 \leq l \leq N-1$ can be eliminated. This will reduce the number of terms in the expansion to 3^N . In addition, the memory in the line code will reduce the number of terms further. For example, since $x_n = 1$ cannot be preceded by $x_{n-1} = 1$, and similarly for -1 , the terms $z_{1,n-1}z_{1,n}$ and $z_{2,n-1}z_{2,n}$ can be eliminated. Elimination of all terms of this type will of course reduce the total number of terms to 2^N , the number of possible input data sequences.

The fact that we end up with 2^N terms by such a cumbersome procedure suggests that there must be an easier approach, and indeed there is. The bipolar encoding can be accomplished by the circuit shown in Figure 3.7 [34]. A modulo 2 accumulation of all the past input bits is first performed, resulting in the binary variable a_n . The three-level x_n is obtained by taking the difference of successive a_n . All we have to do is input to the canceller a_n rather than x_n , since the linear first difference filter can then be thought of as being a part of the echo channel and can be easily compensated by the linear taps of the canceller. When the input data sequence is independent and equally likely, the a_n are likewise independent and equally likely, and the operation of the canceller even when adaptive is not adversely affected. The canceller will require 2^N taps as before, but with a lot less effort!

$$\mathbf{x}_n^T \cdot \mathbf{c} = d^{-1}(\mathbf{x}_n^T \cdot \mathbf{g}) \quad (3.39)$$

Since the right side of (3.39) is a function of N bits, Section 3.1 guarantees the existence of a vector \mathbf{c} satisfying (3.39), and further gives a procedure for finding it. It is interesting to note from (3.39) that even when the echo channel is linear (all but the N linear taps of \mathbf{g} are zero), the canceller needs more than the N linear taps in order to compensate for the D/A nonlinearity.

The addition of extra nonlinear taps should partially or entirely mitigate the effects of D/A nonlinearity, allowing the full resolution of the D/A to be useful. There are monolithic D/A converter realizations which are inherently monotonic, which is sufficient for the existence of $d^{-1}(\cdot)$. Of course, in practice the quantization due to the D/A converter will prevent an exact cancellation of the echo.

The conclusion is that a linear canceller algorithm can still be used in the face of a nonlinear channel and nonlinear canceller implementation. What is necessary is to augment the N bits which are input to the canceller by the remaining bits in the augmented signal vector, resulting in a nonlinear canceller with 2^N taps. Of course, the hope is that considerably fewer taps than this will be required in practice.

3.2.3. Line Codes with Memory

It is often desirable to use line codes which incorporate memory for the purpose of limiting dc wander, RFI, crosstalk, etc. A common example is the "bipolar" or "alternate mark inversion" line code, in which a binary signal is transmitted as a three-level signal. Each input data bit $z_n = 0$ is transmitted as $x_n = 0$, while $z_n = 1$ is transmitted alternately as $x_n = -1$ and $+1$. We will use this example to illustrate how the presence of a line code can be incorporated into the canceller design.

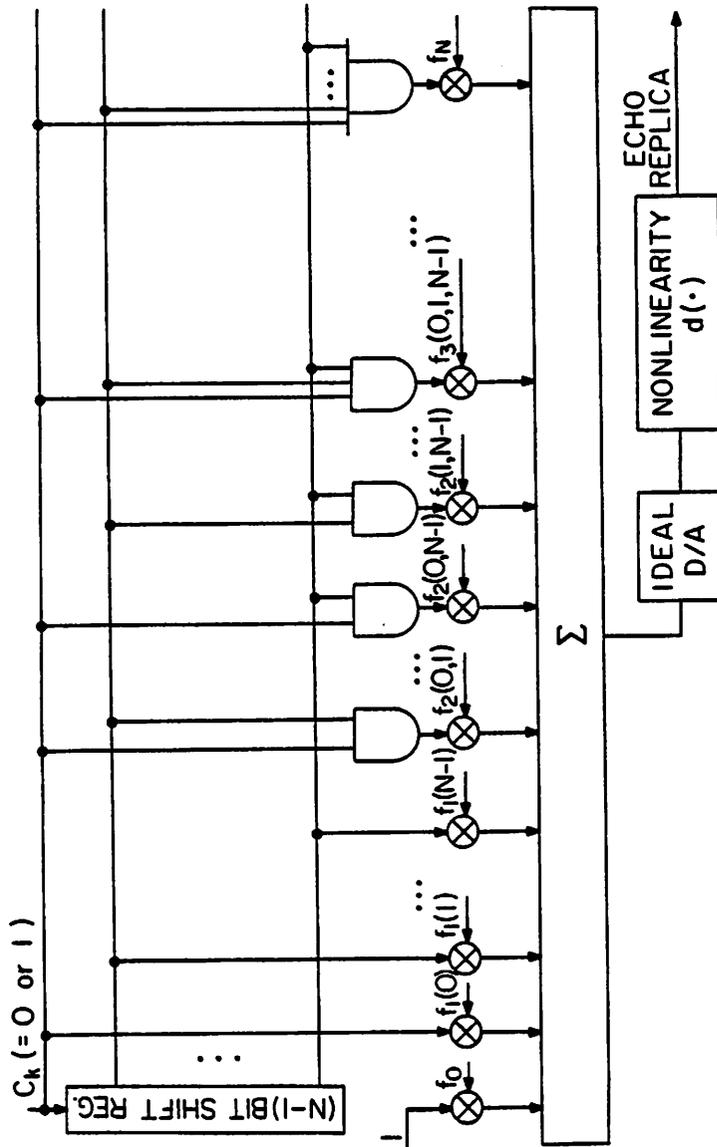


Fig. 3.6. Nonlinear echo canceller for nonlinear channel. Distortion introduced by the channel and by the D/A nonlinearity $d(\cdot)$ are both compensated.

$$\mathbf{g} = (g_0, g_1(0), g_1(1), \dots, g_1(N-1), g_2(0,1), \dots, g_2(N-2, N-1), \dots, g_N)^T \quad (3.33)$$

which is a vector of coefficients of an expansion of the form (3.1) and in accordance with (3.9) is not a function of k . Then the echo is

$$e_n = \mathbf{x}_n^T \cdot \mathbf{g} \quad (3.34)$$

It is natural to assume that the canceller implements an expansion of the form of (3.12) with tap vector \mathbf{c}

$$\mathbf{c} = (c_0, c_1(0), c_1(1), \dots, c_1(N-1), c_2(0,1), \dots, c_2(N-2, N-1), \dots, c_N) \quad (3.35)$$

so that an arbitrary nonlinear echo can be exactly cancelled. A hardware realization of this canceller is shown in Fig. 3.6 Also included in Fig. 3.6 is a nonlinearity $d(\cdot)$ which models the undesired but unavoidable nonlinearity of the D/A converter at the canceller output in Fig. 3.1b. This nonlinearity follows an ideal D/A converter. Ignoring the quantization due to the D/A converter, the echo replica can be written as

$$e_n = d(\mathbf{x}_n^T \cdot \mathbf{c}) \quad (3.36)$$

The interesting question which arises is whether the incorporation of the augmented transmitted data vector into the canceller can compensate for the (D/A) nonlinearity $d(\cdot)$ as well as the echo channel nonlinearity. To answer this question we make use of identity (3.20):

$$d[\mathbf{x}_n^T \cdot \mathbf{c}] = \mathbf{x}_n^T \cdot \mathbf{D}[\mathbf{c}] \quad (3.37)$$

where $\mathbf{D}[\mathbf{c}]$ is a 2^N -dimensional vector-valued nonlinear transformation induced by the nonlinear function $d(\cdot)$ on the coefficient vector \mathbf{c} . Note that this relation is still linear in the augmented transmitted signal vector. As long as a vector \mathbf{c} can be found such that

$$\mathbf{x}_n^T \cdot \mathbf{g} = \mathbf{x}_n^T \cdot \mathbf{D}[\mathbf{c}] = d(\mathbf{x}_n^T \cdot \mathbf{c}) \quad (3.38)$$

for every signal vector \mathbf{x}_n then $e_n = e_n$ and in principle the echo canceller can cancel the echo completely even in the face of the nonlinearities. A simple sufficient condition for (3.16) to be possible is if the inverse D/A nonlinearity $d^{-1}(\cdot)$ exists, since then (3.38) becomes

3.2.2. Nonlinear Channel With Nonlinear Canceller

The most interesting application of the expansion of Section 3.1 is to the compensation of nonlinear as well as linear effects in the channel, as well as in the canceller itself. The method by which this can be done will be considered in this Section.

Assume that the echo signal is not a linear superposition of data digits as in (3.1), but rather that the echo is a general nonlinear function of the current and past $N-1$ data digits,

$$e_n = f(x_n, x_{n-1}, \dots, x_{n-N+1}) \quad (3.31)$$

This model precludes the possibility of the function f being a function of n , and thus requires that the nonlinear echo channel be time-invariant. This is the same assumption required for the Volterra series representation of a nonlinear system [54]. However, when the canceller is made adaptive as in Section 3.2.5, the canceller can compensate for a nonlinear echo channel which is a slowly varying function of time.

Further assume for simplicity that the data digits are binary, assuming one of two values. As was mentioned in Section 3.1.2, the expansion of (3.1) is valid for an arbitrary two-level signal x_n as well as for a signal z_n which assumes the values 0 and 1. It is convenient to write this expansion in a vector inner product notation. In analogy with Section 3.1.3, define a 2^N -dimensional "augmented transmitted data vector"

$$\mathbf{x}_n = (1, x_n, \dots, x_{n-N+1}, \\ x_n x_{n-1}, x_n x_{n-2}, \dots, x_n x_{n-N+2} x_{n-N+1}, \\ \dots, x_n x_{n-1} \dots x_{n-N+1})^T \quad (3.32)$$

The subscript n on \mathbf{x}_n reflects the fact that this vector is changing with time in accordance with the current and last $N-1$ bits of the data sequence.

In a similar way, define a 2^N -dimensional "augmented echo path vector"

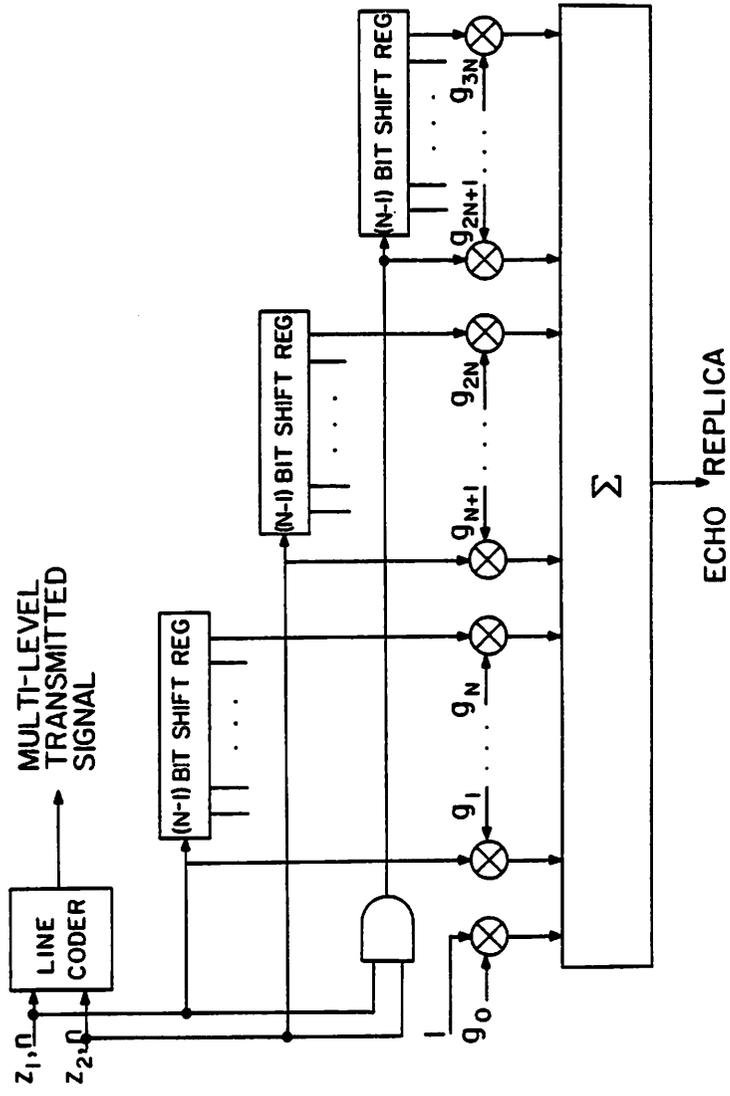


Fig. 3.5. Linear echo canceller for multilevel transmitted signals.

$$e_n = \sum_{j=0}^{N-1} h(j) (a + bz_{1,n-j} + cz_{2,n-j} + dz_{1,n-j}z_{2,n-j}), \quad (3.30)$$

a representation of which is shown in Figure 3.5. The echo response is represented as a transversal filter with $3N+1$ taps, each of which needs to store only a single bit. The delay line can thus be implemented by a shift register as in the binary transmission case, and the tap-weights do not require multiplies. The equivalent echo impulse response (g_0, \dots, g_{3N+1}) is a function of the actual channel impulse response as well as the constants (a, b, c, d) . If an adaptive echo canceller is constructed from the model of Figure 3.5, there is no need to explicitly incorporate these latter constants into the design, since the adaptation mechanism will automatically incorporate them. This should become clearer in Section 3.2.5 where adaptive filtering in the context of the expansion (3.1) will be elaborated.

When the transmission is two-level, then only one of the three shift registers is required, so that there are $N+1$ total taps. For three-level transmission, only two of the shift registers are required, so that there are $2N+1$ total taps.

In general, for M -level transmission, the structure of a multiply-free binary transversal filter can be retained and the details of the multiple level transmitted signal can be left to the adaptation mechanism to sort out. In each case, a maximum of $(M-1)N+1$ taps are required in the binary transversal filter. This technique has two advantages. First, the implementation is simplified by incorporating the details of the multilevel signal into the tap weights. Second, in practice there will be some uncertainty in the transmitted levels due to component tolerances, etc., for which the canceller will automatically compensate. For example, a mismatch between a positive and negative transmitted level will have no adverse effect on the echo attenuation which can be achieved.

corresponding to the 11 bit pattern, which is c_2 , can be set to zero, resulting in an expansion of the form

$$x_n = a + bz_{1,n} + cz_{2,n} \quad (3.26)$$

where there are three constants. Alternatively, if the bit pattern 01 is not assigned, then the $c_1(2)$ coefficient can be set to zero and

$$x_n = a + bz_{1,n} + cz_{1,n}z_{2,n} \quad (3.27)$$

which is of a slightly different form but still has three constants. Similarly, there are two other possibilities for the expansion, corresponding to not assigning the 00 or 10 bit patterns.

When the number of transmitted levels is four, the expansion of (3.1) becomes directly of the form

$$x_n = a + bz_{1,n} + cz_{2,n} + dz_{1,n}z_{2,n} \quad (3.28)$$

for some constants (a, b, c, d) . Finally, for $M=5$, choose $L=3$, and assign the bit patterns 000, 001, 010, 100, and 011 to the five levels. Then the expansion is of the form

$$x_n = a + bz_{1,n} + cz_{2,n} + dz_{3,n} + ez_{2,n}z_{3,n} \quad (3.29)$$

where there are in this case 55 other ways in which the five levels can be assigned to patterns of three bits, each resulting in a different form of the expansion.

It should be emphasized that in any of these illustrative expansions one or more of the constants can be zero. In fact, one criterion for choosing from among the possible expansions is the number of non-zero terms which result for the particular transmitted levels.

Using these expansions, the received echo signal of (3.1) can be represented in a different form, in which the terms are represented in terms of binary rather than M -level data signals. For example, for the four-level signal of (3.6), (3.1) becomes

3.2.1. Multilevel Transmitted Signals With Linear Canceller

It was first pointed out by Mueller [20] that the echo canceller for data transmission is particularly attractive to implement when the transmitted data bits are inputted directly to the canceller, resulting in a "binary transversal filter" in which the delay elements store individual bits rather than analog values and the need for multiplies is eliminated. When the transmitted data symbols are multilevel, as is usually the case for example in voiceband data transmission, then this advantage would seem to be partially negated. For M transmitted levels, the transversal filters require the storage of M values at each stage and multiplies of the tap-weights by one of the transmitted data levels. For certain signal constellations, the latter values can be particularly inconvenient, as for example the square root of two.

In the instance of multilevel data the expansion (3.1) can be used to obtain a simpler implementation. Let L be an integer such that $2^L \geq M$. Then the transmitted level x_n can be represented as a function of L bits,

$$x_n = f(z_{1,n}, z_{2,n}, \dots, z_{L,n}) \quad (3.24)$$

which in turn can be expanded as in (3.1). As shown in Section 3.1.1, at most M terms are required in this expansion.

This result will now be illustrated for $M=2$ through $M=5$. For $M=2$ level transmission, $L=1$ and (3.1) becomes

$$x_n = a + bz_{1,n} \quad (3.25)$$

for some constants a and b . Section 3.1.2 gives a procedure for finding the two constants, but in this case it is not necessary to find them since as will be shown shortly the adaptation mechanism of an adaptive canceller will automatically find the right constants without need for the designer to specify them.

For a three-level transmitted signal, let $L=2$ and assign the bit patterns 00, 01, and 10 to the three levels. Then in the expansion of (3.1) the term

$$D(\mathbf{c}) = \frac{1}{2^N} \sum_{k=1}^{2^N} d(\mathbf{x}_k^T \cdot \mathbf{c}) \mathbf{x}_k \quad (3.22)$$

3.2. APPLICATION TO ECHO CANCELLATION

The usual assumption in the design of an echo canceller for data transmission is that the echo signal consists of a linear superposition of N data symbols,

$$e_n = \sum_{k=0}^{N-1} x_{n-k} g(k) \quad (3.23)$$

where e_n is the current echo signal, x_n is the current transmitted data symbol, and $g(0), \dots, g(N-1)$ are the impulse response samples of the echo channel. In this Section we relax this linearity assumption and show how nonlinearities in the echo channel and in the echo canceller itself can be compensated in the canceller using the series expansion of (3.1). It will be shown that this method is considerably more attractive than the table look-up method [25], particularly when the number of bits N is large and the nonlinearities are mild.

In Section 3.2.1 the application of this expansion to multilevel data transmission will be discussed. Then in Section 3.2.2 the application to a nonlinear channel and/or a canceller which for implementation reasons is nonlinear will be explored. Section 3.2.3 explores the modifications which are desirable when typical line codes are employed. Section 3.2.4 derives an adaptive algorithm which can be used to "learn" the characteristics of the nonlinear channel and the nonlinearity of the canceller itself (this adaptation algorithm turns out to be essentially the same as for a linear canceller). Section 3.2.5 considers the truncation of expansion (3.1) to a relatively small number of terms, and describes a procedure for determining which terms to retain. Finally Section 3.3 will give numerical results based on computer simulations for reasonable channel and canceller models to illustrate the viability of the techniques.

we shall see now.

To compute the coefficients of (3.1) we must solve the system of 2^N linear equations

$$\mathbf{M}\mathbf{c} = \mathbf{f} \quad (3.18)$$

Because of the properties of \mathbf{M} just shown, system (3.18) admits a closed form solution

$$\mathbf{c} = 2^{-N}\mathbf{M}^T\mathbf{f} = 2^{-N}\mathbf{M}\mathbf{f} \quad (3.19)$$

Expression (3.19) allows the direct calculation of the coefficients of the expansion once the values of f for all possible sequences of N bits are known. It is at the same time another proof of the existence of expansion (3.1) since \mathbf{M} , being orthogonal, is always nonsingular, and so system (3.18) can always be solved.

Let $d(\cdot)$ be a nonlinear function of a real variable. Then it is easy to see that

$$d(\mathbf{x}^T \cdot \mathbf{c}) = \mathbf{x}^T \cdot \mathbf{D}(\mathbf{c}) \quad (3.20)$$

where $\mathbf{D}(\mathbf{c})$ is a 2^N dimensional nonlinear transformation induced by $d(\cdot)$ on vector \mathbf{c} . This identity will be useful in Section 3.2.2 to analyze a nonlinear channel with a nonlinear echo canceller.

To prove identity (3.20), it is sufficient to realize that $d(\mathbf{x}^T \cdot \mathbf{c})$ is just another nonlinear function of x_0, x_1, \dots, x_{N-1} , and consequently admits an expansion of the form (3.1), with a coefficient vector $\mathbf{D}(\mathbf{c})$.

We can solve for $\mathbf{D}(\mathbf{c})$ observing that if \mathbf{x}_k is an element of basis \mathbf{B} , then the 2^N dimensional matrix $\mathbf{x}_k \mathbf{x}_k^T$ is an orthogonal projector, and

$$\sum_{k=1}^{2^N} \mathbf{x}_k \mathbf{x}_k^T = 2^N \mathbf{I} \quad (3.21)$$

Writing identity (3.20) with \mathbf{x}_k instead of \mathbf{x} , then multiplying both sides by \mathbf{x}_k , summing for all k , and using (3.21), we get:

$$\begin{aligned}\mathbf{x} \cdot \mathbf{y} &= 1 + x_0 y_0 + x_1 y_1 + \dots + x_{N-1} y_{N-1} + \dots + x_0 x_1 \dots x_{N-1} y_0 y_1 \dots y_{N-1} \\ &= (1 + x_0 y_0)(1 + x_1 y_1) \dots (1 + x_{N-1} y_{N-1})\end{aligned}\quad (3.14)$$

It is clear that this inner product is different from zero only if $x_k = y_k$ for all $k = 0, \dots, N-1$, which occurs only if $\mathbf{x} = \mathbf{y}$. Thus, any two different vectors in \mathbf{B} are orthogonal, and \mathbf{B} is an orthogonal basis of R^{2^N} as we claimed. Also from (3.14), the norm of a basis vector is 2^N .

Now matrix

$$\mathbf{M} = \left[\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_{2^N} \right] \quad (3.15)$$

is orthogonal, and so the following identity holds:

$$\mathbf{M}\mathbf{M}^T = 2^N \mathbf{I} \quad (3.16)$$

where \mathbf{I} is the 2^N dimensional identity matrix. Also note that all elements of \mathbf{M} are either 1 or -1. Orthogonal matrices whose elements are 1 or -1 are called *Hadamard matrices*, and have been used in other fields of signal processing, like image encoding [53].

Furthermore, if the basis vectors are ordered as follows:

$$\begin{aligned}\mathbf{x}_1^T &= (+1, +1, +1, \dots, +1) \\ \mathbf{x}_2^T &= (-1, +1, +1, \dots, +1) \\ \mathbf{x}_3^T &= (+1, -1, +1, \dots, +1) \\ &\dots \\ \mathbf{x}_{N+1}^T &= (+1, +1, +1, \dots, +1) \\ \mathbf{x}_{N+2}^T &= (-1, -1, +1, \dots, +1) \\ &\dots \\ \mathbf{x}_{2N}^T &= (-1, +1, +1, \dots, -1) \\ &\dots \\ \mathbf{x}_{2^N}^T &= (-1, -1, -1, \dots, -1)\end{aligned}\quad (3.17)$$

matrix \mathbf{M} is also symmetric, and so, up to a factor, it is its own inverse. This ordering was used in writing (3.13).

These properties of \mathbf{M} are useful in computing the coefficients of expansion (3.1) when it is written in terms of a binary variable which assumes values 1 and -1, as

properties of expansion (3.1) will be proved.

It is convenient to use a vector inner product notation. Toward this end, define the 2^N -dimensional "augmented binary vector"

$$\mathbf{x} = (1, x_0, \dots, x_{N-1}, x_0 x_1, x_0 x_2, \dots, x_{N-2} x_{N-1}, \dots, x_0 x_1 \dots x_{N-1})^T \quad (3.10)$$

where each term in the series representation of (3.1) is represented and the superscript T denotes transpose.

In a similar way, define a 2^N -dimensional "coefficient vector"

$$\mathbf{c} = (c_0, c_1(0), c_1(1), \dots, c_1(N-1), c_2(0,1), \dots, c_2(N-2, N-1), \dots, c_N)^T \quad (3.11)$$

which is a vector of coefficients of an expansion of the form (3.1). Then a more compact notation for expansion (3.1) is as an inner product of an augmented binary vector with the augmented coefficient vector.

$$f(x_0, x_1, \dots, x_{N-1}) = \mathbf{x}^T \cdot \mathbf{c} \quad (3.12)$$

It is also convenient to define a 2^N -dimensional vector \mathbf{f} whose components are the values of the nonlinear function f for all the 2^N combinations of +1 and -1 values of the variables x_0, \dots, x_{N-1}

$$\mathbf{f} = \begin{pmatrix} f(+1, +1, +1, \dots, +1) \\ f(-1, +1, +1, \dots, +1) \\ f(+1, -1, +1, \dots, +1) \\ f(+1, +1, -1, \dots, +1) \\ \vdots \\ f(+1, +1, +1, \dots, -1) \\ f(-1, -1, +1, \dots, +1) \\ f(-1, +1, -1, \dots, +1) \\ \vdots \\ f(-1, +1, +1, \dots, -1) \\ \vdots \\ f(-1, -1, -1, \dots, +1) \\ \vdots \\ f(-1, -1, -1, \dots, -1) \end{pmatrix} \quad (3.13)$$

When the vectors \mathbf{x} are also formed for all the combinations of values +1 and -1 of the binary variables x_k , a set \mathbf{B} of 2^N vectors is obtained. We are going to show that \mathbf{B} is an orthogonal basis of the 2^N -dimensional vector space R^{2^N} (the space of all 2^N -tuples of real numbers). To prove this, consider any two vectors \mathbf{x} and \mathbf{y} in \mathbf{B} and form their inner product:

terms corresponding to bit patterns for which the function is not specified results in precisely $(2^N - M)$ zero coefficients, leaving a maximum of M non-zero coefficients.

A natural application is to obtaining an expansion such as (3.1) for a function $f(y)$ where y assumes one of M values. Then for N chosen such that $2^N \geq M$, M different N -bit sequences can be assigned to each of the M values of y . An expansion of the form of (3.1) with a maximum of M terms then results. This procedure will be illustrated in Section 3.1 for a multilevel transmission application.

3.1.2. Expansion in Terms of Other Binary Variables

In some applications, it is desirable to obtain an expansion of the form of (3.1) in a set of variables which each assume two values, like z_k , but not the particular values 0 and 1. For example, in data transmission, it is common to transmit levels 1 and -1 rather than 0 and 1. The former values have the advantage, as will be seen later, of having statistical properties which are easier to handle.

Let the variable x_k assume one of two values. Then it follows from (3.1) that

$$x_k = a + bz_k \quad (3.9)$$

where z_k assumes the values 0 and 1 and a and b are some appropriately chosen constants. Then an expansion of the form (3.1) can be obtained in terms of x_k . This can be seen by direct replacement of (3.9) in (3.1) and will be shown in more detail in the next Section for the particular case of a variable which assumes the values 1 and -1 .

3.1.3. Some Important Properties of the Nonlinear Expansion

If the binary variable x_k assumes values 1 and -1 , the coefficients of expansion (3.1) have a simple closed form expression. In order to derive it, some pro-

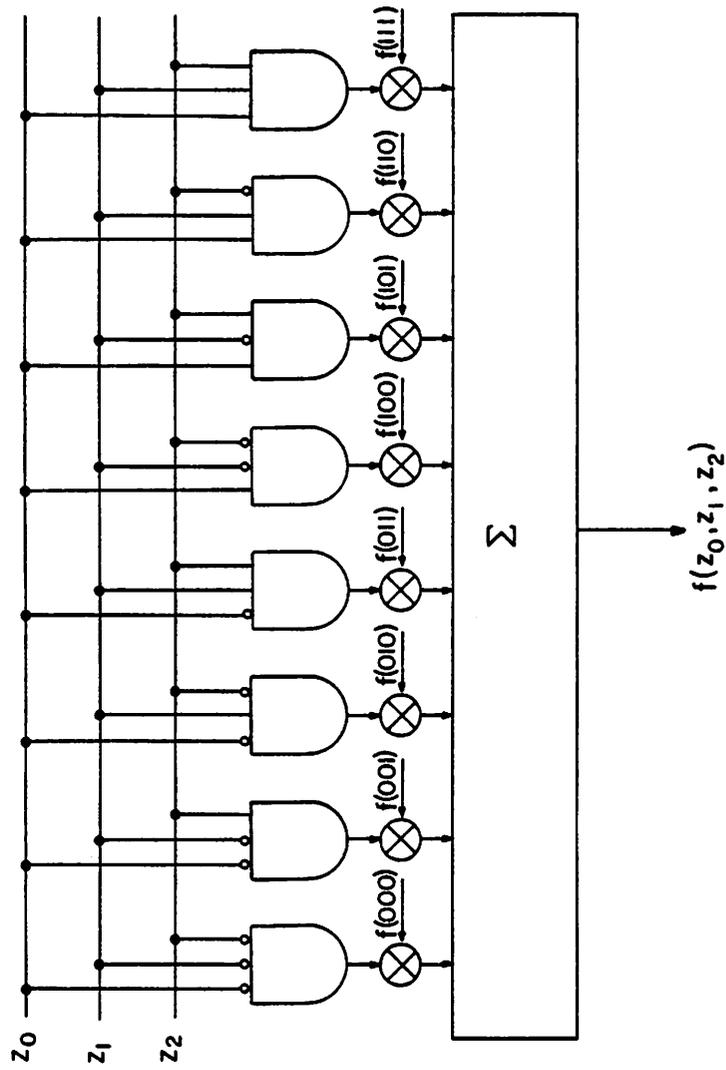


Fig. 3.4b. Hardware implementation of nonlinear function by table lookup method.

