הטכניון – מכון טכנולוגי לישראל
Technion – Israel Institute of Technology

ספריות הטכניון
The Technion Libraries

בית הספר ללימודי מוסמכים ע"ש ארווין וג'ואן ג'ייקובס
Irwin and Joan Jacobs Graduate School

REAL TIME SPEECH AND DATA PROCESSING BY A COMPUTER CONTROLLED DIGITAL FILTER

Project Thesis

Submitted in Partial Fulfillment of the Requirements
for the Degree of Master of Science
in Electrical Engineering

by

Roland D. Wilk

Submitted to the Senate of the Technion
Israel Institute of Technology

Elul 5736 Haifa September 1976

The project was carried out under the supervision of
Dr David Malah in the Signal Processing Laboratory of the
Faculty of Electrical Engineering

I wish to express my gratitude to Dr Malah for his guidance during the execution of the project and to thank Mr Eli Pollak, the Signal Processing Laboratory engineer, for his assistance.

I am grateful to the Lady Davis Foundation for having awarded me a Fellowship for the duration of my studies.

CONTENTS

LIST OF FIGURES

ABSTRACT

The project deals with the design and construction of a general purpose computer controlled transversal digital filter with ten coefficients and a sampling rate of up to 20kHz. The filter is controlled by means of a NOVA 2 mini-computer. Two applications viz. a) a keyboard controlled FIR filter and b) a real time speech synthesizer are implemented.

For the first application software was developed for the computation of filter coefficients to approximate a given frequency specification in the least squares sense. The computed coefficients can then be transfered to the filter.

Since the filter is computer controlled, its characteristics can be made to be time varying by updating the coefficients. This feature is fully exploited in the speech synthesizer.

The speech synthesizer is used to synthesize a speech wave in real time from a set of parameters characterizing the human vocal system i.e. larynx, vocal tract and oral and nasal apertures. During normal speech the vocal system characteristics vary with time. In the synthesizer this is achieved by updating the set of parameters at regular intervals of 15ms.

The functional model of the vocal system consists of an all pole digital filter realized by means of a transversal filter in a feedback loop. The all pole filter is excited either by a sequence of pulses (for voiced speech) or by band limited white noise (for unvoiced speech). This model is known as the linear prediction model of the vocal system

Software was developed for the extraction of the required parameters from a sampled speech wave. An exciter to generate the required excitation for the all pole filter was constructed as an extension of the transversal filter so that the exciter parameters can be updated simultaneously with the filter coefficients. For the analysis, speech is sampled at 10kHz and parameters computed over a 450 sample segment, succesive segments overlapping by 300 samples i.e. a set of parameters is computed every 150 samples. During synthesis, therefore, the parameters are updated every 15ms. Each set of parameters is encoded into 96 bits, thus giving a bit rate of 6400 bits/sec. However, it is probably possible to reduce the sampling and update rates significantly without causing much deterioration in the quality of the synthesized speech, thereby achieving a great saving in bits per second of speech.

* * * * * * * *

GLOSSARY OF SYMBOLS

a(k)	Predictor coefficients
d(n)	Desired signal
e(n)	Error signal
h(k)	Impulse response
p(k)	Complex poles of linear predictor
r(k)	Autocorrelation coefficients
w	Angular frequency (discrete time systems)
x(n)	Input signal
y(n)	Output signal
j,k,n	Integer indices
l,m,q	Integer constants
s,z	Transform domain variables
u,v	Real constants
ws	Angular sampling frequency
A(n)	Discrete Fourier transform
E	Quantization step size
H(z)	Z-transform of h(k)
P	Order of linear predictor
Q	Parameter of generalized Hamming window
T	Sampling period
W(k)	Window
X(z)	Z-transform of x(n)
Y(z)	Z-transform of y(n)
M,N	Integer constants
$\emptyset$	Correlation coefficient
Ω	Angular frequency (continuous time systems)

CHAPTER 1

INTRODUCTION

Digital filters are readily realized by standard computer software either directly from the difference equations characterizing the filter or by means of the Fast Fourier Transform. The reason for designing special purpose hardware is to relieve the computer of a task or tasks which can be executed more efficiently and if necessary in real time, which the general purpose computer is often unable to perform.

A computer controlled filter is very versatile. It can be used with constant coefficients as a standard digital filter, but since the loading of coefficients can be performed by the computer, sets of coefficients corresponding to standard type frequency responses can be stored by the computer and used to program the filter. The advantage of computer loading compared to the laborious task of manual loading becomes greater as the number and length (in bits) of the coefficients increases.

In applications where high speed real time signal processing by a time varying filter is required, the computer controlled filter becomes indispensible. Most general purpose computers are incapable of processing signals sampled at 10kHz or more. The computer controlled filter is then used as a special purpose extension of the computer. Applications include speech synthesis and adaptive equalization.

In the former application the filter, in a suitable feedback configuration, is used to model the time varying

transfer function of the vocal tract.

In the latter application the filter is used to compensate for the distortion caused by the communication channel. This compensation is known as equalization. Since the channel characteristics vary slowly with time, the computer is used to sample the equalizer output, to compute optimal coefficients and to transmit these to the filter.

The project deals with the design and construction of a computer controlled transversal digital filter with 10 coefficients and a sampling rate of up to 20kHz. Software is developed for the computation of the filter coefficients given an ideal frequency response which is approximated in the least squares sense.

A further application of the filter, that of a speech synthesizer, is developed in full. For this application additional equipment had to be built namely an exciter and the feedback loop in which the filter is used. Software for the analysis of sampled speech is developed to extract the required parameters for the synthesizer.

Chapter 2 reviews standard design procedures for and methods of realization of digital filters in general. Some of these are applicable to the constructed filter. Chapter 3 deals with the actual design and construction of a 10 coefficient computer controlled transversal digital filter while Chapter 4 presents the principles of speech synthesis by linear prediction as well as a description of the actual hardware of

the synthesizer and its supporting software. All computer programs are listed in Appendix A. A sample run of the speech analysis program can be found in Appendix B. Data sheets of special devices or ready made modules are given in Appendix C while complete schematics of the built units are provided in Appendix D.

* * * * * * * *

CHAPTER 2

DIGITAL FILTERS

2.1 Definition:

Rabiner et al.[11] aptly define a digital filter as "a computational process or algorithm by which a digital signal or sequence of numbers (acting as input) is transformed into a second sequence of numbers termed the output signal. The numbers are limited to a finite precision. The algorithm may be implemented in software as a computer subroutine for a general purpose machine, or in hardware as a special purpose computer. The term digital filter is then applied to the specific routine in execution or to the hardware".

Let x(n) be the input sequence to a digital filter and y(n) be the output sequence. In general y(n) is a function both of previous outputs and present and previous inputs i.e.

$$y(n)=F(y(n-1),y(n-2),\ldots,y(n-M),x(n),x(n-1),\ldots,x(n-N)) \quad (2.1)$$

2.2 Terminology:

2.2.1 Impulse Response:

The impulse response of a digital filter is the response y(n) to a unit sample

$$x(n)=1 \quad n=0$$
$$=0 \quad \text{otherwise} \quad (2.2)$$

If the filter is causal, then $y(n)=0$ for $n<0$.

2.2.2 FIR and IIR Filters:

Let h(n) be the impulse response of a digital filter. The

filter is of the FIR variety if both finite M and N exist such that

$$h(n)=0 \text{ for all } n<N$$
$$h(n)=0 \text{ for all } n>M,\ M>N \qquad (2.3)$$

and is of the IIR variety otherwise.

2.2.3 Recursive and Nonrecursive Realizations:

Let x(n) be the input sequence to a digital filter and y(n) be the output sequence. The filter is said to be realized nonrecursively if

$$y(n)=F(x(n),x(n-1),x(n-2),\ldots,x(n-M)) \qquad (2.4)$$

i.e. y(n) is a function of past and present inputs only. If y(n) is explicitly dependent on previous outputs, then the filter is said to be realized recursively.

FIR and IIR filters can be realized both recursively and nonrecursively. In general, however, FIR filters are realized more efficiently nonrecursively while recursive realizations are used for IIR filters.

2.2.4 Z-Transform:

The z-transform of a sequence x(n) n=0,1,2,... is defined as

$$X(z)=\sum_{n=0}^{\infty} x(n)z^{-n} \qquad (2.5)$$

2.2.5 Transfer Function:

The transfer function of a discrete time system is defined as

$$H(z)=Y(z)/X(z) \qquad (2.6)$$

where X(z) and Y(z) are the z-transforms of the input and output sequences respectively. H(z) is also the z-transform of the system impulse response.

2.2.6 Frequency Response:

The frequency response of a system is its response to the complex sinusoid exp(jwn) which can be shown to be obtained from the transfer function H(z) by

$$H(\exp(jw)) = H(z)\Big|_{z=e^{jw}} \qquad (2.7)$$

H(exp(jw)) is a periodic function of w with period 2π.

2.2.7 Stability:

A discrete time system is stable if all the poles of its transfer function H(z) lie within the unit circle of the complex z-plane.

2.2.8 Discrete Fourier Transform (DFT):

The discrete Fourier transform A(n) of a sequence x(k) k=0,1,2,...,N-1 defined as

$$A(n) = \sum_{k=0}^{N-1} x(k)\exp(-j2\pi k/N) \qquad (2.8)$$

2.3 Design of FIR Filters:

One of the advantages of FIR filters is that they can be designed to have an exactly linear phase characteristic. This is a desirable feature and most design methods usually incorporate this condition.

Since design procedures are available in the literature,

only a summary is presented.

2.3.1 Fourier Series and Windowing:

The frequency response H(exp(jw)) of the filter is a periodic function and can therefore be expanded in a Fourier series. The coefficients of this series are the filter impulse response coefficients. The Fourier series generally has an infinite number of nonzero terms. Since an FIR filter which approximates the original frequency response is required, the Fourier series must be truncated. Direct truncation is equivalent to multiplying the impulse response by a rectangular window. Multiplication in the time domain corresponds to convolution in the frequency domain. The Fourier transform of the rectangular window is a sin(x)/x function, which has large side lobes, thus grossly distorting the frequency response. The search for a finite window with most of its energy in its main lobe has led to many designs, namely the Hamming, Blackman, Kaiser and Helms designs.[10]

The Hamming and Blackman designs are easy to evaluate and provide a fixed peak sidelobe ripple. The Kaiser design provides a window where a tradeoff can be obtained between peak sidelobe ripple and the width of the main lobe by the adjustment of a parameter. The disadvantage is that Bessel functions must be computed. The Helms design uses the Dolph-Chebyshev window which is optimal in the sense that the main lobe width is as small as possible for a given peak ripple. The main disadvantage is that inverse hyberbolic

cosines must be evaluated to determine the filter coefficients.

The discrete Fourier transform can be used to compute a good approximation to the Fourier series by sampling the frequency response at a much larger number of points than the number of coefficients required. This is the method used in the computer program described in 2.7.

2.3.2 Frequency Sampling:

This method is used for realizing the filter by a comb filter in cascade with a bank of complex resonators (see fig. 2.1). The frequency response is sampled at N equispaced frequencies where N is the number of coefficients in the required filter impulse response. Let these samples be H(k) k=0,1,...N-1.

The filter transfer function H(z) can be expanded as

$$H(z)=\sum_{n=0}^{N-1}h(n)z^{-n} = \frac{1-z^{-N}}{N}\sum_{k=0}^{N-1}\frac{H(k)}{1-z^{-1}e^{j\frac{2\pi k}{N}}} \qquad (2.9)$$

The expresssion in (2.9) shows that the filter frequency response is linearly related to the the frequency samples H(k). Thus linear optimization techniques can be used to select optimally values of several, or all, the frequency samples to give the best approximation to the desired filter.

2.3.3 Polynomial Filters:

This is a time domain design approach whereby a polynomial of degree q is fitted through the r last values of the sequence. If q=r+1 the fit is exact. If q<r+1 an error

criterion, usually least squares, must be used while if q>r+1, the number of possible fits is infinite.

Applications of polynomial filters include interpolation, extrapolation (prediction), differentiation and integration.

2.3.4 Inverse Filter:

The aim of inverse filter design is to find the filter coefficients h(n) such that for a prescribed input x(n), the output y(n) is an impulse or some other desired function.

In the Wiener filter design,[12] the output of a filter is required to approximate a desired signal d(n) in the least squares sense when a prescribed signal is input to the filter (see fig. 2.2).

The requirement of minimum error energy gives rise to the system of correlation equations

$$\sum h(n)\phi_{xx}(j-n)=\phi_{dx}(j) \quad j=0,1,2,\ldots,N \qquad (2.10)$$

where $\phi_{ab}(k)$ is the expected value of a(n+k)b(n).

2.4 Design of IIR Filters:

Since design techniques for analog filters are well established, it is logical that this material should be used in the design of digital filters. This is done by transforming the s-domain analog transfer function to a z-domain sampled data transfer function. The different methods of doing so correspond to which characteristics are required to be preserved.

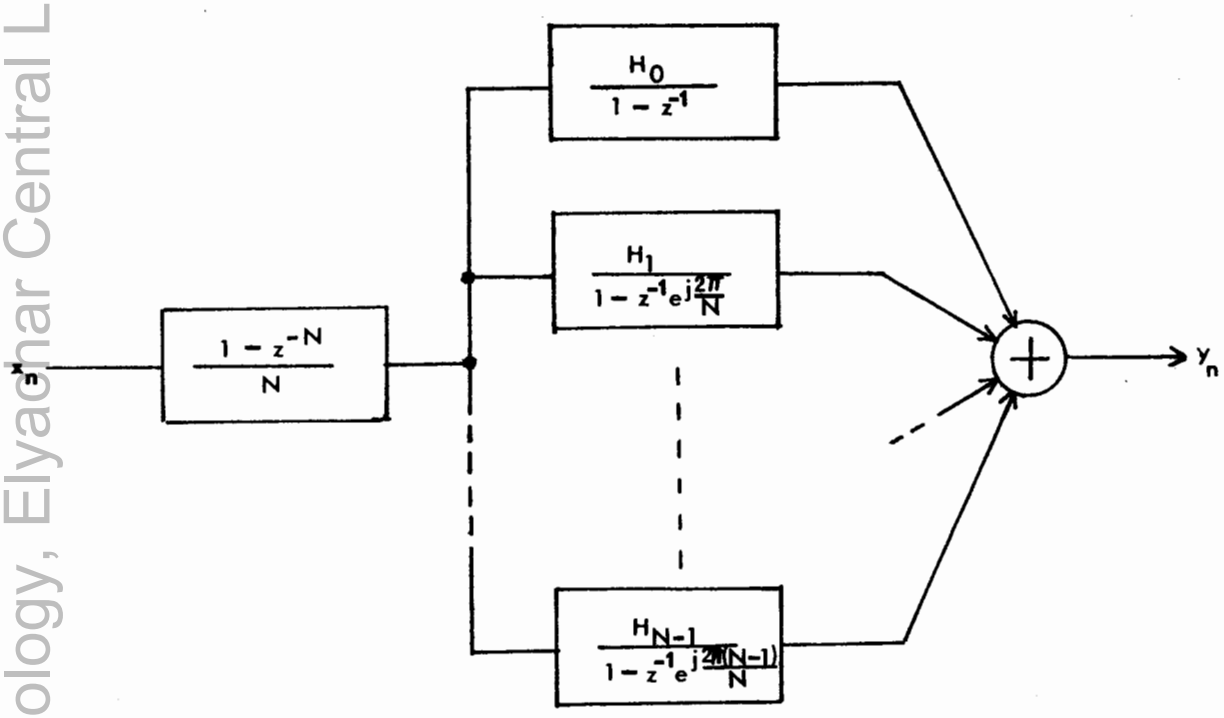


Fig. 2.1
Realization of an FIR Filter Using a Comb Filter and a Bank of Complex Resonators

ציור 2.1
ממוש מסנן FIR באמצעות מסנן מסרק ומערך של רזונטורים קומפלכסיים

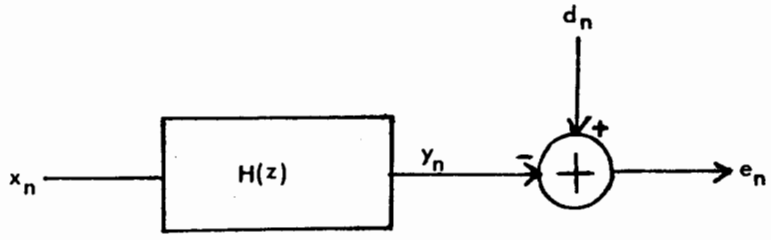


Fig. 2.2
Wiener Filter

ציור 2.2
מסנן Wiener

2.4.1 Standard Z-Transform:

This transformation preserves the impulse response. The analog transfer function is first transformed into the time domain (inverse Laplace) after which it is sampled and transformed into the z-domain.

2.4.2 Bilinear transformation:

This transformation maps the analog frequency response on $\Omega\epsilon(0,\infty)$ to $w\epsilon(0,ws/2)$ where $2\pi/ws$ is the sampling period T.

The mapping function is given by

$$\Omega=(ws/\pi)\tan(w\pi/ws) \qquad (2.11)$$

The transformation is performed simply by replacing s with $2(z-1)/T(z+1)$, but in order to retain the correct critical frequency, prewarping of the analog response has to be performed.

2.4.3 Matched Z-Transform:[4]

This transformation matches the poles and zeros of the z-domain transfer function to those of the s-domain function.

The mapping function is given by

$$s \dashrightarrow \exp(sT)=z \qquad (2.12)$$

The transformation is performed by replacing terms of the form

$s-u$ with $1-z^{-1}\exp(uT)$ and

$(s-u)^2+v^2$ with $1-2z^{-1}\exp(uT)\cos(vT)+z^{-2}\exp(2uT)$

Note that while poles are treated in the same way as in the standard z-transform, zeros are not.

2.4.4 Linear Programming Design:[9]

All of the previous methods have taken standard analog designs (high pass, low pass, band pass and band stop) and transformed them to discrete time designs. When a nonstandard frequency response is required, then some algorithmic, as opposed to closed form, design procedure is generally used.

Rabiner et al. describe a frequency domain procedure using linear programming techniques to choose filter coefficients to approximate an arbitrary magnitude characteristic.

2.4.5 Time Domain Designs:

One of the cases where time domain design of IIR filters is required is in linear prediction.[6] A linear predictor of order P is a filter of the form

$$y(n)=a(1)y(n-1)+a(2)y(n-2)+\ldots+a(P)y(n-P) \qquad (2.13)$$

This filter has the transfer function

$$H(z)=1/(1+a(1)z^{-1}+a(2)z^{-2}+\ldots+a(P)z^{-P}) \qquad (2.14)$$

and can be seen to have only poles in the complex z-plane excluding the possibility of multiple zeros at the origin.

Given a sequence x(n) we wish to find the coefficients a(k) such that the prediction residual energy

$$\sum_{n=M}^{N}\left(y(n)-\sum_{k=1}^{P}y(n-k)a(k)\right)^2 \qquad (2.15)$$

is minimal. This condition gives rise to a system of autocorrelation equations which can be solved for numerically. This topic is discussed more fully in Chapter 4.

2.5 Realization of FIR Filters:

2.5.1 Nonrecursive:

a) Direct Convolution

This can be performed by software or by a digital transversal filter, which is the sampled data equivalent of a tapped delay line (see fig. 2.3).

b) Discrete Fourier Transform

The DFT is implemented using the Fast Fourier Transform (FFT) algorithm. In order to use the FFT to perform the convolution of a finite duration signal (e.g. a finite duration impulse response) with an infinite duration or finite duration signal where the number of samples is greater than can be handled by the FFT program, it is necessary to decompose the signal into sections and compute the discrete convolution as many smaller convolutions. In order to preserve the mutual effect of each segment on its neighbour, some form of overlapping is required.

One of two methods is generally used. The "overlap save" method uses segments of N samples with an overlap of M-1 samples where N is the the number of points of the DFT and M is the duration of the impulse response. The "overlap add" method uses non-overlapping segments of N-M samples. A full discussion can be found in Brigham.[2]

2.5.2 Recursive:

For designs where the majority of of frequency samples are

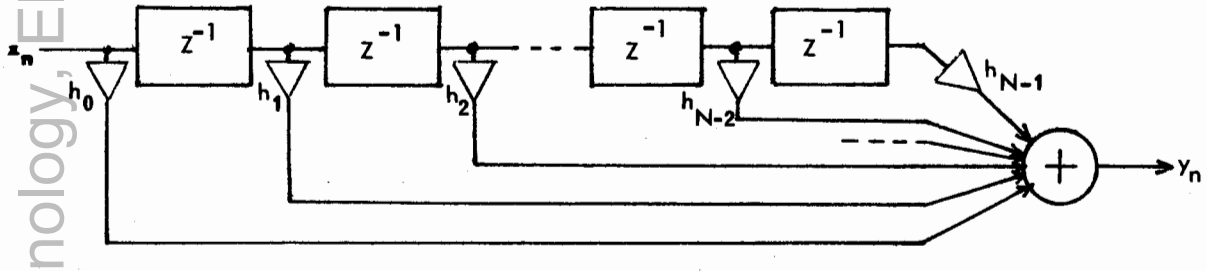


Fig. 2.3
Transversal Filter

ציור 2.3
מסנן טרנסוורסלי

either 0 or 1, a recursive realization using the frequency sampling technique of 2.3.2 can prove to be much more efficient. This realization, which uses a comb filter in cascade with a bank of complex resonators, has been shown in fig. 2.1.

2.6 Realization of IIR Filters:

IIR filters can be realized directly as

$$y(n)=\sum_{k=1}^{M}a(k)y(n-k)+\sum_{k=0}^{N}b(k)x(n-k) \qquad (2.16)$$

When finite length arithmetic is used, truncation and/or roundoff errors can have a noticible effect on the stability of the filter. For this reason filters are usually realized either as a cascade of first or second order sections or as a sum of first or second order sections in parallel. The cascade and parallel realizations are shown in figures 2.4 and 2.5 respectively. A direct form second order section is shown in fig. 2.6. The corresponding first order section is obtained by setting a(2) and b(2) to zero.

The linear predictor described in 2.4.5 can be realized by using a transversal filter in a closed loop as in fig. 2.7.

2.7 Computer Program:

A conversational FORTRAN program, Program A, was written for the NOVA 2 for the computation of coefficients for a transversal filter. The listing is given in Appendix A. The desired frequency response is approximated by a piecewise

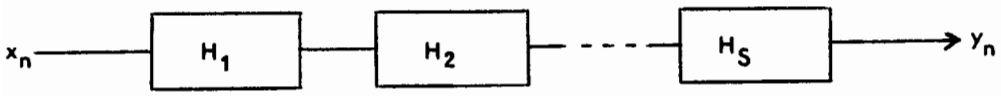


Fig. 2.4
Cascade Realization of
an IIR Filter

ציור 2.4
ממוש טורי של
מסנן IIR

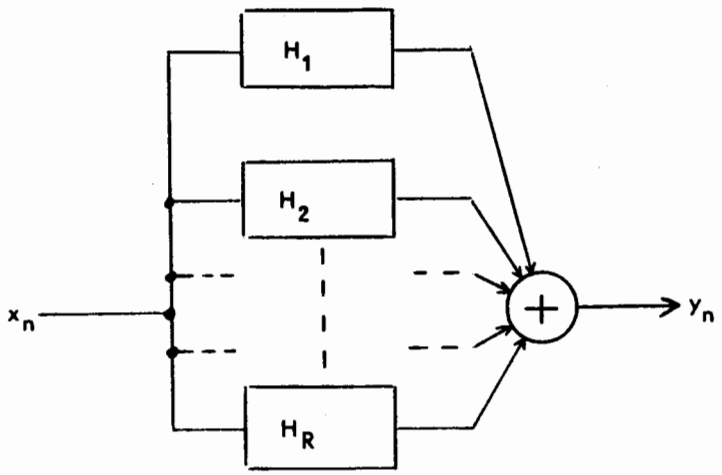
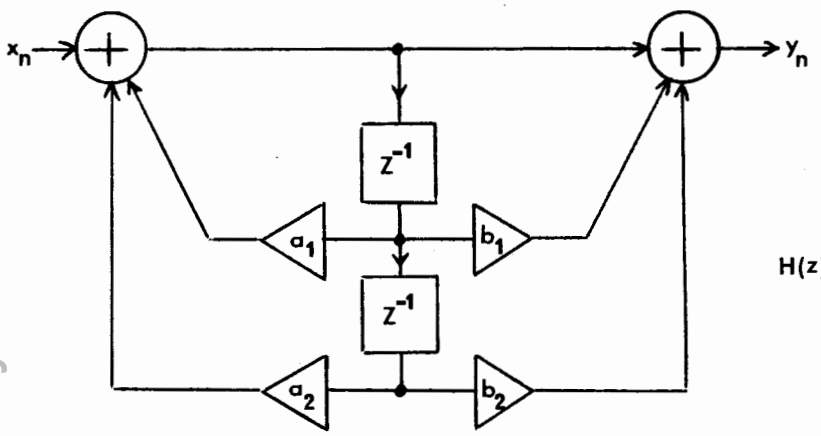


Fig. 2.5
Parallel Realization of
an IIR Filter

ציור 2.5
ממוש מקבילי של
מסנן IIR



$$H(z) = \frac{1+b_1z^{-1}+b_2z^{-2}}{1-a_1z^{-1}-a_2z^{-2}}$$

Fig. 2.6
Direct Form Second Order Section

ציור 2.6
ממוש ישיר של מסנן מסדר שני

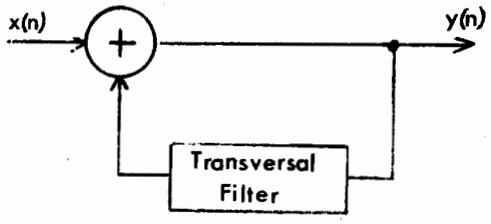


Fig. 2.7
Linear Predictor

ציור 2.7
מערכת לחזוי ליניארי

linear function and then linearly interpolated over 1024 points. The DFT is used to compute the filter impulse response which can then be truncated to any length and multiplied by the generalized Hamming window

$$W(k)=Q+(1-Q)\cos(2\pi k/N) \quad k=-N/2,\ldots,-1,0,1,\ldots,N/2 \qquad (2.17)$$

where N is the length of the truncated impulse response and Q is a parameter. Q=1 gives a rectangular window, Q=.54 a Hamming window while Q=.5 gives a Hanning window.

The graphics terminal is used to display the piecewise linear approximation, its impulse response, the truncated and windowed impulse response and the resulting frequency response of the designed filter. A flowchart of the program is given in fig. 2.8 and examples of filter designs using this program are given in figures 2.9 to 2.17.

* * * * * * * *

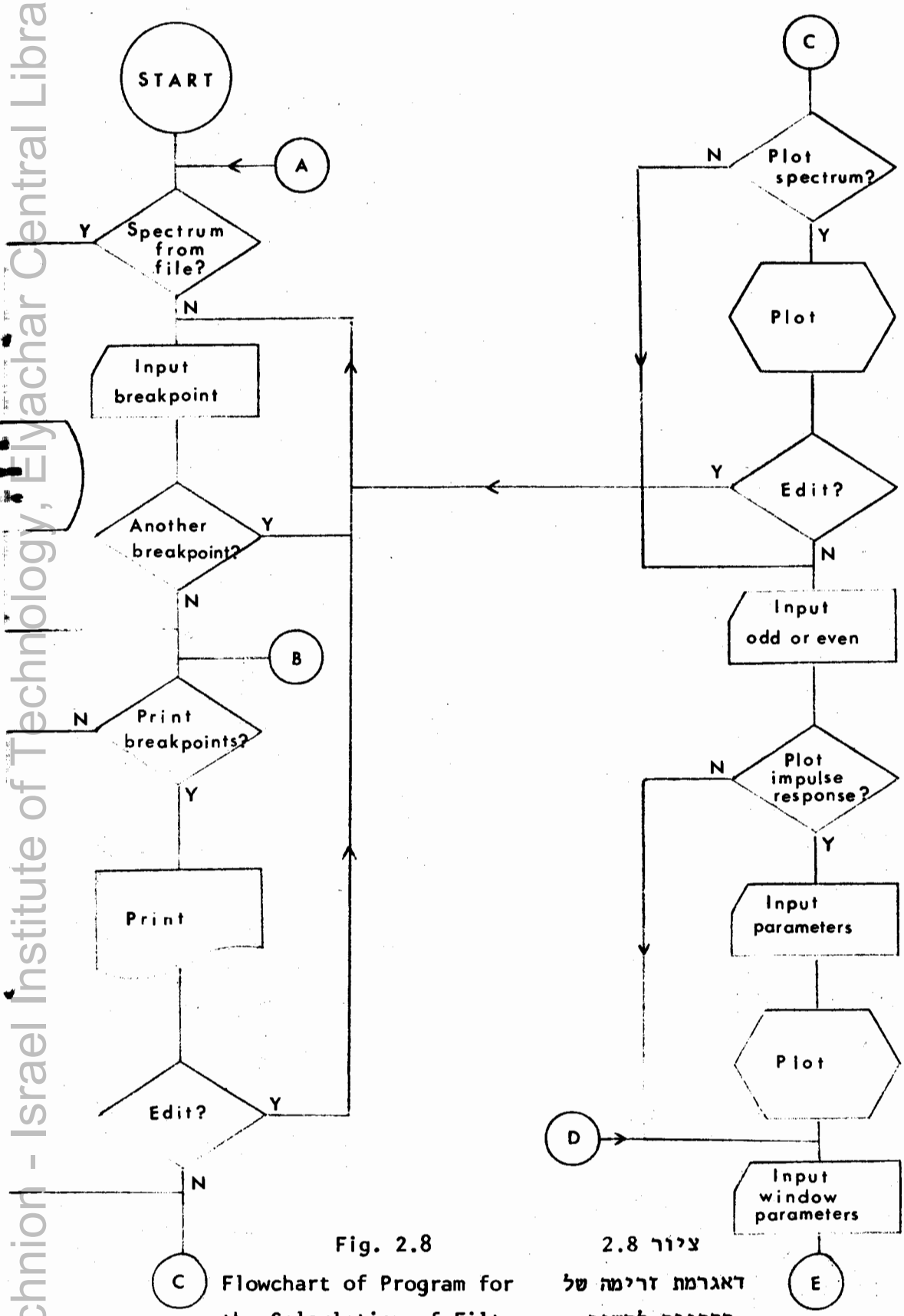


Fig. 2.8 Flowchart of Program for the Calculation of Filter Coefficients (Program A)

ציור 2.8 דיאגרמת זרימה של התכנית לחשוב מקדמי המסנן (תכנית A)

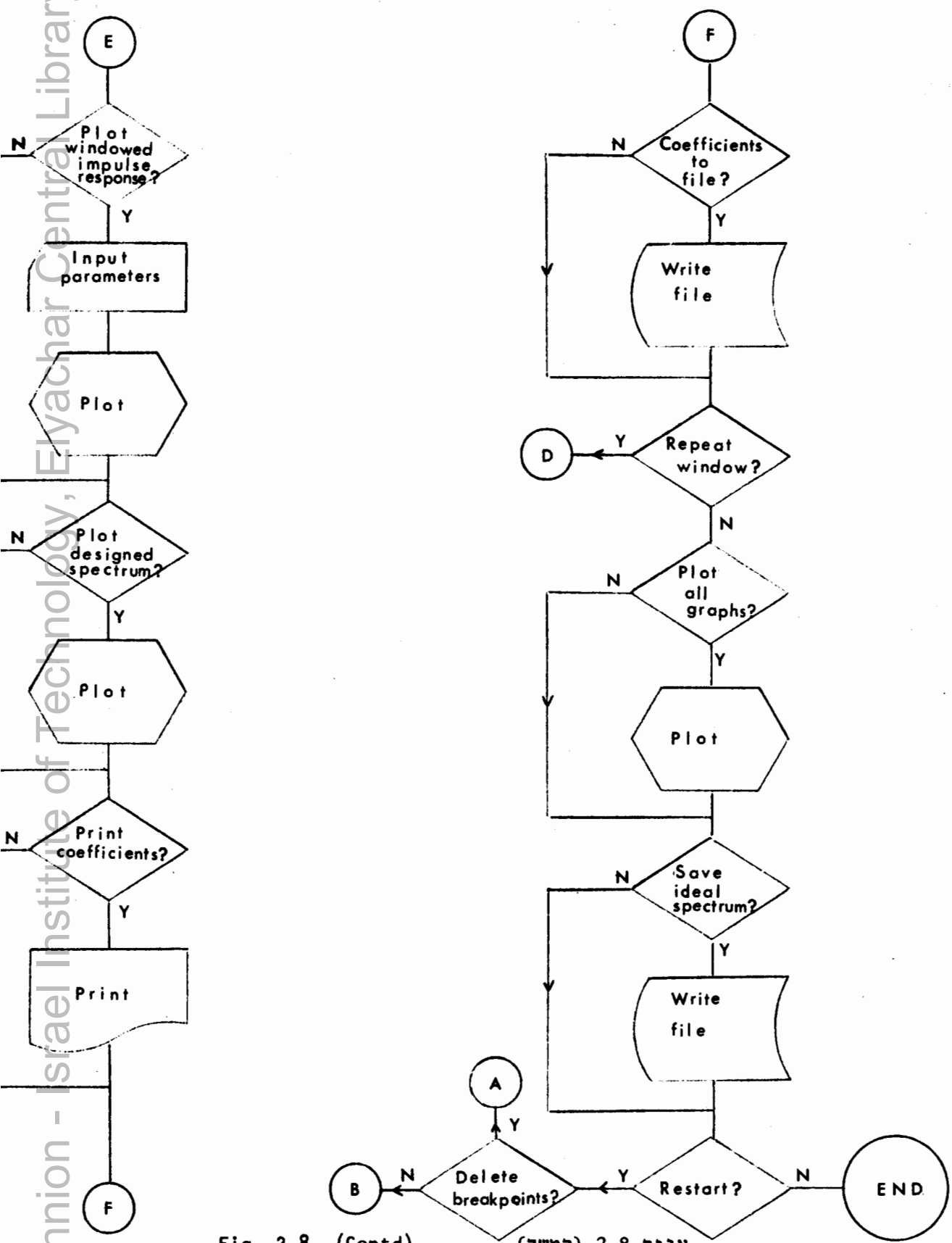


Fig. 2.8 (Contd)
Flowchart of Program for the Calculation of Filter Coefficients (Program A)

