To the Graduate Council:

I am submitting herewith a thesis written by Naeem Uddin Syed entitled "Minicomputer Controlled Fast Expansion of Speech Signals for Improving Intelligibility." I have examined the final copy of this thesis for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Master of Science, with a major in Electrical Engineering.

Donald W. Bouldin

D. W. Bouldin, Major Professor

We have read this thesis and recommend its acceptance:

Accepted for the Council:

L. Evans Roth

Vice Chancellor
Graduate Studies and Research

MINICOMPUTER CONTROLLED FAST EXPANSION OF SPEECH SIGNALS FOR IMPROVING INTELLIGIBILITY

A Thesis

Presented for the

Master of Science

Degree

The University of Tennessee, Knoxville

Naeem Uddin Syed

March 1983

3066047

ACKNOWLEDGMENTS

I wish to express my deepest gratitude to Dr. Donald Wayne Bouldin who had been really instrumental and helpful in the preparation and polishing of the thesis. Special thanks go to Dr. Igor V. Nabelek for his continued assistance and advice during each phase of this thesis. He suffered through innumerable questions, and still maintained a smile. I also appreciate Drs. Robert W. Rochelle and Robert E. Bodenheimer serving on my committee and their valuable advice. Thanks is also due to Dr. William E. Blass for his help in interfacing and programming.

I also feel greatly indebted to the Audiology and Speech Pathology Department of The University of Tennessee, Knoxville, (UTK) for providing the necessary cooperation, facilities and equipment.

My gratitude to all my friends who, though they may not have been directly involved, still provided support.

Financial support provided to me as a Teaching Assistant by Electrical Engineering Department of The University of Tennessee, Knoxville, is gratefully acknowledged.

I especially want to recognize the encouragement of my parents, Mr. and Mrs. Qutub U. Syed, and my sisters Husna and Ishrat.

Ultimate thanks go to God Almighty who made it all possible.

ABSTRACT

A new system for speech enhancement named "fast expansion," has been designed, developed and demonstrated. It removes reverberating tails and low amplitude noise from speech. The system developed has two channels under minicomputer control and can be expanded to more channels. The minicomputer allows great flexibility, and threshold level can be adjusted independently for each channel. The completed system gives the expected results, and this hopefully will open a new venue for research in speech enhancement.

TABLE OF CONTENTS

CHAPTER		PAGE
I.	INTRODUCTION	1
II.	REVIEW OF DEVELOPMENTS IN SPEECH PROCESSING	2
	Eras of Advancement	2
	Techniques for Measuring and Improving Intelligibility	5
III.	SPEECH PROCESSING TECHNIQUES	11
	Fast Expansion	11
	Center Clipping	21
	Squelch	23
IV.	MINICOMPUTER-BASED SPEECH PROCESSING SYSTEM	25
	Structure of the Experimental System	25
	Sampling of Audio Signals	27
	Input/Output Module	29
	Computer	30
	Software	30
	Fast Expansion	32
	Input	34
	Get	34
	Center Clipping	35
	Peak Clipping	35
V.	RESULTS	36
	Experiments with Pure Tones	36
	Experiments with Speech	37
	Results with Spectrum Analysis	40
VI.	SUMMARY AND CONCLUSIONS	48
LIST OF REFERENCES		51
APPENDICES		53
	Appendix A	54
	Appendix B	58
	Appendix C	75
VITA		81

LIST OF FIGURES

FIGURE		PAGE
1.	Syllable Articulation for High-Pass and Low-Pass System Operating in Quite with Optimal Gain	7
2.	Articulation Index as a Function of Frequency	7
3.	Input/Output Curve Showing the Fast Expansion Technique and Its Effect on an Applied Signal	12
4.	Input/Output Curve with the Expansion Ratio; One is to Infinity	13
5.	Input/Output Curve with Different Expansion Ratios	13
6.	Sine Wave Used for Fast Expansion	14
7.	Sine Wave after Fast Expansion	14
8.	Pulses of Pure Sine Wave	16
9.	White Noise	16
10.	Pulses of Sine Wave after Addition of Noise	17
11.	Noise Added Sine Wave Pulses after Fast Expansion	17
12.	A Simple Sine Wave	18
13.	Sine Wave after Processing of Fast Expansion	18
14.	Piecewise Continuous Function Introduced with Sine Wave Because of Processing	19
15.	Sine Wave Used for Center Clipping	22
16.	Sine Wave after Center Clipping	22
17.	Input Signal	24
18.	Input Signal after Passing through Squelch Circuit	24
19.	Block Diagram of the Minicomputer-Based System for Speech Processing	26
20.	Minimum Sampling Rate for First-Order Sampling of Band-Pass Signals	28

FIGURE PAGE

21. General Flow Chart 31

22. Timing Diagram of a Typical Channel 33

23. Sampling Algorithm 33

24. A Spectrum of Pure Sine Wave 41

25. A Spectrum of Sine Wave after Fast Expansion 42

26. A Spectrum of Sine Wave after Center Clipping 43

27. A Spectrum of Sine Wave after Fast Expansion and Peak Clipping 44

28. A Spectrum of Speech Signal 45

29. A Spectrum of Speech Signal after Fast Expansion with a Threshold Value of 20% 46

30. A Spectrum of Speech Signal after Fast Expansion with a Threshold Value of 35% 47

31. A Flowchart of Get 76

32. A Flowchart of Input 77

33. A Flowchart of Fast Expansion 78

34. A Flowchart of Center Clipping 79

35. A Flowchart of Peak Clipping 80

CHAPTER I

INTRODUCTION

The problems of measuring and improving the intelligibility of speech are as old as speech communication itself. Speech signals which have been combined with noise often become unintelligible, yet the understanding of contaminated speech is essential in many practical situations.

Extensive research in this general area has been performed by many investigators including Professor Igor Nabelek of the UTK Audiology and Speech Pathology Department. He recognized the need and usefulness of studying the effects of a new technique of enhancing speech intelligibility called "fast expansion." However, a computer-aided search of the literature revealed that no work on this specific subject has been published.

This thesis describes the development of a system to investigate fast expansion as a tool for improving the intelligibility and quality of an audio signal. It begins by presenting history and background material on the general area of speech intelligibility in Chapter II. The next chapter describes fast expansion and the overall structure of the system. Chapter IV describes the speech processing system. In Chapter V results are evaluated. Finally a summary, conclusions and suggestions for further study are presented in Chapter VI.

CHAPTER II

REVIEW OF DEVELOPMENTS IN SPEECH PROCESSING

I. ERAS OF ADVANCEMENT

Although the problems of measuring and improving the intelligibility of speech are as old as speech communication itself, very few scientific investigations were undertaken until the telephone was invented. Ancient theaters used reflectors and resonators to improve audibility, but there seems to have been no work for speech enhancement (improving intelligibility of speech). Medievel buildings had reflectors and daises (raised platforms), but little attempt was made to suppress echos or to find their effects on intelligibility until the nineteenth century. Spoken word testing for deafness began in the sixteenth century or earlier, but there were no scientific tests of hearing loss for speech until the twentieth century.

Then the field seems to break rather conveniently into three eras, each with a dominant academic discipline, each with a new device that made major advances possible. The first period is that of the physicists, the vacuum tube amplifier, and the telephone laboratories; the second is that of psychologists and otologists, the tape recorders and the university laboratories. And the third is that of the stored program digital computers and its engineers.

The first period extends from before World War I to the middle of World War II. World War I created new problems. Two early patents for

noise-canceling, pressure gradient microphones specifically cite airplane noise as the origin of the requirement (1). Soon the vacuum tube amplifier and its associated electrical components allowed investigators for the first time to control and to measure accurately and easily the levels at which signals and noise are received, transmitted, and presented to listeners over a wide variety of transducers. Before that, voices could be made to sound louder to the listener only by speaking more loudly. This procedure changes the spectrum, timing and pronunciation of the speech and the distribution of the sound field around the head of the talker or listeners. Now, losses due to transmission or filtering could be compensated, spectra shaped, and noise introduced in controlled amounts at various points in the experiments. The whole field of signal-to-noise and band-width experiments was open to investigators. The investigators who generally had been trained in physics departments were learning and doing things that today are more likely to be taught in engineering, audiology or psychology departments. The laboratories of telephone organizations were dominant. A conspicious and important establishment was the Bell Telephone Laboratores in the United States of America. A great deal of pioneer research on speech and hearing has been done there.

The second period in the field extends from the middle of World War II to the early 1960's. Although during the war all nations had the same problem of communication in noise in ships' engine rooms, armoured vehicles, and aircrafts (where high altitude and oxygen masks introduced additional problems), the USA was the only nation that placed a high

priority on the problems and had sufficient scientific resources to allow considerable concentration of work on them by several laboratories and training stations. As a result, at the end of the war there was a reserve of research and development achievements and manpower trained in this branch. In this period an important development was the tape recorder (2), a device as well suited to encourage research in experimental audiology as the amplifier had been in physics. The device made possible for the first time the easy and accurate reproduction of noise and speech, precisely the same every time, or with changes in sequence duration, intensity, phase, frequency, or other modifications all precisely controlled. No doubt disc recording had permitted some of these controls but discs did not permit as many variations in technique as tape recorders did.

In the early 1960's the digital computer became an important tool in speech intelligibility research. First as a controller, analyzer, and synthesizer of speech itself and then as a controller of experiments and as an analyzer of the results. The physicists, engineers, and audiologists learned to use the machines for their purposes but the chief contributions during this period have come from the investigators who are sufficiently familiar with the computers to be able to identify themselves as computer engineers. We are living in this period and it seems that research and development on the intelligibility of speech will continue to show great progress.

II. TECHNIQUES FOR MEASURING AND IMPROVING INTELLIGIBILITY

Speech signals which have been combined with noise often become unintelligible but the understanding of speech contaminated by the presence of noise is an important consideration in many practical communication situations. For example, factory workers who receive information from public address systems or pilots who require accurate instructions from ground stations need to obtain intelligible signals. One method for improving the intelligibility of speech in noise is to simply decrease the output level of the noise generating source. Another procedure involves increasing the volume of the speech signal, thereby producing a more favorable signal-to-noise ratio. But sometimes these techniques are not feasible or economical.

Human sensory perception is facilitated by the brain's processing of information transmitted to it along nerve bundles from peripheral organs. In auditory perception it is known that a stimulating acoustic waveform is first processed by auditory mechanisms which are peripheral to the brain. This peripheral processing ultimately converts acoustic energy into neural impulses. Neural information is then transmitted along the auditory nerve into a complex of ascending neural pathways reaching to the auditory cortex of the brain (3).

The idea of acoustic signal processing is to identify the type of distortion present and to do something electrically to the signal that is equal but opposite to that distortion. This approach, when effective, accomplishes what has sometimes been called the "transparent amplifier" (4).

In principle it is possible to undo in advance, electronically, the damage that is going to be done to the signal when it gets into the ear of an impaired listener. To determine exactly what distortions would take place in the ear, clinical tests are required such as those reported in the case of Mary Ann Mirmelstein of Newport News, Virginia (5). She became strangely unresponsive after a bout with measles when she was three years old. She went through a long series of tests and at last was declared as deaf. Then at age 27, she met Charles I. Berlin, an audiologist who heads the Kresey Hearing Research Laboratory of the South in New Orleans. Using special equipment Berlin was able for the first time to provide a precise diagnosis of her problem as "ultra-audiometric" hearing (which is very uncommon) the capacity to hear, but only at extremely high frequencies. Once the problem was identified, Berlin developed a device that "translates" low frequency sounds into the range at which ultra-audiometrics could hear them.

In most situations which require speech processing, the redundancy of human speech is exploited. Figure 1 is from the old experiments by French and Steinbert (6). Using various filters, they determined how much speech information was in the high frequencies and in the low frequencies, finding a crossover point at about 1900 hz. As it can be seen, about 70% intelligibility if the filter passes everything above 1900 hz, and about 70% if passes everything below. If there were no redundancy, the figures would be 50% above and below.

Figure 2 says essentially the same thing, showing the cumulative relative intelligibility attributed to frequency. The ordinate, the

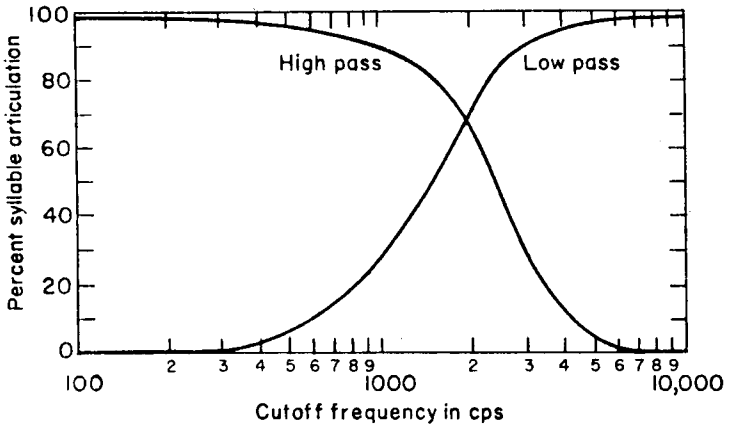


Figure 1. Syllable articulation for high-pass and low-pass system operating in quite with optimal gain.

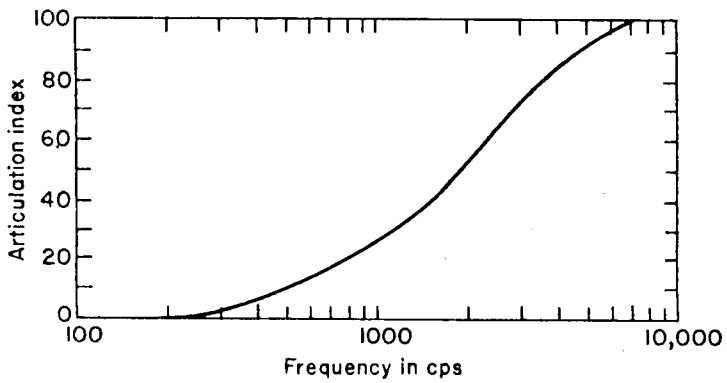


Figure 2. Articulation index as a function of frequency.

articulation index, departs from zero at about 250 Hz and reaches an asymptote at about 7,000 Hz, implying that there is no speech information below 250 Hz or above 7,000 Hz. Thus, when speech is completely filtered below 250 Hz and above 7,000 Hz, there should be no decrement in the intelligibility score.

These data form the basis for the articulation index, which involves breaking this 250-7,000 Hz range into several bands, each of which contributes the same amount of information to the overall intelligibility. If one really wants to predict what the effect of noise on the perception of speech is going to be, he must use the articulation index. The theory is straightforward enough. The speech range is divided into 20 segments, each of which contributes, theoretically, 5% to total intelligibility. It is necessary to compute the speech signal-to-noise (S/N) ratio in each of these bands for the particular speech and noise concerned. These ratios are then divided by 30, multiplied by 5%, and added together. The tacit assumptions are that when the S/N ratio is 0, there is no contribution to intelligibility by the speech energy in that band, that at S/N = 30 db, it is contributing all it can, and that the contribution is otherwise proportional to the S/N ratio. The articulation index works fairly well in predicting about how much speech will be heard in a particular situation (7).

Measuring and improving speech intelligibility has application to many fields like architectural acoustics, audiology, communication engineering, speech synthesis and analysis, speaker recognition, etc. Audiology was the first field studied, beginning in the 19th century.

Remarkable development occurred in the 20th century mainly due to advancements in the sciences of neurology, linguistics, perceptual psychology, phonetics, electronics and computers.

Several methods have been proposed to date to enhance the quality and intelligibility of speech signals embedded in noise. The following is a list of some methods used for speech enhancement (S):

1. Peak Clipping
2. Center Clipping
3. Signal Companding
4. High Pass Filtering
5. Selective Processing
6. Multi-microphone Mixing
7. Linear Spectral Subtraction
8. Power Spectral Subtraction
9. INTEL Technique
10. Perceptual Subtraction
11. Optimum Weiner Filtering
12. Adaptive Comb Filtering
13. Harmonic Location and Selection
14. Adaptive Noise Cancelling Technique
15. Vocoder based on using All Pole Model of speech
16. A Critical Band All-Pole Vocoder
17. Vocoder based on using a Pole-Zero Model of Speech
18. Vocoder based on a Nonparametric Model of Speech
19. Squelch Circuit
20. Combination of two or more of the above techniques

All these methods have their own advantages and disadvantages and combinations may be chosen. But surprisingly, fast expansion is not mentioned anywhere as a method for improving the intelligibility of audio signals. In this thesis the implementation of this new technique will be described.

CHAPTER III

SPEECH PROCESSING TECHNIQUES

I. FAST EXPANSION

Fast expansion is the name given to a process in which the central band of a time varying signal is cut down and brought to zero while the remaining portion is left unaffected. The level of signal where this operation takes place is adjustable and called the threshold value. This definition is absolutely true only when the ratio of expansion is one to infinity, i.e. if the output goes from zero to the input signal value after the threshold point in practically no time. This is shown in Figure 3 and Figure 4. When the expansion ratio is other than one to infinity, the graphs will be as shown for three cases in Figure 5 for the expansion ratios of 1:2, 1:5 and 1:10. When the input signal reaches a point "A" the output starts increasing gradually. This increment depends upon the ratio of expansion. Beyond the threshold value "B" the output becomes the same as the input.

To understand the process more fully, consider the simple example of the sine wave shown in Figure 6. It is processed with an infinite expansion ratio with the threshold level set at "B." After processing, the signal will look like that shown in Figure 7. We can see that harmonics are introduced due to processing. This causes the distortion. So we pass it through a bandpass filter to eliminate the harmonics. After filtering we again get the same curve as in Figure 6.

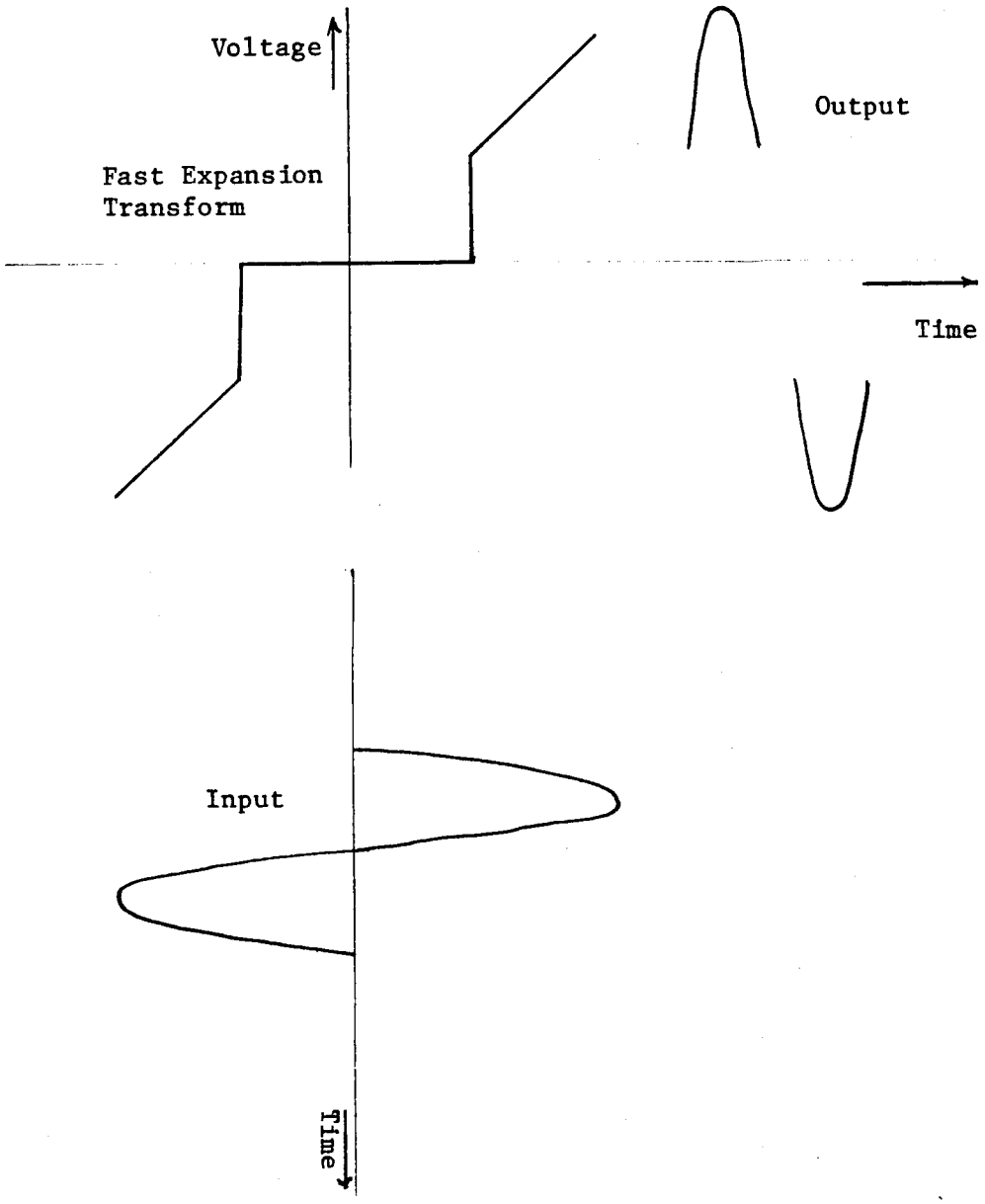


Figure 3. Input/output curve showing the fast expansion technique and its effect on an applied signal.