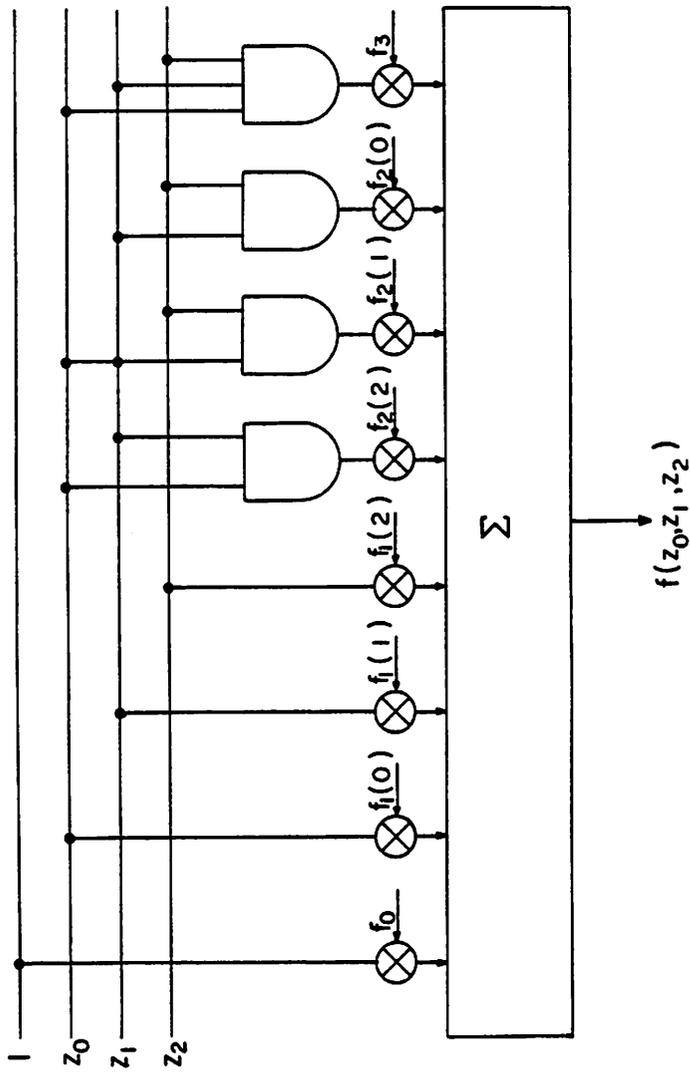


Fig. 3.4a. Hardware implementation of nonlinear function by binary series expansion.

cal situations not all of the terms in the series expansion need be retained. For example, if the function is "linear", then

$$f(z_0, z_1, z_2) = c_1(0)z_0 + c_1(1)z_1 + c_1(2)z_2 \quad (3.8)$$

and only three terms of the expansion of Figure 3.3a need to be retained while all eight terms of the expansion of Figure 3.3b are always required. This is of considerable importance when N is large and the function is linear or nearly linear. This utility will be demonstrated in Sections 2.2 and 2.3 for the application to echo cancellation of data streams.

Representations of these two methods in a form more appropriate for hardware realization are shown in Figure 3.4. In Figure 3.4a, note that the products of bits are easily generated using "and" gates. While the representation of Figure 3.4b for the table look-up method does not make any sense (simply storing the values in a RAM or ROM is more reasonable), this form is conceptually valuable when the adaptive filtering application is considered in the next Section.

3.1.1. Expansion of Incompletely Specified Functions

When the function $f(z_0, \dots, z_{N-1})$ is not specified for some particular N -bit sequences, a corresponding reduction in the number of terms in the expansion of (3.1) can be obtained. An application of this fact is given in Section 3.2.1.

Suppose that the function is specified for $M < 2^N$ values of its arguments. Then no more than M terms in the expansion are required. To see this, the procedure to determine the expansion coefficients can be modified as follows. When the procedure reaches one of the N -bit sequences for which the value of the function is not specified, the value of the expansion is a "don't care" for this particular argument. Therefore the expansion coefficient being determined can be set to any arbitrary value. In particular, a value of zero effectively eliminates one of the terms of the expansion. Setting to zero the coefficients of all the

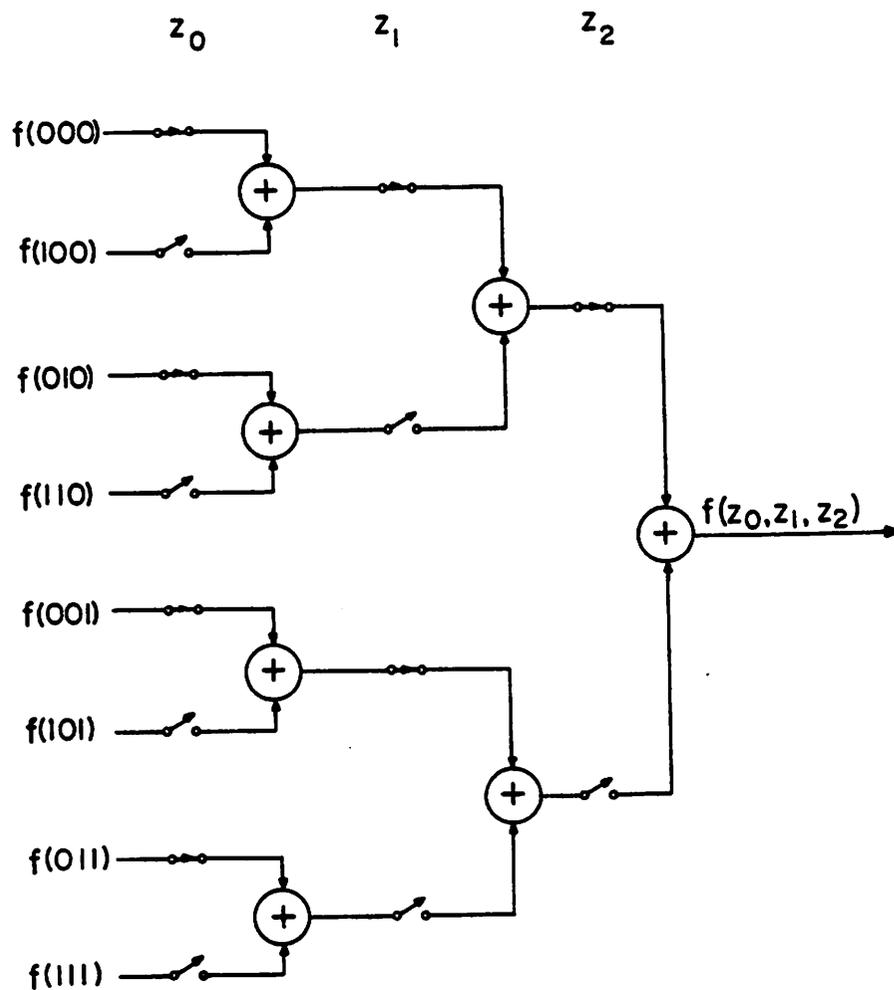


Fig. 3.3b. Binary tree representation of nonlinear function by table lookup method.

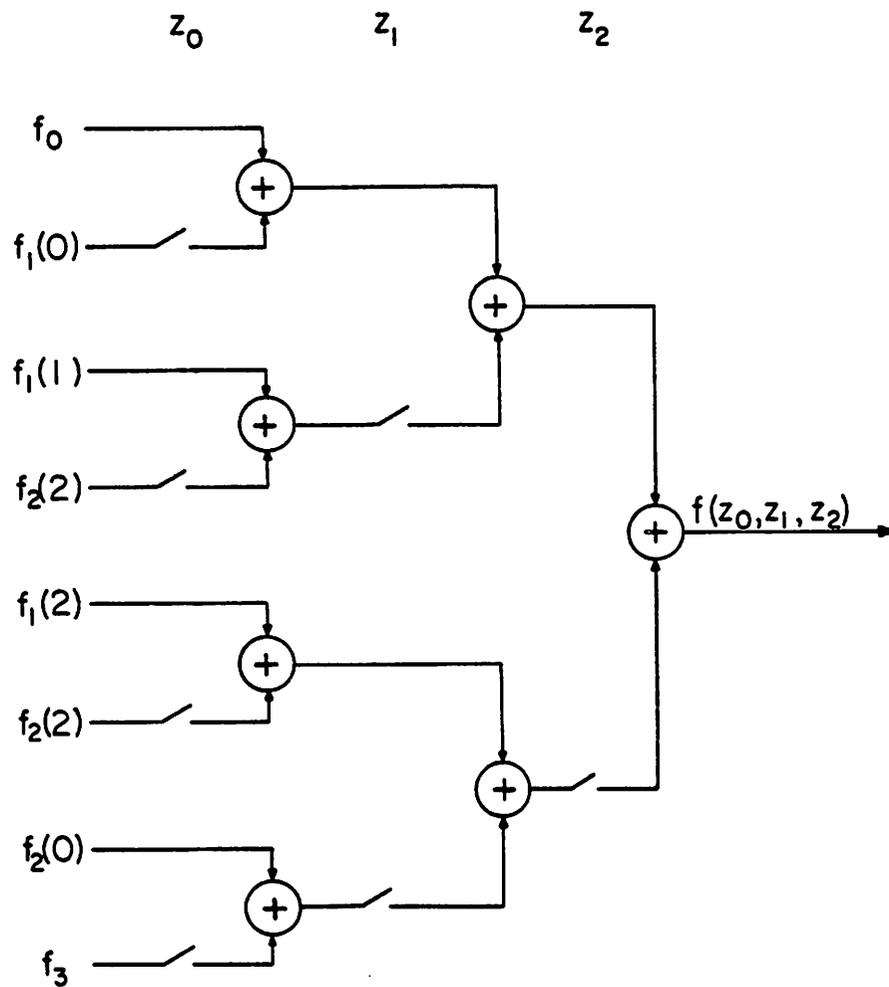


Fig. 3.3a. Binary tree representation of nonlinear function by binary series expansion.

$$f(z_0, z_1, z_2) = c_0 + c_1(0)z_0 + c_1(1)z_1 + c_1(2)z_2 \\ + c_2(2)z_0z_1 + c_2(1)z_0z_2 + c_2(0)z_1z_2 + c_3z_0z_1z_2 \quad (3.6)$$

where there are $2^3 = 8$ terms total. Interestingly, this expansion can be written in the form

$$f(z_0, z_1, z_2) = c_0 + c_1(0)z_0 + z_1(c_1(1) + c_2(2)z_0) \\ + z_2(c_1(2) + c_2(1)z_0) + z_1(c_2(0) + c_3z_0) \quad (3.7)$$

a form which easily generalizes to the general case of N bits. This latter form results in a tree of switches and adders as shown in Figure 3.3a. The leaves of the tree are the values of the constants in the expansion, and the switches closest to the leaves are closed when $z_0 = 1$ and are open when $z_0 = 0$, and similarly for the switches in the other two levels of the tree. Note that in general a number of constants in the expansion contribute to the value of the function, from a minimum of one for the all-zeros case to a maximum of eight for the all-ones case. A number of summations have to be evaluated to determine the function, from a minimum of zero in the all-zeros case to a maximum of seven in the all-ones case.

An alternate representation for the function, also requiring eight constants, is shown in Figure 3.3b. This tree also has three levels (or in general N levels for N bits) but in this case every branch has a switch. The convention is that the switches are shown for the $z=0$ condition, and reverse for the $z=1$ condition. In Figure 3.3b, when the function is evaluated one and only one path through the tree has all the switches closed. Thus, only one of the constants contribute to the function evaluation, and no summations are actually required. This method is of course simply a table look-up, in which the eight functional values are stored.

One might ask what value the expansion of Figure 3.3a has when it requires the storage of eight constants, the same as for the method of Figure 3.3b, but unlike Figure 3.3b also requires summations. The answer is that in many practi-

representation is

$$\sum_{L=0}^N \binom{N}{L} = 2^N. \quad (3.2)$$

Since there are thus 2^N free c parameters in the sum of (3.1), it is not surprising that a function with 2^N values can be represented. When the N bits in (3.1) are taken as N bits out of a continuous bit stream, then the expansion of (3.1) becomes similar to a Volterra series expansion of a general nonlinear time-invariant system [54], with the important exception that only a finite number of terms is required for an exact representation.

The proof that the expansion of (3.1) is general is simple by induction. Note first that

$$c_0 = f(000 \cdots 0), \quad (3.3)$$

the function evaluated for all zeros, since all the higher order terms are zero. Then evaluating the function for a single 1 in the argument at position k , all the second and higher order terms are zero and

$$c_1(k) = f(00 \cdots 010 \cdots 0) - c_0, \quad (3.4)$$

where the single 1 is in position k . Similarly, when the function is evaluated for two ones in positions k_1 and k_2 , all the third and higher order terms are zero and

$$c_2(k_1, k_2) = f(0 \cdots 010 \cdots 010 \cdots 0) - c_1(k_1) - c_1(k_2) - c_0. \quad (3.5)$$

Proceeding by induction, all the constants in the expansion can be evaluated. Not only does this prove the result, but it also elaborates a procedure by which the constants of the expansion can actually be evaluated for a given function of N bits.

Understanding of this expansion can perhaps be enhanced by a simple example. For a function of three bits $f(z_0, z_1, z_2)$ the expansion becomes

simulation results for the types of nonlinearities typically encountered in MOS D/A converters. These results indicate that, depending on the number of bits in the D/A converter, a 20 dB or greater increase in echo attenuation can be obtained by incorporating compensation for the D/A nonlinearity with a modest increase in canceller complexity.

3.1. A BINARY SERIES EXPANSION OF A NONLINEAR FUNCTION

Let $f(z_0, \dots, z_{N-1})$ be an arbitrary (nonlinear) function of N bits, where z_i assumes one of the values 0 or 1. Over all combinations of N bits this function assumes a total of 2^N possible values (which are not necessarily distinct). We now show that this function can be represented as a series with a finite number of terms,

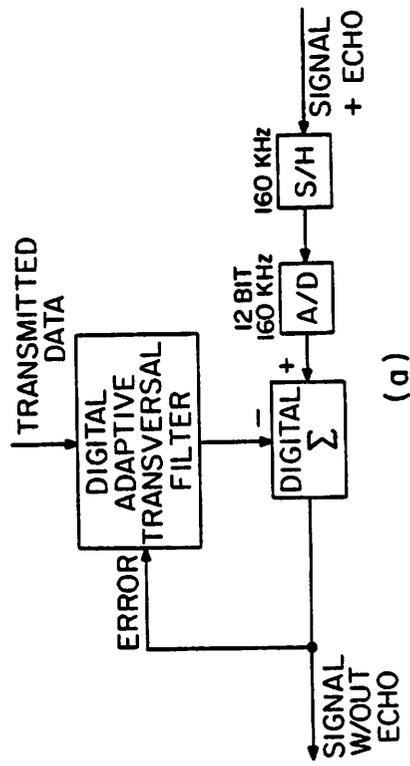
$$\begin{aligned}
 f(z_0, \dots, z_{N-1}) = & c_0 + \sum_{k=0}^{N-1} c_1(k) z_k \\
 & + \sum_{k_1 \neq k_2} c_2(k_1, k_2) z_{k_1} z_{k_2} \\
 & + \dots \\
 & + \sum_{k_1 \neq k_2 \neq \dots \neq k_r} f_r(k_1, k_2, \dots, k_r) z_{k_1} z_{k_2} \dots z_{k_r} \\
 & + \dots \\
 & + \sum_{k=0}^{N-1} c_{N-1}(k) z_0 z_1 \dots z_{k-1} z_{k+1} \dots z_{N-1} \\
 & + c_N z_0 z_1 \dots z_{N-1}.
 \end{aligned} \tag{3.1}$$

The general r -th order sum is over all combinations of r of the N indexes. Thus, for example in the second order term $z_1 z_2$ only appears once, and not as a separate $z_2 z_1$ term. To reduce the number of arguments, the subscripts of the *missing* bits have been used as arguments of the last $\frac{N}{2}$ coefficients $c_r(\cdot)$. The total number of terms can be obtained by observing that the number of combinations of N bits taken L at a time is $\binom{N}{L}$. Thus, the total number of terms in the

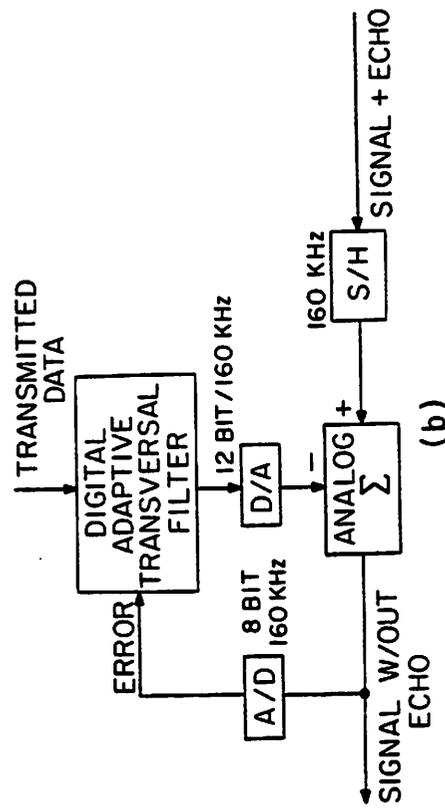
modems. Specifically, the configuration of Fig. 3.1b is particularly attractive, since D/A converters of the required speed and resolution have already been demonstrated in MOS technology [56]. However the linearity requirements can only be achieved using self-correction or trimming, which are costly solutions. An alternative solution to the problem in which the transversal filter summation is done by analog circuitry and thus the adaptation can compensate for the D/A nonlinearity is shown in Fig. 3.2 and will be discussed in the next Chapter. However it cannot correct other sources of distortion like pulse asymmetry or saturation in transformers. Furthermore, digital circuits benefit more from the shrinking design rules and are easier to design than their analog counterparts, and thus a technique amenable to digital implementation like the one presented here is likely to be preferred in the future.

The technique described here is also interesting in other respects. It leads to a systematic design procedure for using a binary transversal filter (where the delays are implemented by shift registers) even in the context of a multilevel transmitted signal and line codes with memory. The design procedure can also take advantage of redundancies in the transmitted line code in the form of simplification of the echo canceller hardware. The technique is also applicable to the implementation of the decision feedback equalizer feedback filter, which has a structure very similar to that of the echo canceller, although the requirements on this filter are relatively relaxed and compensation for nonlinearities may not be required.

In Section 3.1 a method of expanding an arbitrary nonlinear function of a number of bits in a series with a finite number of terms is presented. This expansion serves as the basis for the nonlinear echo canceller design procedures described later. Then in Section 3.2 the application of this expansion to multilevel transmitted signals, redundancies in the line code, and nonlinearities in the echo channel and the canceller itself are considered. Section 3.3 gives



(a)



(b)

Fig. 3.1. (a) Fully digital echo canceller. (b) Digital transversal filter and analog cancellation.

dB or more of echo cancellation, a linear echo canceller would require that the uncompensated transmitted pulse asymmetry be kept below some -60 dB, which can be achieved only with careful circuit design and at the cost of increased complexity.

2. *Saturation in transformers.* This will lead to a slight nonlinearity which can be controlled by choice of a bulkier transformer.
3. *Nonlinearity of data converters.* The echo canceller is typically implemented as a digital processor, since its input consists of an inherently digital bit stream. This suggests that the actual cancellation be done digitally, requiring A/D conversion of the received signal (containing the echo), or the cancellation can be done in the analog domain, requiring D/A conversion of the canceller output. These data converters constitute the most important source of nonlinearity, particularly where monolithic converters without trimming are to be employed.

Examining the alternatives in the use of data converters in greater detail, Fig. 3.1 shows two possible configurations. In Fig. 3.1a, a purely digital echo canceller using a front end A/D is considered. For a digital subscriber loop with a bit rate of 80 kb/s and a minimum of 50 dB of echo cancellation, simulation shows that the A/D needs a resolution of 12 or more bits with 1/2 LSB integral linearity and a conversion time of 6 microseconds or less. In Fig. 3.1b the output of the echo canceller is converted to analog and the cancellation is performed in the analog domain. The error signal which is the input to the digital processor echo canceller is digitized by a lower resolution A/D converter. The required resolution is at least 12 bits with 1/2 LSB integral linearity in the D/A and as many as 8 bits in the A/D (which needs to be monotonic but not necessarily linear). The problem of the linearity of the data converters is a very important one in the context of an MOS monolithic implementation of these

canceller of Section 2.1.2. Another disadvantage is the adaptation speed. Adaptation of a given memory location in the table look-up echo canceller is performed only when that location is accessed, which occurs infrequently because it is associated with one out of 2^N transmitted patterns. In the linear canceller all taps are adapted every bit period, and thus convergence occurs much faster.

The large increase in storage requirement and the decrease in convergence speed of the table look-up echo canceller are a direct consequence of the generality of the nonlinear function of N bits that it can generate. However, this generality is not required in practice, because the kinds of distortion that are likely to be encountered are greatly restricted in their nature, and furthermore, they are generally very small. Thus we are paying a high price in complexity and low speed to achieve more generality than we need. It would be very desirable to find an algorithm able to correct small amounts of nonlinear distortion without a large complexity penalty or adaptation speed penalty. In this Chapter we present an algorithm that is intermediate between the linear and the completely general nonlinear table look-up algorithms mentioned above. It can correct nonlinearity using extra taps in the transversal filter of Fig. 1.6, and the number of extra taps increases in direct relation to the degree of the nonlinearity that it is required to track. As a limiting case for very strong nonlinearity, 2^N taps are required, in which case it becomes equivalent to the table-lookup algorithm [25].

As an example of the kinds of nonlinearity appearing in a practical system, consider the subscriber loop modem shown in Fig. 1.4. The following sources of nonlinear distortion can be identified:

1. *Transmitted pulse asymmetry.* When nominally balanced positive and negative pulses are transmitted, in practice there will be a slight imbalance which cannot be compensated for by a linear echo canceller. To achieve 50

CHAPTER 3

NONLINEAR ECHO CANCELLATION TECHNIQUES

The echo cancellation techniques described in Chapter 2 neglect the effect of nonlinear distortion in the echo path or in the echo replica. However, in real systems small amounts of nonlinear distortion may exist, which result in a degradation of the echo cancellation.

Achieving a 50 to 60 dB cancellation in a monolithic echo canceller is challenging in the face of the inherent nonlinearities in monolithic A/D and D/A converters due to processing variations and component variations. Nevertheless, these systems also have to deal with small amounts of nonlinear distortion in the channel, i.e. a channel which is "almost" linear. Thus some technique must be found to correct the effect of nonlinear distortion and prevent it from degrading the echo cancellation.

A possible solution to the problem is the so called "table look-up echo canceller" [25]. This simply consists of a random access memory that stores the values of the echo for each one of the possible sequences of the latest N transmitted bits. Since there are 2^N such sequences, 2^N memory locations are required. By assigning a possibly different value to each sequence, any function, linear or nonlinear, of the N bits can be represented. Thus the table look-up echo canceller can correct even large amounts of nonlinear distortion, which is a significant advantage over the technique of Chapter 2. Another advantage of this structure is that it requires no additions to compute the echo replica. This is instead simply generated by reading the memory.

A disadvantage is the storage requirement. The table look-up canceller needs 2^N memory words as said before, as compared to the N required by the

jitter in the timing signal. A second order filter with a $Q=100$ has been used. The output of this filter is a nearly sinusoidal waveform with a significant amount of jitter.

2.2.10. Frequency Multiplier

The phase lock loop (PLL) locks to the timing tone obtained in the previous stage and, by using a very large loop time constant ($\tau=0.1\text{sec}$) reduces the jitter. It simultaneously performs a frequency multiplication by a factor of 40, in order to obtain the processor clock at 3.2 MHz (in the second implementation of the echo canceller the processor clock frequency was $f_c=1.6$ Mhz but the same PLL was used in both receivers, adding a divide by 2 stage in the second).

Phase locking occurs after the echo canceller has converged, since an echo-free signal is required to recover the timing. Synchronous operation starts after an initial period during which first the echo canceller is allowed to converge and then the PLL locks. In the subscriber modem, the derived timing clock is also used as the transmit data clock. This synchronous operation is not required for the proper operation of this type of echo canceller [24], but is a requirement of the digital switch with which the modem must interface.

In the central office modem, the transmitter utilizes an external clock that is provided by the switch. The receiver must recover timing from the received signal in much the same way as in the subscriber set. This is so because, although the correct frequency is available from the clock that feeds the transmitter, the correct phase is not, since the phase depends on the delay in the line.

2.2.6. Detector

The output of the echo canceller is the far-end transmitted data signal, which has a three-level eye. Detection requires slicing it at two thresholds and conversion to a binary signal by the inverse of the differentiation introduced in the transmitter. This is done with logic circuits in this implementation. If a decision feedback equalizer were used as proposed later, it could also perform the decoding without any further logic. After decoding, the binary signal is descrambled with the circuit described in Section 2.2.1.

2.2.7. Reconstruction Filter

This filter reconstructs a continuous-time waveform from the samples at the output of the echo canceller to be used in the subsequent timing recovery circuit. Since a nonlinear operation will follow in the timing recovery circuit, and this will create high frequency components not present in the original spectrum, it is necessary to remove first the aliases generated by the sampling. The design of the filter is somewhat simplified using a bandpass instead of a lowpass characteristic. A fourth order maximally flat filter with a center frequency at half the data rate and a bandwidth of 0.4 times the center frequency has been used. It is shown in [27] that use of this filter minimizes the jitter in the timing waveform.

2.2.8. Full-wave Rectifier

Full-wave rectification creates a discrete line [27] in the spectrum at the data rate, even when the data signal has a continuous spectrum bandlimited to less than the data rate (typically between $0.5f_B$ and f_B).

2.2.9. Bandpass Filter

This filter recovers the spectral line created by the full-wave rectification and removes other spectral components whose presence would increase the

$$p_1 = -2.1409 + j 1.1538$$

$$p_2 = -2.1409 - j 1.1538$$

$$p_3 = -1.3130 + j 2.9702$$

$$p_4 = -1.3130 - j 2.9702$$

Similarly the transmitter filter pole locations are:

$$p_1 = -2.1544 + j 2.4755$$

$$p_2 = -2.1544 - j 2.4755$$

$$p_3 = -2.5990$$

2.2.4. Equalizer

A single-parameter manually adjusted equalizer was used in this system. As in the standard automatic line build-out (ALBO) circuit used in T-carrier systems, this equalizer had a single adjustable zero. Performance proved satisfactory under most circumstances. The adjustment of the zero frequency could be made automatic using standard techniques. Use of adaptive decision feedback equalization (DFE) after the echo cancellation would further improve the eye opening, particularly in the presence of bridged taps, but was not included. Eight taps of DFE or fewer should be completely sufficient.

2.2.5. Echo Canceller

Two different versions of the echo canceller have been constructed. They are described in detail in Chapter 4. One of them used a 32 tap transversal filter, the other used only 16 taps. Since the sample rate is twice the data rate, the first filter covered impulse response lengths up to 200 μs , and the second, lengths up to 100 μs . The reason for this is that at the moment in which the first filter was designed, not enough information was available concerning worst case impulse response length of telephone lines. Experimentation proved that 200 μs is an extremely pessimistic estimation, and in the second design only 16 were used.

data rate provides a good indication of the transmitted pulse symmetry. A more quantitative result can be obtained driving the circuit with a sequence of alternating 1's and 0's. A periodic waveform with a fundamental frequency at half the data rate results at the output. The even harmonic content of that signal measures the nonlinear pulse asymmetry distortion.

A significant consideration in the design of the modem is timing recovery. It is possible to derive timing in a half-duplex mode during initialization. However if timing is lost during normal operation, resynchronization requires to interrupt the transmission and go back to a training mode until timing is reacquired. Before this can happen, the loss of synchronization must be detected, perhaps by detecting an unusually high error rate. This technique seems complex and unreliable. Thus, it was decided to take a more conservative design approach in which the canceller is capable of deriving timing properly when the data transmission in the two directions is asynchronous. This allows for proper recovery of timing during startup or after loss of timing, but also requires that the sampling be done at twice the transmitted pulse rate. One of the functions of the receive filter, in addition to removing the out-of-band noise, is thus to remove all potentially aliasing frequency components above the data rate. Aliasing distortion would impair the operation of the timing recovery circuits, although it has no adverse consequences on the echo cancellation and data detection. The receive filter used here provides only 27 dB of attenuation at the data rate. However, the signal has a spectral zero at that frequency, which further decreases the high frequency components. Experimental evaluation has shown that this filter is completely satisfactory. A computer program (given in Appendix A) has been written to solve equations (10) of reference [49]. From use of that program and transient analysis check, the following normalized (for unity data rate) pole locations result for the receiver filter:

The most important limitation is the *integral linearity* of the converter, which is limited by the mismatches in the components used in its monolithic implementation [55,56,57]. In this section we analyze both theoretically and by computer simulation the effect of D/A (or A/D) nonlinearity on the echo canceller configurations of Figs. 4.1b, 4.1c, and 4.1d. In the process we will also be able to characterize the effect of any nonlinearities in the echo response, such as are introduced by transmitted pulse asymmetry. To do this we will make extensive use of the theory of nonlinear echo cancellation that was developed in Chapter 3.

If nonlinear distortion is present in the echo path or in the echo replica path, it is natural to assume that the current echo sample or alternatively the current sample at the canceller output is given by some time-invariant but otherwise arbitrary nonlinear function of the current and last $N-1$ transmitted bits. Let x_n denote the current nominal transmitted level, which for purposes of this Section we will assume takes on the values $+1$ and -1 (other cases can be handled similarly). It was shown in Chapter 3 that an arbitrary time-invariant nonlinear function of the current and the last $N-1$ bits $f(x_n, \dots, x_{n-N+1})$ can be represented by an expansion of the form (3.1), which is analogous to a Volterra series expansion in continuous amplitude signals. This expansion can be used to represent both the echo response and the canceller output in the presence of arbitrary time-invariant nonlinearities. It is also convenient to introduce an augmented transmitted symbol vector, an augmented echo path vector and a coefficient vector as in (3.32), (3.33) and (3.35). With this notation the expansion of (3.1) can be represented as the inner product

$$f(x_n, \dots, x_{n-N+1}) = \mathbf{x}_n^T \mathbf{c} . \quad (4.3)$$

When the data digits x_k are mutually independent and assume the values $+1$ and -1 with equal probability, it is easy to show that the components of the vector \mathbf{x}_n are uncorrelated. This fact will be used when calculating the effect of nonlinearities in the canceller.

This nonlinear expansion can be used to determine the effect of nonlinearities in the echo response. For the general nonlinear case, if it is assumed that the current echo sample is a (nonlinear) function of the current and last $N-1$ data bits, then it can be written in the form

$$\mathbf{e}_n = \mathbf{x}_n^T \cdot \mathbf{g} \quad (4.4)$$

For the case of a linear echo response, only the components $g_1(0), \dots, g_1(N-1)$ of vector \mathbf{g} are nonzero (and equal to the samples of the echo path impulse response).

The representation of the canceller output is different for each of the three cases of Fig. 4.1c, 4.1d, and 4.1b so that they will now be considered separately.