ציור 2.8 (המשך)
דיאגרמת זרימה של תכנית לחשוב מקדמי המסנן (תכנית A)

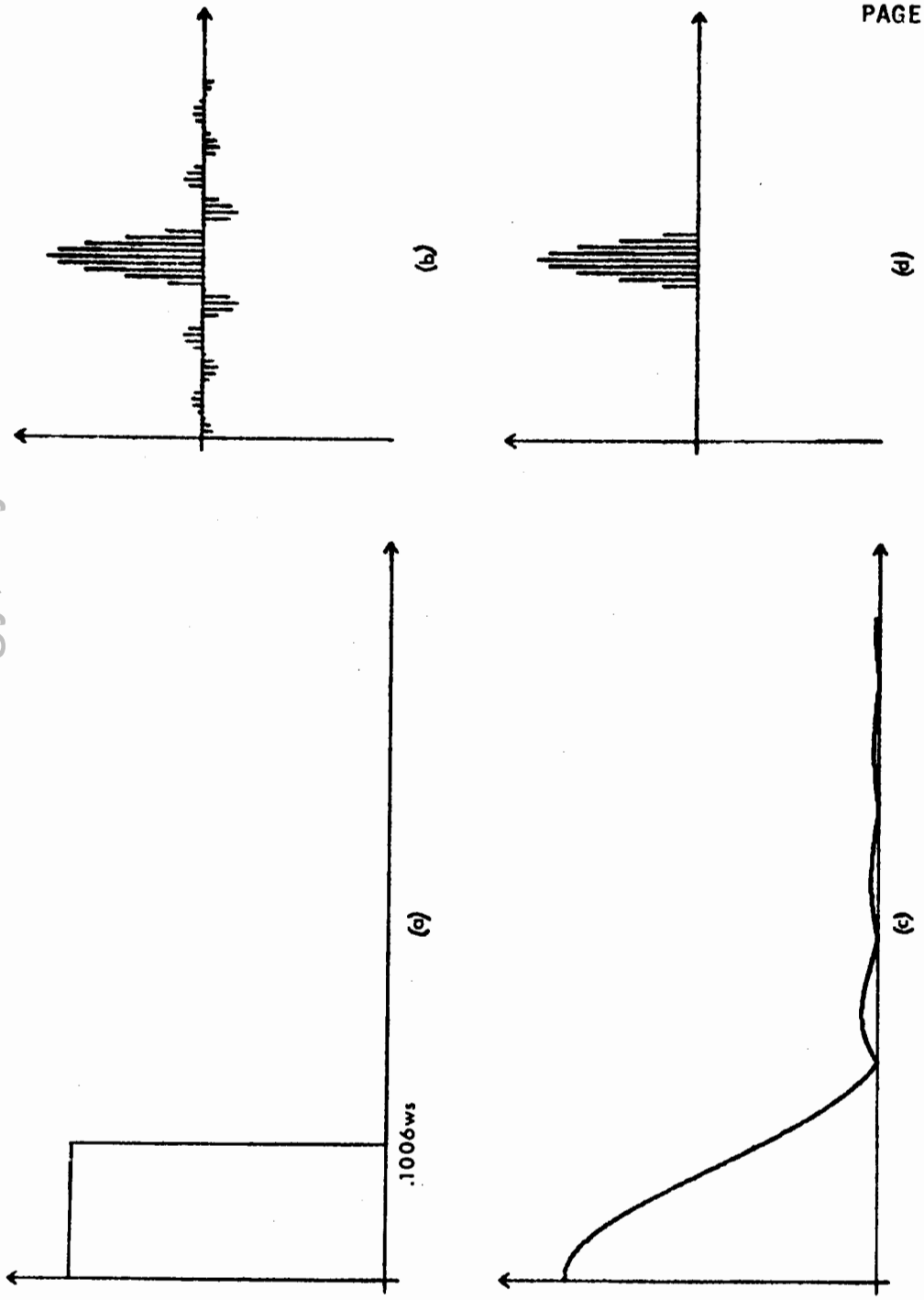


Fig. 2.9
Design Example (Low Pass)
3 Coefficients, Rectangular Window
a) Required Frequency Response
b) Impulse Response of (a)
c) Approximated Frequency Response
d) Impulse Response of (c)

ציור 2.9
דוגמת תכנון (מסנן מעביר נמוכים)
9 מקדמים, חלון מלבני
a) תגובת התדר הדרוש
b) תגובה להלם של (a)
c) תגובת התדר המקורבת
d) תגובה להלם של (c)

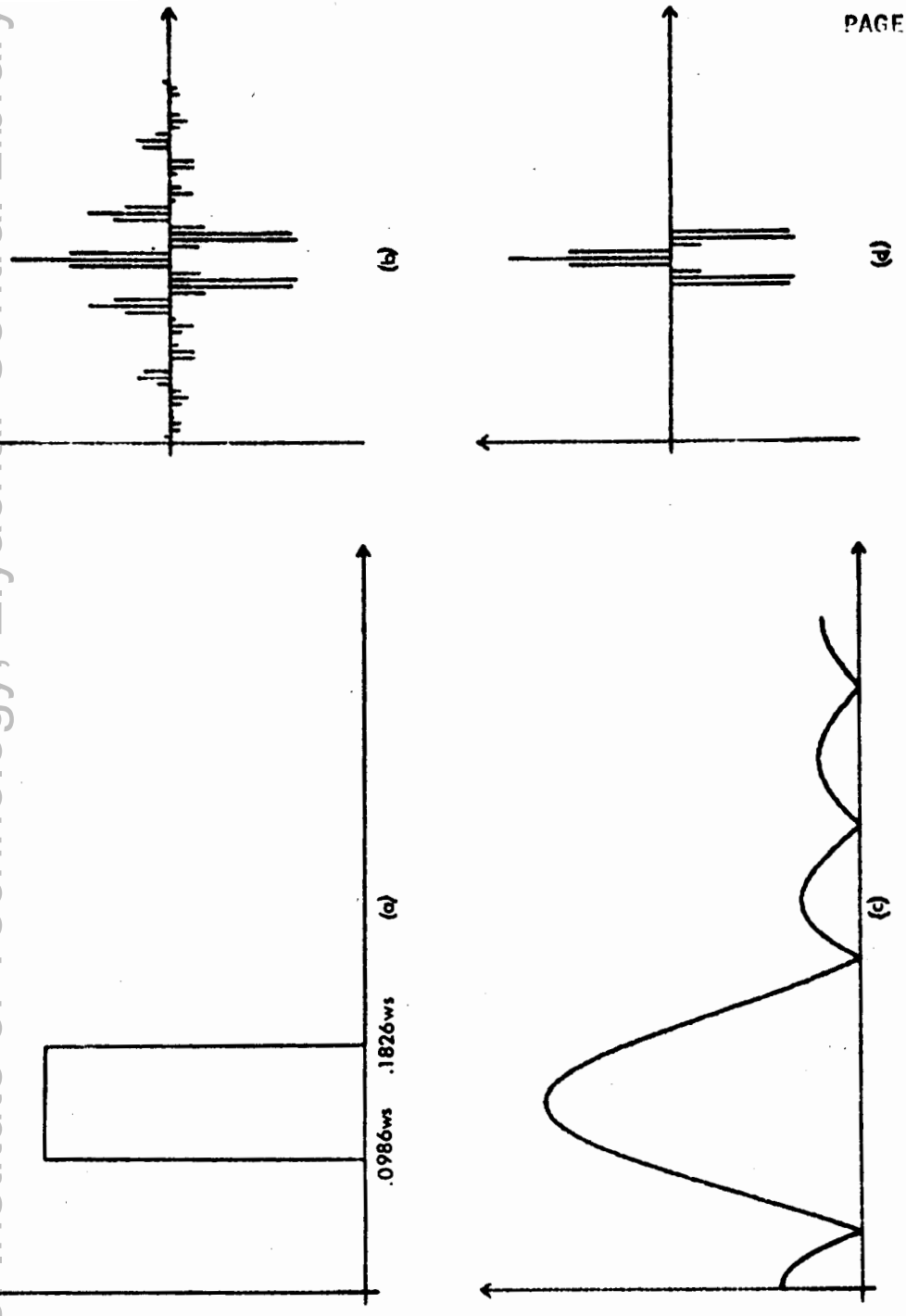


Fig. 2.10

Design Example (Band Pass)

9 Coefficients, Rectangular Window

a) Required Frequency Response

b) Impulse Response of (a)

c) Approximated Frequency Response

d) Impulse Response of (c)

ציור 2.10

דוגמת תכנון (מסנן מעביר סרט)

9 מקדמים, חלון מלבני

a) תגובת התדר הדרוש

b) תגובה להלם של (a)

c) תגובת התדר המקורבת

d) תגובה להלם של (c)

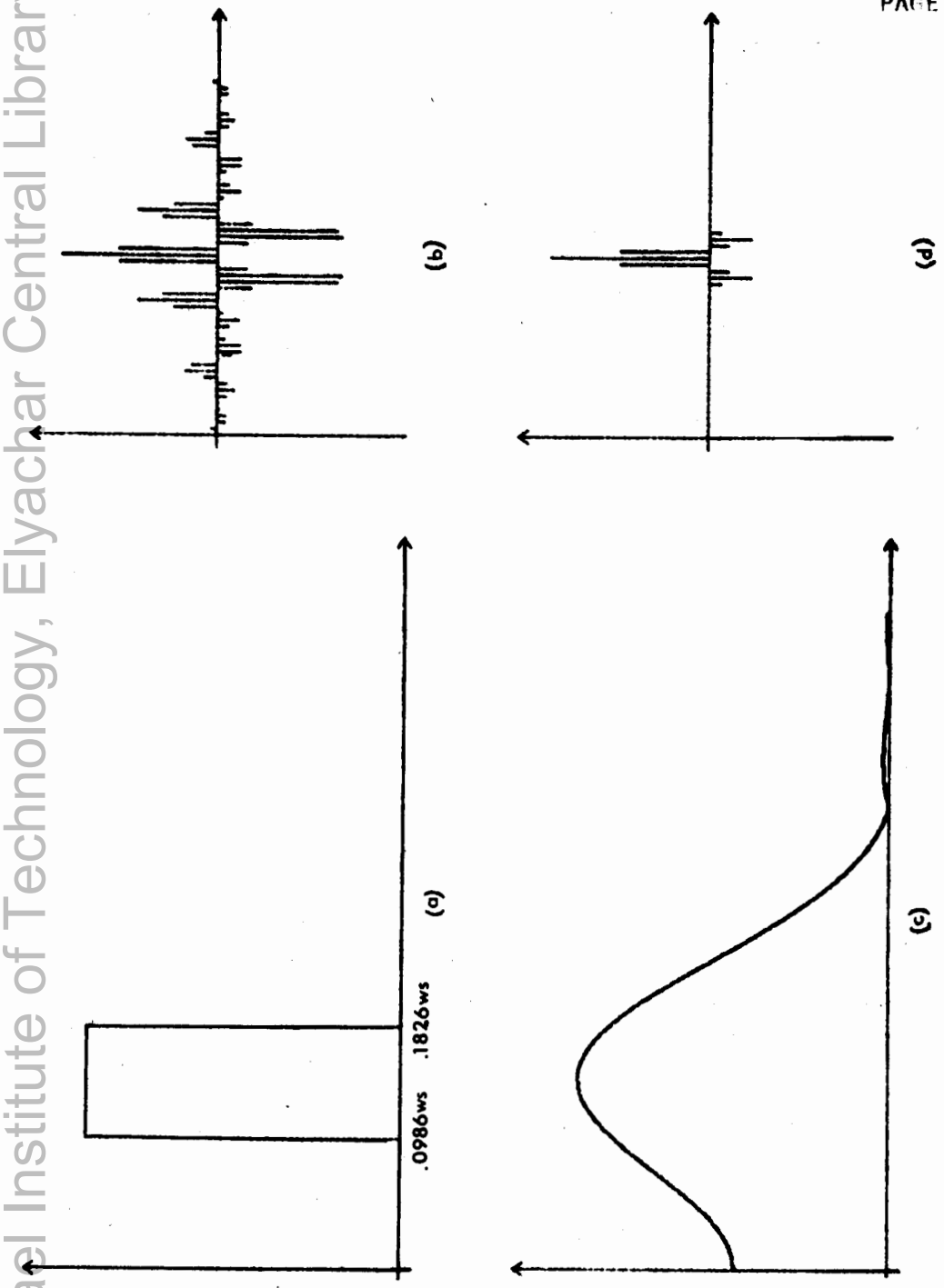


Fig. 2.11

Design Example (Band Pass)

9 Coefficients, Hanning Window

a) Required Frequency Response

b) Impulse Response of (a)

c) Approximated Frequency Response

d) Impulse Response of (c)

צילור 2.11

דוגמת תכנון (מסנן מעביר סרט)

9 מקדמים, חלון Hanning

a) תגובת התדר הדרוש

b) תגובה להלם של (a)

c) תגובת התדר המקורבת

d) תגובה להלם של (c)

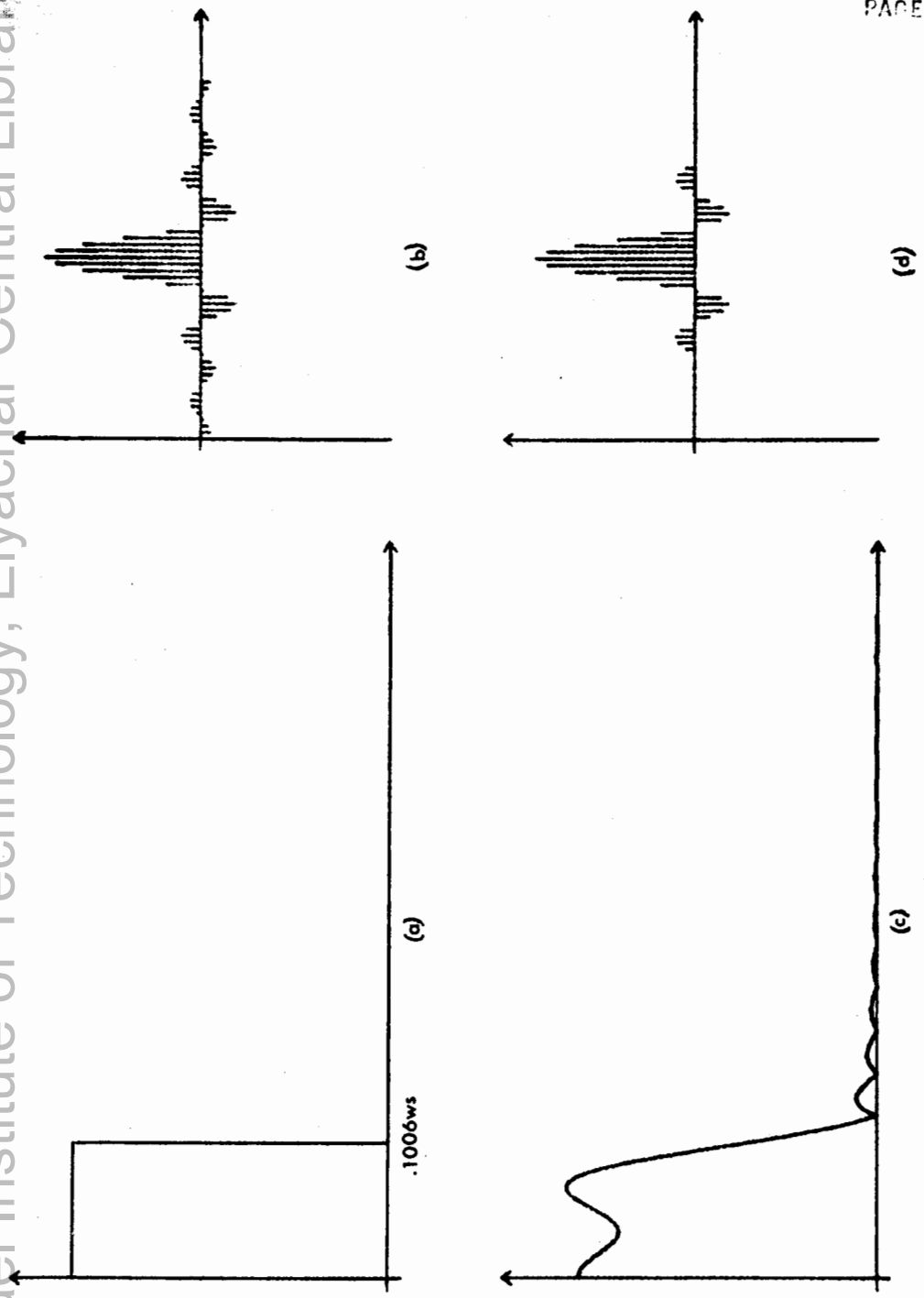


Fig. 2.12

Design Example (Low Pass)

29 Coefficients, Rectangular Window

a) Required Frequency Response

b) Impulse Response of (a)

c) Approximated Frequency Response

d) Impulse Response of (c)

ציור 2.12

דוגמת תכנון (מסנן מעביר עמוכים)

29 מקדמים חלון מלבני

a) תגובת התדר הדרוש

b) תגובה להלם של (a)

c) תגובת התדר המקורבת

d) תגובה להלם של (c)

CHAPTER 3

THE DESIGN AND CONSTRUCTION OF A COMPUTER CONTROLLED TRANSVERSAL FILTER

3.1 Specifications:

a) 10 coefficients such that $y(n)=\sum_{k=1}^{10} h(k)x(n-k)$

b) 8 bit coefficient word length

c) 8 bit data word length

d) Overflow detection and limiting of output

e) Capability of replacing the set of coefficients within one sampling period

f) Sampling rate of at least 10kHz

g) Analog input and output with range (-5,+5) volts

3.2 System Description:

The system consists of three units:

a) The Data General NOVA 2 Mini-computer

b) The Loader Unit

c) The Filter Unit

The block diagram of the system is shown in fig. 3.1.

3.2.1 NOVA 2:

The NOVA 2 is a 16 bit mini-computer with a cycle time of 1.2 μs. Its input/output operations are controlled by two flip-flops (BUSY and DONE) for each peripheral. When a peripheral is free, its BUSY=0 and DONE=1 (logical). An output operation is performed as follows:

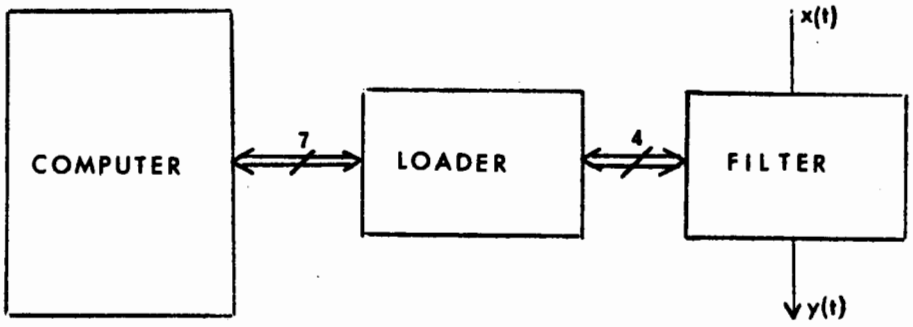


Fig. 3.1
Block Diagram of
Computer Controlled Filter

ציור 3.1
דיאגרמה מלבנית
של המסנן המבוקר ע"י מחשב

a) The data to be output is placed on the I/O bus and a pulse (DOA) is provided. This pulse is used to load the data into a buffer.

b) A second pulse (START) is provided to initiate the peripheral. This pulse sets BUSY (i.e. BUSY=1) and clears DONE (i.e. DONE=0).

c) When the peripheral has completed its task and is ready to accept another datum, it sends a pulse (COMP) to clear BUSY and to set DONE.

d) I/O control in the executing program is performed by sensing the states of the BUSY and DONE flip-flops of the relevant peripheral.

e) An instruction (CLEAR) is available. This produces a pulse which clears both BUSY and DONE and can be used to reset the peripheral.

f) An instruction (IOPULSE) provides the peripheral with a pulse which has no effect on BUSY and DONE.

3.2.2 The Loader:

The purpose of this unit is to accept a set of coefficients, one at a time, from the computer or from a keyboard, to allow editing of these coefficients and, at the command of the computer or operator, to transfer the edited set of coefficients to the filter. Once having performed the transfer, the loader plays no further role and can be disconnected.

3.2.3 The Filter:

This is the special hardware unit which samples the input signal, performs the convolution of the input sequence with the impulse response and converts the output sequence to analog form.

3.3 Description of Filter Hardware:

3.3.1 Design Considerations:

a) The number of coefficients was determined by the minimum number required for the successful linear prediction of speech.

b) The number of bits for the coefficients was influenced by the computer word length. The NOVA 2 has a word length of 16 bits. Thus using 8 bit coefficients, two coefficients can be stored in one computer word.

c) The filter is designed to perform the convolution $y(n)=\sum_{k=1}^{10} h(k)x(n-k)$ in the following order:

$$y(n)=x(n-10)h(10)+x(n-9)h(9)+\ldots+x(n-1)h(1) \qquad (3.1)$$

Since the binary representation of x(n-1) is only required at the end of the computation cycle, the maximum amount of time is available for the analog to digital converter to perform the conversion.

d) The timing for the filter was designed as follows. A master oscillator running at 100 times the desired sampling rate feeds a decoding circuit to provide 10 intervals each of 10 clock periods. One multiplication and addition are performed during each of these 10 intervals. Of the 10 clock periods available

in each interval, 8 are used for the multiplication operation, one for the accumulation addition and one for clearing the multiplier. The timing diagram is shown in fig. 3.9 and is described fully in 3.3.10.

e) The 2's complement fixed point representation of data is used to provide easy handling of bipolar input and output. The permitted values for data and coefficients are $x=-1+n2^{-7}$ $n=0,1,2,\ldots,255$.

f) Since the filter output is to be truncated to 8 bits for the digital to analog converter, the accuracy of each of the ten products needs to be to 12 bits. Thus only the 12 most significant bits of each product are computed and accumulated.

g) Each product is in the range (-1.000,0.774) (oct.) i.e. (-1.0000000,0.1111111) (binary). The range of the filter output is therefore (-12.000,11.730) (oct.). In order to detect and to compensate for overflow, the accumulator was designed to accomodate the range (-20,17.774) (oct.). At the end of the computation cycle, the accumulator contents is tested to see whether it is in the range (-1.000,0.774) (oct.). If not, the constant -1.000 (oct.) or 0.774 (oct.) is output depending on the sign of the accumulator contents.

h) The analog to digital and digital to analog converters were chosen so that an input range of (-5,+5) volts corresponded to the binary range (-1.000,0.774) (oct.).

i) Transistor-transistor logic (TTL) was used for most of the hardware, MOS being used for the storage devices.

3.3.2 Filter Block Diagram:

The filter can be divided into the following modules:

a) Sample and hold
b) Analog to digital converter
c) Storage of data and coefficients
d) Multiplier
e) Accumulator
f) Overflow detector and output limiter
g) Digital to analog converter
h) Master oscillator and control

The block diagram of the filter unit is shown in fig. 3.2.

3.3.3 Sample and Hold:

This is a digitally controlled analog circuit the purpose of which is to provide the analog to digital converter with a constant level input while it is performing the conversion.

The aperture time of the analog to digital converter, i.e. the time interval during which its analog input is sensitive to level variations, is the time during which it performs a conversion. Since its aperture time is of the order of 40μs, level changes in the input signal can cause the conversion to be erroneous. The sample and hold circuit, which has an aperture time of 1μs, is used to keep the signal level constant during the time a conversion is being performed.

Control is provided by a logic signal from the control module. When this signal is high, the output of the sample and hold follows the input. When the control signal goes low, the

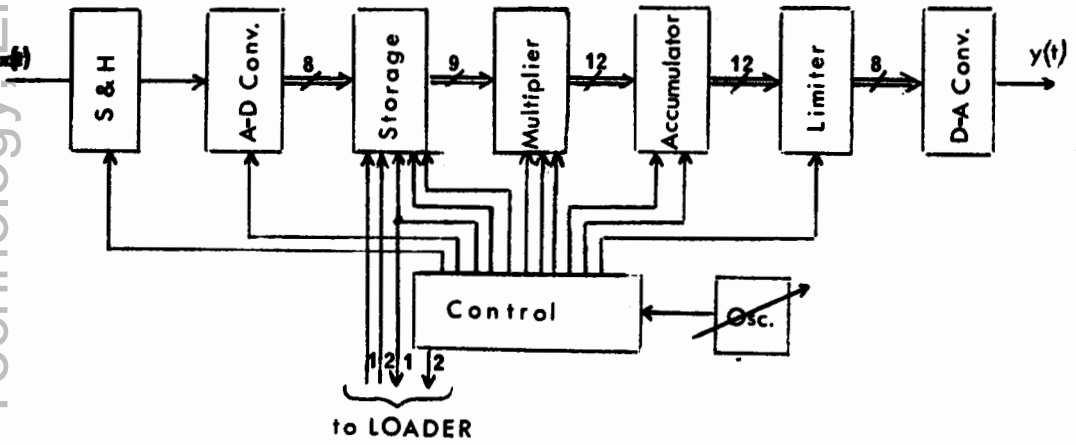


Fig. 3.2

Block Diagram of Filter Unit

3.2 ציור

דיאגרמת מלבנים של יחידת המסנן

output is held constant.

This module was built by the Signal Processing Laboratory at the Technion.

All inputs to modules are given with the source of the input in brackets while outputs are given with their destinations.

Inputs:

1) Analog input (Filter Input)
2) Control (Control Module)

Outputs:

1) Analog output (Analog to Digital Converter)

3.3.4 Analog to Digital Converter:

A commercial CMOS 8 bit device performs this function. The input range for the device is (0,+10) volts, the output range (00000000,11111111) (bin.), the transfer function being linear. Since an input range of (-5,+5) volts is required, the input signal must be shifted by +5V. A 2's complement format output is obtained by inverting the most significant bit, thus giving an output range of (10000000,01111111) (bin.).

The conversion time is 40μs. Since the converted datum is to be used during the same sampling interval (so that the delay between the filter input and output is one sampling period, an essential requirement for when the filter is to be operated in a closed loop), the minimum allowable sampling period is somewhat greater than the conversion time. In the filter design, up to 80% of the sampling period is allowed for the

conversion, setting the maximum sampling rate at 20kHz. A data sheet is provided in Appendix C.

The peripheral circuitry for this module (see data sheet) was built by the Signal Processing Laboratory.

Inputs:

1) Analog input (Sample & Hold)
2) Start convert (Control Module)

Outputs:

1-8) 2's complement binary datum (Storage)

3.3.5 Storage:

This module provides cyclic storage for both data and coefficients. It is shown in fig. 3.3.

10 multiplications are performed during each sampling period. After 8 of these have taken place, the digital output of the analog to digital converter is inserted into the data loop. During the 9th multiplication, this new datum is clocked into the multiplicand latch where it will be available for the 10th multiplication. Since the data loop has only 9 data, after each sampling period the data in the loop has effectively advanced by one datum.

The coefficient loop contains 80 bits for the 10 coefficients. The selector is controlled by the loader. The normal mode of operation is with the loop closed.

Inputs:

1) Coefficient load control (Loader Unit)
2) Coefficient data input (Loader Unit)

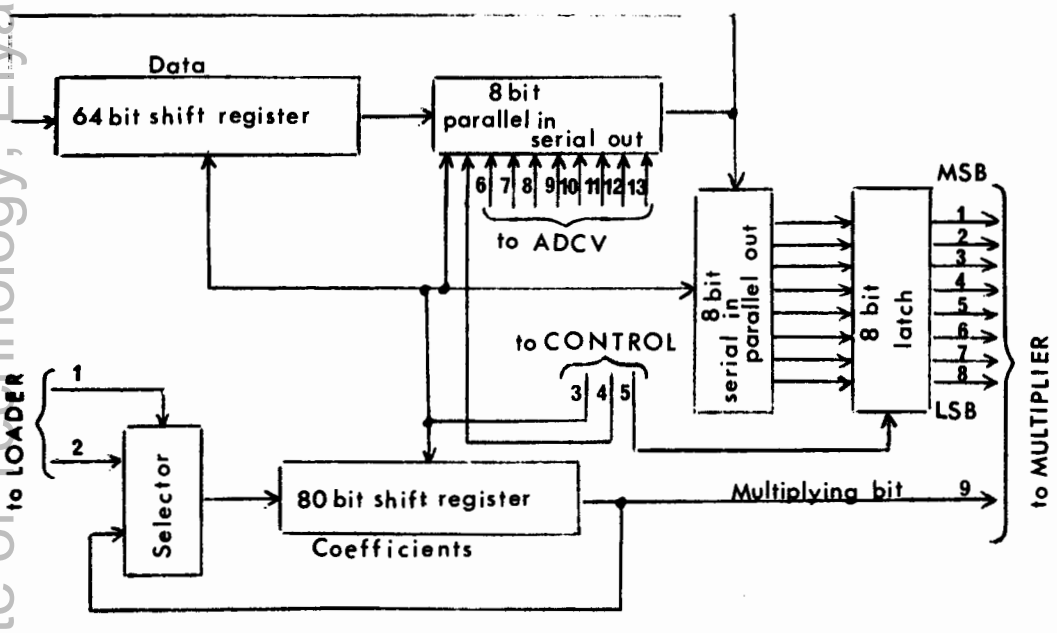


Fig. 3.3

Filter Storage Module

3.3 ציור

תת יחידה לאגירת נתונים במסנן

3) Shift register clock (Control)

4) Parallel input control (Control)

5) Latch control (Control)

6-13) Digitized data input (Analog to Digital Converter)

Outputs:

1-8) Multiplicand (Multiplier)

9) Multiplying bit (Multiplier)

3.3.6 Multiplier:

Let B(X,k) be the k th bit of word X such that B(X,1) denotes the LSB and B(X,8) denotes the MSB.

Let S(X,C) be the result of shifting word X one bit to the right with carry C i.e. S(0110,0)=0011 and S(0110,1)=1011.

Let MP=multiplier, MC=multiplicand, $\overline{MC}$=2's complement of MC and Product=MP*MC. The multiplication algorithm can be represented by the flowchart in fig. 3.4. The output product is the exact 16 bit product truncated to 12 bits.

The logic diagram of the multiplier is shown in fig. 3.5. The zero-one-true-complement device provides a choice of four outputs: all lines logical 0 (controls a=0,b=0), all logical 1 (a=0,b=1), true version of input (a=1,b=0) and 1's complement of input (a=1,b=1). If the MSB of the multiplying datum is logical 1 (indicating a negative number), the 2's complement of the multiplicand is required. This is obtained by making the carry input of the adder logical 1, thus effectively adding 1 to the LSB of the 1's complement. The one bit latch performs the function of CARRY in the flowchart of fig. 3.4.

