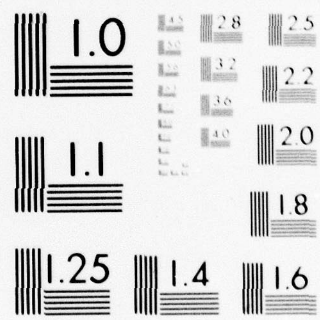
AD-A073 401 NAVAL AIR DEVELOPMENT CENTER WARMINSTER PA COMMUNICA--ETC F/G 17/2
TIES NARROWBAND SIGNAL CONVERSION UNIT DESIGN REPORT.(U)
JUL 79 Y LUI

UNCLASSIFIED NADC-79153-40 NL

1 OF 2
AD
A073401



MICROCOPY RESOLUTION TEST CHART

(14)

REPORT NO. NADC-79153-40

(12)

LEVEL II

(6)

TIES NARROWBAND SIGNAL CONVERSION UNIT DESIGN REPORT.

(10)
Yau-Foon Lui
Communication Navigation Technology Directorate
NAVAL AIR DEVELOPMENT CENTER
Warminster, Pennsylvania 18974

(11) 30 July 1979 (12) 124p.

(9) FINAL REPORT
AIRTASK NO. WF21-200-000
Work Unit No.ZC902

(16) F21200 (17) WF21200000

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

Prepared for
NAVAL AIR SYSTEMS COMMAND
Department of the Navy
Washington, DC 20361

393 533

NOTICES

REPORT NUMBERING SYSTEM - The numbering of technical project reports issued by the Naval Air Development Center is arranged for specific identification purposes. Each number consists of the Center acronym, the calendar year in which the number was assigned, the sequence number of the report within the specific calendar year, and the official 2-digit correspondence code of the Command Office or the Functional Directorate responsible for the report. For example: Report No. NADC-78015-20 indicates the fifteeth Center report for the year 1978, and prepared by the Systems Directorate. The numerical codes are as follows:

CODE	OFFICE OR DIRECTORATE
00	Commander, Naval Air Development Center
01	Technical Director, Naval Air Development Center
02	Comptroller
10	Directorate Command Projects
20	Systems Directorate
30	Sensors & Avionics Technology Directorate
40	Communication Navigation Technology Directorate
50	Software Computer Directorate
60	Aircraft & Crew Systems Technology Directorate
70	Planning Assessment Resources
80	Engineering Support Group

PRODUCT ENDORSEMENT - The discussion or instructions concerning commercial products herein do not constitute an endorsement by the Government nor do they convey or imply the license or right to use such products.

APPROVED BY: William F. Lyons Jr. DATE: 7-30-79

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER NADC-79153-40	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) TIES Narrowband Signal Conversion Unit Design Report		5. TYPE OF REPORT & PERIOD COVERED Final Report
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Yau-Foon Lui		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS Communication Navigation Technology Directorate Code 40, Naval Air Development Center Warminster, PA 18974		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS AIRTASK NO. WF21-200-000
11. CONTROLLING OFFICE NAME AND ADDRESS Naval Air Systems Command Department of the Navy Washington, DC 20361		12. REPORT DATE 30 July 1979
		13. NUMBER OF PAGES 120
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report) UNCLASSIFIED
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

TIES Program
Communication
Digital Signal Processing

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

The task was to design a digital signal processor that is capable of processing communication signals (AM/FM/SSB) in the half duplex mode by using presently available digital signal processing hardware.

Detailed description of various functions to be performed by the processor, processing algorithms, as well as circuits designed to accomplish these junctions are presented in the report.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

S/N 0102-LF-014-6601

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF FIGURES (Continued)

LIST OF FIGURES (Continued)

LIST OF TABLES

1. SUMMARY

The objective of the TIES (Tactical Information Exchange System) program is the integration of the presently used separate CNI (Communication, Navigation and Identification) subsystems into a large system that is programmable, reconfigurable, uses common modules, and can process a variety of signals simultaneously that are currently performed one at a time by dedicated hardware. The system is partitioned into three parts — the frequency conversion unit, the frequency distribution unit, and the signal conversion unit as shown in figure 1.

Because of the difference in signal structure, the signal conversion unit is further broken down into the narrow band unit which handles narrow band signals such as AM, FM, SSB, etc., and the wideband unit which handles signals such as TACAN, IFF, JTIDS, and GPS in the foreseeable future.

As an in house effort to investigate the narrowband signal processing requirements for TIES, a narrowband signal conversion unit (NBSCU) was designed that will handle AM, FM, and SSB signals in the half duplex mode using presently available digital hardware. The unit uses FIR (Finite Impulse Response) filters because FM signals are extremely phase sensitive and FIR filters have exactly linear phase response.

The Arctangent algorithm is used for signal demodulation because of its efficiency. The amplitude of the signal is given by the equation $A = R\cos\theta + Q\sin\theta$. The angle θ is given by $\theta = \tan^{-1}(Q/R)$. In both cases, R and Q are the real and quadrature components of the incoming signal that was frequency translated to baseband.

Detailed descriptions of various functions to be performed by the NBSCU; processing algorithms for various signals, as well as circuits designed to accomplish these functions are presented in the following pages.

2. BASIC CONCEPTS

2.1 Sampling

In order to have a better understanding of the work involved, the basic concept of sampling has to be reviewed. The sampling theorem states that if a band limited signal is sampled at a sufficiently high rate, the original signal can be reconstructed by a set of interpolating functions on the sampled values of the signal. The minimum sampling rate is the well known Nyquist rate of twice the highest frequency of the incoming signal. The sampled spectrum of a signal is shown in figure 2.

Undersampling will cause the repeated spectrum to fold over one another causing aliasing. The reconstruction of the original signal, depending on how severe aliasing is, becomes extremely difficult and can only be recovered with distortion. The amount of distortion will depend on the amount of undersampling and on the shape of the spectrum of the signal being sampled. This is shown in figure 3.

If the incoming signal is at a relatively high frequency, sampling at twice the highest frequency will become impractical. It is known that for a bandpass signal, the sampling rate can be as small as twice the bandwidth of the signal. Another constraint for the bandpass sampling frequency is that the sampling

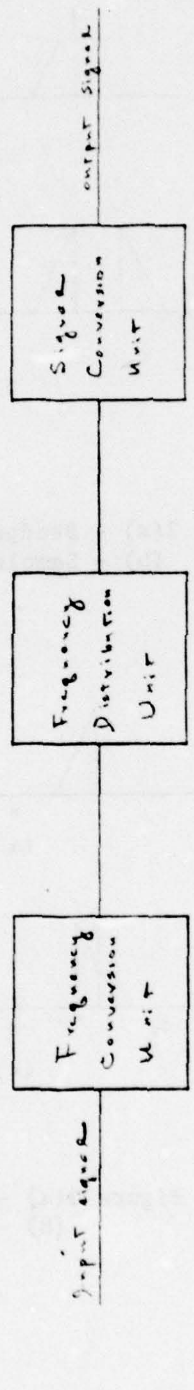


Figure 1 - Simplified TIES Architecture

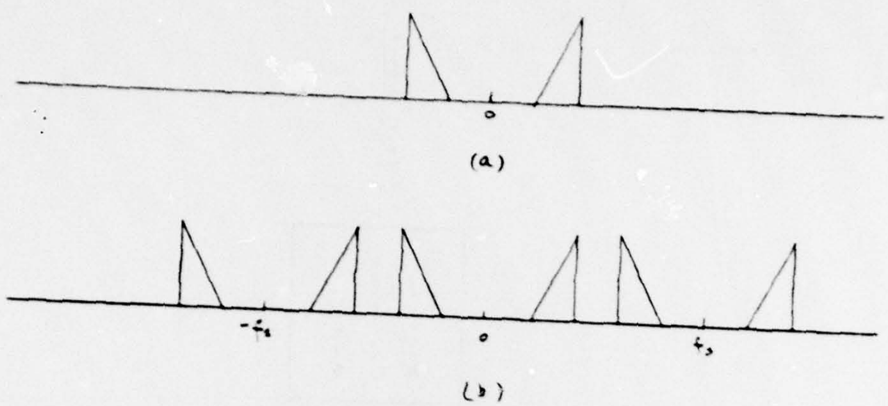


Figure 2(a) - Bandpass Signal
(b) - Sampled Signal Spectrum

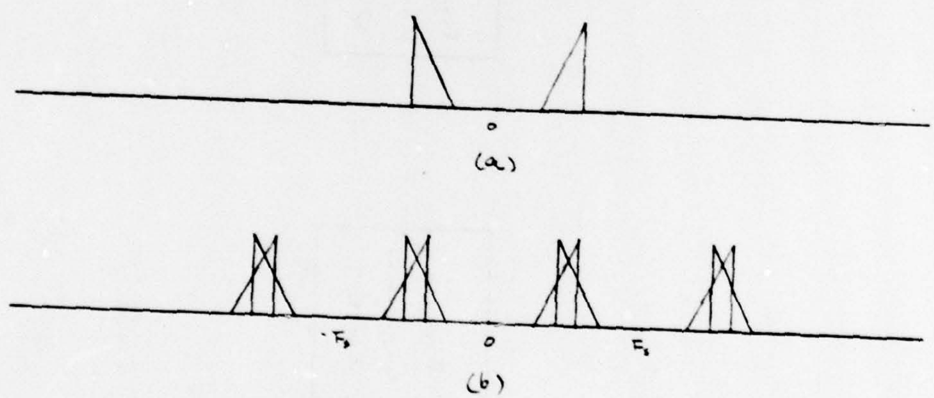


Figure 3(a) - Bandpass Signal
(b) - Effect of Undersampling

rate must satisfy the relationship

$$F_s = \frac{2F_h}{INT(F_h/B)}$$

where F_h is the highest signal frequency,

B is the bandwidth of the signal,

INT is the integer value of the quantity.

Therefore, in bandpass sampling, the sampling rate varies between twice to four times the signal bandwidth. Another effect for bandpass sampling is that of frequency translation. This is shown in figure 4.

It can be seen then, the original signal can be recovered independently of the sampling technique by passing the sampled data through a low pass filter with the proper bandwidth as shown in figure 5.

It is important to note that the duration and shape of the pulses in the sampling waveform do not affect the ability to recover the signal from its samples. The task for processing the conventional signals will be to design a set of digital hardware that will perform the interpolating function of input samples to produce the original baseband signal. The state-of-the-art circuit in sampling would be a sample-and-hold circuit which consists of a high speed electronic switch followed by a capacitor. Upon external command, the switch is closed and the capacitor is charged up to the incoming signal level. The switch is then opened and the charge is then held on by the capacitor until the next command for closing the switch comes along. A simple diagram for a sample-and-hold circuit is shown in figure 6. Sample-and-holds normally have unity gain and are non-inverting. There are several noise sources that are inherently associated with sample-and-hold circuits that will be discussed in Appendix A.

2.2 A/D Conversion

The sampled output of the signal has to be converted into a binary number before any processing can be performed. This process is known as A/D conversion and is accomplished by A/D converters. Most commonly used A/D converters are of 2's complement notation due to the inherent advantage of the 2's complement numbering system in hardware processing. Since the input voltage to the converter needs an infinitely large number of binary digits for representation, another source of error will occur due to the finite digit representation of the analog input signal. This will also be discussed in Appendix A. The output digits of the A/D converter will be the inputs to the narrowband processor. To reduce circuit complexity to a minimum only AM/FM/SSB will be processed one function at a time in the half duplex mode.

3. SIGNAL PROCESSING ALGORITHMS

3.1 AM Transmit

The AM transmit algorithm is shown in figure 7.