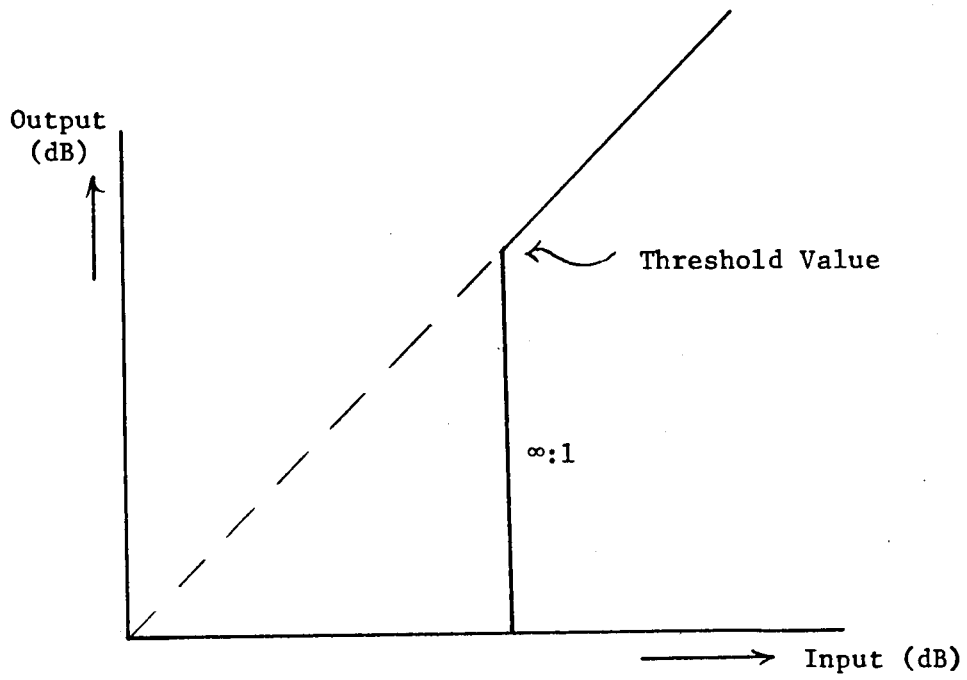


Figure 4. Input/output curve with the expansion ratio; one is to infinity.

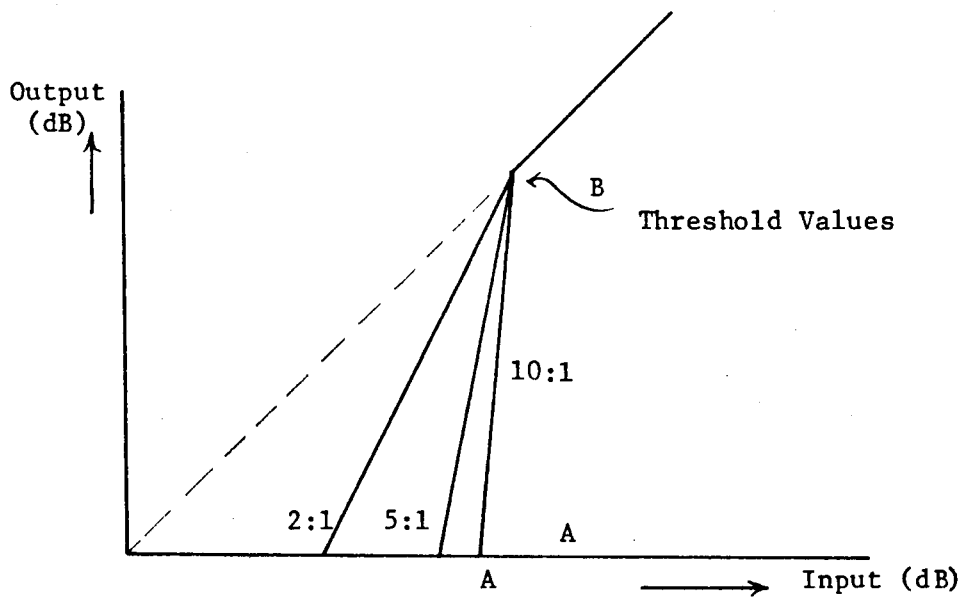


Figure 5. Input/output curve with different expansion ratios.

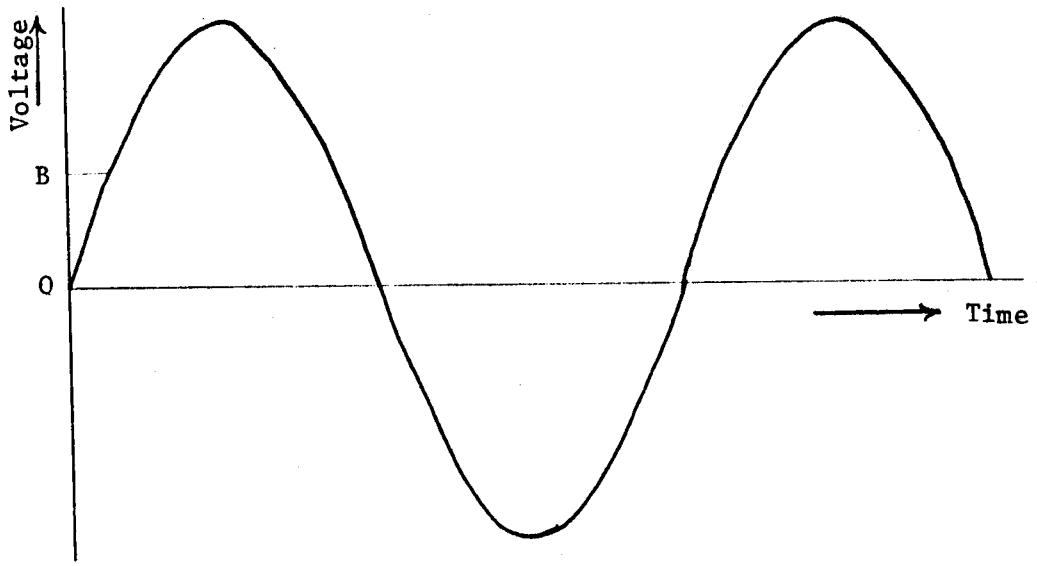


Figure 6. Sine wave used for fast expansion.

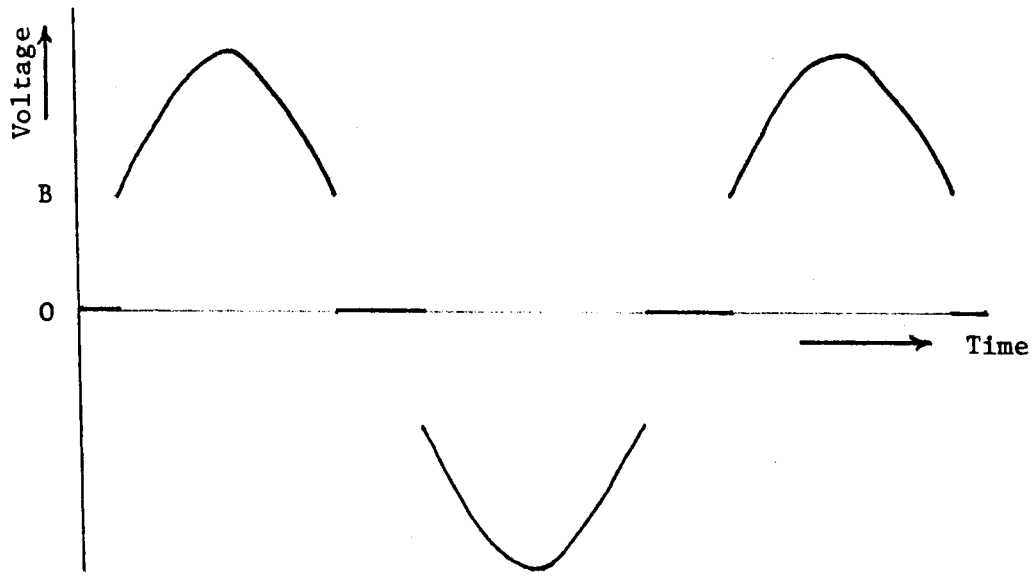


Figure 7. Sine wave after fast expansion.

A more realistic situation is evident when the pulses of the pure tone signal of Figure 8 are combined with the white noise of Figure 9. After addition, the signal becomes like that shown in Figure 10. When processed using infinite ratio fast expansion transform, the signal becomes like that shown in Figure 11 after filtering. It is evident that noise between the two pulses has been removed completely yet the signal to noise ratio has remained unaffected. The same process can be employed for speech signals to remove the reverberating tails and eliminate the noise in pauses between words.

The characteristic of the waveform obtained by fast expansion can be evaluated mathematically. It can be seen that the waveform shown in Figure 12 after processing is the summation of Figure 13 and 14 The function of Figure 14 can be defined as follows.

$$f(\theta) = -\operatorname{Sin}\theta \qquad 0 \leq \theta \leq B$$

$$\pi - B \leq \theta \leq \pi + B$$

$$2\pi - B \leq \theta \leq 2\pi$$

$$= 0 \qquad \text{elsewhere}$$

This is an odd function so $a_n = 0$ and the period is from $-\pi$ to π.

$$b_n = \frac{2}{2\pi} \int_{-\pi}^{\pi} f(\theta) \operatorname{Sin} n\theta \, d\theta$$

$$= \frac{2}{\pi} \int_{0}^{\pi} f(\theta) \operatorname{Sin} n\theta \, d\theta$$

$$= -\frac{2}{\pi} \int_{0}^{\pi} \operatorname{Sin}\theta \operatorname{Sin} n\theta \, d\theta$$

$$= \frac{-2}{\pi} \left[\int_{0}^{B} \operatorname{Sin}\theta \operatorname{Sin} n\theta \, d\theta + \int_{\pi-B}^{\pi} \operatorname{Sin}\theta \operatorname{Sin} n\theta \, d\theta \right]$$

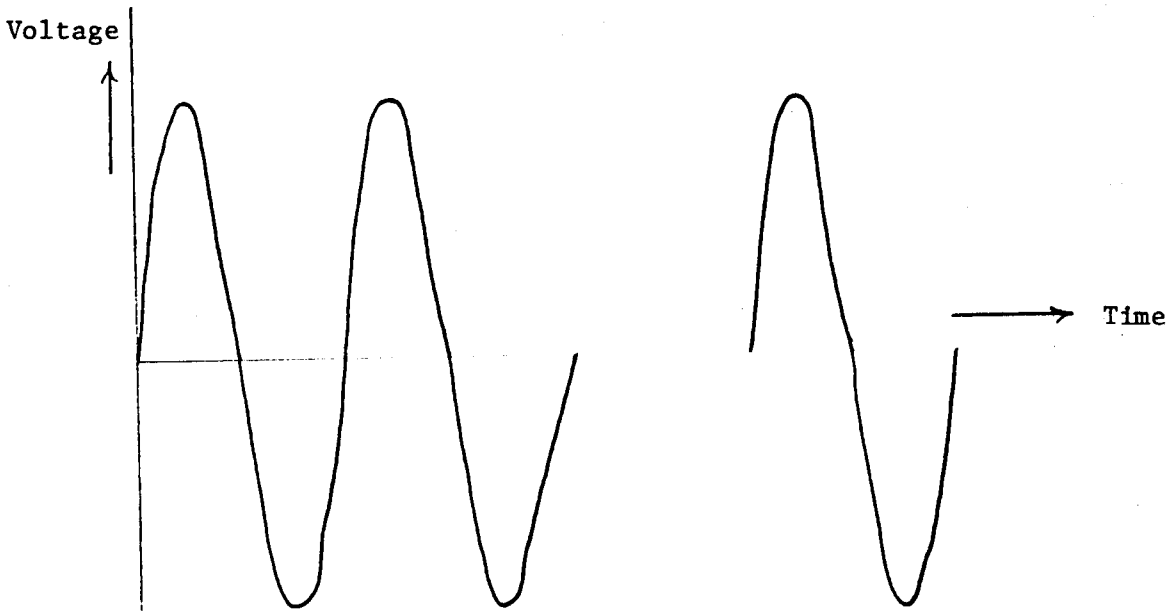


Figure 8. Pulses of pure sine wave.

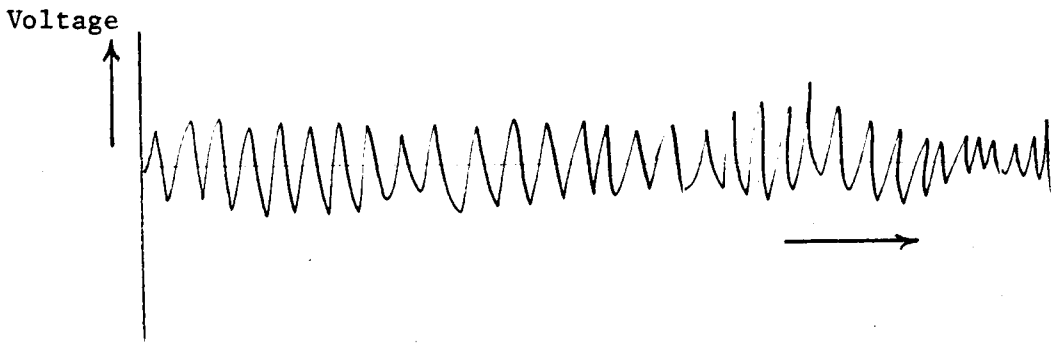


Figure 9. White noise.

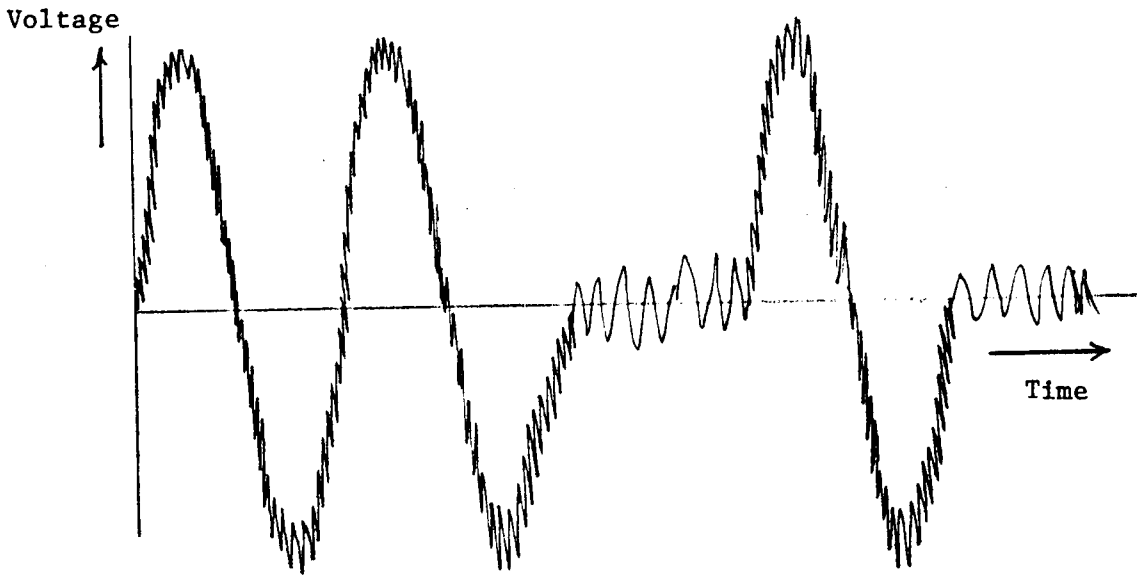


Figure 10. Pulses of sine wave after addition of noise.

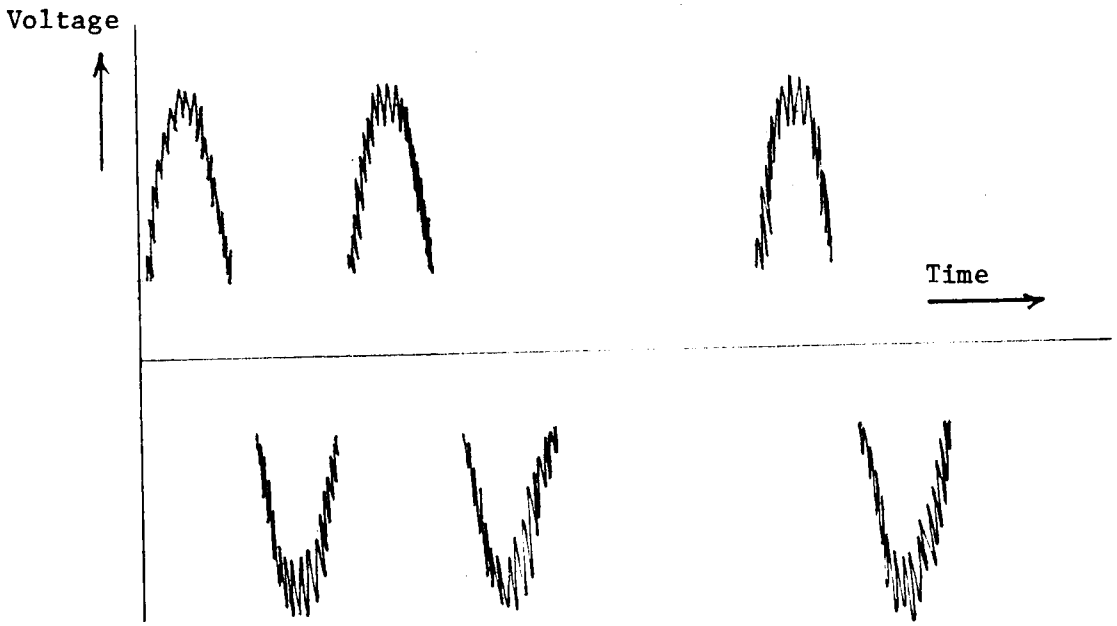


Figure 11. Noise added sine wave pulses after fast expansion.

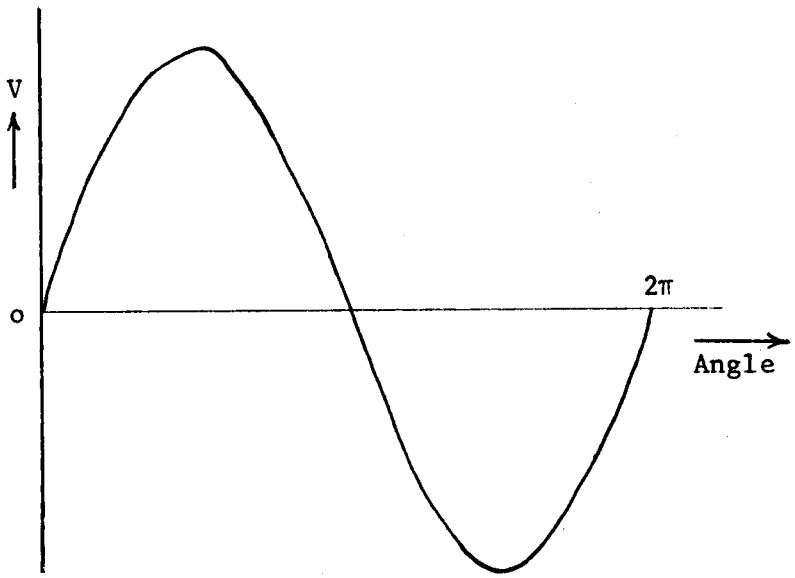


Figure 12. A simple sine wave.

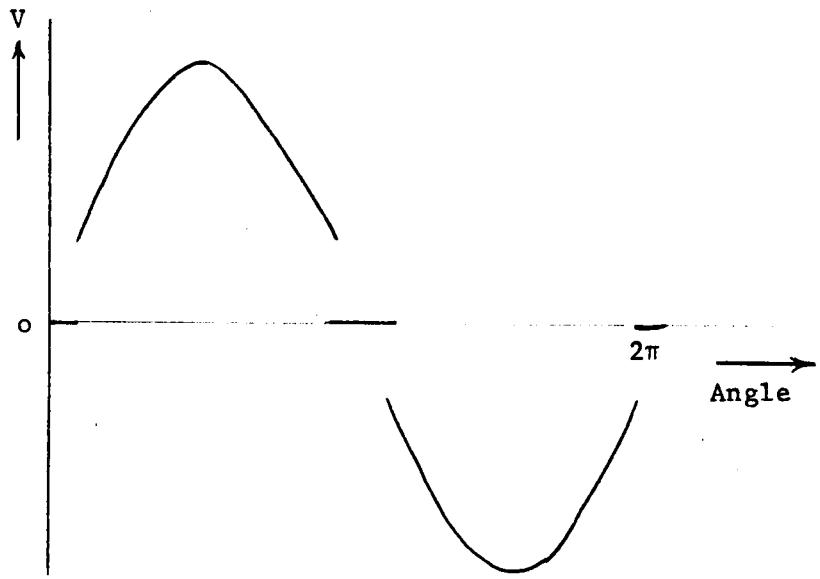


Figure 13. Sine wave after processing of fast expansion.