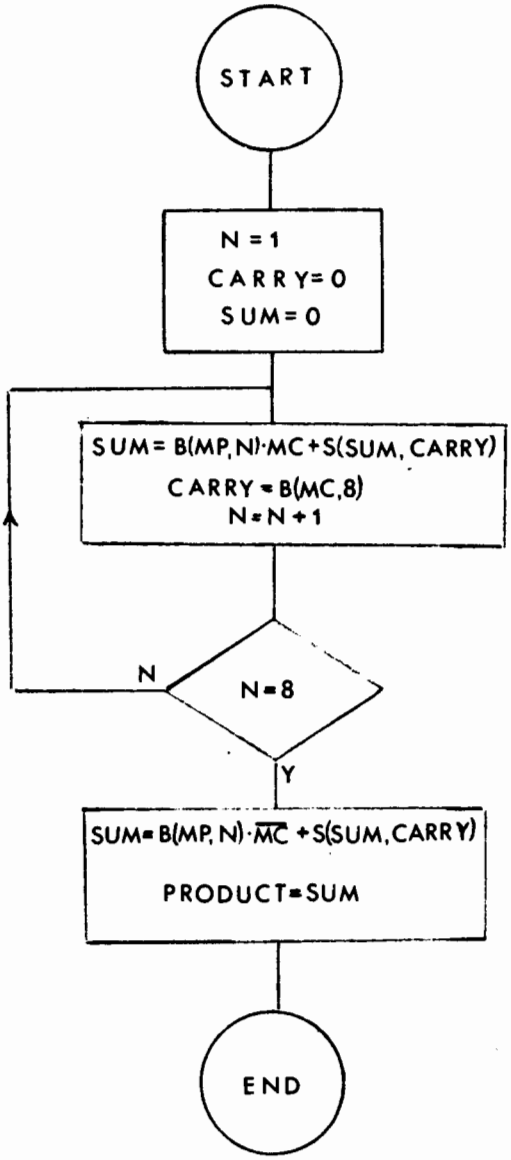


Fig. 3.4
Flowchart of Multiplier Algorithm

3.4 ציור
דיאגרמת זרימה של
אלגוריתם המכפל

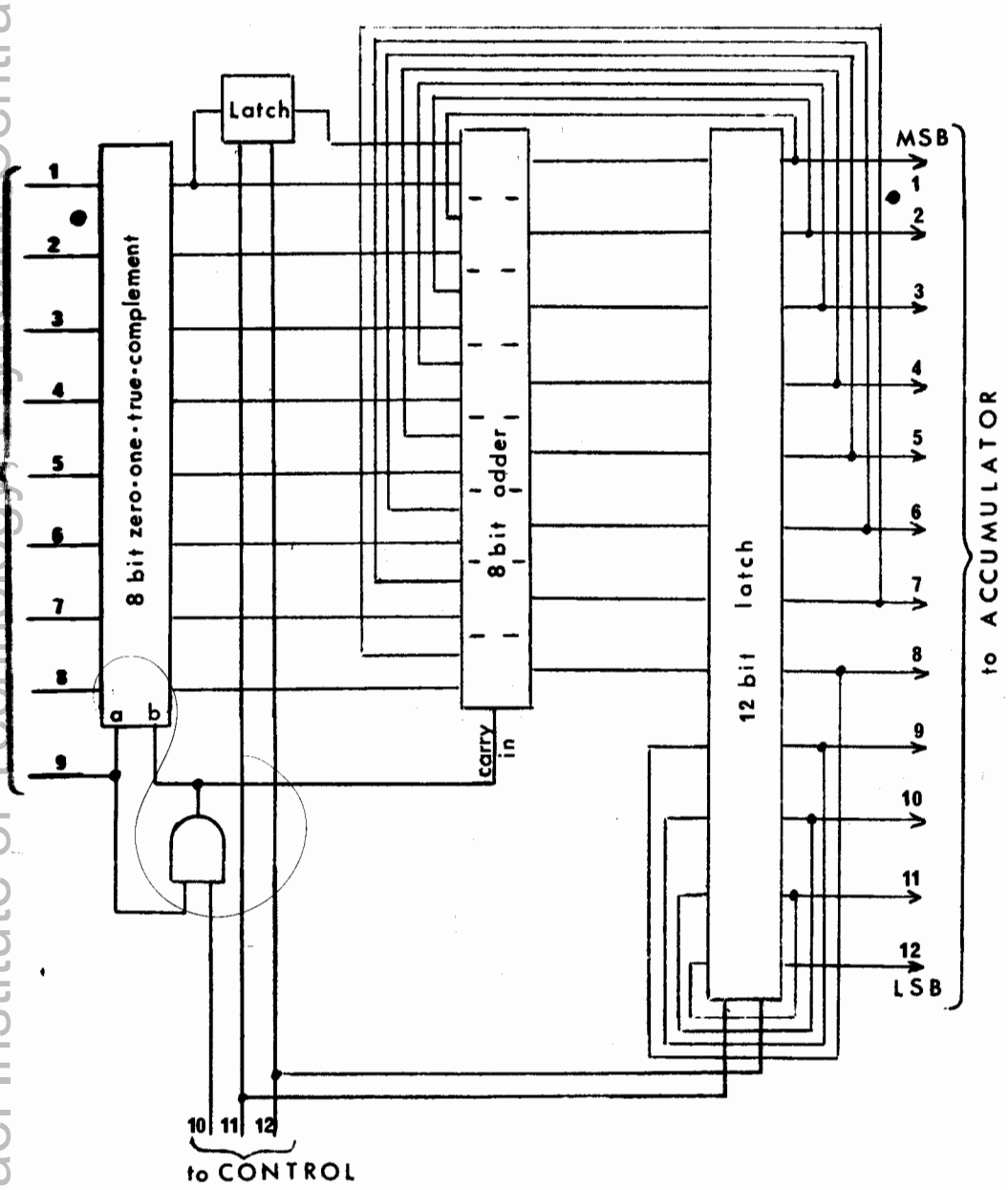


Fig. 3.5

Filter Multiplier Module

ציור 3.5

תת יחידת המכפל במסנן

Inputs:

1-8) Multiplicand (Storage)

9) Multiplying bit (Storage)

10) Sign bit; indicates MSB of multiplying bit (Control)

11) Latch control (Control)

12) Clear (Control)

Outputs:

1-12) Product (Accumulator)

3.3.8 Accumulator:

This is simply an adder-latch combination which can also be cleared. The input range is (-1.0000,0.7776) (oct.), the output range (-20.000,17.774) (oct.). The logic diagram is shown in fig. 3.6.

Inputs:

1-12) Product (Multiplier)

13) Latch control (Control)

14) Clear (Control)

Outputs:

1-12) Running total (Limiter)

3.3.7 Overflow Detector and Output Limiter:

The purpose of the detector is to examine the output of the accumulator after the final addition and to decide whether the output is in the range (-1.000,0.774) (oct.) or not. This is accomplished by examining the 5 most significant bits of the output. If all 5 bits have the same logical value, then no

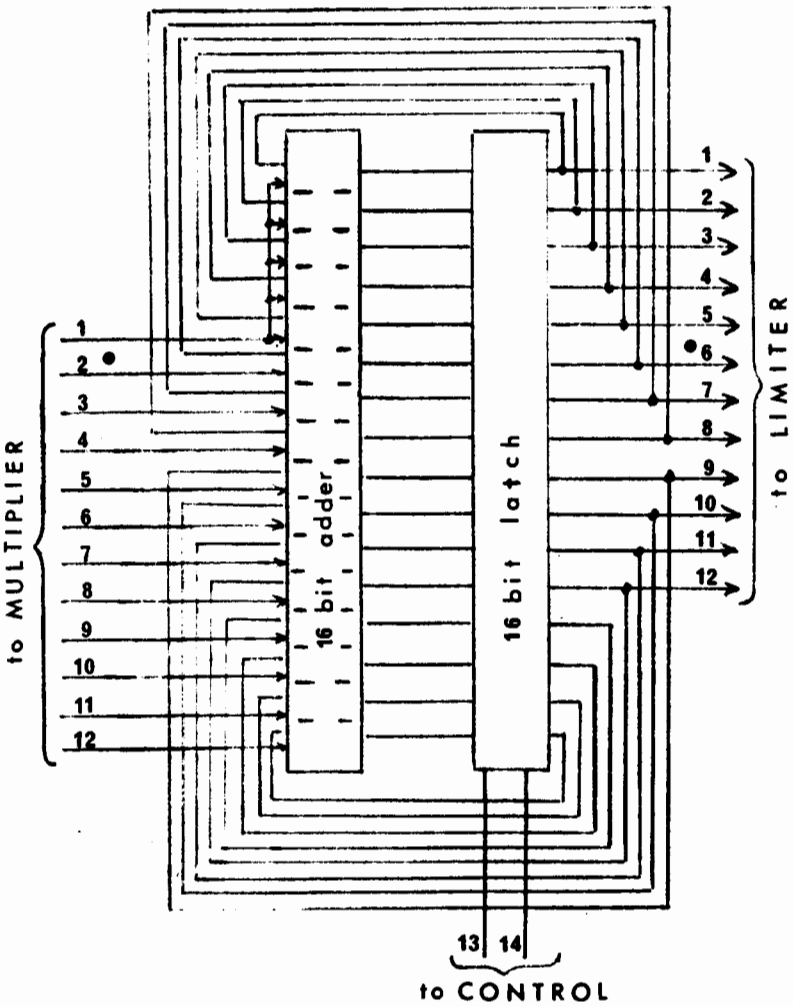


Fig. 3.6

Filter Accumulator Module

ציור 3.6

תת יחידת האקומולטור במסנן

overflow correction is required. If, howevever, the 5 bits differ in any way from one another, then overflow has occured. Then, instead of loading the output buffer with the accumulator contents, one of the two constants -1.000 (oct.: or 0.774 (oct.) is loaded, the former if the most significant of the 5 bits is high and vice versa.

The logic diagram of the overflow detector and output limiter is shown in fig. 3.7. The output of the OR gate is low when overflow is detected.

Inputs:

1-12) Accumulator total (Accumulator)

13) Latch control (Control)

Outputs:

1-8) Output (Digital to Analog Converter)

3.3.9 Digital to Analog Converter:

The device used is a standard commercial 8 bit 2's complement converter with bipolar output. A data sheet is provided in Appendix C.

Inputs:

1-8) 2's complement binary datum (Limiter)

Outputs:

1) Analog output (Filter Output)

3.3.10 Master Oscillator and Control:

A master oscillator with a frequency range of 20kHz to 2MHz feeds two divide by ten counters in cascade, the output of

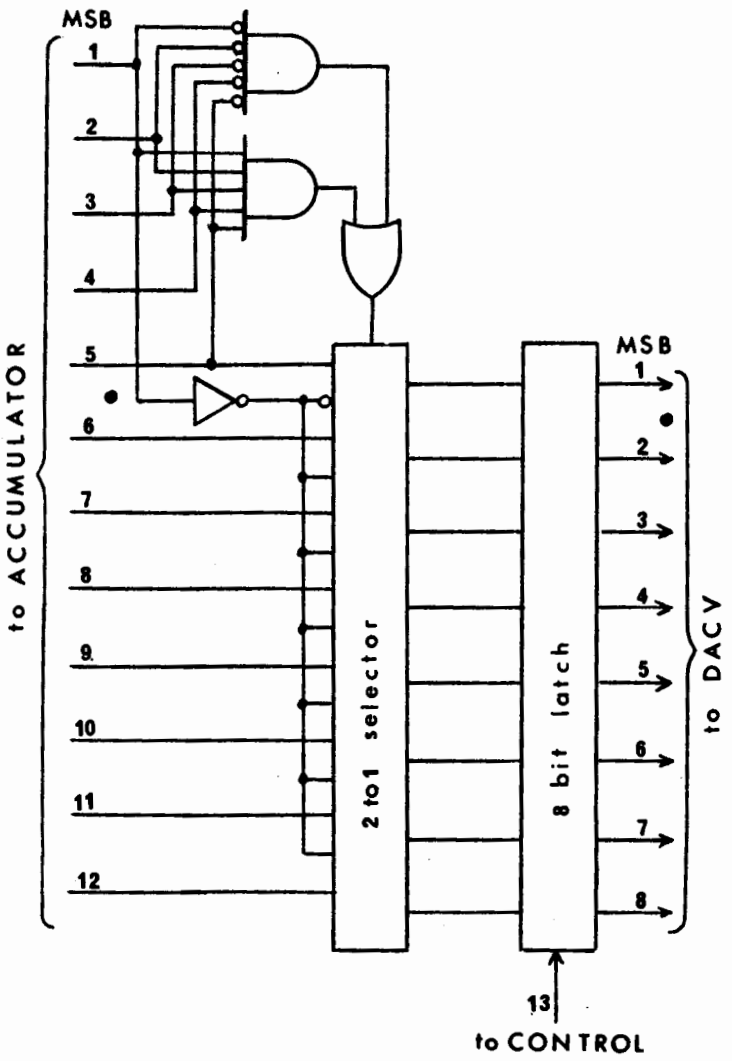


Fig. 3.7
Filter Limiter Module

3.7 ציור
תת יחידת המגבל במסנן

each of which is used as the input of a BCD (binary coded decimal) to decimal decoder. This provides a flexible method of selecting any of the 100 clock periods in each sampling period for control purposes. E.g. if pulse #53 is required for a particular function, outputs A3 and B5 are used (see fig. 3.8) while if pulse #79 is required as in control output #8, A9 and B7 are used.

The decoding circuits provide the following control signals:

a) The control signals for the sample and hold circuit and analog to digital converter

b) The clock pulses for the shift registers in the storage module and loader unit

c) The loading of the registers and latches in the storage module

d) The timing for the multiplier module

e) The timing for the accumulator module

f) The loading of the output buffer

g) Ready pulse for the loader unit

The oscillator was built by the Signal Processing Laboratory. The logic diagram of the control module is shown in fig. 3.8. The control signals are shown in fig. 3.9.

In the following list the number in brackets refers to the respective control input of named module.

Outputs;

1) Master oscillator

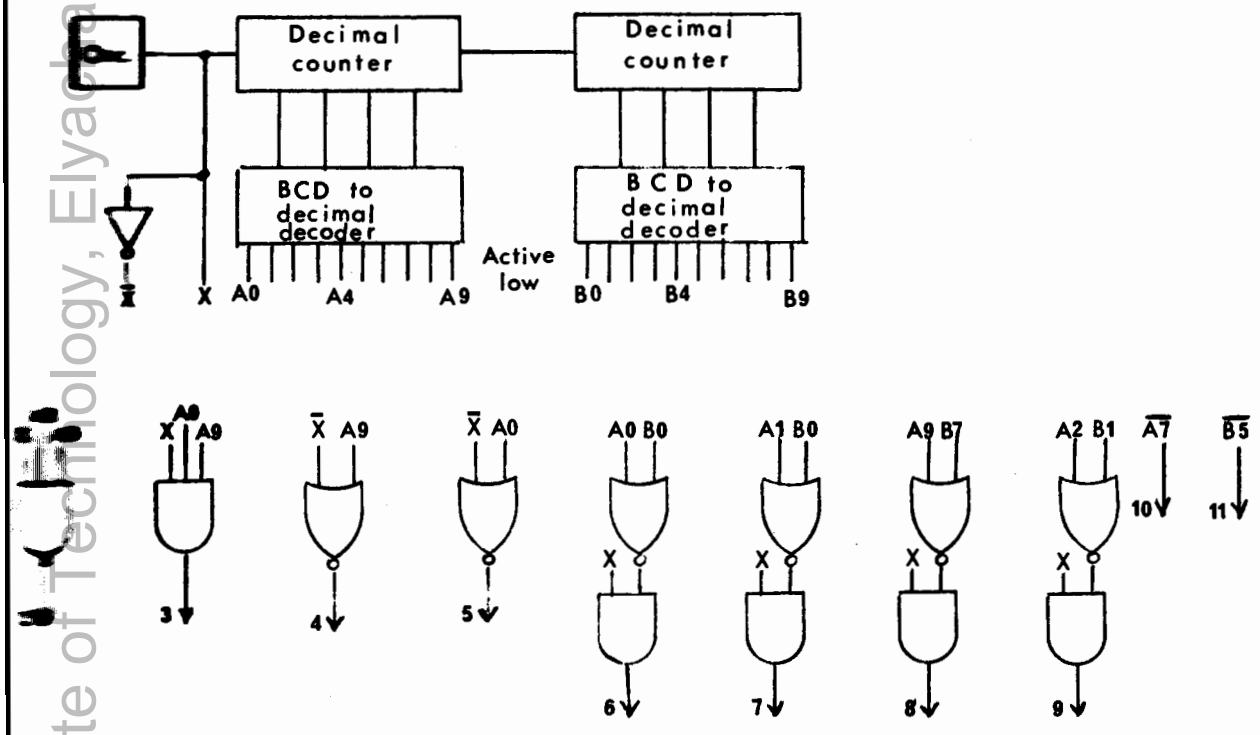


Fig. 3.8
Counter Control Module

ציור 3.8
תת יחידת הבקרה במסנן

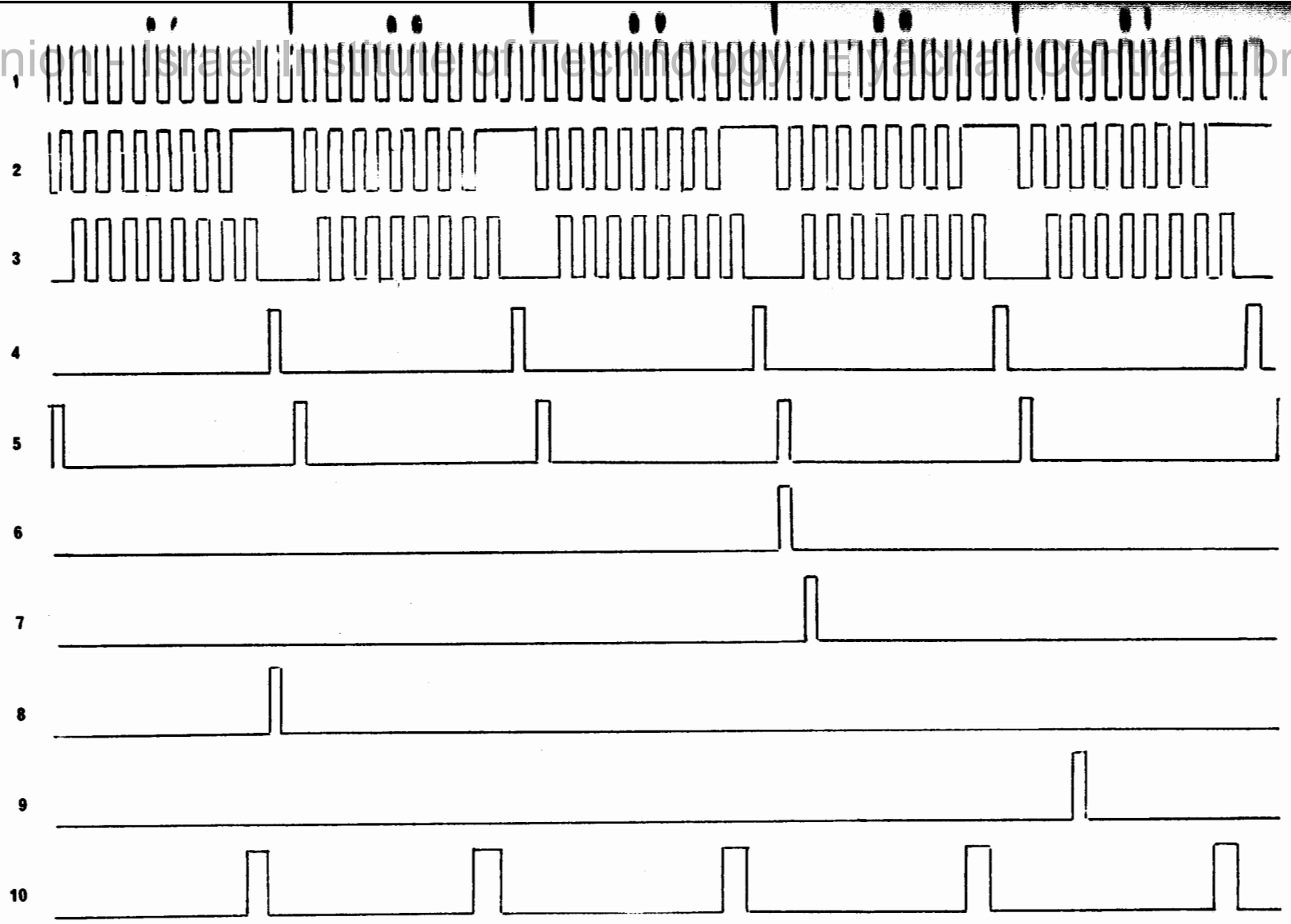


Fig. 3.9
Filter Timing Diagram

ציור 3.9
דיאגרמת זמון של המסנן

2) Storage(3), Loader Unit(1), Computer(1)

3) Multiplier(11)

4) Storage(4), Accumulator(13)

5) Multiplier(12)

6) Sample & Hold(2), Limiter(13), Loader Unit(2)

7) Accumulator(14)

8) Storage(5)

9) Analog to Digital Converter(2)

10) Multiplier(10)

11) Exciter(1)

3.4 Description of Loader Hardware:

3.4.1 Loader Block Diagram:

The loader can be divided into the following modules:

a) Storage

b) Keyboard Encoder

c) Edit Control

d) Load Control

e) Interface Control

The block diagram of the loader unit is shown in fig. 3.10.

3.4.2 Storage:

The storage module provides serial storage for 80 bits (i.e. 10 coefficients), 8 of which are contained in a parallel in/parallel out shift register. The contents of this register is displayed as two hexadecimal digits each on a 7 segment

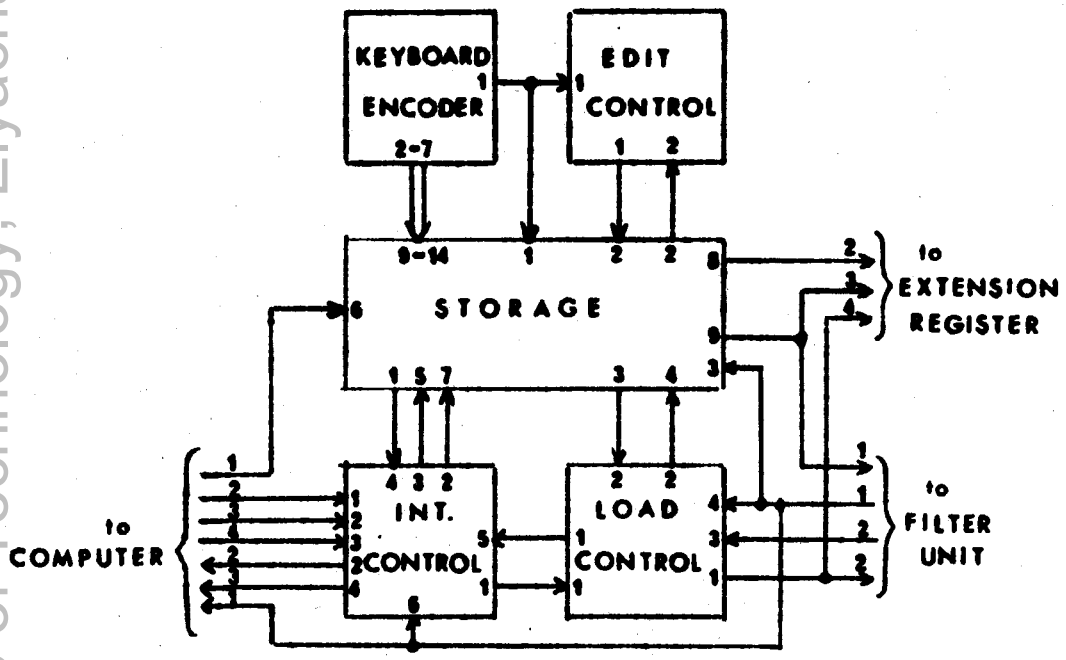


Fig. 3.10
Block Diagram of Loader Unit

ציור 3.10
דאגרמת מלבנים של יחידת מטען תמקרמים

display. Each of these digits can be altered on the keyboard. A third display is used for the number of the coefficient in the register.

Three counters are employed to keep track of the position the data in the loop. Counter A divides by 8 and signals the last bit of each word (END OF WORD). Counter B divides by 9 and is used to signal the end of 9 words (END OF 9 WORDS). Counter C divides the output of counter A by 10 and signals the end of the coefficient sequence (END OF SEQUENCE) as well as being used by the coefficient number display. Since the coefficients are in descending order, the counter's output is inverted and the constant 1001 (bin.) added to give a displayed output of 0 to 9.

Logic is included for extending the storage register. This is used in the speech synthesizer where 16 bits of data are passed on to the exciter. The logic diagram is shown in fig. 3.11.

Inputs:

1) Edit clock (Keyboard Encoder)
2) Shift (Edit Control)
3) Filter shift register clock (Filter Control Module)
4) Shift (Load Control)
5) Reset (Interface Control)
6) Data (Computer)
7) Computer-loader transfer (Interface Control)
8) Load left byte (Keyboard Encoder)

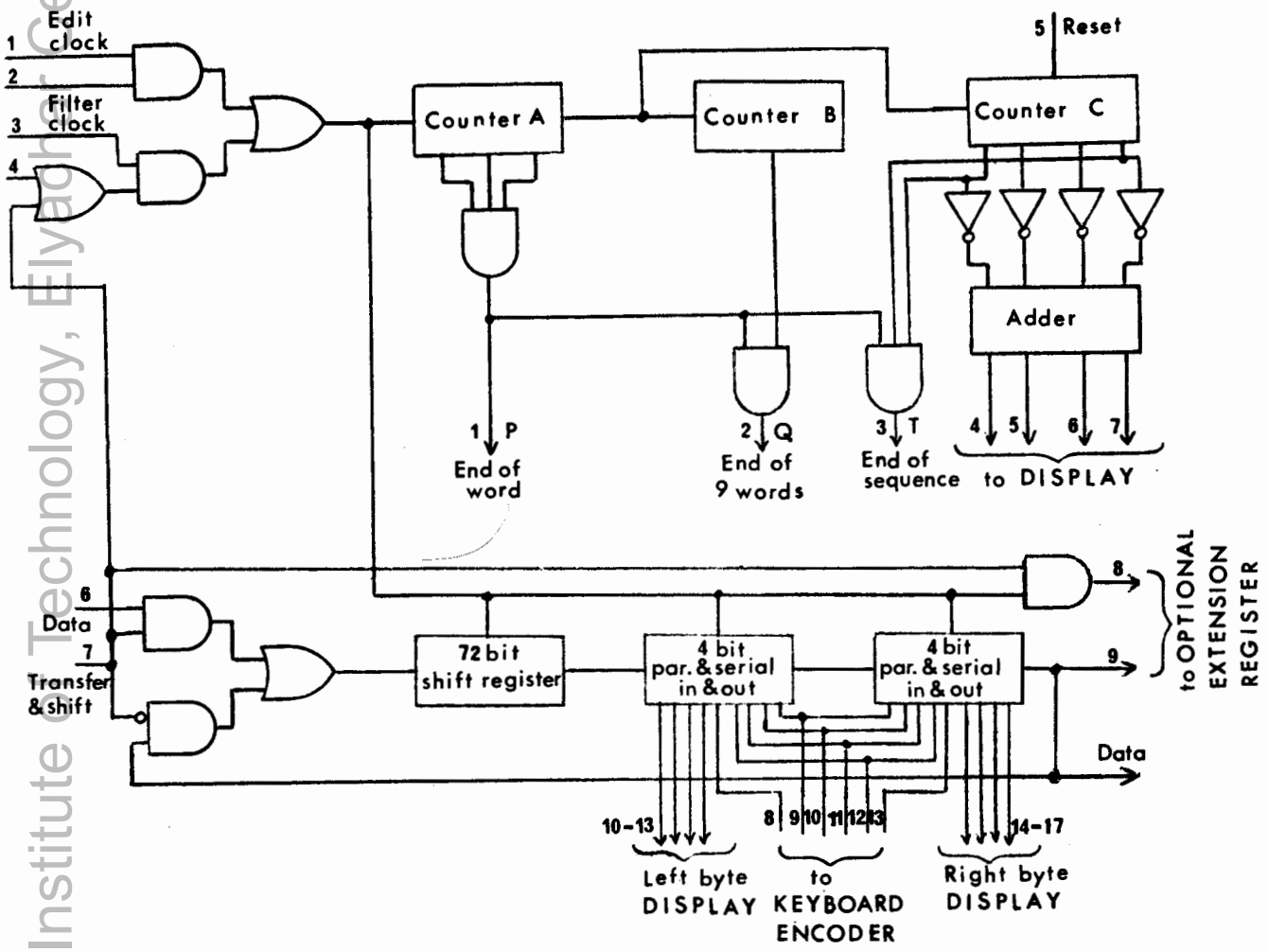


Fig. 3.11
Loader Storage Module

3.11 ציור
תת יחידת האגירה במטען

9-12) Key code (Keyboard Encoder)

13) Load right byte (Keyboard Encoder)

Outputs:

1) End of word (Interface Control)

2) End of 9 words (Edit Control)

3) End of sequence (Load Control)

4-7) Coefficient number (Display)

8) Clock (Extension Register)

9) Data (Extension Register, Filter Unit)

10-13) Left byte (Display)

14-17) Right byte (Display)

3.4.3 Keyboard Encoder:

The purpose of the keyboard encoder is to detect the depression of one of the keyboard's 16 keys, put the code of the depressed key onto 4 data lines and to provide a latching pulse. This is performed by scanning the keyboard at a low rate (400Hz) using a 16 line selector. When a depressed key is detected, the selector output is high and the state of that key at the previous scan determines whether or not a latching pulse is to be generated. Only if that key was not depressed at the previous scan is a latching pulse produced. The memory is provided by a 16 bit shift register. The latching pulse is alternately applied to the left and right halves of the 8 bit register mentioned in 3.4.2. The logic for the encoder is shown in Fig. 3.12.

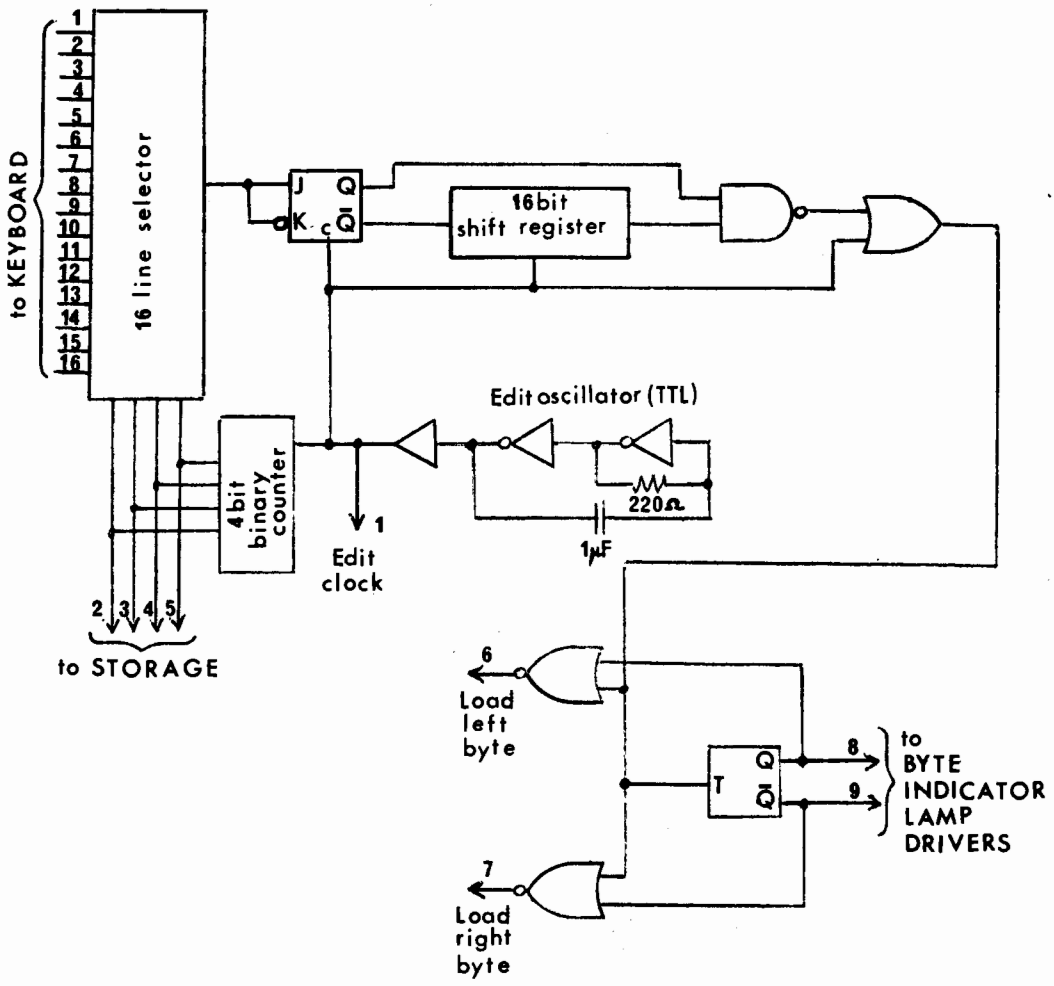


Fig. 3.12

Loader Keyboard Encoder Module

3.12 ציור

תת יחידת מצפן המקשים במטען

Inputs:

1-16) Panel keyboard switches

Outputs:

1) Edit clock (Storage, Edit Control)

2-5) Key code (Storage)

6) Load left byte (Storage)

7) Load right byte (Storage)

8) Right byte indicator (Panel Lamp Driver)

9) Left byte indicator (Panel Lamp Driver)

3.4.4 Edit Control:

The natural order of the coefficients in both the filter and loader is from h(10) to h(1). In order to be able to display the coefficients in ascending order, a panel switch is provided whereby each depression advances the data cyclicly by 72 bits. On the display, therefore, the coefficients appear from h(1) to h(10).