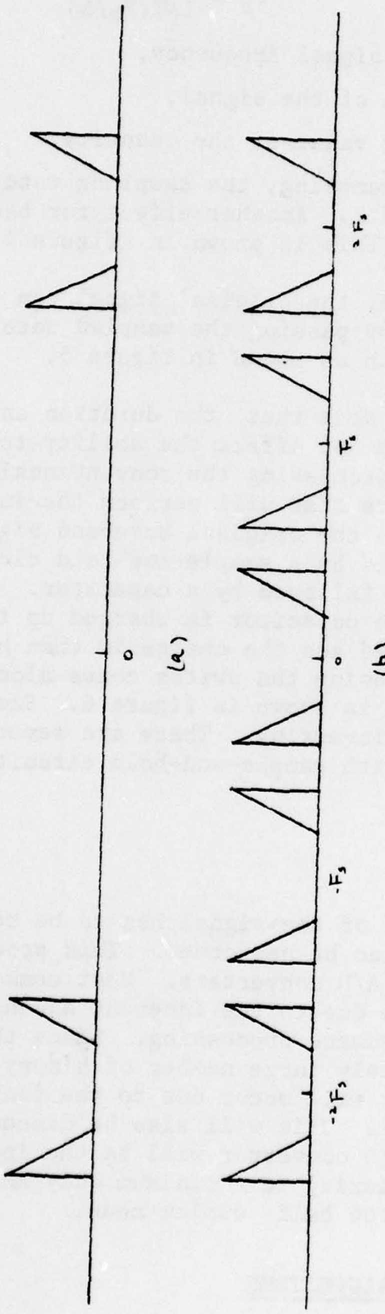


Figure 4(a) - Bandpass Signal
(b) - Frequency Translation by Sampling

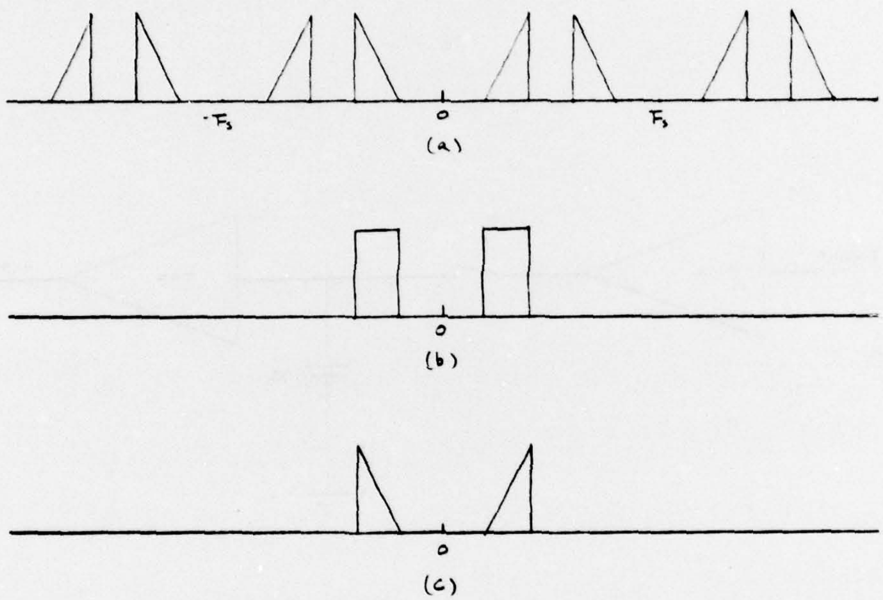


Figure 5(a) - Sampled Signal
(b) - Filter Characteristic
(c) - Signal Recovery by Filtering

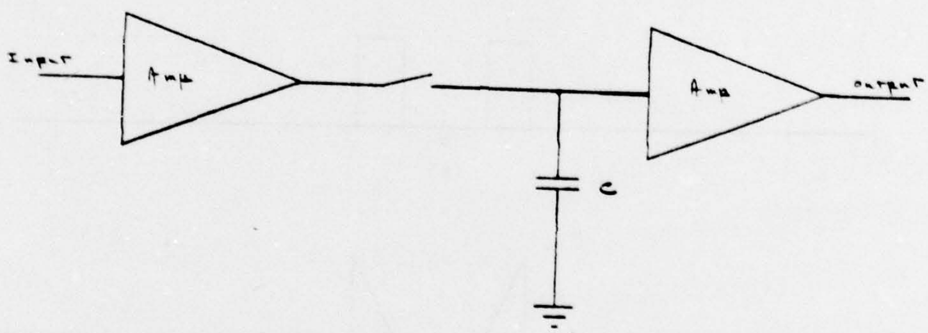


Figure 6 - Simplified Sample and Hold Circuit

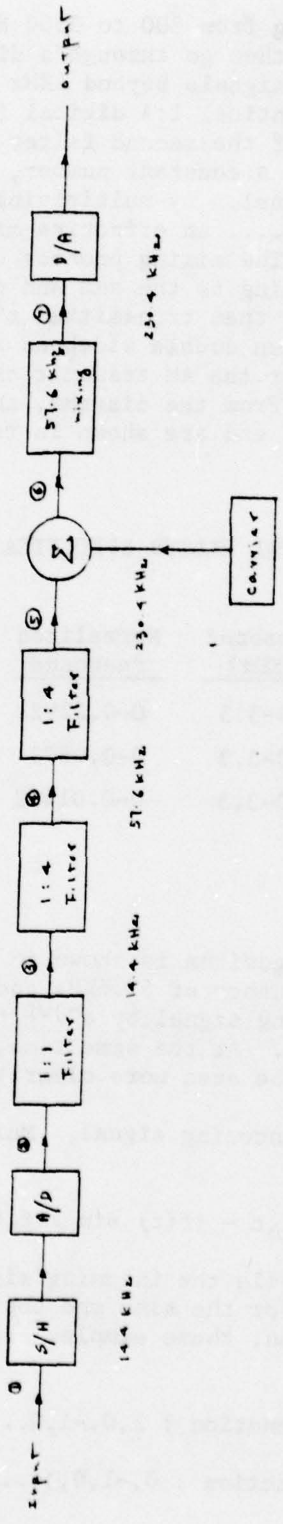


Figure 7 - AM Transmit Algorithm

Audio input ranging from 300 to 3300 Hz is sampled at a rate of 14.4KHz. The sampled input will then go through a digital filter that passes the audio signal, but attenuates signals beyond 4KHz to 50 db. The output will then be interpolated by two identical 1:4 digital filters to change the sampling rate to 230.4KHz. The output of the second filter will be at audio frequency with change in sampling rate only. A constant number, representing the carrier, will be added to this audio signal. By multiplying the resulting samples with the sequence 1, 0, -1, 0, ... an effective mixing process with samples of a cosine function is obtained. The mixing process causes the signal to shift in frequency by ±57.6KHz, corresponding to the sum and difference frequencies in analog mixing. This signal is then transmitted through the D/A converter. If the carrier is set to 0, then double sideband suppressed carrier will be obtained. The spectral diagram for the AM transmit chain at points 1, 2, 3, 4, 5, 6, 7 is shown in figure 8. From the diagram, the specifications for the digital filters can be obtained and are shown in table 1.

TABLE 1 - DIGITAL FILTER SPECIFICATIONS FOR AM TRANSMIT MODE

Filter	Sampling Frequency (KHz)	Passband (KHz)	Normalized Passband	Stopband (KHz)	Normalized Stopband	Attenuation (db)
1	14.4	0-3.3	0-0.2292	4-7.2	0.2780-0.5	50
2	57.6	0-3.3	0-0.0573	10.4-28.8	0.1806-0.5	50
3	230.4	0-3.3	0-0.01432	53.6-115.2	0.2326-0.5	50

3.2 AM Receive

The AM receive algorithm is shown in figure 9. The input signal is converted into an IF frequency of 57.6KHz and sampled at a rate of 230.4KHz. By multiplying the incoming signal by e^{-jwt} the incoming IF frequency is then translated to baseband. At the same time, two channels (one real, one quad) are formed. This can be seen more clearly as follows:

Let f(t) be the incoming signal. Multiplication by $e^{-j2\pi f_o t}$ yields the following:

$$f(t) \cos 2\pi f_o t - jf(t) \sin 2\pi f_o t$$

Since f_o is 57.6KHz while the incoming signal is sampled at 230.4KHz, four samples are required for the sine and cosine functions in the above expression. For ease of computation, these samples can be chosen to have the following values:

For the cosine function : 1,0,-1,0........

For the -sine function : 0,-1,0,1.......

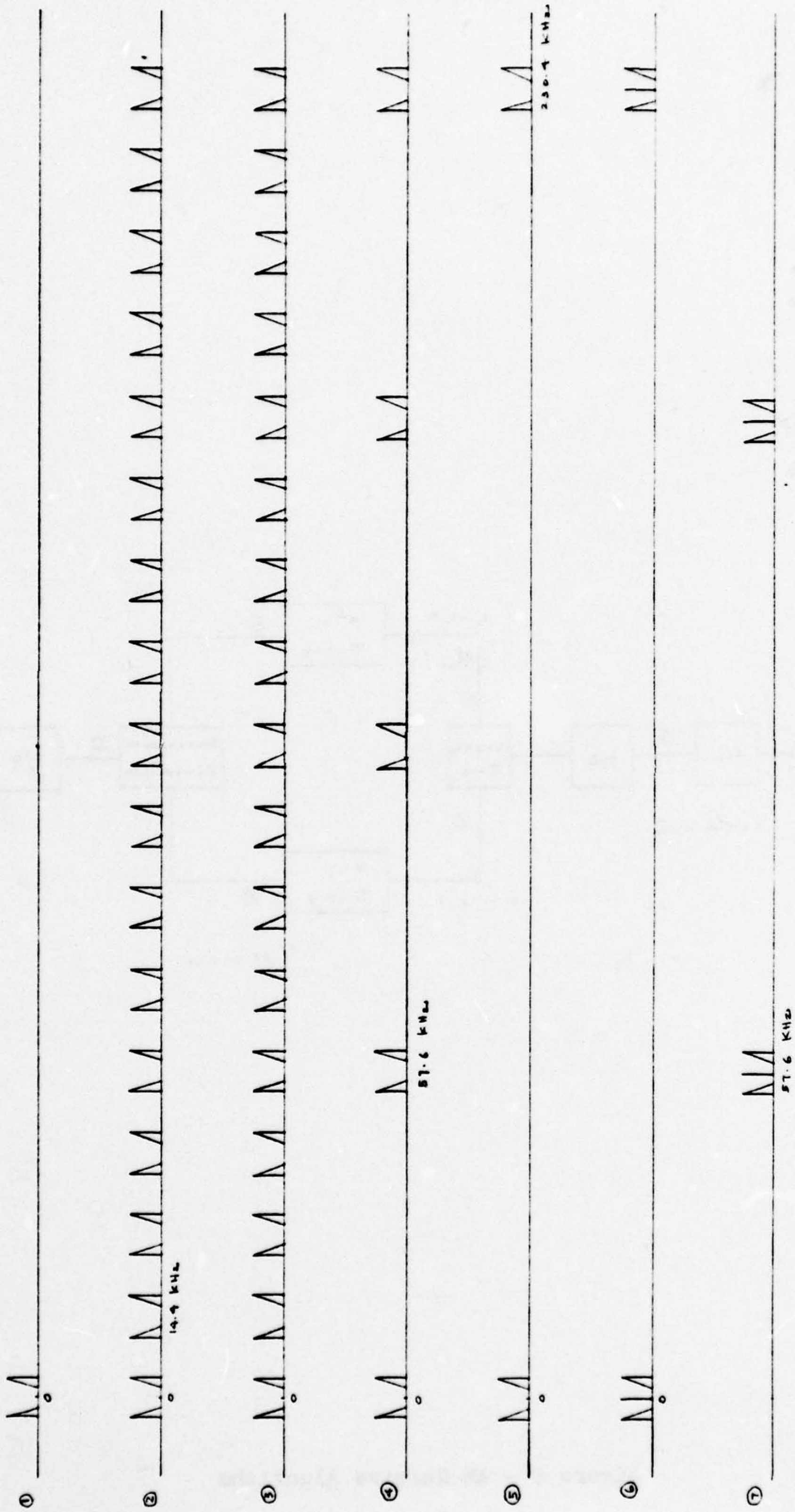


Figure 8 - AM Transmit Spectra

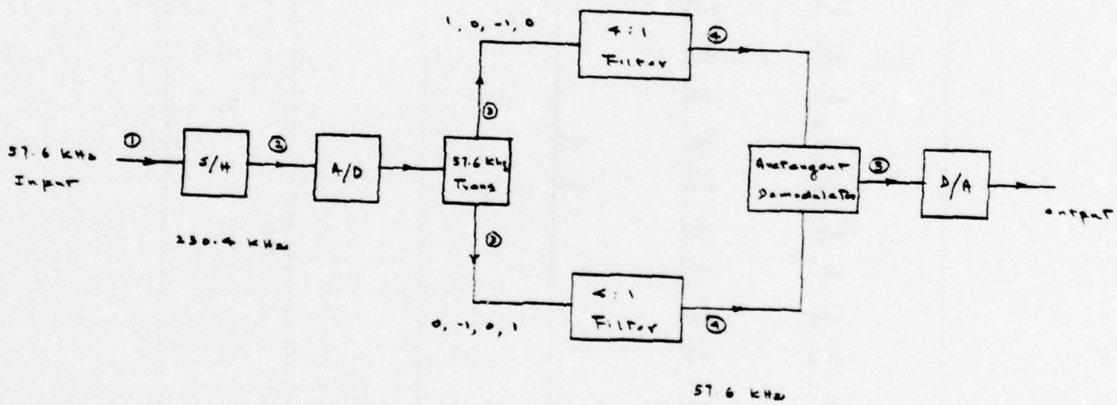


Figure 9 - AM Receive Algorithm

Therefore, by changing the signs of the input samples and routing the samples alternately to the 4:1 resampling filter with appropriate zero filling as shown in figure 9, the frequency translation can be accomplished rather easily. The output of the frequency translater will feed the two 4:1 desampling filters on the real and quad channels. However, close examination of the block diagram indicates that the function of the frequency translation can be readily absorbed into that of the two 4:1 filters. This can be seen from the following:

If it is assumed that the 4:1 filter is of N stages (where N can be any integer) with coefficients for the impulse response h_o, h_1,......h_{N-1}, the output of the filter to a set of input samples is given by the convolution sum,

$$Y(n) = \sum_{k=o}^{N-1} h(k)\, X\,(n-k)$$

where $Y_{(n)}$ is the output sample and X(n-k) are the input samples. This process is shown in figure 10.

However, the input data to the 4:1 resampling filter in this case is multiplied by 1,0,-1,0.....for the real channel and by 0,-1,0,1,........in the quadrature channel. It can be seen that in both channels, alternate samples have value 0 while the signs change alternately for samples that are not zero. Using these input samples for the 4:1 filter, figure 11 is obtained.

Notice that instead of multiplying the input samples by 1,0,-1,0....and 0, -1,0,1....for the two channels, the coefficients for the filter can be varied to obtain output samples as shown in figure 12.

Notice also that while the two outputs of the two figures look identical, they are in fact not the same upon close examination. A comparison made between the two methods used for the real channel shows the difference for the case N=4. This is shown in figure 13.