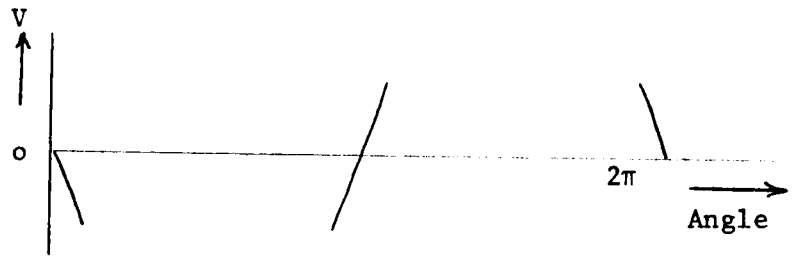


Figure 14. Piecewise continuous function introduced with sine wave because of processing.

$$= -\frac{1}{\pi}\left\{\left[\frac{\text{Sin }(1-n)B}{1-n} - \frac{\text{Sin }(1+n)B}{1+n}\right] + \left[\frac{\text{Sin }\pi - n\pi}{1-n}\right.\right.$$

$$\left. - \frac{\text{Sin }(\pi + n\pi)}{1+n} - \frac{\text{Sin }(\pi - B - n\pi + nB)}{1-n}\right.$$

$$\left.\left. + \frac{\text{Sin }(\pi - B + n\pi - nB)}{1+n}\right]\right\}$$

$$= -\frac{1}{\pi}\left\{\left[\frac{\text{Sin }(1-n)\,B}{1-n} - \frac{\text{Sin }(1+n)\,B}{1+n}\right] + \left[-\frac{\text{Sin }(n\pi + B - nB)}{1-n}\right.\right.$$

$$\left.\left. - \frac{\text{Sin }(n\pi - B - nB)}{1+n}\right]\right\}$$

$$= -\frac{1}{\pi}\left[\frac{\text{Sin }(1-n)\,B}{1-n} - \frac{\text{Sin }(1+n)\,B}{1+n} - \frac{\text{Cos } n\pi \text{ Sin }(B - nB)}{1-n}\right.$$

$$\left. + \frac{\text{Cos } n\pi \text{ Sin }(n+1)B}{1+n}\right]$$

for n = even

$$b_n = 0$$

for n = odd and $n \neq 1$

$$b_n = -\frac{1}{\pi}\left[\frac{\text{Sin }(1-n)\,B}{1-n} - \frac{\text{Sin }(1+n)\,B}{1+n} + \frac{\text{Sin }(1-n)\,B}{1-n}\right.$$

$$\left. - \frac{\text{Sin }(1+n)\,B}{1+n}\right]$$

$$= -\frac{2}{\pi}\left[\frac{\text{Sin }(1-n)\,B}{1-n} - \frac{\text{Sin }(1+n)\,B}{1+n}\right]$$

$$= \frac{2}{\pi}\left[\frac{\text{Sin }(1+n)\,B}{1+n} - \frac{\text{Sin }(1-n)\,B}{1-n}\right]$$

$$\text{As } f(\theta) = \frac{a_o}{2} + \sum_{n=1}^{\infty} (a_n \text{ Cos } n\theta + b_n \text{ Sin } n\theta$$

Therefore, for the function described

$$f(\theta) = \frac{2}{\pi}\left[\left(\frac{\text{Sin } 2B}{2} - B\right)\text{Sin }\theta - \left(\frac{\text{Sin } 2B}{2} - \frac{\text{Sin } 4B}{4}\right)\text{Sin } 3\theta - \left(\frac{\text{Sin } 4B}{4} - \frac{\text{Sin } 6B}{6}\right)\text{Sin }\theta + \ldots\right]$$

Therefore, the waveform obtained by fast expansion can be represented as follows.

$$f(\theta) = \text{Sin }\theta + \frac{2}{\pi}\left[\left(\frac{\text{Sin } 2B}{2} - B\right)\text{Sin }\theta - \left(\frac{\text{Sin } 2B}{2} - \frac{\text{Sin } 4B}{4}\right)\text{Sin } 3\theta - \left(\frac{\text{Sin } 4B}{4} - \frac{\text{Sin } 6B}{6}\right)\text{Sin }\theta + \ldots\right]$$

where $\theta = W_o\, t$

$B = \text{Sin}^{-1}(\text{threshold value})$

Assuming peak value as unity.

II. CENTER CLIPPING

It will be helpful for comparison and evaluation of results to describe another technique in use for speech enhancement called "center clipping." In this technique, the center portion of the signal is cut down and brought to zero while the threshold value is subtracted from the remaining portion.

Again consider a sine wave like that shown in Figure 15, having a peak value greater than the threshold value. Center clipping produces the signal shown in Figure 16, which is obviously different from that shown in Figure 7, page 14, produced from fast expansion.

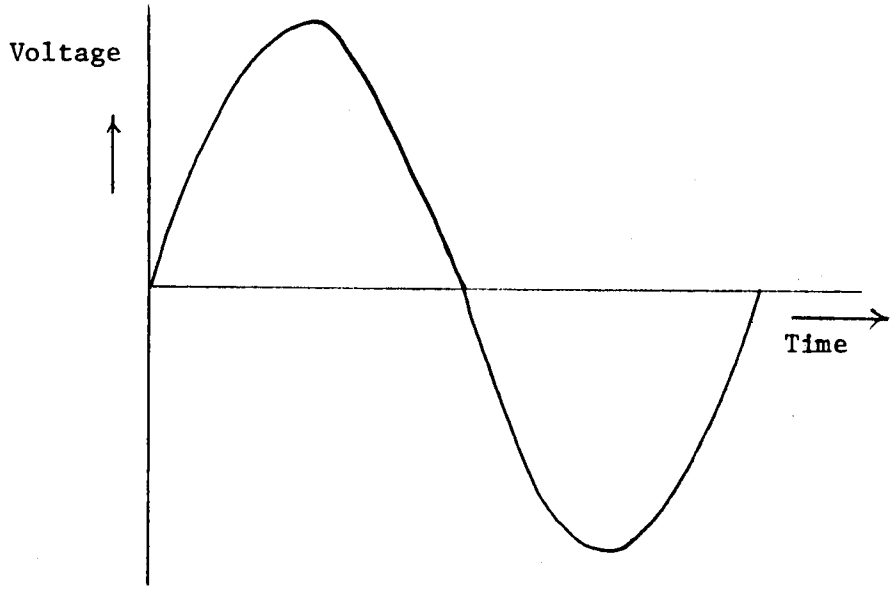


Figure 15. Sine wave used for center clipping.

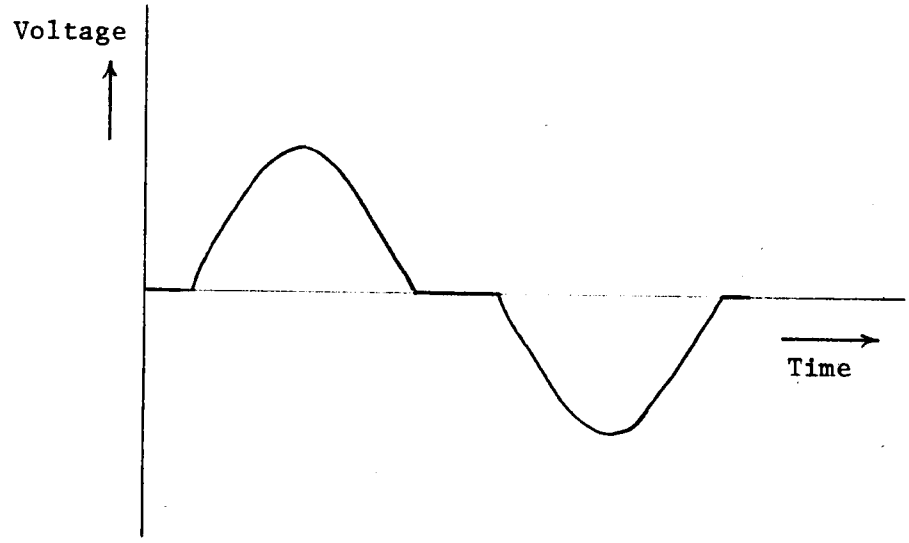


Figure 16. Sine wave after center clipping.

III. SQUELCH

For many years squelch circuits have been employed in radio receivers to cut off the receiver output when the incoming signal is below a threshold value (8). This technique is illustrated in Figure 17 and Figure 18. Fast expansion differs from the squelch technique in that the digital method totally removes the central portion of the signal. This procedure eliminates the reverberating tails and noise present in between words completely.

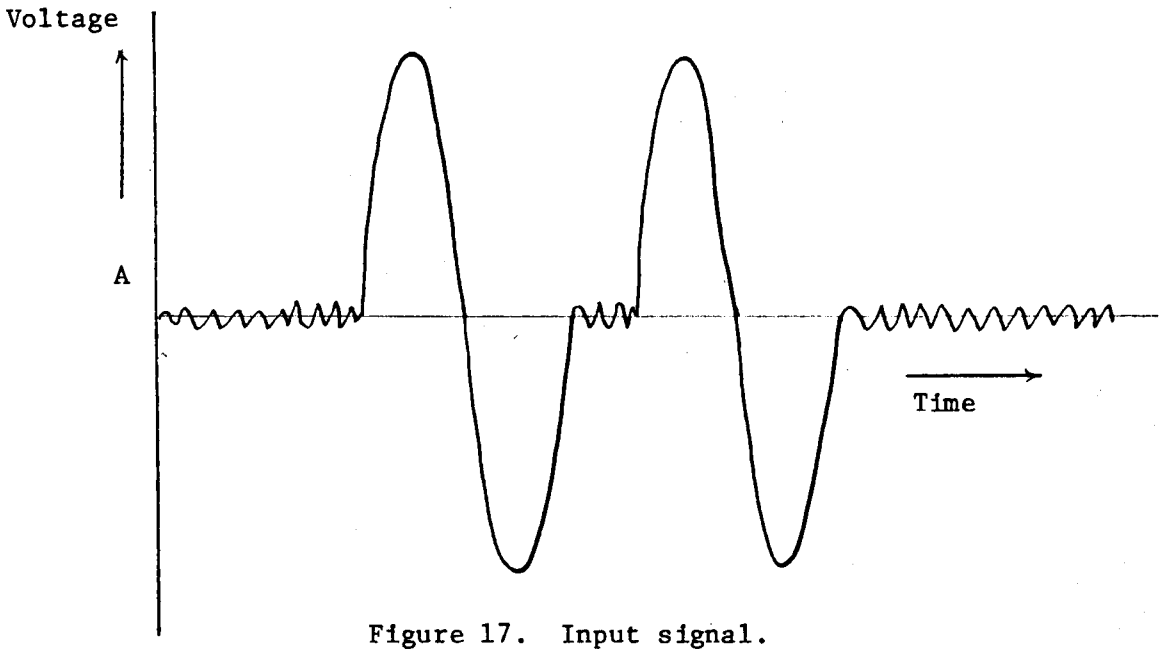


Figure 17. Input signal.

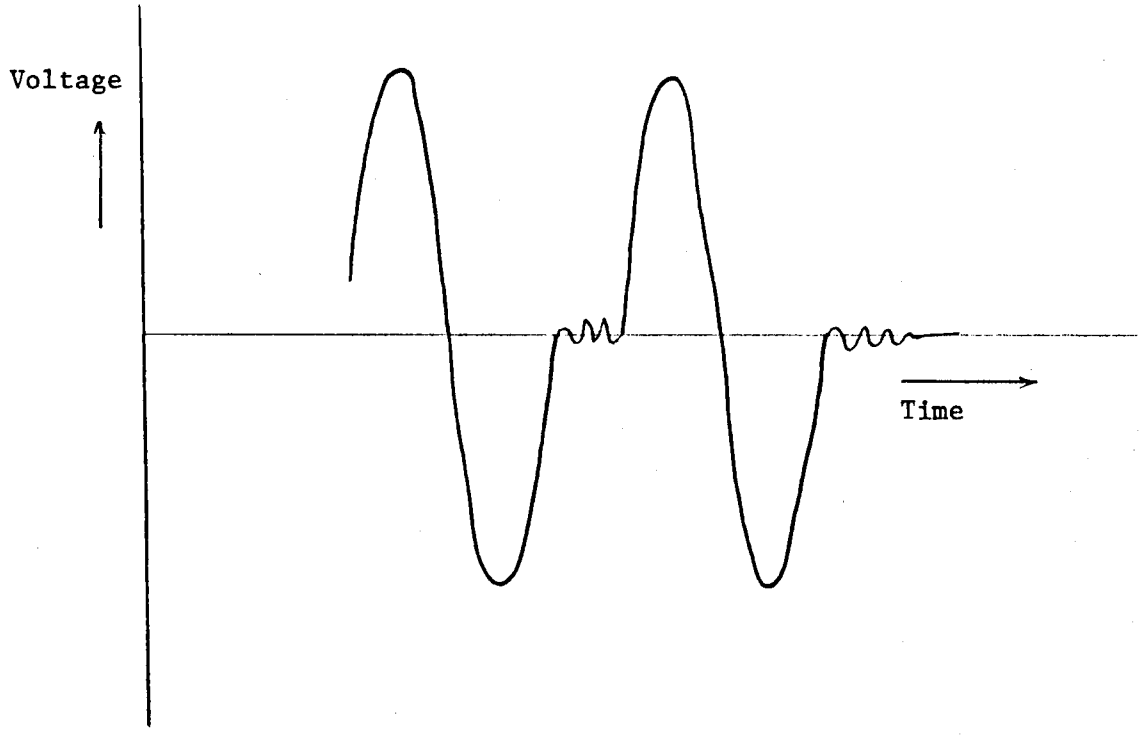


Figure 18. Input signal after passing through squelch circuit.

CHAPTER IV

MINICOMPUTER-BASED SPEECH PROCESSING SYSTEM

I. STRUCTURE OF THE EXPERIMENTAL SYSTEM

A block diagram of the minicomputer-based system for speech processing is shown in Figure 19. Signals from the tape player and noise generator are mixed through attenuators so that the levels of speech signal and noise can be adjusted independently. The tone generator can also be used in some experiments instead of the tape player. After mixing, the signal is amplified to the desired level. Then this signal is fed to the computer through two bandpass filters, with one band an octave higher than the other. The analog to digital converter (A/D) encodes the analog signal into a digital code and makes it available to the computer for further processing. The second channel of higher frequency bandwidth is sampled twice as fast as the first one. After the computer does the desired processing, analog output is made available through the digital-to-analog converter (D/A) at two output terminals called x and y. This discrimination is done so that bandpass filters of respective bandwidth can be employed after processing to avoid the distortion produced by the processing. These two signals are mixed again after passing through the bandpass filters. Then the signal is amplified or attenuated to adjust the desired level of output. Now this signal can be listened to, displayed on an oscilloscope or recorded for further use. The

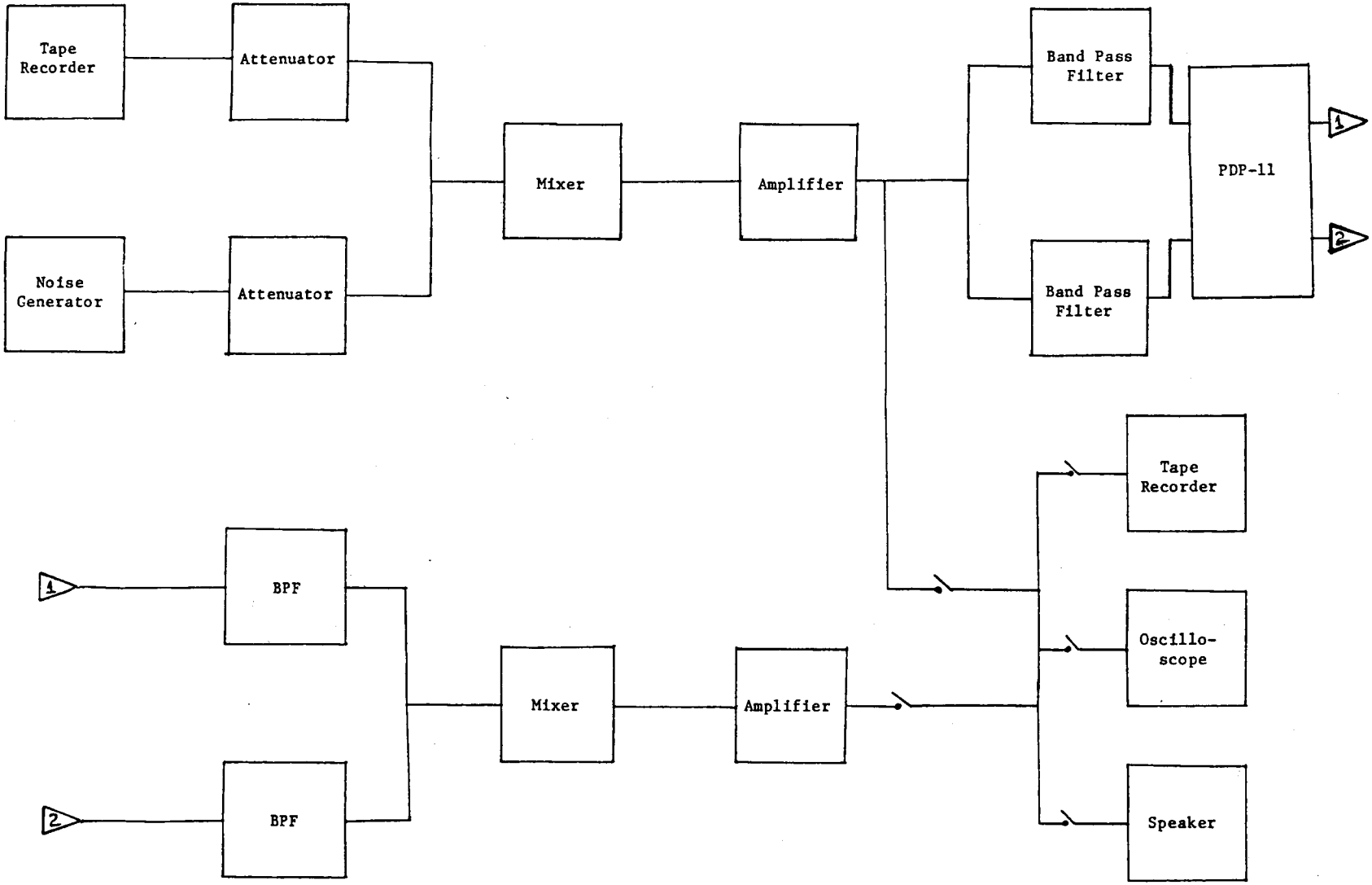


Figure 19. Block diagram of the minicomputer-based system for speech processing.

original input signal is also made available here through switches in order to compare the processed signal with the unprocessed one.

II. SAMPLING OF AUDIO SIGNALS

To implement fast expansion in real time some care in writing software must be taken. The Nyquist theorem suggests that a signal must be sampled at least twice the highest frequency component present in it in order to reconstruct it faithfully. This imposes an upper limit of time between two samples. For example, a band of 0-4,000 Hz should be sampled and processed at the rate of 8,000 samples/second. So it is obvious that the amount of software that can be written for processing is limited primarily by the processor execution speed rather than the amount of memory available.

In multichannel processing sampling no longer need be twice the highest frequency component of a particular band sampled. Because a bandpass signal can be sampled in proportion to its bandwidth instead of the highest frequency component present in it. Since it has spectral components only in the range $W_o < w < W_o + W_f$, the minimum sampling rate $2W_f$ can now be realized only for certain discrete values of $(W_o + W_f)/W_f$. For other ratios the minimum rate will be somewhere between $2W_f$ and $4W_f$ as determined from the sawtooth plot shown in Figure 20 (10). This procedure can be exploited to allow more time for processing when the minicomputer based system is expanded to more channels.

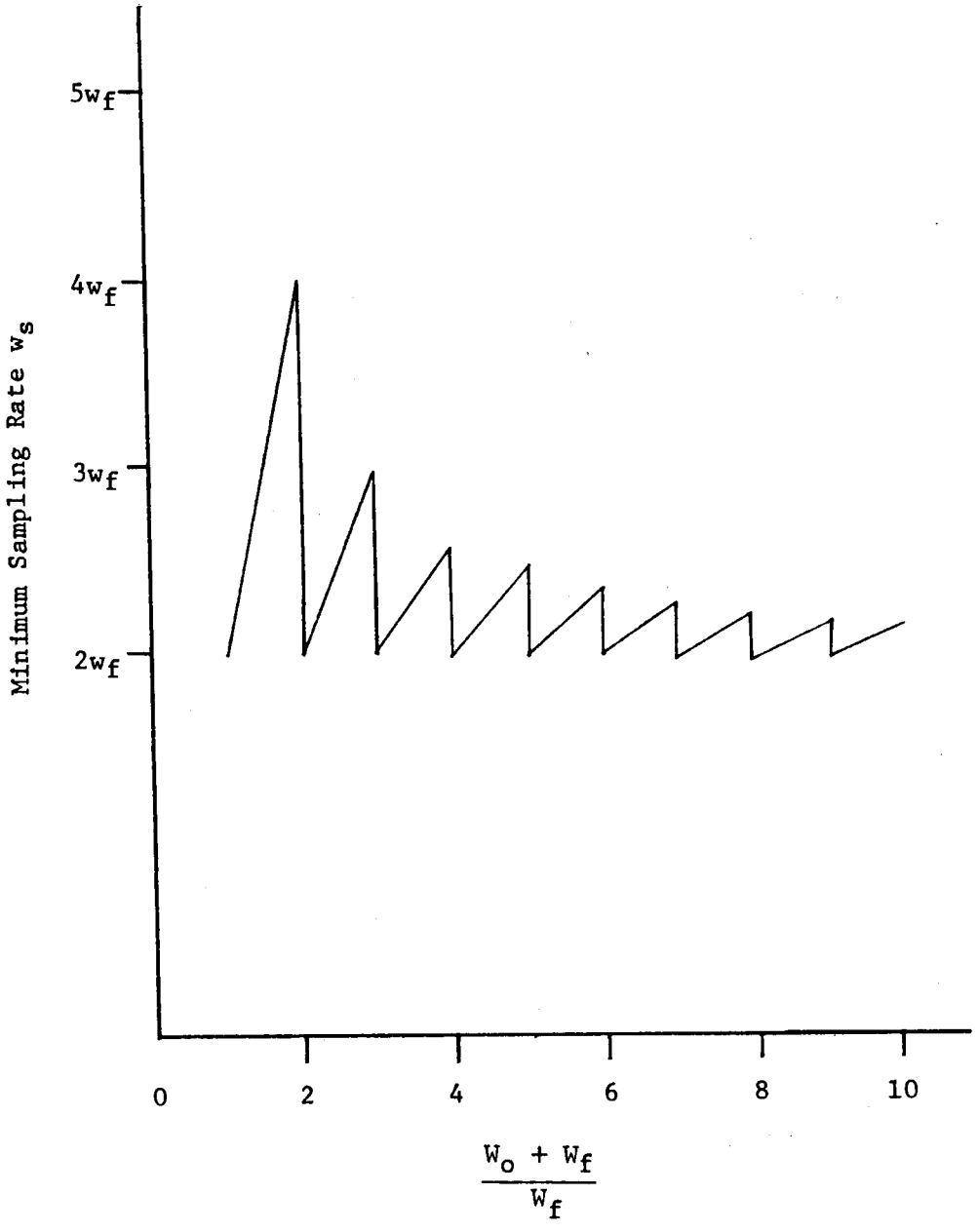


Figure 20. Minimum sampling rate for first-order sampling of band-pass signals.