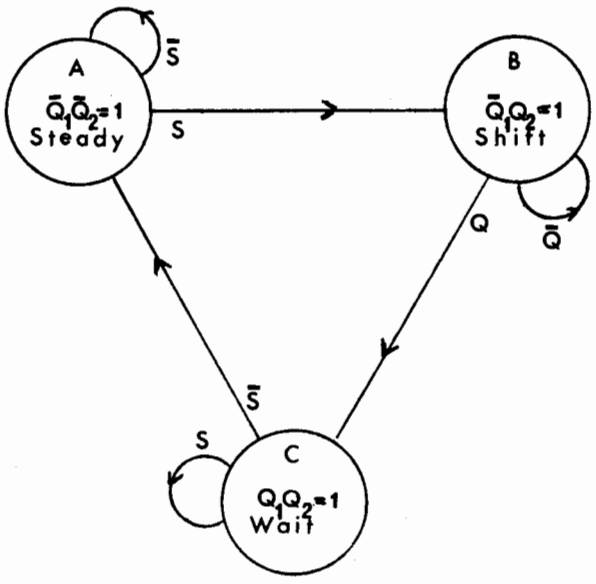
The edit control consists of a 3 state machine. The states of the machine are:

a) Steady state ---- changes to state B when step switch depressed

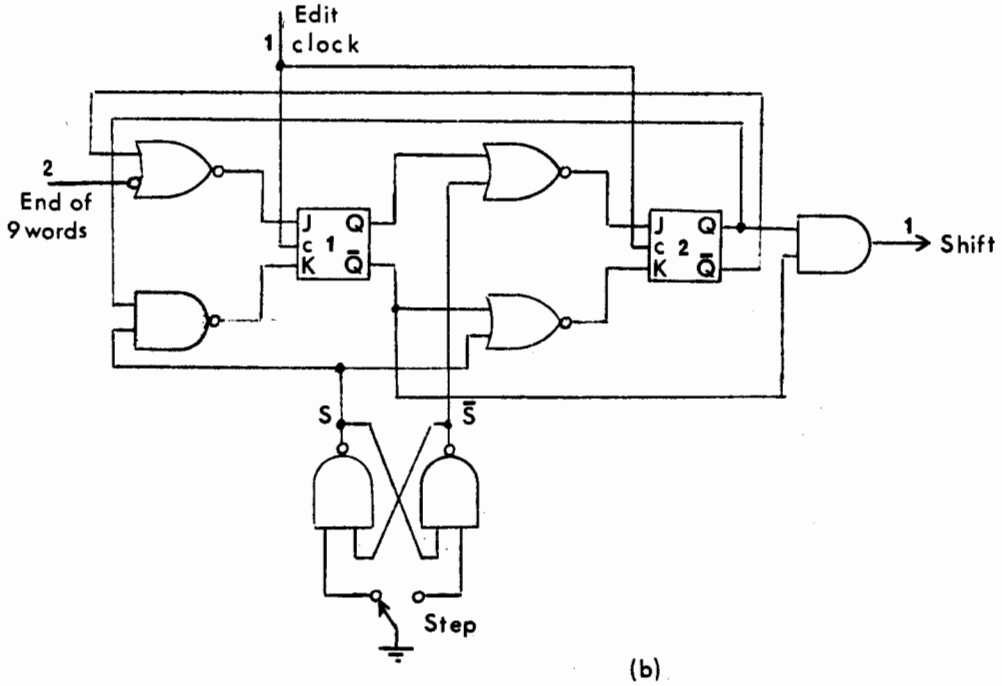
b) Shift ---- data are shifted until the END OF 9 WORDS signal is received

c) Wait ---- return to state A when step switch is released

The state diagram of the machine and logic diagram of the module are shown in figures 3.13(a) and 3.13(b) respectively.



(a)



(b)

Fig. 3.13
Loader Edit Control Module
a) State Diagram
b) Logic Diagram

ציור 3.13
תת יחידת בקרת העריכה במטען
a) דיאגרמת מצבים
b) דיאגרמה לוגית

Inputs:

1) Edit clock (Keyboard Encoder)

2) End of 9 words (Storage)

Outputs:

1) Shift (Storage)

3.4.5 Load Control:

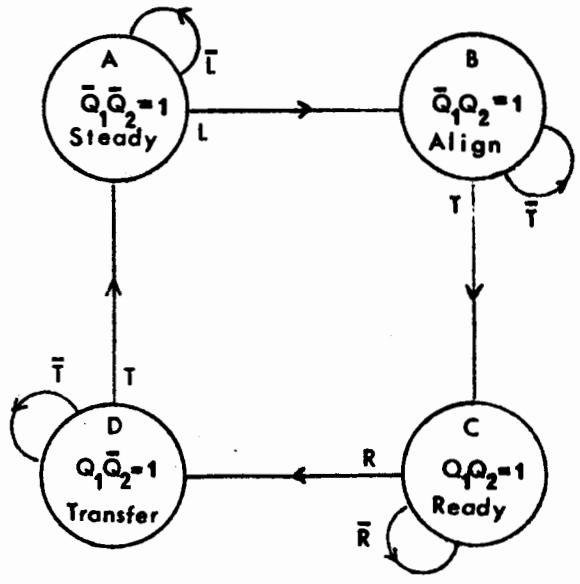
When the LOAD command for transfer of the ten coefficients in the loader unit to the filter unit is given either by the computer (an IOPULSE followed by a START pulse) or by the load switch on the loader, the following sequence of operations takes place:

a) The data in the loader is aligned so that the first bit to be output will be the LSB of coefficient h(10) (.01 to 1 sampling period).

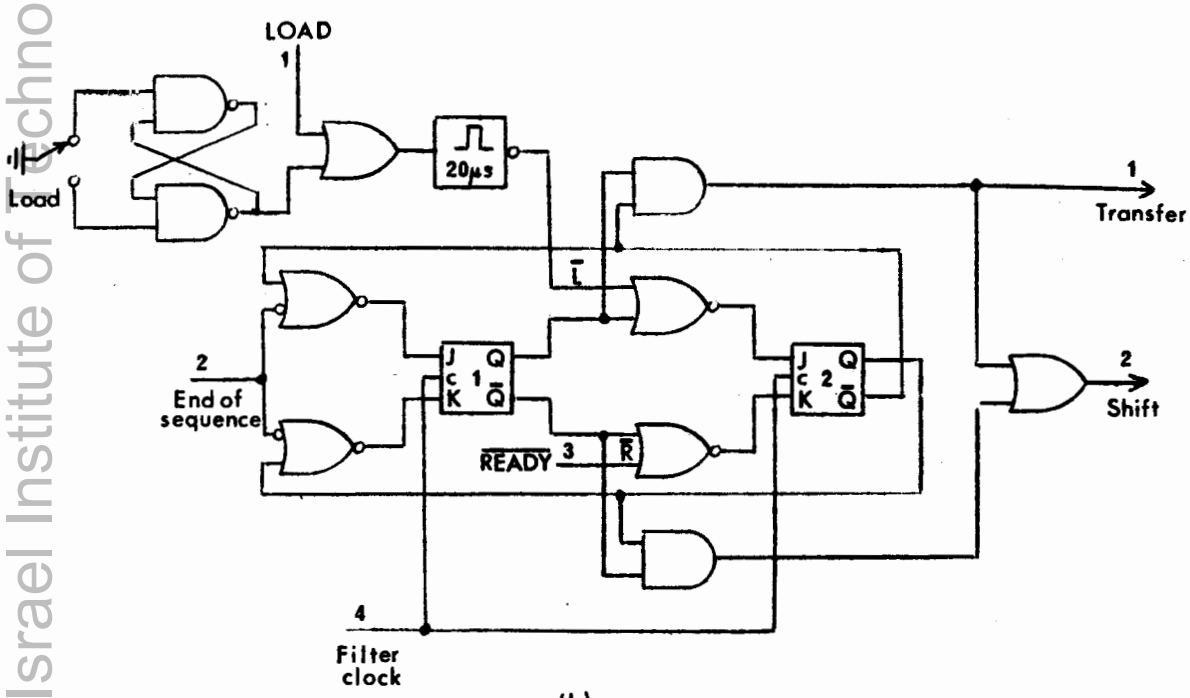
b) The load control waits for a signal (READY) from the filter signaling the beginning of a computation cycle (0 to .99 sampling period).

c) 80 bits of data are transfered serially to the filter (1 sampling period).

This sequence of operations is controlled by a four state machine, the state diagram of which is shown in fig. 3.14(a). The full sequence takes from 1.01 to 2.99 sampling periods. Note, however, that the actual replacement of the coefficients is effected in one sampling period. The logic diagram of the module is shown in fig. 3.14(b).



(a)



(b)

Fig. 3.14

Loader Load Control Module

a) State Diagram

b) Logic Diagram

ציור 3.14

תת יחידה בקרת הטעינה במטען

a) דיאגרמת מצבים

b) דיאגרמה לוגית

Inputs:

1) Load command (Interface Control)
2) End of sequence (Storage)
3) Ready signal (Filter Control Module)
4) Filter clock (Filter Control Module)

Outputs:

1) Loader-filter transfer (Interface Control, Filter Storage Module)
2) Shift (Storage)

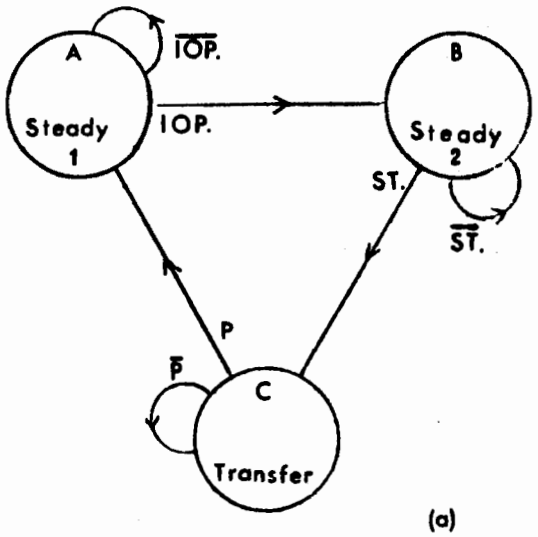
3.4.6 Interface Control:

The purpose of this module is to control the output of data from the computer to the loader and to initiate a LOAD sequence. When a START pulse is received, 8 bits of data, i.e. one coefficient, is transfered serially from the output buffer situated in the computer to the loader storage module. This having been done, the interface control sends a pulse (COMP) to the computer to set DONE and clear BUSY. When, however, first an IOPULSE and then a START pulse is received, a LOAD sequence is initiated. At the completion of the LOAD sequence a COMP pulse is sent to the computer.

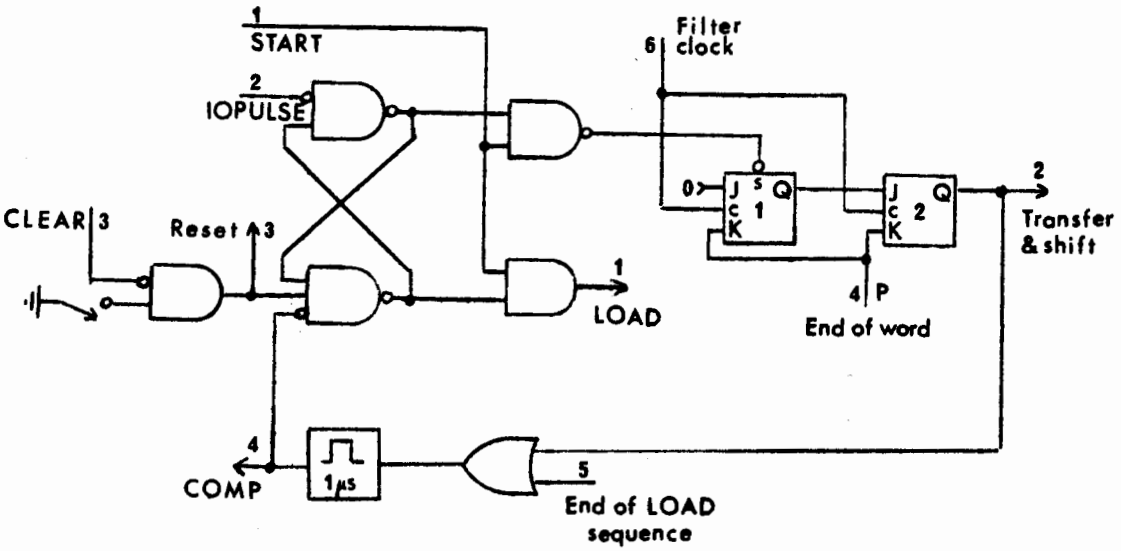
The state diagram of the interface module is shown in fig. 3.15(a). The logic diagram is shown in fig. 3.15(b).

Inputs:

1) Begin task (Computer)
2) Special purpose pulse (Computer)
3) End of word (Storage)



(a)



(b)

Fig. 3.15

Loader Interface Control Module

a) State Diagram

b) Logic Diagram

ציור 3.15

תת יחידת הבקרה של יחידת התאום במטען

a) דיאגרמת מצבים

b) דיאגרמה לוגית

4) Filter loaded (Load Control)

5) Filter clock (Filter Control Module)

Outputs:

1) Load filter (Load Control)

2) Computer-loader transfer (Storage, Computer)

3) Task completed (Computer)

3.5 Description of Computer Output Hardware:

The output buffer of the computer is a parallel & serial in & out shift register. When the mode control of this shift register is logical 0, a pulse on the clock input causes the data on the data bus to be latched. When the mode control is logical 1, the data can be clocked out serially. The logic diagram of the computer output stage is shown in fig. 3.16. When the transfer input is low, a DOA pulse causes latching of the data.

Inputs:

1) Filter shift register clock (Filter Control Module)

2) Computer-loader transfer (Loader Interface Control Module)

3) Task completed (Loader Interface Control Module)

Outputs:

1) Data (Loader Storage Module)

2) Begin task (Loader Interface Control Module)

3) Special purpose pulse (Loader Interface Control Module)

4) Clear (Loader Interface Control Module)

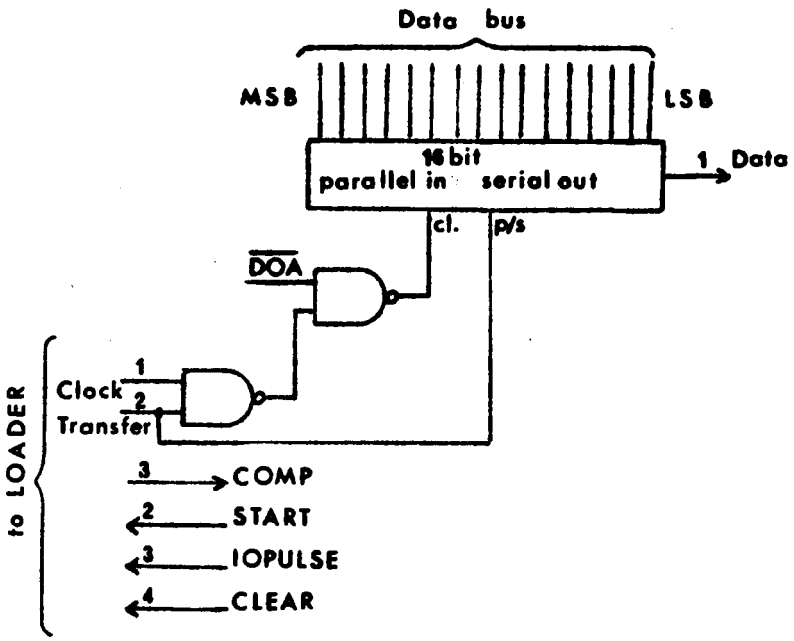


Fig. 3.16
Computer Output Hardware

3.16 ציור
חמרת יציאת המחשב

3.6 Characteristics of the Transversal Filter:

The transversal filter as built can be operated in two modes, open and closed loop. The loop is implemented using an operational amplifier with an option of two loop gains, 1 and 4.

3.6.1 Open Loop:

Difference equation:

$$y(n)=\sum_{k=1}^{10} h(k)x(n-k) \tag{3.2}$$

Transfer function:

$$H(z)=\sum_{k=1}^{10} h(k)z^{-k} \tag{3.3}$$

This filter has only zeros in the finite z-plane and is thus always stable. It has a finite duration impulse response of not more than 10 sampling periods.

3.6.2 Closed Loop:

The transversal filter can also be used in the mode shown in fig. 2.7.

Difference equation:

$$y(n)=x(n)+\sum_{k=1}^{10} h(k)y(n-k) \tag{3.4}$$

Transfer function:

$$H(z)=1/(1-\sum_{k=1}^{10} h(k)z^{-k}) \tag{3.5}$$

This filter has only poles (except possibly for zeros at the origin). Its stability thus depends upon all its poles lying within the unit circle. It has an infinite duration impulse response.

3.7 Software:

Program B, listed in Appendix A, can be used to transfer a set of coefficients from any of the computer's input peripherals (magnetic tape, cassette, disc, paper tape) to the filter. The coefficients are assumed to be in the range (-1,127/128). A flowchart of this program is shown in fig. 3.17.

* * * * * * * *

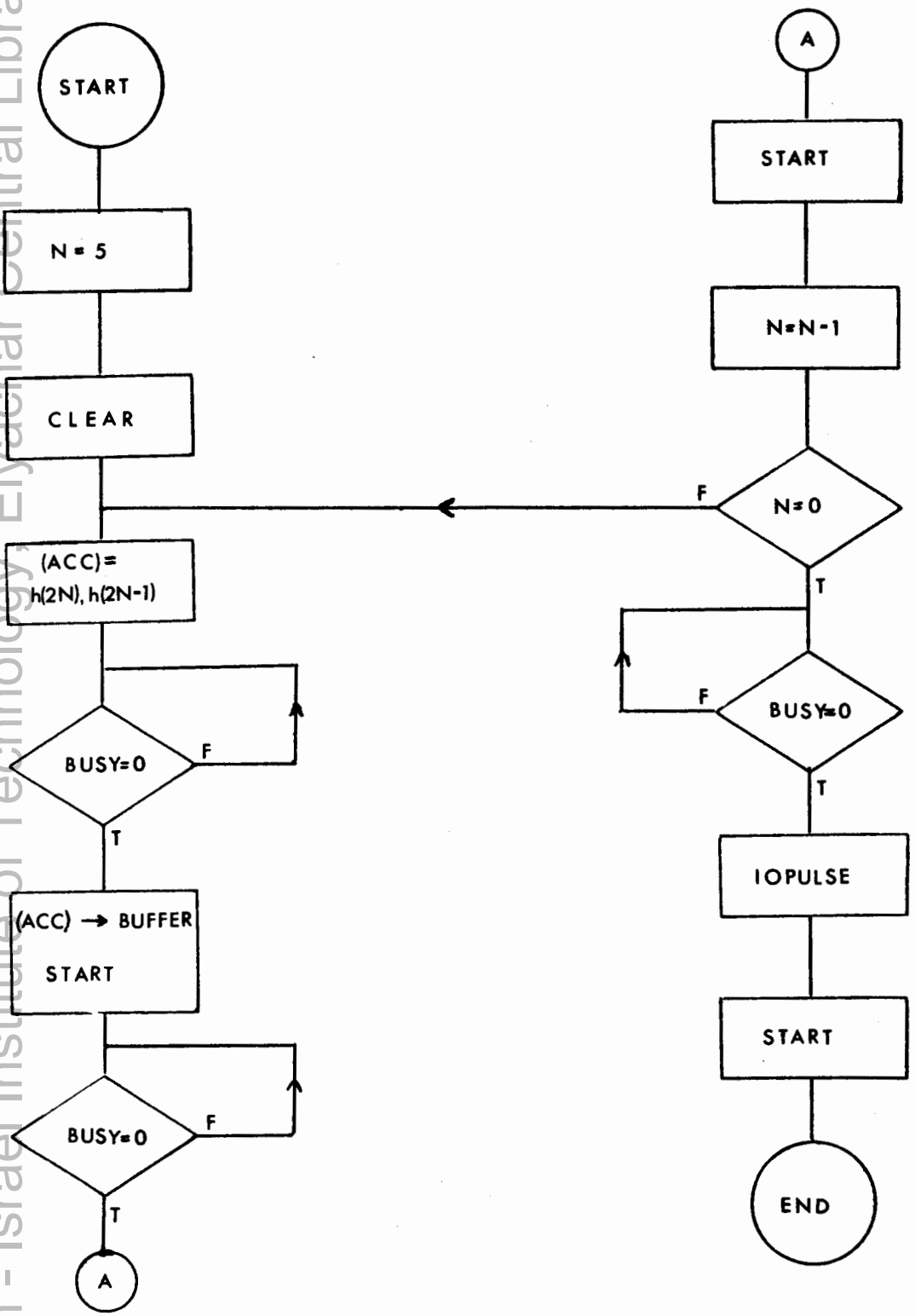


Fig. 3.17
Flowchart of Coefficient Loading Program (Program B)

ציור 3.17
דיאגרמת זרימה של
תכנית טעינת המקדמים (תכנית B)

CHAPTER 4

SPEECH SYNTHESIS BY LINEAR PREDICTION

4.1 Introduction:

The speech wave, generally, is a band limited signal in the range 30 to 4000Hz. The conventional method of storing speech digitally is by sampling it at at least 8kHz with a precision of 8 or more bits per sample. Thus one second of speech is represented by at least 64000 bits. This is a lot of storage and methods have been sought whereby the speech signal can be represented by a small number of slowly varying parameters. The linear prediction model is one of these representations. A system will be described in which the computer controlled filter is used as a time varying linear predictor. This model requires only 6400 bits per second of speech.

4.2 Linear Prediction Model:[1]

The following are characteristics of the vocal tract which indicate the use of the linear prediction model:

a) For nonnasal voiced speech the transfer function of the vocal tract has no zeros.

b) For unvoiced and nasal speech the transfer function of the vocal tract has both poles and zeros, but since the zeros all lie within the unit circle, each zero can be approximated by multiple poles within the unit circle.

Thus an all pole filter can approximate the transfer function of the vocal tract to any desired degree of accuracy. The linear prediction model of the speech wave is shown in fig. 4.1.

The output of the linear filter of fig. 4.1 is given by

$$x(n)=e(n)-\sum_{k=1}^{P}a(k)x(n-k) \qquad (4.1)$$

The transfer function of this filter is given by

$$H(z)=1/(1+\sum_{k=1}^{P}a(k)z^{-k})$$
$$=1/\prod_{k=1}^{P}(1-p(k)z^{-1}) \qquad (4.2)$$

where the p(k) are the complex poles of H(z).

The coefficients of the linear predictor are chosen for each segment of speech so that the prediction residual energy

$$\sum_{n=1}^{N}(x(n)+\sum_{k=1}^{P}a(k)x(n-k))^{2} \qquad (4.3)$$

is minimal. N is the number of speech samples in the segment.

Since the a(k) are real, the poles of H(z) are either real or occur in conjugate pairs. For the filter to be stable

$$|p(k)|<1 \quad k=1,2,\ldots,P \qquad (4.4)$$

The vocal tract is excited by one or both of the following:

a) A series of nearly periodic pulses generated by the vocal chords. This gives rise to voiced speech.

b) The turbulence of air forced through constrictions in the vocal tract. The speech is then unvoiced.

In the model, the two forms of excitation are mutually exclusive even though in actual speech combinations do occur (e.g. r,v,z,zh).

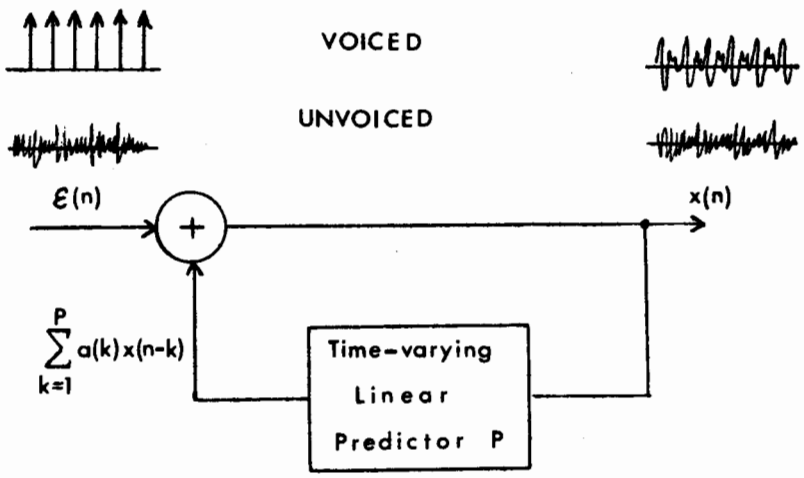


Fig. 4.1
Block Diagram of the Speech Model Based on the Linear Prediction of the Speech Wave

ציור 4.1
דיאגרמת מלבנים של מודל יצירת אות הדיבור המבוסס על חזוי ליניארי של גל הדבור

According to Atal and Hanauer,[1] the minimum number of poles necessary for a reasonable approximation of the transfer function of the vocal tract is 10-12. Thus the constructed computer controlled filter can be used. In the hardware realization of the model, the linear filter of fig. 4.1 is excited by band limited white noise for unvoiced speech or by a series of impulses spaced from one another by the pitch period for voiced speech.

Thus the parameters of the speech model are:

a) 10 filter coefficients

b) Voiced or unvoiced speech indicator

c) Pitch period (if voiced)

d) Amplitude of the excitation

4.3 Analysis of the Speech Signal:

4.3.1 Outline of Analysis:

The refresh rate for the synthesizer was chosen to be every 150 samples with a sampling rate of 10kHz. 150 samples corresponds to 1 to 4 pitch periods. In order to be able to extract pitch accurately, the analysis is performed over 450 samples, each analysis interval overlapping the previous interval by 150 samples (see fig. 4.2).

For the purpose of describing the analysis, let each 450 sample interval be represented as a sequence $x(n)$ $n=1,2,\ldots,450$. The predictor coefficients are computed using the whole interval but the prediction residual is computed only

for the segment n=151,152,...,300.

x(n) n=1,2,...,450 -- Analysis interval of sampled speech

W(n) n=1,2,...,450 -- Hamming window

y(n) n=1,2,...,450 -- Windowed analysis interval

a(k) k=1,2,...,10 --- Predictor coefficients

r(k) k=0,1,...,10 --- Autocorrelation coefficients of y(n)

e(n) n=151,152,...,300 - Prediction residual at sample n of the analysis interval

For each analysis interval of 450 samples, the following operations are performed:

a) The input signal x(n) is multiplied by a Hamming window

b) The filter coefficients a(k) are computed

c) The prediction residual e(n) is computed

d) The poles of the synthesizer are computed from the coefficients

e) The excitation amplitude is computed from the residual

f) A voiced/unvoiced decision is made. If the former, then the pitch period is extracted

g) The poles of the synthesizer are forced to lie within a prescribed circle of radius<1 to ensure stability

h) The stabilized filter coefficients are computed from the poles

i) The parameters are arranged in the format required by the synthesizer

The analysis is performed by two FORTRAN programs on the

NOVA 2. The first, Program C, executes stages a to f while the second, Program D, executes stages g to i. The programs are listed in Appendix A.

4.3.2 Window:

In order to reduce the effect of computing the predictor coefficients on a truncated signal, the 450 sample analysis interval is multiplied by a Hamming window

$$W(n)=.54+.46\cos(2\pi(n-225)/225) \quad n=1,2,\ldots,450 \qquad (4.5)$$

This window is shown in fig. 4.3.

4.3.3 Computation of the Predictor Coefficients:

The required coefficients are those which minimize the prediction residual energy over the analysis interval

$$\sum_{n=1}^{450}\left(y(n)+\sum_{k=1}^{10}a(k)y(n-k)\right)^2 \qquad (4.6)$$

By setting the partial derivatives of (4.6) with respect to the a(k) all to zero, a system of linear equations is obtained which reduces to a system of normal autocorrelation equations.

$$\begin{bmatrix} r(0) & r(1) & \ldots & r(8) & r(9) \\ r(1) & r(0) & \ldots & r(7) & r(8) \\ r(2) & r(1) & \ldots & r(6) & r(7) \\ \vdots & \vdots & & \vdots & \vdots \\ r(8) & r(7) & \ldots & r(0) & r(1) \\ r(9) & r(8) & \ldots & r(1) & r(0) \end{bmatrix} \begin{bmatrix} a(1) \\ a(2) \\ a(3) \\ \vdots \\ a(9) \\ a(10) \end{bmatrix} = \begin{bmatrix} r(1) \\ r(2) \\ r(3) \\ \vdots \\ r(9) \\ r(10) \end{bmatrix} \qquad (4.7)$$

where $r(k)=\sum_{n=1}^{450}y(n)y(n-k)$. The algorithm used to solve (4.7) is that given by Markel and Gray.[7]

4.3.4 Computation of the Prediction Residual:

For each 450 sample interval, the prediction residual of

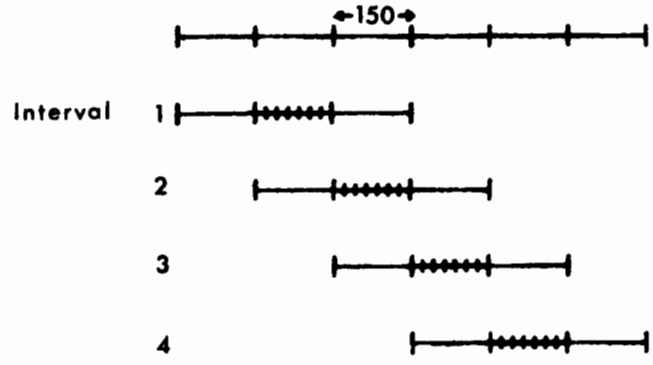


Fig. 4.2
Time Relation of Analysis Interval

ציור 4.2
יחסי זמן בין קטעי האבלידזה

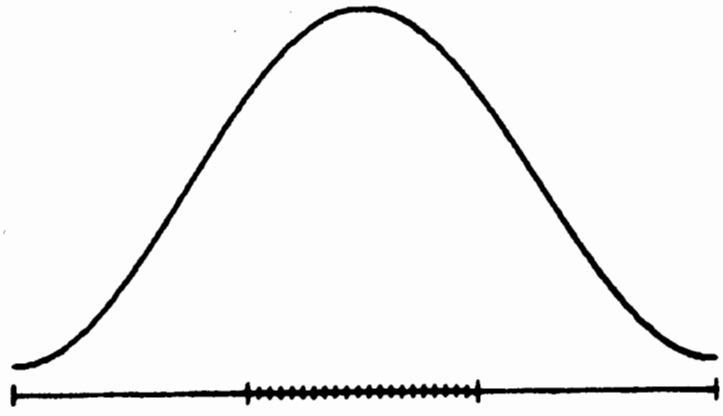


Fig. 4.3
Hamming Window

ציור 4.3
חלון Hamming

the center 150 samples are computed by

$$e(n)=x(n)+\sum_{k=1}^{10}a(k)x(n-k) \quad n=151,152,....,300 \tag{4.8}$$

4.3.5 Computation of the Excitation Amplitude:

The average excitation amplitude for the analysis interval is computed by

$$\text{Amplitude} = \sqrt{\sum_{n=151}^{300} e(n)^2} \tag{4.9}$$

4.3.6 Voiced / Unvoiced Decision:

Two characteristics of the speech signal are used to make this decision.

a)[1] The ratio

$$\frac{\sum_{n=1}^{450} e(n)^2}{\sum_{n=1}^{450} x(n)^2} \tag{4.10}$$

is about .04 for voiced speech and about .25 for unvoiced when using 10 predictor coefficients. This has been verified by observing the analysis of various speech records. During voiced intervals the ratio (4.10) varies between 0.02 and 0.15 while for unvoiced intervals (4.10) ranges from 0.1 to 0.5. A threshold of 0.2 is used in the analysis program.

b)[8] The prediction residual has strong almost periodic spikes corresponding to the pitch pulses. If the normalized autocorrelation of the prediction residual is computed, these spikes give rise to spikes at integral multiples of the pitch period. A peak picking procedure is used to find the largest spike and its value is compared to a fixed threshold, empirically found to be about 0.3. A spike greater than 0.3 indicates voiced speech. The location of the spike gives the

pitch period (see fig. 4.4).

These two criteria are used together. Only if both criteria indicate voiced speech is such a decision made. In order to reduce the occurence of a single wrong decision, a single "voiced" decision between two "unvoiced" decisions is changed to "unvoiced" and vice versa i.e. the sequence of decisions VVUVUUVUVV becomes VVVVUUUUVV.

4.3.7 Time Reversal:[13]

When the autocorrelation of a truncated signal is computed, a triangular window effect is produced (see fig. 4.5). In order to overcome this, a fixed overlap autocorrelation function is used for computing the autocorrelation of the prediction residual (see fig. 4.6). Either the first or the last 225 samples are correlated with the whole 450 sample interval. Which half is used depends on which half is more likely to have pitch spikes. This enables the onset and termination of a "voiced" segment to be more accurately detected. If the previous voiced/unvoiced decision is "voiced", then the earlier half is used while if the previous decision is "unvoiced", the later half is used, the latter option being effectively time reversal.