4.3.1. Nonlinear Distortion in Fig. 4.1c

For the configuration of Fig. 4.1c, it is assumed that the D/A converter has an inherent undesired nonlinearity $d(\cdot)$. Since techniques are available to implement D/A converters which are inherently monotonic and have no discontinuities, it is reasonable to assume that the function $d(\cdot)$ is monotonically increasing and continuous. This implies that the inverse function $d^{-1}(\cdot)$ exists. The canceller can then be modeled as a linear combination of the N data bits followed by the nonlinear function $d(\cdot)$,

$$\mathbf{e}_n = d\left(\sum_{k=0}^{N-1} c(k)x_{n-k}\right) \quad (4.5)$$

where the tap-weights in the canceller summation are $c(k)$. Since (4.5) is a general nonlinear function it can be written in the form

$$\mathbf{e}_n = \mathbf{x}_n^T \cdot \mathbf{D}(\mathbf{c}) \quad (4.6)$$

where \mathbf{c} is the N -dimensional tap-weight vector and $\mathbf{D}[\mathbf{c}]$ is a 2^N -dimensional nonlinear transformation of \mathbf{c} induced by the nonlinear function $d(\cdot)$. We thus see that the effect of the D/A nonlinearity is to add the terms of order two and higher in the expansion of the form of (3.1) into the echo replica. Those added

terms are uncorrelated to the echo when the echo response is assumed to be linear, and contribute together with the noise and far-end data to the residual mean squared cancellation error.

The cancellation error is

$$\tau_n = \mathbf{x}_n^T \cdot (\mathbf{g} - \mathbf{D}[\mathbf{c}]) + s_n + \tau_n \quad (4.7)$$

where s_n is the far-end signal and τ_n is the noise. Since all the terms in (4.7) are uncorrelated, including the components of \mathbf{x}_n , the mean squared error (MSE) is

$$\rho = (\mathbf{g} - \mathbf{D}[\mathbf{c}])^T \cdot (\mathbf{g} - \mathbf{D}[\mathbf{c}]) + E[s_n^2] + E[\tau_n^2] \quad (4.8)$$

where E stands for expected value. The condition for minimum MSE is to choose \mathbf{c} so as to minimize the first term in (4.8). Because \mathbf{c} only has N components, in general it is not possible to force this term to zero, even when the echo response is linear, except of course when $d(\cdot)$ is linear. The minimization of the MSE was discussed in Chapter 3, where it was shown that in the case of small nonlinearity the familiar gradient technique applies.

The net effect of nonlinear distortion is to increase the residual error after cancellation, since it causes the first term in (4.7) to be non-zero. Further understanding of this effect can be obtained by grouping the terms in (4.7) in the manner

$$\tau_n = \sum_{k=0}^{N-1} [\mathbf{g} - \mathbf{D}[\mathbf{c}]]_{k+1} x_{n-k} + [\mathbf{g} - \mathbf{D}[\mathbf{c}]]_0 + v_n + s_n + \tau_n \quad (4.9)$$

where the notation $[\cdot]_n$ denotes the n -th component of the vector. The first term in this residual cancellation error is a linear distortion which will be present due to the nonlinearity in the D/A. The minimum MSE is not necessarily achieved when the first term vanishes, since the second and third terms of (4.9) are also functions of \mathbf{c} . The second term is a d.c. offset due to the nonlinearity, while v_n represents all the accumulated nonlinear distortion terms (defined as terms containing a product of two or more data bits). It is perhaps surprising

that all these terms are uncorrelated with one another.

The representations of (4.7) through (4.9) are useful for gaining insight into the nature of the distortion introduced by the D/A nonlinearity. When it is desired to compute the MSE of the cancellation residual, however, it is more convenient to use the simpler relation

$$\rho = E[x_n \cdot g - d(\sum_{k=0}^{N-1} c(k)x_{k-n})]^2 + E[s_n^2] + E[n_n^2] . \quad (4.10)$$

The additional residual echo introduced by D/A nonlinearity is illustrated by the computer simulations of the adaptive canceller shown in Fig. 4.4. A second order nonlinearity bx^2 in the D/A transfer function was considered here. The ratio of the mean squared error (averaged over 1000 samples) to the mean squared echo is plotted for $b=0, 0.01, 0.005$, and 0.001 . In order to achieve 60 dB of cancellation b must be less than 0.001, which requires 12 bit integral linearity in the D/A.

4.3.2. Nonlinear Distortion in Fig. 4.1d

Let $d_k(\cdot)$ be the nonlinear transfer function of the D/A combined with the nonlinear transfer function of the k^{th} S/H in Fig. 4.2. By allowing for nonlinearity and offset in the S/H's, their design can be simplified and chip area can be saved. Moreover, as we shall see later, the adaptation algorithm can easily compensate for these impairments in the structure of Fig. 4.2.

The echo replica in this instance is

$$e_n = \sum_{k=0}^{N-1} d_k(c(k))x_{n-k} \quad (4.11)$$

If we define a new 2^N -dimensional tap-weight vector

$$\mathbf{D}[\mathbf{c}] = (0, d_0(c(0)), d_1(c(1)), \dots, d_{N-1}(c(N-1)), 0, \dots, 0) \quad (4.12)$$

which has only the N non-zero linear components, then the cancellation error and MSE are again given by (4.7) and (4.8) with the new definition of $\mathbf{D}[\mathbf{c}]$. In this

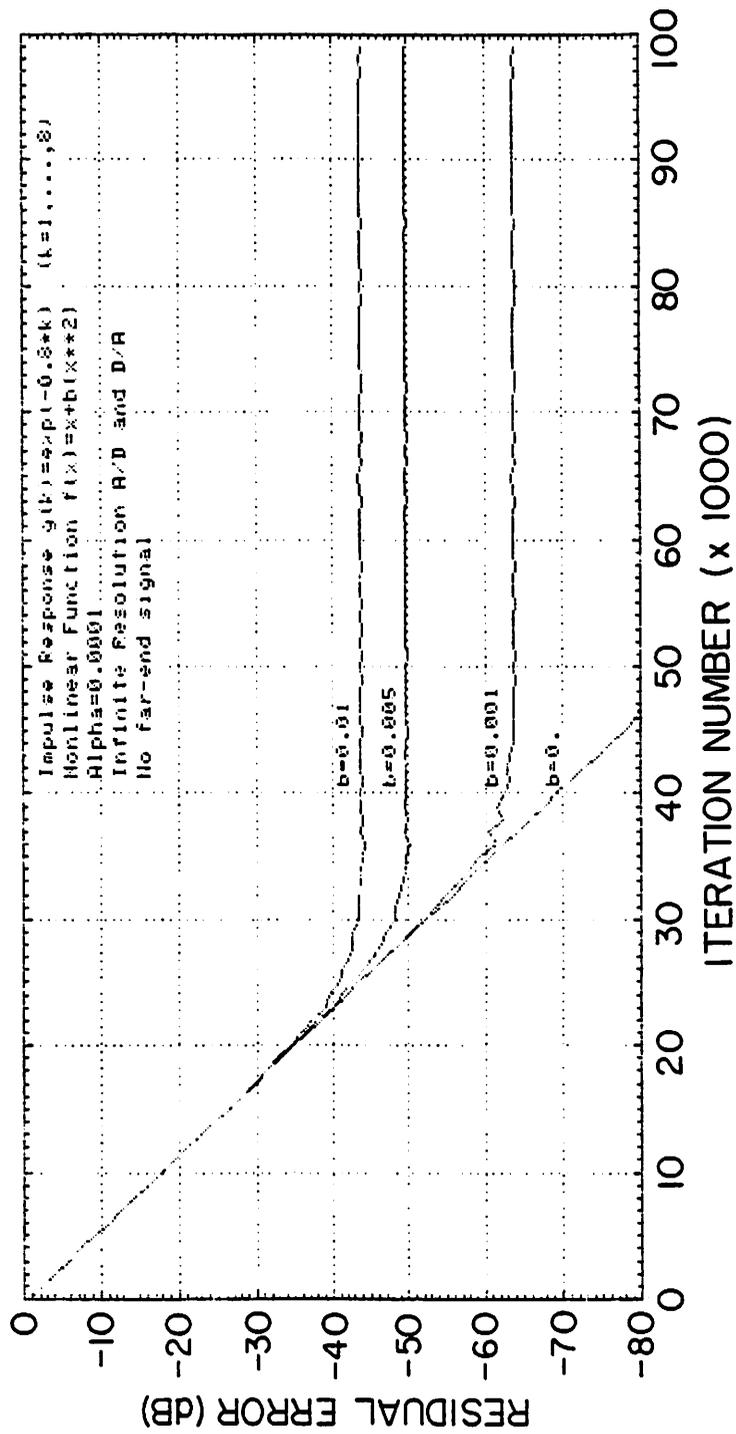


Fig. 4.4. Computer simulations of nonlinear distortion effects in D/A.

instance the condition for minimization of the MSE is much simpler; namely, from (4.7)

$$c(k) = d_k^{-1}(g_1(k)) \quad (4.13)$$

which is valid whether or not the echo response is linear. As before the assumption of the existence of $d_k^{-1}(\cdot)$ is reasonable. While nonlinearity in the echo response does not affect the optimal tap-weight vector in (4.13), it will cause excess MSE in the minimized MSE as seen from (4.8). When the echo response is linear, the first term in (4.8) can be forced to zero, and there is no excess residual echo. This represents a significant advantage of the technique of Fig. 4.1d over 4.1c.

The MSE can be minimized adaptively using the gradient technique. The gradient of ρ is

$$(\nabla\rho)_k = -2(g_1(k) - d_k(c(k))) \cdot \frac{\partial d_k(c(k))}{\partial c(k)} \quad (4.14)$$

Because $d_k(\cdot)$ is monotonic, $\frac{\partial d_k(c(k))}{\partial c(k)}$ is never zero, and hence the only minimum of ρ is associated with the solution (4.13), so that the gradient technique will be able to find the minimum. Furthermore, since for practical situations $\frac{\partial d_k(c(k))}{\partial c(k)}$ is close to unity (because the nonlinear distortion is small), the nonlinearity does not significantly slow convergence. As usual the gradient is replaced by a noisy estimate, and the algorithm becomes

$$c^{(n+1)}(k) = c^{(n)}(k) + 2\alpha\tau_n x_{n-k} \quad (4.15)$$

where a superscript (n) has been added to the tap-weights to indicate that they are now functions of time.

A further modification of the algorithm is introduced by the fact that in Fig. 4.2 the error is computed using the values of the coefficients stored in the S/H, which are not the latest versions since only one S/H is refreshed each pulse period (although the latest versions of *all* the coefficients are always available in

the memory of the digital processor, refreshing only one S/H at a time considerably relaxes the speed requirements of the D/A). Thus in the definition of $D[\mathbf{c}]$ in (3.2.2.2), $c^{(n)}(k)$ must be replaced by $c^{(n-m(k))}(k)$ where $m(k)$ is a cyclic permutation of the sequence $(0, 1, \dots, N-1)$. Which one of the N possible cyclic permutations it is depends on the initialization of the selector circuit, but no attempt is made to control that parameter. Since α is very small, the coefficients change very slowly and this modification does not appreciably alter the convergence.

The preceding analysis has been verified by computer simulation. Fig. 4.5 shows the convergence transient of the echo canceller when no external signal is present and an infinite resolution A/D converter is used. DAC's with 10 to 13 bits resolution and with infinite resolution were considered. It is interesting to observe that the speed of convergence is still experimentally determined to be

$$\nu_{20} = \frac{1.15}{\alpha} \quad (4.16)$$

as predicted in [24], in spite of the modifications introduced in the algorithm. Although one particular case of nonlinearity is shown, we experimented with different nonlinearities and even with independent nonlinear functions for each tap (in anticipation of possibly different S/H nonlinearities), with results very similar to those of Fig. 4.5. We conclude that this technique is extremely insensitive to small nonlinear distortion in the D/A converter. It cannot, however, compensate for nonlinearities in the echo response, as can the technique described in Chapter 3.

4.3.3. Nonlinear Distortion in Fig.4.1b

The effect of nonlinear distortion in the all-digital organization of Fig. 4.1b is somewhat more involved than the previous two cases. That is why it was relegated to the end of this discussion. Since in Fig. 4.1b the nonlinear distortion introduced by the A/D affects the input signal *before* echo cancellation,

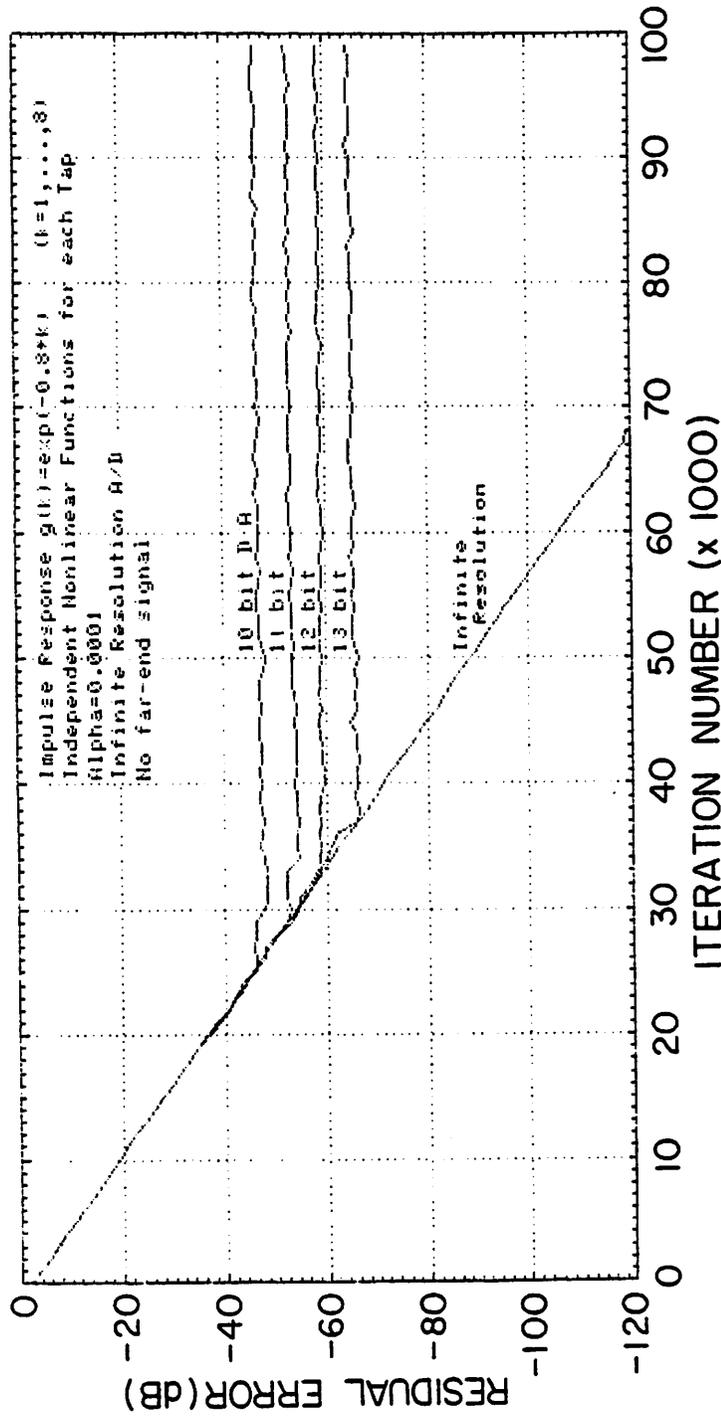


Fig. 4.5. Computer simulation of convergence transient in analog/digital echo canceller.

intermodulation components between the far-end signal and the echo will be created. To see this in detail, let now $\{x_k\}$ be the sequence of transmitted bits and $\{y_k\}$ the sequence of bits transmitted from the far-end of the line. Furthermore, let us assume that the received signal is a nonlinear function of only the last N near-end and M far-end transmitted bits. We can now define an $(N+M)$ -dimensional transmitted symbol vector \mathbf{z}_n , such that

$$\mathbf{z}_n = (x_n, x_{n-1}, \dots, x_{n-N+1}, y_n, y_{n-1}, \dots, y_{n-M+1}) \quad (4.17)$$

and a $2^{(N+M)}$ -dimensional augmented transmitted symbol vector \mathbf{z}_n , defined as in (3.32), but with $(N+M)$ replacing N . An "augmented channel vector" \mathbf{g} can also be defined such that for the most general nonlinear case, the input signal to the receiver is $\mathbf{z}_n^T \mathbf{g}$. If both the transmission and the echo path are linear, \mathbf{g} will have only $(N+M)$ (out of $2^{(N+M)}$) nonzero components, which are the components $g_1(0), \dots, g_1(N+M)$. The first N of them are the samples of the echo path impulse response, and the last M are the samples of the transmission path impulse response.

If a nonlinear A/D with transfer function $d(\cdot)$ is used in Fig. 4.1b, the input to the receiver is:

$$u_n = d(\mathbf{z}_n^T \mathbf{g}) = \mathbf{z}_n^T \mathbf{D}[\mathbf{g}] \quad (4.18)$$

where $\mathbf{D}[\mathbf{g}]$ is a $2^{(N+M)}$ -dimensional nonlinear transformation of \mathbf{g} . In general, all components of $\mathbf{D}[\mathbf{g}]$ may be different from zero, even when the echo response is linear. The terms of (4.18) which are linear functions of the near-end transmitted bits represent a linear echo. The terms which are linear functions of the far-end transmitted bits represent a linear received signal. The nonlinear terms in (4.18) include an offset term and, in addition,

- a) *Nonlinear echo components* consisting of nonlinear interactions among the local transmitted data bits,

- b) *Nonlinear far-end signal components* consisting of nonlinear interactions among the far-end transmitted data bits, and
- c) *Intermodulation terms* consisting of nonlinear interactions among local and far-end transmitted signal bits.

All these nonlinear terms represent an uncancellable noise for a linear echo canceller.

A nonlinear echo canceller as described in Chapter 3 could remove component (a). In order to remove (b) and (c), a generalized transversal filter structure which is a nonlinear version of Mueller's combined echo cancellation and decision feedback equalization [22] could be used. This transversal filter would generate the most general nonlinear function of the last N transmitted and the last M received bits. However such a structure would be impractical because of the requirement for synchronous operation between the near-end and the far-end systems, as discussed in Section 2.1.6.

4.4. EXPERIMENTAL REALIZATIONS

In this Section a detailed description of the four echo canceller configurations introduced in Section 4.1 will be given, including breadboard realizations of alternatives A and B, and an integrated circuit realization of alternative D. Each configuration will be evaluated in terms of the feasibility of its LSI implementation.

4.4.1. Analog Echo Canceller

Fig. 4.6 shows a detailed circuit schematic of an analog echo canceller. This echo canceller has been experimentally evaluated in a breadboard realization, which in addition to the echo cancellation function implemented simultaneous decision feedback equalization as well.

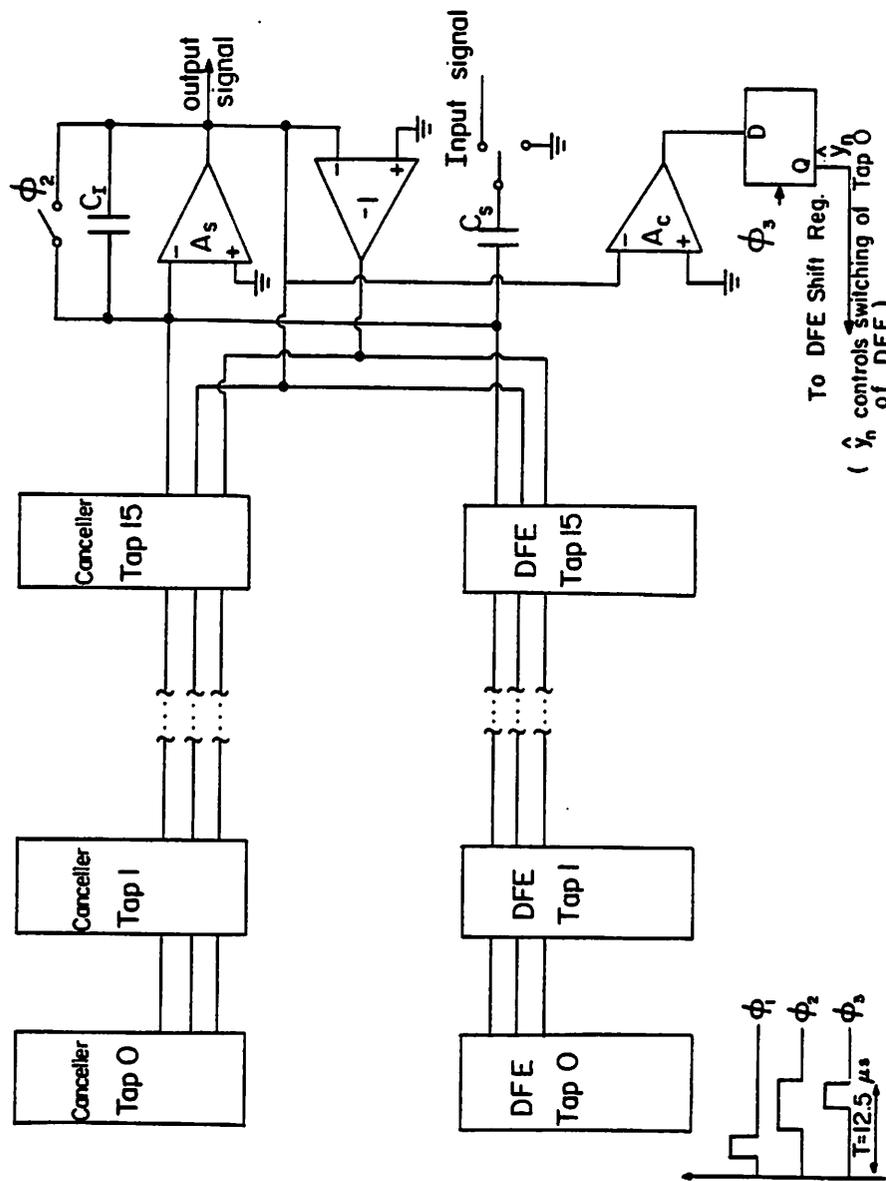


Fig. 4.6a. Circuit schematic of analog echo canceller including decision feedback equalization.

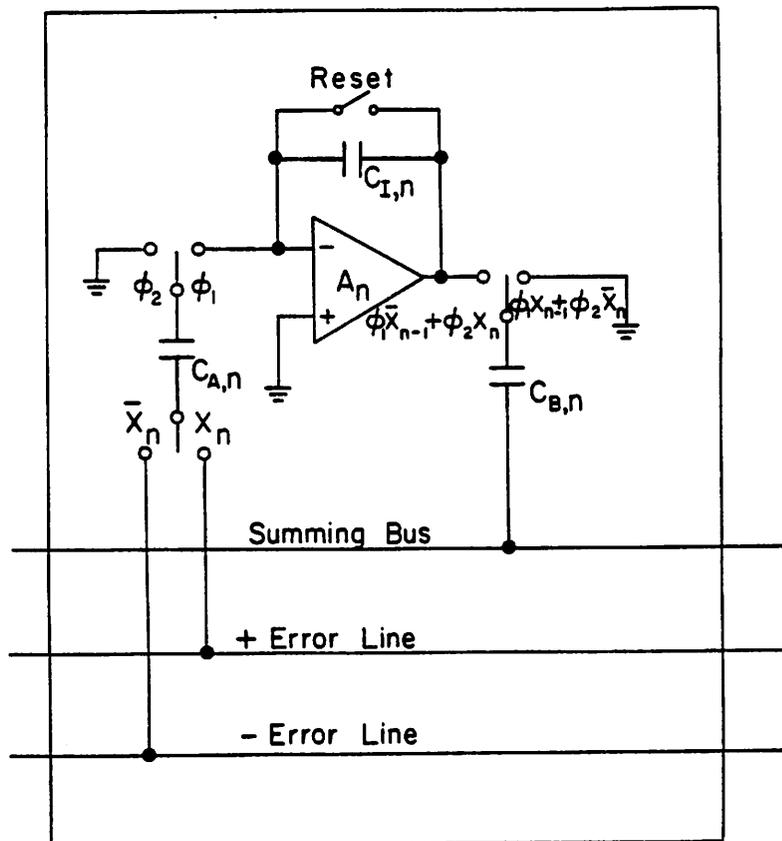


Fig. 4.6b. Detail of an echo canceller tap.

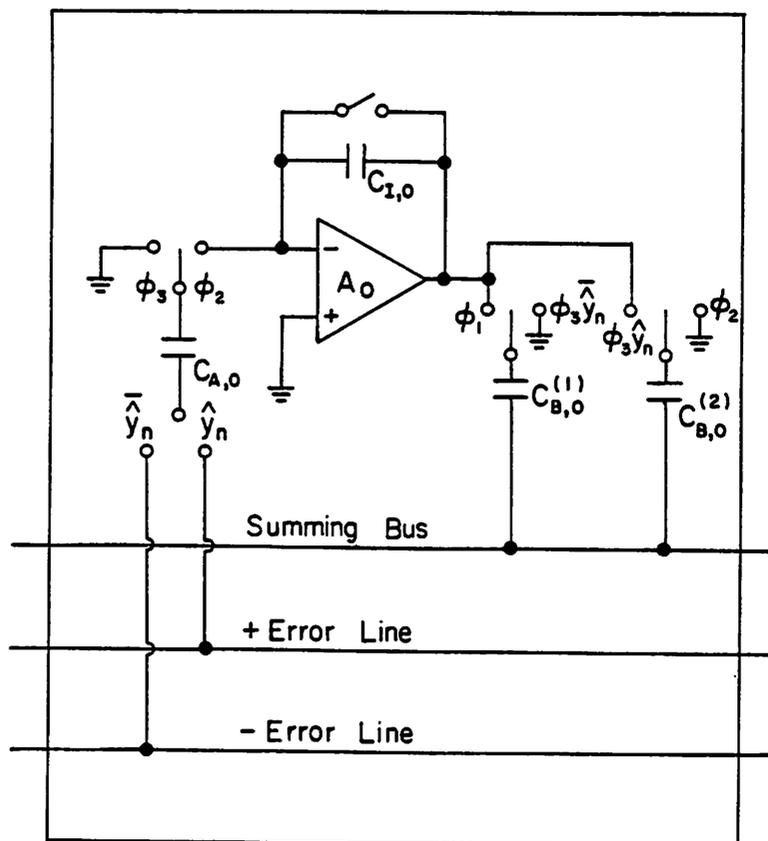


Fig. 4.6c. Detail of a DFE tap.

The coefficients of the echo canceller and the DFE are stored in integrators A_k , while the transmitted data is present in a shift register (not explicitly shown in Fig. 4.6). Weighting of the coefficients by either 1 or -1 (depending on the value of the data bit stored in the shift register stage associated with the coefficient under consideration) is done by summing capacitors $C_{B,n}$. The contributions of the summing capacitors associated with all the taps are added in the summing amplifier A_S . The cancellation error is sampled by capacitors $C_{A,n}$ and used to correct the coefficients according to the LMS algorithm, as described in detail below. The circuit operation takes place in three clock phases Φ_1 , Φ_2 , and Φ_3 , shown at the bottom of Fig. 4.6.

To compute the convolution sum and perform the echo cancellation, capacitors $C_{B,n}$ are switched as follows: During clock phase Φ_1 they are initialized to ground if the data bit stored in the *previous* shift register stage, x_{k-1} , is 0, or to the value of the coefficient present at the output of the integrator if the bit is 1. The reason to use the previous shift register stage in the initialization is that the shift register is shifted by the rising edge of Φ_2 , so that the bit under consideration will not be in the present shift register stage until Φ_2 . The switch position during Φ_2 is controlled by x_n . If $x_n=0$ the switch is returned to ground and the positive value of the coefficient appears at the output of the summing amplifier. If $x_n=1$, the switch is thrown to the output of the summing amplifier and the *negative* value of the coefficient appears at the output of the summing amplifier. The operation of all the other echo canceller taps is similar. Also similar is the operation of the decision feedback equalizer with coefficients $d(n)$ and shift register bits y_n . Also present at the input of the summing amplifier is capacitor C_S , which samples the input signal. C_S is initialized to the input voltage during Φ_1 and switched to ground during Φ_2 , so that it contributes $+s_n$ to the summing amplifier output. Thus, at the end of clock phase Φ_2 , when the summing amplifier has settled, the quantity:

$$s_n - \sum_{k=0}^{N-1} c(k)x_{n-k} - \sum_{k=1}^{M-1} d(k)y_{n-k} \quad (4.19)$$

will be present at its output. Comparator A_C monitors this output to determine its sign, which corresponds to the value of the present data bit. The rising edge of clock Φ_3 loads this present received bit y_n in a flip-flop which is the first stage of the DFE shift register (although this stage is clocked differently from the others, because as we said before, the other stages are clocked by Φ_2 and not by Φ_3). Bit y_n is used to either add or subtract coefficient $d(0)$ of the DFE to the signal present at the output of the summing amplifier. If $c(k) = g(k)$, $k=0, \dots, N-1$ and $d(k) = h(k)$, $k=0, \dots, M-1$, and if no noise is present in the input signal, the output of the summing amplifier should go to zero at the end of the clock phase Φ_3 . This should occur after convergence of the echo canceller and DFE. Prior to convergence, or during convergence, the output of the summing amplifier during Φ_3 represents the *cancellation and equalization error*. This error is used by the adaptation algorithm to correct the coefficients.

Correction of the coefficients is performed as follows:

During Φ_3 , $+\varepsilon_n$ is sampled by capacitor $C_{A,n}$ (Fig. 4.6) if $x_n=1$, or $-\varepsilon_n$ if $x_n=0$. The top plate of the capacitor is grounded during this phase. During the next phase (Φ_1) the top plate of $C_{A,n}$ is connected to the input of the integrator. The bottom plate is not switched in this instance. However, since the summing amplifier is reset during Φ_1 , the bottom plate of $C_{A,n}$ is effectively grounded and the correction term $\frac{C_{A,n}}{C_{I,n}} \varepsilon_n \text{sign}(x_n)$ is added to the previous value of the coefficient stored in the integrator.

It is interesting to analyze in detail which are the critical elements in this circuit. The following points should be discussed:

- a) *Capacitor matching.*
- b) *Offset, low gain and nonlinearity of the summing amplifier.*
- c) *Offset of the integrators, clock feedthrough of the switches, and leakage currents of the integrators (and associated switches).*

These points will be considered in detail now.

4.4.1.1. Capacitor Matching

The ratio of capacitors $C_{B,n}$ to C_I determines the gain factor by which the convolution sum will be affected, and contributes to the overall adaptation gain α . The nominal value of this gain was 1 in our design, so that capacitors $C_{B,n}$ were nominally equal to C_I . Small mismatches in the ratio $\frac{C_{B,n}}{C_I}$ are of no consequence to the circuit performance because the adaptation algorithm will force coefficient $c(n)$ to a value slightly larger than the nominal to compensate for a ratio $\frac{C_{B,n}}{C_I} < 1$, or to a value slightly smaller to compensate for a ratio $\frac{C_{B,n}}{C_I} > 1$.

The only situation in which a mismatch in $\frac{C_{B,n}}{C_I}$ can be harmful is if it becomes much smaller than 1, forcing $C(n)$ to a large value that may saturate the amplifier. Of course this will not happen for small mismatches. A mismatch in $\frac{C_{B,n}}{C_I}$ also affects the gain of the adaptation algorithm. This effect can be lumped together with the effect of mismatches in $\frac{C_{A,n}}{C_{I,n}}$.