III. INPUT/OUTPUT MODULE

Analog signal voltages are made available to the computer by a multiplexed A/D converter. Multiplexing allows all sixteen channels (however, only two channels have been implemented to date) to share a single high speed converter, saving hardware and decreasing the cost. Since the minicomputer is a general purpose machine, it makes sense to have an analog to digital conversion subsystem that is capable of all expected sampling requirements. A moderate speed converter (10 microsecond conversion time) with twelve bit capability was chosen, to allow full audio bandwidth signals to be digitized with reasonable accuracy. The analog data acquisition module used is a Data Translation model DT 1761. This module provides results up to a 100 khz rate and includes a software programmable multiplexer for sixteen channels. The multiplexer can be programmed to increment automatically to the next channel on each conversion if desired. In addition, DMA (direct memory access) capability is included so that conversion can be made and data stored at high speed without processor intervention. Interrupts can be generated on completion of conversion or of DMA transfer. The complete analog interface is contained on a single 10½ by 8½ inch (quad size) printed circuit card that plugs into the PDP11/23 backplane. The detailed specification are given in Appendix A.

IV. COMPUTER

Several factors were considered in choosing the computer. The processor had to be capable of signal analysis and desired processing at the required speed. Secondly it must provide facilities for software development. The above considerations led to the selection of Digital Equipment Corporation PDP 11/23 minicomputer for the system. This processor uses data 16-bit words and can operate efficiently on 8-bit bytes. The winchester disk associated with it has 31.5M bytes of memory. Although it is dedicated to fast expansion for our purposes the computer can be used for monitoring other experiments as well.

V. SOFTWARE

The general flow chart is shown in Figure 21. In the first step, the user is asked to specify the threshold values of both channels. These values are converted to octal and then stored in memory locations allocated for them. Then the main program starts the data conversion. After processing, the signal is converted to an analog signal and is sent to the outside world.

The easiest and most direct method of sampling the two channels is to sample each channel once per program loop. No doubt this is a simple method but this scheme is inefficient. Since the sampling rate of a signal depends upon its frequency, it was decided to sample the band which is one octave higher than the other, at twice the rate.

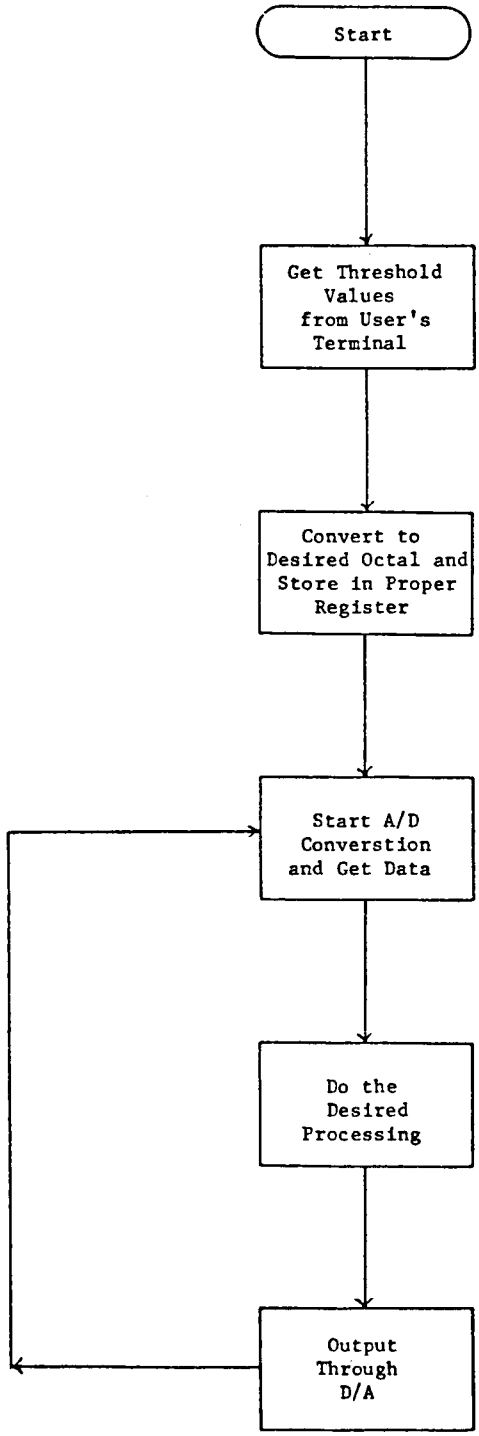


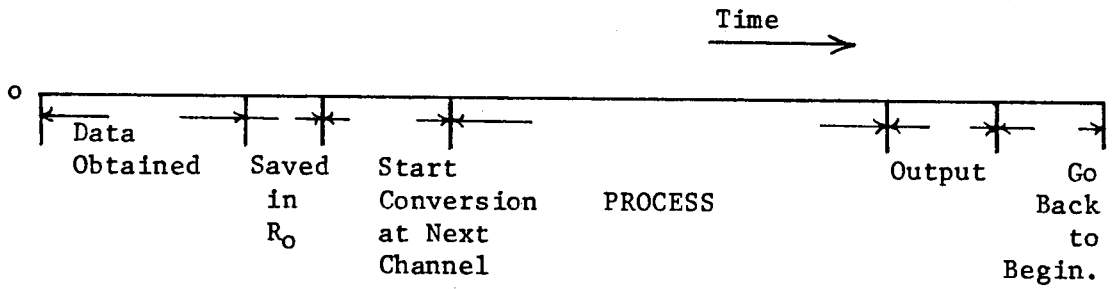
Figure 21. General flow chart.

All programs were written in PDP-11 assembly language and are given in Appendix B. Detailed flow charts are given in Appendix C. The source code was edited, assembled and converted to object code using the RT-11 operating system (10).

VI. FAST EXPANSION

First of all, threshold values of the two channels are stored in different registers. This is done in order to get the maximum speed of processing. After that, A/D conversion is started at channel 0. The program waits until data is available at the A/D Data Buffer Register (ADDBR). Then this data is transferred to register R0 for further processing. This digitized data is then compared with its respective threshold value. If it falls below it, D/A Data Buffer Register (DACDBR) is cleared; otherwise, the same value is transferred to DACDBR. Then output is made available to the outside world through output channel X. Now control is transferred to get the data from channel 6 and the same processing is performed except that now D/A output is made available at terminal Y. The system samples and processes channel 6 once more before going back to channel 0. The entire procedure is then repeated indefinitely.

In Figure 22 the timing diagram of a typical channel is shown. It takes 43.8μsec. to complete the entire process for one sample. The sampling rate is therefore 22,800 samples/sec. Figure 23 shows the sampling algorithm employed that results in sampling channel 6 twice as fast as channel 0. It shows that sampling of channel 6 is



Data Obtained	=	8.24 μsec
Saved in Register	=	3.97 μsec
Start Conversion at Next Channel	=	5.39 μsec 17.66 μsec
PROCESS	=	17.66 μsec
Output	=	4.27 μsec
Go Back to Begin.	=	4.27 μsec
TOTAL		43.80 μsec

Figure 22. Timing diagram of a typical channel.

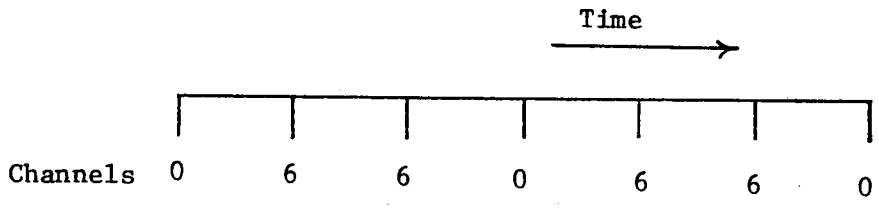


Figure 23. Sampling algorithm.

uneven which caused no problem since the sampling rate was much higher than suggested by the Nyquist theorem.

VII. INPUT

In this subroutine, values are taken from the terminal and checked for the legality of the characters. If any illegal character is typed or if more than four digits are input, then after hitting the return key causes the subroutine to give an error message and go back to the beginning of the subroutine. If the delete or rubout key is struck on the terminal, the program deletes the previous input character. If the user inputs the legal value defined by the program, it will store this value in register R0 properly and control will be returned from the subroutine.

VIII. GET

This program was written to assist the user in adjusting the threshold values from the terminal by answering the questions asked by the computer. First of all, it gives the message that information regarding the use of this and related programs can be found in the file named Proj.Hlp. Then it asks for threshold values (positive or negative) for a particular channel. After asking this question control is transferred to subroutine INPUT. Then the input value entered from the terminal is converted into the desired octal number and stored in a location reserved for it. The next question is then asked and so on until the user reaches the end of the program, where control is transferred to the main program.

IX. CENTER CLIPPING

This program implements the center clipping technique described in Chapter III. First of all, it compares the digitized value of the input signal to the threshold value. If it falls below the value, the output buffer is cleared, forcing the output signal to zero. Otherwise, the threshold value is subtracted from the input value and the result is moved to the output buffer. The reverse is true for the negative half cycle. Like fast expansion, it also samples and processes channel 6 twice as often as channel 0. The output for both channels are made available at output channels X and Y respectively.

X. PEAK CLIPPING

This program also makes use of GET and INPUT to receive the threshold values from the user's terminal. Once these values are obtained, it operates in the same manner as fast expansion or center clipping. It initiates the data conversion at channels 0 and 6; processes the data and delivers the output signals to the respective output ports. The only way it differs from the other two programs is the processing algorithm. The digitized signal is transferred directly to the output buffer if it falls below the threshold value. Otherwise, it is clipped by transferring the threshold value to DACDBR for output. The reverse is true for the negative half cycle.

CHAPTER V

RESULTS

Two types of experiments were carried out using this system of fast expansion: one using a pure tone of different frequencies, while in the other, effects on speech signals were studied. The experiments performed were preliminary in nature. They were designed to verify that the minicomputer system implements the fast expansion technique as intended. Additional extensive experimentation on the intelligibility of speech processed in this way are planned. Such experiments involve multiple trials using several human subjects to judge the effectiveness of this new technique.

I. EXPERIMENTS WITH PURE TONES

After processing as discussed in Chapter IV, harmonics were introduced. The effect of harmonics was more prominent with lower frequencies than with higher ones. It was observed that above 6,500 Hz the effect of harmonics introduced by processing is very little and increasing the threshold value from 5% to 70% of peak value does not seem to affect the quality of the tone, while at lower frequenices increasing the threshold value does increase the distortion. In another experiment, it was found that low frequencies like 50 Hz, were heard after processing but not beforehand.

In another set of experiments, it was found that the intensity of the signal remains unaffected until the threshold level of 90% is

reached. Between 90% and 95% it sounds very distorted and becomes harder and harder to understand. Beyond 95% it is practically impossible to comprehend. In the preceding experiments, distortion was completely removed when a bandpass filter of proper bandwidth was employed after processing.

When a continuous signal mixed with noise was processed it remained practically unaffected. (Here for comparison we took the signal passing through the computer and coming out of the bandpass filters, but without any processing, so that filtering could not mislead the evaluation of the results.) But when pulses mixed with continuous background noise were used, it was observed that, with proper adjustment of threshold level the noise was completely eliminated between pulses (i.e., during pauses between the tones). These pulses sounded the same and the overall effect on perception was good. It was observed that the noise level should be below 70% in order to accomplish the above mentioned results.

These results are tabulated in Table 1. All of these results were very encouraging and the system response was better than originally anticipated.

II. EXPERIMENTS WITH SPEECH

Processing of speech signals imposed the problem of selecting the bandwidth to be employed before and after the processing because there are only two output channels available. After countless trials, bands of 600-1,150 Hz. and 1,550-3,000 Hz. were found better than other combinations. It was observed that after passing through the computer

Table I

Fast Expansion Results Using Pure Tone

Experiments Performed with	Threshold Value	Comments
Pure tone 0-100Hz.	0-5%	Hard to hear.
	5-30%	Could be heard.
100-6500Hz	0-5%	Distortion was introduced.
	5-30%	Distortion was increased by increasing the threshold value.
	0-5%	
6500-10,000Hz.	0-5%	A little distortion in tone was detected.
	5-70%	No difference was detected by increasing the threshold value.
	70-90%	No change in intensity.
	90-100%	Became harder and harder to listen.
Pure tone mixed with noise.	0-70%	Practically remained unaffected.
	70-90%	Became more distorted.
Pulses of pure tone mixed with continuous white noise.	0-70%	Quality was improved, best results depend upon adjusting threshold according to S/N ratio.
	70-90%	Quality started decreasing.

and bandpass filters, the quality of the speech deteriorated considerably. It was due to eliminating the frequencies below 600 Hz, above 3,000 Hz. and between 1,150 Hz. to 1,550 Hz.

First, ordinary continuous speech with added noise was processed. No improvement on the intelligibility was observed; in fact, the quality was degraded considerably. In other experiments words with pauses were taken for processing. Here noise in between words was removed but again there was no improvement on intelligibility.

Keeping in mind the effect of filtering and limited bandwidth, another set of experiments was conducted. This time the speech signal passing through the computer and coming out of the filters, but without any processing, was used for comparison to the original signal. When continuous speech mixed with noise was processed, it was found that intelligibility improved a little at a threshold value of about 15-20% of peak value of signal. For words with longer pauses in between and mixed with noise, intelligibility was improved even more than in the previous case. Again the best threshold level was about 15-20%.

In another series, a prerecorded set of words with reverberating tails were processed and studied. It was found that reverberating tails were removed at the threshold level of about 35-40%. But at this level, intelligibility did not seem to improve. However, the quality was a little better than the original one. The best tradeoff was found at a threshold level of about 25% since intelligibility was increased and yet the reverberating tail was reduced to an extent it was detected only by an experienced listener.

III. RESULTS WITH SPECTRUM ANALYSIS

Several spectra were obtained using a spectrum analyzer in order to judge the performance of the system. Figure 24 shows the spectrum of a sine wave of 500 Hz. Figure 25 shows the spectrum of the same signal after fast expansion. Threshold value was about 20% of the peak value of the applied signal. Figure 26 shows the signal after center clipping. In Figure 27 the spectrum of the signal after fast expansion plus peak clipping is shown. For peak clipping the threshold value was about 80% of peak value whereas for fast expansion it was again at a level of about 20%. It can be seen that the level of harmonics was lowest in the case of fast expansion, since the amplitude of the frequencies in between the odd harmonics is almost zero. In the other cases the harmonics were higher plus all other frequencies in between have low amplitude.

Spectra of a speech signal before and after processing are shown in Figure 28, 29 and 30. Figure 29 was obtained when the threshold level was about 20%, while for Figure 30, it was about 35%. It can be seen that distortion is introduced during the process and Figure 30 looks more distorted than Figure 29.

Even though the performance of the system with speech signals was not very impressive, results obtained from using a pure tone are very encouraging. After going through all of these findings one can predict that there should be a considerable improvement on quality and intelligibility, if this system is expanded to four or eight channels.

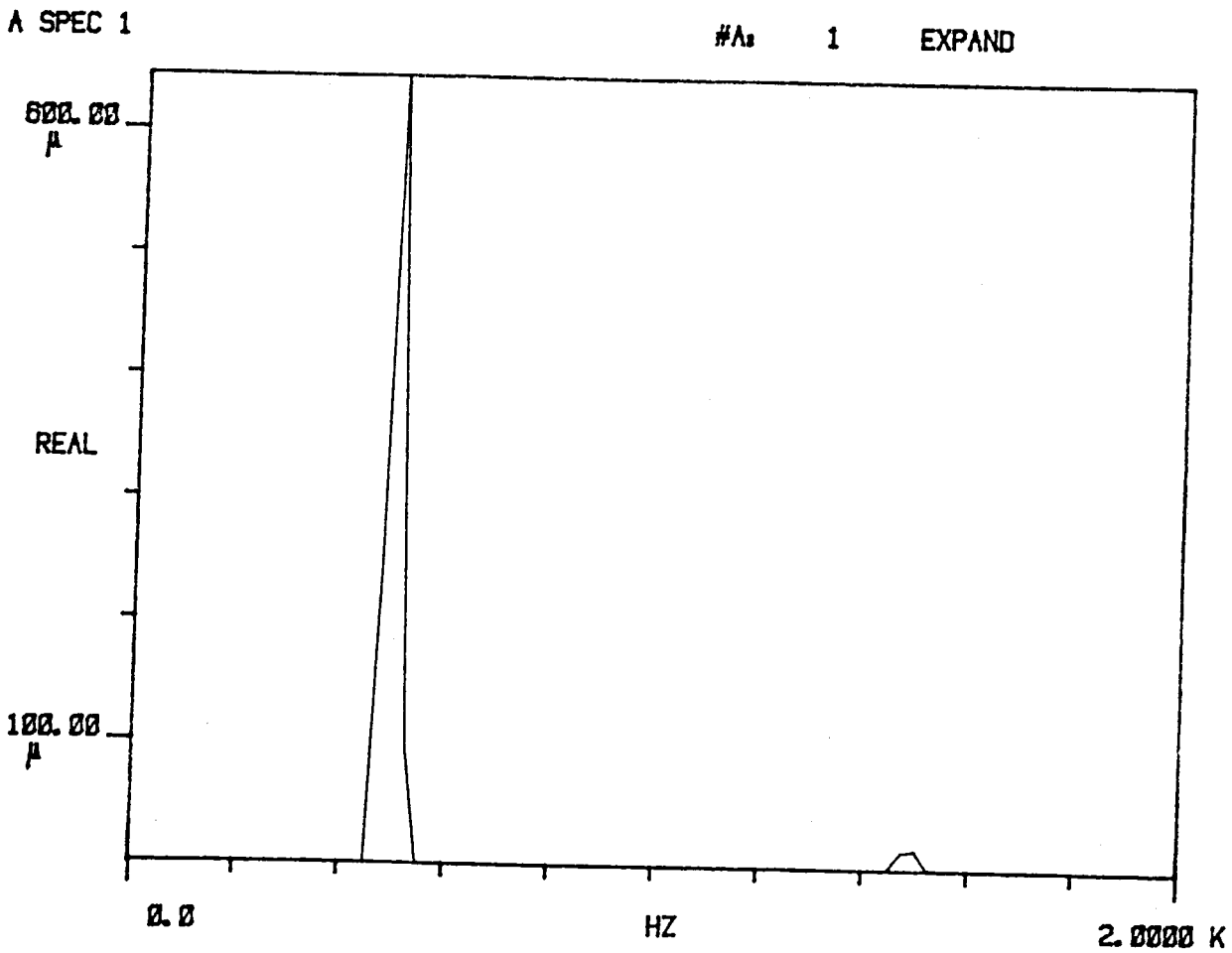


Figure 24. A spectrum of pure sine wave.

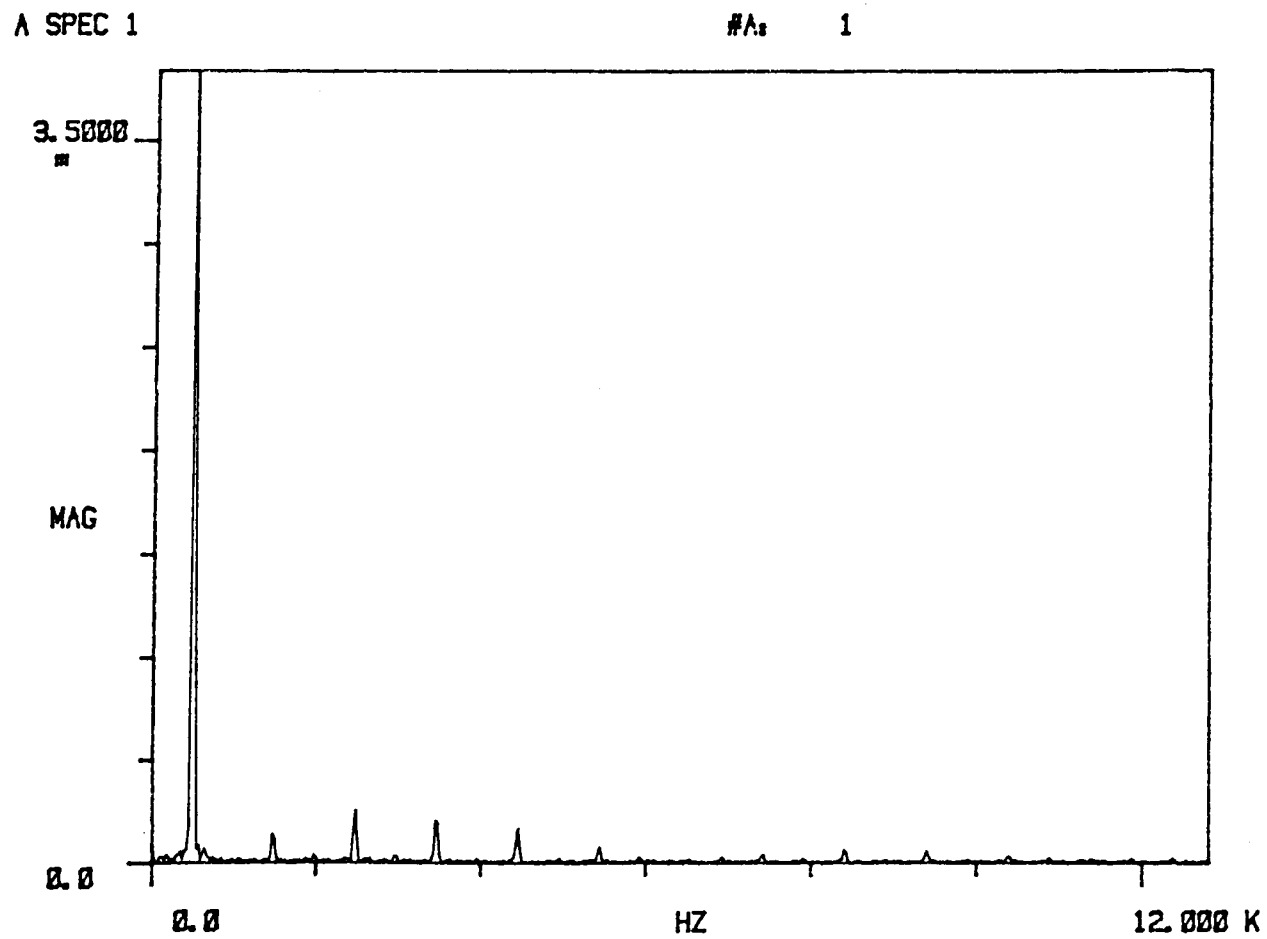


Figure 25. A spectrum of sine wave after fast expansion.