It can be seen clearly that only every fourth output of the two methods are identical. While the other two sets of outputs are different, no difficulty is encountered because in a 4:1 filtering process only one out of every fourth consecutive output is used. Thus, for this application, the functions for the two methods are identical. Consequently, the frequency translation hardware can be eliminated. The filtering process can be simplified by combining the two filters in the two channels into a single filter. This can be accomplished by changing the sign of alternate pairs of coefficients and routing alternate sum of products to two sets of registers for storage. This is shown in figure 14. While this looks more complicated than the single channel operation, it is in fact very simple to implement as will be seen later on.

The baseband signal, due to the process of frequency translation, will be complex. It has a real component as well as a quadrature component as shown in figure 15.

The magnitude of the vector will then be the desired AM output. Two methods can be used for this computation and are given by the following two equations.

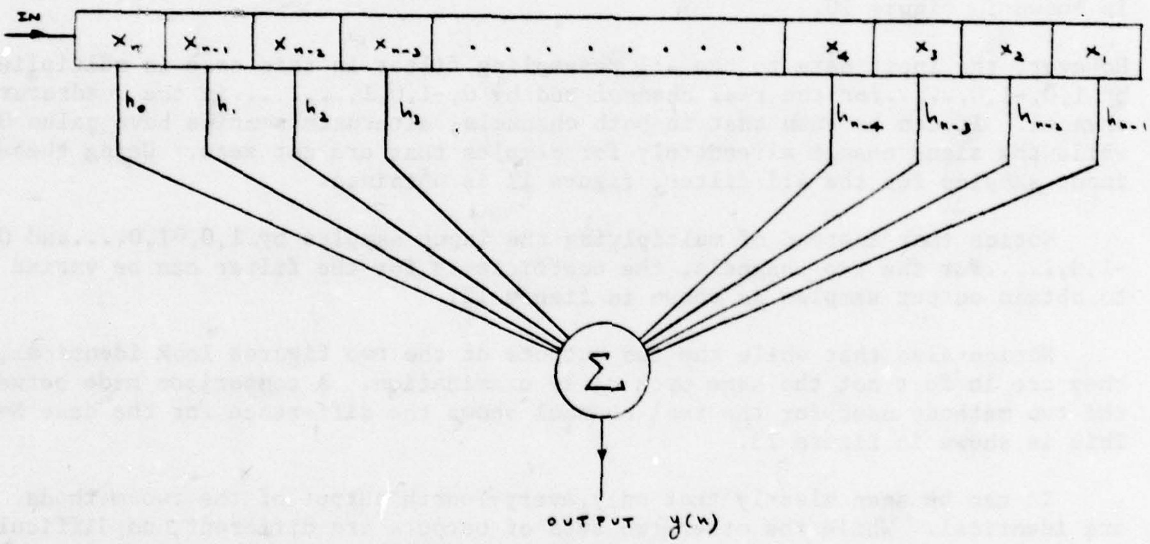
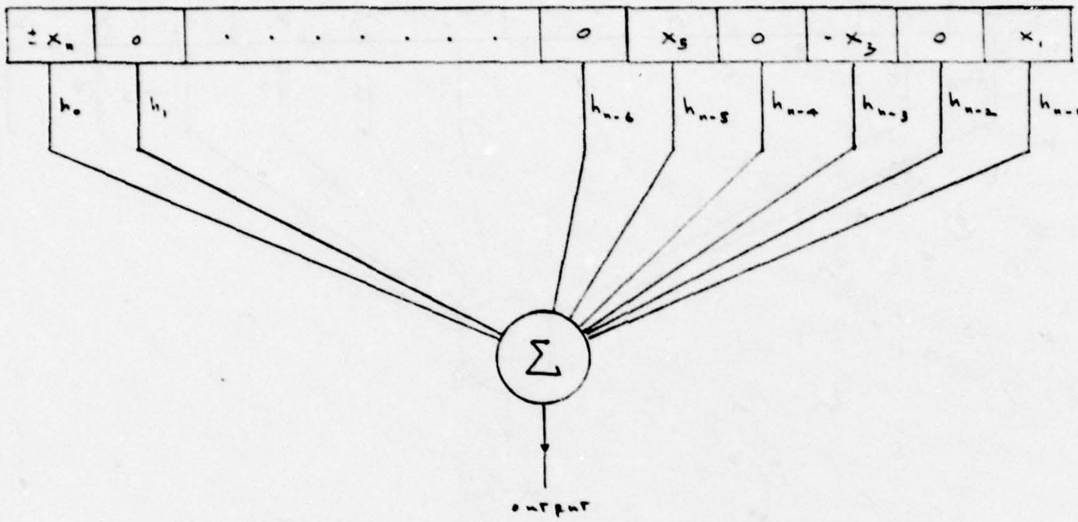
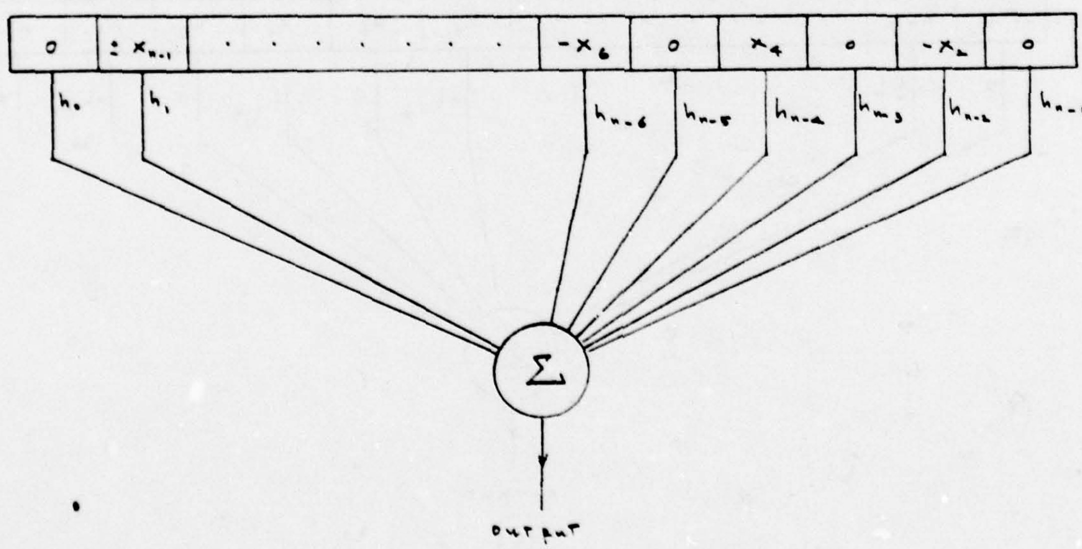


Figure 10 - Sum of Product Computation for Convolution Filter



(a)



(b)

Figure 11 - Filtering of Frequency Translated Signal

(a) Real Channel
(b) Quad Channel

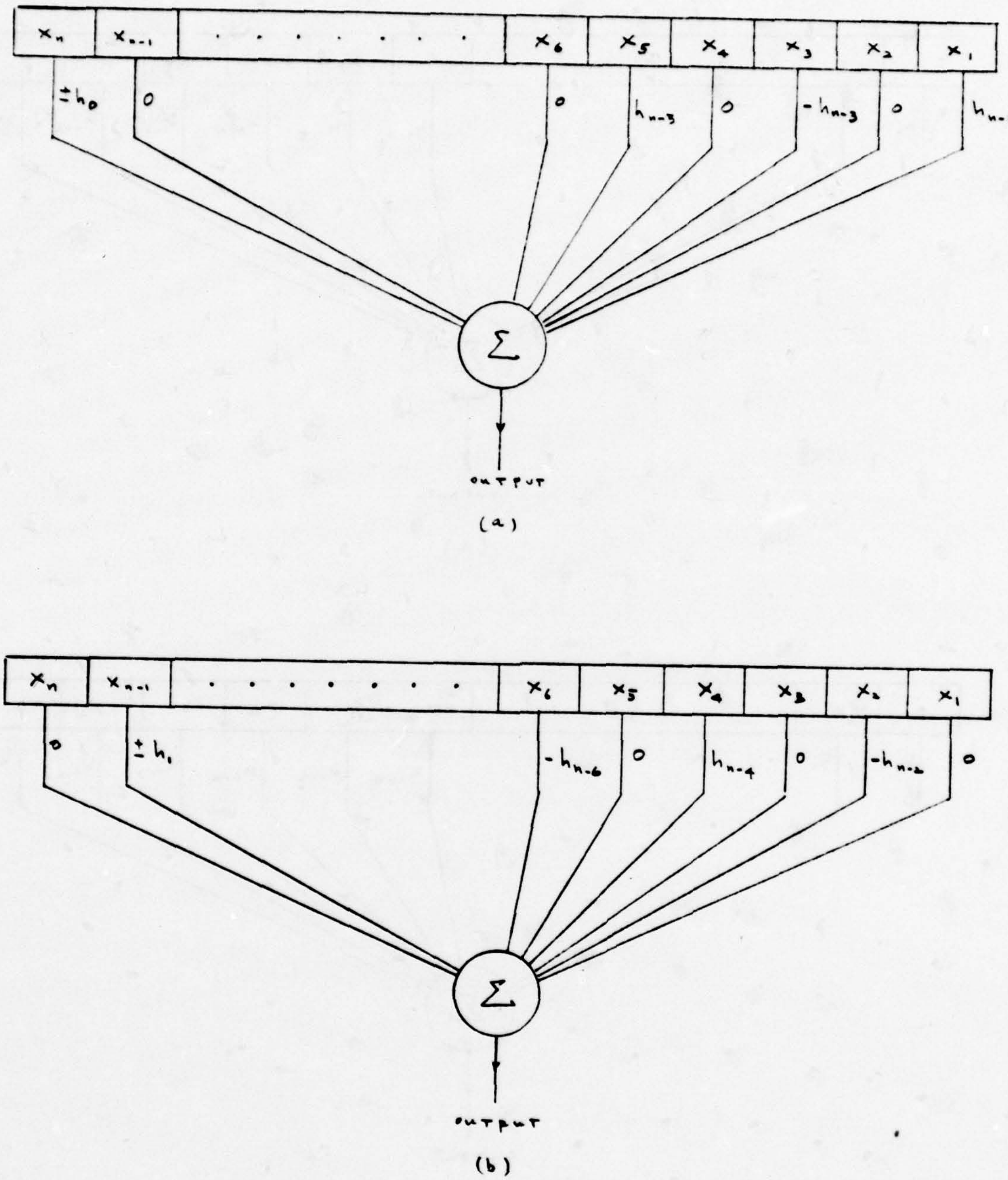
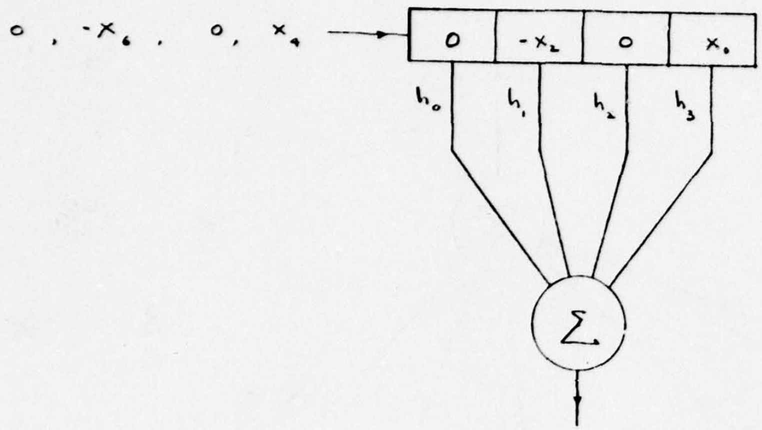
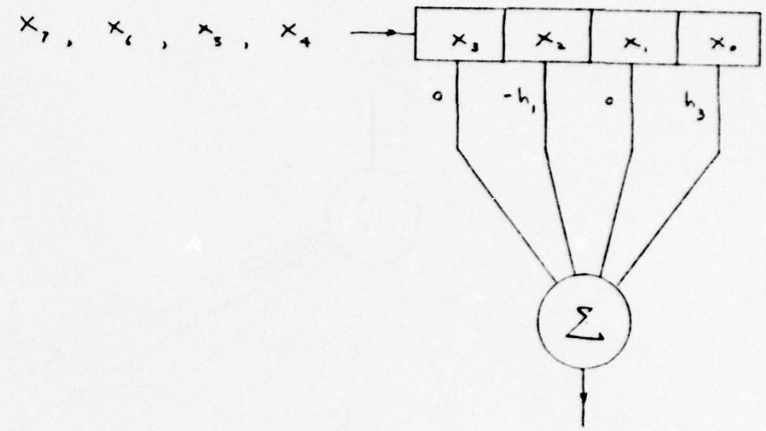


Figure 12 - Alternate Method for Filtering Frequency Translated Signal
(a) Real Channel
(b) Quad Channel



$$
\begin{aligned}
Y_n &= -h_1 x_2 + h_3 x_0 \\
Y_{n+1} &= h_0 x_4 - h_2 x_2 \\
Y_{n+2} &= h_1 x_4 - h_3 x_2 \\
Y_{n+3} &= -h_0 x_6 + h_2 x_4 \\
Y_{n+4} &= -h_1 x_6 + h_3 x_4 \\
&\vdots
\end{aligned}
$$



$$
\begin{aligned}
Y_n &= -h_1 x_2 + h_3 x_0 \\
Y_{n+1} &= -h_1 x_3 + h_3 x_1 \\
Y_{n+2} &= -h_1 x_4 + h_3 x_2 \\
Y_{n+3} &= -h_1 x_5 + h_3 x_3 \\
Y_{n+4} &= -h_1 x_6 + h_3 x_4 \\
&\vdots
\end{aligned}
$$