The gain of the adaptation algorithm is given by:

$$\alpha_n = \frac{1}{2} \frac{C_{A,n}}{C_{I,n}} \frac{C_{B,n}}{C_I} \quad (4.20)$$

The LMS algorithm (Eq. 2.21) is implemented only if $\alpha_n = \text{const} = \alpha$. If as a result of capacitor mismatches a different value of α is obtained for each tap, the algorithm becomes:

$$c_k^{(n+1)} = c_k^{(n)} + 2 \alpha \varepsilon_n x_{n-k} \quad (4.21)$$

Large α_k mismatches may affect the convergence speed and even render the algorithm nonconvergent. However this happens only for capacitor mismatches which are orders of magnitude larger than those typically encountered in monolithic or discrete realizations of the circuit. We found experimentally that even mismatches of 20 % have negligible effect in the adaptation.

As a conclusion, capacitor mismatch is not a significant impairment for the performance of an analog echo canceller.

4.4.1.2. Offset, Low Gain and Nonlinearity of the Summing Amplifier

Let

$$y = A(x) \quad (4.22)$$

be the transfer function of the summing amplifier. By letting $A(x)$ be a nonlinear function, all the effects mentioned in the title of this Section can be accounted for. This analysis is done in Section 4.4.4, in the context of another echo canceller implementation, but is equally applicable here. The conclusion is that distortion, low gain and offset in the summing amplifier have negligible effect on the performance of the echo canceller.

4.4.1.3. Offset of the Integrators, Clock Feedthrough and Leakage Currents

These, particularly clock feedthrough, are the major limitations of the analog approach to echo cancellation. To understand the effect of the offset and clock feedthrough, consider the integrator of Fig. 4.7, which is a simplified model for a tap of the echo canceller of Fig. 4.6. Analysis is simplified by grounding the bottom plate of capacitor $C_{A,n}$, which amounts to setting the error line of Fig. 4.6 to zero. In practice this voltage is not zero, but it does not interact with the offset and clock feedthrough under consideration, and so it can be neglected here. From Fig. 4.7, clock feedthrough results in the injection of a charge

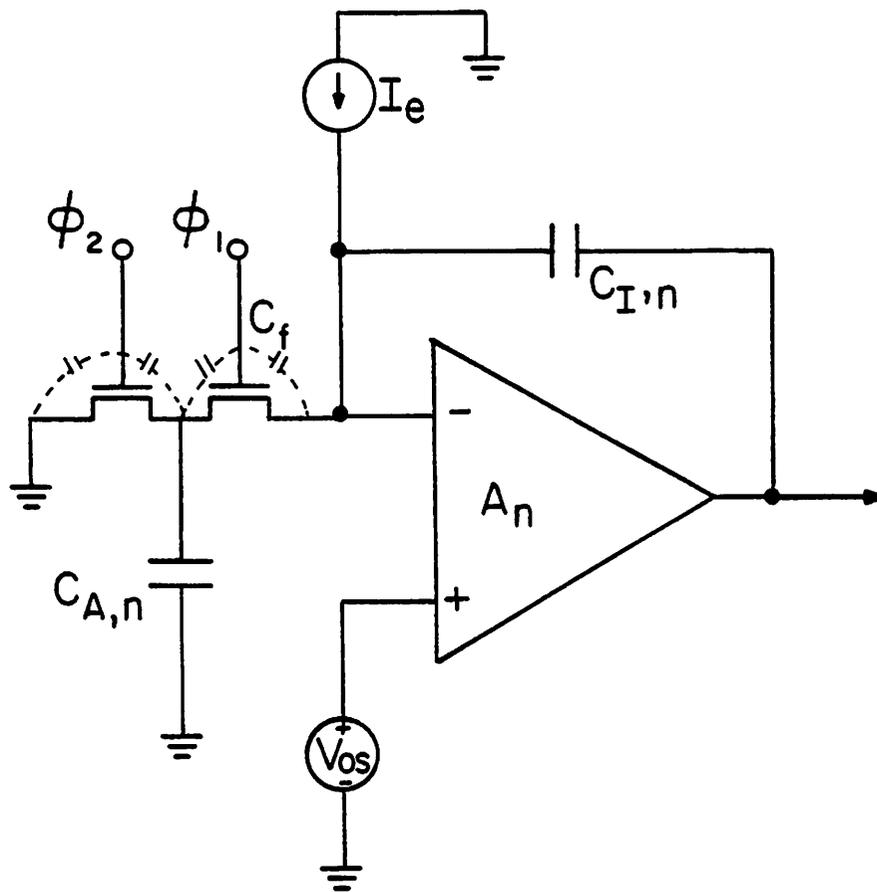


Fig. 4.7. Model of an integrator to analyze clock feedthrough.

$$q_f = V_{clock} C_f \quad (4.23)$$

during each clock cycle, where V_{clock} is the clock voltage swing and C_f is the feedthrough capacitance of the switches.

Offset results in the injection of a charge

$$q_{os} = V_{os} C_{A,n} \quad (4.24)$$

where V_{os} is the offset voltage of the integrator.

Similarly, leakage currents result in a charge

$$q_l = I_l T \quad (4.25)$$

being integrated, where T is the baud period.

The total parasitic charge injected per clock cycle is

$$q = V_{clock} C_f + V_{os} C_{A,n} + I_l T \quad (4.26)$$

and the error voltage added per cycle to the coefficients is:

$$p = \frac{q}{C_{I,n}} \quad (4.27)$$

One is interested in implementing low values of α , which for this implementation is given by

$$\alpha = \frac{C_{A,n}}{C_{I,n}} \quad (4.28)$$

α can be decreased increasing $C_{I,n}$ or decreasing $C_{A,n}$. However, area and settling time considerations limit the maximum size of $C_{I,n}$ to values on the order of 10 pF in an integrated realization of this structure, particularly considering that many integrators must be implemented. Thus, the other possible approach, decreasing $C_{A,n}$ must be studied.

The offset term decreases with decreasing size of $C_{A,n}$. However, the clock feedthrough depends on the ratio of the feedthrough capacitance to the integrator capacitor, rather than the ratio of the sampling to the integrator capacitor, and so it does not decrease with decreasing α . The same can be said of the leakage current term, but this is much smaller than feedthrough, and can usually be

neglected.

Let us consider now what the effect of this parasitic charge is. To this end, let us introduce a vector

$$\mathbf{p} = (p_1, p_2, \dots, p_N)^T \quad (4.29)$$

whose components are the parasitic voltages of (4.26), and the possibility of different values of them for different integrators has been accounted for.

The adaptation algorithm becomes

$$\mathbf{c}^{(n+1)} = \mathbf{c}^{(n)} + 2\alpha \mathbf{x}_n^T \mathbf{r}_n + \mathbf{p} \quad (4.30)$$

A derivation similar to that leading to equation (2.23) can be carried out. The resulting difference equation for the error is

$$\varepsilon_n = (1 - 4\alpha + 4\alpha^2 N) \varepsilon_{n-1} + 4\alpha^2 N U + P \quad (4.31)$$

where $P = |\mathbf{p}|^2$. From here, the steady state MSE becomes:

$$\varepsilon_{\infty} = \frac{\alpha N U + \frac{P}{4\alpha}}{1 - \alpha N} \quad (4.32)$$

which can be compared to (2.26). The effect of the feedthrough is to introduce a term $\frac{P}{4\alpha}$ in the MSE, which *increases* for decreasing α . This term can be physically interpreted as follows: without the feedback introduced by the adaptation algorithm, the feedthrough would be integrated until the integrators saturate. The adaptation feedback prevents the integrators from saturating by forcing the error to a value different from zero, in order to counter the effect of the feedthrough. However, the smaller α (physically, the smaller capacitor $C_{A,n}$), the larger this error has to be to balance the effect of the feedthrough, which is independent of α . Thus, feedthrough becomes more and more troublesome for smaller values of α . The term $\frac{P}{4\alpha}$ is relatively large for all practical values of α and clock feedthrough. Although techniques like feedthrough cancellation using a cancelling switch could be applied, they do not seem to be reliable enough, or to provide sufficient cancellation, and so integration of completely analog echo

cancellers is not feasible. However, in Section (4.4.4) we will see that analog circuits also have important advantages in some respects, and so a hybrid implementation looks very attractive.

4.4.2. Digital Echo Canceller

This was also experimentally tested in a breadboard implementation. Because of the high speed and accuracy requirements of the input A/D, this configuration is not considered appropriate for integration. However, a breadboard implementing it was built with the purpose of doing system-level field measurements. The reason to choose this architecture was that, being completely digital, it offers more reliability and a less risky design. Since the purpose of this breadboard was not to evaluate the circuit, but rather the system, the choice was considered appropriate. A rather detailed description of the digital processor will be given here, because it is considered that it can serve as a basis for new designs, and also because it illustrates the hardware complexity involved.

The echo canceller was implemented as two identical 16 tap transversal filters operating in time-interleaved fashion (Section 2.1.2.1). Each transversal filter was a digital processor built using six 2901 bit slice microprocessors. The internal arithmetic precision was 24 bits. Pipelined microcontrol was used, anticipating the need to perform tests at higher data rates, which would require faster processor operation.

Fig. 4.8 shows a block diagram of the processor. This block diagram does not have a direct correspondence with the actual partition in chips, but is rather designed to explain the way the processor operates. A detailed chip-level schematic is given in Appendix D. The state diagram, timing diagram and microprogram tables are also shown in Appendix D.

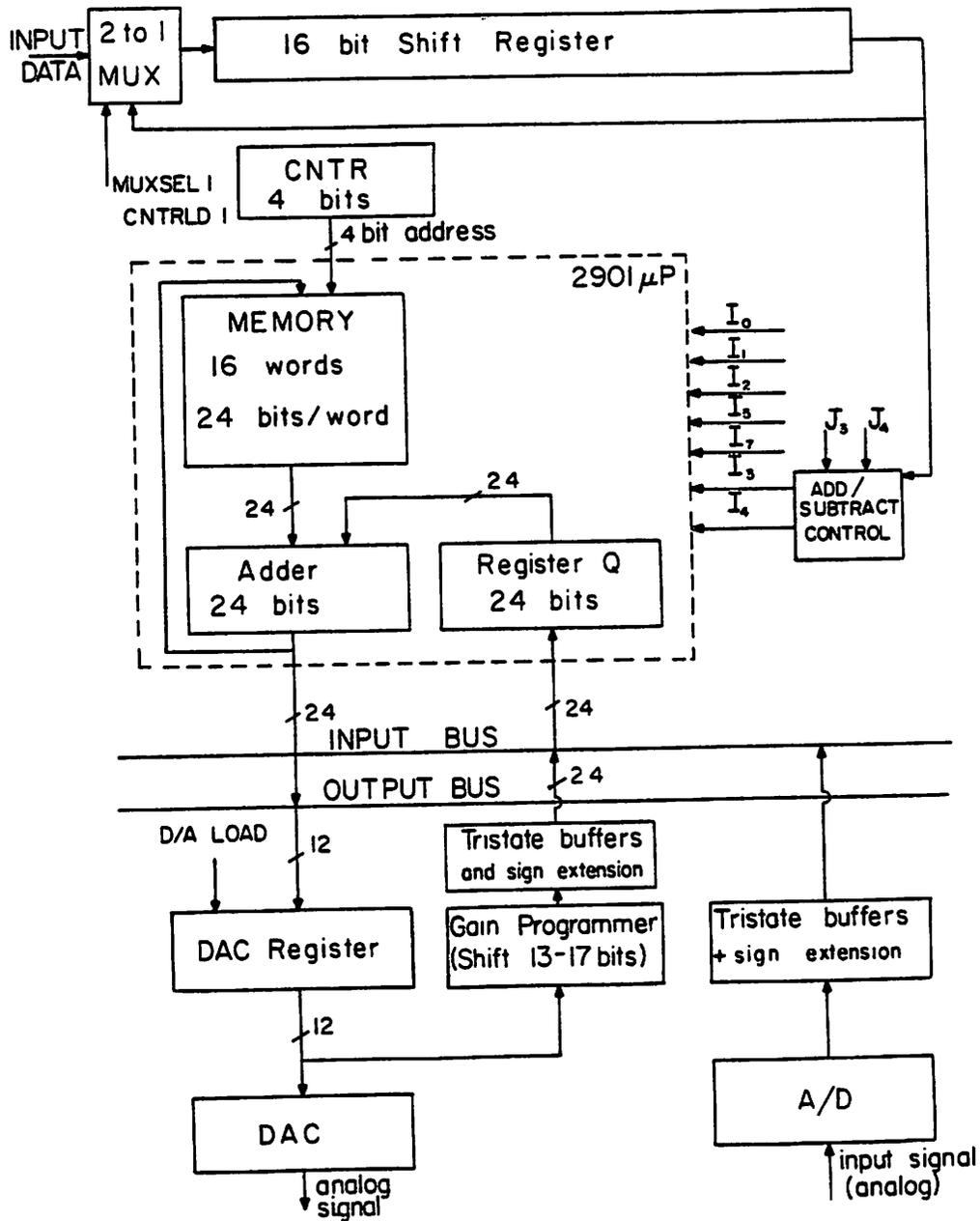


Fig. 4.8a. Block diagram of data path of digital processor.

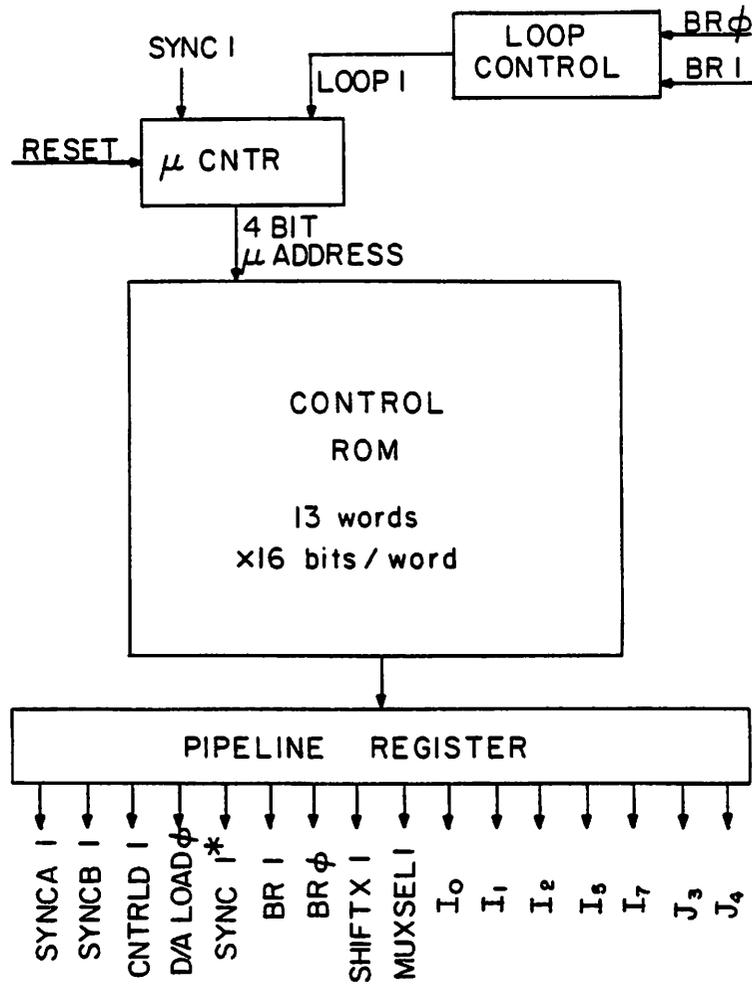


Fig. 4.8b. Block diagram of control unit of digital processor.

The memory (internal to the 2901), supports *read-modify-write* cycles. It consists of 16 words of 24 bits each, representing the echo canceller coefficients. The adder is also internal to the 2901 and is used to perform both the convolution sums and the coefficient adaptation. For maximum speed, external 2902 carry lookahead units were used. The microprogram ROM uses 14 control words of 16 bits each (this gives an idea of the simplicity of the algorithm) to generate control functions for the 2901 and other blocks. The data shift register (16 bits long) holds the binary signal used in the convolution and acts as the transversal filter delay line. It is cyclically rotated 32 bit positions during each data bit period ($12.5 \mu s$), the first 16 times to compute the convolution sum, and the next 16 to perform the adaptation. An additional shift is performed to read an input bit through multiplexer M_1 , discarding at the same time the oldest bit stored in the shift register. The *add/subtract* lines of the 2901 are controlled by the last bit of the shift register, through some additional logic which also depends on the control signals issued by the control ROM.

The microprogram can loop through certain states a number of times defined by the control signals $BR0$ and $BR1$. The following code was used

$BR0 = 0$	$BR1 = 0$	<i>do not loop</i>
$BR0 = 0$	$BR1 = 1$	<i>loop 8 times</i>
$BR0 = 1$	$BR1 = 0$	<i>loop 16 times</i>
$BR0 = 1$	$BR1 = 1$	<i>do not loop</i>

These two control signals, applied to the select control of multiplexer M_2 determine what input to the multiplexer is selected to generate the LOOP1 signal. LOOP1, applied to the enable input of the microprogram counter, ($\mu CNTR$), determines whether a new microinstruction is fetched (LOOP1=0) or the same one is executed again (LOOP1=1). The data inputs to multiplexer M_2 come from outputs 5 and 14 of decoder D_1 . When the address counter (CNTR) reaches a prespecified count, the loop is exited.

The μCNTR is reset to zero by the RESET signal, and to 2 by the SYNC signal, which is used for internal or external synchronization. As said before, the echo canceller uses two identical time interleaved processors. To ensure the correct phase relationship between them, one of the processors works as a slave of the other (the master). The master generates a signal SYNC1 which is used to reset the μCNTR of the slave and thus synchronize its operation. The master, in turn, is synchronized by a signal coming from the timing recovery circuit.

The output of the echo canceller is loaded into the D/A register. The processor receives inputs from the A/D and from the D/A register. This latter is used to read back the output signal (which is the residual signal r_n of equation (2.15)), shifted by a number of bits that can be programmed in the block "programmer", depending on the desired value of α in the adaptation algorithm. Programming is done simply by substituting pre-wired plug-in headers.

The state diagram (Appendix D) shows the operation of the processor. A reset cycle is entered each time the RESET signal is asserted, forcing the μCNTR to zero. The reset cycle clears all the memory locations and register Q. Normal operation starts with state 2. This cycle includes computation of the echo replica, adaptation of the coefficients, and generation of the synchronization signals and the control of the A/D, S/H and D/A.

The timing diagram shows clearly the relationship between the operation of the two processors and between the processors and the sampling of the external signal by the S/H, and the analog to digital and digital to analog conversions.

The microprogram table is compiled from the state diagram, taking into account that there is a delay of one clock cycle between the instruction fetching and its execution, due to the pipelined microcode, which must be considered when branchings are performed.

4.4.3. Digital Echo Canceller with Analog Cancellation

A monolithic implementation of an echo canceller using this approach looks promising in the context of the nonlinear echo cancellation technique presented in Chapter 3, or perhaps using self-calibration techniques to correct the nonlinear distortion of the D/A converter. However none of these techniques has been experimentally evaluated as of this writing, although experiments are underway.

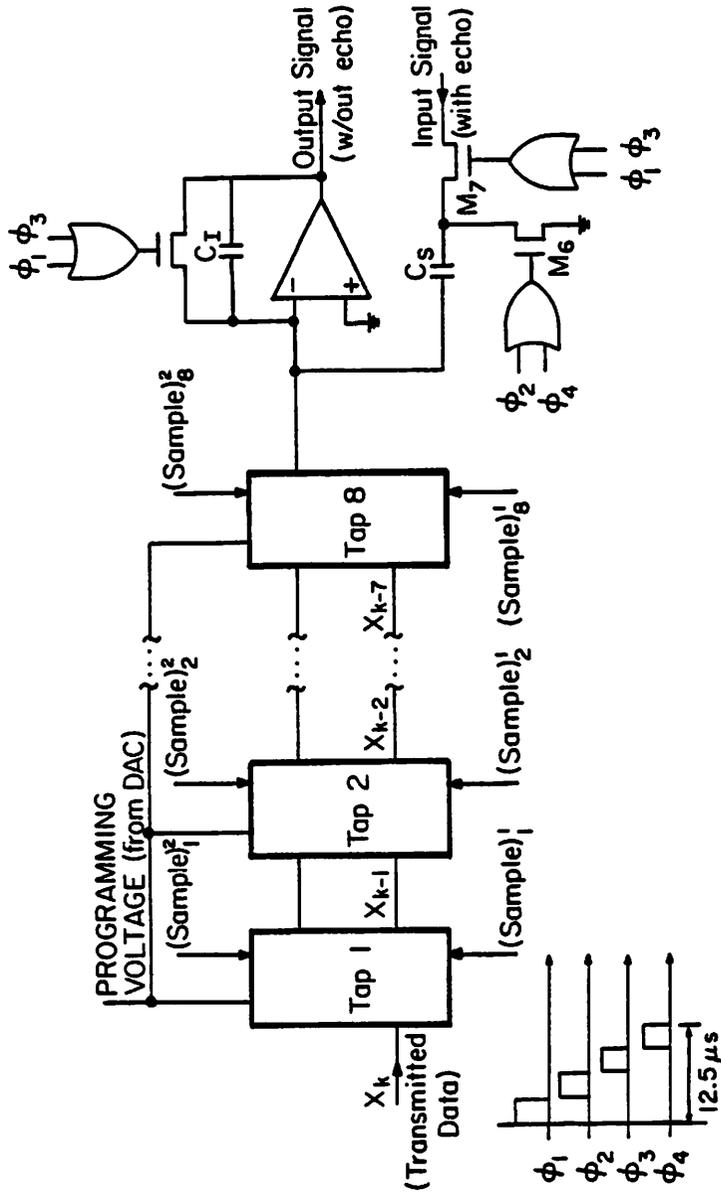
4.4.4. Analog-Digital Echo Canceller

From the analysis of Section 4.3, it is clear that the configuration of Fig. 4.1d is the most attractive of those presented for MOS realization. This approach has been experimentally evaluated by fabricating an MOS chip that implements the functions shown within dotted lines in Fig. 4.2. Two 8-tap programmable transversal filters operate in time-interleaved fashion, sampling at twice the data rate.

A circuit schematic is shown in Fig. 4.9. The data bits are shifted through the dynamic shift register formed by $E_n^{(1)}$ and $E_n^{(2)}$, where the subscript n identifies the tap number and the superscript (1,2) indicates which one of the 2 time-interleaved sections is being considered. Capacitors $C_{A,n}^{(i)}$ are the sample and hold capacitors, with the associated sampling switch $M_{S,n}^{(i)}$ and source follower buffer $M_{1,n}^{(i)}$, $M_{2,n}^{(i)}$. Summing capacitors $C_{B,n}^{(i)}$ are initialized to ground when the corresponding data bit is 0, and to the S/H coefficient when the bit is 1. At the same time the input signal is sampled by capacitor C_s . During the next clock phase the position of the switches associated with the summing capacitors and with capacitor C_s are reversed, and the quantity

$$r_k = s_k - \sum_{n=1}^N a(n)x_{k-n} \quad (4.33)$$

is computed in the charge domain and appears at the output of the summing



(a)

Fig. 4.9a. Circuit schematic of transversal filter. Circuitry inside each box "Tap n" is shown in Fig. 4.9b. $(Sample)_i^j$ represents the sampling signal for the S/H of tap n of the i^{th} of the two interleaved sections of the filter.

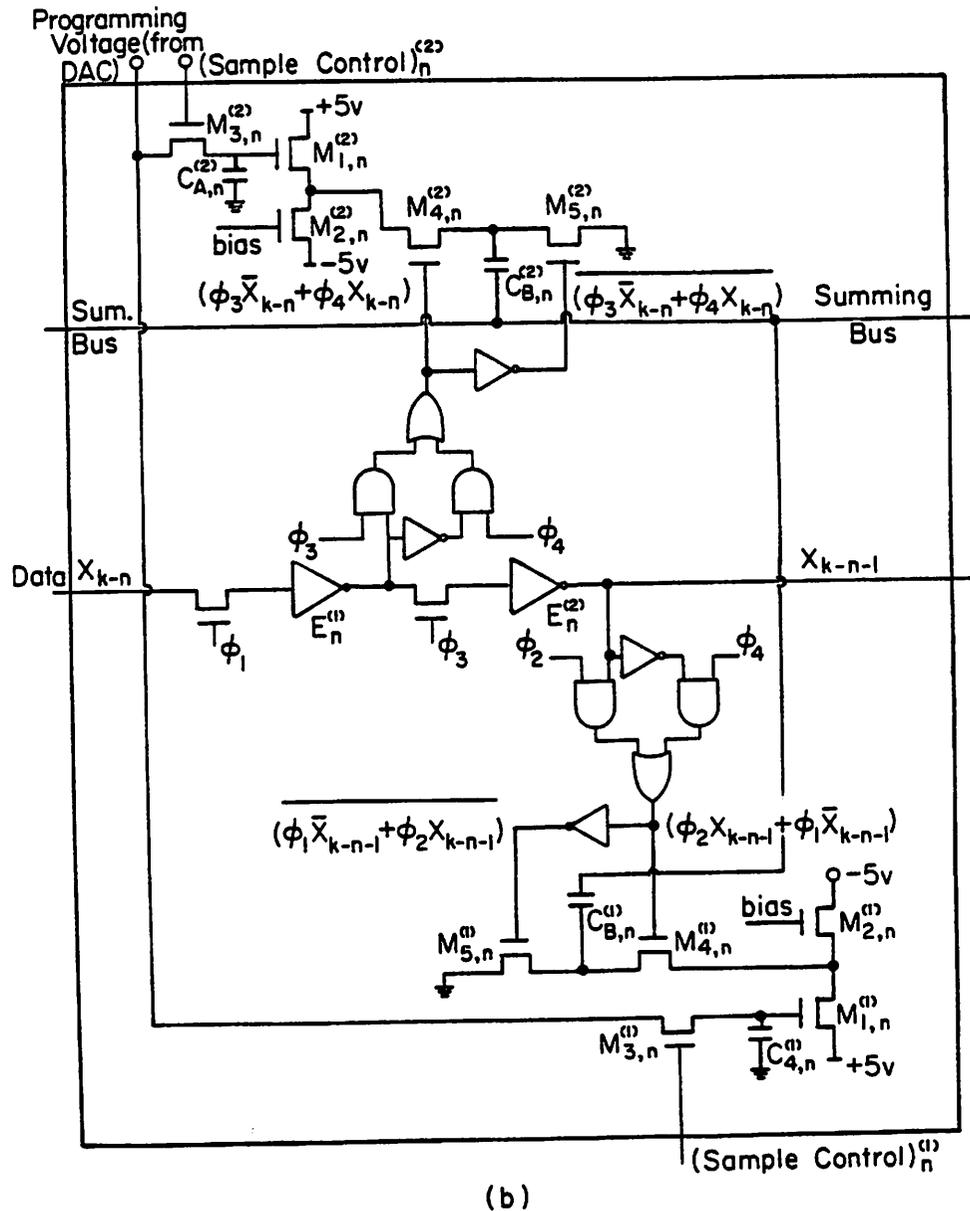


Fig. 4.9b. Circuit schematic of tap n of echo canceller. Superscript (i) identifies each one of the two interleaved sections of the transversal filter.

amplifier.

While Section 4.3.2 concentrated on the effect of the D/A nonlinearity, in the context of Fig. 4.2 the following additional factors are worth considering:

- 1) *Linearity requirements on the sample/hold (S/H) circuits.*
- 2) *Linearity and gain requirements on the summing amplifier.*
- 3) *Linearity requirements on the sampling capacitors.*

With respect to 1), the S/H's are implemented as capacitors buffered by source followers. No attempt was made to linearize them or compensate their offset, since their effect is the same as D/A nonlinearity and is therefore taken care of by the adaptation algorithm, as shown in the Section 4.3.2. By keeping the S/H simple much die area can be saved, since a total of 16 are implemented on chip.

To consider points 2) and 3), refer to Fig. 4.10, and assume that voltages

$$\begin{aligned} V_1(2) - V_1(1) \\ \text{and} \\ V_2(2) - V_2(1) \end{aligned} \tag{4.34}$$

are to be added using a switched capacitor summing amplifier as shown. The arguments 1 and 2 refer to the sampling and charge redistribution phases. This serves as a model of a single-tap transversal filter, with $V_1(1)$ equal to the input signal with echo, $V_1(2)=0$ and either

$$\begin{aligned} V_2(1) &= \text{tap coefficient} \\ V_2(2) &= 0 \\ \text{or} \\ V_2(1) &= 0 \\ V_2(2) &= \text{tap coefficient} \end{aligned} \tag{4.35}$$

depending on whether a weight of +1 or -1 is given to the tap coefficient (as determined by the corresponding data bit). The conclusions extracted from this simple model can be easily extended to a multitap transversal filter. Let also

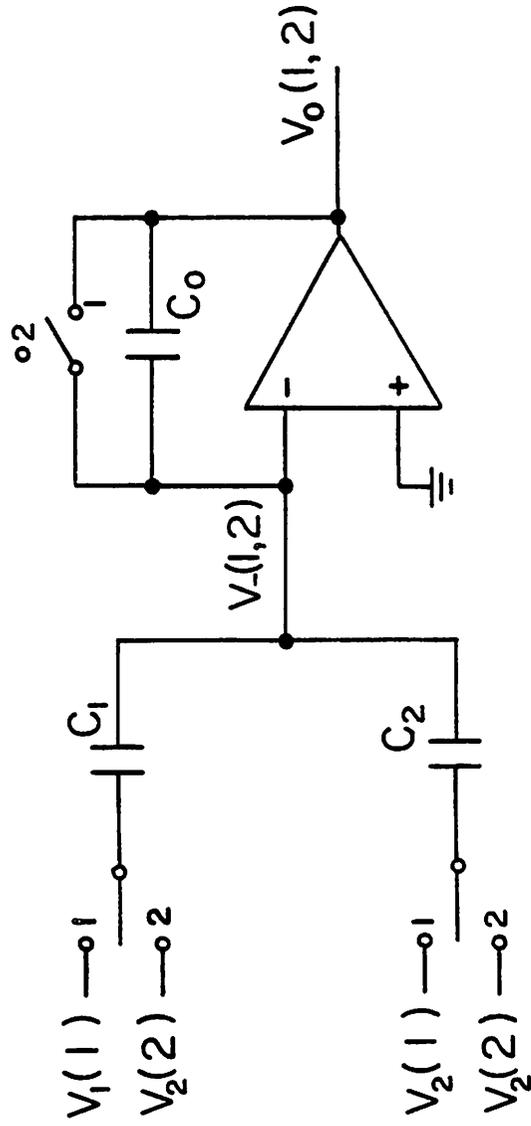


Fig. 4.10. Model of a single-tap transversal filter to compute the effect of nonlinearity.

$$V_0(k) = A(V_-(k)) \quad (k=1,2) \quad (4.36)$$

be the (nonlinear) transfer function of the summing amplifier. It is *not* assumed here that the gain of the amplifier is high. Finally, let

$$q_\tau = h_\tau(V) \quad (\tau=0,1,2) \quad (4.37)$$

define the (possibly nonlinear) capacitors of Fig. 4.10. Using charge conservation,

$$\begin{aligned} & h_0[A(V_-(2)) - V_-(2)] - h_0(0) \\ &= [h_1(V_1(1) - V_-(1)) - h_1(V_1(2) - V_-(2))] + \\ &+ [h_2(V_2(1) - V_-(1)) - h_2(V_2(2) - V_-(2))] \end{aligned} \quad (4.38)$$

In the case of linear capacitances

$$h_\tau(V) = C_\tau V \quad (4.39)$$

and Eq. (4.38) becomes:

$$C_0[A(V_-(2)) - V_-(2)] + (C_1 + C_2) \Delta V_- = C_1 \Delta V_1 + C_2 \Delta V_2 \quad (4.40)$$

where

$$\Delta V_k = V_k(1) - V_k(2)$$

Solving (4.40) for $V_-(2)$, using the fact that $V_-(1)$ is a constant (the offset of the amplifier) and also using (4.36), the output $V_0(2)$ can be found as a function of the *linear* combination

$$C_1 \Delta V_1 + C_2 \Delta V_2 \quad (4.41)$$

This shows that in the case of *linear capacitors*, the nonlinear distortion of the summing amplifier affects the total sum, which for the echo canceller of Fig. 4.2 is the *residual signal*, after echo cancellation. Thus no uncancellable echo distortion components, or intermodulation components between the received signal and the echo, are created. Observe that this is true even in the case when the amplifier gain is relatively low, since no high gain assumption has been made.