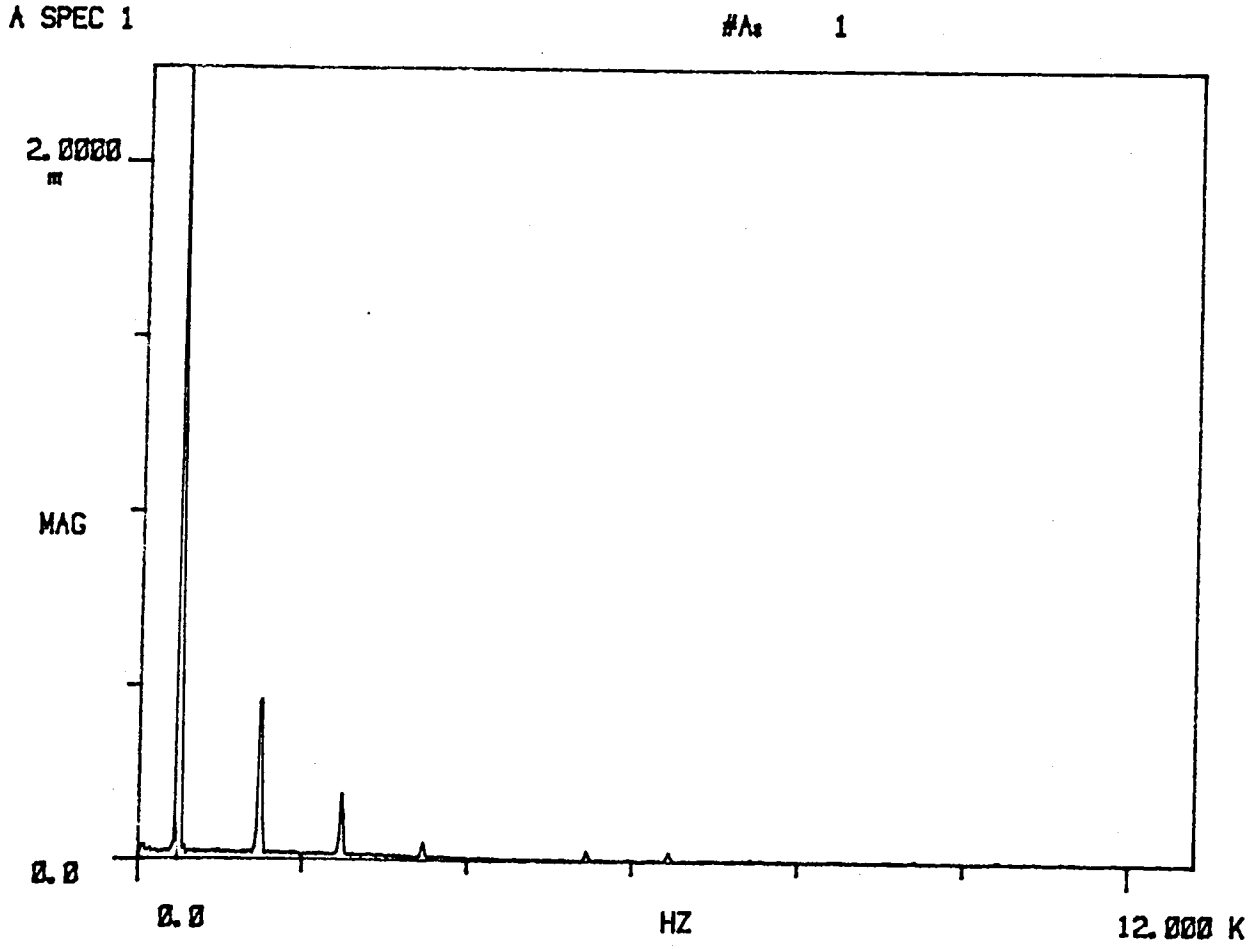


Figure 26. A spectrum of sine wave after center clipping.

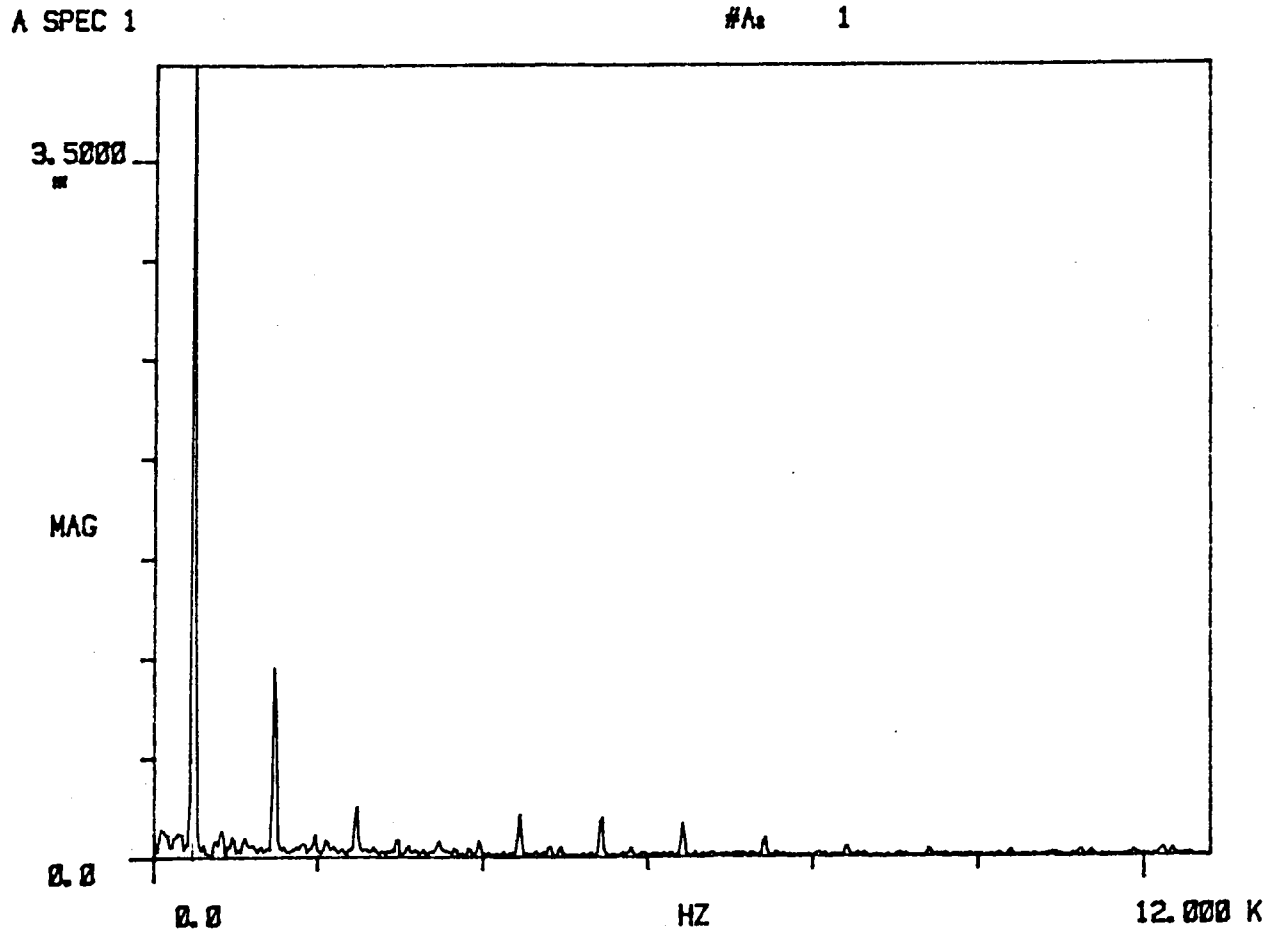


Figure 27. A spectrum of sine wave after fast expansion and peak clipping.

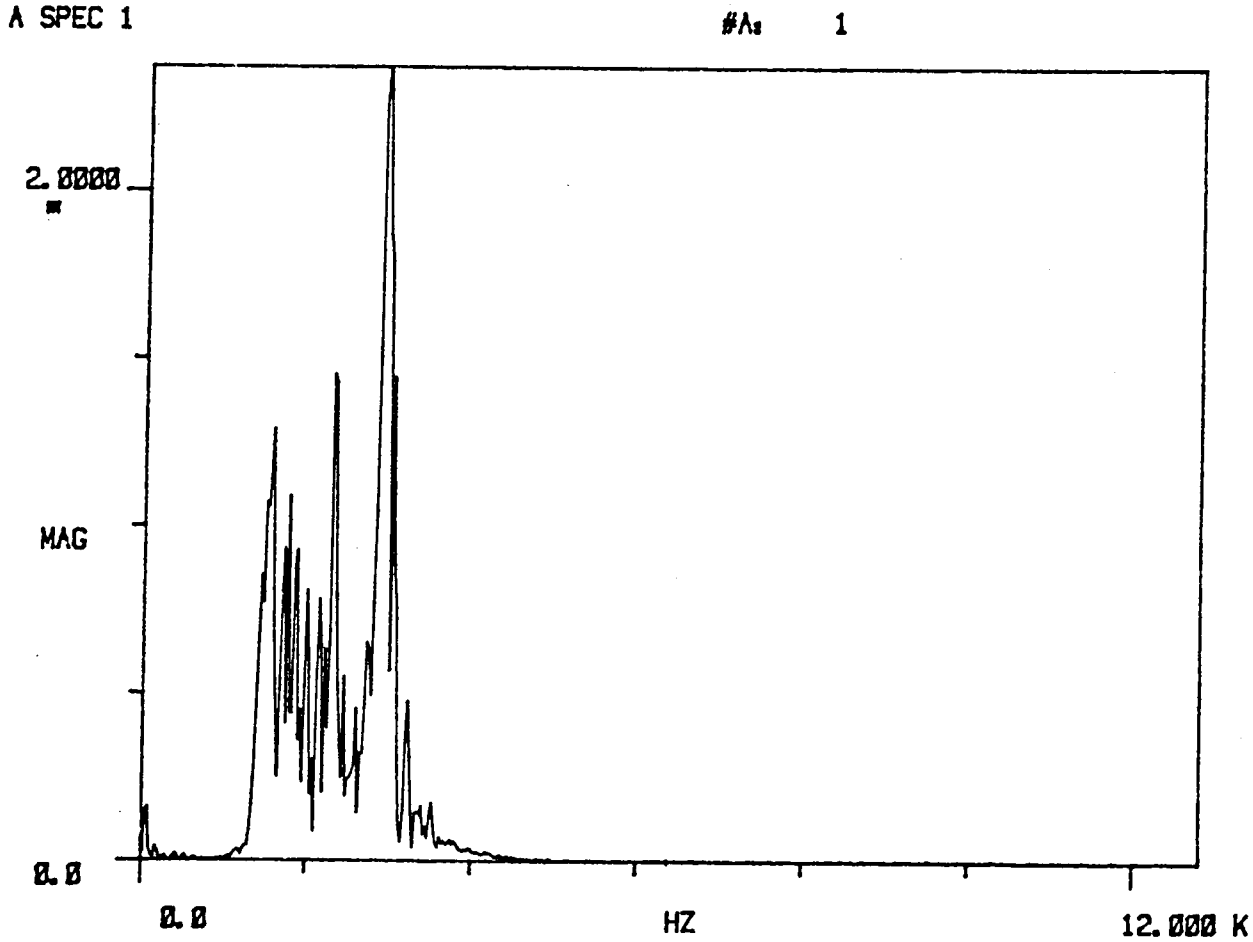


Figure 28. A spectrum of speech signal.

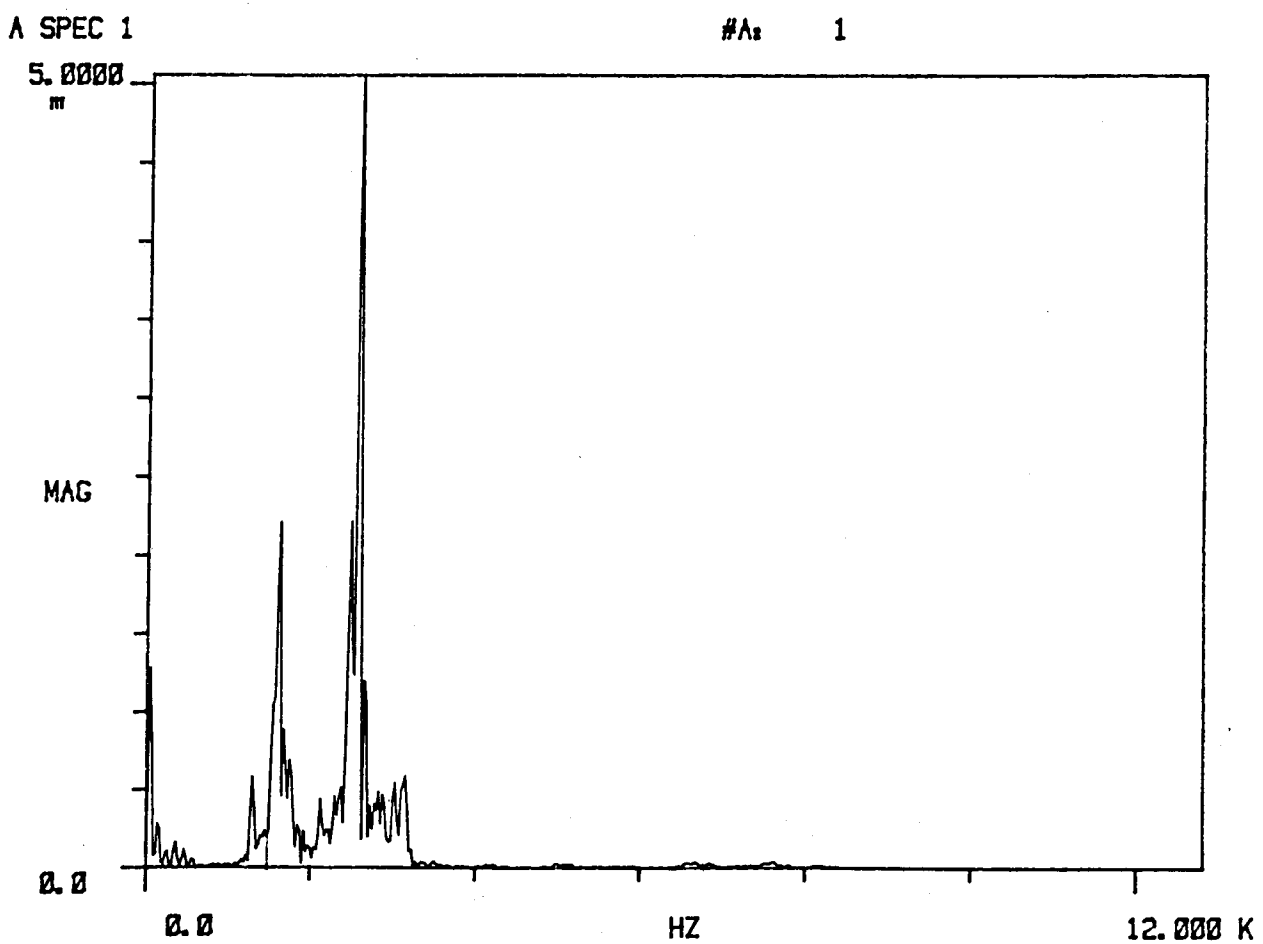


Figure 29. A spectrum of speech signal after fast expansion with a threshold value of 20%.

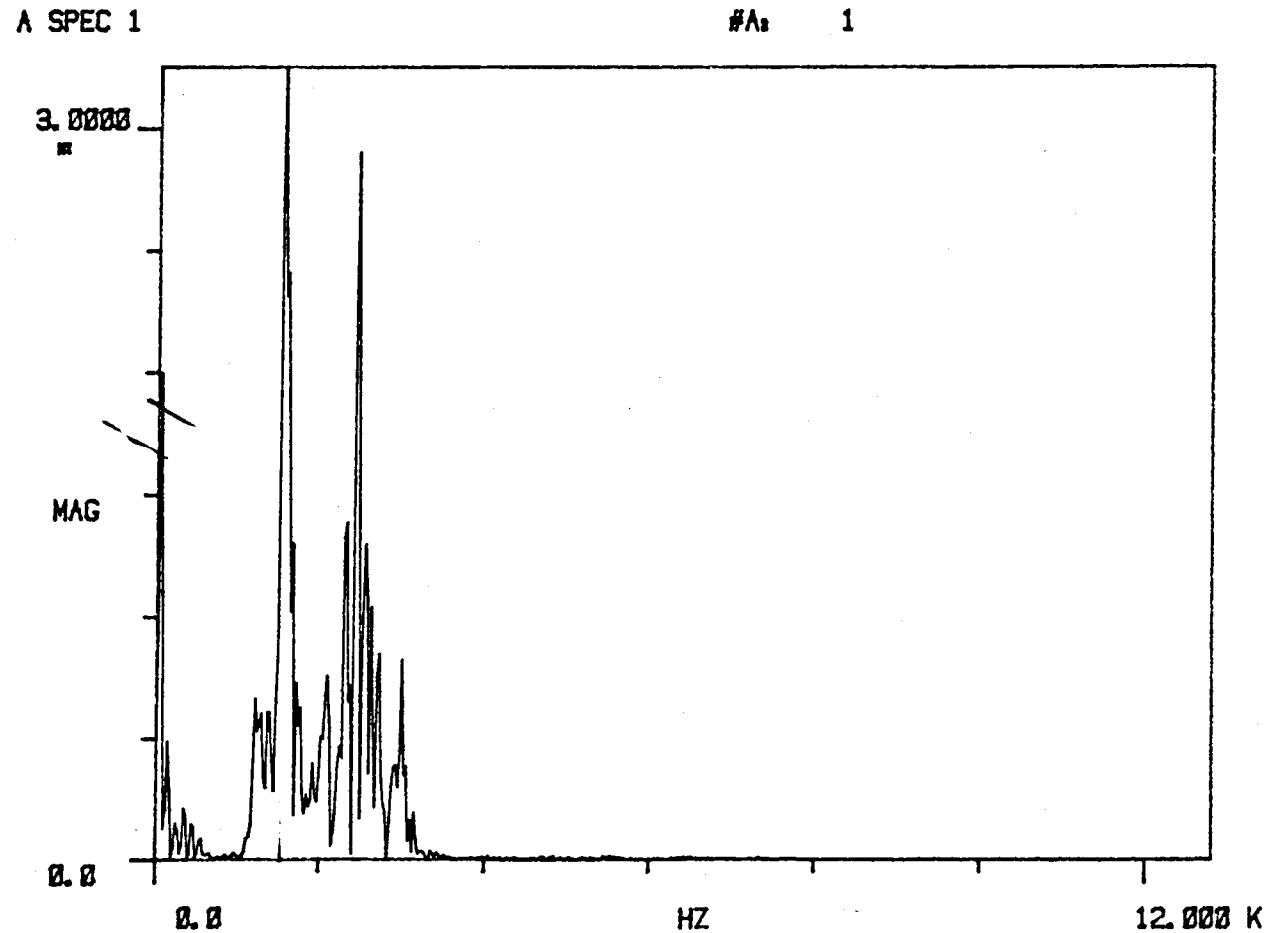


Figure 30. A spectrum of speech signal after fast expansion with a threshold value of 35%.

CHAPTER VI

SUMMARY AND CONCLUSIONS

A system using fast expansion was designed by the author for speech enhancement. Previous research (11) indicates the compression (the opposite transform to that of expansion) does not remove the low amplitude background noise nor the reverberating tails and the noise in pauses between words. On the other hand, center clipping does remove the low amplitude noise but it brings down considerably the signal to noise ratio for low energy consonants thus ruining the intelligibility of the speech. This minicomputer-based system for fast expansion not only removes the noise in pauses and reverberating tails but also preserves the S/N ratio.

Even though results are good, using fast expansion, they can be improved by expanding the system to more channels, preferably eight so that all the audible frequencies can be processed in narrow bands. Also investigation is needed to exploit its use in communication, acoustic research, hearing aids and other fields by itself or in conjunction with other processing techniques.

Another limitation is the available solid-state memory. In order not to lose any information, sampling of the speech signal must occur at least 16,000 times per second. That sets an upper limit of about 62 microseconds for acquiring and processing the sample completely. Therefore, any complicated processing such as digital filtering cannot be done in real time using this system. One solution is to forego

implementing the technique in the real time by storing the data in memory and then processing it afterwards. In our case, the solid-state memory available for storing the data is about 20K words which is insufficient to store speech signals lasting even two or three seconds. The winchester disk has a very large memory of 31.5M bytes but writing on the disk restricts the acquisition of information about the incoming signal. This is due to the need to stop writing on the disk after every other operation. The time taken to write a block on the disk is too long to be tolerated.