In the analysis program only that range of autocorrelation values corresponding to the range of possible pitch periods specified at the beginning of the analysis is computed. This is done to save run time since computation of the autocorrelation function is very time consuming and the search for pitch is

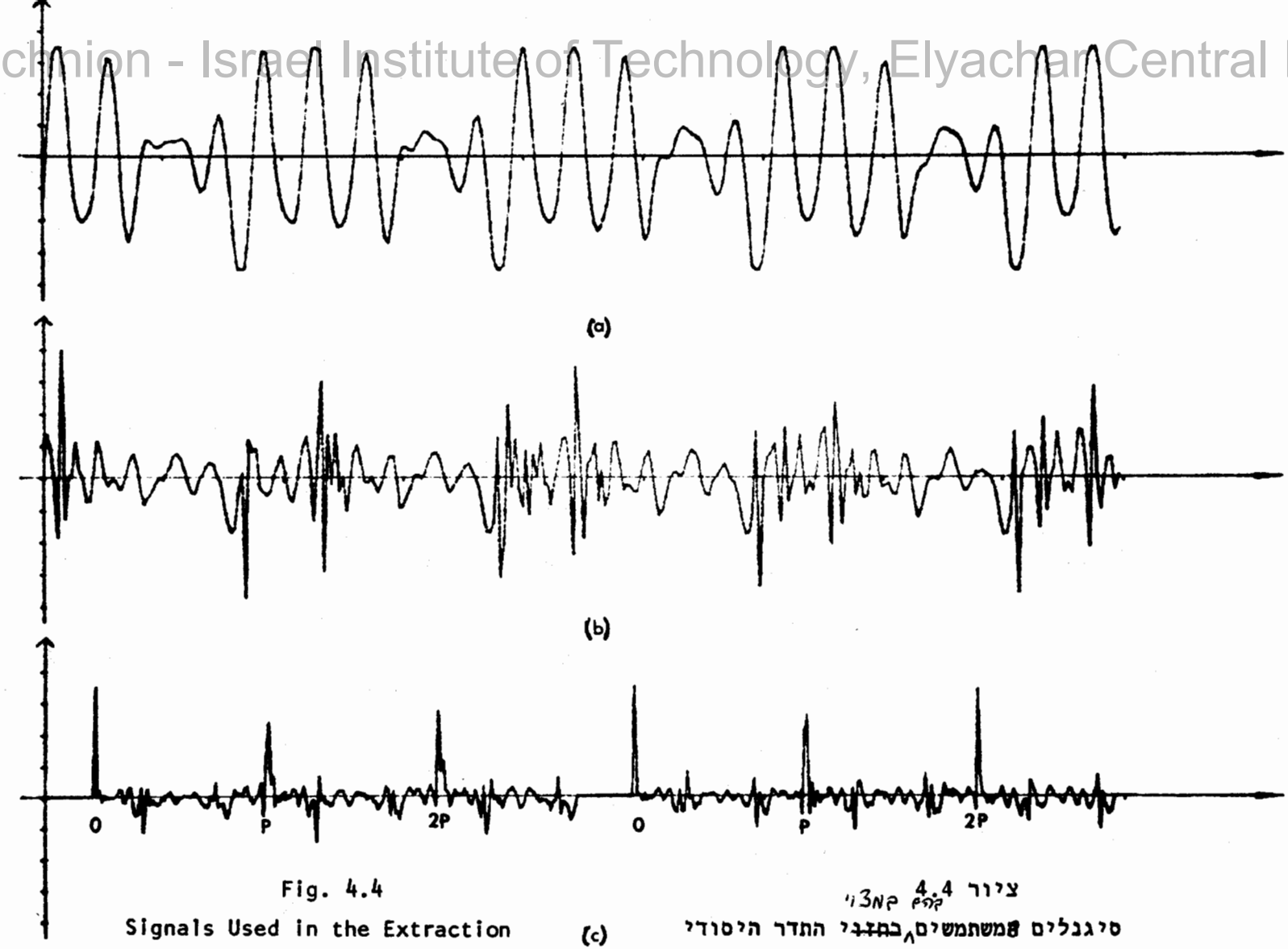


Fig. 4.4
Signals Used in the Extraction of Pitch
a) Original Speech Segment
b) Prediction Residual
c) Autocorrelation of Prediction Residual

ציור 4.4
סיגנלים שמשתמשים בחיזוי התדר היסודי
a) קטע דבור מקורי
b) שגיאת החזוי
c) פונקציית אוטוקורלציה של אות שגיאת החזוי.

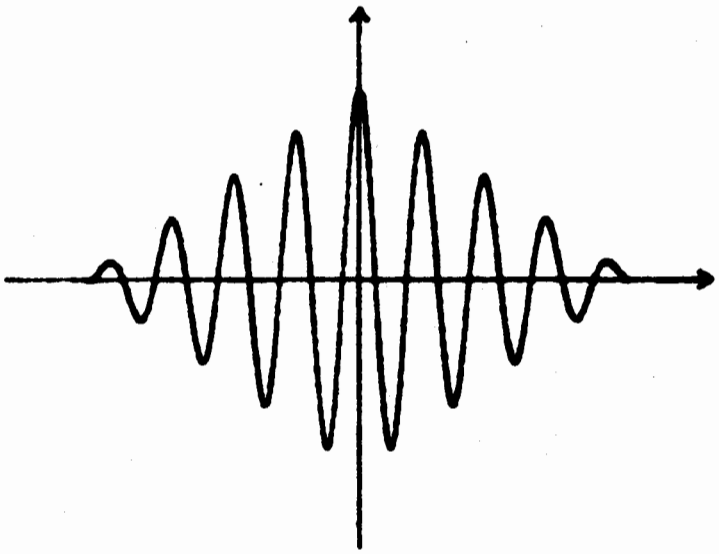


Fig. 4.5
Standard Autocorrelation of
a Truncated Sinusoid

ציור 4.5
פונקצית אוטוקורלציה
סטנדרטית של אות סינוסוידלי מוגבל זמן

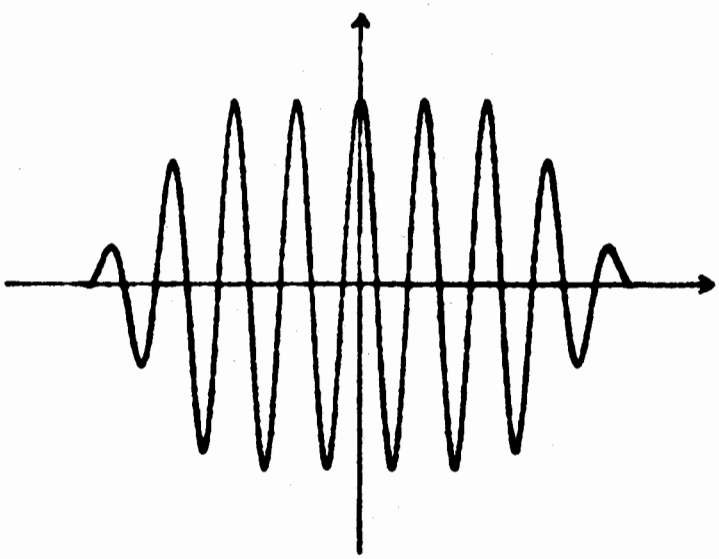


Fig. 4.6
Fixed Overlap Autocorrelation
of a Truncated Sinusoid

ציור 4.6
פונקצית אוטוקורלציה
עם חפיפה קבועה של אות סינוסוידלי
מוגבל זמן

conducted only in the predetermined range.

4.3.9 Computation of the Poles of the Transfer Function:

There are two reasons for wanting to compute the poles:

a) Roundoff errors in the solution of (4.7) might result in some of the poles lying outside the unit circle.

b) Because only 8 bit precision is used in the synthesizing filter, poles inside, but close to the unit circle cause significant limit cycles or even instability. Gold and Rader[3] show that in the case of a filter realized in a direct form (as is the case of the constructed filter), the error in the output due to truncation is

$$e < E/|1+\sum_{k=1}^{10} a(k)| \qquad (4.11)$$

where E is the quantization step size. To eliminate limit cycles it is necessary for this error to be smaller than E. Although (4.11) can be used to indicate the possibility of instability, it does not provide a remedy. The method used to eliminate instability was to force all poles to lie within a circle of radius<1. The radius of this circle was empirically found to be about 0.93.

The computation of the poles by the extraction of the roots of the polynomial $1+a(1)z^{-1}+\ldots+a(9)z^{-9}+a(10)z^{-10}$ is performed by an IBM developed subroutine using the Bairstow method of extracting quadratic factors from a polynomial. This method eliminates the need for complex arithmetic.

The poles are forced to lie within a prescribed circle in the following manner. Only poles lying outside the prescribed

circle are affected. If a pole lies outside, it is made to lie at the intersection of the prescribed circle and the straight line joining the original pole with the origin.

4.3.10 Formatting:

After the poles have been forced to lie within a circle of radius 0.93, the coefficients a(k) of the polynomial $1+\sum_{k=1}^{10} a(k)z^{-k}$ are recomputed. All the parameters for the analysis interval are then packed into 96 bits and stored in 6 computer words (see 4.4.2). The format is shown in fig. 4.7. First output to the loader are the exciter parameters. Because two 8 bit words have then already been output, the first coefficient output is not a(10) but a(8). This is followed by a(7) to a(1) and then a(10) and a(9). Note that two coefficients are stored in each computer word.

The program also provides for setting all parameters to zero if the excitation is too small i.e. during silence periods. This is to prevent unwanted limit cycles.

4.4 The Synthesizer:[5,8]

A block diagram of the synthesizer is shown in fig. 4.8. The original speech is sampled at 10kHz. Each set of parameters represents 15ms of real time. Thus the filter sampling rate must be 10kHz and the computer must update the parameters at 67Hz.

The synthesizer has two independent time bases. The master oscillator of the filter unit controls the pitch of the

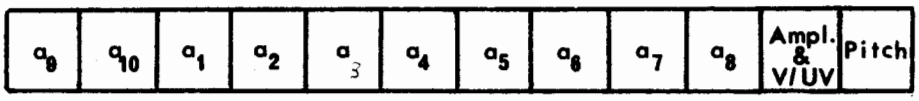


Fig. 4.7
Format of the Set of
Parameters for the Synthesizer

ציור 4.7
מבנה סט הפרמטרים
עבור הסינתסייזר

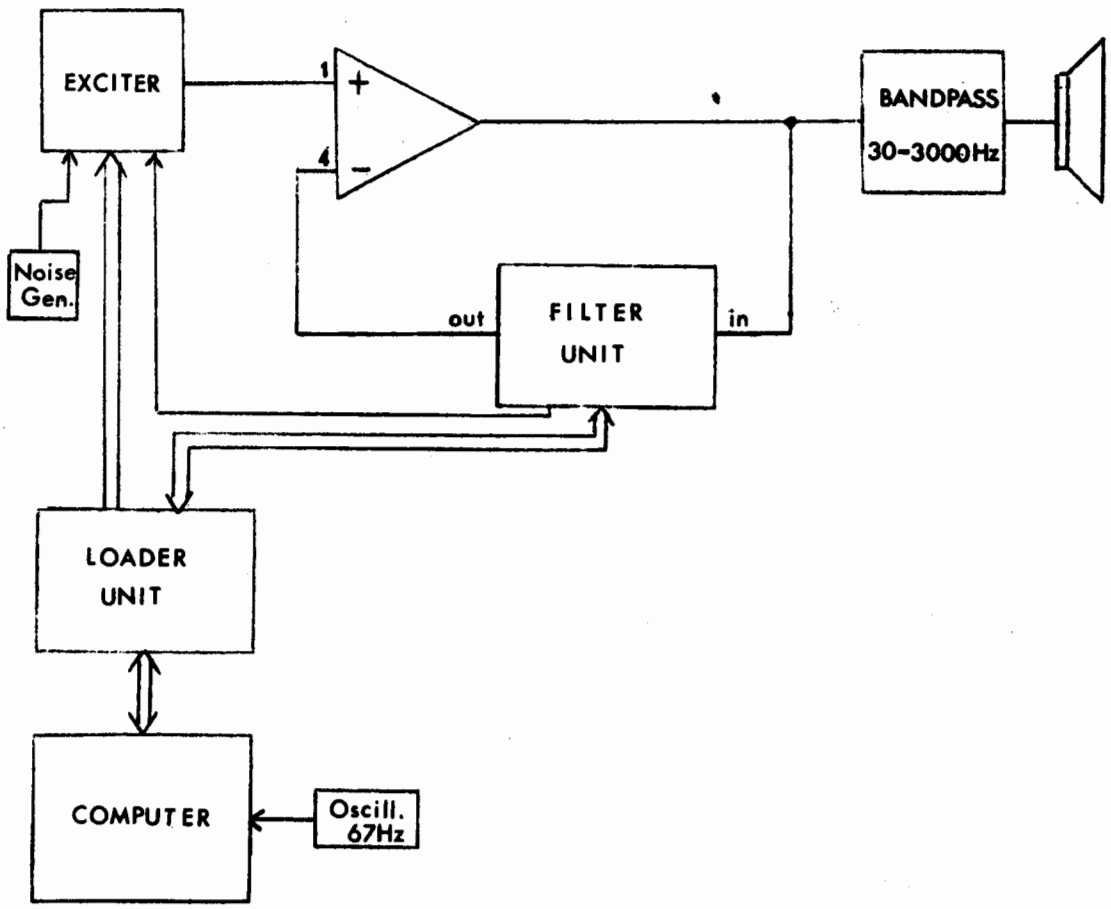


Fig. 4.8
Block Diagram of
Speech Synthesizer

ציור 4.8
דאגרמת מלבנים
של הסינתסייזר

synthesized speech while a second oscillator of frequency 67Hz is used by the computer to regulate the rate at which it updates the synthesizer parameters. Thus the pitch and/or the rate of the synthesized speech can be varied independently.

4.4.1 The Exciter:

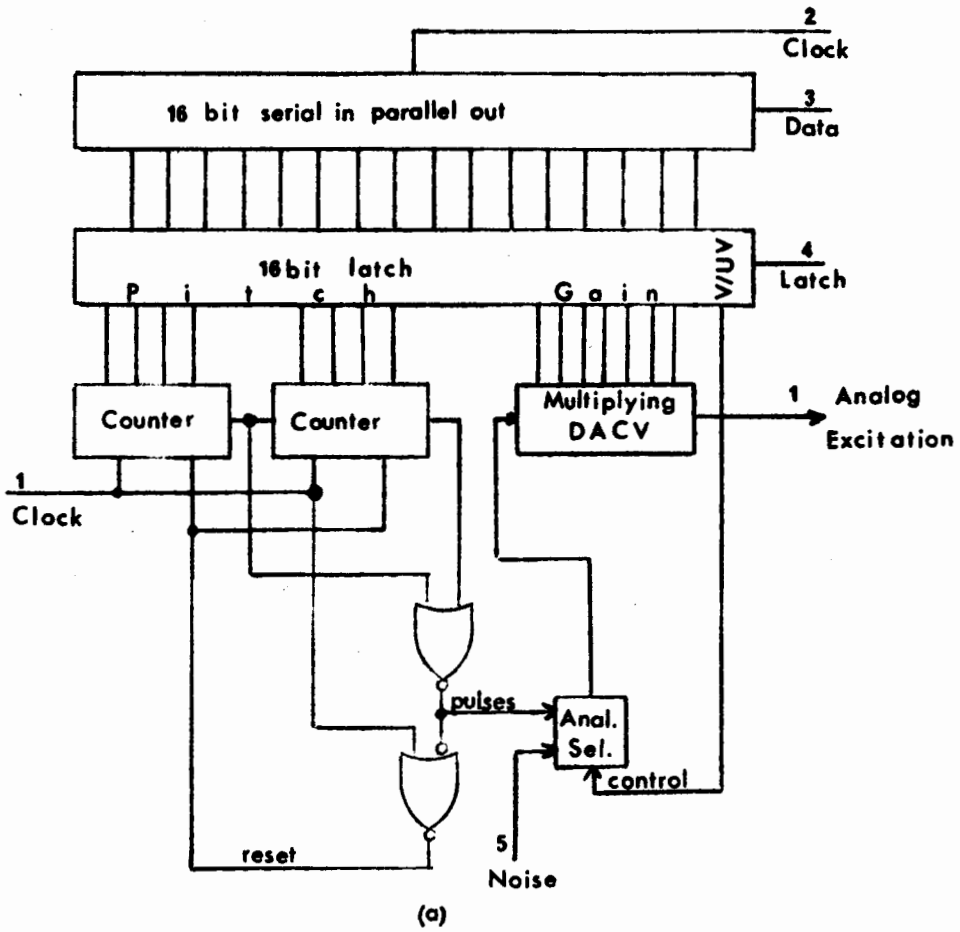
The function of the exciter is to provide either a pulse train of programmable period and amplitude or noise of programmable amplitude. The pulse train is produced by a resettable down counter which is reset to the value (pitch period) contained in a buffer every time it reaches zero. Data is latched after the transfer of a set of coefficients to the filter unit. An analog multiplexer controlled by a logic signal passes either the pulses or noise obtained from a pseudo-random binary noise generator. A CMOS multiplying digital to analog converter performs the amplitude modulation by multiplying the excitation by a positive digital number of 7 bits. The major portion of the exciter module was assembled from CMOS devices. The logic diagram of the exciter is shown in fig. 4.9.

Inputs:

1) Clock at filter sampling rate (Filter Control Module)
2) Loader shift register clock (Loader Storage Module)
3) Data (Loader Storage Module)
4) Latch control (Loader Load Control Module)
5) Noise (Pseudo-random Noise Generator)

Outputs:

1) Analog excitation (Filter Input)



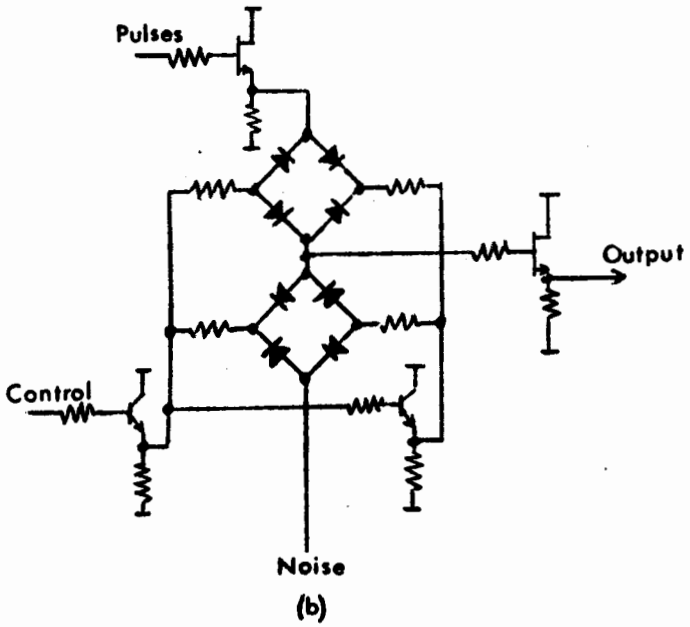


Fig. 4.9
Synthesizer Exciter
a) Logic Diagram
b) Analog Selector

ציור 4.9
יחידת עירור הסנתסייזר
a) דאגרמה לוגית
b) בורר אנלוגי

4.4.2 Parameter Coding:[14]

Although more efficient encoding methods for encoding the parameters exist, all are encoded linearly into binary format so that no decoding whatsoever is necessary in the synthesizer. The encoding is done as follows:

a) Filter coefficients:

Since the range of the coefficients was found to be (-4,+4), the loop gain of the synthesizer was made to be 4 so that the range of the actual coefficients could be (-1,+1), thus utilizing the full dynamic range of the filter unit. Each coefficient is encoded into the 8 bit 2's complement format according to the filter specifications.

b) Pitch:

The pitch of speech is generally in the range 50 to 200Hz. The corresponding pitch period is from 5 to 20ms i.e. 50 to 200 samples at a sampling rate of 10kHz. The pitch parameter is encoded into 8 bits giving a range of 2 to 255 samples per pitch period.

c) Excitation amplitude:

This parameter is linearly encoded into 7 bits.

d) Voiced/unvoiced decision:

This is a binary parameter and is encoded as the 8th (MSB) bit of the excitation amplitude.

Thus a complete set of parameters consists of 96 bits made up as follows:

10 coefficients	@ 8 bits	80 bits
Voiced/unvoiced	@ 1 bit	1 bit
Pitch	@ 8 bits	8 bits
Excitation ampl.	@ 7 bits	7 bits
Total		96 bits per frame

4.4.3 Control Program:

The control program, Program E, is written in FORTRAN together with the NOVA 2 assembly language. The rate at which the synthesizer is updated is controlled by a 67Hz clock. The program senses the clock pulses in the following manner. The 67Hz clock is connected to the computer analog to digital converter. For every clock pulse the converter performs a conversion at the end of which it sets its own DONE flip-flop. The control program senses the state of this DONE flip-flop and can thus synchronize its output to the synthesizer with the 67Hz oscillator.

The sequence of operations is as follows:

a) The first of the six parameter words (each 16 bits) is output to the buffer.

b) A START pulse is given and the replying COMP pulse awaited.

c) When the COMP pulse is received (i.e. BUSY=0), a second START pulse is given to transfer the second 8 bits.

d) When the COMP pulse is received the second of the six parameter words is output.

e) After all six parameter words have been transfered to the loader unit, the program 0waits for the 67Hz clock pulse by

sensing the state of the computer's analog to digital converter. When the pulse is received, an IOPULSE followed by a START pulse is sent to the loader unit for the transfer of the coefficients to the filter unit.

f) When the COMP pulse is received from the loader unit, the program begins the transfer of the next set of parameters to the loader unit.

The program is listed in Appendix A.

4.5 Example:

The utterance "Shalom al Yisrael" by a male speaker is analyzed. The printout of the analysis program can be found in Appendix B. After the analysis of each segment of 150 samples, the following is output:

a) Segment number

b) Energy of the signal in the center third of the analysis interval

c) The ratio (4.10)

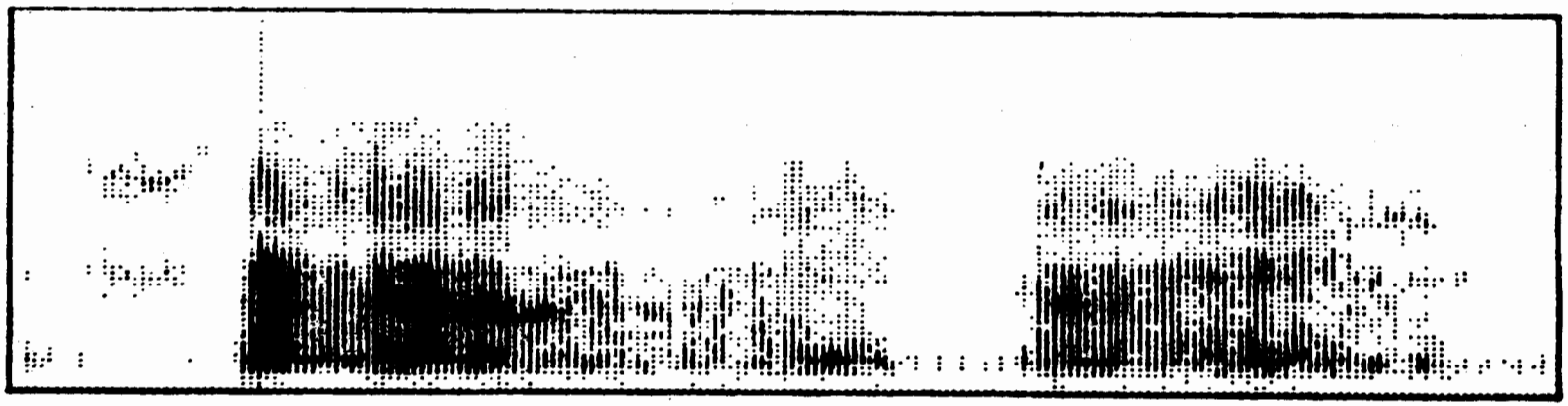
d) The peak normalized autocorrelation of the prediction residual

e) Pitch if decision is "voiced", otherwise zero.

Sonograms of the original utterance and of the synthesized utterance are shown in fig. 4.10. Plots of the poles of the predictor for an interval of unvoiced and of voiced speech are shown in figures 4.11 and 4.12 respectively.

(a)

SH - A - L - O - M A - L YI - S - R - A - E - L



(b)

Fig. 4.10

Sonograms

a) Original Speech

b) Synthesized Speech

ציור 4.10

סונוגרמות

a) אות הדבור המקורי

b) אות המסונתז

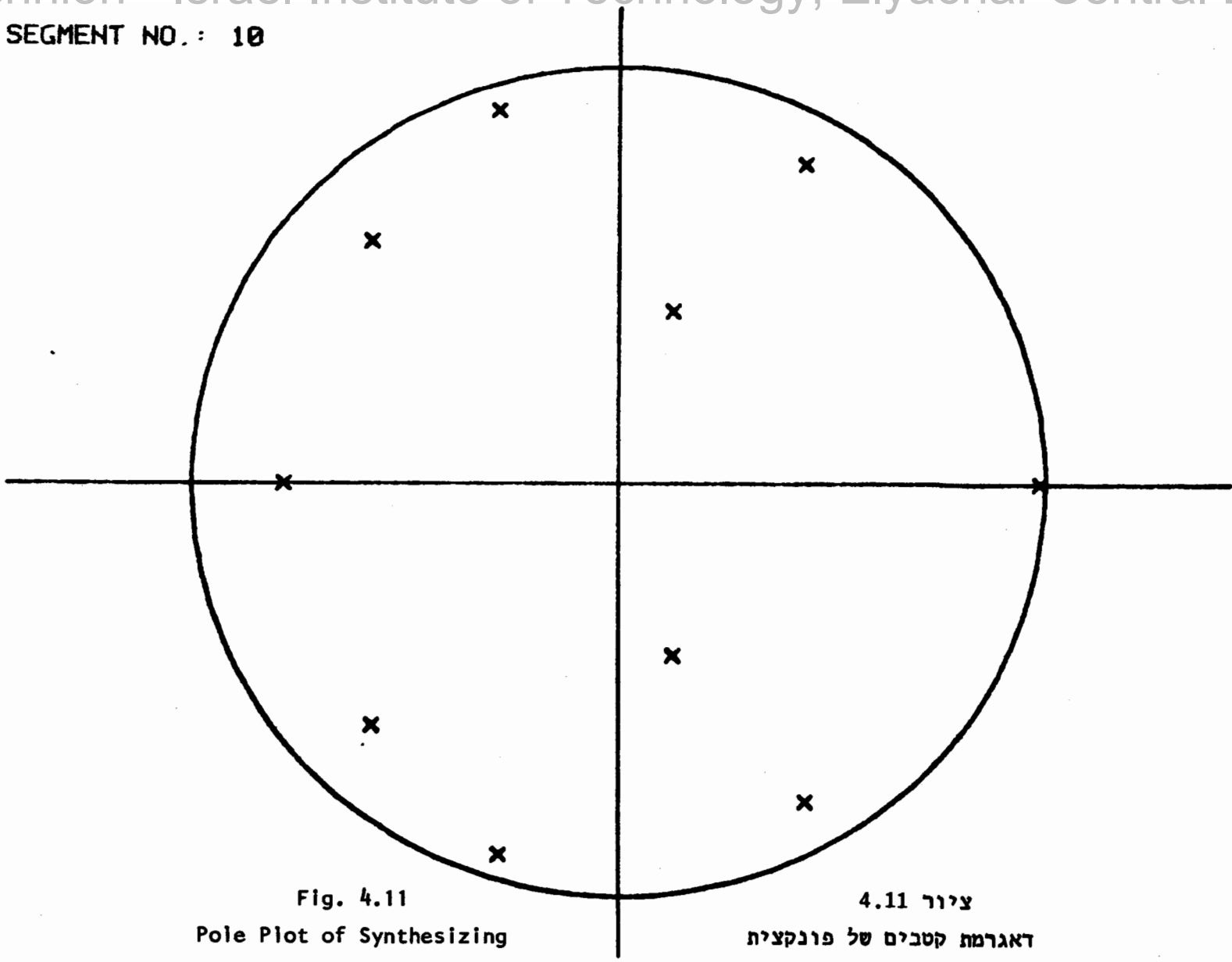


Fig. 4.11
Pole Plot of Synthesizing
Filter Transfer Function for a
Segment of Unvoiced Speech

ציור 4.11
דאגרמת קטבים של פונקציית
התמסורת של מסנן הסינתזה
עבור קטע של אות דבור אל - קולי

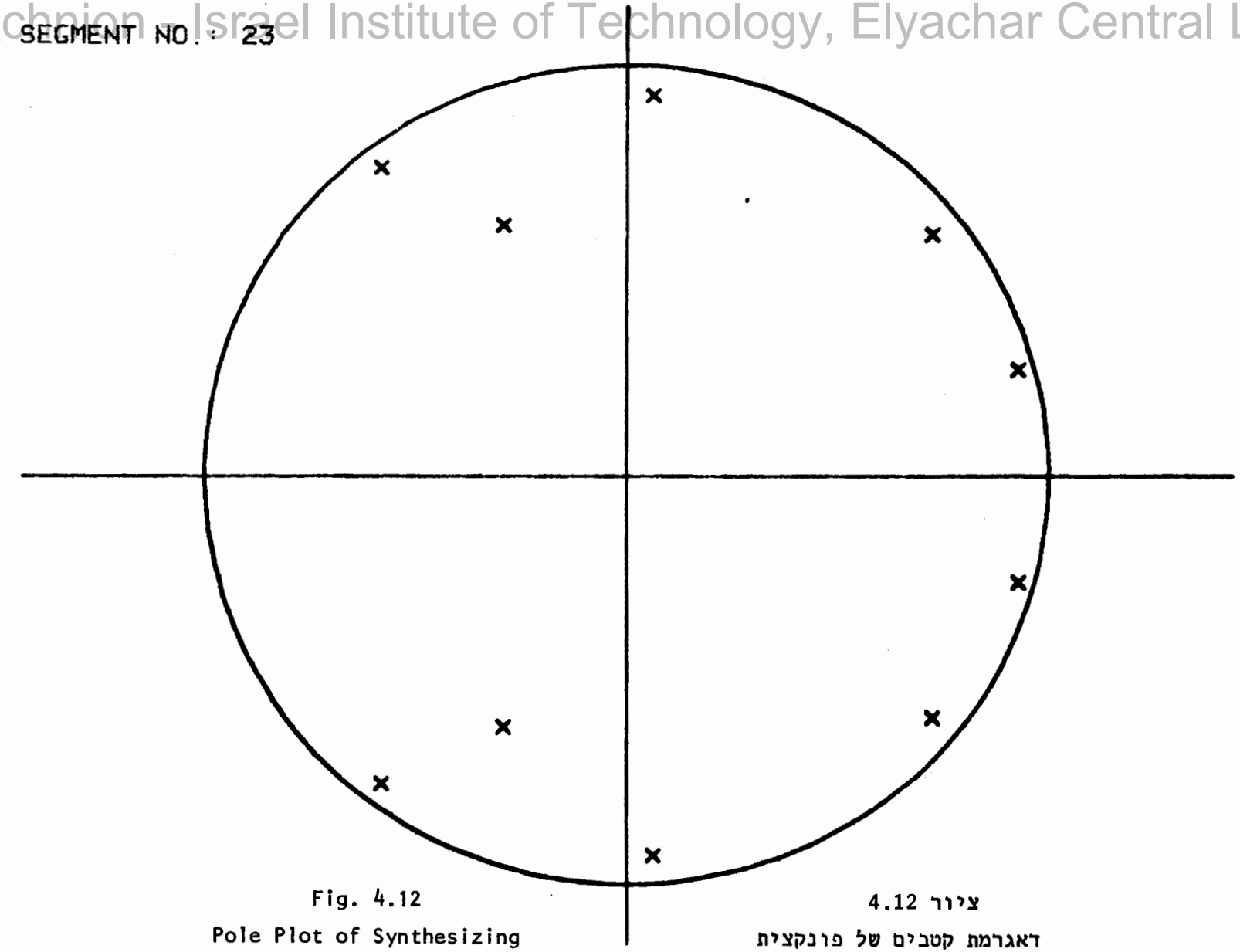


Fig. 4.12
Pole Plot of Synthesizing
Filter Transfer Function for a
Segment of voiced Speech

ציור 4.12
דאגרמת קטבים של פונקצית
התמסורת של מסנן הסינתזה
עבור קטע של אות דבור קולי

CHAPTER 5

CONCLUSIONS

The filter, as built, behaves as expected when used in the open loop mode i.e. with no feedback. With only 10 coefficients a sharp cutoff in the frequency response is impossible to attain. However, without much additional hardware but with a proportional decrease in the maximum sampling frequency, the number of coefficients can be increased. Since the coefficients are usually computed by the computer and the loading is done automatically, an increase in the number of coefficients does not present any operating problems. The additional hardware includes more storage, redesign of the counting circuits and, if necessary, an increase in the internal word length. Examples of the frequency responses of filters having 29 and 51 coefficients have already been shown in figures 2.12 to 2.17.

When the filter is used in the closed loop mode, problems do occur. In order to keep the design simple, the feedback loop was implemented using an operational amplifier. The conversion of the filter output to analog form and its reconversion to digital form introduces errors which are difficult to eliminate. A far better way, at the expense of more hardware, would be to implement the feedback loop digitally. Precise loop gains of 1,2,4 etc. are then easy to implement simply by shifting the fed back datum.

A further problem which only occurs in the feedback mode is that of internal precision. By increasing the length of the

data words to 12 bits, the amplitude of limit cycles would be reduced significantly and the stability of the filter for poles much closer to the unit circle can be assured.

The quality of the speech produced by the synthesizer is mediocre. The speech is understandable but lacks resonance. This is due to the deliberate damping of the closed loop impulse response by the relocation of poles close to the unit circle in order to prevent instability. The quality of the synthesized speech can be improved by following the recommendations mentioned above i.e. by increasing the data word length to 12 bits and by implementing the feedback loop digitally. The accuracy of the synthesized speech depends on the number of coefficients used. 10 coefficients is the minimum recommended in the literature.

When the filter is used in the speech synthesizer, the sample and hold circuit can be bypassed. This is because the input to the filter is a pulse, the level of which is constant during the analog to digital conversion. A further improvement can be made in order to utilize the full dynamic range of the coefficients. The coefficients can be normalized so that the largest coefficient (in absolute value) is in the range (0.5,1.0). The normalization factor would be contained in an additional parameter to be transfered together with the coefficients. This parameter needs only to be of two bits in order to give integer normalization factors of 1,2,4 and 8, which is sufficient for most purposes.