Now consider the effect of a voltage coefficient in the capacitors. Assume that

$$h_*(V) = C_* V(1 + \gamma V) \quad (4.42)$$

Then, substituting in (4.38) and collecting terms, it can be seen that the linear combination of (4.41) is corrupted by the distortion term:

$$\begin{aligned} & \gamma C_1 [(V_1(1) - V_-(1))^2 - (V_1(2) - V_-(2))^2] + \\ & \gamma C_2 [(V_2(1) - V_-(1))^2 - (V_2(2) - V_-(2))^2] \end{aligned} \quad (4.43)$$

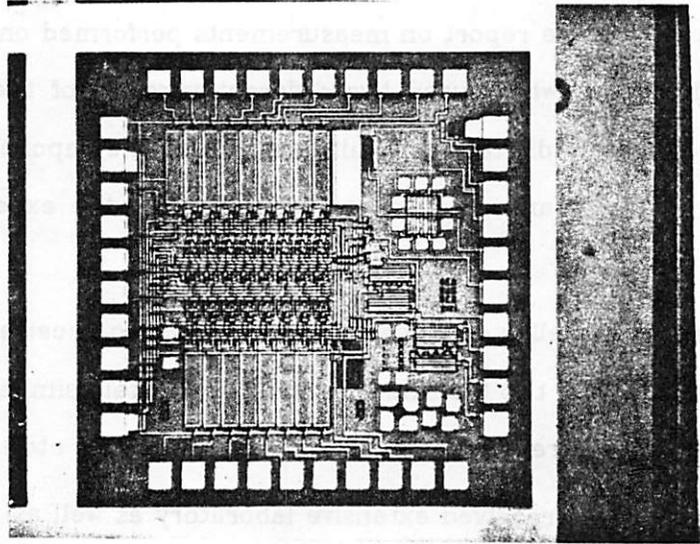
Since only an upper bound for the voltage coefficient γ is required, let us assume that all the voltages in (4.43) are bounded by some voltage V_{\max} . Then, in order to ensure a distortion lower than -60 dB, all we require is:

$$\gamma V_{\max} \ll 10^{-3} \quad (4.44)$$

In MOS technology, low voltage coefficient capacitors can be easily obtained. One option is to use poly to poly capacitors, if a process with double layer of polysilicon is available. Another option (the one we used) are poly to n+ capacitors. A high dose implant forms the bottom plate of the capacitors, providing adequately low voltage coefficients. Reference [57] gives a voltage coefficient as low as 20 ppm for a bottom plate doping of 10^{20} cm^{-3} . This ensures that (4.44) is satisfied.

To complete the description of this analog-digital echo canceller, the processor performing the adaptation must be considered. The processor was implemented again using 2901 bit slice microprocessors (as in the fully digital implementation). A circuit schematic is given in Appendix D. In this case, for simplicity, the processor was designed in such a way that no loops exist in the microprogram, which as a result has a strictly linear operation. Then the timing diagram coincides with the state diagram. The design of this processor follows identical criteria as that of Section 4.4.2, so that no further details will be given here.

Figure 4.11 shows a photograph of the chip, fabricated in NMOS silicon gate process with $7 \mu\text{m}$ design rules. Chip size is 3 mm on each side (between centers of scribe lines). The area of the active parts is 1.5 mm^2 , and can be



Chip Photo

Fig. 4.11. Experimental echo canceller chip in NMOS $7\mu\text{m}$ silicon gate technology.

decreased using smaller S/H capacitors (instead of the conservative 20 pF used here) and a more advanced process.

4.5. EXPERIMENTAL RESULTS

In this Section we report on measurements performed on an experimental breadboard modem which used two different versions of the echo canceller, namely a completely digital one (built using discrete components) as described in Section 4.4.2 and an analog/digital version using the experimental custom chip of Section 4.4.4.

Each echo canceller was used in one of the two receivers of a complete two-way system. The two transmitters were completely similar, as were all the other sections of the receiver (equalizer, timing recovery, etc.).

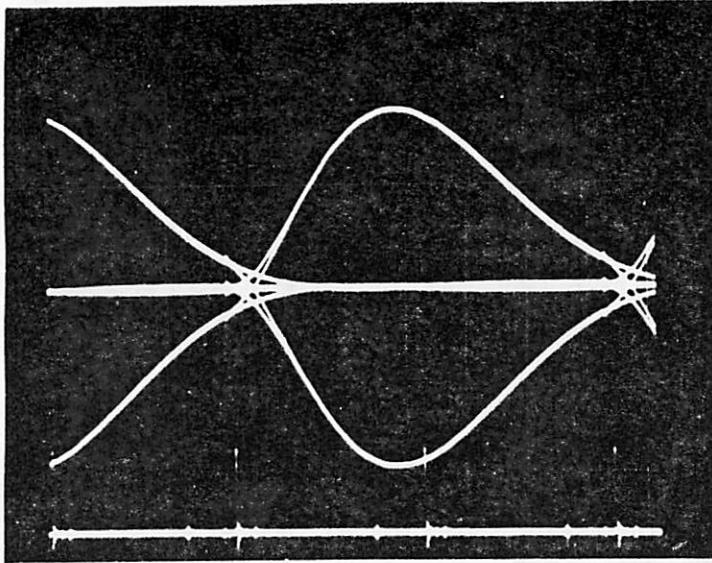
The modem has received extensive laboratory as well as field testing, with line attenuations up to 44 dB. The field test was performed in Ora Loma, California, is a small (300 line) local office where the crosstalk and impulse noise impairments were minimal. Subscriber loops were remotely looped back so that both ends of the subscriber loop were available in the office.

The degree of cancellation achieved was measured by measuring the SNR at the input and output of the canceller. Specifically, both SNR's were determined by measuring the peak far-end signal voltage (with the local transmitter turned off) and the peak echo signal (with the far-end signal turned off). By this measurement technique, the achieved degree of cancellation was 63 dB. Error rate measurements showed a BER (bit error rate) lower than 10^{-8} with line attenuations at 40 kHz up to 40 dB. BER increased to 10^{-4} when the line attenuation was 44 dB. This increase was attributed not to the echo canceller, but rather the noise from the digital circuitry and the effect of equalization errors due to the very simple equalizer that was used.

Figure 4.12a shows the eye diagram at the output of the transmitter. Figure 4.12b shows the spectrum of the transmitted signal, measured at the output of the transmitter (top trace) and on an adjacent pair (bottom trace). The settings of the spectrum analyzer were changed after recording the top trace, so that the crosstalk attenuation (not directly observable in the picture) was 70 dB. Crosstalk from analog frequency division multiplexing channels caused the narrow lines observable above 80 kHz.]. Figure 4.13a shows the eye diagram at the input of the echo canceller with the local transmitter turned off, while Figure 4.13b shows the signal at the same point with the local transmitter on and set to the nominal output level of 600 mV peak. Note the different scales in the two pictures. Prior to echo cancellation, the echo was about 38 dB higher than the far-end signal. The eye diagram of the echo signal is closed because the equalizer has been adjusted to open the received signal eye, not the eye for the echo. Figure 4.14 shows the eye diagram at the output of the echo canceller when operating with a line attenuation of 40 dB. The quantization error as well as the sampled nature of the signal can be clearly observed.

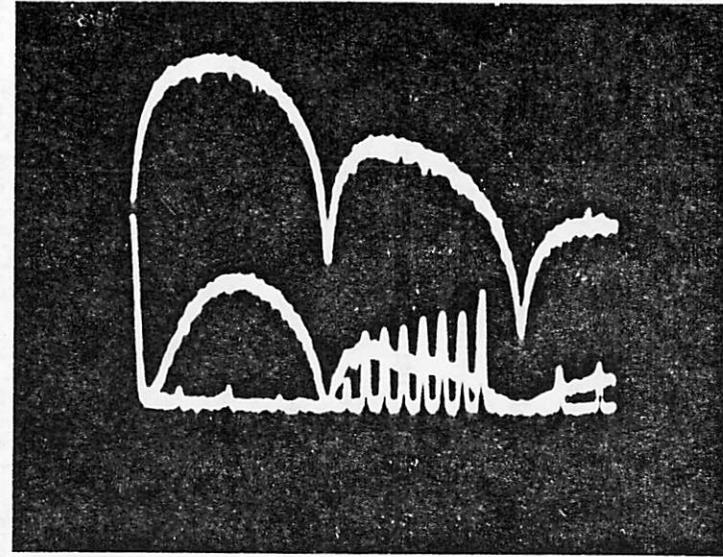
Impulse noise was the dominant source of errors we observed in both laboratory and field testing. Impulse noise almost always caused a single bit error on the line, which expanded to three errors in passing through the scrambler. Such errors are inconsequential for voice communication. For data communication, error detection followed by re-transmission might well provide a satisfactory solution. Alternatively, a simple form of burst error-correcting code would be highly effective in overcoming the impulse noise errors we observed. Digital circuitry for such encoding could be included on the same chip as the functions described above.

Measurements were made to determine the magnitude of near-end crosstalk from other (unsynchronized) channels at 80 kb/s on adjacent pairs in typical cables. Of course, such crosstalk could not be reduced by the echo



(a)

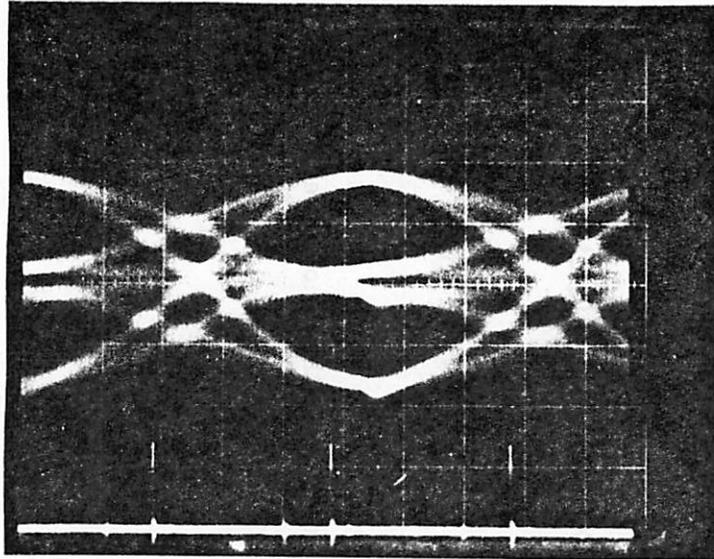
Transmitted pulse shape
 200 mV/cm
 2 μ s/cm



(b)

Transmitter spectrum
 and crosstalk spectrum
 20 dB/cm
 20 kHz/cm
 70 dB between spectra

Fig. 4.12. (a) Eye diagram at the output of the transmitter. Peak transmitted signal is 600 mV. (b) Spectrum of the transmitted signal at the output of the transmitter (top trace) and on an adjacent pair (bottom trace). Crosstalk loss for this worst observed case is 70 dB. Also seen is the crosstalk from analog carriers above 80 kHz.

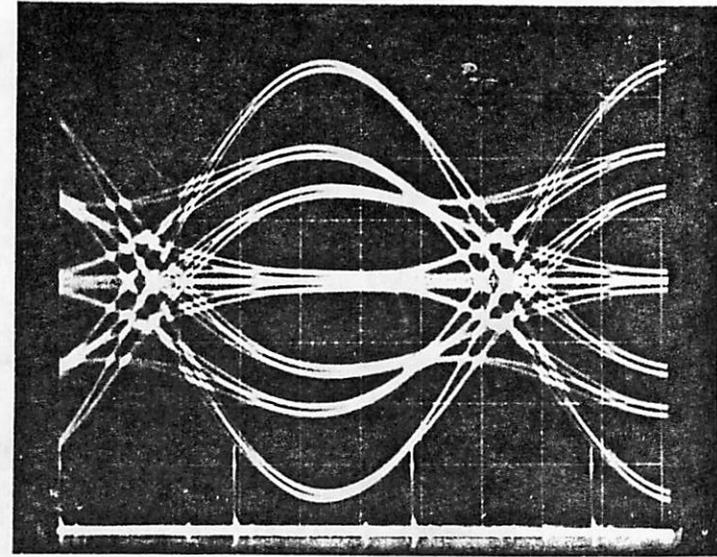


(a)

Local Transmitter off

20 mV/cm

2 μ s/cm



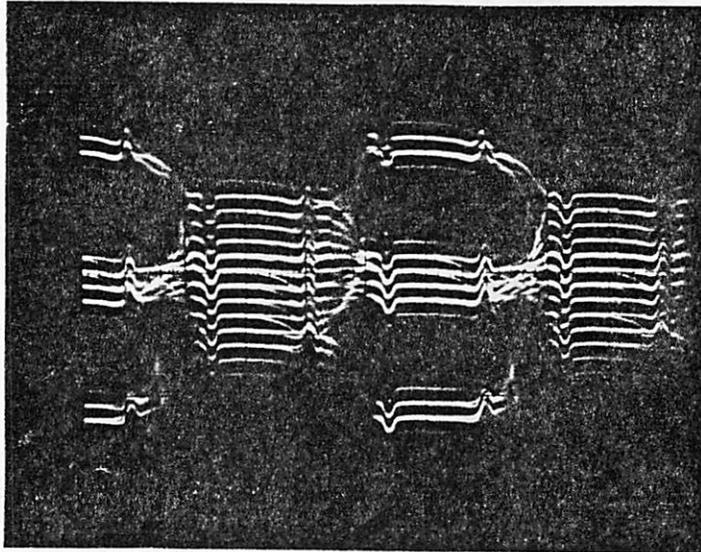
(b)

Local Transmitter on

1 V/cm

2 μ s/cm

Fig. 4.13. Eye diagram at the input of the echo canceller. (a) With local transmitter off. Scale: 20 mV/cm, 5 μ s/cm. (b) With local transmitter on. Scale: 1 V/cm, 5 μ s/cm.

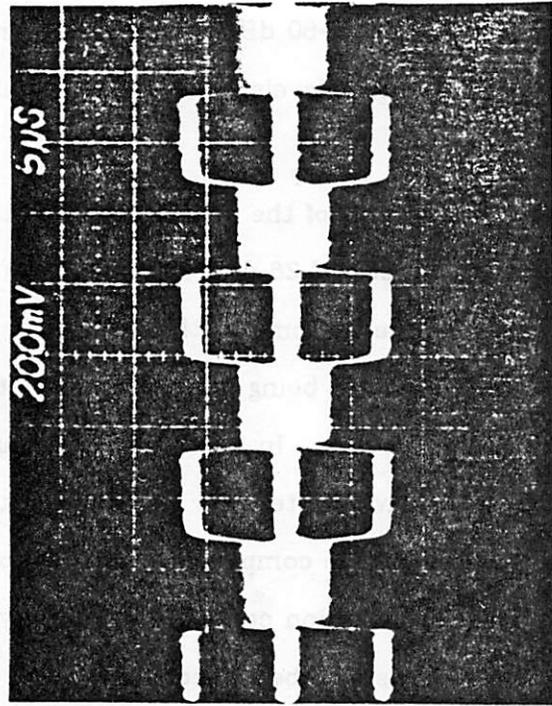


Eye diagram at output of Echo Canceller :

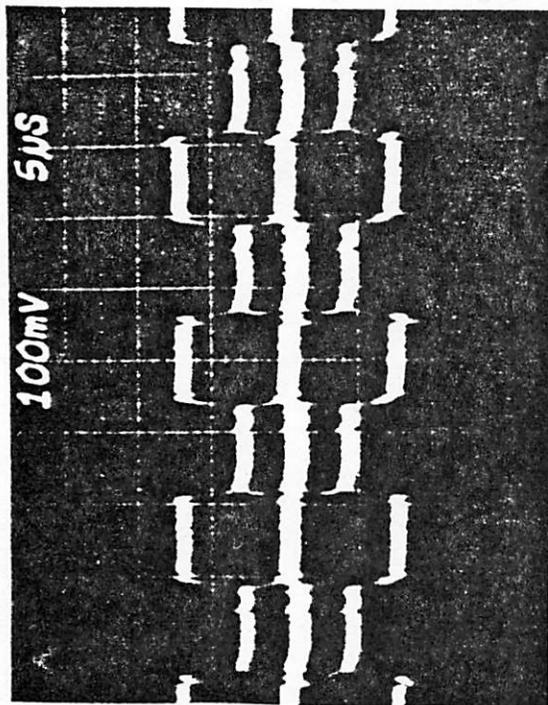
$2 \mu\text{s/cm}$

100 mV/cm

Fig. 4.14. Eye diagram at the output of the echo canceller with 40 dB of line attenuation.



(b)



(a)

Fig. 4.15. Eye diagrams at the outputs of the two receivers. (a) Using the echo canceller of Fig. 4.1d. (b) Using the echo canceller of Fig. 4.1b.

canceller. Measured crosstalk was at -70 to -80 dB relative to the transmitted signal level for a single interferer. At these levels, near-end crosstalk is not a limitation.

Figure 4.15 compares the eye diagrams of the breadboard and the custom chip when operating with a line attenuation of 26 dB. The maximum measured echo cancellation of the modem using the custom chip (measured as previously described) was 40 dB, with the limiting factor being not nonlinear distortion but digital noise picked up by the analog circuits. In fact, the nonlinearity of the sample and hold circuits alone would have limited the cancellation to approximately 20 dB were the adaptation not able to compensate for this nonlinearity. We are confident that this 40 dB of cancellation can be substantially improved with additional effort in designing circuits with better common mode and power supply rejection ratios. In addition, monolithic integration of the entire modem would reduce the problems of digital noise pickup experienced in both implementations.

CHAPTER 5

CONCLUSION

Several alternative echo cancellation techniques have been reported in previous chapters. This Chapter will attempt to synthesize the experience gained through the experimental evaluation of those systems and extract conclusions for the monolithic implementation of fully integrated hybrid-method digital subscriber loops. Fig. 5.1 shows a block diagram of the chip. For the purpose of the analysis that will be carried out in this Chapter, four blocks can be considered, namely, the echo canceller, the cancellation block, the timing recovery block and the input and output filters. Section 5.1 covers the digital processor implementing the echo canceller and the decision feedback equalizer. Section 5.2 describes the (analog) cancellation block in which the digital to analog conversion of the echo canceller output, the echo cancellation and the analog to digital conversion of the residual signal are performed using a single multiplexed capacitor array. Section 5.3 analyzes the timing recovery block, and Section 5.4 considers the input and output filters.

5.1. DIGITAL PROCESSOR

The echo canceller uses two interleaved sections ("phases") to generate echo replica samples at twice the data rate (160 kHz). Each section has 14 linear taps. In addition, 6 nonlinear taps (as described in Chapter 3) are used to correct nonlinear distortion introduced by the DAC and pulse asymmetry of the transmitter filter. Four taps of decision feedback equalization are implemented by the digital processor as well. However the DFE works independently of the echo canceller, and is not functional during the start-up cycle, when timing has not been acquired yet. It only starts working when the timing recovery block

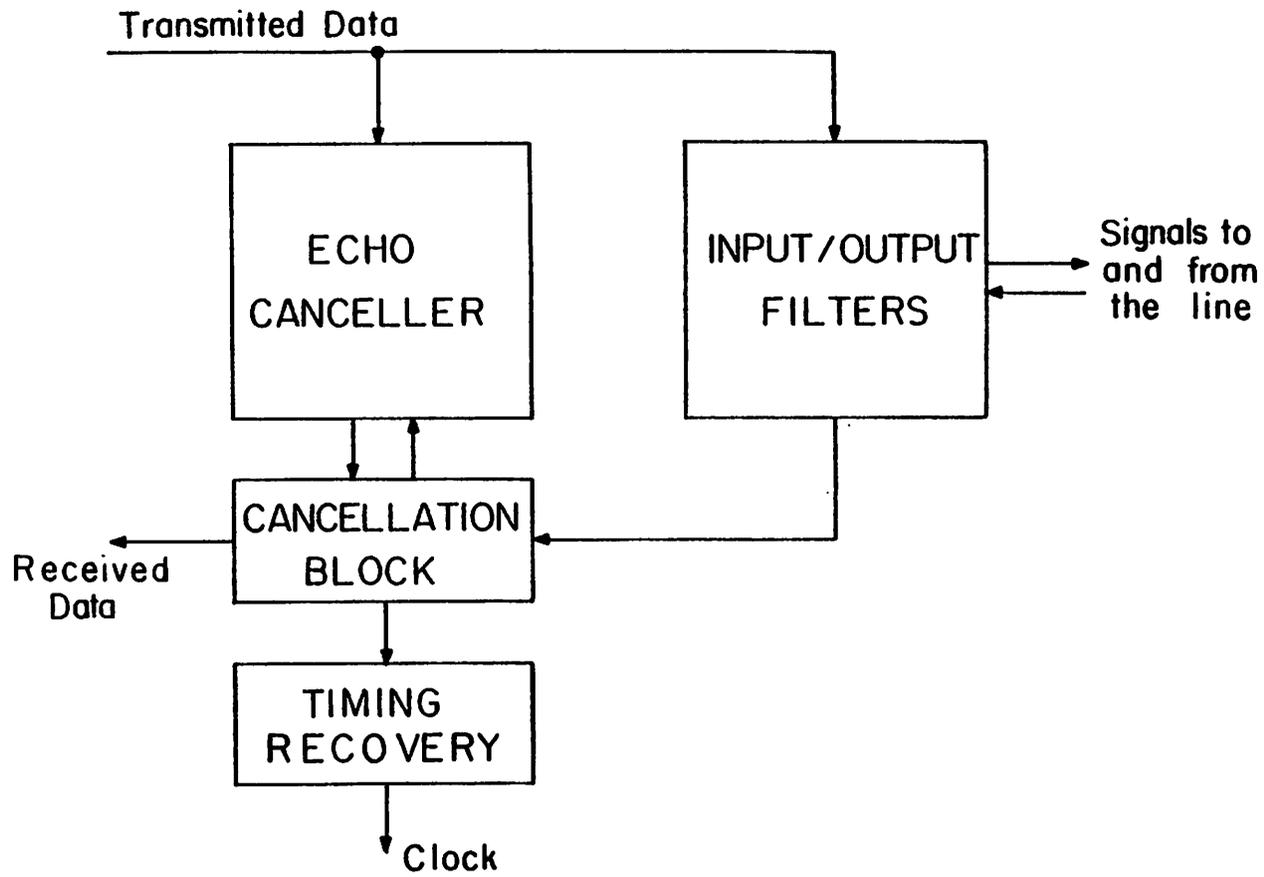


Fig. 5.1. Block diagram of proposed new implementation of transceiver chip.

issues a signal signifying that the phase lock loop is in lock.

5.1.1. Digital Processor Operation The operation of the digital processor can be divided into three cycles:

- 1) *Reset cycle*
- 2) *Echo canceller convergence cycle*
- 3) *Decision feedback equalizer convergence and normal operation*

5.1.1.1. Reset Cycle

During this cycle all memory locations and registers are cleared.

5.1.1.2. Echo Canceller Convergence Cycle

The echo canceller is allowed to converge and the DFE taps are kept clear. Gearshifting is performed during convergence of the echo canceller, to use more effectively the dynamic range of the A/D and speed up convergence. This cycle ends when the phase detector in the timing recovery module issues the IN_LOCK control signal, informing that lock conditions have been achieved.

5.1.1.3. DFE Convergence and Normal Operation

When lock has been acquired, the DFE is allowed to operate and converge, and normal operation begins.

5.1.2. Processor Architecture

Fig. 5.2 shows a block diagram of the processor. A 1056 bit memory is used to store the 20 taps coefficients of each of the two echo canceller phases and the 4 tap coefficients of the DFE. Internal arithmetic precision is 24 bits, so the memory is organized as 44 words of 24 bits each. The memory uses a three transistor cell with independent *data-in* and *data-out* lines, as well as independent *read* and *write* accesses. Although this may seem to render the cell layout

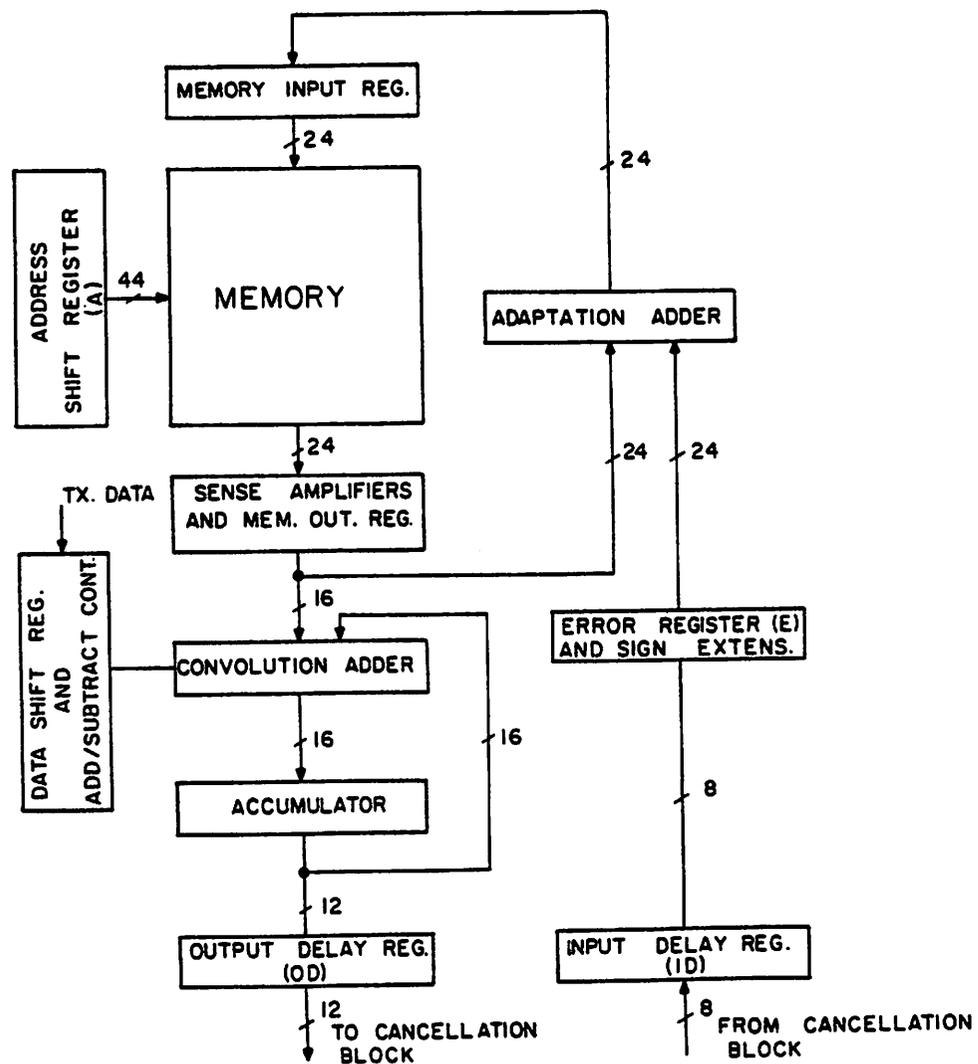


Fig. 5.2. Block diagram of the digital processor implementing the echo cancellation and equalization algorithms.

inefficient, it actually allows us to perform independent read and write accesses to different cells, and so *read-modify-write* cycles can be implemented, that is, while one coefficient is being read, the *previous* coefficient is being written over and corrected simultaneously, all in one clock cycle. This cuts down the number of states, and the clock frequency, by one half.

There are two adders, one to compute the convolution sum and the other to correct the coefficients. The first only needs a precision of 16 bits, since the output is a 12 bit number routed to the D/A (4 extra bits are used to prevent accumulation of arithmetic errors). Since the second performs the adaptation, which requires high dynamic range, its precision is 24 bits. A 2-phase clock is used. Phase Φ_1 is used in the memory to precharge the data-in and data-out lines, whereas Φ_2 enables the read and/or write controls. The adders are implemented in CMOS domino logic, which also requires a 2-phase clock.

Each row of the memory corresponds to an independent coefficient, and the "*row decoder*" is implemented using a shift register which shifts a 1 to enable the rows in a sequential fashion. The write control of a given row is connected to the read control of the following row, so that while one row is being read, the previous one is being written.