The following suggestions are made to improve the system performance.

1. Synchronize the tape recorders so that the same signal can be processed two or more times with different bands each time. These processed signals can be mixed or played together. This will produce the entire range of audible frequencies.

2. Inferface two available computers so that one can exclusively do the job of data conversion while the other can fetch data from there and store onto the disk for further processing. This requires a high speed parallel interfacing card and handshaking arrangemnets between the two computers. If this is done, it will eliminate most of the analog circuitry (and permit the luxury of employing 32 bands) by making operations like mixing, filtering, amplifying and attenuating possible in software. In this way, everything will be done digitally, but this will not be real-time processing.

For further research, it is suggested that the effect of fast expansion with and/or without other techniques (like peak clipping or compression) on handicapped hearing should be studied carefully.

If found useful, all of this circuitry and programming can be implemented in hardware. Today's technology has come to a point where all of the hardware necessary to perform these operations can be put in a small chip that can be placed inside a hearing aid.

LIST OF REFERENCES

LIST OF REFERENCES

1. Pridham, E. S., and P. L. Jensen, U. S. Patent 1,430,356, Sept. 26, 1922, applied for Aug. 17, 1917.

2. Miller, G. A., and F. M. Weiner, Transmission and reception of sounds under combat conditions, New York, Columbia University Press, (1948).

3. Tiffany, W. R., Phonetics Theory and Application, 2nd edition, McGraw Hill Book Company, New York, 145-153, (1977).

4. Ward, W. D., Effects of Noise on Speech Communication, The Application of Signal Processing Concepts to Hearing Aids, Edited by Paul Yanick, Jr., Grune and Stratton Publishers, New York, (1978).

5. Help for High Frequency Hearers, Time Magazine, September 13, vol 120 no. 11, (1982).

6. French, N. F., and J. C. Steinberg, Factors Governing the Intelligibility of Speech Sounds, J. Acoust. Soc. Am., 19, 90-119, (1947).

7. Ward, W. D., Further Observation on Contralateral Remote Masking and Related Phenomena, J. Acoust. Soc. Am., 42, 593-600, (1967).

8. The Radio Amateur's Handbook, American Radio League Publisher, 117, (1980).

9. Peebles, Peyton Z., Communication System Principles, 2nd edition, Addison Weseley Publishing Co., Massachusetts, 305, (1979).

10. Microcomputer Processor Handbook, Maynard Mass., Digital Equipment Corp., (1980).

11. Coln, Michael C., A Computer Controlled Multiband Amplitude Compressor, E. E. and C. S. Thesis, Mass. Inst. of Techn., (1979).

APPENDICES

APPENDIX A

ANALOG INPUT SPECIFICATIONS

Explanation of Input Types

Data Translation offers the user three distinct types of analog inputs. High level systems are available where the input range can be selected to be 0-5V, ±5V, 0-10V, or ±10V. We had ±10V

High Level Analog Input Specifications: Models DT1761)

NUMBER OF CHANNELS:	DT1761 8 DI or 16 SE with 2 D/A channels
MUX INPUT CHARACTERISTICS	Break before Make on new channel selection ±35 volts overvoltage protection
INPUT IMPEDANCE:	100 megohms in "OFF" or "ON" position of multiplexer
A/D CONVERTER RESOLUTION:	12 bits binary weighted
CODING:	or 2's complement.
RANGE:	±10 volts
PERFORMANCE:	
a. Gain Error	Adjustable to zero
b. Offset Error	Adjustable to zero
c. Linearity	± ½ LSB
d. Accuracy	± 0.03% FSR
e. Stability	± 25 ppm/ C°FSR
f. Throughput	100 KHz Option "C"

ANALOG OUTPUT SPECIFICATIONS

Dual Channel D/A Specifications

The DT1761 offers the user a sophisticated dual channel D/A Converter. The converter and its associated interface circuitry simplify a number of control tasks.

RESOLUTION:	12 bit binary weighted
LINEARITY:	± ½ LSB
RANGE:	± 10 volts
RELATIVE ACCURACY:	± 0.025%
SETTLING TIME:	0.1% in 1 uS, 0.01% in 3 uS driven into 50 ft. coaxial cable terminated with 470 ohms.
TEMPERATURE COEFFICIENT:	25 ppm/°C
Z AXIS CONTROL:	The DT1761 contains all the control circuitry for Z axis and scope control mode bits.
Z OUTPUT (INTENSIFY):	TTL compatible pulse driven into 50 ohm termination. Normally LOW (max. 0.8 volts), generates HIGH (min. 2.4 volts) pulse on intensify.
Z RISETIME:	100 nS into 50 ft. of coaxial cable terminated with 470 ohms.
Z PULSE WIDTH:	0.5 uS or 5 uS jumper selectable. External capacitor supplied by the user can lengthen the pulse width from 0.2 uS to 100 uS.
SET-UP DELAY:	3 uS or 70 uS jumper selectable. External capacitor supplied by the user can lengthen the delay from 1 uS to 0.5 mS.
GAIN & OFFSET:	Both outputs adjustable with external pot.

MODE BITS:	Two input bits are decoded to four mode control output lines. the output lines are LOW true and can drive up to 10 standard TTL loads.