Figure 13 - Difference Between the Two Computations for N = 4

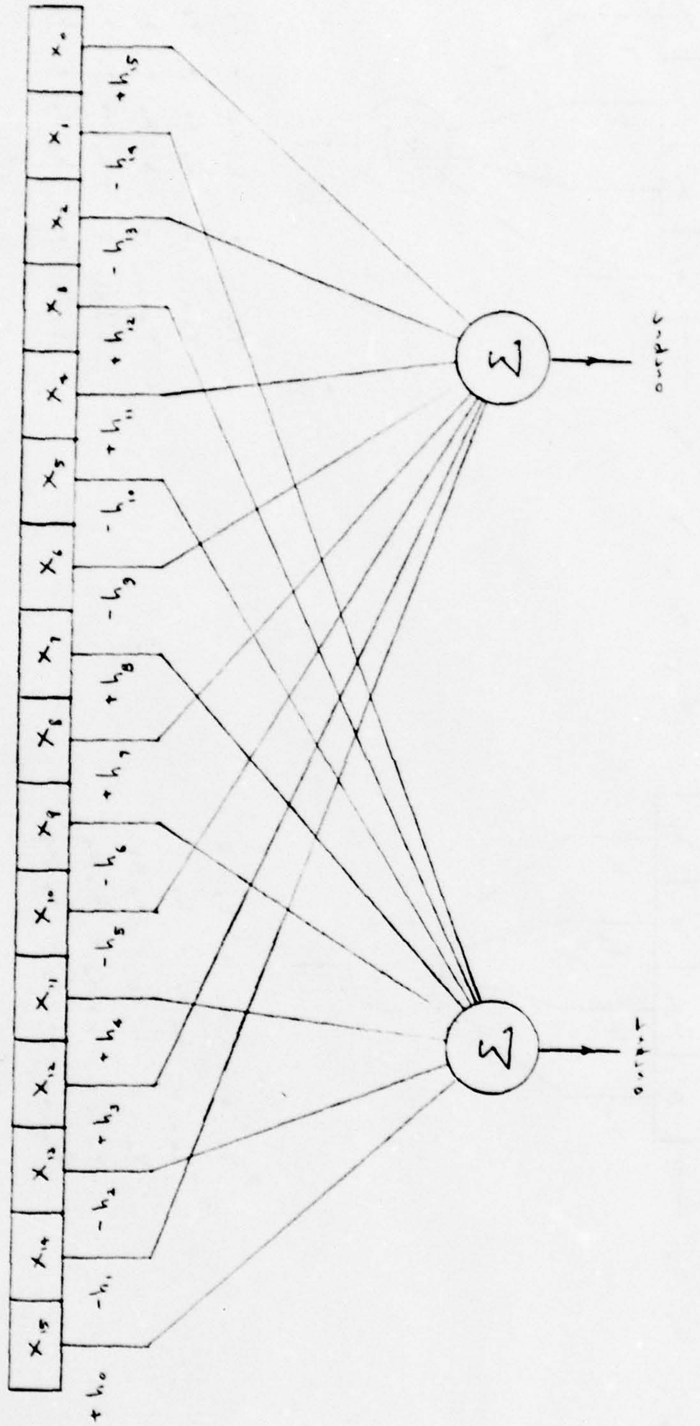


Figure 14 - Dual Channel Filtering of Frequency Translated Signal

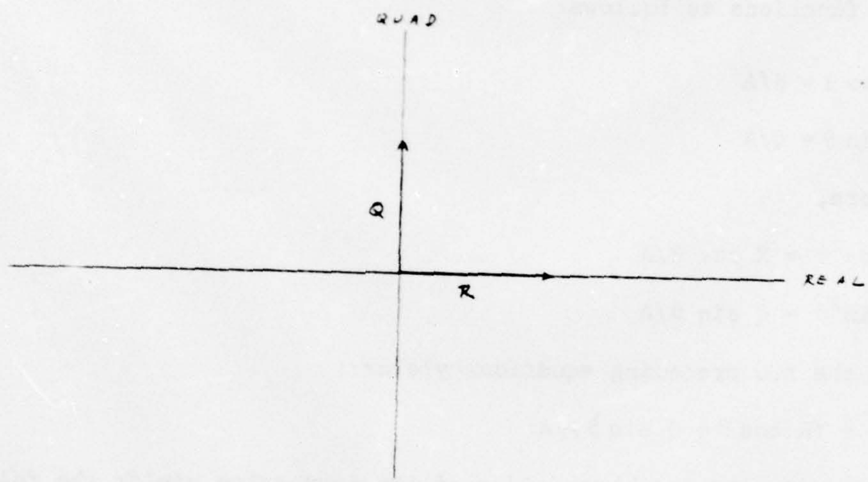


Figure 15 - Real and Quad Components of Frequency Translated Signal

1. $A = \sqrt{R^2 + Q^2}$

2. $A = R \cos\theta + Q \sin\theta$

where A = Amplitude of vector
R = Real component of vector
Q = Quadrature component of vector
θ = Phase angle of vector

The first equation is obtained from Pythagoras Theorem while the second equation is obtained from the trigonometric relationship between the sine and cosine functions as follows:

$\cos\theta = R/A$

$\sin\theta = Q/A$

Therefore,

$\cos^2\theta = R \cos\theta/A$

$\sin^2\theta = Q \sin\theta/A$

Adding the two preceding equations yields:

$1 = (R \cos\theta + Q \sin\theta)/A$

Multiplication by A on both sides of the expression yields the following equation:

$A = R \cos\theta + Q \sin\theta$

Since taking the square root of a number is a complicated process, the second method is chosen. Notice that A is the amplitude of a vector and therefore, the sign is always positive. The spectra at points 1,2,3,4,5 for the AM receive mode is shown in figure 16. The specifications for the 4:1 resampling filter are listed in table 2.

TABLE 2 - DIGITAL FILTER SPECIFICATIONS FOR AM RECEIVE MODE

Filter	Sampling Frequency (KHz)	Passband (KHz)	Normalized Passband	Stopband (KHz)	Normalized Stopband	Attenuation (db)
1	230.4	0-3.3	0-0.01432	53.6-115.2	0.2326-0.5	50

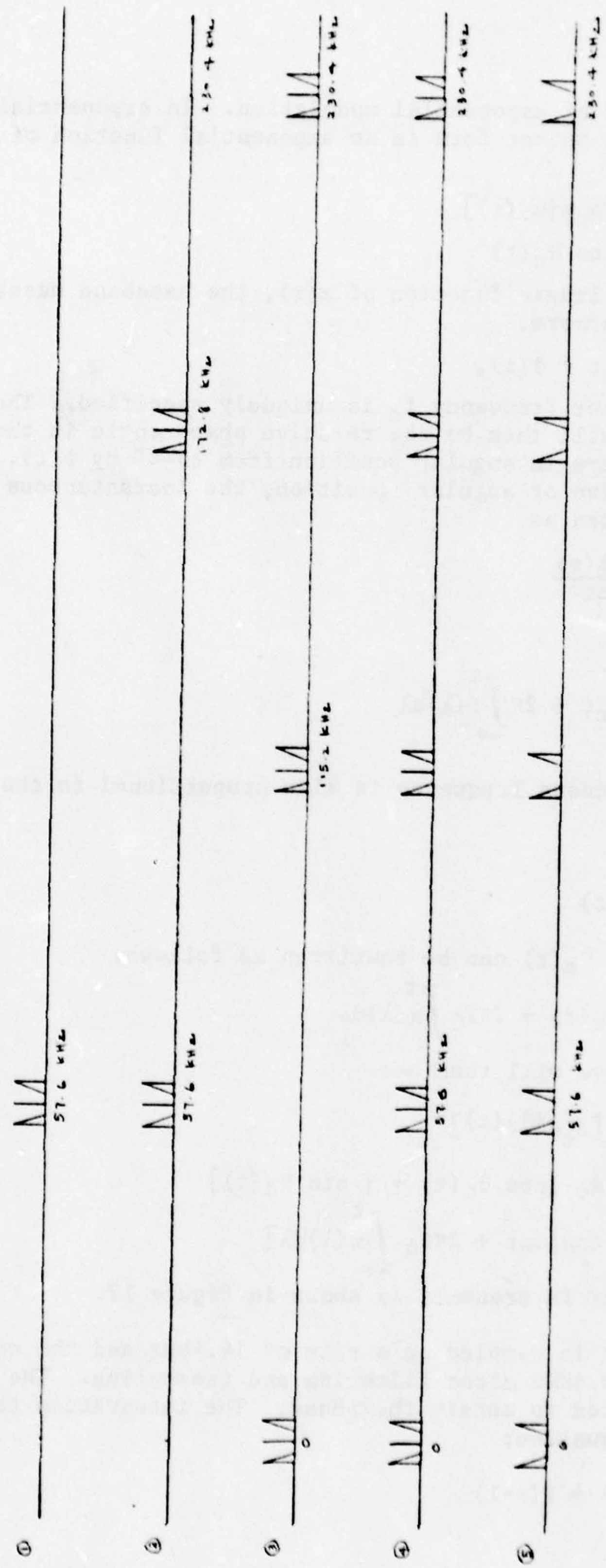


Figure 16 - AM Receive Spectra

3.3 FM Transmit

FM is a form of exponential modulation. In exponential modulation, the modulated wave in phasor form is an exponential function of the message; that is,

$$X_c(t) = \text{Re}\,[A_c e^{j\theta_c(t)}]$$
$$= A_c \cos\theta_c(t)$$

where $\theta_c(t)$ is a linear function of $x(t)$, the baseband message, and A_c is constant. Furthermore,

$$\theta_c(t) = 2\pi f_c t + \phi(t),$$

so that the carrier frequency f_c is uniquely specified. The second term in the equation, $\phi(t)$, will then be the relative phase angle in the sense that the phasor $e^{j\theta_c}$ differs in angular position from $e^{j\omega_c t}$ by $\phi(t)$. Since frequency is the time derivative of angular position, the instantaneous frequency deviation $f(t)$ can be written as

$$f(t) = \frac{1}{2\pi}\frac{d\phi(t)}{dt}$$

Therefore,

$$\theta_c(t) = 2\pi f_c t + 2\pi\int_{-\infty}^{t} f(\lambda)d\lambda$$

But the instantaneous frequency is also proportional to the message.

Therefore,

$$f(t) = f_\Delta X(t)$$

The equation for $\theta_c(t)$ can be rewritten as follows:

$$\theta_c(t) = 2\pi f_c(t) + 2\pi f_\Delta\int_{-\infty}^{t} x(\lambda)d\lambda$$

The modulated wave will then be:

$$X_c(t) = \text{Re}\,[A_c e^{j\theta_c(t)}]$$
$$= \text{Re}\,A_c\,[\cos\theta_c(t) + j\sin\theta_c(t)]$$
$$= A_c\cos[\omega_c t + 2\pi f_\Delta\int_{-\infty}^{t} x(\lambda)d\lambda]$$

The algorithm for FM transmit is shown in figure 17.

Audio input is sampled at a rate of 14.4KHz and the sampling rate is up converted to 230.4KHz after filtering and resampling. The output of the filter is then integrated to obtain the phase. The integration process is given by the following equation:

$$Y(n) = X(n) + Y(n-1)$$

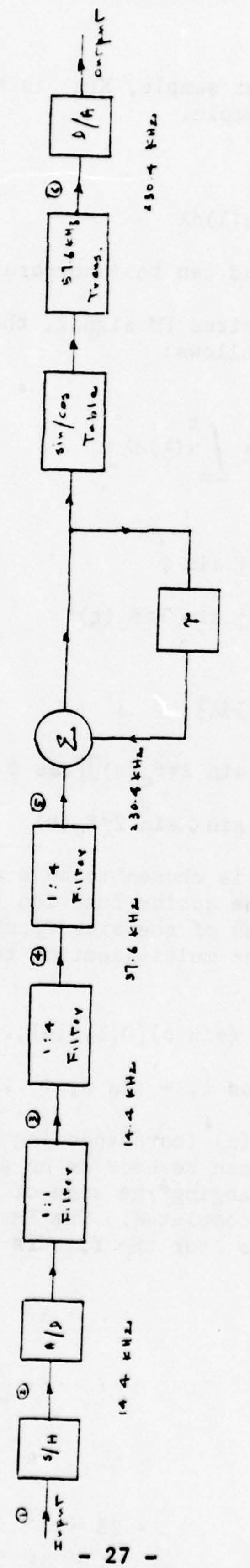


Figure 17 - FM Transmit Algorithm

Where Y(n) is the present output sample, X(n) is the present input sample, and Y(n-1) is the delayed output sample.

Recall that

$$\theta_c(t) = 2\pi f_c(t) + 2\pi f_\Delta \int_{-\infty}^{t} x(\lambda)d\lambda$$

The product $2\pi f_\Delta$ is a scaler and can be incorporated into the filter coefficients.