5.1.3. State and Timing Diagrams Figs. 5.3 and 5.4 show the state and timing diagrams of the processor. The *reset cycle* is entered each time the *reset* signal is asserted. At completion of the execution of the reset routine, the *echo canceller convergence cycle* is entered. After convergence of the echo canceller a timing signal will be available for the timing recovery block to lock on, and when this occurs, the IN_LOCK signal is asserted. This signal causes the processor to exit the echo canceller convergence cycle and enter the *DFE convergence and normal operation cycle*. The processor keeps running in this state indefinitely, unless a loss of lock is detected, in which case a jump to the previous cycle is

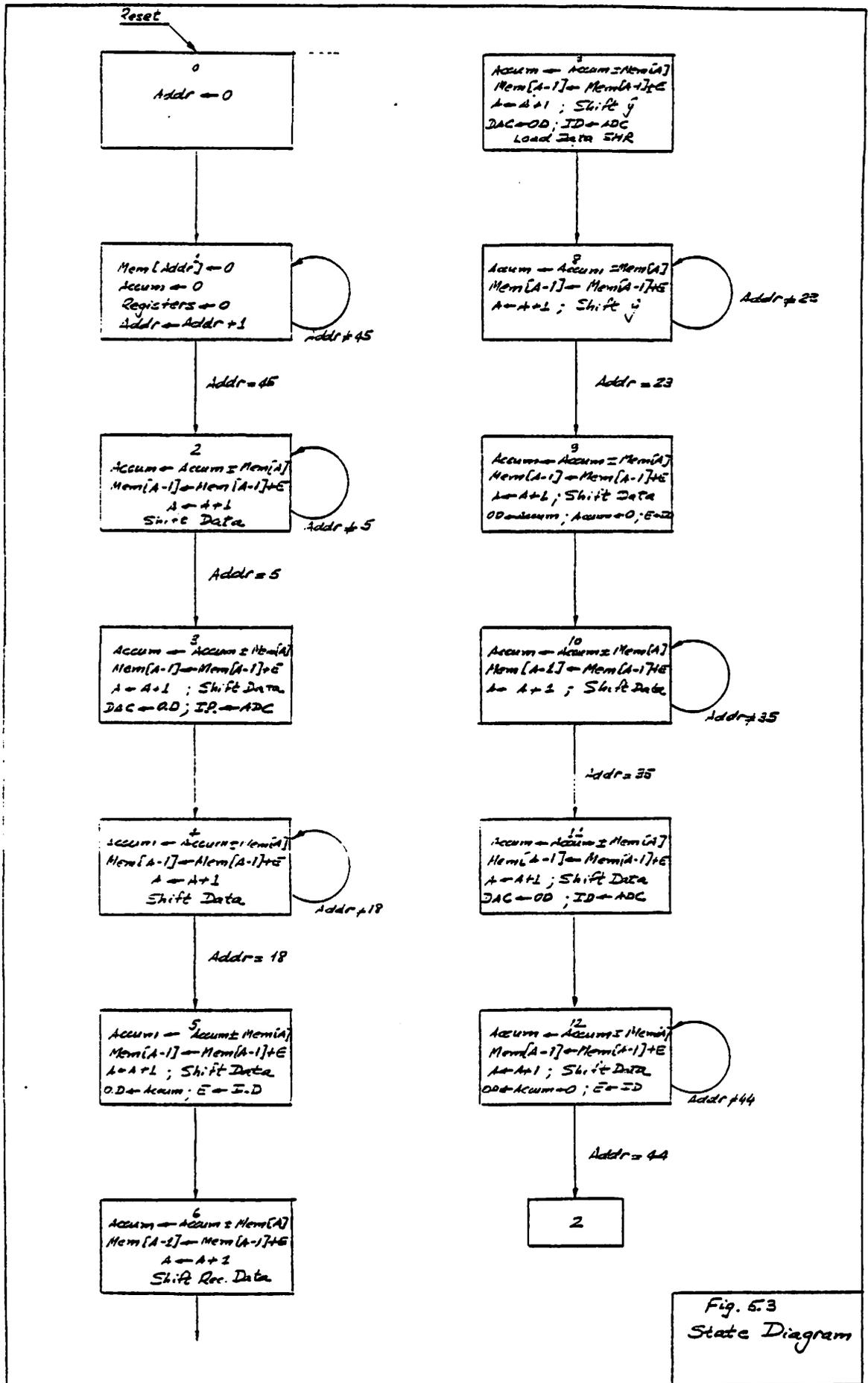


Fig. 5.3
State Diagram

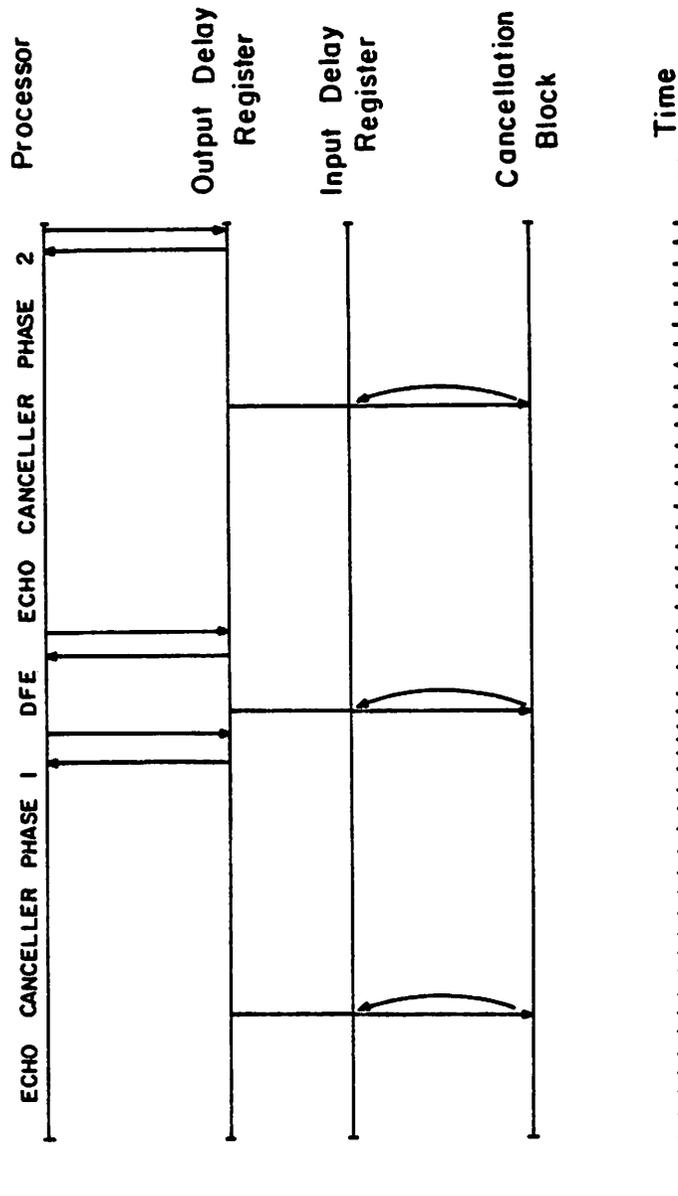


Fig. 5.4. Timing diagram of the operation of the digital processor and the cancellation block.

performed.

In order to speed up convergence of the echo canceller, and also to decrease the dynamic range of the A/D, gearshifting is used. However, gearshifting is controlled by an independent digital circuit and thus it is transparent to the processor. A *shift counter* counts the number of bits that the A/D output must be shifted before being sent to the processor. The shift counter is incremented when the adaptation gain is to be decreased. A counter keeps track of the number of adaptation cycles that have elapsed and decides when gearshifting must be performed.

5.2. CANCELLATION BLOCK

Fig. 5.5 shows the cancellation block. A single multiplexed capacitor array and a resistor string voltage divider are used for both the D/A and the A/D conversions. The capacitor array provides 8 bits and the resistor string the remaining 6 bits. Moreover, cancellation of the echo of the incoming signal also takes place in this capacitor array. The input signal is sampled by an extra capacitor of maximum size (C'_{32}). The circuit performs simultaneously the digital to analog conversion of the echo replica, the subtraction of this echo replica from the input signal in the charge domain, and the analog to digital conversion of the residual signal. The digital to analog conversion of the echo replica and the echo cancellation take place in a very natural way, as a part of the initialization cycle of the capacitor array that is required anyway for the analog to digital conversion. Thus the digital to analog conversion and the echo cancellation are done almost for free. In the analysis that follows it will be assumed that the echo replica is a two's complement 12 bit integer. The residual signal coming out of the A/D is a two's complement 12 bit integer, but only 8 of these bits (or fewer if speed constraints so require) are tested. Before convergence of the echo canceller the residual signal that the A/D must handle is very large, and so

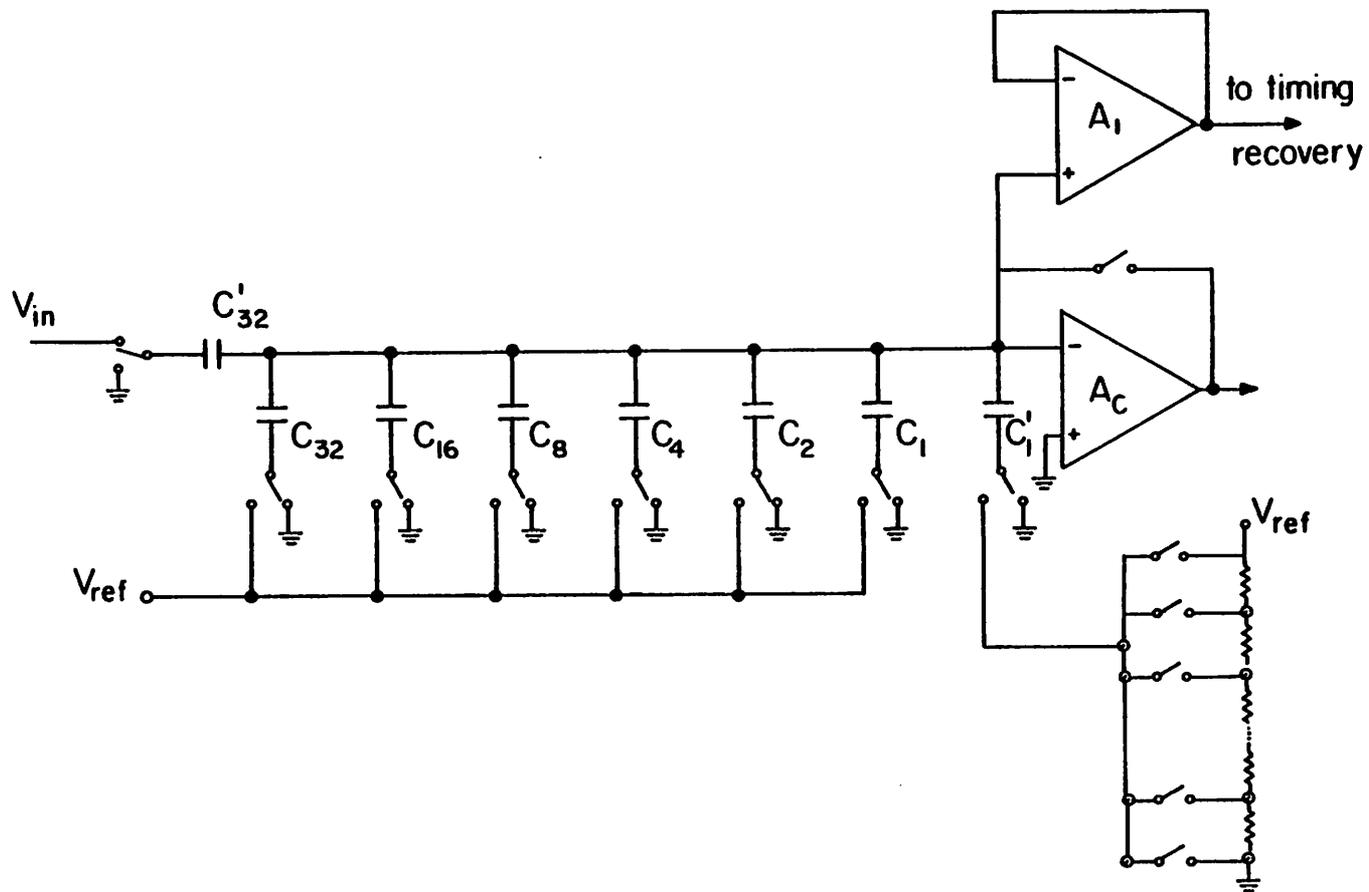


Fig. 5.5. Cancellation block. A single multiplexed capacitor array is used to perform the D/A conversion of the echo replica, the echo cancellation, and the A/D conversion of the residual signal. The D/A conversion and the sampling of the input signal are performed simultaneously with the capacitor initialization and so they take no extra time.

the most significant bits must be tested. However, after the echo canceller has converged, the residual signal may be very small (for large line attenuation) and most of the available A/D resolution would be wasted if the test started with the most significant bits. Therefore the test starts at a bit position that is adaptively changed to track the level of the residual signal, and higher order bits are simply set equal to the sign bit. Thus the *relative* as opposed to the *absolute* quantization error of the residual signal is kept approximately constant. Admittedly, this technique requires a comparator that can resolve the least significant bit that is going to be tested under the worst case condition of minimal received signal, but this is considered a feasible requirement.

Furthermore, before starting the test of the bits that are going to be tested in a given conversion, the least significant bits that *are not* going to be tested, can be switched at random in order to provide dithering noise to the feedback signal, which increases the effective resolution of the A/D [].

Operation of the circuit takes place in the following steps:

- 1) The comparator is reset, so that the top plate of the array becomes a virtual ground. At the same time, capacitor C'_{32} samples the input signal, and capacitors C_1 to C_{16} are initialized to V_{ref} if the corresponding input bit to the DAC is 1, or to ground if such bit is 0. Capacitor C_{22} is initialized to ground if the sign bit is 1 or to V_{ref} if the sign bit is 0, and capacitor C'_1 is initialized to the tap voltage of the resistor array that has been selected by the decoder, according to the 6 least significant bits of the echo replica.
- 2) The reset switch of the comparator is opened, and switches S'_1, S_1, \dots, S_{16} , and S'_{32} are thrown to ground, and S_{32} is thrown to V_{ref} (if not already there). At this point, the *difference* between the input signal and the echo canceller output appears on the top plate of the capacitor array, and is buffered by A_1 and sampled and held by S/H for subsequent use in the

timing recovery circuit.

- 3) The low order bits that are not going to be tested are switched at random to introduce dithering noise.
- 4) The comparator output is tested. If the voltage of the top plate is negative, S_{10} is left in its position at V_{ref} . If the test voltage is positive, the switch is thrown to ground. All the high order bits which are not going to be tested are set to the value of the sign bit and the associated switches thrown accordingly.
- 5) Testing of the low order bits continues until completion.

Digital logic can be used to decide what bits of the A/D will be tested. At the beginning of the processor operation testing starts from the most significant bit. After the echo canceller starts converging and the residual signal decreases, some of the higher order bits will remain equal to the sign bit for every sample, a situation that can be easily recognized with simple digital logic. Thus testing can start at the first *significant* bit position.

Although single ended half-circuits have been used throughout this discussion for the sake of simplicity, the actual implementation is fully differential, in order to improve the noise rejection of the circuits. A high noise rejection of the analog circuits is of primary importance, considering that these circuits are going to be located on chip close to the digital processor.

5.3. TIMING RECOVERY BLOCK

Timing recovery operates in this chip basically in the same way as in the breadboard system that was described in Chapter 2. One important difference is that the filters are implemented here using switched capacitor techniques, and so they are sampled-data filters, as opposed to the continuous-time filters used in the breadboard implementation. As said in Chapter 2, the reconstruc-

tion filter is a fourth order maximally flat bandpass filter, centered at 40 kHz, with a bandwidth of 16 kHz. Fig. 5.6 shows a circuit schematic of the switched capacitor realization and Fig. 5.7 shows computer simulations using the program DIANA.

5.4. INPUT AND OUTPUT FILTERS

The input and output filters realize the response that was discussed in Chapter 2. Here, switched-capacitor versions are presented. The circuit schematics are given in Figs. 5.8 and 5.10, and the computer simulations in Figs. 5.9 and 5.11. The sampling rate is 960 kHz, which corresponds to 12 samples per period of the data signal. Continuous-time second order Sallen and Key sections are used as antialias at the input of the receiver and as a smoothing filter at the output of the transmitter.

5.5. CONCLUDING REMARKS

The research reported here has proved the feasibility of the integration of hybrid-method digital subscriber loops. The major difficulties in implementing such a system have been evaluated both theoretically and experimentally. The system whose integration is proposed here is now well understood, and the techniques that can be used have been presented. We expect to see in the near future commercial implementations of this system, which will hopefully benefit from the results of this research.

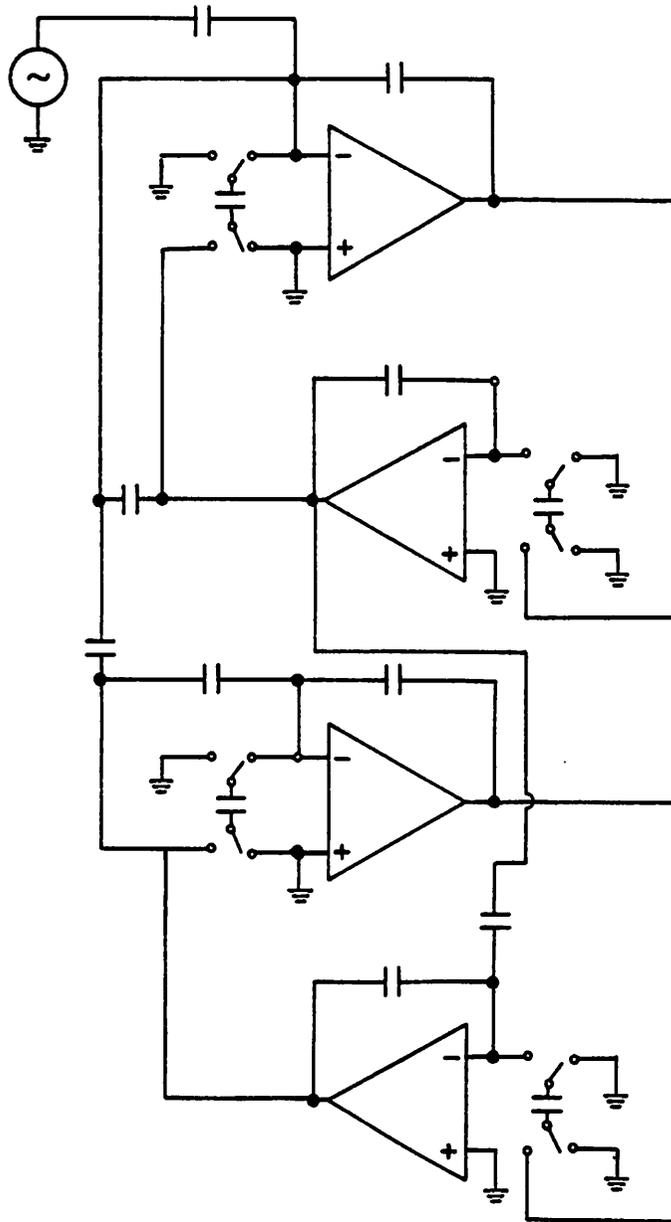


Fig. 5.6. Circuit schematic of reconstruction filter.

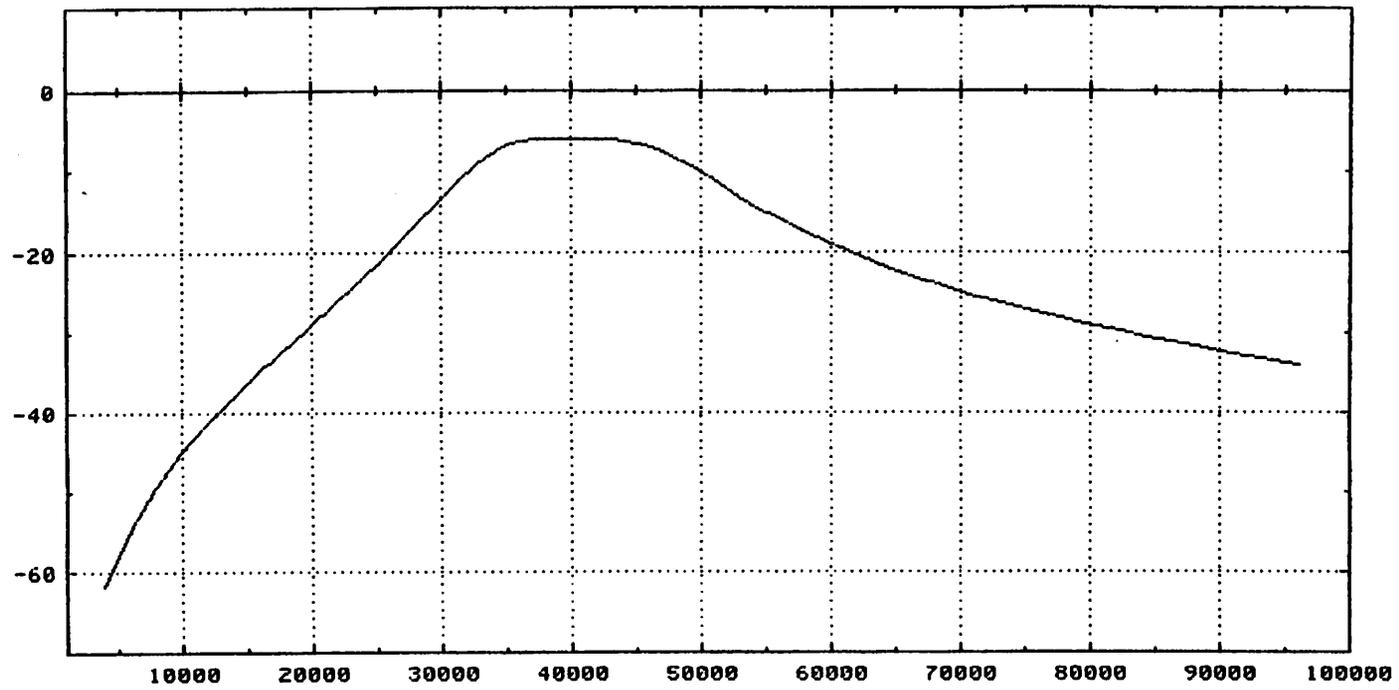


Fig. 5.7. Computer simulation of the frequency response of the reconstruction filter using the program DIANA.

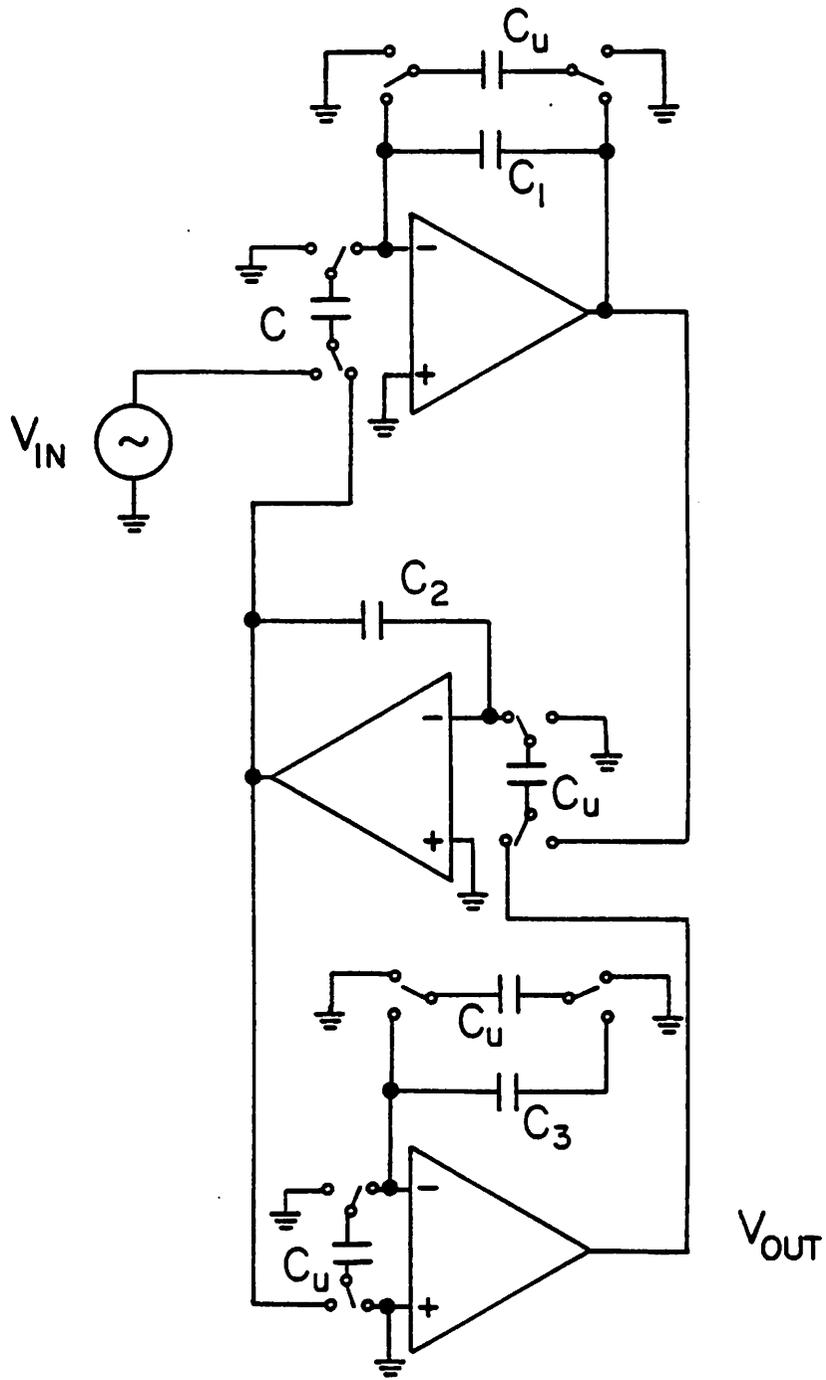


Fig. 5.8. Circuit schematic of transmitter output filter.

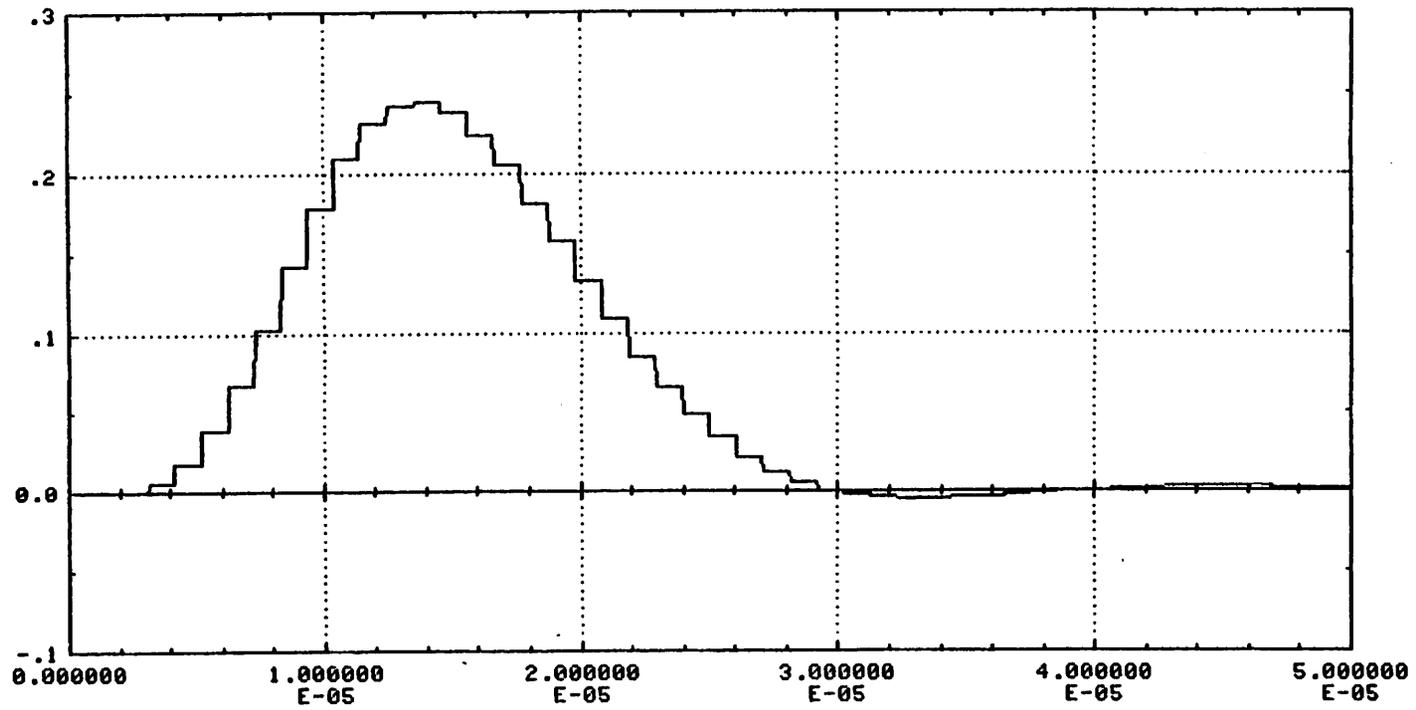


Fig. 5.9a. Transient response of transmit filter.

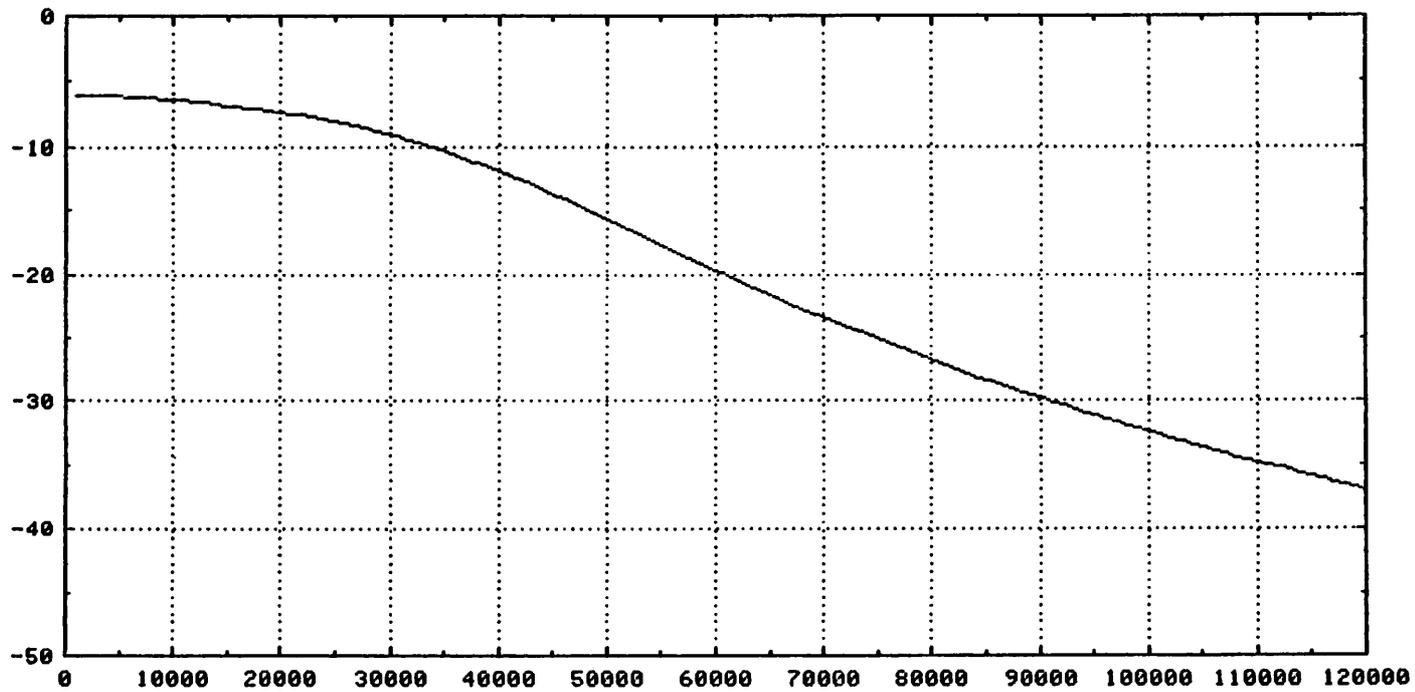


Fig. 5.9b. Amplitude response of transmit filter.