APPENDIX B

```
                                                  ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
                                                  ;                                            ;
                                                  ;                    *EGIT1*                 ;
                                                  ;                                            ;
                                                  ;THIS PROGRAM WAS WRITTEN TO ASK THE         ;
                                                  ;USER FOR THRESHOLD VALUES, THEN             ;
                                                  ;GETTING THOSE VALUES AND STORING            ;
                                                  ;IT PROPERLY                                 ;
                                                  ;                                            ;
                                                  ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;



                                            .MCALL  .PRINT, .TTYIN, .EXIT
                                            .MCALL  .REGDEF, ..V2..
000000                               ST:    .PRINT  #M1             ;GIVE THE MESSAGE OF HELP
000006  000414                              BR      Q1              ;SKIP THE MESSAGE
000010     106     117     122       M1:    .ASCIZ/FOR HELP READ PROJ.HLP./
000013     040     110     105
000016     114     120     040
000021     122     105     101
000024     104     040     120
000027     122     117     112
000032     056     110     114
000035     120     056     000
                                            .EVEN
000040                               Q1:    .PRINT  #M2             ;ASK THE USER TO INPUT THRESHOLD VALUE
000046  000430                              BR      Q2              ;SKIP THE MESSAGE JUST ASKED
000050     111     116     120       M2:    .ASCIZ/INPUT POSITIVE VALUE  FROM 0 TO 1000 FOR CH.0./
000053     125     124     040
000056     120     117     123
000061     111     124     111
000064     126     105     040
000067     126     101     114
000072     125     105     040
000075     040     106     122
000100     117     115     040
000103     060     040     124
000106     117     040     061
000111     060     060     060
```

```
    000114      040     106     117
    000117      122     040     103
    000122      110     056     060
    000125      056     000
22                                          .EVEN                   ;MAKE LOCATIONS EVEN
23  000130  004767  000376          Q2:     JSR     PC,INPUT                ;GO AND GET THE VALUE
24  000134  005001                          CLR     R1              ;CLEAR REGISTER R1
25  000136  062701  000002          1$:     AUD     #2,R1           ;CHANGE IT INTO 2'S COMPLIMENT
26  000142  005300                          DEC     R0              ;DECREMENT R0 TO SEE PROCESS COMPLETED
27  000144  001374                          BNE     1$              ;IF NO GO BACK AND DO IT AGAIN
28  000146  010167  000346                  MOV     R1,PLOW         ;STORE THIS VALUE IN LOCATION DESIGNATED
29  000152                                   .PRINT  #M3             ;ASK THE USER FOR NEXT VALUE
30  000160  000427                          BR      Q3              ;SKIP THE MESSAGE
31  000162      111     116     120  M3:     .ASCIZ/INPUT POSITIVE VALUE FROM 0 TO 1000 FOR CH.6./
    000165      125     124     040
    000170      120     117     123
    000173      111     124     111
    000176      126     105     040


    000201      126     101     114
    000204      125     105     040
    000207      106     122     117
    000212      115     040     117
    000215      040     124     117
    000220      040     061     060
    000223      060     060     040
    000226      106     117     122
    000231      040     103     110
    000234      056     066     056
    000237      000
32                                          .EVEN                   ;MAKE THE LOCATIONS EVEN
33  000240  004767  000266          Q3:     JSR     PC,INPUT        ;GO AND GET THE VALUE FROM INPUT
34  000244  005001                          CLR     R1              ;CLEAR REGISTER TO STORE NEW VALUE
35  000246  062701  000002          1$:     AUD     #2,R1           ;CHANGE IT TO REQUIRED VALUE
36  000252  005300                          DEC     R0              ;DECREMENT REGISTER R0
37  000254  001374                          BNE     1$              ;IF PROCESS NOT COMPLETE GO BACK
38  000256  010167  000240                  MOV     R1,PHIGH        ;STORE THE VALUE AT PROPER LOCATION
39  000262                                   .PRINT  #M4             ;ASK THE USER FOR NEGATIVE THRESHOLD
40  000270  000427                          BR      Q4              ;SKIP THE MESSAGE
41  000272      111     116     120  M4:     .ASCIZ/INPUT NEGATIVE VALUE FROM 0 TO 1000 FOR CH.0/
```

```
    000275      125      124      040
    000300      116      105      107
    000303      101      124      111
    000306      126      105      040
    000311      126      101      114
    000314      125      105      040
    000317      106      122      117
    000322      115      040      060
    000325      040      124      117
    000330      040      061      060
    000333      060      060      040
    000336      106      117      122
    000341      040      103      110
    000344      056      060      000
42                                          .EVEN                   ;MAKE THE LOCATIONS EVEN
43  000350  004767  000156          Q4:     JSR     PC,INPUT        ;GO AND GET THE RESULT
44  000354  005001                          CLR     R1              ;CLEAR R1 FOR FURTHER USE
45  000356  062701  000002          1$:     ADD     #2.,R1          ;CHANGE THE INPUT TO DESIRED AND STORE IN R1
46  000362  005300                          DEC     R0              ;DECREMENT R0
47  000364  001374                          BNE     1$              ;IF NOT COMPLETE GO BACK
48  000366  005401                          NEG     R1              ;MAKE THIS VALUE NEGATIVE
49  000370  042701  170000                  BIC     #170000,R1      ;MASK THE BITS 12 TO 15
50  000374  010167  000124                  MOV     R1,NLOW         ;STORE THE VALUE AT PROPER LOCATION
51  000400                                  .PRINT  #M5             ;ASK THE USER FOR NEXT THRESHOLD LEVEL
52  000406  000427                          BR      Q5              ;SKIP THE MESSAGE GIVEN
53  000410      111      116      120  M5:     .ASCIZ/INPUT NEGATIVE VALUE FROM 0 TO 1000 FOR CH.6./
    000413      125      124      040
    000416      116      105      107
    000421      101      124      111
    000424      126      105      040
    000427      126      101      114
    000432      125      105      040
    000435      106      122      117
    000440      115      040      060
    000443      040      124      117
    000446      040      061      060
    000451      060      060      040
    000454      106      117      122
    000457      040      103      110
```

```
        000462      056      066      056
        000465      000
     54                                          .EVEN                   ;MAKE THE LOCATIONS EVEN
     55 000466  004767  000040           Q5:     JSR     PC,INPUT        ;GO AND GET THE VALUES
     56 000472  005001                           CLR     R1              ;CLEAR REGEISTER ONE
     57 000474  062701  000002           1$:     ADD     #2.,R1          ;CHANGE THE INPUT VALUE TO DESIRED ONE
     58 000500  005300                           DEC     R0              ;DECREMENT R0 TO SEE THE COMPLETION
     59 000502  001374                           BNE     1$              ;IF PROCESS NOT COMPLETE GO BACK
     60 000504  005401                           NEG     R1              ;NEGATE THE VALUE
     61 000506  042701  170000                   BIC     #170000,R1      ;MASK THE BITS FROM 12 TO 15
     62 000512  010167  000010                   MOV     R1,NHIGH        ;STORE THE VALUE AT PROPER LOCATION
     63 000516  000404                           BR      NEX             ;GET OUT FROM HERE AND GO TO NEXT
     64 000520  000000                   PLOW:   0                       ;MEMORY LOCATIONS RESERVED  FOR
     65 000522  000000                   PHIGH:  0                       ;THRESHOLD VALUES
     66 000524  000000                   NLOW:   0
     67 000526  000000                   NHIGH:  0
     68 000530                           NEX:    .EXIT                   ;DUMMY PROCESS
     69 000532  000207                   INPUT:  RTS     PC              ;DUMMY SUBROUTINE PUT HERE
     70         000000'                          .END    ST              ;STARTING LOCATION OF PROGRAM
SYMBOL TABLE

INPUT   000532R         M4      000272R         NLOW    000524R         Q1      000040R         Q4      000350R
M1      000010R         M5      000410R         PHIGH   000522R         Q2      000130R         Q5      000460R
M2      000050R         NEX     000530R         PLOW    000520R         Q3      000240R         ST      000000R
M3      000162R         NHIGH   000526R

. ABS.  000000      000
        000534      001
ERRORS DETECTED:  0

VIRTUAL MEMORY USED:  8448 WORDS  ( 33 PAGES)
DYNAMIC MEMORY AVAILABLE FOR  48 PAGES
GET,GET=GET
```

```
                                                ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
                                                ;                                           ;
                                                ;                 **INPUT**                 ;
                                                ;                                           ;
                                                ;THIS SUBROUTINE GETS THE VLUES FROM USER'S ;
                                                ;TERMINAL. IF ANY ILLEGAL CHARACTER OR MORE ;
                                                ;THAN FOUR DIGITS ARE TYPED, IT GIVES AN    ;
                                                ;ERROR MESSAGE AND CONTROL IS TRANSFERED    ;
                                                ;AGAIN TO THE BEGINNING OF THIS SUBROUTINE. ;
                                                ;OTHERWISE VALUE ENTERED WILL BE STORED     ;
                                                ;PROPERLY AND CONTROL WILL BE TRANSFERED    ;
                                                ;FORM THE SUBROUTINE. IF RUB KEY IS HIT AT  ;
                                                ;ANY MOMENT IT DELETS THE LAST CHARACTER    ;
                                                ;ENTERED BY USER.                           ;
                                                ;                                           ;
                                                ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;



                                        .MCALL  .PRINT, .TTYIN, .EXIT
                                        .MCALL  .REGDEF, ..V2..
000000  012701  000242'         INPUT:  MOV     #TEMP2,R1       ;STORE LOCATION OF TEMP2 IN R1
000004                          1$:     .TTYIN                  ;GET THE CHARACTER FROM TERMINAL
000010  042700  177600                  BIC     #177600,R0      ;MASK BITS 6 TO 15
000014  120027  000015                  CMPB    R0,#15          ;WAS CARRIAGE RETURN HIT
000020  001771                          BEQ     1$              ;IF YES GO AND WAIT FOR LINE FEED
000022  120027  000012                  CMPB    R0,#12          ;CHECK FOR LINE.FEED
000026  001425                          BEQ     10$             ;IF YES GO TO STORE THE VALUE
000030  120027  000177                  CMPB    R0,#177         ;LOOK FOR THE RUB KEY
000034  001415                          BEQ     2$              ;IF YES GO AND DELETE THE LAST CHARACTER
000036  120027  000060                  CMPB    R0,#60          ;IS IT LEGAL CHARACTER
000042  103437                          BLO     20$             ;IF SURE IT IS ILLEGAL GIVE ERROR MESSAGE
000044  120027  000071                  CMPB    R0,#71          ;CHECK AGAIN FOR LEGAL CHARACTER
000050  101034                          BHI     20$             ;IF IT IS ILLEGAL GIVE THE ERROR MESSAGE
000052  162700  000060                  SUB     #60,R0          ;SUBTRACT SIXTY FROM ASCI TO GET THE NUMBER
000056  110021                          MOVB    R0,(R1)+        ;STORE THIS IN LOCATION RESERVED FOR IT
000060  020127  000247'                 CMP     R1,#TEMP2+5     ;CHECK FOR THE MAXIMUM NUMBER ALLOWED
000064  103026                          BHIS    20$             ;IF MORE GIVE THE ERROR MESSAGE
000066  000746                          BR      1$              ;IF NOT REPEAT THE PROCESS
000070  020127  000242'         2$:     CMP     R1,#TEMP2       ;WAS THERE ANY CHARCTER BEFORE
```

```
40 000074  101743                        BLOS    1$              ;IF NOT DONE'T CARE
41 000076  105301                        DECB    R1              ;OTHERWISE DELETE THE LAST CHARACTER
42 000100  000741                        BR      1$              ;GO AND GET THE VALUE FROM TERMINAL
43 000102  005000                10$:    CLR     R0              ;CLEAR THE REGISTER R0
44 000104  012702  000242'               MOV     #TEMP2,R2       ;STORE THE LOCATION OF TEMP2 IN R2
45 000110  005004                        CLR     R4              ;CLEAR THE REGISTER R4
46 000112  020201                11$:    CMP     R2,R1           ;IS JOB COMPLETED
47 000114  103011                        BHIS    30$             ;IF YES RETURN FROM THIS SUBROUTINE
48 000116  010003                        MOV     R0,R3           ;STORE THE CONTENTS OF R0 IN R3
49 000120  006303                        ASL     R3              ;MAKE ROOM AVAILABL
50 000122  006303                        ASL     R3              ;FOR NEXT INPUT
51 000124  006303                        ASL     R3              ;CHARACTER
52 000126  060003                        ADD     R0,R3           ;ADD CONTENTS OF R0 TO R3
53 000130  060300                        ADD     R3,R0           ;ADD CONTENTS OF R3 TO R0
54 000132  112204                        MOVB    (R2)+,R4        ;MOVE NEXT LOCATION TO R4
55 000134  060400                        ADD     R4,R0           ;ADD CONTENTS OF R4 TO R0
56 000136  000765                        BR      11$             ;GO AND REPEAT THE PROCESS
57 000140  000207                30$:    RTS     PC              ;RETURN FROM THE SUBROUTINE
58 000142                        20$:    .TTYIN                  ;GET THE VALUE FROM TERMINAL
59 000146  042700  177600                BIC     #177600,R0      ;MASK THE BITS FROM 7 TO 15
60 000152  020027  000012                CMP     R0,#12          ;IS IT LINE FEED
61 000156  001371                        BNE     20$             ;WAIT FOR LINE FEED
62 000160                                .PRINT  #MSG            ;PRINT THE ERROR MESSAGE
63 000166  000704                        BR      INPUT           ;GO BACK TO GET THE VALUE FROM TERMINAL
64 000170  000000                TEMP1:  0
65 000172     015      012      131     MSG:    .ASCIZ<15><12>/YOU INPUT ILLEGAL NUMBER, TRY AGAIN!!/
   000175     117      125      040
   000200     111      116      120
   000203     125      124      040
   000206     111      114      114
   000211     105      107      101
   000214     114      040      116
   000217     125      115      102
   000222     105      122      054
   000225     040      124      122
   000230     131      040      101
   000233     107      101      111
   000236     116      041      041
   000241     000
```

```
     66 000242                              TEMP2:  .BLKB   6
     67                                             .EVEN                   ;MAKE THE LOCATIONS EVEN
     68          000000                             .END    INPUT           ;STARTING LOCATION OF PROGRAM
SYMBOL TABLE

INPUT   000000R         MSG     000172R         TEMP1   000170R         TEMP2   000242R

  ABS   000000      000
        000250      001
ERRORS DETECTED:  0

VIRTUAL MEMORY USED:  8448 WORDS  ( 33 PAGES)
DYNAMIC MEMORY AVAILABLE FOR  48 PAGES
INPUT,INPUT=INPUT
```

```
                                                            ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
                                                            ;                                            ;
                                                            ;               **FEXP**                     ;
                                                            ;                                            ;
                                                            ;THIS PROGRAM WAS WRITTEN TO DO THE          ;
                                                            ;FAST EXPANSION. THIS ALLOWS TWO             ;
                                                            ;CHANNELS TO BE CONTROLLED INDEPENDENTLY.    ;
                                                            ;HIGHER FREQUENCY CHANNEL IS SAMPLED         ;
                                                            ;AND PROCESSED TWICE AS FAST AS OF           ;
                                                            ;LOWER FREQUENCY.                            ;
                                                            ;                                            ;
                                                            ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;


                                                  .MCALL  .REGDEF,.EXIT
                                                  .MCALL  ..V2..
         177000                          ADCSR=177000                      ;LOCATION OF AD CONTROL STATUS REGISTER
000000   112767   000000   177001        NEX:     MOVB    #0,ADCSR+1       ;START CONVERSION AT CHANNEL 0
000006   016701   000256                          MOV     PLOW,R1          ;MOVE THREESHOLD FOR CH 0 TO R1
000012   016703   000254                          MOV     PHIGH,R3         ;MOVE +VE THRESHOLD FOR CH6 TO R3
000016   016702   000252                          MOV     NLOW,R2          ;MOVE -VE THRESHOLD FOR CH0 TO R2
000022   016704   000250                          MOV     NHIGH,R4         ;MOVE -VE THRESHOLD FOR CH6 TO R4
000026   105737   177000                 PROCS:   TSTB    @#ADCSR          ;IS DATA READY
000032   100375                                   BPL     PROCS            ;NO WAIT FOR IT
000034   013700   177002                          MOV     @#ADCSR+2,R0     ;MOVE THE DATA TO REGISTER R0
000040   112767   000006   177001                 MOVB    #6,ADCSR+1       ;START CONVERSION AT CHANNEL 6
000046   042700   170000                          BIC     #170000,R0       ;MASK THE UNNESSARY BITS
000052   020027   003777                          CMP     R0,#3777         ;IS IT POSITIVE VALUE
000056   103406                                   BLO     1$               ;IF YES GO TO PROCESS POSITIVE HALF CYCLE
000060   020002                                   CMP     R0,R2            ;DOES IT FALL WITHIN THE THRESHOLD BAND
000062   103401                                   BLO     2$               ;NO GO FOR OUTPUT
000064   005000                                   CLR     R0               ;BRING DOWN THE OUTPUT VALUE TO ZERO
000066   010067   177002                 2$:      MOV     R0,ADCSR+2       ;OUTPUT THE VALUE AT CHANNEL Y
000072   000405                                   BR      PROCS1           ;GO FOR NEXT SIGNAL PROCESSING
000074   020001                          1$:      CMP     R0,R1            ;IS IT ABOVE THRESHOLD VALUE
000076   101001                                   BHI     3$               ;YES GO TO OUTPUT
000100   005000                                   CLR     R0               ;BRING DOWN THE OUTPUT TO ZERO
000102   010067   177002                 3$:      MOV     R0,ADCSR+2       ;OUTPUT THE DATA AT CHANNEL Y
000106   105737   177000                 PROCS1:  TSTB    @#ADCSR          ;IS DATA READY
```

```
 40 000112  100375                          BPL     PROCS1          ;IF NOT WAIT FOR IT
 41 000114  013700  177002                  MOV     @#ADCSR+2,R0    ;STORE THE DATA IN REGISTER R0
 42 000120  112767  000006  177001          MOVB    #6,ADCSR+1      ;START CONVERSION AT CHANNEL 6
 43 000126  042700  170000                  BIC     #170000,R0      ;MASK THE FOUR HIGHEST BITS
 44 000132  020027  003777                  CMP     R0,#3777        ;IS IT POSITIVE VALUE
 45 000136  103410                          BLO     1$              ;IF YES GO FOR APPROPRIATE PROCESSING
 46 000140  020004                          CMP     R0,R4           ;IS IT WITHIN THE THRESHOLD VALUE
 47 000142  103401                          BLO     2$              ;NO GO FOR OUTPUT
 48 000144  005000                          CLR     R0              ;OTHERWISE CLEAR THE REGISTER R0
 49 000146  052700  100000          2$:     BIS     #100000,R0      ;SET BIT 15 FOR OUTPUT AT PORT X
 50 000152  010067  177002                  MOV     R0,ADCSR+2      ;MOVE THE DATA TO OUTPUT BUFFER
 51 000156  000407                          BR      PROCS2          ;GO FOR NEXT PROCESS
 52 000160  020003          1$:             CMP     R0,R3           ;IS IT WITH IN THE THRESHOLD LEVEL
 53 000162  101001                          BHI     3$              ;NO GO TO OUTPUT
 54 000164  005000                          CLR     R0              ;CLEAR REGISTER R0
 55 000166  052700  100000          3$:     BIS     #100000,R0      ;SET BIT 15 FOR OUTPUT AT PORT X
 56 000172  010067  177002                  MOV     R0,ADCSR+2      ;MOVE THE DATA TO OUTPUT BUFFER
 57 000176  105737  177000          PROCS2: TSTB    @#ADCSR         ;IS DATA READY
 58 000202  100375                          BPL     PROCS2          ;IF NOT WAIT FOR IT
 59 000204  013700  177002                  MOV     @#ADCSR+2,R0    ;MOVE THE DATA TO REGISTER R0
 60 000210  112767  000000  177001          MOVB    #0,ADCSR+1      ;START CONVERSION AT CHANNEL0
 61 000216  042700  170000                  BIC     #170000,R0      ;MASK THE BITS FROM 12 TO 15
 62 000222  020027  003777                  CMP     R0,#3777        ;IS IT POSITIVE
 63 000226  103410                          BLO     1$              ;IF YES GO FOR IT
 64 000230  020004                          CMP     R0,R4           ;IS IT WITHIN THE THRESHOLD LEVEL
 65 000232  103401                          BLO     2$              ;IF NO GO FOR OUTPUT
 66 000234  005000                          CLR     R0              ;BRING DOWN THE OUTPUT VALUE TO ZERO
 67 000236  052700  100000          2$:     BIS     #100000,R0      ;SET THE BIT 15 FOR OUT PUT
 68 000242  010067  177002                  MOV     R0,ADCSR+2      ;MOVE THE DATA TO OUTPUT BUFFER
 69 000246  000667                          BR      PROCS           ;REPEAT THE ENTIRE PROCESS AGAIN
 70 000250  020003          1$:             CMP     R0,R3           ;IS IT WITH IN THHE THRESHOLD LEVEL
 71 000252  101001                          BHI     3$              ;IF NOT GO FOR OUTPUT
 72 000254  005000                          CLR     R0              ;CLEAR THE REGISTER R0
 73 000256  052700  100000          3$:     BIS     #100000,R0      ;SET BIT 15 FOR OUTPUT
 74 000262  010067  177002                  MOV     R0,ADCSR+2      ;MOVE THE DATA TO OUTPUT BUFFER
 75 000266  000657                          BR      PROCS           ;REPEAT THE ENTIRE PROCESS AGAIN
 76 000270  000000          PLOW:   0
 77 000272  000000          PHIGH:  0
 78 000274  000000          NLOW:   0
```

```
     79 000276  000000                    NHIGH:  0
     80         000000                     END    NEX             ;STARTING LOCATION OF PROGRAM
.MAIN.  MACRO V04.00   00:27:00 PAGE 1-2
SYMBOL TABLE

ADCSR = 177000          NHIGH   000276R         PHIGH   000272R         PROCS   000026R         PROCS2  000176R
NEX     000000R         NLOW    000274R         PLOW    000270R         PROCS1  000106R

  ABS.  000000      000
        000300      001
ERRORS DETECTED:  0

VIRTUAL MEMORY USED:  8192 WORDS  ( 32 PAGES)
DYNAMIC MEMORY AVAILABLE FOR  48 PAGES
FEXP,FEXP=FEXP
```

```
                                                  ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
                                                  ;                                               ;
                                                  ;                   **CCLIP**                   ;
                                                  ;                                               ;
                                                  ;THIS PROGRAM IMPLEMENTS THE CENTER CLIPPING    ;
                                                  ;TECHNIQUE. IT CENTER CLIPS THE INPUT SIGNAL.   ;
                                                  ;THRESHOLD VALUE FOR BOTH THE CHANNELS CAN      ;
                                                  ;BE ADJUSTED INDEPENDENTLY.                     ;
                                                  ;                                               ;
                                                  ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;


                                            .MCALL  .REGDEF,.EXIT
                                            .MCALL  ..V2..
        177000                      ADCSR=177000                    ;LOCATION OF AD CONTROL STATUS REGISTER
000000  112767  000000  177001  NEX:    MOVB    #0,ADCSR+1          ;START CONVERSION AT CHANNEL 0
000006  016701  000316                  MOV     PLOW,R1             ;MOVE THREESHOLD FOR CH 0 TO R1
000012  016703  000314                  MOV     PHIGH,R3            ;MOVE +VE THRESHOLD FOR CH6 TO R3
000016  016702  000312                  MOV     NLOW,R2             ;MOVE -VE THRESHOLD FOR CH0 TO R2
000022  016704  000310                  MOV     NHIGH,R4            ;MOVE -VE THRESHOLD FOR CH6 TO R4
000026  105737  177000          PROCS:  TSTB    @#ADCSR             ;IS DATA READY
000032  100375                          BPL     PROCS               ;NO WAIT FOR IT
000034  013700  177002                  MOV     @#ADCSR+2,R0        ;MOVE THE DATA TO REGISTER R0
000040  112767  000006  177001          MOVB    #6,ADCSR+1          ;START CONVERSION AT CHANNEL 6
000046  042700  170000                  BIC     #170000,R0          ;MASK THE UNNESSARY BITS
000052  020027  003777                  CMP     R0,#3777            ;IS IT POSITIVE VALUE
000056  103412                          BLO     1$                  ;IF YES GO TO PROCESS POSITIVE HALF CYCLE
000060  020002                          CMP     R0,R2               ;DOES IT FALL WITHIN THE THRESHOLD BAND
000062  103404                          BLO     2$                  ;NO GO FOR OUTPUT
000064  005000                          CLR     R0                  ;BRING DOWN THE OUTPUT VALUE TO ZERO
000066  010067  177002                  MOV     R0,ADCSR+2          ;OUTPUT THE VALUE AT CHANNEL Y
000072  000415                          BR      PROCS1              ;GO FOR NEXT PROCESS
000074  160200                  2$:     SUB     R2,R0               ;SUBTRACT THE THRESHOLD VALUE
000076  010067  177002                  MOV     R0,ADCSR+2          ;MOVE THE DATA TO OUTPUT BUFFER
000102  000411                          BR      PROCS1              ;GO FOR NEXT SIGNAL PROCESSING
000104  020001                  1$:     CMP     R0,R1               ;IS IT ABOVE THRESHOLD VALUE
000106  101004                          BHI     3$                  ;YES GO TO OUTPUT
000110  005000                          CLR     R0                  ;BRING DOWN THE OUTPUT TO ZERO
000112  010067  177002                  MOV     R0,ADCSR+2          ;OUTPUT THE DATA AT CHANNEL Y
```

```
40 000116  000403                          BR      PROCS1          ;GO FOR NEXT PROCESS
41 000120  160100                  3$:     SUB     R1,R0           ;SUBTRACT THE THRESHOLD VALUE
42 000122  010067  177002                   MOV     R0,ADCSR+2      ;MOVE THE DATA TO OUTPUT BUFFER
43 000126  105737  177000          PROCS1: TSTB    @#ADCSR         ;IS DATA READY
44 000132  100375                          BPL     PROCS1          ;IF NOT WAIT FOR IT
45 000134  013700  177002                   MOV     @#ADCSR+2,R0    ;STORE THE DATA IN REGISTER R0
46 000140  112767  000006  177001          MOVB    #6,ADCSR+1      ;START CONVERSION AT CHANNEL 6
47 000146  042700  070000                   BIC     #70000,R0       ;MASK THE FOUR HIGHEST BITS
48 000152  020027  003777                   CMP     R0,#3777        ;IS IT POSITIVE VALUE
49 000156  103412                          BLO     1$              ;IF YES GO FOR APPROPRIATE PROCESSING
50 000160  020004                          CMP     R0,R4           ;IS IT WITHIN THE THRESHOLD VALUE
51 000162  103404                          BLO     2$              ;NO GO FOR OUTPUT
52 000164  005000                          CLR     R0              ;OTHERWISE CLEAR THE REGISTER R0
53 000166  010067  177002                   MOV     R0,ADCSR+2      ;MOVE THE DATA TO OUTPUT BUFFER
54 000172  000415                          BR      PROCS2          ;GO FOR NEXT PROCESS
55 000174  160400                  2$:     SUB     R4,R0           ;SUBTRACT THE THRESHOLD VALUE
56 000176  010067  177002                   MOV     R0,ADCSR+2      ;MOVE THE DATA TO OUTPUT BUFFER
57 000202  000411                          BR      PROCS2          ;GO FOR NEXT PROCESS
58 000204  020003                  1$:     CMP     R0,R3           ;IS IT WITH IN THE THRESHOLD LEVEL
59 000206  101004                          BHI     3$              ;NO GO TO OUTPUT
60 000210  005000                          CLR     R0              ;CLEAR REGISTER R0
61 000212  010067  177002                   MOV     R0,ADCSR+2      ;MOVE THE DATA TO OUTPUT BUFFER
62 000216  000403                          BR      PROCS2          ;GO FOR NEXT PROCESS
63 000220  160300                  3$:     SUB     R3,R0           ;SUBTRACT THE THRESHOLD VALUE
64 000222  010067  177002                   MOV     R0,ADCSR+2      ;MOVE THE DATA TO OUTPUT BUFFER
65 000226  105737  177000          PROCS2: TSTB    @#ADCSR         ;IS DATA READY
66 000232  100375                          BPL     PROCS2          ;IF NOT WAIT FOR IT
67 000234  013700  177002                   MOV     @#ADCSR+2,R0    ;MOVE THE DATA TO REGISTER R0
68 000240  112767  000000  177001          MOVB    #0,ADCSR+1      ;START CONVERSION AT CHANNEL0
69 000246  042700  070000                   BIC     #70000,R0       ;MASK THE BITS FROM 12 TO 15
70 000252  020027  003777                   CMP     R0,#3777        ;IS IT POSITIVE
71 000256  103412                          BLO     1$              ;IF YES GO FOR IT
72 000260  020004                          CMP     R0,R4           ;IS IT WITHIN THE THRESHOLD LEVEL
73 000262  103404                          BLO     2$              ;IF NO GO FOR OUTPUT
74 000264  005000                          CLR     R0              ;BRING DOWN THE OUTPUT VALUE TO ZERO
75 000266  010067  177002                   MOV     R0,ADCSR+2      ;MOVE THE DATA TO OUTPUT BUFFER
76 000272  000655                          BR      PROCS           ;GO FOR NEXT PROCESS
77 000274  160400                  2$:     SUB     R4,R0           ;SUBTRACT THE THRESHOLD VALUE
78 000276  010067  177002                   MOV     R0,ADCSR+2      ;MOVE THE DATA TO OUTPUT BUFFER
```

```
     79 000302  000651                          BR      PROCS           ;REPEAT THE ENTIRE PROCESS AGAIN
     80 000304  020003                  1$:     CMP     R0,R3           ;IS IT WITH IN THHE THRESHOLD LEVEL
     81 000306  101004                          BHI     3$              ;IF NOT GO FOR OUTPUT
     82 000310  005000                          CLR     R0              ;CLEAR THE REGISTER R0
     83 000312  010067  177002                  MOV     R0,ADCSR+2      ;MOVE THE DATA TO OUTPUT BUFFER
     84 000316  000643                          BR      PROCS           ;GO FORNEXT PROCESS
     85 000320  160300                  3$:     SUB     R3,R0           ;SUBTRACT THE THRESHOLD VALUE
     86 000322  010067  177002                  MOV     R0,ADCSR+2      ;MOVE THE DATA TO OUTPUT BUFFER
     87 000326  000637                          BR      PROCS           ;REPEAT THE ENTIRE PROCESS AGAIN
     88 000330  000000                  PLOW:   0
     89 000332  000000                  PHIGH:  0
     90 000334  000000                  NLOW:   0
     91 000336  000000                  NHIGH:  0
     92         000600'                          .END    NEX             ;STARTING LOCATION OF PROGRAM
SYMBOL TABLE

ADCSR = 177000          NHIGH   000336R         PHIGH   000332R         PROCS   000026R         PROCS2  000226R
NEX     000600R         NLOW    000334R         PLOW    000330R         PROCS1  000126R

. ABS.  000000      000
        000340      001
ERRORS DETECTED:  0

VIRTUAL MEMORY USED:  8192 WORDS  ( 32 PAGES)
DYNAMIC MEMORY AVAILABLE FOR  48 PAGES
CCLIP,CCLIP=CCLIP
```

```
                                                          ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
                                                          ;                                             ;
                                                          ;                  **PCLIP**                  ;
                                                          ;                                             ;
                                                          ;THIS PROGRAM WAS WRITTEN TO PERFROM THE      ;
                                                          ;PEAK CLIPPING OPERATION. ONCE THE THRESHOD   ;
                                                          ;VALUES ARE OBTAINED FROM GET AND INPUT, IT   ;
                                                          ;STARTS WORKING. IT CLIPS THE INPUT SIGNAL    ;
                                                          ;IF IT IS BEYOND THE THRESHOLD VALUE. THIS    ;
                                                          ;IS TWO INDEPENDENTLY CONTROLLED CHANNELS     ;
                                                          ;SYSTEM.                                      ;
                                                          ;                                             ;
                                                          ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;


                                              .MCALL  .REGDEF,.EXIT
                                              .MCALL  ..V2..
        177000                        ADCSR=177000                        ;LOCATION OF AD CONTROL STATUS REGISTER
000000  112767  000000  177001        NEX:    MOVB    #0,ADCSR+1          ;START CONVERSION AT CHANNEL 0
000006  016701  000302                        MOV     PLOW,R1             ;MOVE THREESHOLD FOR CH 0 TO R1
000012  016703  000300                        MOV     PHIGH,R3            ;MOVE +VE THRESHOLD FOR CH6 TO R3
000016  016702  000276                        MOV     NLOW,R2             ;MOVE -VE THRESHOLD FOR CH0 TO R2
000022  016704  000274                        MOV     NHIGH,R4            ;MOVE -VE THRESHOLD FOR CH6 TO R4
000026  105737  177000                PROCS:  TSTB    @#ADCSR             ;IS DATA READY
000032  100375                                BPL     PROCS               ;NO WAIT FOR IT
000034  013700  177002                        MOV     @#ADCSR+2,R0        ;MOVE THE DATA TO REGISTER R0
000040  112767  000006  177001                MOVB    #6,ADCSR+1          ;START CONVERSION AT CHANNEL 6
000046  042700  170000                        BIC     #170000,R0          ;MASK THE UNNESSARY BITS
000052  020027  003777                        CMP     R0,#3777            ;IS IF POSITIVE VALUE
000056  103411                                BLO     1$                  ;IF YES GO TO PROCESS POSITIVE HALF CYCLE
000060  020002                                CMP     R0,R2               ;DOES IT FALL WITHIN THE THRESHOLD BAND
000062  103403                                BLO     2$                  ;NO GO FOR OUTPUT
000064  010067  177002                        MOV     R0,ADCSR+2          ;OUTPUT THE VALUE AT CHANNEL Y
000070  000414                                BR      PROCS1              ;GO FOR NEXT PROCESS
000072  010200                        2$:     MOV     R2,R0               ;MOVE THE THRESHOLD VALUE
000074  010067  177002                        MOV     R0,ADCSR+2          ;MOVE THE DATA TO OUTPUT BUFFER
000100  000410                                BR      PROCS1              ;GO FOR NEXT SIGNAL PROCESSING
000102  020001                        1$:     CMP     R0,R1               ;IS IT ABOVE THRESHOLD VALUE
000104  101003                                BHI     3$                  ;YES GO TO OUTPUT
```

```
40 000106  010067  177002                  MOV     R0,ADCSR+2      ;OUTPUT THE DATA AT CHANNEL Y
41 000112  000403                          BR      PROCS1          ;GO FOR NEXT PROCESS
42 000114  010100                  3$:     MOV     R1,R0           ;MOVE THE THRESHOLD VALUE
43 000116  010067  177002                  MOV     R0,ADCSR+2      ;MOVE THE DATA TO OUTPUT BUFFER
44 000122  105737  177000          PROCS1: TSTB    @#ADCSR         ;IS DATA READY
45 000126  100375                          BPL     PROCS1          ;IF NOT WAIT FOR IT
46 000130  013700  177002                  MOV     @#ADCSR+2,R0    ;STORE THE DATA IN REGISTER R0
47 000134  112767  000006  177001          MOVB    #6,ADCSR+1      ;START CONVERSION AT CHANNEL 6
48 000142  042700  070000                  BIC     #70000,R0       ;MASK THE FOUR HIGHEST BITS
49 000146  020027  003777                  CMP     R0,#3777        ;IS IT POSITVE VALUE
50 000152  103411                          BLO     1$              ;IF YES GO FOR APPROPRIATE PROCESSING
51 000154  020004                          CMP     R0,R4           ;IS IT WITHIN THE THRESHOLD VALUE
52 000156  103403                          BLO     2$              ;NO GO FOR OUTPUT
53 000160  010067  177002                  MOV     R0,ADCSR+2      ;MOVE THE DATA TO OUTPUT BUFFER
54 000164  000414                          BR      PROCS2          ;GO FOR NEXT PROCESS
55 000166  010400                  2$:     MOV     R4,R0           ;MOVE THE THRESHOLD VALUE
56 000170  010067  177002                  MOV     R0,ADCSR+2      ;MOVE THE DATA TO OUTPUT BUFFER
57 000174  000410                          BR      PROCS2          ;GO FOR NEXT PROCESS
58 000176  020003                  1$:     CMP     R0,R3           ;IS IT WITH IN THE THRESHOLD LEVEL
59 000200  101003                          BHI     3$              ;NO GO TO OUTPUT
60 000202  010067  177002                  MOV     R0,ADCSR+2      ;MOVE THE DATA TO OUTPUT BUFFER
61 000206  000403                          BR      PROCS2          ;GO FOR NEXT PROCESS
62 000210  010300                  3$:     MOV     R3,R0           ;MOVE THE THRESHOLD VALUE
63 000212  010067  177002                  MOV     R0,ADCSR+2      ;MOVE THE DATA TO OUTPUT BUFFER
64 000216  105737  177000          PROCS2: TSTB    @#ADCSR         ;IS DATA READY
65 000222  100375                          BPL     PROCS2          ;IF NOT WAIT FOR IT
66 000224  013700  177002                  MOV     @#ADCSR+2,R0    ;MOVE THE DATA TO REGISTER R0
67 000230  112767  000000  177001          MOVB    #0,ADCSR+1      ;START CONVERSION AT CHANNEL0
68 000236  042700  070000                  BIC     #70000,R0       ;MASK THE BITS FROM 12 TO 15
69 000242  020027  003777                  CMP     R0,#3777        ;IS IT POSITIVE
70 000246  103411                          BLO     1$              ;IF YES GO FOR IT
71 000250  020004                          CMP     R0,R4           ;IS IT WITHIN THE THRESHOLD LEVEL
72 000252  103403                          BLO     2$              ;IF NO GO FOR OUTPUT
73 000254  010067  177002                  MOV     R0,ADCSR+2      ;MOVE THE DATA TO OUTPUT BUFFER
74 000260  000662                          BR      PROCS           ;GO FOR NEXT PROCESS
75 000262  010400                  2$:     MOV     R4,R0           ;MOVE THE THRESHOLD VALUE
76 000264  010067  177002                  MOV     R0,ADCSR+2      ;MOVE THE DATA TO OUTPUT BUFFER
77 000270  000656                          BR      PROCS           ;REPEAT THE ENTIRE PROCESS AGAIN
78 000272  020003                  1$:     CMP     R0,R3           ;IS IT WITH IN THHE THRESHOLD LEVEL
```

```
     79 000274  101003                        BHI     3$              ;IF NOT GO FOR OUTPUT
     80 000276  010067  177002                MOV     R0,ADCSR+2      ;MOVE THE DATA TO OUTPUT BUFFER
     81 000302  000651                        BR      PROCS           ;GO FORNEXT PROCESS
     82 000304  010300                3$:     MOV     R3,R0           ;MOVE THE THRESHOLD VALUE
     83 000306  010067  177002                MOV     R0,ADCSR+2      ;MOVE THE DATA TO OUTPUT BUFFER
     84 000312  000645                        BR      PROCS           ;REPEAT THE ENTIRE PROCESS AGAIN
     85 000314  000000                PLOW:   0
     86 000316  000000                PHIGH:  0
     87 000320  000000                NLOW:   0
     88 000322  000000                NHIGH:  0
     89         000000'                       .END    NEX             ;STARTING LOCATION OF PROGRAM
SYMBOL TABLE

ADCSR = 177000          NHIGH   000322R         PHIGH   000316R         PROCS   000026R         PROCS2  000216R
NEX     000000R         NLOW    000320R         PLOW    000314R         PROCS1  000122R

. ABS.  000000      000
        000324      001
ERRORS DETECTED:  0

VIRTUAL MEMORY USED:  8192 WORDS  ( 32 PAGES)
DYNAMIC MEMORY AVAILABLE FOR  48 PAGES
PCLIP,PCLIP=PCLIP
```