In order to obtain the desired FM signal, the real part has to be extracted from the function $e^{j\theta_c(t)}$ as follows:

$$FM = \mathrm{Re}\,[e^{j\theta_c(t)}]$$

$$= \mathrm{Re}\,[e^{j2\pi f_c(t)}][e^{j2\pi f_\Delta \int_{-\infty}^{t} x(\lambda)d\lambda}]$$

Let $e^{j2\pi f_\Delta \int_{-\infty}^{t} x(\lambda)d\lambda} = e^{j\phi}$

$$= \cos\phi + j\sin\phi$$

and $e^{j2\pi f_c(t)} = \cos 2\pi f_c(t) + j \sin 2\pi f_c(t)$

Therefore,

$$\mathrm{Re}\,[e^{j2\pi f_c(t)} e^{j2\pi f_\Delta \int_{-\infty}^{t} x(\lambda)d\lambda}]$$

$$= \mathrm{Re}\,[\cos 2\pi f_c(t) + j\sin 2\pi f_c(t)][\cos\phi + j\sin\phi]$$

$$= \cos\phi\,\cos 2\pi f_c(t) - \sin\phi\,\sin 2\pi f_c(t)$$

If the carrier frequency is chosen to be a quarter of the input sampling rate, then, four samples of the cosine function (which can be chosen to be 1,0,-1,0,....) and four samples of the sine function (which can be chosen to be 0,1,0,-1,.....) can be used for multiplication to produce the desired output as follows:

$$(\cos\phi)[1,0,-1,0,.....] - (\sin\phi)[0,1,0,-1,....]$$

$$= \cos\phi,\ -\sin\phi,\ -\cos\phi,\ +\sin\phi,\ + \ldots\ldots\ldots\ldots$$

Therefore, the output, Y(n) (corresponding to ϕ in the above expression), from the integration process can be used as an address to obtain the sine and cosine values of Y(n). By changing the sign of every other pair of samples, the frequency translation is completed. The FM transmit spectra is shown in figure 18. The specifications for the filters are listed in table 3.

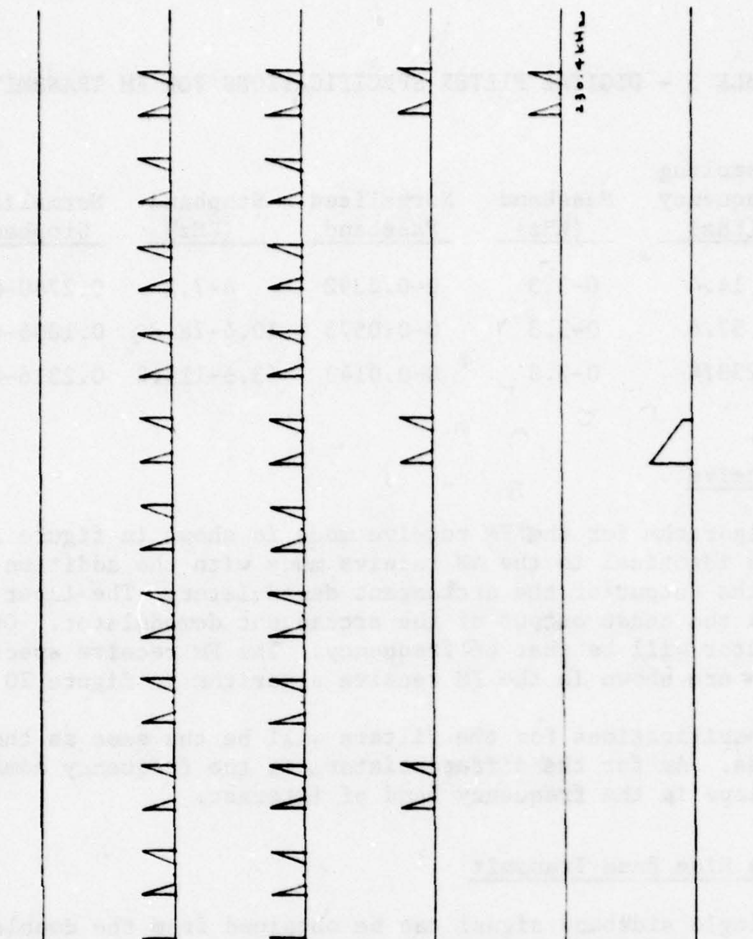


Figure 18 - FM Transmit Spectra

TABLE 3 - DIGITAL FILTER SPECIFICATIONS FOR FM TRANSMIT MODE

Filter	Sampling Frequency (KHz)	Passband (KHz)	Normalized Passband	Stopband (KHz)	Normalized Stopband	Attenuation (db)
1	14.4	0-3.3	0-0.2292	4-7.2	0.2780-0.5	50
2	57.6	0-3.3	0-0.0573	10.4-28.8	0.1806-0.5	50
3	230.4	0-3.3	0-0.0143	53.6-115.2	0.2326-0.5	50

3.4 FM Receive

The algorithm for the FM receive mode is shown in figure 19. As can be seen, it is identical to the AM receive mode with the addition of a differentiator at the output of the arctangent demodulator. The input to the differentiator is the phase output of the arctangent demodulator. Output of the differentiator will be that of frequency. The FM receive spectra at points 1,2,3,4,5,6 are shown in the FM receive algorithm in figure 20.

The specifications for the filters will be the same as those for the AM receive mode. As for the differentiator, in the frequency domain, it will have a linear slope in the frequency band of interest.

3.5 Single Side Band Transmit

The single sideband signal can be obtained from the double sideband signal by suppressing the carrier and filtering out one of the sidebands (upper or lower sideband). The suppression of the carrier can be easily done by adjusting the constant added to the output waveform before frequency translation. However, since the two sidebands are only several hundred hertz apart, the filter has to be extremely selective and represents a relatively large load on the multipliers. Consequently, this approach is abandoned.

The algorithm for single sideband transmit is shown on figure 21. Audio signals ranging from 300 to 3300 Hz are sampled at 7.2KHz and frequency translated 1800 Hz to DC frequency by multiplying the input by $e^{\pm j2\pi(1800)}$. The positive sign indicates a frequency up shift to the right of the DC frequency while a negative sign indicates a down shift to the left of the frequency origin. Due to the frequency translation process, the signal is split into the real and quadrature components. A low pass selectivity filter will then eliminate one of the sidebands depending on which way the frequency translation took place. This can be seen more clearly on the spectra diagram for single sideband transmit on figure 22 in which the spectra at points 1,2, 3,4,5,6,7,8 on the SSB transmit algorithm are shown.

The signal then goes through three interpolating filters to convert the sampling rate to 230.4KHz. By multiplying the signal with 4 samples of