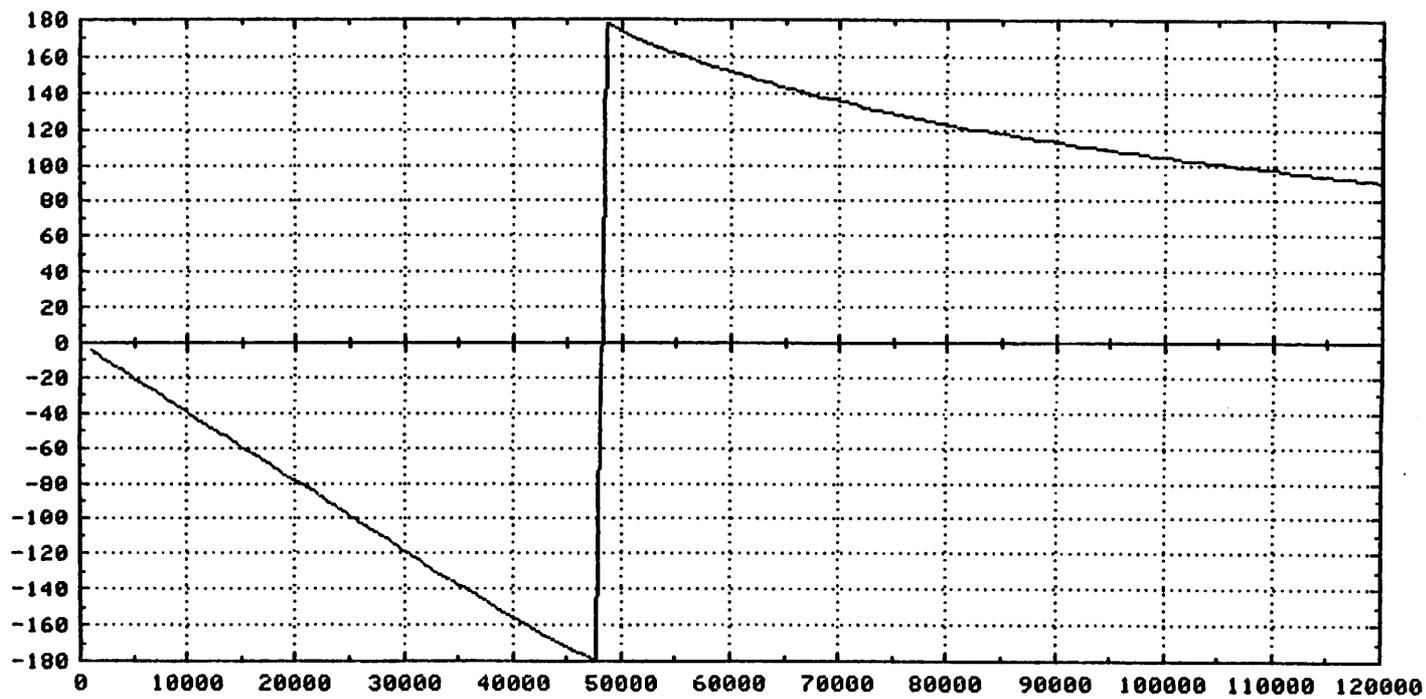


Fig. 5.9c. Phase response of transmit filter.

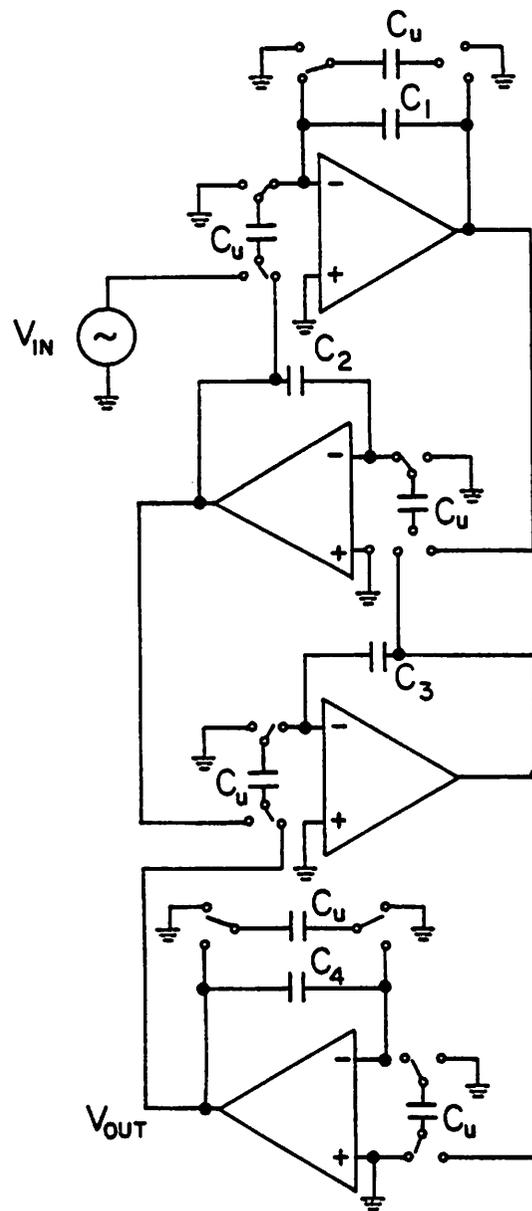


Fig. 5.10. Circuit schematic of receive filter.

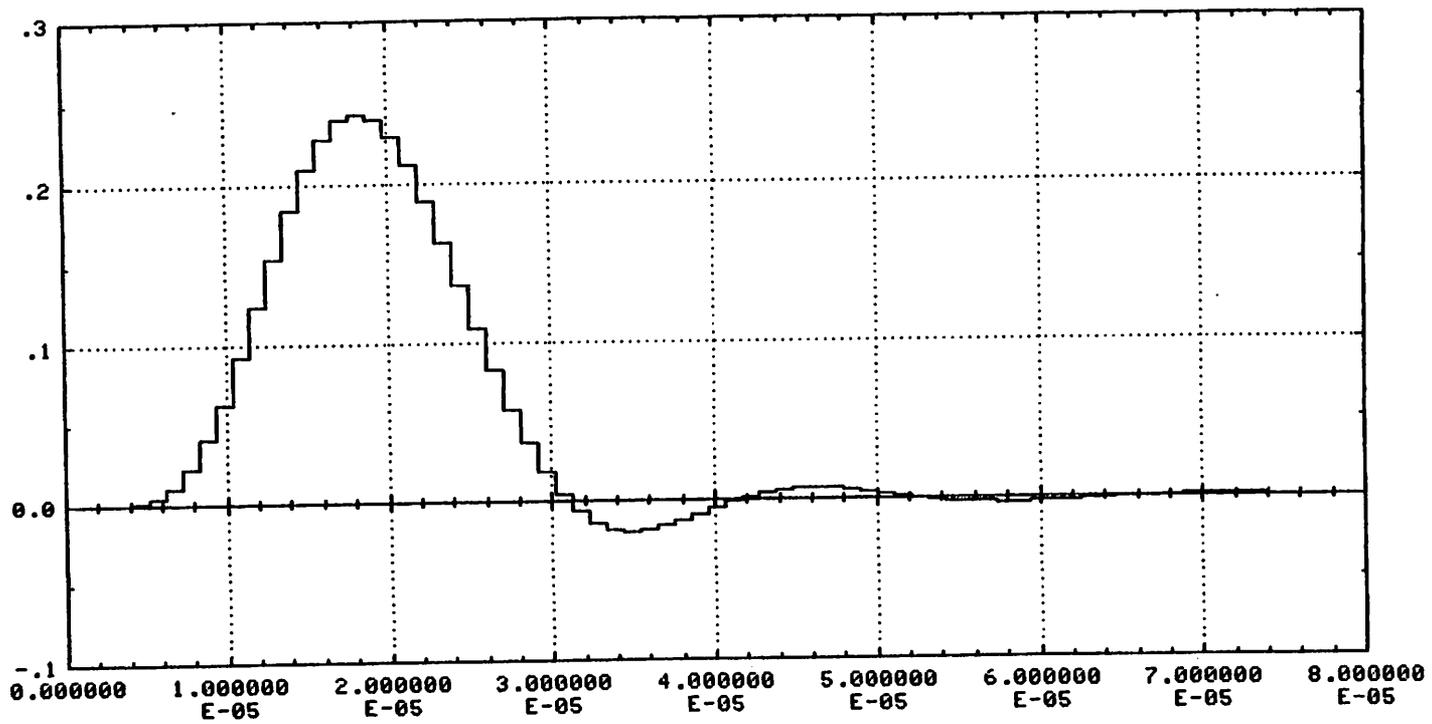


Fig. 5.11a. Transient response of receive filter.

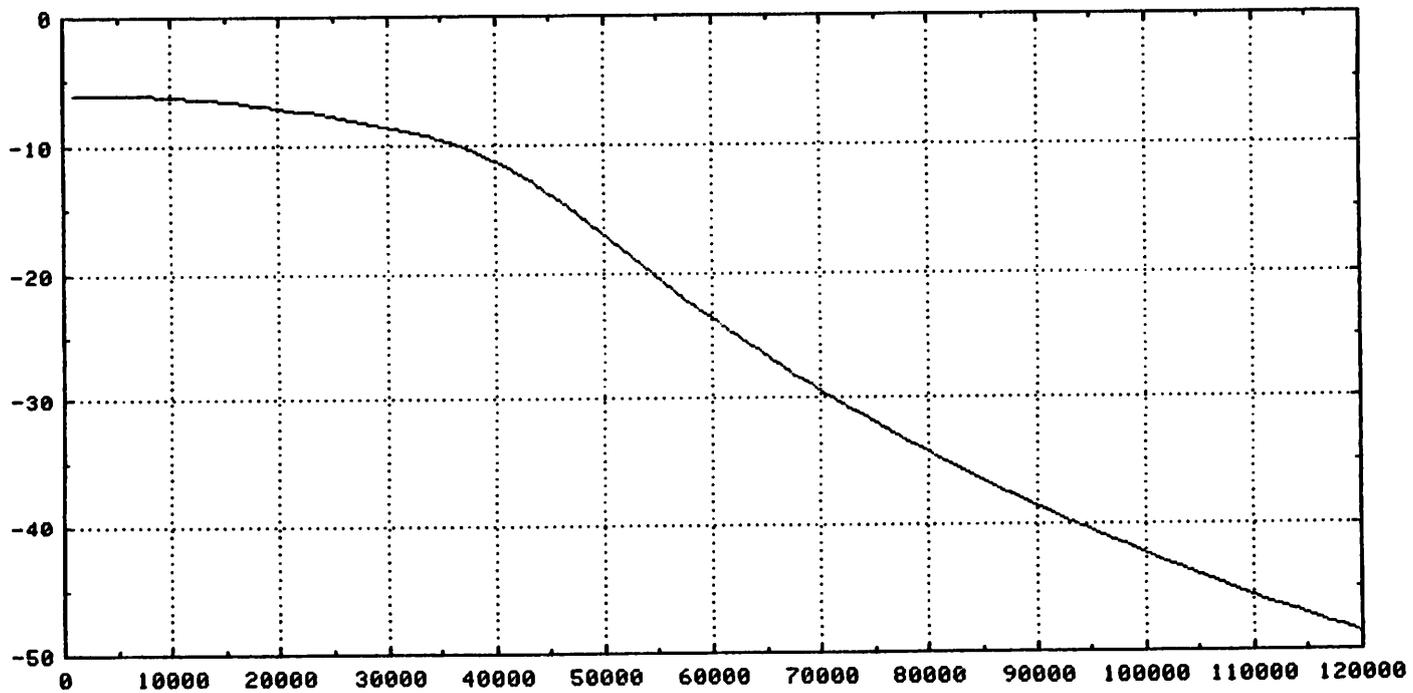


Fig. 5.11b. Amplitude response of receive filter.

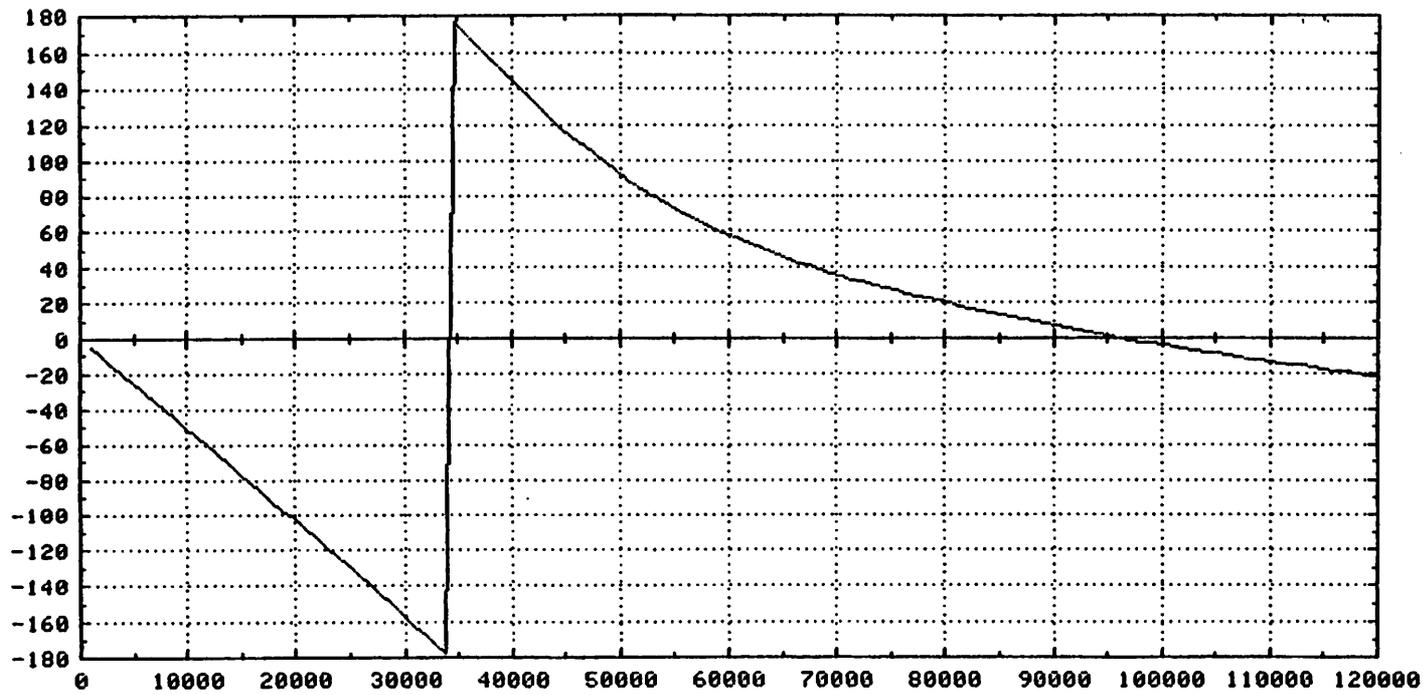


Fig. 5.11c. Phase response of receive filter.

APPENDIX A

MINIMAL INTERSYMBOL INTERFERENCE FILTERS

The design of minimal intersymbol interference filters is covered in references [49,52]. These references present a set of nonlinear equations to determine the pole locations of the desired filters. A program to solve the equations by a linearized Taylor expansion is also mentioned, but not explicitly given. Since designs other than those given in [51] are sometimes necessary, the programs shown below have been written.

The set of equations that we want to solve is [49]:

$$\prod_{j=1}^n \left(\frac{\sinh \frac{(p_i - p_j)}{2}}{\frac{(p_i - p_j)}{2}} \right) = \frac{2^{n-1} f(t_0)}{\prod_{j=1}^n (-p_j) \sinh \left(\sum_{j=1}^n p_j \right)} e^{-p_i t_0} \prod_{j=1}^n \sinh \left(\frac{p_i + p_j}{2} \right) \quad (\text{A.1})$$

$i = 1, 2, \dots, n$

$$E = \frac{1}{\left[e^{-\sum_{j=1}^n p_j} - 1 \right]}$$

where p_i are the poles, n is the order of the filter, E is the target residual intersymbol interference power, and $f(t_0)$ is the maximum of the impulse response $f(t)$ occurring at time t_0 .

An initial approximate solution is given by iteration of:

$$C_0 = \frac{3(2t_0 + n)^2}{2n}$$

$$C_{j+1} = C_j - \Delta E_j \frac{\sqrt{2C_j}}{n} \quad (\text{A.2})$$

$$\Delta E_j = \sum_{i=1}^n \sqrt{C_j + \sqrt{C_j^2 + (n+1-2i)^2 \pi^2}} - \sqrt{3n} (2t_0 + n)$$

which is computed by the subroutine APPROX. The iteration starts computing C_0 , and then for each j , ΔE_j and C_{j+1} are computed until a stable value of C_j is

obtained. Then the approximate poles are computed using the equation:

$$(p_i + 6 + \frac{12t_0}{n} \frac{S}{n})^2 = \frac{24C}{n} + j \frac{24\pi}{n} (N+1-2i) \quad (\text{A.3})$$

where

$$S = \sum_{j=1}^n p_j \quad (\text{A.4})$$

is computed using the last equation of (A.1) for a given value of E .

Starting with this initial solution, equations (A.1) are solved by Newton's method, by subroutine NEWTON, which calls subroutine INVERT to perform the matrix inversion. The program listing follows.

```

complex sum,p,const,s,sum1
common /s1/sum/s2/p(4),s/s4/const
e=0.0001
su=0.5*log(e/(2.+e))
sum=cplx(su,0.)
sum1=-sum
const=csinh(sum1)
call approx
call newton
print 1,(p(k),k=1,4)
1 format(1x,2f20.6)
stop
end

subroutine approx
common /s1/sum/s2/p(4),s
complex p,s,sum
n=4
s=1.
c=3*((n+(sum/6.))**2)/(3.141592*n)
50 continue
delta=0.
do 100 l=1,n
delta=delta+sqrt(c+sqrt((c**2)+float((n+1-2*l)**2)))
100 continue
delta=delta-(sqrt(6.*float(n)/3.141593))*(float(n)+sum/6.)
correc=delta*(sqrt(2.*c))/float(n)
c=c-correc
test=abs(correc)
if(test .le. 0.00001) go to 150
go to 50
150 continue
x=c

```

```

do 200 k=1,n
y=float(n)+1.-2.*float(k)
p(k)=-6.+csqrt((12.*3.141593/float(n))*cplx(x,y))
200 continue
return
end

```

```

subroutine newton
complex p1,p2,p3,p4,a,x,y,z,s
complex f
common /s2/p1,p2,p3,p4,s/s3/a(5,10)
dimension y(5),x(5),z(5)
x(1)=p1
x(2)=p2
x(3)=p3
x(4)=p4
x(5)=s
5 continue
do 10 i=1,5
y(i)=f(i,p1,p2,p3,p4,s)
a(i,1)=1000.*(f(i,p1+.001,p2,p3,p4,s)-y(i))
a(i,2)=1000.*(f(i,p1,p2+.001,p3,p4,s)-y(i))
a(i,3)=1000.*(f(i,p1,p2,p3+.001,p4,s)-y(i))
a(i,4)=1000.*(f(i,p1,p2,p3,p4+.001,s)-y(i))
a(i,5)=1000.*(f(i,p1,p2,p3,p4,s+.001)-y(i))
10 continue
call invert
error=0.
do 20 i=1,5
z(i)=0.
do 15 j=1,5
z(i)=z(i)+a(i,j)*y(j)
15 continue
x(i)=x(i)-z(i)
error=error+z(i)*conjg(z(i))
20 continue
error=sqrt(error)
p1=x(1)
p2=x(2)
p3=x(3)
p4=x(4)
s=x(5)
if(error .lt. 0.00001) return
go to 5
end

```

```

complex function f(k,p1,p2,p3,p4,s)
complex p1,p2,p3,p4,s,const,sum
complex csinh
common /s1/sum/s4/const
if(k .eq. 1) go to 1
if(k .eq. 2) go to 2
if(k .eq. 3) go to 3

```

```

if(k .eq. 4) go to 4
if(k .eq. 5) go to 5

1  f=2.*s*(csinh(p1)/(p1*const))*((p1**2-p2**2)/(p2**2))*
1  ((p1**2-p3**2)/(p3**2))*((p1**2-p4**2)/(p4**2))*
2  {csinh((p1+p2)/2.)/csinh((p1-p2)/2.)}*
3  {csinh((p1+p3)/2.)/csinh((p1-p3)/2.)}*
4  {csinh((p1+p4)/2.)/csinh((p1-p4)/2.)}-1.
   return
2  f=2.*s*(csinh(p2)/(p2*const))*((p2**2-p3**2)/(p3**2))*
1  ((p2**2-p4**2)/(p4**2))*((p2**2-p1**2)/(p1**2))*
2  {csinh((p2+p3)/2.)/csinh((p2-p3)/2.)}*
3  {csinh((p2+p4)/2.)/csinh((p2-p4)/2.)}*
4  {csinh((p2+p1)/2.)/csinh((p2-p1)/2.)}-1.
   return
3  f=2.*s*(csinh(p3)/(p3*const))*((p3**2-p4**2)/(p4**2))*
1  ((p3**2-p1**2)/(p1**2))*((p3**2-p2**2)/(p2**2))*
2  {csinh((p3+p4)/2.)/csinh((p3-p4)/2.)}*
3  {csinh((p3+p1)/2.)/csinh((p3-p1)/2.)}*
4  {csinh((p3+p2)/2.)/csinh((p3-p2)/2.)}-1.
   return
4  f=2.*s*(csinh(p4)/(p4*const))*((p4**2-p1**2)/(p1**2))*
1  ((p4**2-p2**2)/(p2**2))*((p4**2-p3**2)/(p3**2))*
2  {csinh((p4+p1)/2.)/csinh((p4-p1)/2.)}*
3  {csinh((p4+p2)/2.)/csinh((p4-p2)/2.)}*
4  {csinh((p4+p3)/2.)/csinh((p4-p3)/2.)}-1.
   return
5  f=p1+p2+p3+p4-sum
   return
end
complex function csinh(x)
complex x
csinh=(cexp(x)-cexp(-x))/2.
return
end
subroutine invert
common /s3/a(5,10)
complex a,ctemp
do 8 i=1,5
do 7 j=6,10
a(i,j)=0.
7 continue
8 continue
do 9 k=1,5
a(k,k+5)=1.
9 continue
do 50 j=1,5
do 20 i=j,5
temp=a(i,j)*conjg(a(i,j))
temp=sqrt(temp)
k=i
if(temp .gt. 0.000001) go to 30
20 continue
call abort(sing)
30 continue

```

```
do 100 l=1,10
  ctemp=a(j,l)
  a(j,l)=a(k,l)
  a(k,l)=ctemp
100 continue
  do 120 l=10,j,-1
    a(j,l)=a(j,l)/a(j,j)
120 continue
    if(j .eq. 5) go to 150
    do 140 l=j+1,5
      do 130 ll=10,j,-1
        a(l,ll)=a(l,ll)-a(j,ll)*a(l,j)
130 continue
140 continue
150 continue
    if(j .eq. 1) go to 49
    do 300 l=1,j-1
      do 290 ll=10,j,-1
        a(l,ll)=a(l,ll)-a(j,ll)*a(l,j)
290 continue
300 continue
49 continue
50 continue
    do 450 i=1,5
      do 400 j=1,5
        a(i,j)=a(i,j+5)
400 continue
450 continue
return
end
```

APPENDIX B

EFFECT OF PULSE ASYMMETRY ON THE SPECTRUM
OF A BIPOLAR SIGNAL

This Appendix will establish that the components of the transmitted data signal at multiples of the bit rate is a direct and useful measure of transmitted pulse asymmetry. As is well known, the bipolar encoded signal has zeros in the power spectrum at all harmonics of the bit rate. This Appendix will show that in the presence of pulse asymmetry the transmitted signal in fact has a line component at the bit frequency.

Assume that the transmitted data bits assume the values $x_n = +1$ and $x_n = -1$, that the difference of successive data values are used to modulate a train of pulses, and that there is a asymmetry in the pulses such that a positive pulse has shape $h_+(t)$ and negative pulse has shape $h_-(t)$. Then the transmitted signal can be represented as

$$s(t) = \sum_k (f_+(x_{k-1}, x_k)h_+(t-kT) + f_-(x_{k-1}, x_k)h_-(t-kT)) \quad (\text{B.1})$$

where f_+ is 1 when $x_k = 1, x_{k-1} = -1$, and 0 otherwise, and similarly f_- is 1 only when $x_k = -1, x_{k-1} = 1$. Since f_+ and f_- are nonlinear functions of x_k and x_{k-1} , the expansion of (3.1) is valid and it is simple to show that

$$\begin{aligned} f_+(x_{k-1}, x_k) &= \frac{1 + x_k - x_{k-1} - x_k x_{k-1}}{4} \\ f_-(x_{k-1}, x_k) &= \frac{1 - x_k + x_{k-1} - x_k x_{k-1}}{4} \end{aligned} \quad (\text{B.2})$$

Thus, (B.1) becomes

$$\begin{aligned} s(t) &= \frac{1}{4} \sum_k (1 - x_k x_{k-1})(h_+(t-kT) - h_-(t-kT)) + \\ &\quad \frac{1}{4} \sum_k (x_k - x_{k-1})(h_+(t-kT) + h_-(t-kT)) \end{aligned} \quad (\text{B.3})$$

which demonstrates the nature of the nonlinearity introduced when there is

pulse asymmetry: nonlinear distortion in the form of second order products of adjacent bits is introduced into the transmitted signal. This demonstrates the terms which would have to be added to the nonlinear canceller to compensate for pulse asymmetry.

Assume that the data bits x_k are independent and assume the value +1 with probability p . Then the average value of the transmitted signal is easily determined from (B.3) to be

$$E[s(t)] = p(1-p) \sum_k (h_+(t-kT) - h_-(t-kT)) . \quad (\text{B.4})$$

The total signal can then be written as a zero-mean random component plus the deterministic component of (B.4). The random component will have a continuous power spectrum, while the deterministic component consists of a line spectrum since it is periodic in T sec. Expanding (B.4) in a Fourier series,

$$E[s(t)] = \frac{p(1-p)}{T} \sum_n (H_+(n \frac{2\pi}{T}) - H_-(n \frac{2\pi}{T})) e^{jn \frac{2\pi}{T}} . \quad (\text{B.5})$$

Thus, the transmitted signal contains line components at multiples of the bit rate due to the pulse asymmetry. Taking into account the effect of the transmit filter, the only significant component will be that at the bit rate, which is proportional to the difference of the Fourier transforms of the positive and negative pulses at that frequency. As expected, this undesired component is maximum when the data is equally likely, and goes away when the data is always one polarity since the transmitted signal is zero in this case. Of course, it also goes away when the positive and negative pulses are equal, or even if they only have equal Fourier transforms at the bit frequency.

APPENDIX C

NMOS SILICON GATE PROCESS

I. WAFER CLEANING

1. TCE 60° C 10 min.
2. Acetone room temp. 2 min.
3. DI H_2O rinse, N_2 blow dry.
4. Piranha ($H_2O : H_2SO_4 / 1:5$) 15 min.
5. DI, N_2
6. Dip in $HF : H_2O / 1 : 5$ 30 sec.
9. DI, N_2

II. GATE OXIDE (TEMP = 1000° C)

1. Push O_2 (dry) 6.5cm 3 min.
2. Oxidize O_2 (dry) 6.5cm 55 min. (for 850 A oxide)
3. Anneal N_2 4.0cm 15 min.
4. Pull N_2 4.0cm 3 min.

III. ACTIVE AREA

1. Deposit 1000 A Si_3N_4
2. Active area mask
 - i. HMDS 3 min with small amount of vapor
 - ii. N_2 purge 5 min with larger amount of N_2 flow
 - iii. Spin Photoresist 6000rpm 30 sec.
 - iv. 90° prebake 15 min.
 - v. Align and expose
 - vi. Develop in AZ developer : $H_2O / 1 : 1$ 60 sec.
 - vii. DI, N_2
 - viii. Inspect under IR
 - ix. 110° C postbake 15 min.
3. Dip in $HF : H_2O / 1 : 10$ 10 sec, DI, N_2 .
4. Plasma etch nitride
5. Field implant Boron, 120 KeV, $1.5 \times 10^{13} cm^{-2}$
6. Strip Photoresist in acetone 5 min.; DI, N_2

IV. LOCOS

1. Piranha 8 min.
2. DI, N_2
3. Push O_2 (dry) 6.5 cm 3 min 1000° C.
4. O_2 (dry) 6.5 cm 20 min 1000° C.
5. Stand by N_2 4.0 cm
6. Wet O_2 2.0 cm 360 min. 900° C.
7. Anneal N_2 4.0 cm 15 min. 900° C.
8. Pull N_2 4.0 cm 3 min. 900° C.

V. BOTTOM PLATE

1. Bottom plate mask (same as in III.2)
2. Dip in $HF : H_2O / 1 : 10$ 30 sec.
3. DI, N_2 .
4. Plasma etch Si_3N_4 .
5. Bottom plate implant Phos 50 KeV $8 \times 10^{14} \text{ cm}^{-2}$
6. Plasma etch photoresist.
7. Piranha 5 min.
8. DI, N_2 .
9. Re-gate (1000° C)
 - i. Push O_2 (dry) 6.5 cm 3 min
 - ii. Oxidize O_2 (dry) 6.5 cm 55 min.
 - iii. Anneal N_2 4.0 cm 15 min.
 - iv. Pull N_2 4.0 cm 3 min.