If the filter is to be used only for speech synthesis, the synthesizer can be made completely digital, only the synthesized output being converted to an analog signal. No analog to digital converter is required, since the excitation signal can be generated simply in a digital form and the feedback loop implemented digitally.

In the speech analysis example, the speech is sampled at 10kHz and a set of parameters extracted for each 15ms interval. It is probably possible to reduce these rates to 8kHz and 20ms respectively or even more without causing the quality of the synthesized speech to deteriorate too much, thus achieving an even greater saving in bits per second of speech.

* * * * * * * *

APPENDIX A

LISTING OF COMPUTER PROGRAMS

A.1 Program A:

```
*******   COMPUTATION OF TRANSVERSAL FILTER COEFFICIENTS    *******
**        FROM LINEAR PIECEWISE APPROXIMATION OF SPECTRUM        **

      DIMENSION Y(513),ZS(1028),ZT(1028)
      COMPLEX ZC(1028)
      COMMON DATAX(513)
      CALL FOPEN(0,"SCRATCH0")
      CALL FOPEN(1,"SCRATCH1")
      CALL FOPEN(2,"SCRATCH2")
      PI=4.*ATAN(1.)

***  LINEAR PIECEWISE APPROXIMATION OF SPECTRUM  ***

    1 ACCEPT "SPECTRUM FROM FILE? (1=YES,0=NO):  ",IX
      CALL TEST(IX,0,1,$1)
      IF(IX)10,10,2
    2 CALL INOUT(3,"FILE NAME:     ",0)
      READ BINARY(3,END=3) (Y(I),I=1,513)
      CALL FCLOS(3)
      GO TO 120
    3 TYPE "THE SPECIFIED FILE NAME WAS INCORRECT"
      TYPE
      CALL FCLOS(3)
      GO TO 1
   10 DO 20 I=1,513
   20 Y(I)=-1.
      TYPE
   30 ACCEPT "X,Y:  ",IX,YTEMP
      IF(IX)50,40,40
   40 IF(IX-512)60,60,50
   50 TYPE "ILLEGAL X VALUE. RE-INPUT:"
      GO TO 30
   60 Y(IX+1)=YTEMP
   70 ACCEPT "ANOTHER PAIR? (1=YES,0=NO):  ",IX
      CALL TEST(IX,0,1,$70)
      IF(IX)80,80,30
   80 IF(Y(1))90,100,100
   90 ACCEPT "Y(0):  ",Y(1)
      GO TO 80
```

```
100 IF(Y(513))110,120,120
110 ACCEPT "Y(512):  ",Y(513)
    GO TO 100
120 ACCEPT "PRINT BREAKPOINTS? (1=YES,0=NO):  ",IX
    CALL TEST(IX,0,1,$120)
    IF(IX)170,170,130
130 TYPE
    TYPE "        X             Y"
    TYPE
    DO 150 I=1,513
    IF(Y(I))150,140,140
140 J=I-1
    TYPE J,Y(I)
150 CONTINUE
    TYPE
160 ACCEPT "EDIT? (1=YES,0=NO):  ",IX
    CALL TEST(IX,0,1,$160)
    IF(IX)170,170,30
C
C  ***  PLOTTING OF SPECTRUM  ***
C
170 IFLAG3=0
180 ACCEPT "PLOT SPECTRUM? (1=YES,0=NO):  ",IX
    CALL TEST(IX,0,1,$180)
    IF(IX)280,280,190
190 YM=0.
    DO 210 I=1,513
    IF(Y(I)-YM)210,210,200
200 YM=Y(I)
210 CONTINUE
    IFLAG3=1
    CALL INITT(0)
    CALL MOVABS(450,600)
    CALL ANMODE
    TYPE "SPECTRUM"
    CALL AXES(246,800,100,90,550,256)
    CALL MOVABS(257,INT(350*Y(1)/YM)+100)
    IX=2
220 IF(Y(IX))230,240,240
230 IX=IX+1
    GO TO 220
240 CALL DRWABS(IX+256,INT(350*Y(IX)/YM)+100)
    IF(IX-513)250,260,260
250 IX=IX+1
    GO TO 220
260 CALL MOVABS(1000,1000)
    PAUSE
    CALL ERASE
    CALL ANMODE
270 ACCEPT "EDIT? (1=YES,0=NO):  ",IX
```

```
      CALL TEST(IX,0,1,$270)
      IF(IX)280,280,30
C
C  ***  INTERPOLATION OF SPECTRUM  ***
C
  280 IXOLD=1
      IX=2
  290 IF(Y(IX))300,310,310
  300 IX=IX+1
      GO TO 290
  310 DO 320 I=IXOLD,IX
  320 ZC(I)=CMPLX(Y(IXOLD)+(Y(IX)-Y(IXOLD))/
     *FLOAT(IX-IXOLD)*FLOAT(I-IXOLD),0.)
      IXOLD=IX
      IX=IX+1
      IF(IX-514)290,330,330
C
C  ***  FREQUENCY DOMAIN TO TIME DOMAIN  ***
C
  330 ACCEPT "ODD OR EVEN SPECTRUM? (-1=ODD,1=EVEN):  ",L2
      CALL TEST(L2,-1,1,$330)
      DO 340 I=514,1024
  340 ZC(I)=L2*ZC(1026-I)
      CALL DFT4(ZC,1024,1)
      IF(L2-1)370,350,370
  350 DO 360 I=1,513
  360 ZS(I)=REAL(ZC(I))
      GO TO 390
  370 DO 380 I=1,513
  380 ZS(I)=AIMAG(ZC(I))
C
C  ***  PLOTTING OF IMPULSE RESPONSE  ***
C
  390 IFLAG0=0
  400 ACCEPT "PLOT IMPULSE RESPONSE? (1=YES,0=NO):  ",IX
      CALL TEST(IX,0,1,$400)
      IF(IX)470,470,410
  410 ACCEPT "EXPANSION FACTOR (INTEGER):  ",L0
  420 ACCEPT "DISCRETE OR CONTINUOUS? (1=DISCR.,0=CONT.):  ",IT0
      CALL TEST(IT0,0,1,$420)
      YMAX=0.
      DO 440 I=1,513
      IF(ZS(I)-YMAX)440,440,430
  430 YMAX=ZS(I)
  440 CONTINUE
      IFLAG0=1
      CALL INITT(0)
      CALL MOVABS(350,760)
      CALL ANMODE
      TYPE "IMPULSE RESPONSE"
```

```
      DO 450 I=1,200
450   DATAX(I)=L2*ZS(202-I)/YMAX
      DO 460 I=1,201
460   DATAX(I+200)=ZS(I)/YMAX
      WRITE BINARY(0) (DATAX(I),I=1,401)
      REWIND 0
      CALL GRAPH(0,L0,250.,246,700,400,100,700,256,401,IT0,200)
      CALL MOVABS(1000,1000)
      PAUSE
      CALL ERASE
      CALL ANMODE

 ***  MULTIPLICATION OF IMPULSE RESPONSE BY WINDOW  ***

470   ACCEPT "WINDOW SIZE (ODD,1-1025):  ",L3
      IF(MOD(L3,2))470,470,480
      IF(L3-1)470,480,480
480   IF(L3-1025)490,490,470
490   ACCEPT "ALPHA? (1=RECTANGULAR,.54=HAMMING,.5=HANNING):  ",ALPHA
      L4=1+L3/2
      DO 500 I=1,L4
500   ZT(I)=ZS(I)*(ALPHA+(1-ALPHA)*COS((I-1)*PI/(L4-1)))
      IX=L4+1
      IF(IX-513)510,510,530
510   DO 520 I=IX,513
520   ZT(I)=0.
530   DO 540 I=514,1024
540   ZT(I)=L2*ZT(1026-I)

 *** PLOTTING OF WINDOWED IMPULSE RESPONSE  ***

      IFLAG1=0
550   ACCEPT "PLOT WINDOWED IMPULSE RESPONSE? (1=YES,0=NO):  ",IX
      CALL TEST(IX,0,1,$550)
      IF(IX)620,620,560
560   ACCEPT "EXPANSION FACTOR (INTEGER):  ",L1
570   ACCEPT "DISCRETE OR CONTINUOUS? (1=DISCR.,0=CONT.):  ",IT1
      CALL TEST(IT1,0,1,$570)
      YMAX=0.
      DO 590 I=1,513
      IF(ZT(I)-YMAX)590,590,580
580   YMAX=ZT(I)
590   CONTINUE
      IFLAG1=1
      CALL INITT(0)
      CALL MOVABS(320,760)
      CALL ANMODE
      TYPE "WINDOWED IMPULSE RESPONSE"
      CALL MOVABS(750,500)
      CALL ANMODE
```

```
      WRITE(10,1010) L3
      CALL MOVABS(750,300)
      CALL ANMODE
      WRITE(10,1020) ALPHA
      DO 600 I=1,200
600   DATAX(I)=ZT(824+I)/YMAX
      DO 610 I=1,201
610   DATAX(I+200)=ZT(I)/YMAX
      WRITE BINARY(1) (DATAX(I),I=1,401)
      REWIND 1
      CALL GRAPH(1,L1,250.,246,700,400,100,700,256,401,IT1,200)
      CALL MOVABS(1000,1000)
      PAUSE
      CALL ERASE

***  PLOTTING OF SPECTRUM OF WINDOWED IMPULSE RESPONSE  ***

      IFLAG2=0
620   ACCEPT "PLOT SPECTRUM OF WINDOWED IMPULSE RESPONSE?
     * (1=YES,0=NO):  ",IX
      CALL TEST(IX,0,1,$620)
      IF(IX)710,710,625
625   IF(L2-1)650,630,650
630   DO 640 I=1,1024
640   ZC(I)=CMPLX(ZT(I),0.)
      GO TO 670
650   DO 660 I=1,1024
660   ZC(I)=CMPLX(0.,ZT(I))
670   CALL DFT4(ZC,1024,0)
      YMAX=0.
      DO 690 I=1,513
      IF(CABS(ZC(I))-YMAX)690,690,680
680   YMAX=CABS(ZC(I))
690   CONTINUE
      DO 700 I=1,513
700   DATAX(I)=CABS(ZC(I))/YMAX
      WRITE BINARY(2) (DATAX(I),I=1,513)
      REWIND 2
      IFLAG2=1
      CALL INITT(0)
      CALL MOVABS(400,620)
      CALL ANMODE
      TYPE "SPECTRUM OF WINDOWED"
      CALL MOVABS(425,590)
      CALL ANMODE
      TYPE "IMPULSE RESPONSE"
      CALL GRAPH(2,1,350.,246,800,100,90,550,256,513,0,256)
      CALL MOVABS(1000,1000)
      PAUSE
      CALL ERASE
```

```
      CALL ANMODE
C
C  ***  OUTPUT OF COMPUTED DATA  ***
C
  710 ACCEPT "PRINT IMPULSE RESPONSE COEFFICIENTS? (1=YES,0=NO):  ",IX
      CALL TEST(IX,0,1,$710)
      IF(IX)770,770,720
  720 TYPE
      YMAX=0.
      DO 740 I=1,L4
      IF(ABS(ZT(I))-YMAX)740,740,730
  730 YMAX=ABS(ZT(I))
  740 CONTINUE
      L5=L4-1
      DO 750 I=1,L5
      L6=I-L4
      IX=1025+L6
      ZTEMP=ZT(IX)/YMAX
  750 TYPE L6,ZTEMP
      DO 760 I=1,L4
      L6=I-1
      ZTEMP=ZT(I)/YMAX
  760 TYPE L6,ZTEMP
      TYPE
  770 ACCEPT "OUTPUT COEFFICIENTS TO FILE? (1=YES,0=NO):  ",IX
      CALL TEST(IX,0,1,$770)
      IF(IX)830,830,780
  780 CALL INOUT(3,"FILE NAME:    ",1)
      YMAX=0.
      DO 800 I=1,L4
      IF(ABS(ZT(I))-YMAX)800,800,790
  790 YMAX=ABS(ZT(I))
  800 CONTINUE
      L5=L4-1
      DO 810 I=1,L5
      L6=I-L4
      IX=1025+L6
      ZTEMP=ZT(IX)/YMAX
  810 WRITE BINARY(3) ZTEMP
      DO 820 I=1,L4
      ZTEMP=ZT(I)/YMAX
  820 WRITE BINARY(3) ZTEMP
      CALL FCLOS(3)
  830 ACCEPT "REPEAT WINDOW? (1=YES,0=NO):  ",IX
      CALL TEST(IX,0,1,$830)
      IF(IX)840,840,470
C
C  ***  PLOTTING OF ALL GRAPHS  ***
C
  840 ACCEPT "PLOT ALL GRAPHS? (1=YES,0=NO):  ",IX
```

```
      CALL TEST(IX,0,1,$840)
      IF(IX)960,960,850
  850 CALL ERASE
      IF(IFLAG3)910,910,860
  860 CALL AXES(10,590,435,425,735,20)
      CALL MOVABS(21,INT(Y(1)*250./YM)+435)
      IX=2
  870 IF(Y(IX))880,890,890
  880 IX=IX+1
      GO TO 870
  890 CALL DRWABS(IX+20,INT(Y(IX)*250./YM)+435)
      IF(IX-513)900,910,910
  900 IX=IX+1
      GO TO 870
  910 IF(IFLAG0)925,925,920
  920 CALL GRAPH(0,L0,125.,660,1000,585,435,735,670,280,IT0,200)
  925 IF(IFLAG2)935,935,930
  930 CALL GRAPH(2,1,250.,10,590,45,35,345,20,513,0,256)
  935 IF(IFLAG1)950,950,940
  940 CALL GRAPH(1,L1,125.,660,1000,195,45,345,670,280,IT1,200)
  950 CALL MOVABS(1000,1000)
      PAUSE
      CALL ERASE
      CALL ANMODE
C
C *** SAVING THE IDEAL SPECTRUM  ***
C
  960 ACCEPT "SAVE IDEAL SPECTRUM? (1=YES,0=NO):  ",IX
      CALL TEST(IX,0,1,$960)
      IF(IX)980,980,970
  970 CALL INOUT(3,"FILE NAME:    ",1)
      WRITE BINARY(3) (Y(I),I=1,513)
      CALL FCLOS(3)
C
C ***  RESTART  ***
C
  980 ACCEPT "RESTART? (1=YES,0=NO):  ",IX
      CALL TEST(IX,0,1,$980)
      IF(IX)1000,1000,990
  990 ACCEPT "DELETE ALL BREAKPOINTS? (1=YES,0=NO):  ",IX
      CALL TEST(IX,0,1,$990)
      IF(IX)120,120,1
 1000 CALL RESET
      CALL DELETE("SCRATCH0")
      CALL DELETE("SCRATCH1")
      CALL DELETE("SCRATCH2")
      STOP
 1010 FORMAT(1X,"WINDOW WIDTH:",1X,I3)
 1020 FORMAT(1X,"ALPHA:",1X,F5.3)
      END
```

```
C
C *******    PLOT AXES     *******
C
      SUBROUTINE AXES(IXMIN,IXMAX,IYCON,IYMIN,IYMAX,IXCON)
      CALL MOVABS(IXCON,IYMAX)
      CALL DRWREL(-5,-7)
      CALL MOVREL(10,0)
      CALL DRWREL(-5,7)
      CALL DRWABS(IXCON,IYMIN)
      CALL MOVABS(IXMIN,IYCON)
      CALL DRWABS(IXMAX,IYCON)
      CALL DRWREL(-7,5)
      CALL MOVREL(0,-10)
      CALL DRWREL(7,5)
      RETURN
      END
*$
C
C *******    OPENING INPUT AND OUTPUT FILES     *******
C
      SUBROUTINE INOUT(CHANNEL,TEXT,K)
      INTEGER DEVICE(5),CHANNEL,TEXT(7)
      TYPE
   10 FORMAT(7A2)
   20 FORMAT(5A2)
      DO 30 I=1,5
   30 DEVICE(I)=0
      WRITE(10,10) TEXT
      READ(11,20,ERR=40) DEVICE
   40 IF(K.EQ.1) DELETE DEVICE
      CALL FOPEN(CHANNEL,DEVICE)
      RETURN
      END
*$
C
C *******    TEST FOR ACCEPTABLE OPTIONS     *******
C
      SUBROUTINE TEST(ITEST,IA,IB,IR)
      IF(ITEST-IA)10,30,10
   10 IF(ITEST-IB)20,30,20
   20 RETURN IR
   30 RETURN
      END
```

```
*******    PLOT GRAPH    *******

      SUBROUTINE GRAPH(IFILE,IEX,FACTOR,IXMIN,IXMAX,IYCON,
     *IYMIN,IYMAX,IXCON,IMAX,ITYPE,ICENT)
      COMMON DATAX(513)
      CALL AXES(IXMIN,IXMAX,IYCON,IYMIN,IYMAX,IXCON)
      L=IMAX/IEX
      IF(MOD(L,2).EQ.0) L=L-1
      IST=ICENT+(1-L)/2
      K=IST+L
      READ BINARY(IFILE) (DATAX(I),I=1,K)
      IF(ITYPE)30,10,30
10    CALL MOVABS(1+IXCON,INT(DATAX(1+IST)*FACTOR)+IYCON)
      DO 20 I=2,L
20    CALL DRWABS(IEX*I+IXCON,INT(DATAX(I+IST)*FACTOR)+IYCON)
      GO TO 50
30    DO 40 I=1,L
      CALL MOVABS(IEX*I+IXCON,IYCON)
40    CALL DRWREL(0,INT(DATAX(I+IST)*FACTOR))
50    REWIND IFILE
      RETURN
      END
```

A.2 Program B:

```
C
C  *******    LOAD A SET OF COEFFICIENTS    *******
C
      REAL COEFF(10)
      INTEGER OUT(5)
    1 DO 2 I=1,10
    2 COEFF(I)=0.
      CALL INOUT(0,"COEFF. FILE:  ",0)
      READ BINARY(0,END=3) (COEFF(I),I=1,10)
C
C  ***  PREPARE COEFFICIENTS IN PAIRS FOR TRANSFER  ***
C
    3 DO 4 I=1,5
      J1=INT(128*COEFF(2*I-1))
      J2=INT(128*COEFF(2*I))
    4 OUT(I)=400K*J1+IAND(377K,J2)
      TYPE
      DO 5 I=1,10
      J1=I-1
    5 WRITE(10,7) J1,COEFF(I)
C
C  ***  CLEAR LOADER  ***
C
A     NIOC 45
      DO 6 I=1,5
      L=OUT(6-I)
C
C  ***  LOAD ACCUMULATOR WITH DATUM (2 COEFFICIENTS)  ***
C
A     LDA 0,T.+12,3
C
C  ***  SENSE IF LOADER IS DONE  ***
C
A     SKPBZ 45
A     JMP .-1
C
C  ***  OUTPUT DATUM TO BUFFER AND SEND START PULSE  ***
C
A     DOAS 0,45
C
C  ***  SENSE IF LOADER IS DONE  ***
C
A     SKPBZ 45
A     JMP .-1
```

```
C
C  ***  SEND SECOND START PULSE  ***
C
A     NIOS 45
 6  CONTINUE
C
C  ***  SENSE IF LOADER IS DONE  ***
C
A     SKPBZ 45
A     JMP .-1
C
C  ***  SEND IOPULSE  ***
C
A     NIOP 45
C
C  ***  SEND START PULSE TO INITIATE LOAD SEQUENCE  ***
C
A     NIOS 45
      GO TO 1
 7  FORMAT(10X,I2,10X,F6.3)
      END
```

A.3 Program C:

```
*******    COMPUTATION OF SPEECH SYNTHESIZER PARAMETERS    *******

    THE SYNTHESIZER PARAMETERS ARE COMPUTED FOR EACH 150 SAMPLES
        OF SPEECH OVER AN ANALYSIS INTERVAL OF 450 SAMPLES.

    INPUT:  ORIGINAL SIGNAL
    X:      WINDOWED ANALYSIS INTERVAL
    WINDOW: HAMMING WINDOW
    ERROR:  PREDICTION RESIDUAL
    AUTOCOR:AUTOCORRELATION OF X
    COEFF:  PREDICTOR COEFFICIENTS
    ENG:    SIGNAL ENERGY
    ERG:    PREDICTION RESIDUAL ENERGY
    RR:     REAL PART OF THE POLES OF THE SYNTHESIZING FILTER
    RC:     IMAGINARY PART OF THE POLES OF THE FILTER
    POL:    DUMMY ARRAY FOR ROOT FINDING ROUTINE
    POLY:   INPUT POLYNOMIAL ARRAY FOR ROOT FINDING ROUTINE
    ECOR:   AUTOCORRELATION OF PREDICTION RESIDUAL
    GAIN:   EXCITATION AMPLITUDE
    DECISION:VOICED/UNVOICED DECISION
    PITCH:  PITCH IF VOICED, ZERO IF UNVOICED

 DOUBLE PRECISION INPUT(600),X(450),WINDOW(450),ERROR(450),
*AUTOCOR(11),COEFF(30),ENG(3),ERG(3),RR(11),RC(11),POL(11),
*POLY(11),ECOR(225),ALPHA,BETA,PARCOR,TEMP,SUM,MAX
 REAL RRS(10),RCS(10),GAIN(2),STORE(450)
 INTEGER DECISION(3),PITCH(2),WORD,W1,W2,FIX
 LOGICAL LAST1,LAST2
 DATA INPUT,ERROR,ENG,ERG,ECOR,COEFF/1486*0./
 DATA DECISION,PITCH,GAIN/7*0/,LAST1,LAST2/2*.FALSE./
 CALL INOUT(0,14HINPUT FILE:   ,0)
 CALL INOUT(1,14HGAIN & PITCH: ,1)
 CALL INOUT(2,14HROOTS FILE:   ,1)
 TYPE
 ACCEPT "OUTPUT EXTRA DATA? (1=YES,0=NO):  ",IX
 IF(IX.EQ.0) GO TO 1
 CALL INOUT(3,14HWINDOWED DATA:,1)
 CALL INOUT(4,14HERROR FILE:   ,1)
 CALL INOUT(5,14HAUTOCOR. FILE:,1)
1 TYPE
 ACCEPT "PITCH WINDOW:<15>",W1,W2
 IF(W1.LT.1) GO TO 1
 IF(W2.GT.225) GO TO 1
```

```
      IF(W1.GE.W2) GO TO 1
      TYPE
      TWOPI=8.*ATAN(1.)
      TYPE
      TYPE "    SEGMENT    ENERGY       ERROREN/ENERGY    MAX/ERROREN
     *      PITCH"
      TYPE

***   COMPUTE WINDOW   ***

      DO 2 I=1,450
    2 WINDOW(I)=(.54+.46*COS((FLOAT(I)/450.-.5)*TWOPI))

***   READ FIRST TWO SEGMENTS   ***

      NUMBER=1
      DO 3 I=301,600
      READ BINARY(0,END=38) FIX
    3 INPUT(I)=FLOAT(FIX)/2048.

***   APPLY WINDOW   ***

    4 DO 5 I=1,450
    5 X(I)=INPUT(I+150)*WINDOW(I)

***   COMPUTATION OF LPC'S   ***

      CALL AUTO(X,AUTOCOR,450,1,11)
      COEFF(21)=-AUTOCOR(2)/AUTOCOR(1)
      ALPHA=AUTOCOR(1)+COEFF(21)*AUTOCOR(2)
      BETA=AUTOCOR(3)+COEFF(21)*AUTOCOR(2)
      DO 7 I=1,9
      PARCOR=-BETA/ALPHA
      DO 6 J=1,(I+1)/2
      TEMP=COEFF(J+20)+PARCOR*COEFF(I-J+21)
      COEFF(I-J+21)=COEFF(I-J+21)+PARCOR*COEFF(J+20)
    6 COEFF(J+20)=TEMP
      COEFF(I+21)=PARCOR
      ALPHA=ALPHA+PARCOR*BETA
      BETA=0.
      IF(I.LE.8) BETA=AUTOCOR(I+3)
      DO 7 J=1,I+1
      BETA=BETA+COEFF(J+20)*AUTOCOR(I-J+3)
    7 CONTINUE

***   COMPUTATION OF ERROR SIGNAL   ***

      DO 9 I=1,150
      SUM=0.
      DO 8 J=1,10
```

```
    SUM=SUM+COEFF(J+20)*INPUT(I-J+300)
    ERROR(I+300)=INPUT(I+300)+SUM

***  DETERMINATION OF PITCH AND GAIN  ***

    PITCH(2)=0
    DECISION(3)=0
    ENERGY=0.
    ERROREN=0.
    ENG(3)=0.
    ERG(3)=0.
    MAX=0.
    DO 10 I=301,450
    ENG(3)=ENG(3)+INPUT(I)*INPUT(I)
10  ERG(3)=ERG(3)+ERROR(I)*ERROR(I)
    ENERGY=ENG(1)+ENG(2)+ENG(3)
    ERROREN=ERG(1)+ERG(2)+ERG(3)
    GAIN(2)=SQRT(ERG(3))
    IF(ERROREN/ENERGY.GT..2) GO TO 22
    IF(DECISION(2))15,15,11
11  DO 13 I=W1,W2
    SUM=0.
    DO 12 J=1,225
12  SUM=SUM+ERROR(J)*ERROR(J+I)
13  ECOR(I+1)=SUM
    SUM=0.
    DO 14 I=1,225
14  SUM=SUM+ERROR(I)*ERROR(I)
    ECOR(1)=SUM
    GO TO 19
15  DO 17 I=W1,W2
    SUM=0.
    DO 16 J=226,450
16  SUM=SUM+ERROR(J)*ERROR(J-I)
17  ECOR(I+1)=SUM
    SUM=0.
    DO 18 I=226,450
18  SUM=SUM+ERROR(I)*ERROR(I)
    ECOR(1)=SUM
19  DO 21 I=W1,W2
    IF(MAX-ECOR(I))20,21,21
20  MAX=ECOR(I)
    INDEX=I
21  CONTINUE
    IF(MAX/ECOR(1).GE..3) DECISION(3)=1
    PITCH(2)=INDEX-1
22  IF(DECISION(1).NE.DECISION(3)) GO TO 23
    DECISION(2)=DECISION(1)
23  PITCH(1)=DECISION(2)*PITCH(1)
    WRITE (10,24) NUMBER,ENERGY,ERROREN/ENERGY,MAX/ECOR(1),PITCH(1)
```

```
24 FORMAT(5X,I3,3X,G13.6,1X,G13.6,5X,G13.6,5X,I3)
C
C  ***  COMPUTATION OF POLES OF TRANSFER FUNCTION  ***
C
   POLY(11)=1.
   DO 25 I=1,10
25 POLY(11-I)=COEFF(I)
   CALL PRBM(POLY,11,RR,RC,POL,IR,IER)
   DO 26 I=1,10
   RRS(I)=SNGL(RR(I))
26 RCS(I)=SNGL(RC(I))
   IF(IR.NE.10) TYPE "IER:",IER,"   NO. OF COMPUTED ROOTS:",IR
C
C  ***  OUTPUT  ***
C
   IF(IX.EQ.0) GO TO 30
   DO 27 I=1,450
27 STORE(I)=SNGL(X(I))
   WRITE BINARY(3) (STORE(I),I=1,450)
   DO 28 I=1,150
28 STORE(I)=SNGL(ERROR(I+300))
   WRITE BINARY(4) (STORE(I),I=1,150)
   DO 29 I=1,150
29 STORE(I)=SNGL(ECOR(I)/ECOR(1))
   WRITE BINARY(5) (STORE(I),I=1,150)
30 WRITE BINARY(1) GAIN(1),PITCH(1)
   IF(IR)32,32,31
31 WRITE BINARY(2) (RRS(I),RCS(I),I=1,IR)
   IF(IR-10)32,33,33
32 WRITE BINARY(2) (0.E0,0.E0,I=IR+1,10)
C
C  ***  SHIFT DATA  ***
C
33 DO 34 I=1,450
34 INPUT(I)=INPUT(I+150)
   DO 35 I=1,300
35 ERROR(I)=ERROR(I+150)
   DO 36 I=1,20
36 COEFF(I)=COEFF(I+10)
   GAIN(1)=GAIN(2)
   PITCH(1)=PITCH(2)
   ENG(1)=ENG(2)
   ENG(2)=ENG(3)
   ERG(1)=ERG(2)
   ERG(2)=ERG(3)
   DECISION(1)=DECISION(2)
   DECISION(2)=DECISION(3)
C
C  ***  INPUT FRESH SEGMENT  ***
```

```
      IF(LAST1.AND.LAST2) GO TO 40
      DO 37 I=451,600
      READ BINARY(0,END=38) FIX
   37 INPUT(I)=FLOAT(FIX)/2048.
      NUMBER=NUMBER+1
      GO TO 4
   38 DO 39 J=I,600
   39 INPUT(I)=0.
      IF(LAST1) LAST2=.TRUE.
      LAST1=.TRUE.
      NUMBER=NUMBER+1
      GO TO 4
   40 TYPE "<7>"
      END
*$
C
C  ***  SUBROUTINE FOR COMPUTING AUTOCORRELATION  ***
C
      SUBROUTINE AUTO(X,AUTOCOR,NUM1,NUM2,NUM3)
      COMPILER DOUBLE PRECISION
      REAL X(1),AUTOCOR(1)
      DO 20 I=NUM2,NUM3
      SUM=0.
      DO 10 J=1,NUM1-I+1
   10 SUM=SUM+X(J)*X(I+J-1)
   20 AUTOCOR(I)=SUM
      RETURN
      END
*$
C
C  *******    OPENING INPUT AND OUTPUT FILES    *******
C
      SUBROUTINE INOUT(CHANNEL,TEXT,K)
      INTEGER DEVICE(5),CHANNEL,TEXT(7)
      TYPE
   10 FORMAT(7A2)
   20 FORMAT(5A2)
      DO 30 I=1,5
   30 DEVICE(I)=0
      WRITE(10,10) TEXT
      READ(11,20,ERR=40) DEVICE
   40 IF(K.EQ.1) DELETE DEVICE
      CALL FOPEN(CHANNEL,DEVICE)
      RETURN
      END
```

A.4 Program D:

```
C
C  *******    FORMATTING OF PARAMETERS    *******
C
      REAL COEFF(10)
      COMPLEX ROOT(10),POLY(11)
      INTEGER OUT1(5),OUT2,PITCH
      CALL INOUT(0,14HROOTS FILE:   ,0)
      CALL INOUT(1,14HGAIN & PITCH: ,0)
      CALL INOUT(2,14HOUTPUT FILE:  ,1)
      ACCEPT "<15>REQUIRED CIRCLE:<15>",CIRCLE
      ACCEPT "<15>COEFFICIENT FACTOR:<15>",FACTORC
      ACCEPT "<15>NOISE FACTOR:<15>",FACTORN
      FACTORN=2.*FACTORN
      ACCEPT "<15>LOWER GAIN BOUND:<15>",LB
      TEMP=0.
C
C  ***  SELECT SUITABLE GAIN  ***
C
    1 READ BINARY(1,END=2) GAIN,PITCH
      IF(PITCH.EQ.0) GAIN=GAIN*FACTORN
      IF(TEMP.GE.GAIN) GO TO 1
      TEMP=GAIN
      GO TO 1
    2 FACTORS=127./TEMP
      TYPE
      TYPE "MAXIMUM GAIN FACTOR IS:  ",FACTORS
      REWIND 1
    3 ACCEPT "<15>REQUIRED GAIN FACTOR:<15>",FACTORS
      NZERO=0
    4 READ BINARY(0,END=17) (ROOT(I),I=1,10)
C
C  ***  RELOCATE PROBLEMATIC POLES  ***
C
      DO 5 I=1,10
      IF(CABS(ROOT(I)).GT.CIRCLE) ROOT(I)=ROOT(I)*CIRCLE/CABS(ROOT(I))
    5 CONTINUE
      READ BINARY(1,END=17) GAIN,PITCH
      GAIN=GAIN*FACTORS
      IF(PITCH)6,6,7
    6 GAIN=GAIN*FACTORN
C
C  ***  COMPUTE COEFFICENTS FROM POLES  ***
C
    7 POLY(1)=ROOT(1)
```

```
      POLY(2)=(-1.,0.)
      DO 9 I=1,9
      POLY(I+2)=-POLY(I+1)
      DO 8 J=1,I
    8 POLY(I-J+2)=POLY(I-J+2)*ROOT(I+1)-POLY(I-J+1)
    9 POLY(1)=POLY(1)*ROOT(I+1)
      DO 10 I=1,10
   10 COEFF(I)=REAL(POLY(11-I))
      DO 11 I=1,10
   11 COEFF(I)=COEFF(I)*FACTORC
C
C  ***  FORMAT PAIRS OF COEFFICENTS FOR OUTPUT  ***
C
      DO 12 I=1,5
      J1=INT(32*COEFF(2*I-1))
      J2=INT(32*COEFF(2*I))
   12 OUT1(I)=400K*J1+IAND(377K,J2)
      IGAIN=INT(GAIN)
      IF(IGAIN-LB)16,13,13
   13 IF(IGAIN.GT.127) IGAIN=127
      OUT2=255-PITCH+400K*IGAIN
      IF(PITCH)15,15,14
   14 CALL ISET(OUT2,15)
   15 WRITE BINARY(2) OUT2,OUT1(4),OUT1(3),OUT1(2),OUT1(1),OUT1(5)
      TYPE IGAIN,PITCH
      GO TO 5
   16 WRITE BINARY(2) (NZERO,I=1,6)
      TYPE NZERO,NZERO
      GO TO 5
   17 TYPE "<7>"
      END
```

A.5 Program E:

```
C
C *******    CONTROLLED OUTPUT OF PARAMETERS TO SYNTHESIZER    *******
C
      INTEGER INPUT(15000)
      J=1
    1 CALL INOUT(0,"FILE NAME:     ",0)
      DO 2 I=J,15000
    2 READ BINARY(0,END=3) INPUT(I)
      GO TO 4
    3 CALL FCLOS(0)
      J=I
      ACCEPT "<15>ANOTHER FILE? (YES=1,NO=0):  ",K
      IF(K.EQ.1) GO TO 1
    4 CONTINUE
      K=I
      ACCEPT "<15>WITH OR WITHOUT PAUSE? (YES=1,NO=0):  ",M
A     NIOC 45
    5 DO 7 J=1,K,6
      DO 6 I=1,6
      L=INPUT(J+I-1)
A     LDA 0,T.+7,3
A     SKPBZ 45
A     JMP .-1
A     DOAS 0,45
A     SKPBZ 45
A     JMP .-1
A     NIOS 45
    6 CONTINUE
C
C ***  WAIT FOR A-D CONVERTER  ***
C
A     SKPBZ ADCV
A     JMP .-1
A     NIOS ADCV
C
C ***  67HZ CLOCK PULSE DETECTED  ***
C
A     SKPBZ 45
A     JMP .-1
A     NIOP 45
A     NIOS 45
    7 CONTINUE
      IF(M.EQ.1) PAUSE
      GO TO 5
      END
```

APPENDIX B

SAMPLE RUN OF ANALYSIS PROGRAM

```
R
LPC

INPUT FILE:
SHALOM

GAIN & PITCH:
GAIN

ROOTS FILE:
ROOT

OUTPUT EXTRA DATA? (1=YES,0=NO):  0
PITCH WINDOW:
40,90
```

SEGMENT	ENERGY	ERROREN/ENERGY	MAX/ERROREN	PITCH
1	.286462E-01	.263558E-01	.110563	0
2	.665040E-01	.224159E-01	.117418	0
3	.103657	.525357E-01	.122505	0
4	.114766	.146966	.112612	0
5	.148122	.262772	.000000	0
6	.203807	.330429	.000000	0
7	.277052	.358408	.000000	0
8	.310498	.385065	.000000	0
9	.572761	.244548	.000000	0
10	.586217	.258506	.000000	0
11	.599093	.272658	.000000	0
12	.278276	.447952	.000000	0
13	.545676	.177172	.179227	0
14	3.84592	.105306	.364520	0
15	8.52753	.987057E-01	.779045	66
16	9.74629	.969780E-01	.317665	66
17	7.74741	.843164E-01	.518184	69
18	4.60501	.569314E-01	.184526	71
19	4.58790	.486601E-01	.709729	75
20	4.82424	.522711E-01	.744252	77
21	7.95482	.768540E-01	.455820	77
22	15.1278	.746619E-01	.595380	76
23	22.0315	.652559E-01	.613178	72
24	23.2143	.593355E-01	.513730	71
25	18.9415	.507112E-01	.498755	70
26	14.8085	.629183E-01	.447772	70
27	12.6279	.633767E-01	.488327	67
28	10.4680	.678480E-01	.449598	63
29	6.59403	.529980E-01	.325295	62

APPENDIX C

DATA SHEETS

ANALOG DEVICES

CMOS 10-Bit Monolithic A/D Converter

AD7570

FEATURES

- 8 and 10-Bit Resolution
- 20μsec Conversion Time
- Microprocessor Compatibility
- Very Low Power Dissipation
- Parallel and Serial Outputs
- Ratiometric Operation
- TTL/DTL/CMOS Logic Compatibility
- CMOS Monolithic Construction

GENERAL DESCRIPTION

The AD7570 is a monolithic CMOS 10-bit successive approximation A/D converter on a 120 by 135 mil chip, requiring only an external comparator, reference and passive clocking components. Ratiometric operation is inherent, since an extremely accurate multiplying DAC is used in the feedback loop.