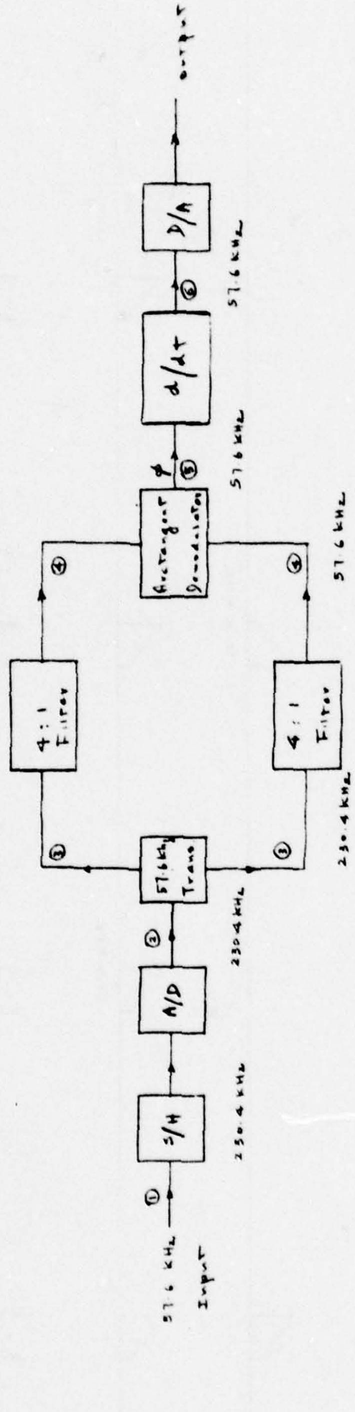


Figure 19 - FM Receive Algorithm

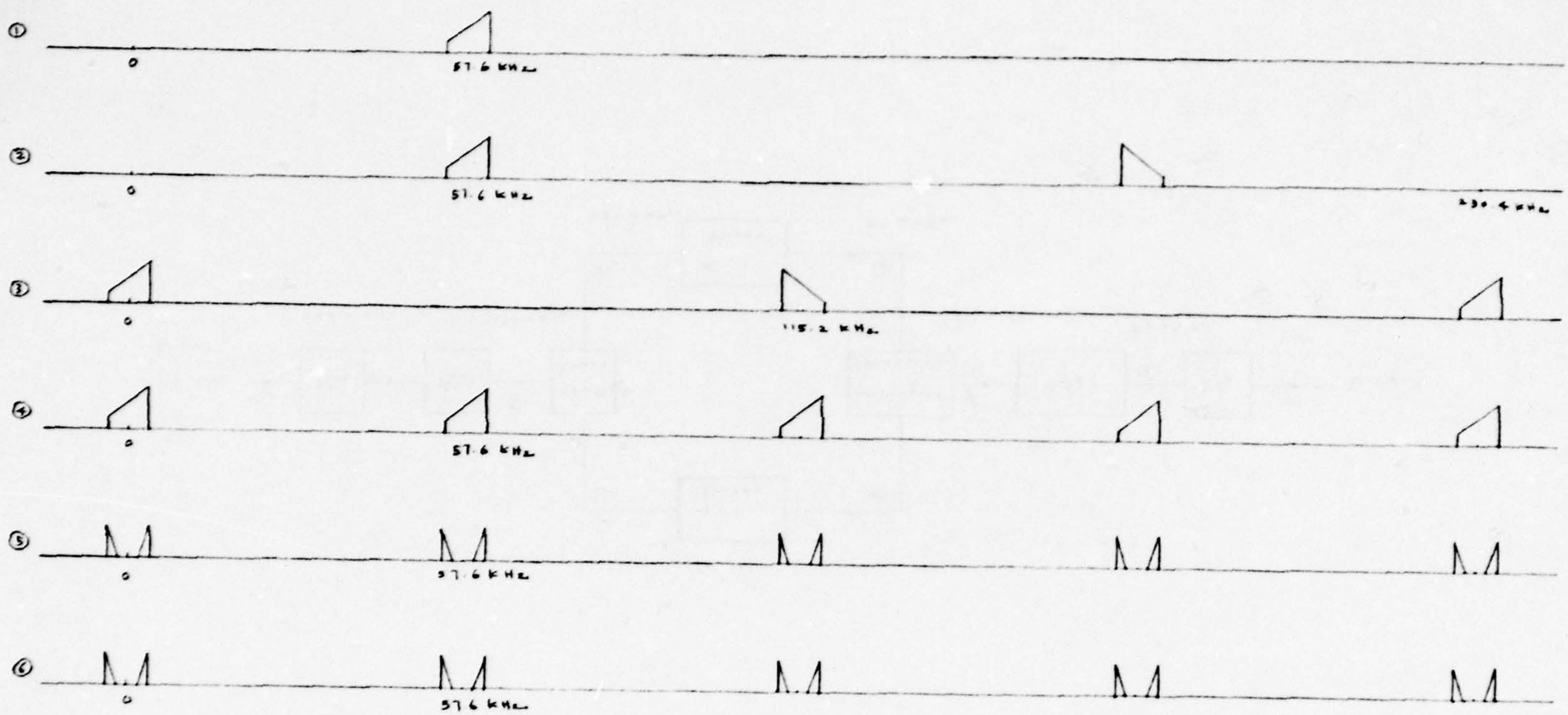


Figure 20 - FM Receive Spectra

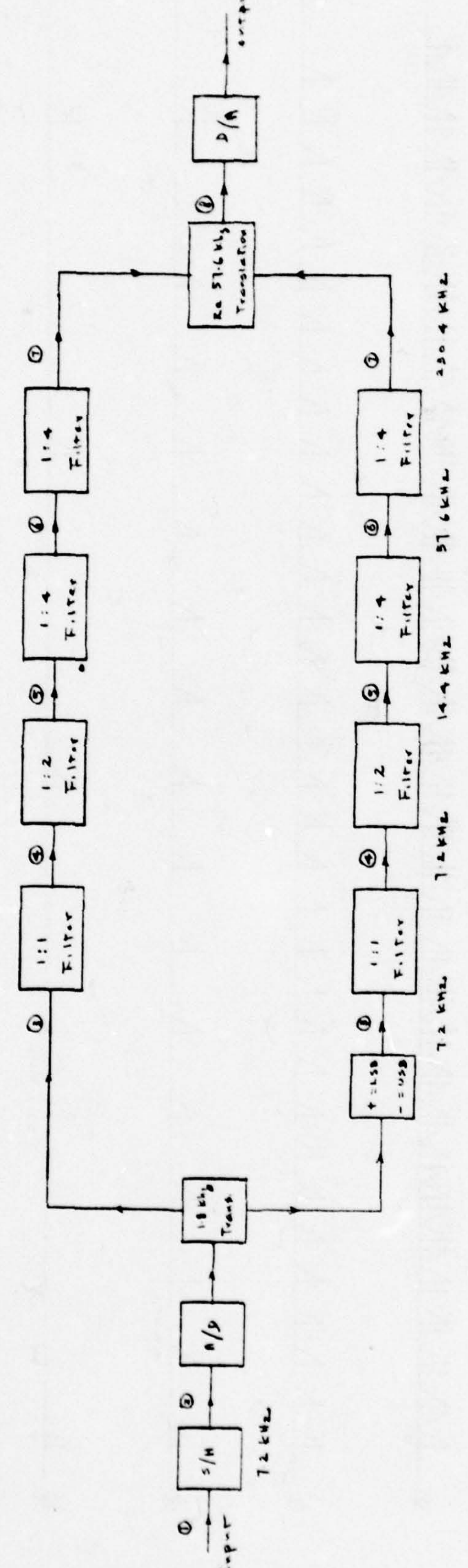


Figure 21 - SSB Transmit Algorithm

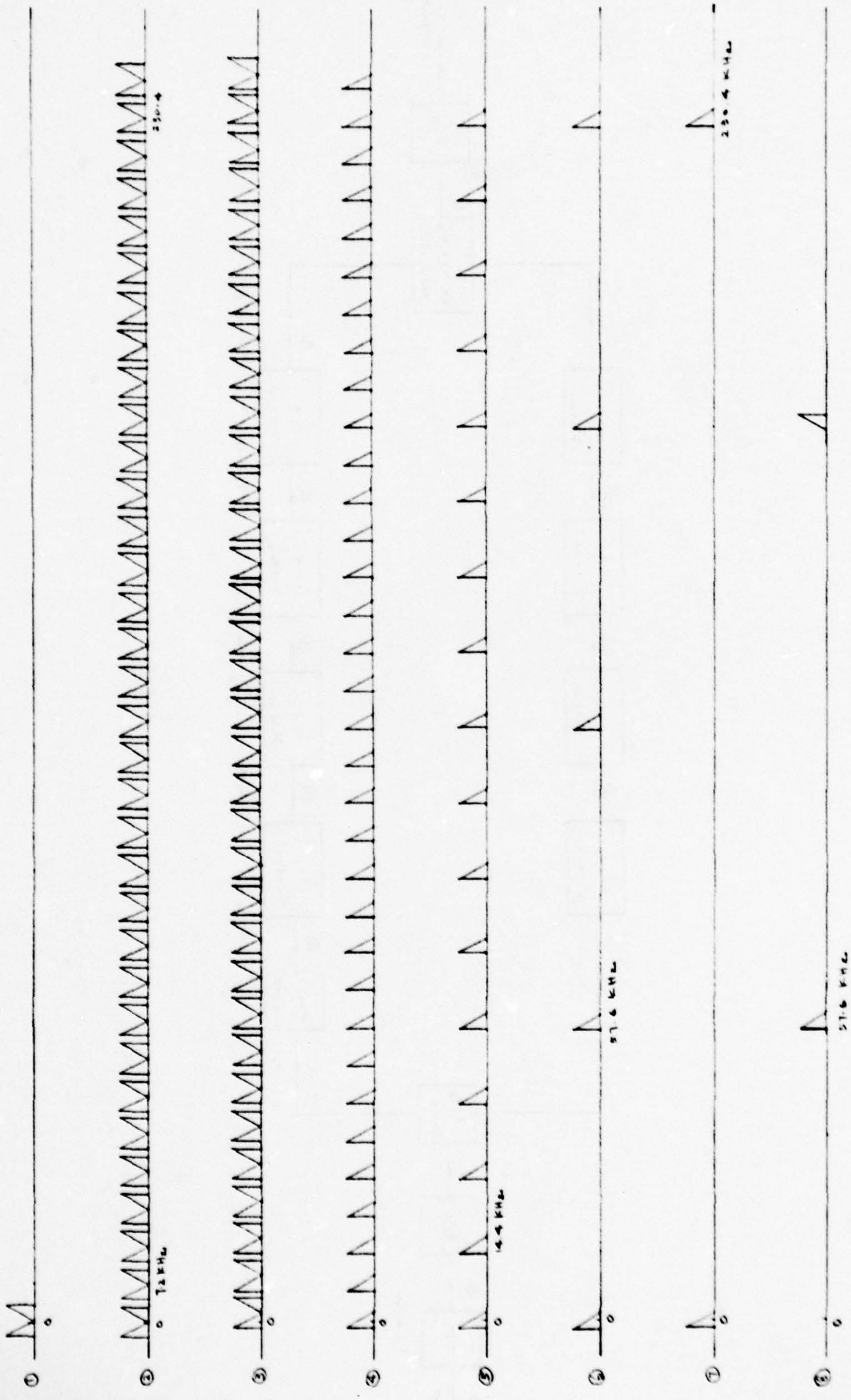


Figure 22 - SSB Transmit Spectra

$e^{\pm j(2\pi 57600)t}$, a frequency translation of 57.6 KHz is accomplished. The real part of this signal is the desired single sideband waveform. To determine whether upper sideband or lower sideband is used, one of the frequency translation processes is chosen to have a fixed shift (up or down). If it is assumed that the 57.6 KHz translation has an up shift, a down shift for the 1.8 KHz translation will produce upper sideband while an up shift of 1.8 KHz will produce lower sideband. Again, the samples for frequency translation are chosen to be the two sets of samples (1,0,-1,0,....; 0,1,0,-1,....) for the sine and cosine functions so that multiplication is reduced to a minimum. For the 1.8 KHz translation, we have the following:

$$X(n)[\cos 2\pi 1800t \pm j \sin 2\pi 1800t]$$

$$= X(n)[(1,0,-1,0,.....) \pm j\ (0,1,0,-1,......)]$$

which is complex.

The case for the 57.6 KHz translation is different because the input to the frequency translator is a complex number. Let the input complex numbers be represented by $x + jy$. The output of the frequency translator is then given by the following:

$$(x + jy)[\cos (2\pi 57600)t + j \sin (2\pi 57600)t]$$

Taking the real parts yield

$$x \cos (2\pi 57600)t - y \sin (2\pi 57600)t$$

Therefore, only every other sample in each channel is needed to produce the output if four samples of sine and cosine are used.

Notice that the final output is not the same as that of AM double sideband with the carrier and one sideband filtered out. This can be seen clearly in figure 23. The difference being that, in using this method of modulation, the transmitted signal is centered on 57.6 KHz, while filtering the carrier and one sideband of the double sideband AM signal produce a signal that is offset from the carrier frequency of 57.6 KHz. If a tone is used as an example, AM modulation will produce the spectrum shown in figure 23(a). SSB modulation which will be used for the NBSCU will produce the spectrum in figure 23(b). By filtering the AM signal, the resulting spectrum is shown in figure 23(c). The difference is obvious. The filter specifications are listed in table 4.

TABLE 4 - DIGITAL FILTER SPECIFICATIONS FOR SSB TRANSMIT MODE

Filter	Sampling Frequency (KHz)	Passband (KHz)	Normalized Passband	Stopband (KHz)	Normalized Stopband	Attenuation (db)
1	7.2	0-1.65	0-0.2292	2-3.6	0.2780-0.5	50
2	14.4	0-1.65	0-0.1146	5.2-7.2	0.3610-0.5	50
3	57.6	0-1.65	0-0.0287	12.4-28.8	0.2153-0.5	50
4	230.4	0-1.65	0-0.0072	55.6-115.2	0.2410-0.5	50

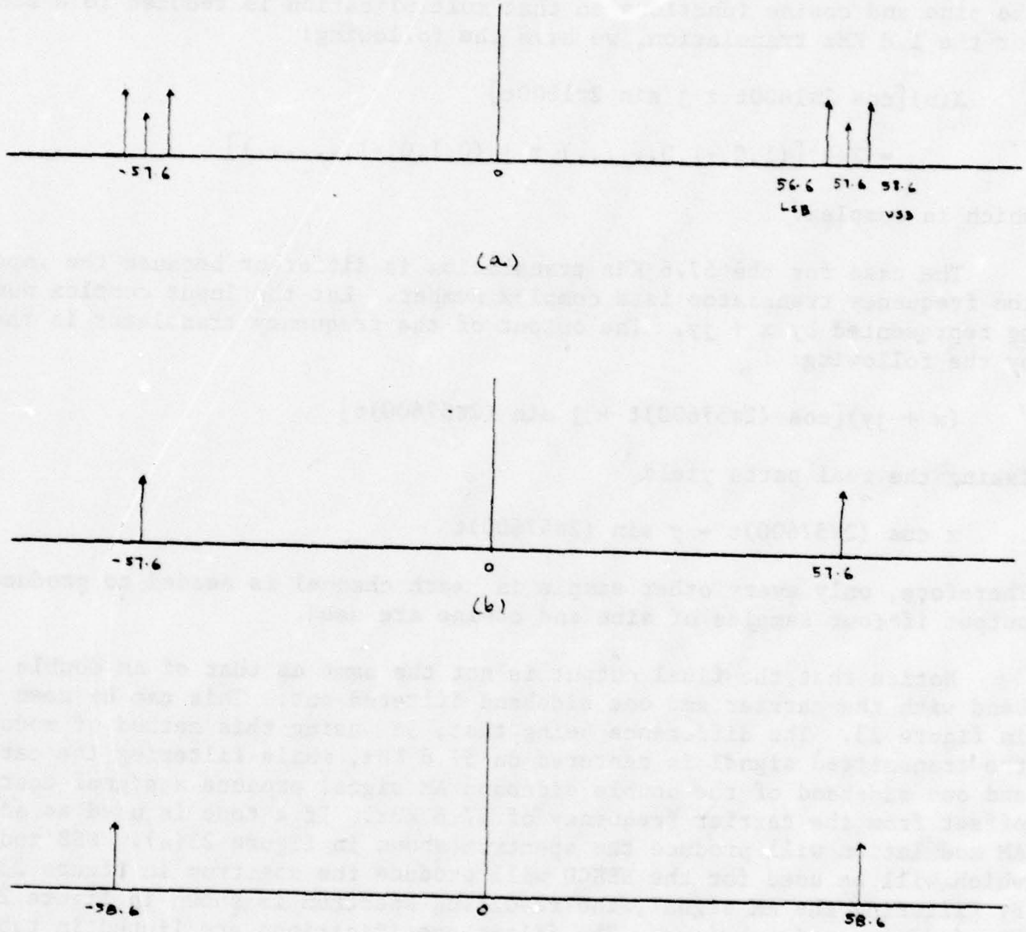


Figure 23 - (a) DSB Spectra
(b) SSB Spectra
(c) SSB Spectra Obtained by Filtering DSB Signal

3.6 Single Sideband Receive

The single sideband receive mode is processed similarly to that of single sideband transmit mode except in the reverse order. The algorithm is shown in figure 24 and the spectra at points 1,2,3,4,5,6,7 are shown in figure 25.

If the 57.6 KHz frequency translation is an up shift operation, the result would be lower sideband. If it is a down shift operation, the result would be upper sideband provided that the 1.8 KHz frequency translation is an up shift operation in both cases. The filter specifications are shown in table 5.

TABLE 5 - DIGITAL FILTER SPECIFICATIONS FOR SSB RECEIVE MODE

Filter	Sampling Frequency (KHz)	Passband (KHz)	Normalized Passband	Stopband (KHz)	Normalized Stopband	Attenuation (db)
1	230.4	0-1.65	0-0.0072	55.6-115.2	0.2410-0.5	50
2	57.6	0-1.65	0-0.0287	12.4-28.8	0.2153-015	50
3	14.4	0-1.65	0-0.1146	5.2-3.6	0.3610-0.5	50

4. CRITICAL FUNCTION DESIGNS

As can be seen from all the algorithms, the most important tasks the NBSCU has to perform are those of filtering and arctangent demodulation. These are discussed in the following paragraphs.

4.1 Digital Filter Design

The design of the digital filter is the most important task for the NBSCU as it is used in all the mod/demod algorithms. In general, a digital filter performs the equivalent of an analog filter in that it is frequency selective. However, the same digital filter can be made to be a differentiator or a Hilbert transformer by varying a set of tap weights for the filter. Unlike analog filters, which can be designed to pass certain frequency bands from DC to infinity, digital filters do not enjoy this luxury. All frequencies are normalized to the sampling frequency F_s, which is taken to be 1 (or π) in the digital domain and frequencies normalized to F_s are known as digital frequencies. Due to spectral folding in a sampled system, the input to a digital filter must not have any significant spectral components beyond $F_s/2$. Otherwise, the original signal cannot be recovered without distortion (the sampling theorem). Therefore, the analog input to an A/D converter must be band limited by an anti-aliasing filter.