VI. THRESHOLD ADJUSTMENT

1. Oxide dip in $HF : H_2O / 1 : 10$ 20 sec.
2. DI, N_2 .
3. Strip Si_3N_4 in H_3PO_4 at 165° C
30 min. (add drops of water during etching to keep temperature constant).
4. DI, N_2 .
5. V_{TD} mask (same as in III.2)
6. V_{TD} implant Phos. 120 KeV 1.4×10^{12}
(do noise compensation in implant).
7. Strip Photoresist in acetone 5 minutes.
8. DI, N_2 .
9. V_{TE} implant Boron 80 KeV $6.0 \times 10^{11} \text{ cm}^{-2}$.
(do noise compensation).
10. Piranha 5 min.
11. DI, N_2 .

VII. GATE DEFINE

1. Oxide dip in $HF : H_2O / 1 : 10$ 5~10 sec.
2. DI, N_2
3. Bake under IR at least 20 min.
3. Poly deposition (5000 A).
4. Poly dope in N-predep furnace.
 - i. Push N_2 5.0 cm 3 min.
 - ii. N_2 / O_2 5.0 cm / 2.5 cm 5 min.
 - iii. Dope $N_2 / O_2 / POCl_3$
5.0 cm / 2.5 cm 6.0 cm. 30 min.
 - iv. N_2 / O_2 5.0 cm / 2.5 cm 2 min.
 - v. N_2 5.0 cm 5 min.
 - vi. Pull N_2 5.0 cm. 3 min.
5. Oxide dip in $HF : H_2O / 1 : 5$ 30 sec.
6. DI, N_2 .
7. Bake under IR 5 min.
8. Gate mask
9. Plasma etch poly
10. Etch backside oxide completely with BHF.
11. DI, N_2 .

12. Strip photoresist in acetone 5 min.
13. *DI, N₂*
14. Piranha 5 min.
15. *DI, N₂* IR 10 min.

VIII. SOURCE AND DRAIN DEFINITION

1. S/D implant As 180 KeV $5 \times 10^{15} \text{ cm}^{-2}$
2. Backside implant BF_2 150 KeV $1 \times 10^{15} \text{ cm}^{-2}$.
3. Piranha 5 min.
4. *DI, N₂*.
5. Drive-in and anneal (1000° C).
 - i. Push O_2 dry 8.5 cm 3 min.
 - ii. Drive-in O_2 (dry) 6.5 cm 30 min.
 - iii. Anneal N_2 4.0 cm 30 min.
 - iv. Pull N_2 4.0 cm 3 min.

IX. SPIN-ON GLASS DEPOSITION

1. Spin glass 3000 rmp, 30 sec.
2. Cure at 900° C for 20 min.
3. Repeat 1 and 2 to grow 7200 Å
(6 applications are required).

X. CONTACT AND METALLIZATION

1. Contact mask
2. Etch contact oxide in BHF
3. *DI, N₂*, *ir* at least 15 min.
4. Deposit 1000 Å of poly
5. Evaporate Aluminum (8000 Å)
6. Metal mask
7. Etch Al in Al etchant type A
8. Plasma etch polysilicon under metal.
9. *DI, N₂*, *IR* 10 min.
10. Etch back side oxide with BHF.
11. *DI, N₂*.
12. Strip photoresist in acetone 5 min.
13. *DI, N₂*, *IR* 15 min.
14. Evaporate Al on the back side $\sim 1 \mu\text{m}$.
15. Sinter Al in sintering furnace at 300° C in forming gas 14 cm 5 min.
16. Test

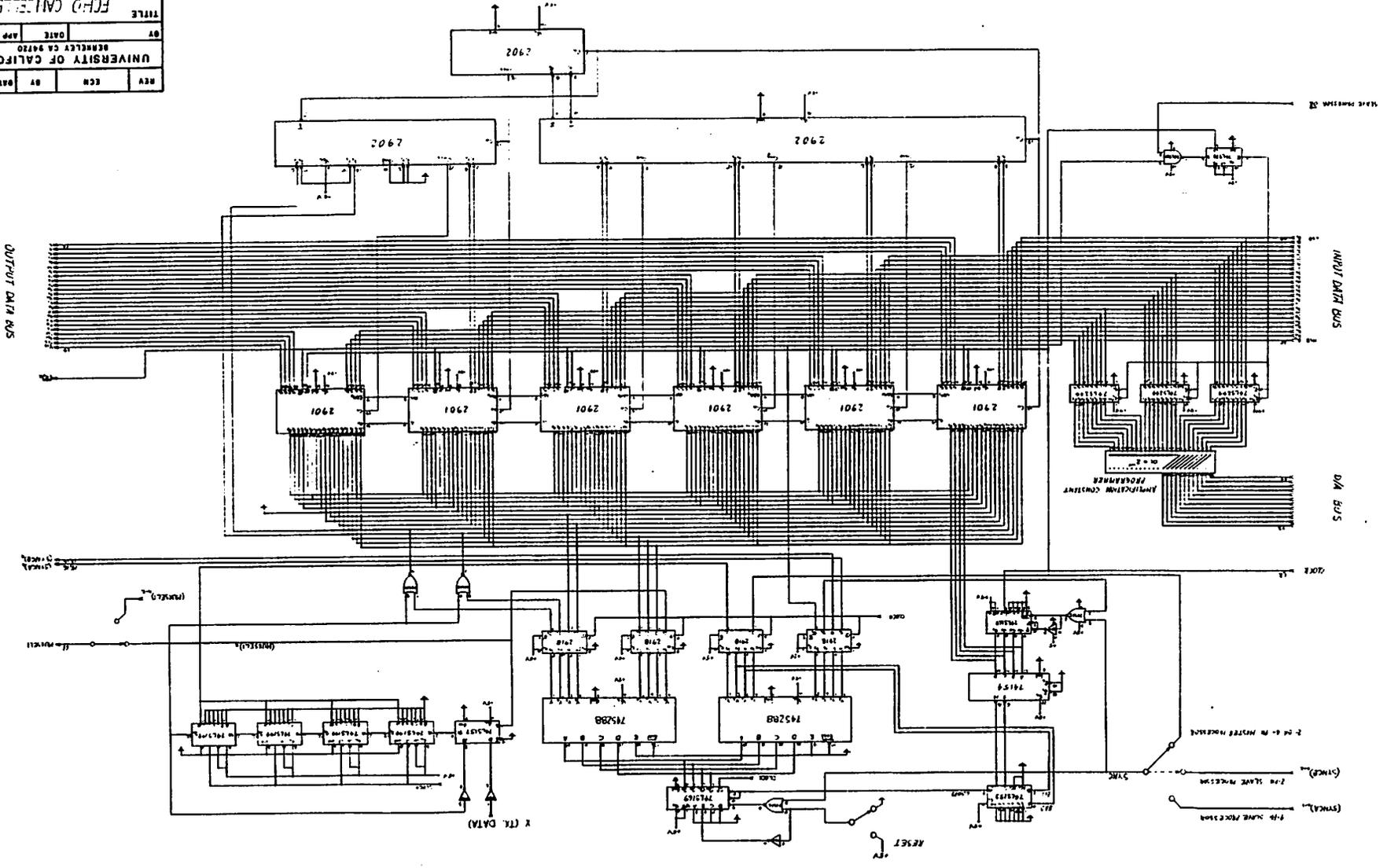
APPENDIX D

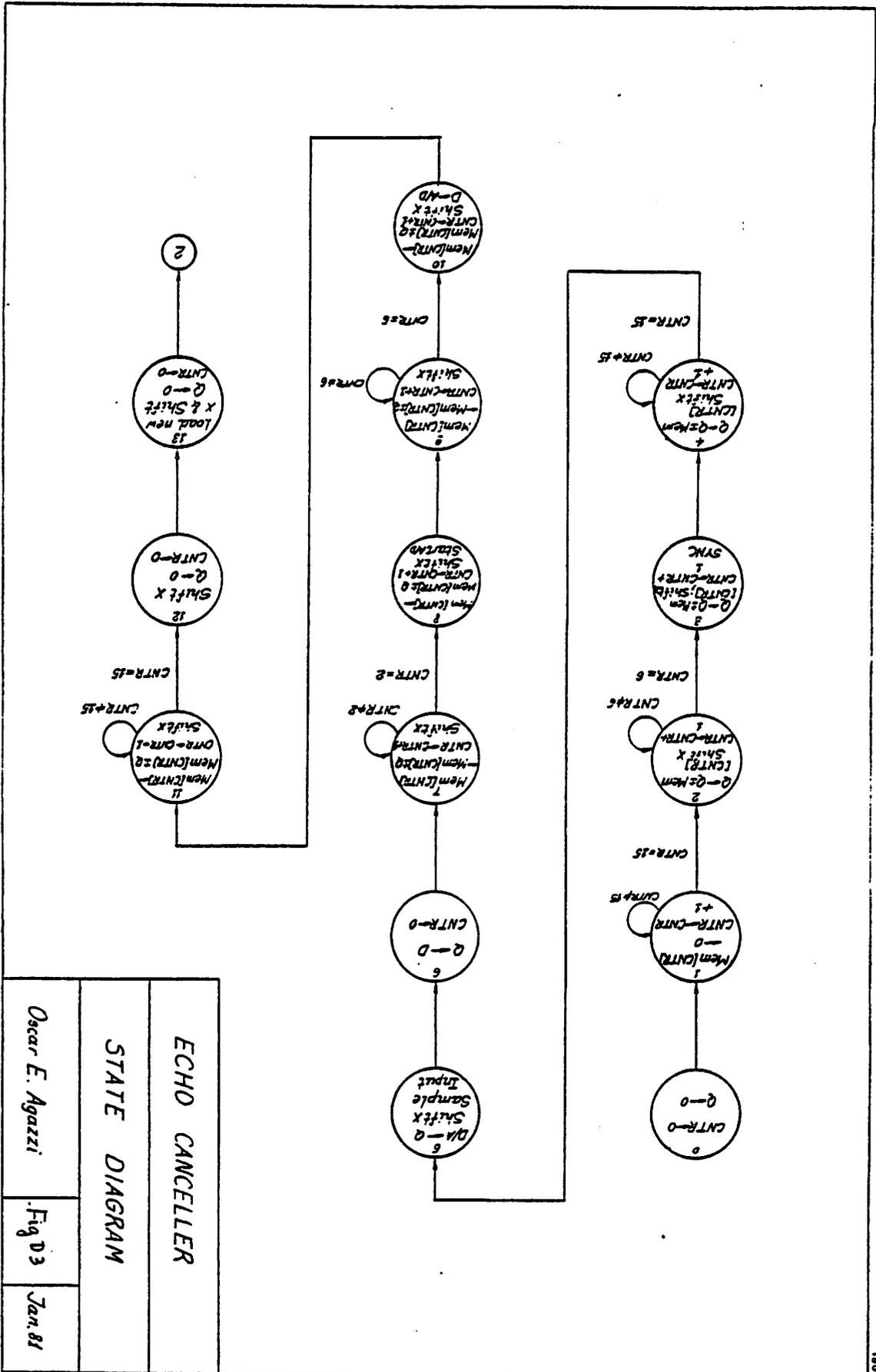
DIGITAL PROCESSOR DOCUMENTATION

This Appendix summarizes the documentation on the digital processors implementing the echo canceller in both versions described in Chapter 4. Also given is the design of the analog interfacing circuits. Fig. D1 shows the digital processor in the fully digital implementation. Fig. D2 is the associated analog interface. Fig. D3 is the state diagram in the same implementation. Fig. D4 shows the microprogram table. Finally, Fig. D5 corresponds to the digital adaptation processor in the analog/digital implementation, and Fig. D6 to its associated analog interface.

REV	ECH	BT	DATE	APP
REV	ECH	BT	DATE	APP
UNIVERSITY OF CALIFORNIA BERKELEY CA 94720				
DATE				
TITLE				
ECHO CHANNEL PROCESSOR				
MODEL NO. D-1				

NOTE: TWO PROCESSORS HAVE BEEN WIRED PER BOARD.
 2. SINGLE BOARD PROCESSOR (ECHO CHANNEL) WITH ASSOCIATED SUPPORTS AND LOGIC IS USED
 IN CONNECTION WITH ALL MODELS.





ECHO CANCELLER

STATE DIAGRAM

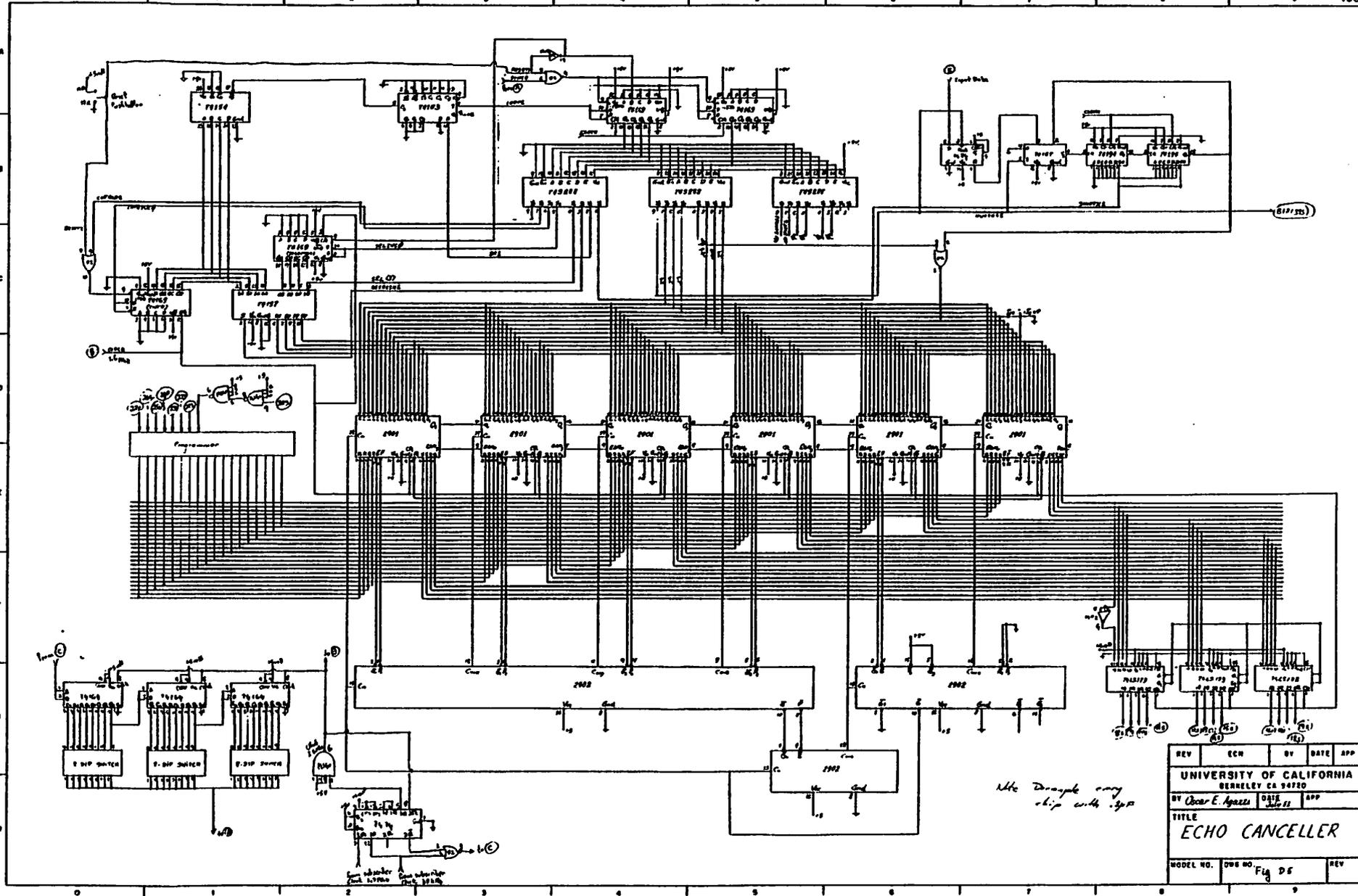
Oscar E. Agazzi

Fig 03

Jan. 81

TABLE II - MICROPROGRAM (Fig D4)

STATE	SYNCA1	SYNCA2	CUTR1	DATA LENGTH	SYNCA*	BR1	BR2	SWTR1	MUXR1	I ₀	I ₁	I ₂	I ₃	I ₄	I ₅	I ₆	I ₇	I ₈	I ₉	ACTION	
0 16	0	0	1	1	0	0	0	0	0	0	0	1	1	0	1	1					R=0; S=A; Q←RAS; K←RAS
1 17	0	0	0	1	0	1	0	0	0	0	0	1	1	1	1	1					R=0; S=A; B←F; F←RAS
2 18	0	0	0	1	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	1	R←A; S←Q; Q←F; F←S±R
3 19	1	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	R←A; S←Q; Q←F; F←S±R
4 20	0	0	0	1	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	1	R←A; S←Q; Q←F; F←S±R
5 21	0	0	1	1	0	0	0	0	0	0	1	0	0	0	1	1					R=0; S←Q; Q←F; F←S±R
6 22	0	0	1	1	0	0	0	0	0	0	1	0	0	0	1	1					R=0; S←Q; Q←F; F←R+S
7 23	0	1	1	1	0	0	0	0	0	0	1	1	0	0	1	1					R←D; S←Q; Q←F; F←RAS
8 24	0	0	1	0	0	0	0	0	0	0	1	0	0	0	1	1					R=0; S←Q; Q←F; F←R+S
9 25	0	0	1	1	0	0	0	0	0	1	1	1	0	0	1	1					R←D; S←0; Q←F; F←R+S
10 26	0	0	0	1	0	1	0	1	0	0	0	0	0	1	1	0					R←A; S←Q; B←F; F←R+S
11 27	0	0	1	1	0	0	0	0	0	0	0	1	1	0	1	1					R=0; S←A; Q←F; F←RAS
12 28	0	0	1	1	1	0	0	0	0	0	0	1	1	0	1	1					R=0; S←A; Q←F; F←RAS
13 29	0	0	1	1	0	0	0	1	1	0	0	1	1	0	1	1					R=0; S←A; Q←F; F←RAS



REV	ECN	BY	DATE	APP
UNIVERSITY OF CALIFORNIA BERKELEY CA 94720				
BY Oscar E. Agazzi		DATE	APP	
		3/16/51		
TITLE				
ECHO CANCELLER				
MODEL NO.	DWG NO.	FILE	DES	REV

Note: Don't change any chip with .3pF

See schematic for notes

REFERENCES

- [1] R. W. Lucky, J. Saltz, and E. J. Weldon, Jr., "*Principles of Data Communication*", Mc Graw Hill, 1968.
- [2] W. R. Bennet and J. R. Davey, "*Data Transmission*", Mc Graw Hill, 1965.
- [3] A. P. Clark, "*Advanced Data Transmission Systems*", John Wiley, 1977.
- [4] W. Kaplan, "*Advanced Calculus*", Addison Wesley, 1952.
- [5] B. Widrow, "Adaptive Filters, I: Fundamentals", *Stanford Electronics Lab. Tech. Rep. 6764-6*, December 1966.
- [6] M. M. Sondhi and A. J. Presti, "A Self-Adaptive Echo Canceller", *Bell System Technical Journal*, Vol. 45, No. 10, December 1966, pp. 1851-1854.
- [7] M. M. Sondhi, "An Adaptive Echo Canceller", *Bell System Technical Journal*, Vol. 46, No. 3, March 1967, pp. 497-511.
- [8] S. J. Campanella, H. G. Suyderhoud, and M. Onufry, "Analysis of an Adaptive Impulse Response Echo Canceller", *COMSAT Technical Review*, Vol. 2, pp. 1-38, Spring 1972.
- [9] N. Demytko and L. K. Mackechnle, "A High-Speed Digital Adaptive Echo Canceller", *Austral. Telecommunications Research*, Vol. 7, pp. 21-29, 1973.
- [10] H. Hoge, "Analysis of Adaptive Echo Canceller with Optimized Gradient Gain", *Siemens F. U. E.*, Vol. 4, pp. 127-131, 1975.
- [11] N. Demytko and K. S. English, "Echo Cancellation on Time-Variant Circuits", *Proceedings of the IEEE*, Vol. 65, No. 3, March 1977, pp. 444-453.
- [12] D. L. Duttweiler, "A Twelve-Channel Digital Echo Canceller", *IEEE Transactions on Communications*, Vol. COM-26, No. 5, May 1978, pp. 847-53.
- [13] D. L. Duttweiler and Y. S. Chen, "A Single-Chip VLSI Echo Canceller", *Bell System Technical Journal*, Vol. 59, No. 2, February 1980, pp. 149-160.

- [14] L. M. Manhire, "Physical and Transmission Characteristics of the Customer Loop Plant", *Bell System Technical Journal*, Vol. 57, No. 1, January 1978, pp. 35-59.
- [15] D. H. Morgen, "Expected Crosstalk Performance of Analog Multichannel Subscriber Carrier Systems", *IEEE Transactions on Communications*, Vol. COM-23, pp. 240-245, February 1975.
- [16] R. K. Even, R. A. Mc Donald, and H. Seidel, "Digital Transmission Capability of the Loop Plant", *Proc. Int. Commun. Conf.*, 1979.
- [17] M. G. Vry and P. J. van Gerwen, "The Design of a 1+1 System for Digital Signal Transmission to the Subscriber", *Proceedings of the International Symposium on Subscriber Loops and Systems*, Zurich, 1980.
- [18] M. G. Vry and P. G. van Gerwen, "Digital Signal Transmission to the Subscriber Using a 1+1 System", *Proceedings of the 2nd International Conference on Telecommunications and Transmission*.
- [19] S. V. Ahamed, P. P. Bohn, and N. L. Gottfried, "A Tutorial on Two-Wire Digital Transmission in the Loop Plant", *IEEE Transactions on Communications*, Vol. COM-29, No. 11, November 1981.
- [20] K. H. Mueller, "A New Digital Echo Canceller for Two-Wire Full-Duplex Data Transmission", *IEEE Transactions on Communications*, Vol. COM-24, No. 9, September 1976, pp. 956-962.
- [21] S. B. Weinstein, "A Passband Data Driven Echo Canceller for Full-Duplex Transmission on Two-Wire Circuits", *IEEE Transactions on Communications*, Vol. COM-25, No. 7, July 1977, pp. 654-666.
- [22] K. H. Mueller, "Combining Echo Cancellation and Decision Feedback Equalization", *Bell System Technical Journal*, Vol. 58, No. 2, February 1979, pp. 491-500.

- [23] D. D. Falconer and K. H. Mueller, "Adaptive Echo Cancellation/AGC Structures for Two-Wire Full-Duplex Data Transmission", *Bell System Technical Journal*, Vol. 58, No. 7, September 1979, pp. 1593-1616.
- [24] N. A. M. Verhoeckx, H. C. van den Elzen, F. A. M. Snijders and P. G. van Gerwen, "Digital Echo Cancellation for Baseband Data Transmission", *IEEE Transactions on Acoustics, Speech, and Signal Processing*, Vol. ASSP-27, No. 6, December 1979.
- [25] N. Holte and S. Stueflotten, "A New Digital Echo Canceller for Two-Wire Subscriber Lines", *IEEE Transactions on Communications*, Vol. COM-29, No. 11, November 1981, pp. 1573-1581.
- [26] Y. Takasaki, "Timing Extraction in Baseband Pulse Transmission", *IEEE Transactions on Communications*, Vol. COM-20, No. 5, October 1972, pp. 877-884.
- [27] L. E. Franks and J. P. Bobrowski, "Statistical Properties of Timing Jitter in a PAM Timing Recovery Scheme", *IEEE Transactions on Communications*, Vol. COM-22, No. 7, July 1974.
- [28] R. D. Gintlin and J. F. Hayes, "Timing Recovery and Scramblers in Data Transmission", *Bell System Technical Journal*, Vol. 54, No. 3, March 1975, pp. 569-593.
- [29] D. L. Duttweiler, "The Jitter Performance of Phase Locked Loops Extracting Timing From Baseband Data Waveforms", *Bell System Technical Journal*, Vol. 55, No. 1, January 1976, pp. 37-58.
- [30] J. E. Mazo, "Jitter Comparison of Tones Generated by Squaring and Fourth Power Circuits", *Bell System Technical Journal*, Vol. 57, No. 5, May-June 1978, pp. 1489-1498.

- [31] J. E. Savage, "Some Simple Self-Synchronizing Digital Data Scramblers", *Bell System Technical Journal*, Vol. 46, No. 2, February 1967, pp. 449-487.
- [32] CCITT Recommendation V-35, "Data Transmission at 48 Kilobits per Second Using 60-108 Khz Group Band Circuits", Mar del Plata, 1968.
- [33] A. Brosio, U. de Julio, V. Lazzari, R. Ravaglia, and A. Tofanelli, "A Comparison of Digital Subscriber Line Transmission Systems Employing Different Line Codes", *IEEE Transactions on Communications*, Vol. COM-29, No. 11, November 1981, pp. 1581-1588.
- [34] H. Kobayashi, "A Survey of Coding Schemes for Transmission or Recording of Digital Data", *IEEE Transactions on Communications*, Vol. COM-19, December 1971, pp. 1087.
- [35] R. A. Tarbox, "A Regenerative Repeater Utilizing Hybrid Integrated Circuit Technology", *Proc. Int. Conf. on Communications*, 1969, paper No. 69CP403-COM.
- [36] Y. Takasaki, "Simple Inductorless Automatic Line Equalizer for PCM Transmission Using New Variable Transfer Function", *IEEE Transactions on Communications*, Vol. COM-26, No. 5, May 1978.
- [37] R. W. Lucky, "Automatic Equalization for Digital Communications", *Bell System Technical Journal*, Vol. 44, No. 4, April 1965, pp. 547.
- [38] R. W. Lucky, "Techniques for Adaptive Equalization of Digital Communications Systems", *Bell System Technical Journal*, February 1966.
- [39] D. L. Lyon, "Timing Recovery in Synchronous Equalized Data Communication", *IEEE Transactions on Communications*, February 1975.
- [40] G. Ungerboeck, "Fractional Tap-Spacing Equalizer and Consequences for Clock Recovery in Data Modems", *IEEE Transactions on Communications*, Vol. COM-24, No. 8, August 1976, pp. 856-864.

- [41] R. D. Gintlin and S. B. Weinstein, "Fractionally Spaced Equalization: An Improved Digital Transversal Equalizer", *Bell System Technical Journal*, Vol. 60, No. 2, February 1981, pp. 275-296.
- [42] R. D. Gintlin and S. B. Weinstein, "On the Required Tap-Weight Precision for Digitally Implemented Adaptive Mean-Square Equalizers", *Bell System Technical Journal*, Vol. 58, No. 2, February 1979, pp. 301-321.
- [43] J. Saltz, "Optimum Mean-Square Decision Feedback Equalization", *Bell System Technical Journal*, Vol. 52, No. 8, October 1973, pp. 1341-1372.
- [44] R. J. Keeler, "Construction and Evaluation of a Decision Feedback Equalizer", *Proc. IEEE Int. Comm. Conf.*, Montreal, Canada, June 14-16, 1971.
- [45] P. Monsen, "Feedback Equalization for Fading Dispersive Channels", *IEEE Transactions on Information Theory*, Vol. IT-17, January 1971.
- [46] D. D. Falconer, "Jointly Adaptive Equalization and Carrier Recovery in Two-Dimensional Digital Communications Systems", *Bell System Technical Journal*, Vol. 55, No. 3, March 1976, pp. 317-334.
- [47] D. D. Falconer, "Analysis of a Gradient Algorithm for Simultaneous Passband Equalization and Carrier Phase Recovery", *Bell System Technical Journal*, Vol. 55, No. 4, April 1976, pp. 409-428.
- [48] D. A. Spaulding, "Synthesis of Pulse Shaping Networks in the Time Domain", *Bell System Technical Journal*, September 1969.
- [49] L. F. Lind and S. E. Nader, "Optimal Analogue Filters for Data Transmission", *Proc. IEEE International Symp. on Circuits and Systems*, Phoenix, Arizona, 1977, pp. 334-347.
- [50] L. F. Lind and S. E. Nader, "Realisable Matched Filters for Minimal Intersymbol Interference", *Proc. IEEE International Symp. on Circuits and Systems*, New York, May 1978, pp. 441-445.

- [51] L. F. Lind and S. E. Nader, "Design Tables for a Class of Data Transmission Filters", *Electronics Letters*, Vol. 13, No. 19, 15 September 1977.
- [52] S. E. Nader and L. F. Lind, "Optimal Data Transmission Filters", *IEEE Transactions on Circuits and Systems*, Vol. CAS-26, No. 1, January 1979, pp. 36-45.
- [53] W. K. Pratt, J. Kane, and H. C. Andrews, "Hadamard Transform Image Coding", *Proceedings of the IEEE*, Vol. 57, No. 1, January 1969.
- [54] M. Schetzen, "Nonlinear System Modeling Based on the Wiener Theory", *Proceedings of the IEEE*, Vol. 69, December 1981, pp. 1557.
- [55] J. L. Mc Creary and P. R. Gray, "All-MOS Charge Redistribution Analog-to-Digital Conversion Techniques-Part I", *IEEE Journal of Solid-State Circuits*, Vol. SC-10, No. 6, December 1975.
- [56] B. Fotouhi and D. A. Hodges, "High Resolution A/D Conversion in MOS/LSI", *IEEE Journal of Solid-State Circuits*, Vol. SC-14, No. 6, December 1979, pp. 920-926.
- [57] W. C. Black, Jr., "High Speed CMOS A/D Conversion Techniques", *Ph.D. Dissertation*, University of California, Berkeley, November 1980.
- [58] D. Senderowicz, D. A. Hodges, and P. R. Gray, "An Integrated Vector Locked Loop", *Proc. IEEE International Symposium on Circuits and Systems*, Rome, May 1982.

M82/41

4
★

Received: by janus.Berkeley.EDU (5.57/1.34)
id AA24147; Fri, 1 Dec 89 08:24:56 PST
Date: Fri, 1 Dec 89 08:24:56 PST
From: pgray (Paul Gray)
Message-Id: <8912011624.AA24147@janus.Berkeley.EDU>
To: flora
Subject: urgent mailing
Status: RO

Could you please get a copy of Oscar Agazzi's PhD dissertation, circa 1985, from Jeff in ERL and FEDEX it to:

~~D. G. Ganeshan Analog Devices 804 Woburn Street Wilmington, Mass 01887-3462~~

~~Chg to 21350. Thnx much PRG~~

check address . put in 558 box
c/o Agnes

EA30

D.G. Ganeshan Analog Devices
181 Ballardvale
Bloomington, Mass 01810

002106779 # to be charge.