The AD7570 parallel output data lines and Busy line utilize three-state logic to permit bussing with other A/D output and control lines or with other I/O interface circuitry. Two enables are available, one controls the two MSB's, the second controls the remaining 8 LSB's. This feature provides the control interface for most microprocessors which can accept only an 8-bit byte.

The AD7570 also provides a serial data output line to be used in conjunction with the serial synchronization line. The clock can be driven externally or, with the addition of a resistor and a capacitor, can run internally as high as 0.6 MHz allowing a total conversion time (8-bits) of typically 20μsec. An 8-bit short cycle control pin stops the clock after exercising 8 bits, normally used for the "J" version (8-bit resolution).

The AD7570 requires two power supplies, a +15V main supply and a +5V (for TTL/DTL logic) to +15V (for CMOS logic) supply for digital circuitry. Both analog and digital grounds are available.

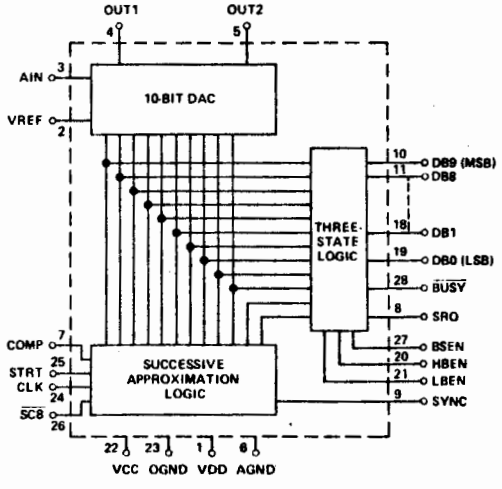
The AD7570 is a monolithic device using a proprietary CMOS process featuring a double layer metal interconnect, on-chip thin-film resistor network and silicon nitride passivation ensuring high reliability and excellent long term stability.

ORDERING INFORMATION

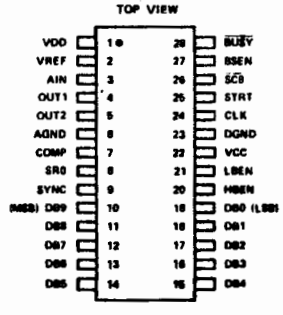
Resolution	Temperature Range
	0 to +75°C
8-Bit	AD7570J
10-Bit	AD7570L

Suffix D: Ceramic Package

FUNCTIONAL DIAGRAM



PIN CONFIGURATION



Route 1 Industrial Park; P.O. Box 280; Norwood, Mass. 02062
Tel: 617/329-4700 TWX: 710/394-6577
West Coast 213/595-1781 Mid-West 312/894-3300 Tex... 214/23...3094

SPECIFICATIONS (VDD = +15V, VCC = +5V, VREF = +10V unless otherwise noted)

PARAMETER[1]	VERSION	TA = +25°C			OVER SPECIFIED TEMP. RANGE		UNITS	TEST CONDITIONS
		MIN	TYP	MAX	MIN	MAX		
ACCURACY								
Resolution	J	8			8		Bits	$\overline{SC8}$ = Logic 0
	L	10			10		Bits	$\overline{SC8}$ = Logic 1
Quantization Uncertainty	J, L			±1/2		±1/2	LSB	
Relative Accuracy	J, L			±1/2		±1/2	LSB	FCLK = 100 kHz
Differential Nonlinearity	J, L			±1/2		±1/2	LSB	See Figure 5 on Page 4
Gain Error	J, L		0.3				% Reading	
Gain Temperature Coefficient	J, L		5			10	ppm Reading per °C	
ANALOG INPUTS								
Analog Input Resistance	J, L		10		5	20	kΩ	
Analog Input Resistance Tempco	J, L		−150				ppm/°C	
Reference Input Resistance	J, L		10		5	20	kΩ	
Reference Input Resistance Tempco	J, L		−150				ppm/°C	
ANALOG OUTPUTS								
Output Leakage Current (IOUT1, IOUT2)	J, L		10			200	nA	VOUT1, 2 = 0V
Output Capacitance COUT1	J, L		120				pF	DB0 through DB9 = Logic 1
COUT2	J, L		40				pF	
COUT1	J, L		40				pF	DB0 through DB9 = Logic "0"
COUT2	J, L		120				pF	
DIGITAL INPUTS								
VINL[2]	J, L		+1.4	+0.8		+0.8	V	VCC = +5V
VINH[2]	J, L	+2.4	+1.4		+2.4		V	
VINL[2]	J, L			+1.5		+1.5	V	VCC = +15V
VINH[2]	J, L	+13.5			+13.5		V	
IINL, IINH[3]	J, L		±0.1	±10			μA	VIN = 0 to VCC
CLK Input Current	J, L		+0.4	+1			mA	During Conversion VCC = +5V; 2.4V ≤ VIN ≤ VCC
CLK Input Current	J, L		+1.7	+3			mA	During Conversion VCC = +15V; 10V ≤ VIN ≤ VCC
CLK Input Current	J, L		±1				μA	VCC = +5V to +15V Conversion Complete or CLK IN ≤ VINL
CIN	J, L		2				pF	
DIGITAL OUTPUTS								
VOUTL	J, L			+0.5		+0.8	V	VCC = +5V, ISINK = 1.6 mA
VOUTH	J, L	+2.4			+2.4		V	VCC = +5V, ISOURCE = 40μA
VOUTL	J, L			+1.5		+1.5	V	VCC = +15V, ISINK = 3 mA
VOUTH	J, L	+13.5			+13.5		V	VCC= +15V, ISOURCE = 1 mA
COUT (Floating) (SYNC, SRO, BUSY, and DB0 through DB9)	J, L		5				pF	VCC = +5V to +15V SRO and SYNC; Conversion Complete $\overline{BUSY}$; BSEN = Logic "0" DB0–DB9; HBEN, LBEN = Logic "0"
ILKG (Floating) (SYNC, SRO, BUSY and DB0 through DB9)			±5				nA	VCC = +5V to +15V SRO and SYNC; Conversion Complete $\overline{BUSY}$; BSEN = Logic "0" DB0–DB9; HBEN, LBEN = Logic "0" VOUT = 0V and VCC

(continued)

PARAMETER[1]	VERSION	TA = +25°C			OVER SPECIFIED TEMP. RANGE		UNITS	TEST CONDITIONS
		MIN	TYP	MAX	MIN	MAX		
DYNAMIC PERFORMANCE								
Conversion Time	J		20	40		40	µs	See Figure 5 on Page 4
	L		40	120		120	µs	
Internal CLK Frequency	J, L		100				kHz	VCC = +5V, CLK Duty Cycle =
(See Figure 2, Page 4 and								50%, R = 33k, C = 760 pF
Section 6, Page 5)	J, L		100				kHz	VCC = +15V, CLK Duty Cycle =
								50%, R = 10k, C = 2500 pF
LBEN, HBEN Propagation Delay								VCC = +5V
tON(EN)	J, L		650				ns	LBEN, HBEN = 0V to +3V
tOFF(EN)	J, L		200				ns	Data Bit Load = 5k, 10 pF
								Measured from 50% of Enable
								Input to 50% Point of Data
								Bit Output
BSEN Propagation Delay								VCC = +5V
tON(BSEN)	J, L		450				ns	BSEN = 0V to +3V
tOFF(BSEN)	J, L		200				ns	$\overline{BUSY}$ Load = 5k, 10 pF
								Measured from 50% Point of
								BSEN Input Waveform to 50%
								Point of $\overline{BUSY}$ Output Waveform
Convert Start (STRT)[4]	J, L	0.5					µs	
Pulse Duration Requirement								
POWER SUPPLIES								
VDD	J, L		+5 to				V	See Figures 3 and 4, Page 4
			+15					
VCC	J, L		+5 to				V	
			VDD					
IDD	J, L		0.2	2			mA	VDD = +15V, fCLK = 0 to 100 kHz
								Continuous Conversion (80% Duty
								Cycle)
ICC	J, L		0.02	2			mA	VCC = +5V, fCLK = 0 to 100 kHz
								Continuous Conversion (80% Duty
								Cycle)
	J, L		0.1	2			mA	VCC = +15V; fCLK = 0 to 100 kHz
								Continuous Conversion (80% Duty
								Cycle)
PRICE (1-49)	AD7570JD			35.00			$	
	AD7570LD			69.00			$	

Specifications subject to change without notice.
[1] "J" version parameters specified for $\overline{SC8}$ = 0.
[2] VINL and VINH specifications applicable to all digital inputs except COMP. COMP terminal must be driven with CMOS levels (i.e., comparator output pullup must be tied to VCC).
[3] IINL, IINH specifications not applicable to CLK terminal. See "CLK input current" in specification table.
[4] STRT falling edge should not coincide with CLK in falling edge.

ABSOLUTE MAXIMUM RATINGS

VDD to GND . . . . . +17V
VCC to GND . . . . . +17V
VCC to VDD . . . . . +0.4V
VREF to GND . . . . . ±25V
Analog Input to GND . . . . . ±25V
Digital Input Voltage Range . . . . . VDD to GND
I_{OUT1}, I_{OUT2} . . . . . ±5 mA
Power Dissipation (package)
up to +50°C . . . . . 1000 mW
Derate above +50°C by . . . . . 10 mW/°C
Operating Temperature . . . . . 0°C to +75°C
Storage Temperature . . . . . −65°C to +150°C

CAUTION:

1. Do not apply voltages higher than VCC or less than GND to any input/output terminal except VREF or AIN.
2. The digital control inputs are zener protected, however permanent damage may occur on unconnected units under high energy electrostatic fields. Keep unused units in conductive foam at all times.
3. VCC should never exceed VDD by more than 0.4V, especially during power ON or OFF sequencing.

INPUT CONTROLS

1. Convert Start (pin 25 – STRT)
When the start input goes to logical 1, the MSB data latch is set to logic 1 and all other data latches are set to logic 0. When the start input returns low, the conversion sequence begins. The start command must remain high for at least 500 nanoseconds. If a start command is reinitiated *during* conversion, the conversion sequence starts over.

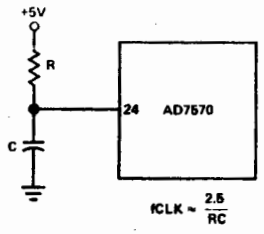
2. High Byte Enable (pin 20 – HBEN)
This is a three-state enable for the bit-9 (MSB) and bit 8. When the control is low, the output data lines for bits 9 and 8 are floating. When the control is high, digital data from the latches appears on the data lines.

3. Low Byte Enable (pin 21 – LBEN)
Same as High Byte Enable pin, but controls bits 0 (LSB) through 7.

4. Busy Enable (pin 27 – BSEN)
This is an interrogation input which requests the status of the converter, i.e., conversion in process or convert complete. The converter status is addressed by applying a logic 1 to the Busy Enable. (See Busy under Output Functions.)

5. Short Cycle 8-Bits (pin 26 – $\overline{SC8}$)
With a logic 0 input, the conversion stops after 8 bits reducing the conversion time by 2 clock periods. This control should be exercised for proper operation of the "J" version. When a logic 1 is applied, a complete 10-bit conversion takes place ("L" version).

6. Clock(pin 24 – CLK)
With an external RC connected, as shown in the Figure below, clock activity begins upon receipt of a Convert-Start command to the A/D, and ceases upon completion of conversion. An external clock (CMOS or TTL/DTL levels) can directly drive the clock terminals, if required.



Generating Internal Clock Frequency

The clock frequency vs. R and C is given in Figure 2.

7. VDD (pin 1)
VDD is the positive supply for all analog circuitry plus some digital logic circuits that are not part of the TTL compatible input/output lines (back-gates to the P-channel devices). Nominal supply voltage is +15V.

8. VCC (pin 22)
VCC is the logic power supply. If +5V is used, all control inputs/outputs (with the exception of comparator terminal) are DTL/TTL compatible. If +15V is applied, control inputs/outputs are CMOS compatible.

OUTPUT FUNCTIONS

1. Busy (pin 28 – $\overline{BUSY}$)
The Busy line indicates whether conversion is complete or in process. Busy is a three-state output and floats until the Busy-Enable line is addressed with a logic 1. When addressed, Busy will indicate either a 1 (conversion complete) or a 0 (conversion in process).

2. Serial Output (pin 8 – SRO)
Provides output data in serial format. Data is available only during conversion. When the A/D is not converting, the Serial Output line "floats." The Serial Sync (see next function) *must* be used, along with the Serial Output terminal to avoid misinterpreting data.

3. Serial Synchronization (pin 9 – SYNC)
Provides 10 positive edges, which are synchronized to the Serial Output pin. Serial Sync is floating if conversion is not taking place.

Note that all digital inputs/outputs are TTL/DTL compatible when VCC is +5V, and CMOS compatible when VCC is +15V.

Table 1. Function Table

PIN NO.	MNEMONIC	FUNCTION
1	VDD	Positive Supply (+15V)
2	VREF	Voltage REFerence (±10V)
3	AIN	Analog INput
4	OUT1	DAC Current OUTput 1
5	OUT2	DAC Current OUTput 2
6	AGND	Analog GrouND
7	COMP	COMParator
8	SRO	SeRial Output
9	SYNC	Serial SYNChronization
10	DB9	Data Bit 9 (MSB)
11	DB8	Data Bit 8
12	DB7	Data Bit 7
13	DB6	Data Bit 6
14	DB5	Data Bit 5
15	DB4	Data Bit 4
16	DB3	Data Bit 3
17	DB2	Data Bit 2
18	DB1	Data Bit 1
19	DB0	Data Bit 0 (LSB)
20	HBEN	High Byte ENable
21	LBEN	Low Byte ENable
22	VCC	Logic Supply (+5V to +15V)
23	DGND	Digital GrouND
24	CLK	CLocK
25	STRT	STaRT
26	$\overline{SC8}$	Short Cycle 8 Bits
27	BSEN	BuSy ENable
28	$\overline{BUSY}$	BUSY

UNIPOLAR BINARY OPERATION

Figure 10 shows the circuit connections required for unipolar analog inputs. If *positive* analog inputs are to be quantized, VREF must be *negative*, and the OUT1 (pin 4) terminal of the AD7570 must be connected to the "+" comparator input. For *negative* analog inputs, VREF must be *positive*, and the OUT1 terminal connected to the "−" input of the comparator.

For clarity, the digital control functions have been omitted from the diagram. For proper use of the digital input/output control functions, refer to the pin function descriptions on page 5.

The input voltage/output code relationship for unipolar operation is shown in Table 2. Due to the inherent multiplying capability of the internal D/A converter, the AD7570 can accurately quantize full scale ranges of 10V to 1V. It should be noted, however, that for smaller full scale ranges, the resolution and speed limitations of the comparator impose a limitation on the maximum conversion rate.

BIPOLAR (OFFSET BINARY) OPERATION

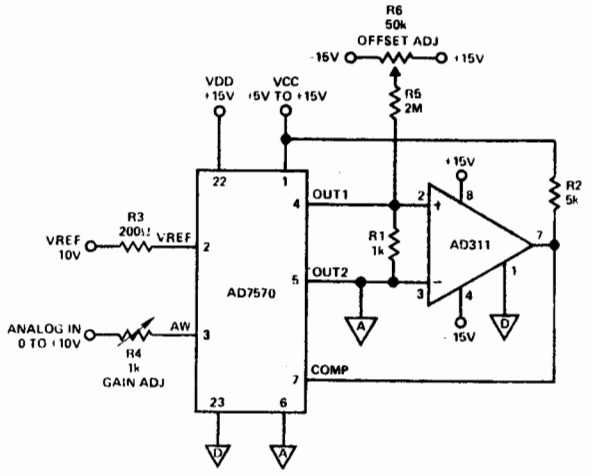
Figure 11 shows the AD7570 configured for offset binary (modified 2's complement) operation. Input voltage/output codes are shown in Table 3.

Amplifier A1, in conjunction with resistors R1, R2, and R3, offsets the bipolar analog input by full scale, and reduces its gain by a factor of 2. The analog signal applied to the AIN terminal is, therefore, a unipolar signal of 0 to +V or 0 to −V, depending on the polarity of VREF. Note that the offset (as well as the scale factor) changes if VREF drifts.

ADJUSTMENT PROCEDURE FOR UNIPOLAR OPERATION

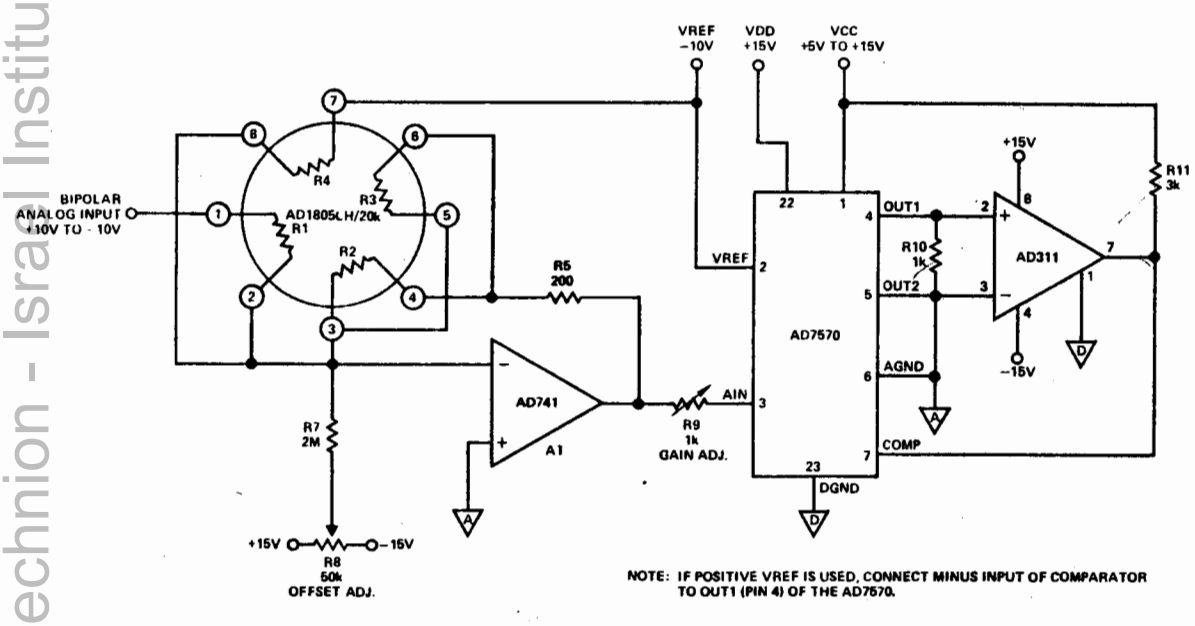
Zero Offset Adjustment

1. Apply continuous STRT command to the AD7570 STRT input. *Note:* STRT must be at intervals long enough to allow a complete conversion.
2. Apply 0 + 1/2LSB (0 − 1/2LSB if positive VREF is used) to the AIN terminal.
3. Observe the SRO output line (synchronize oscilloscope to SYNC terminal of AD7570). Adjust the offset potentiometer (R6) until the LSB flickers between 0 and 1, and all other data bits are logic "0." (See Figure 6, timing diagram, for correlation of SRO and SYNC.)



NOTE:
IF POSITIVE VREF IS USED, THE ANALOG INPUT RANGE IS 0 TO −VREF, AND THE COMPARATOR'S (−) INPUT SHOULD BE CONNECTED TO OUT1 (PIN 4) OF THE AD7570.

Figure 10. Unipolar Operation



NOTE: IF POSITIVE VREF IS USED, CONNECT MINUS INPUT OF COMPARATOR TO OUT1 (PIN 4) OF THE AD7570.

Figure 11. Bipolar Operation

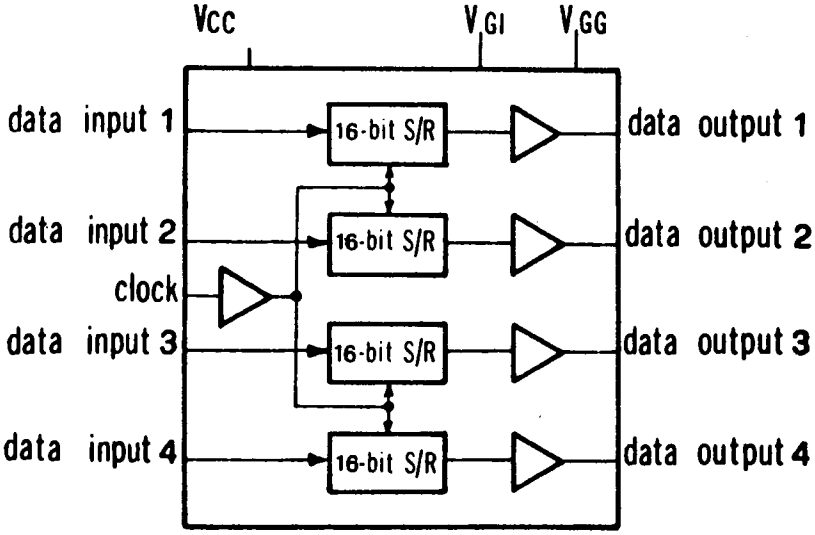
MTNS QUAD 16-BIT STATIC SHIFT REGISTER SL-5-4016/SL-7-4016

Description

The SL-5-4016 / SL-7-4016 are Quad 16 bit shift registers belonging to a standard family of DTL/TTL/MOS compatible registers which are constructed using low threshold Silicon Nitride passivated P-Channel enhancement mode field effect trapsistors. The SL - 5-4016 is packaged in 14-lead ceramic DIL. The SL-7-4016 is the plastic version of this device. Each register has one serial input and one serial output, and the clock input is common to the four registers. All inputs, including clock, can be driven directly from DTL/TTL logic levels, and each output can directly drive DTL/TTL without external interfacing components. Data is read into each register when the clock input is at logic "0" and fs, read out following a positive going (0→1) clock transition. Data can be stored indefinitely in the register with the clock held in either logic state.

Design features

- MTNS Metal Thick Oxide Nitride Silicon Process.
- 64 bit of static shift register per package.
- DC to 2MHz operation.
- Directly interfaces with DTL/TTL at clock data inputs and outputs.
- Data stored independently of period at either clock logic level.



Electrical Characteristics

Standard Conditions for DTL/TTL Compatible Usage :
V_{CC} = + 5v ± 0.5v V_{GG} = - 12v ± 1v V_{GI} = GND
(Substrate at V_{CC})
Temperature Range 0°C ÷ 70°C

Maximum Ratings

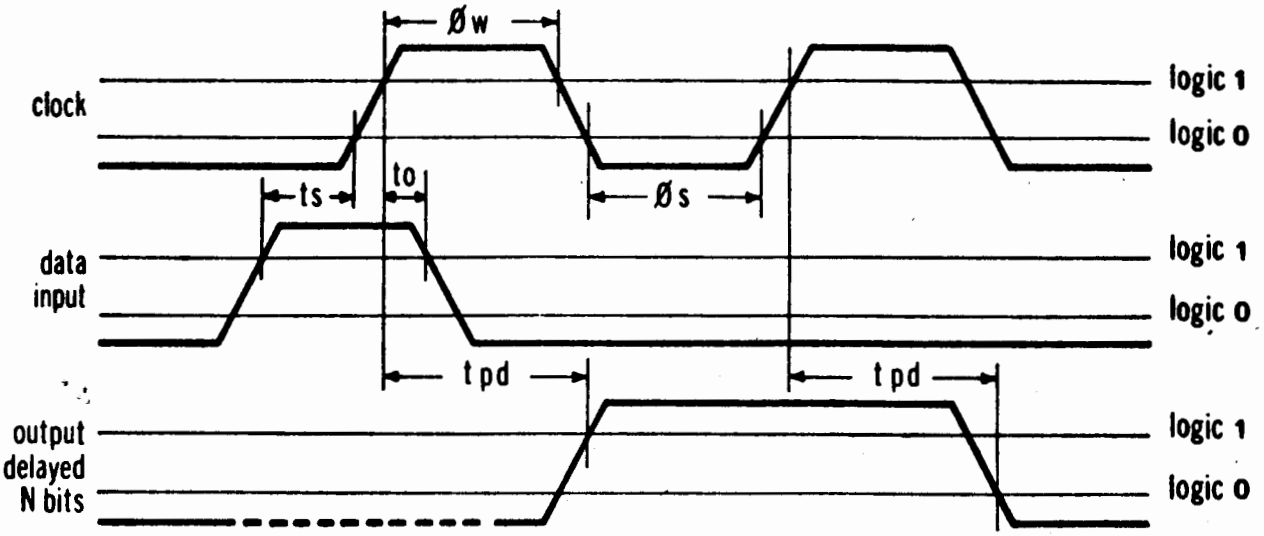
V_{GI} & V_{GG} (with respect to V_{CC})	-20v to ± 0.3v
Clock & logic inputs (with respect V_{CC})	-20v to + 0.3v
Storage Temperature	-55 to + 150°C
Operating Temperature	0°C to + 70°C

Characteristics	Min.	Typ.	Max.	Units	Conditions
Clock Inputs (Fig. 1)					
Repetition Rate,	D.C.	-	2.0	MHz	
Pulse Width, ϕW	230	-	-	nS	
Pulse Separation, ϕS	230	-	-	nS	
Rise & Fall times	-	-	1.0	µS	
Logic 0 Level	-	-	+ 0.8	Volts	
Logic 1 Level	V_{CC} -1.5	-	-	Volts	
Noise Immunity	0.4	-	-	Volts	
Input Leakage	-	-	1.0	µA	Measured at V_{IN} = V_{GG} at 25°C
Input Capacitance	-	15	-	pF	Measured at V_{IN} = V_{CC}
Data Inputs (Fig. 1)					
Set Up Time, ts.	100	-	-	nS	
Data Overlap, to.	50	-	-	nS	
Logic 0 Level	-	-	+ 0.8	Volts	
Logic 1 Level	V_{CC} -1.5	-	-	Volts	
Noise Immunity	0.4	-	-	Volts	
Input Leakage	-	-	1.0	µA	Measured at V_{IN} = V_{GG} at 25°C
Input Capacitance	-	5	-	pf	Measured at V_{IN} = V_{CC}
Data Outputs (Fig. 1)					
Logic 0 Level	-	-	+ 0.4	Volts	I_{OL} = 1.6 mA
Logic 1 Level	V_{CC} -1.0	-	-	Volts	I_{OH} = 100 µA
Propagation Delay, tpd.	-	-	300	nS	1 TTL/DTL Load and 10 pf at 25°C
Power	-	450	-	mW	Nominal Power Supply

Mode of Operation

The 4 separate shift registers share a common CLOCK Input, DATA being clocked in during the Logic 0 state of CLOCK and the OUTPUT changing on the 0 to 1 transition of the CLOCK pulse.
The OUTPUT is in the same sense as the DATA INPUT, i.e., no inversion but delayed by 16 bits.

Figure 1 Waveform Timing



Compatible Usage

The characteristics of the GIANT Standard Family are designed to enable users of other logic forms (TTL, DTL, MTOS, etc.) to take advantage of this range without interfacing problems. As this gives rise to a variety of logic levels the following is a guide to the use of GIANT Standard Family devices in mixed logic systems.

Dual Supply

$V_{CC} = +5_V$, $V_{GG} = 12_V$, $V_{GI} = 0_V$

Logic levels : as defined in electrical characteristics.

1 GIANT devices can drive as many other GIANT devices as speed requirements permit.
2 GIANT devices can drive one TTL/DTL load and as many other GIANT devices as speed requirements permit.
3 TTL/DTL driving GIANT devices may not drive other TTL/DTL loads.

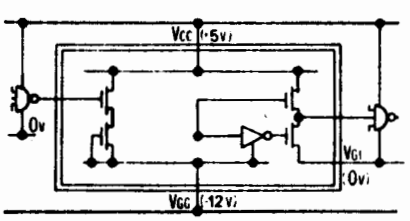
Single Supply

$V_{CC} = 0_V$, $V_{GG} = -17_V$, $V_{GI} = -17_V$

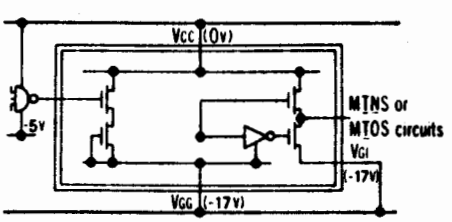
Logic Levels : Input Output

Logic 0 Level - 3.7V Min. - 11V Min.
Logic 1 Level - 1.5V Max. -1.0V Max.

1 GIANT and MTOS devices can be mixed indiscriminately. TTL/DTL referenced to -5 V can be used with GIANT devices as for dual supply conditions.