There are, in general, two classes of digital filters:

1. Infinite impulse response (IIR) filters in which the impulse response of the filter is infinitely long, and

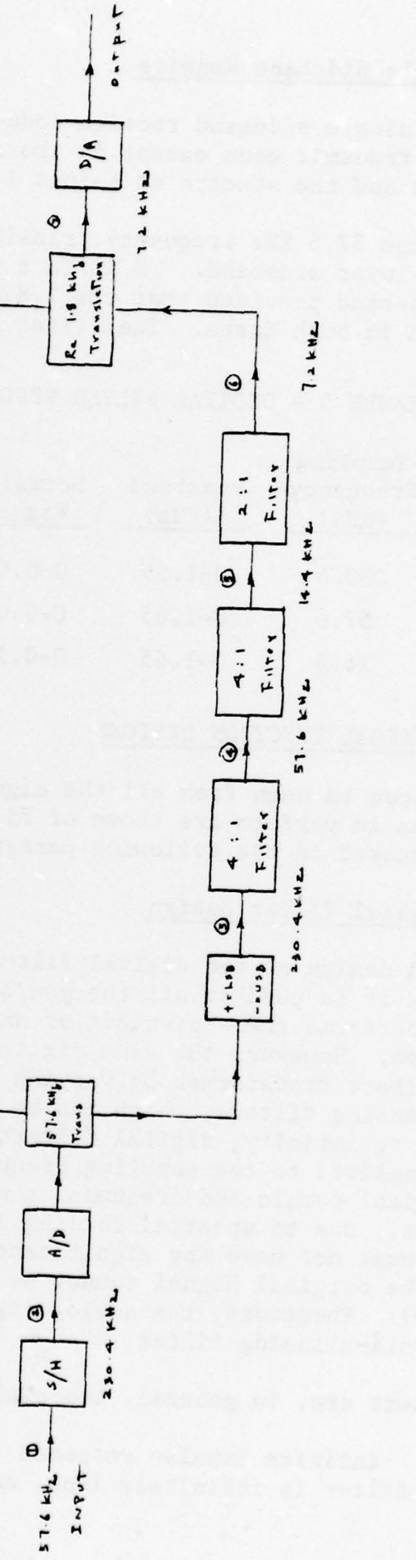


Figure 24 - SSB Receive Algorithm

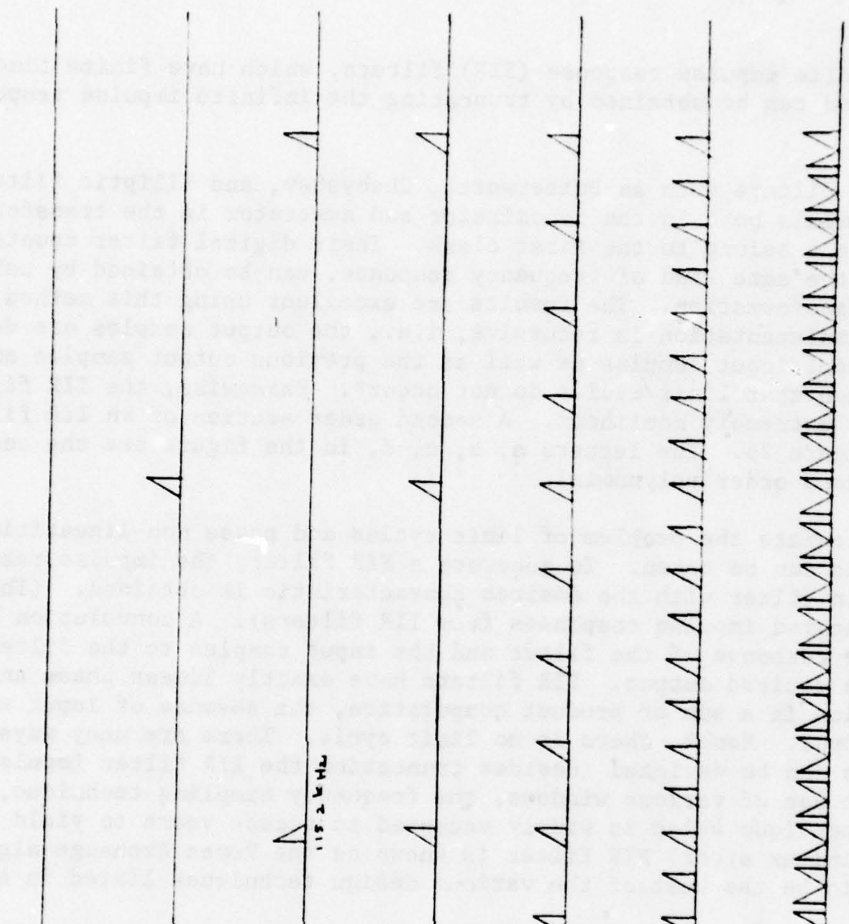


Figure 25 - SSB Receive Spectra

2. Finite impulse response (FIR) filters, which have finite time impulse responses and can be obtained by truncating the infinite impulse response samples.

Analog filters such as Butterworth, Chebyshev, and Elliptic filters which have polynomials both in the denominator and numerator in the transfer function in the S plane belong to the first class. Their digital filter counterparts which have the same kind of frequency response, can be obtained by using a bilinear transformation. The results are excellent using this method. However, the implementation is recursive, i.e., the output samples are dependent on the present input samples as well as the previous output samples and care must be taken that limit cycles do not occur*. Phasewise, the IIR filter response is extremely nonlinear. A second order section of an IIR filter is shown in figure 26. The letters a, b, c, d, in the figure are the coefficients for the second order polynomial.

To alleviate the problem of limit cycles and phase non-linearities, the FIR approach can be taken. To generate a FIR filter, the impulse response of a certain filter with the desired characteristic is obtained. (These may be the truncated impulse responses from IIR filters). A convolution between the impulse response of the filter and the input samples to the filter will produce the desired output. FIR filters have exactly linear phase and since a convolution is a sum of product computation, the absence of input will produce no output. Hence, there is no limit cycle. There are many ways in which FIR filters can be designed (besides truncating the IIR filter impulse response) such as the use of various windows, the frequency sampling technique, etc. A design technique which is widely accepted in recent years to yield an optimum (minimax error) FIR filter is known as the Remez Exchange algorithm. It proved to be the best of the various design techniques listed in Appendix B.

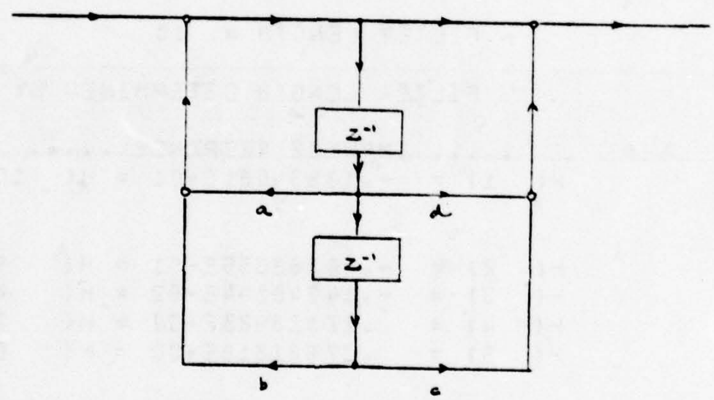
In general, IIR filters are more efficient than the FIR filters when the order of the filter is large. However, since FIR filters have exactly linear phase, i.e., there is no group delay distortion, FIR filters are chosen for the NBSCU.

The Remez Exchange algorithm is implemented using the CDC 6600 to obtain the filter coefficients which are listed in tables 6 through 11.

The frequency response using these coefficients are then obtained using an interpolation formula implemented on a Tektronix 4051 terminal. The output samples are plotted and shown in figure 27 through figure 32 (with 2 db/division vertical scale). The response with the coefficients rounded off to 12 bits are shown in figure 33 through figure 38. Notice that in all cases, the attenuation in the stopband is at least 50 db down with respect to passband amplitude.

---

*Limit cycle is an effect due to the dependence on previous output samples. The filter produces an output sample with no input.



$$H(z) = \frac{1 + a z^{-1} + b z^{-2}}{1 + c z^{-1} + d z^{-2}}$$

Figure 26 - Second Order Section of an IIR Filter

TABLE 6 - IMPULSE RESPONSE FOR 10 STAGE 1:2 DIGITAL FILTER

```
FINITE IMPULSE RESPONSE (FIR)
LINEAR PHASE DIGITAL FILTER DESIGN
REMEZ EXCHANGE ALGORITHM

BANDPASS FILTER

FILTER LENGTH =  10

FILTER LENGTH DETERMINED BY APPROXIMATION

   ***** IMPULSE RESPONSE *****
H(  1) =  -.13634021E-01 = H(  10)

H(  2) =  -.40368055E-01 = H(   9)
H(  3) =  -.14243194E-02 = H(   8)
H(  4) =   .17818923E+00 = H(   7)
H(  5) =   .37921318E+00 = H(   6)

                    BAND  1       BAND  2       BAND
LOWER BAND EDGE     .000000000    .361000000
UPPER BAND EDGE     .114600000    .500000000
DESIRED VALUE      1.000000000    .000000000
WEIGHTING          1.000000000  100.000000000
DEVIATION           .028793100    .000287931
DEVIATION IN DB     .500325669  -70.814231551

EXTREMAL FREQUENCIES
  .0687500   .1146000   .3610000   .3735000   .4172500
  .4735000
```

TABLE 7 - IMPULSE RESPONSE FOR 10 STAGE 1:4 DIGITAL FILTER

```
             FINITE IMPULSE RESPONSE (FIR)
             LINEAR PHASE DIGITAL FILTER DESIGN
             REMEZ EXCHANGE ALGORITHM

             BANDPASS FILTER

             FILTER LENGTH =  10

             FILTER LENGTH DETERMINED BY APPROXIMATION

               ***** IMPULSE RESPONSE *****
        H(  1) =    .82334641E-02 = H(  10)

        H(  2) =    .34954963E-01 = H(   9)
        H(  3) =    .84412599E-01 = H(   8)
        H(  4) =    .14330758E+00 = H(   7)
        H(  5) =    .18419784E+00 = H(   6)

                         BAND  1          BAND  2         BAND
LOWER BAND EDGE       .000000000       .241000000
UPPER BAND EDGE       .007200000       .500000000
DESIRED VALUE        1.000000000       .000000000
WEIGHTING            1.000000000    100.000000000
DEVIATION             .092834110       .000928341
DEVIATION IN DB      1.617350604     -60.645848397

EXTREMAL FREQUENCIES
   .0072000    .2410000    .2597500    .3097500    .3847500
   .4597500
```

TABLE 8 - IMPULSE RESPONSE FOR 25 STAGE 1:4 DIGITAL FILTER

```
FINITE IMPULSE RESPONSE (FIR)
LINEAR PHASE DIGITAL FILTER DESIGN
REMEZ EXCHANGE ALGORITHM

BANDPASS FILTER

FILTER LENGTH =  25

..... IMPULSE RESPONSE .....
H(  1) =   .21100766E-03 = H(  25)
H(  2) =  -.39009108E-03 = H(  24)
H(  3) =  -.28928785E-02 = H(  23)
H(  4) =  -.83599938E-02 = H(  22)
H(  5) =  -.16027747E-01 = H(  21)
H(  6) =  -.22021152E-01 = H(  20)
H(  7) =  -.19541314E-01 = H(  19)
H(  8) =  -.13701737E-02 = H(  18)
H(  9) =   .35973699E-01 = H(  17)
H( 10) =   .88589167E-01 = H(  16)
H( 11) =   .14458815E+00 = H(  15)
H( 12) =   .18771701E+00 = H(  14)
H( 13) =   .20393880E+00 = H(  13)
```

	BAND 1	BAND 2	BAND
LOWER BAND EDGE	.000000000	.180600000	
UPPER BAND EDGE	.057300000	.500000000	
DESIRED VALUE	1.000000000	.000000000	
WEIGHTING	1.000000000	100.000000000	
DEVIATION	.023109854	.000231099	
DEVIATION IN DB	.401530770	-72.724056063	

EXTREMAL FREQUENCIES

.0000000	.0384615	.0573000	.1806000	.1878115
.2070423	.2358885	.2695423	.3056000	.3440615
.3825231	.4209846	.4594462	.5000000	

TABLE 9 - IMPULSE RESPONSE FOR 55 STAGE 1:1 DIGITAL FILTER

FINITE IMPULSE RESPONSE (FIR)
LINEAR PHASE DIGITAL FILTER DESIGN
REMEZ EXCHANGE ALGORITHM

BANDPASS FILTER

FILTER LENGTH = 55

..... IMPULSE RESPONSE .....

H(1)	=	.96952259E-03	=	H(55)
H(2)	=	.10626658E-02	=	H(54)
H(3)	=	-.27980042E-02	=	H(53)
H(4)	=	-.80329778E-02	=	H(52)
H(5)	=	-.62975140E-02	=	H(51)
H(6)	=	.22990183E-02	=	H(50)
H(7)	=	.47051969E-02	=	H(49)
H(8)	=	-.36383487E-02	=	H(48)
H(9)	=	-.67920770E-02	=	H(47)
H(10)	=	.39219972E-02	=	H(46)
H(11)	=	.90704774E-02	=	H(45)
H(12)	=	-.44368474E-02	=	H(44)
H(13)	=	-.12204033E-01	=	H(43)
H(14)	=	.48794071E-02	=	H(42)
H(15)	=	.16271823E-01	=	H(41)
H(16)	=	-.52916459E-02	=	H(40)
H(17)	=	-.21694685E-01	=	H(39)
H(18)	=	.56489810E-02	=	H(38)
H(19)	=	.29229548E-01	=	H(37)
H(20)	=	-.59589844E-02	=	H(36)
H(21)	=	-.40564246E-01	=	H(35)
H(22)	=	.62099160E-02	=	H(34)
H(23)	=	.60088016E-01	=	H(33)
H(24)	=	-.63844805E-02	=	H(32)
H(25)	=	-.10392688E+00	=	H(31)
H(26)	=	.64950402E-02	=	H(30)
H(27)	=	.31757675E+00	=	H(29)
H(28)	=	.49346309E+00	=	H(28)

	BAND 1	BAND 2	BAND
LOWER BAND EDGE	.000000000	.278000000	
UPPER BAND EDGE	.229200000	.500000000	
DESIRED VALUE	1.000000000	.000000000	
WEIGHTING	1.000000000	100.000000000	
DEVIATION	.025721650	.000257216	
DEVIATION IN DB	.446929404	-71.794023574	

EXTREMAL FREQUENCIES

.0000000	.0212054	.0424107	.0636161	.0848214
.1049107	.1250000	.1462054	.1662946	.1852679
.2053571	.2209821	.2292000	.2780000	.2813482
.2891607	.3014375	.3148304	.3304554	.3460804
.3617054	.3784464	.3963036	.4130446	.4297857
.4476429	.4655000	.4822411	.5000000	

TABLE 10 - IMPULSE RESPONSE FOR 16 STAGE 1:4 (SSB) DIGITAL FILTER

FINITE IMPULSE RESPONSE (FIR)
LINEAR PHASE DIGITAL FILTER DESIGN
REMEZ EXCHANGE ALGORITHM

BANDPASS FILTER

FILTER LENGTH = 16

..... IMPULSE RESPONSE .....