APPENDIX C

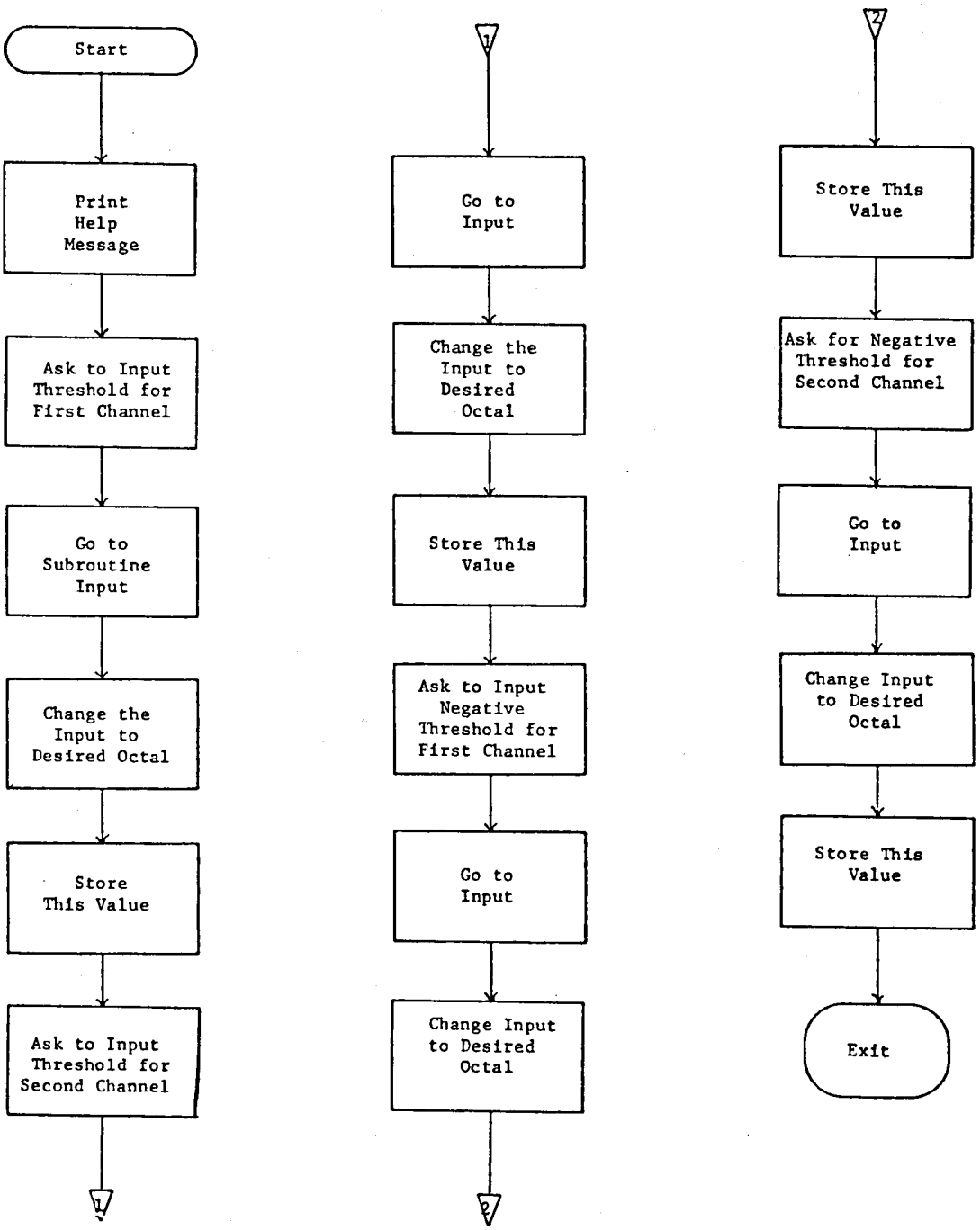


Figure 31. A flowchart of Get.

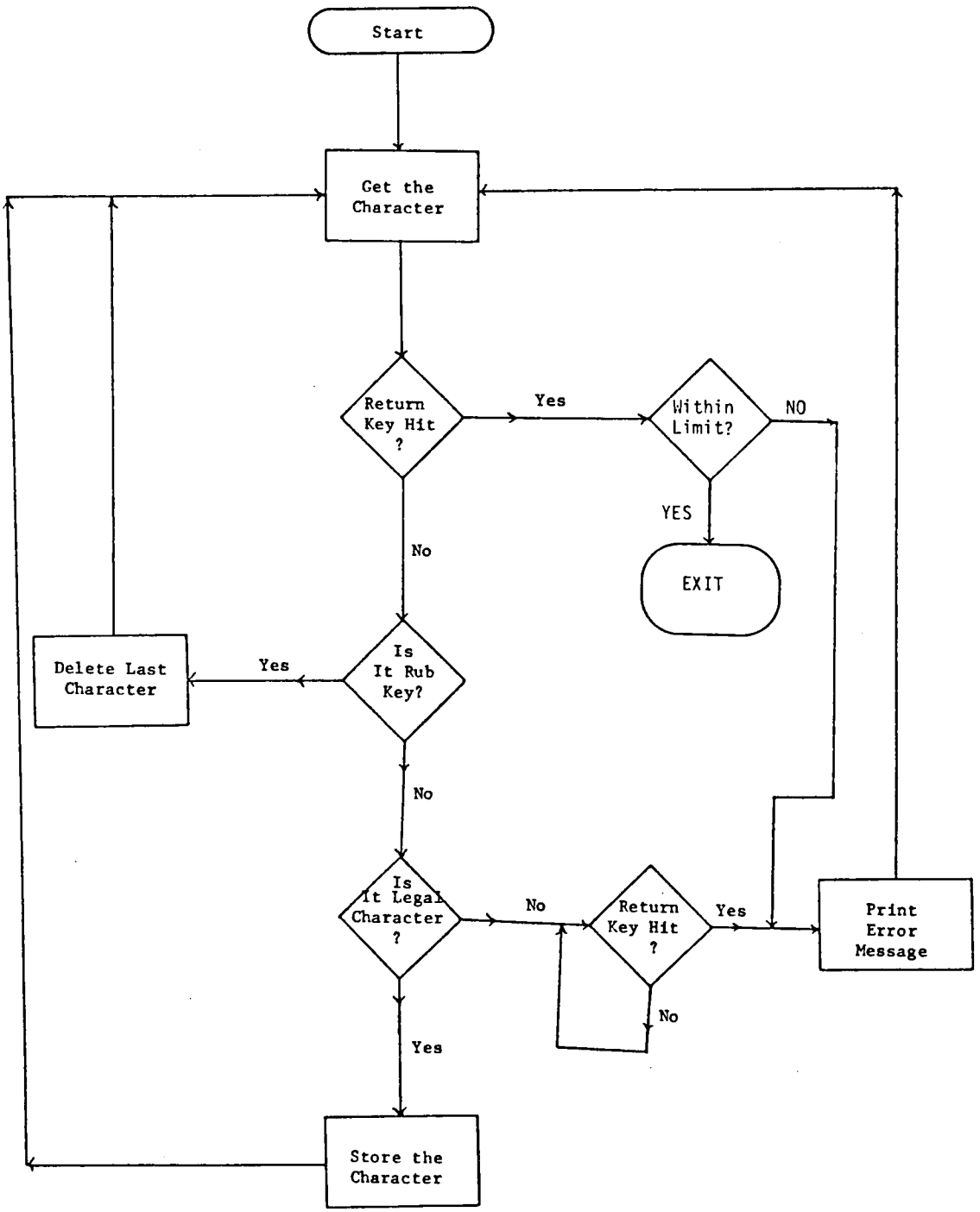


Figure 32. A flowchart of Input.

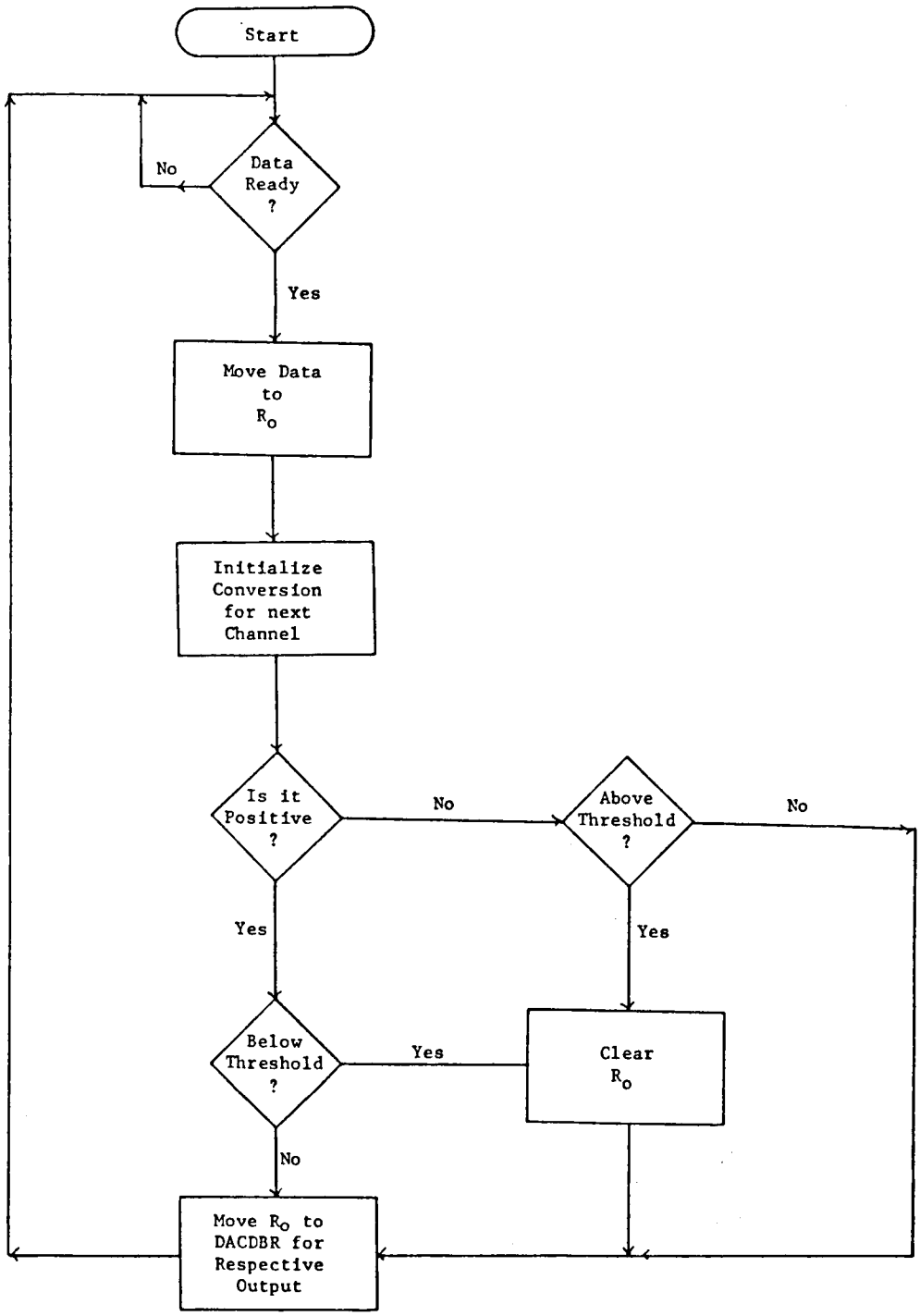


Figure 33. A flowchart of Fast Expansion.

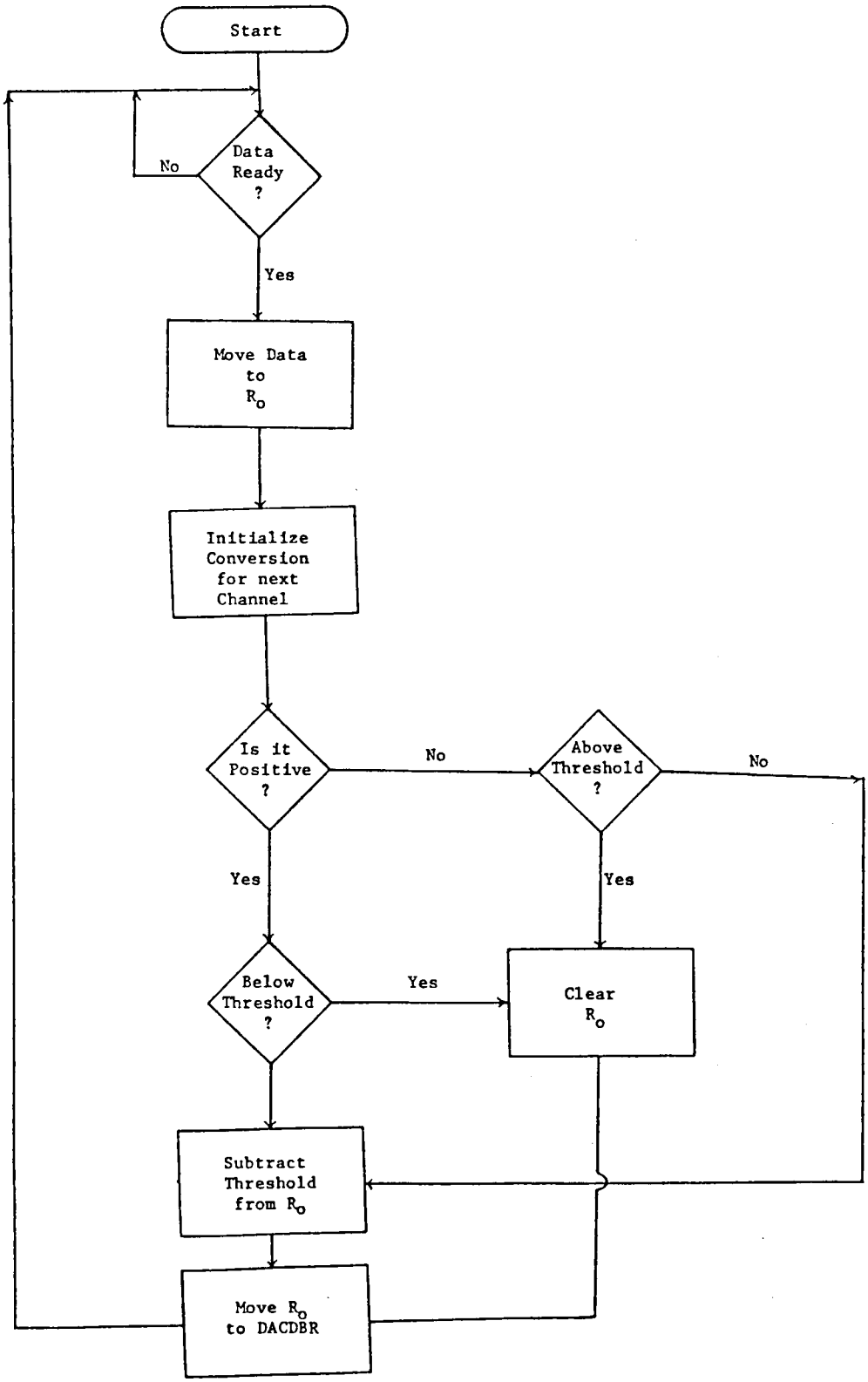


Figure 34. A flowchart of Center Clipping.

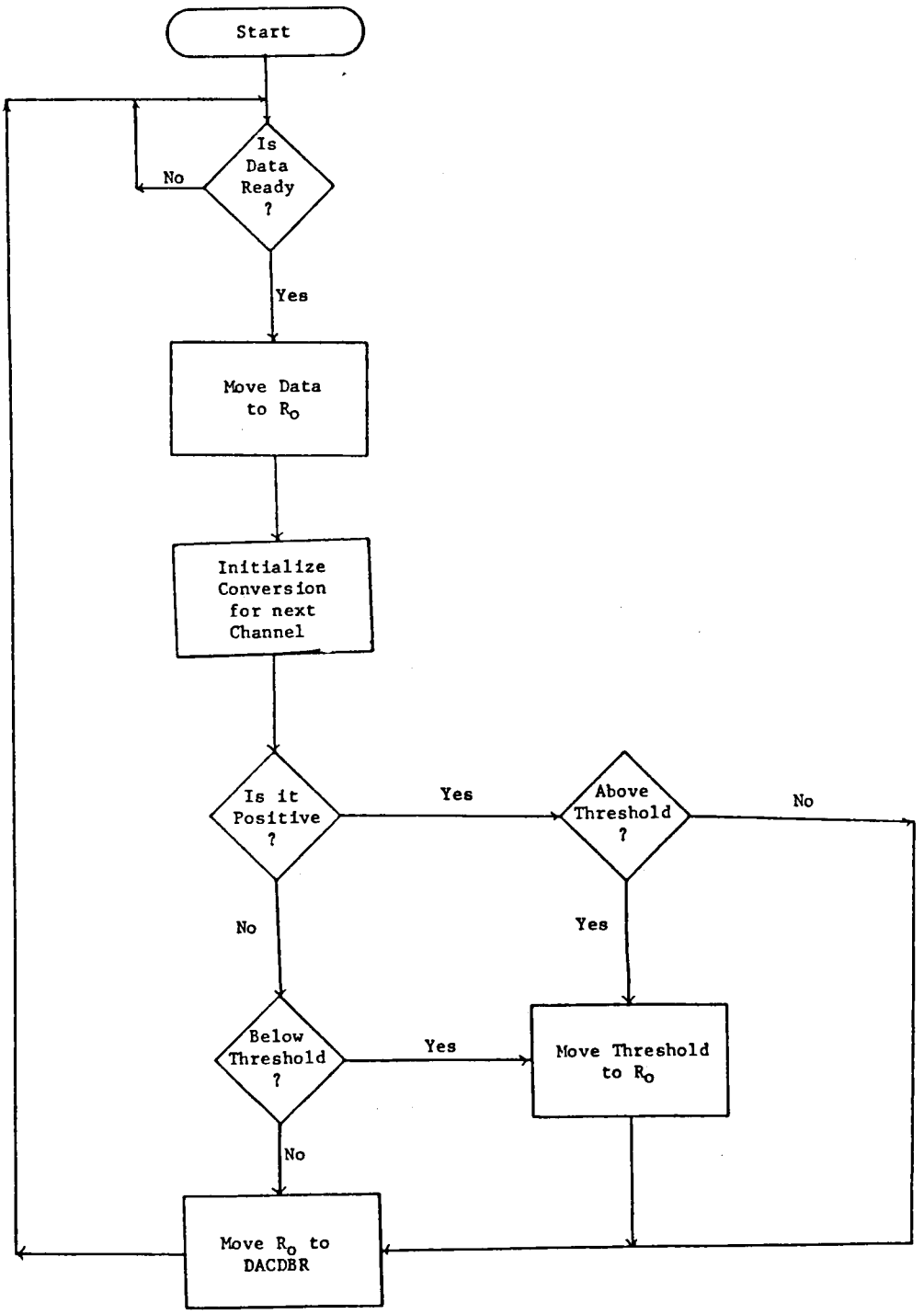


Figure 35. A flowchart of Peak Clipping.

VITA

Naeem Uddin Syed was born on August 26, 1952, in Pakistan. After graduating from Government College Hyderabad in 1970, he attended N.E.D. Government Engineering College of The Karachi University. He received the Bachelor of Engineering degree in electrical engineering in December 1975, from there. He was employed in electrical industries from 1975 to July 1979 in Pakistan. He worked for Saudi Japan Construction Co., Riyadh, in Saudi Arabia, as an electrical engineer from July 1979 to August 1980. His job duties include design, supervision and testing of different electrical systems.

He entered the Graduate School of The University of Tennessee, Knoxville, in September 1980. He has been serving as a Teaching Assistant at the Electrical Engineering Department of The University of Tennessee, Knoxville, since January 1981.