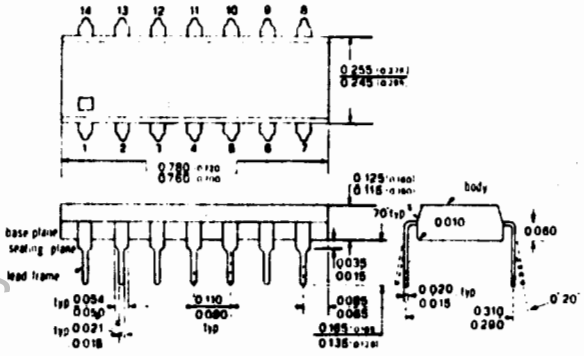


Power Connection Diagram



Power Connection Diagram

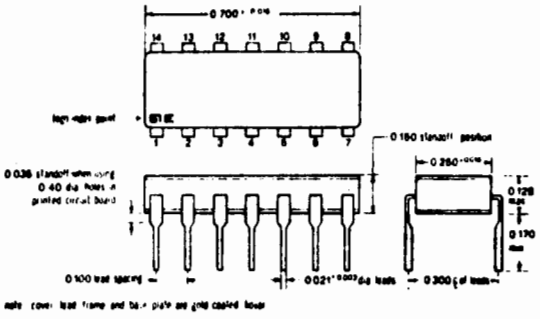
PACKAGE - 14 LEAD PLASTIC DIP



TERMINALS

P/N FUNCTION	P/N FUNCTION
1 - V_{CC}	8 - Input 2
2 - Output 2	9 - Input 3
3 - Output 1	10 - N.C.
4 - V_{GI}	11 - Input 4
5 - N.C.	12 - V_{GG}
6 - Input 1	13 - Output 4
7 - Clock	14 - Output 3

PACKAGE - 14 LEAD DUAL-IN-LINE



TERMINALS

P/N FUNCTION	P/N FUNCTION
1 - V_{CC}	8 - Input 2
2 - Output 2	9 - Input 3
3 - Output 1	10 - N.C.
4 - V_{GI}	11 - Input 4
5 - N.C.	12 - V_{GG}
6 - Input 1	13 - Output 4
7 - Clock	14 - Output 3

GM - 37/6 - CB0069A - 2000

GENERAL INSTRUMENT EUROPE S.p.A.
20149 MILANO P.ZZA AMENDOLA 9 (Italy) - Tel. 469.77.51/2/3/4/5 - Cable: GINEUR Milano - Telex: GINEUR 31454

MOS DIVISION & FACTORY: GIUGLIANO (NAPOLI) Tel. 85 21 56 - Telex. 71108 GINEUR - Cable: GINEUR - GIUGLIANO (NAPOLI)
GENERAL INSTRUMENT FRANCE 11/13 Rue Gandon 75 PARIS 13eme - Tel. 588.74.31 - Telex: 28766
GENERAL INSTRUMENT MICROELECTRONICS LTD. 57/61 Mortimer Street LONDON W1N 7TD - Tel. 01-636 2022 - Telex: 21272
GENERAL INSTRUMENT DEUTSCHLAND GmbH Neumarkter Str. 61 - (8) MUNCHEN 80 - Tel. 45.01.81 - Telex: 524523

VARADYNE SYSTEMS
A DIVISION OF VARADYNE, INC.

DIGITAL-TO-ANALOG CONVERTER

MODEL DAC-29 SERIES

GENERAL DESCRIPTION

Varadyne's DAC-29 digital to analog converter offers a significant breakthrough in cost/performance which makes this device ideally suited for the OEM design engineer. The combination of low cost and moderate output settling time recommends the DAC-29 series for application in computer display systems, data transmission, semiconductor test equipment and programmed/feedback control systems or any requirement where high accuracy or high resolution are not of prime importance.

DAC-29 converters incorporate the latest solid state technology and design concepts in one completely self contained compact plastic case, measuring only 2" x 2" x 0.4" which can be mounted on the users printed circuit board much like any component would be.

Each DAC-29 contains input buffer logic, electronic switches, resistor ladder network, voltage reference source, output buffer amplifier and requires no external components for operation, merely apply D.C. power.

DAC-29 is available in two versions, eight binary bits or two digit BCD. Full scale output can be either unipolar (0 to + 10V @ 5 ma) or bipolar (± 5V @ ± 5 ma) by means of externally programming (jumper wire) the unit.

Output settling time is 20 μsec's to ± 0.2% of full scale. Overall linearity is ± 0.2% with a voltage resolution of 40 millivolts for binary versions and 100 millivolts for B.C.D. units. Input digital coding is straight binary or two digit BCD for unipolar output and two's complement for bipolar outputs. Overall accuracy is ± 0.2% of full scale, ± 1/2 LSB.

DAC-29's are adjustment free over an operating temperature range of 0° to +70°C and have a temperature coefficient of ± 50 ppM/°C with a long term stability of better than ± 0.05%/YR. All digital inputs are compatible with standard DTL/TTL logic levels. Input power requirements are ± 15VDC @ ± 20 ma.*

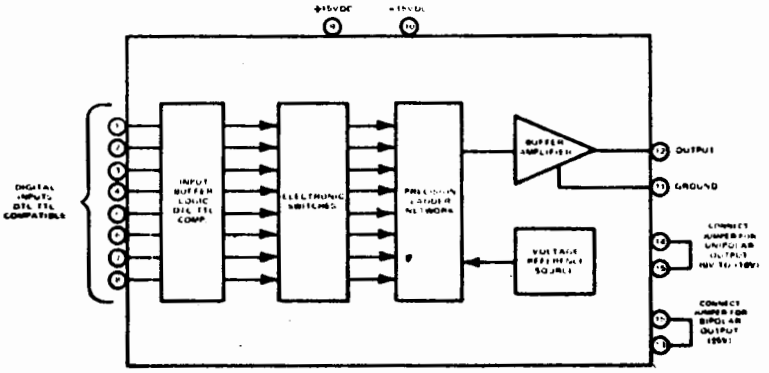
DAC-29 modules are fully encapsulated and feature dual in-line pinning compatibility, .100" grid pin spacing which permits direct plug-in to Augat, Cambion, EECO, etc., circuit boards.

*See Varadyne's ultra-miniature D.C. Power supplies, Powermite Series bulletin #1157010 K

COST / PERFORMANCE BREAKTHROUGH

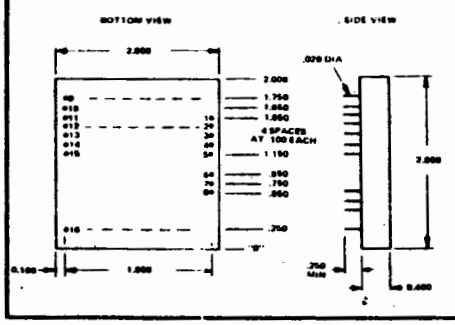
FEATURES

- ☐ Low cost ..... Design for OEM accounts
- ☐ Complete ..... Requires only D.C. power for operation
- ☐ Excellent long term stability.. ±0.05%/YR.
- ☐ Choice of outputs ..... 0 to +10V or ±5V
- ☐ Choice of input coding ..... Binary or BCD
- ☐ Small size ..... 1.6 cubic inches
- ☐ Adjustment free ..... From 0° to +70°c
- ☐ Low power consumption .... 600 milliwatts



BLOCK DIAGRAM
MODEL DAC-29-8B

MODEL DAC 29 D/A CONVERTERS



MODULE INPUT/OUTPUT CONNECTIONS

PIN	FUNCTION	PIN	FUNCTION
1	BIT 1 (MSB)	9	+15V POWER INPUT
2	BIT 2	10	-15V POWER INPUT
3	BIT 3	11	COMMON GROUND
4	BIT 4	12	ANALOG OUTPUT
5	BIT 5	13	- IN
6	BIT 6	14	+ IN
7	BIT 7	15	OFFSET
8	BIT 8 (LSB)	16	NO CONNECTION

SPECIFICATIONS

ELECTRICAL

Digital Inputs:

Resolution ..... 8 binary bits or 2 digit BCD

Coding ..... Parallel data in the following formats:
Straight binary (unipolar output)
BCD (unipolar output)
Two's complement (bipolar output)

Data inputs ..... DTL or TTL compatible, positive logic.

Input Code	V Input Min.	V Input Max.	Bit Status
"0"	0V	+0.8V	Off
"1"	+2.0V	+5.5V	On

Loading: one standard TTL load
IL max. = 1.6 ma @ VIN = 0.4V

Update rate ..... 5MHZ typical, but voltage output limited by output amplifier settling time

Analog output (@ 25°c):

Accuracy ..... ±0.2% of FS ±½ LSB

Output voltage ..... 0 to +10V FS
±5VFS

Output current ..... ±5 ma

Output loading ..... 2K ohms for 0 to +10V output or 1K ohms for ±5V output, in parallel with 1000 pf

Output settling time ..... 20 μsec to ±0.2% of FS (typ.)

Output voltage resolution ..... 40 mV for eight binary bits
100 mV for 2 digit BCD

Linearity ..... ±½ LSB

Temperature coefficient ..... ±50 pp m/°c of FS

Long term stability ..... ±0.05%/YR

Reference source ..... Internal

Input power requirements ..... ±15VDC @ ±20 ma

PHYSICAL – ENVIRONMENTAL:

Operating temperature range ..... 0°c to +70°c

Storage temperature range ..... -55°c to +85°c

Relative humidity ..... Up to 100% non-condensing

Size ..... 2" L x 2" W x 0.4"H plug-in module

Pins ..... 0.020" round gold plated 0.250" long minimum

Case material ..... Black diallyl phthalate, per MIL-M-14

Weight ..... 2 oz.

ORDERING INFORMATION

DAC-29-

NUMBER OF BITS AND CODING
8B – 8 BINARY BITS
8D – 2 DIGIT BCD

INPUT CODING FOR DAC-29 SERIES

ANALOG OUTPUT RANGE (±5V, FS)	BINARY (ONLY) 2'S COMPLEMENT	ANALOG OUTPUT RANGE (0V TO +10V, FS)	STRAIGHT BINARY	ANALOG OUTPUT RANGE (0V TO +10V, FS)	2 DIGIT BCD
+4.96	0111 1111	+9.96	1111 1111	+9.9	1001 1001
+4.37	0111 0000	+8.75	1110 0000	+8.7	1000 0111
+3.75	0110 0000	+7.50	1100 0000	+7.5	0111 0101
+2.50	0100 0000	+5.00	1000 0000	+5.0	0101 0000
0.00	0000 0000	+2.50	0100 0000	+2.5	0010 0101
-2.50	1100 0000	+1.25	0010 0000	+1.2	0001 0010
-3.75	1010 0000	0.00	0000 0000	0.0	0000 0000
-4.37	1001 0000				
-4.96	1000 0001				
-5.00	1000 0000				

VARADYNE SYSTEMS
A DIVISION OF VARADYNE, INC.
1020 TURNPIKE STREET, CANTON, MASS. 02021 TEL. (617) 828-6395 TWX: 710-348-0135

PRINTED IN U.S.A. 3/15/71
Bulletin 136207010K

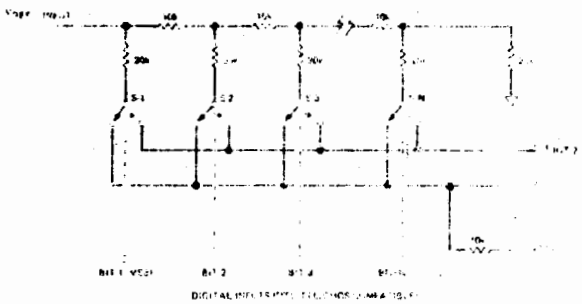
MONOLITHIC CMOS MULTIPLYING 10- & 12-BIT D/A CONVERTERS AD7520 SERIES

GENERAL DESCRIPTION

The AD7520 is a low cost, monolithic 10-bit multiplying D/A converter, packaged in a 16-pin DIP. The AD7521 is a 12-bit resolution multiplying D/A converter, packaged in an 18-pin DIP. Both devices use advanced monolithic CMOS and thin film technologies to provide up to 10-bit differential linearity. The digital inputs directly interface to TTL, DTL, and CMOS logic.

The AD7520 and AD7521 operate from a +5V to +15V supply, and dissipate only 20mW, including the ladder network.

FUNCTIONAL DIAGRAM

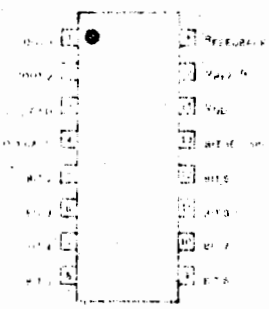


(Switches shown in "High" state)

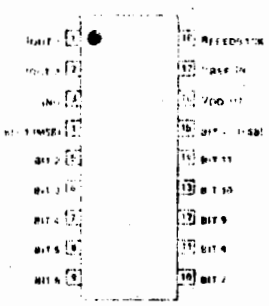
AD7520: N = 10
AD7521: N = 12

PIN CONFIGURATION (TOP VIEW)

AD7520



AD7521



PACKAGE INFORMATION

Suffix D – Ceramic DIP
Suffix N – Plastic DIP

ORDERING INFORMATION

Commercial Grades (0°C to +75°C)

0.2% Diff Nonlinearity	Price (1-49)	0.1% Diff Nonlinearity	Price (1-49)	0.05% Diff Nonlinearity	Price (1-49)
AD7520JD	$20.75	AD7520KD	$29.50	AD7520LD	$39.50
AD7520JN	15.75	AD7520KN	20.50	AD7520LN	27.00
AD7521JD	$25.75	AD7521KD	$34.50	AD7521LD	$44.50
AD7521JN	20.75	AD7521KN	25.50	AD7521LN	32.00

Military Grades (-55°C to +125°C)

0.2% Diff Nonlinearity	Price (1-49)	0.1% Diff Nonlinearity	Price (1-49)	0.05% Diff Nonlinearity	Price (1-49)
AD7520SD	$42.00	AD7520TD	$70.00	AD7520UD	$87.00
AD7521SD	$47.00	AD7521TD	$75.00	AD7521UD	$92.00

SPECIFICATIONS SUMMARY (V_{DD} = +15, V_{REF} = +10V, T_A = +25°C unless otherwise noted)

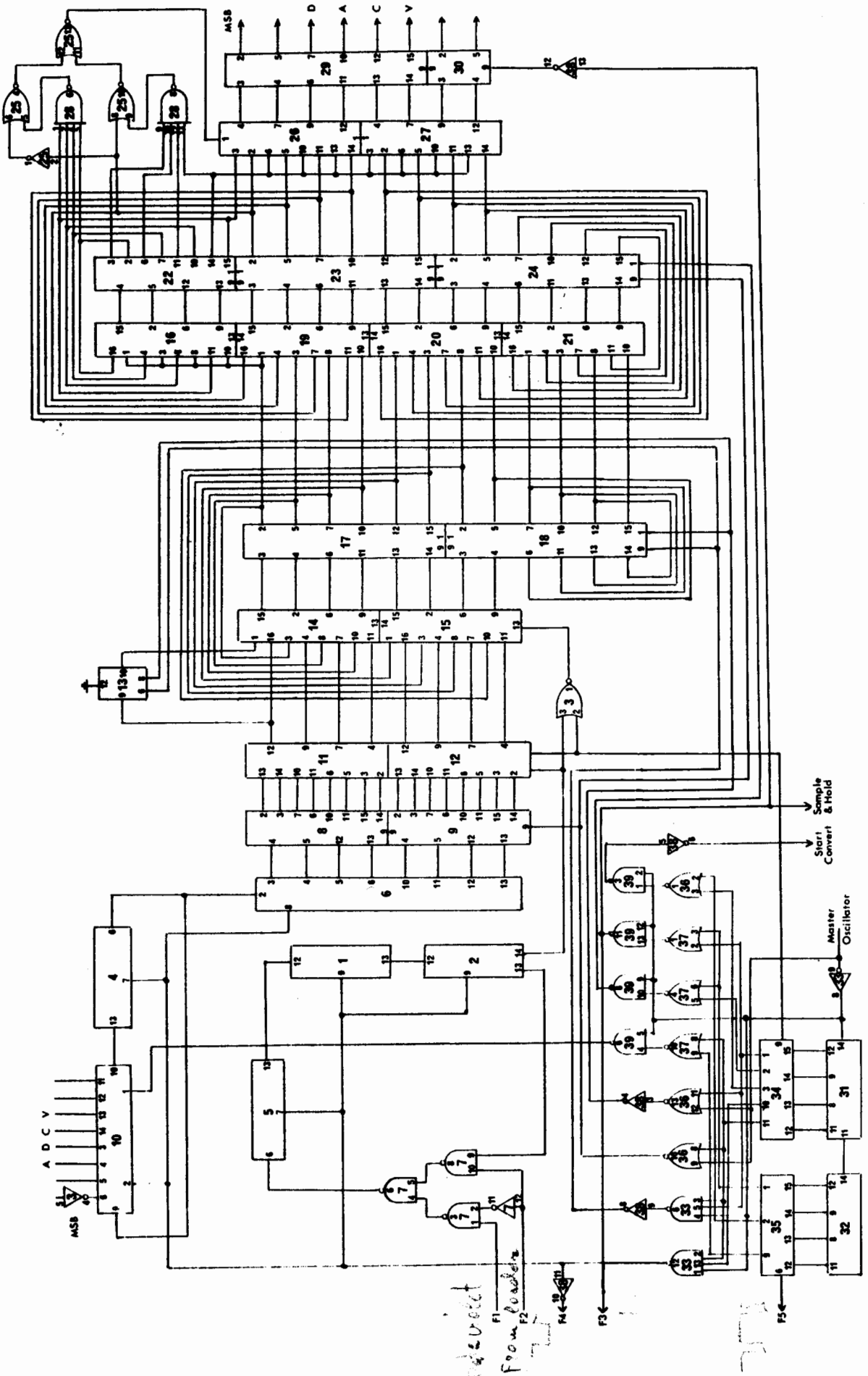
Parameter		AD7520	AD7521	Units	Limit	Test Conditions
DC ACCURACY (Note 1)						
Resolution		10	12	Bits		
Differential Nonlinearity	J S	±0.2 (8 Bit)		% of FSR	Max	S, T, U: over -55°C to +125°C -10V ≤V_{REF} ≤+10V
	K T	±0.1 (9 Bit)		% of FSR	Max	
	L U	±0.05 (10 Bit)		% of FSR	Max	
Relative Accuracy	J S	±0.4		% of FSR	Max	-10V≤V_{REF}≤+10V
	K T	±0.2		% of FSR	Max	
	L U	±0.1		% of FSR	Max	
Differential Nonlinearity Tempco		2		ppm of FSR/°C	Max	-10V ≤V_{REF} ≤+10V
Gain Error (Note 2)		0.3		% of FSR	Typ	
Gain Error Tempco (Note 2)		10		ppm of FSR/°C	Max	
Output Leakage Current (either output)		200		nA	Max	Over specified temperature range
Power Supply Rejection		50		ppm of FSR/%	Typ	
AC ACCURACY						
Output Current Settling Time		500		ns	Typ	To 0.05% All digital inputs low to high and high to low
Feedthrough Error		10		mV p-p	Max	V_{REF} = 20V p-p, 100kHz All digital inputs low
REFERENCE INPUT		±10		V		
Input Resistance		10k		Ω	Typ	
ANALOG OUTPUT						
Voltage Compliance (both outputs)		-100mV, V_{DD}		mV	Max	
Output Capacitance	$I_{OUT\ 1}$	120		pF	Typ	All digital inputs high
	$I_{OUT\ 2}$	37		pF	Typ	
	$I_{OUT\ 1}$	37		pF	Typ	All digital inputs low
	$I_{OUT\ 2}$	120		pF	Typ	
Output Noise (both outputs)		Equivalent to 10kΩ Johnson noise				
DIGITAL INPUTS (Note 3)						
Low State Threshold		0.8		V	Max	Over specified temperature range
High State Threshold		2.4		V	Min	
Input Current (low to high state)		1		μA	Typ	
Input Coding		Binary				
POWER REQUIREMENTS						
Power Supply Voltage Range		+5 to +15		V		
I_{DD}		5		nA	Typ	All digital inputs at GND
		2		mA	Max	All digital inputs high or low
Total Dissipation		20		mW	Typ	

NOTES:
1. Full scale range (FSR) is 10V for unipolar mode and ±10V for bipolar mode.
2. Using the internal $R_{FEEDBACK}$
3. Digital input levels should not go below ground or exceed the positive supply voltage, otherwise damage may occur.

APPENDIX D

SCHEMATIC DIAGRAMS

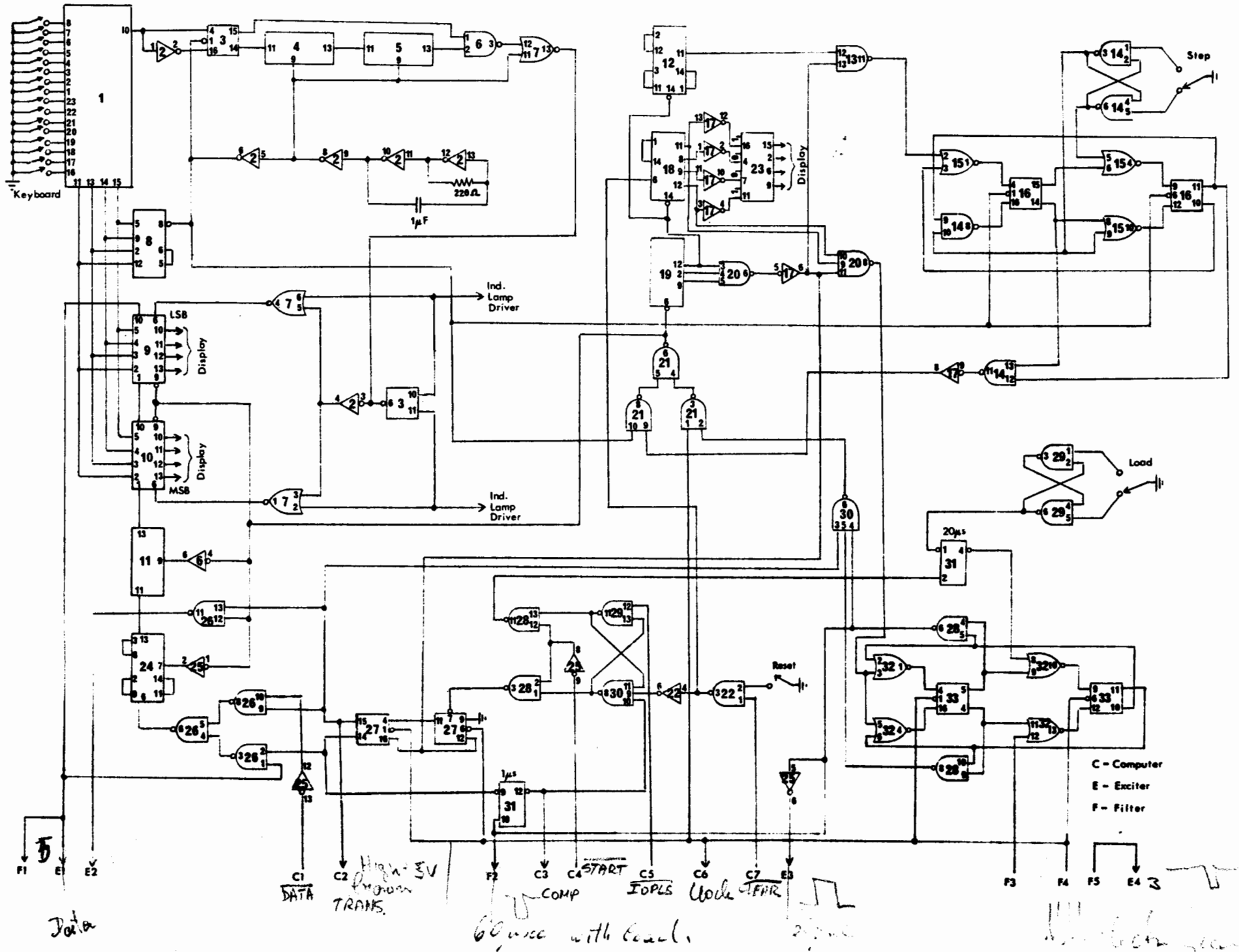
FILTER UNIT



FILTER UNIT

I.C.#	Type	Board #	Position	Description
1	7491	1	1	TTL
2	7491	1	2	TTL
3	7402	1	3	TTL
4	SL-7-4016	1	4	MOS (See Appendix C)
5	SL-7-4016	1	5	MOS (See Appendix C)
6	74164	1	6	TTL
7	7400	1	7	TTL
8	74175	1	8	TTL
9	74175	1	9	TTL
10	74165	1	10	TTL
11	74157	1	11	TTL
12	74157	1	12	TTL
13	7476	2	1	TTL
14	7483	2	2	TTL
15	7483	2	3	TTL
16	7483	2	4	TTL
17	74174	2	5	TTL
18	74174	2	6	TTL
19	7483	2	7	TTL
20	7483	2	8	TTL
21	7483	2	9	TTL
22	74175	2	10	TTL
23	74174	2	11	TTL
24	74174	2	12	TTL
25	7402	3	7	TTL
26	74157	3	8	TTL
27	74157	3	9	TTL
28	7420	3	10	TTL
29	74174	3	11	TTL
30	74174	3	12	TTL
31	7490	4	2	TTL
32	7490	4	3	TTL
33	7420	4	4	TTL
34	7442	4	5	TTL
35	7442	4	6	TTL
36	7402	4	8	TTL
37	7402	4	9	TTL
38	7404	4	11	TTL
39	7400	4	12	TTL

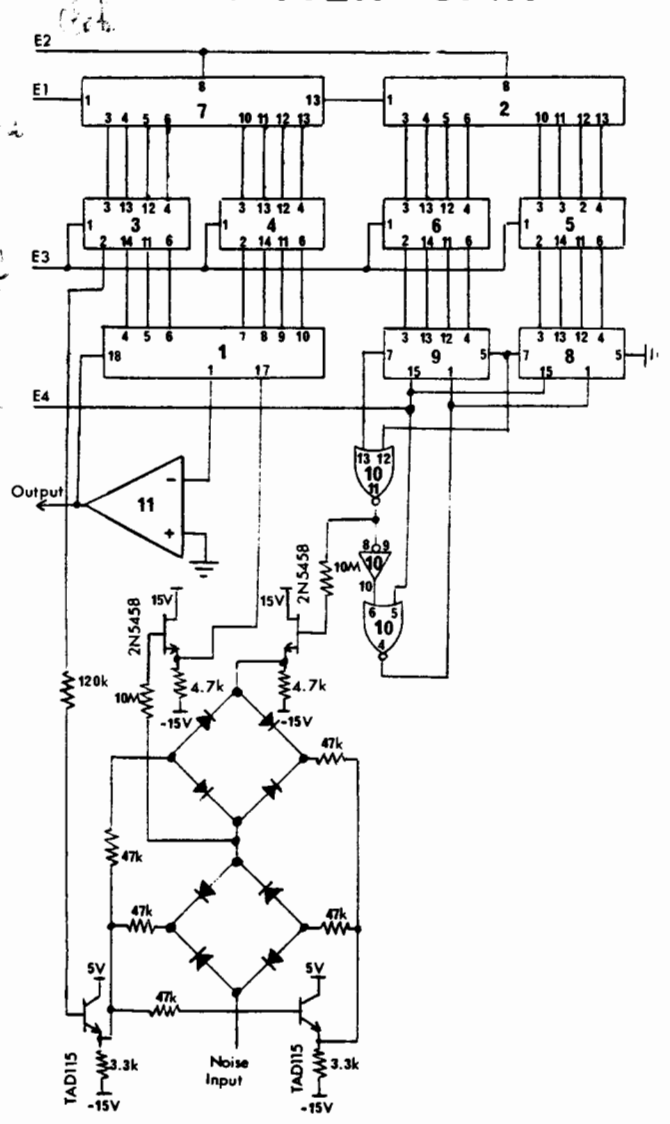
LOADER UNIT



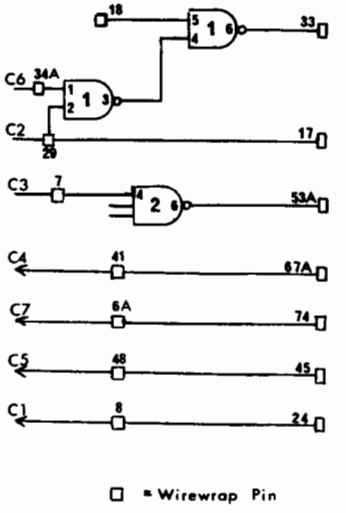
LOADER UNIT

I.C.#	Type	Board #	Position	Description
1	74150	1	11-12	TTL
2	7404	1	2	TTL
3	7476	1	9	TTL
4	7491	1	6	TTL
5	7491	1	5	TTL
6	7400	1	3	TTL
7	7402	1	4	TTL
8	8293	1	8	TTL
9	7495	1	7	TTL
IO	7495	1	10	TTL
11	7491	1	1	TTL
12	7490	2	10	TTL
13	7400	2	6	TTL
14	7400	2	3	TTL
15	7402	2	1	TTL
16	7476	2	2	TTL
17	7404	2	12	TTL
18	7490	2	9	TTL
19	8293	2	7	TTL
20	7410	2	4	TTL
21	7400	2	8	TTL
22	7400	2	5	TTL
23	7483	2	11	TTL
24	SL-7-4016	3	12	MOS (See Appendix C)
25	7404	3	8	TTL
26	7400	3	9	TTL
27	7476	3	1	TTL
28	7400	3	4	TTL
29	7400	3	3	TTL
30	7410	3	7	TTL
31	74123	3	2	TTL
32	7402	3	6	TTL
33	7476	3	5	TTL

EXCITER UNIT



INTERFACE BOARD



□ = Wirewrap Pin

COMPUTER INTERFACE BOARD

I.C.#	Type	Board #	Position	Description
1	7400	1	1	TTL
2	7420	1	2	TTL

EXCITER UNIT

I.C.#	Type	Board #	Position	Description
1	AD7521	1	1	CMOS (See Appendix C)
2	74164	1	2	TTL
3	4029	1	3	CMOS
4	4029	1	4	CMOS
5	4029	1	5	CMOS
6	4029	1	6	CMOS
7	74164	1	7	TTL
8	4029	1	8	CMOS
9	4029	1	9	CMOS
10	4001	1	10	CMOS
11	741	1	11	Operational Amplifier

(כלומר עדכון מקדמים כל 150 דגימות). כל סט פרמטרים מוצפן למילה בת 96 סיביות. פרמטרי הסינתזה מתעדכנים אם כן כל 15 מילישניות. אולם סביר להניח שניתן להפחית בצורה משמעותית את קצבי הדגימה והעדכון מבלי לקלקל הרבה את איכות הקול, כך שניתן להשיג חסכון ניכר בקצב ההעברה (bits/sec) של אות הדבור.

ת ק צ י ר

העבודה הנוכחית עוסקת בתכנון ובניה של מסנן ספרתי רב תכליתי המבוקר באמצעות מחשב. המסנן הינו טרנסוורסלי, בעל עשרה מקדמים וקצב דגימה עד 20 kHz. המסנן, המבוקר ע"י מיני-מחשב Nova-2 הופעל בשני שמושים: א) כמסנן FIR מתוכנת; ב) כמערכת סינתזה של אותות דבור.

ליישום הראשון, כמסנן FIR, פותחה תכנה לחשוב מקדמי המסנן על פי קריטריון שגיאה רבועית מינימלית בקירוב לתגובת תדר נתונה. המקדמים המחושבים נתנים אז להעברה אל המסנן.

היות והמסנן מבוקר ע"י מחשב, תכונותיו ניתנות לשנוי בזמן ע"י עדכון של מקדמיו. אפשרות זו מנוצלת במלואה בסינתזה של הדבור. סינתסייזר הדבור יועד ליצירת גל דבור בזמן אמיתי ממערך של פרמטרים המאפינים את מערכת הדבור האנושית, כלומר מיתרי הקול, מעבר הקול ופתחי הפה והאף. בעת דבור רגיל התכונות של מערכת הקול משתנות בזמן. בסינתסייזר מושג הדבר ע"י עדכון הפרמטרים במרווחים בני 15 מילישניות.

המודל הפונקציונלי של מערכת הקול מורכב ממסנן ספרתי המכיל קטבים בלבד וממומש באמצעות מסנן טרנסוורסלי בחוג סגור. המסנן מעורר הן ע"י סדרת מתקפים (עבור הברות קוליות) או ע"י רעש לבן מוגבל סרט (עבור הברות אל-קוליות). מודל זה ידוע בשם מערכת חזוי ליניארי.

מערכת תכנה פותחה למצוי הפרמטרים הנחוצים מאות דבור דגום. כן נבנה מעורר למסנן כחלק מהמערכת כך שפרמטרי המעורר ניתנים לעדכון בו זמנית עם עדכון מקדמי המסנן. למטרות האנליזה נדגם הדבור בקצב של 10 kHz וחושבו הפרמטרים עבור קטעי דבור בני 450 דגימות כך שקטעים עוקבים חופפים על פני 300 דגימות

(רשימת הציורים - המשך)

עמוד

רשימת הציורים

עמוד

(תוכן עניינים - המשך)

עמוד

תוכן עניינים

הפרויקט נעשה בהנחיית ד"ר דוד מלאך במעבדה לעבוד
אותות של הפקולטה להנדסת חשמל.

ברצוני להודות לד"ר מלאך עבור הנחיתו במשך ביצוע
הפרויקט ולמהנדס המעבדה לעבוד אותות, אלי פולק,
על עזרתו.

תודתי נתונה לקרן ליידי דייויס עבור המלגה
שהוענקה לי במשך תקופת למודי.

עבוד נתונים ואותות דבור בזמן אמיתי
עם מסנן ספרתי מבוקר מחשב

חבור על פרויקט
לשם מילוי חלקי של הדרישות לקבלת התואר
מגיסטר למדעים
בהנדסת חשמל

מאת

רולנד וילק

הוגש לסנט הטכניון - מכון טכנולוגי לישראל
אלול תשל"ו חיפה ספטמבר 1976