H(1) =	-.92734092E-03	= H(	16)
H(2) =	-.12691651E-02	= H(	15)
H(3) =	.33726088E-02	= H(	14)
H(4) =	.20226966E-01	= H(	13)
H(5) =	.54508616E-01	= H(	12)
H(6) =	.10325178E+00	= H(	11)
H(7) =	.15292869E+00	= H(	10)
H(8) =	.18463465E+00	= H(	9)

	BAND 1	BAND 2	BAND
LOWER BAND EDGE	.000000000	.215300000	
UPPER BAND EDGE	.028700000	.500000000	
DESIRED VALUE	1.000000000	.000000000	
WEIGHTING	1.000000000	100.000000000	
DEVIATION	.033453619	.000334536	
DEVIATION IN DB	.581365829	-69.511137847	

EXTREMAL FREQUENCIES

.0000000	.0287000	.2153000	.2270188	.2621750
.3051437	.3598312	.4145187	.4692062	

TABLE 11 - IMPULSE RESPONSE FOR 16 STAGE 1:4 DIGITAL FILTER

```
                 FINITE IMPULSE RESPONSE (FIR)
                 LINEAR PHASE DIGITAL FILTER DESIGN
                 REMEZ EXCHANGE ALGORITHM

                 BANDPASS FILTER

                 FILTER LENGTH =  16

              ..... IMPULSE RESPONSE .....
           H(  1) =  -.27836378E-03 = H(  16)
           H(  2) =   .19176790E-03 = H(  15)
           H(  3) =   .51380971E-02 = H(  14)
           H(  4) =   .21038964E-01 = H(  13)
           H(  5) =   .53192579E-01 = H(  12)

           H(  6) =   .99517446E-01 = H(  11)
           H(  7) =   .14739916E+00 = H(  10)
           H(  8) =   .17825397E+00 = H(   9)

                    BAND  1          BAND  2          BAND
LOWER BAND EDGE    .000000000       .232600000
UPPER BAND EDGE    .014300000       .500000000
DESIRED VALUE     1.000000000       .000000000
WEIGHTING         1.000000000    100.000000000
DEVIATION          .008907236       .000089072
DEVIATION IN DB    .154738637    -81.005140433

EXTREMAL FREQUENCIES
    .0000000    .0143000    .2326000    .2443187    .2716625
    .3146312    .3615062    .4161937    .4708812
```

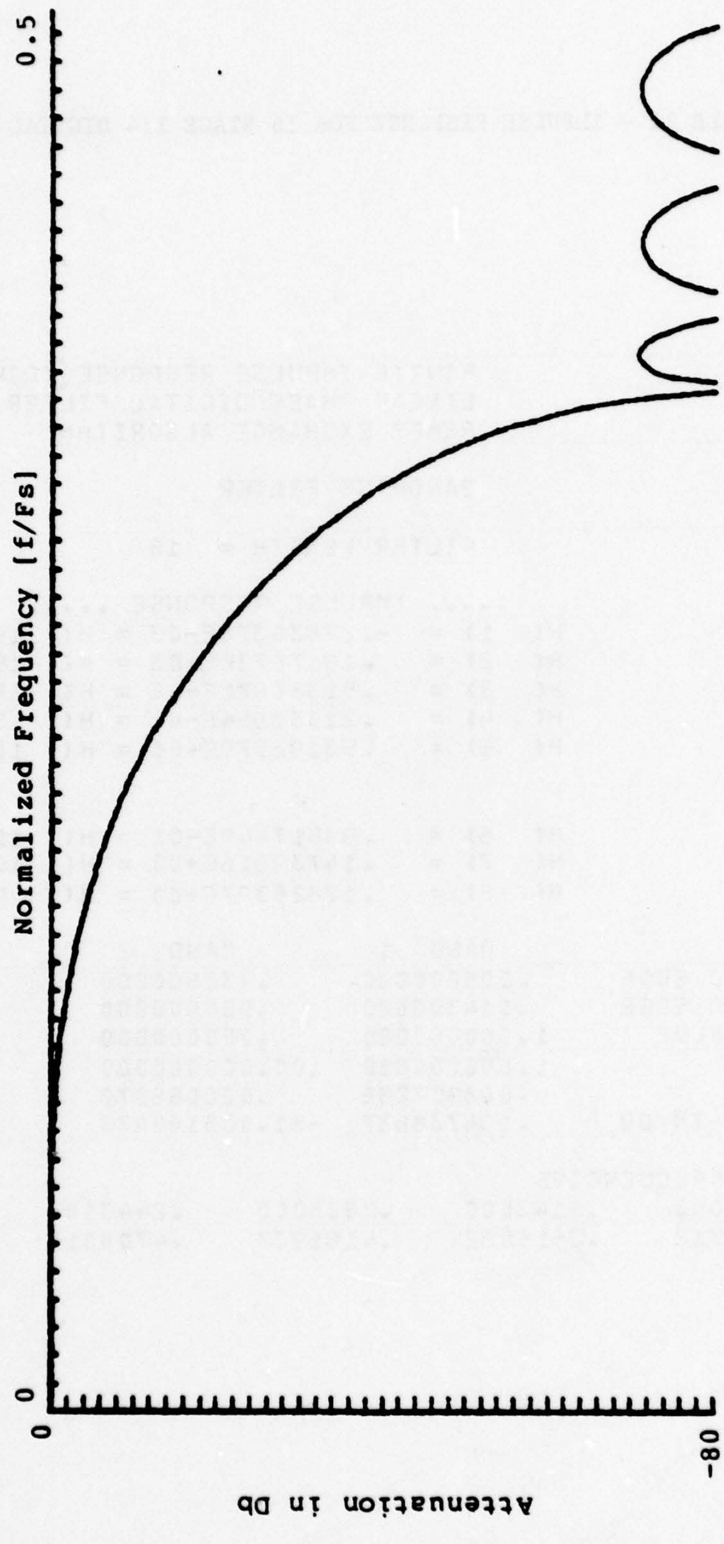


Figure 27 - Frequency Response for 10 Stage 1:2 Digital Filter

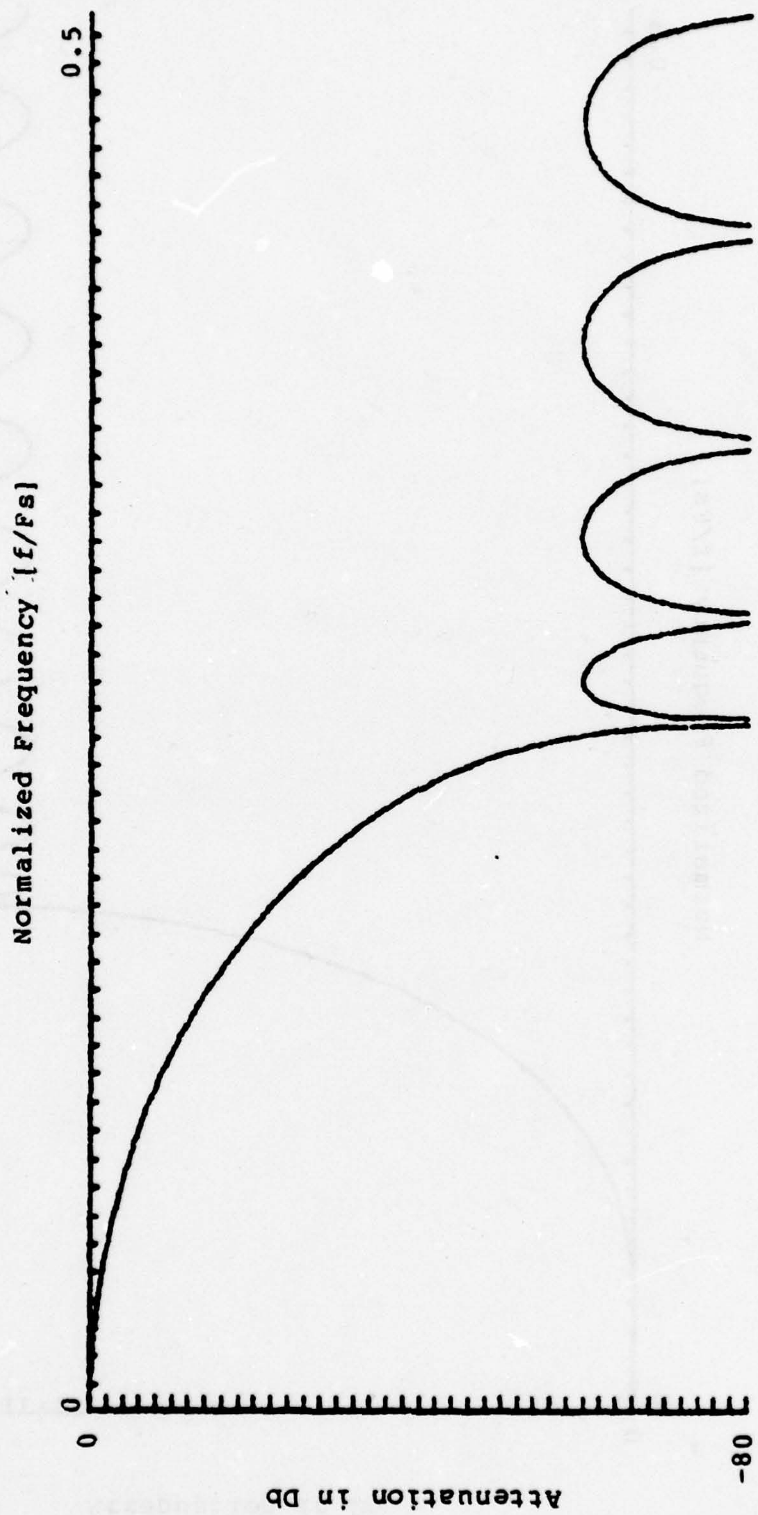


Figure 28 - Frequency Response for 10 Stage 1:4 Digital Filter

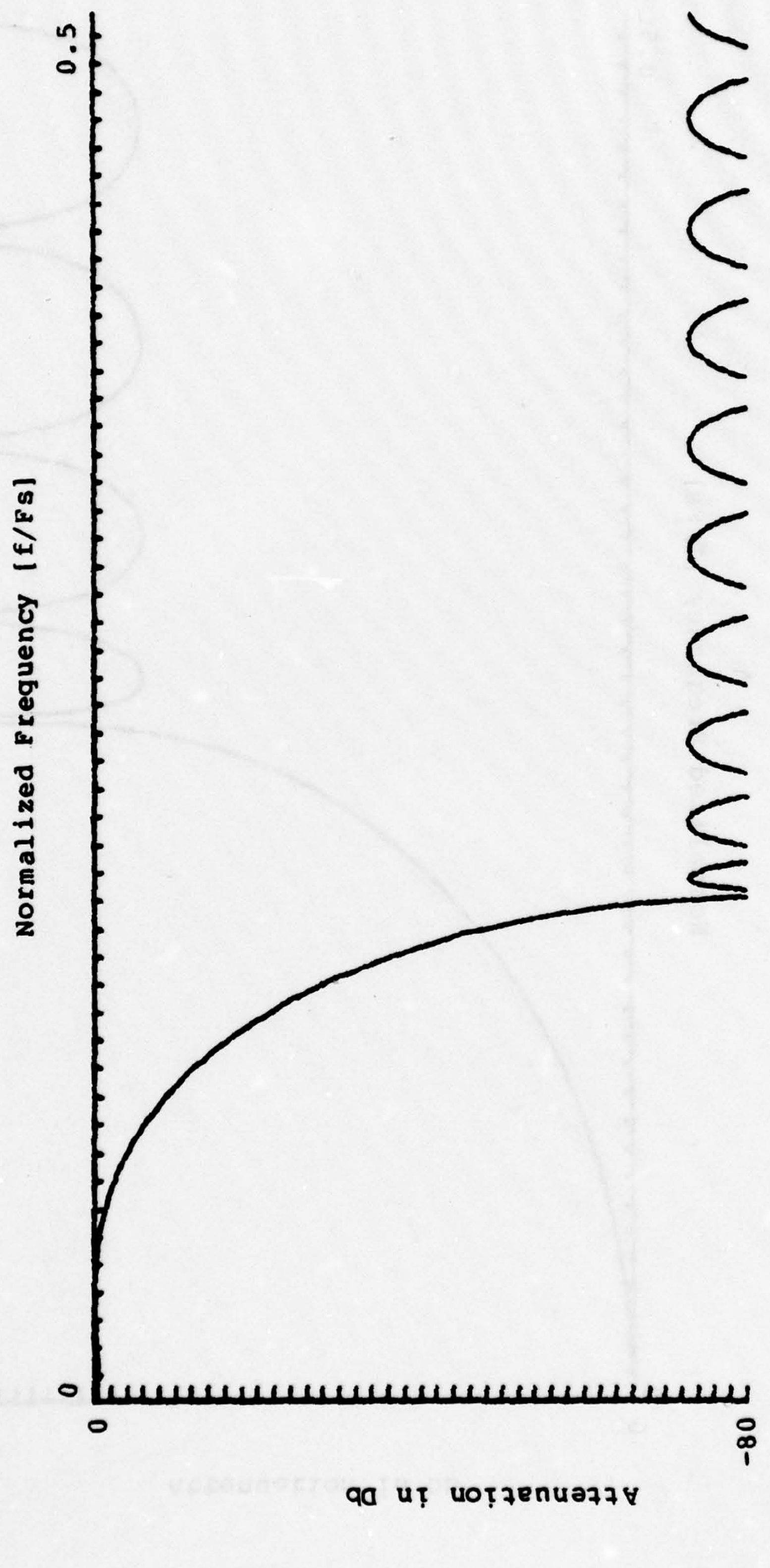


Figure 29 - Frequency Response for 25 Stage 1:4 Digital Filter

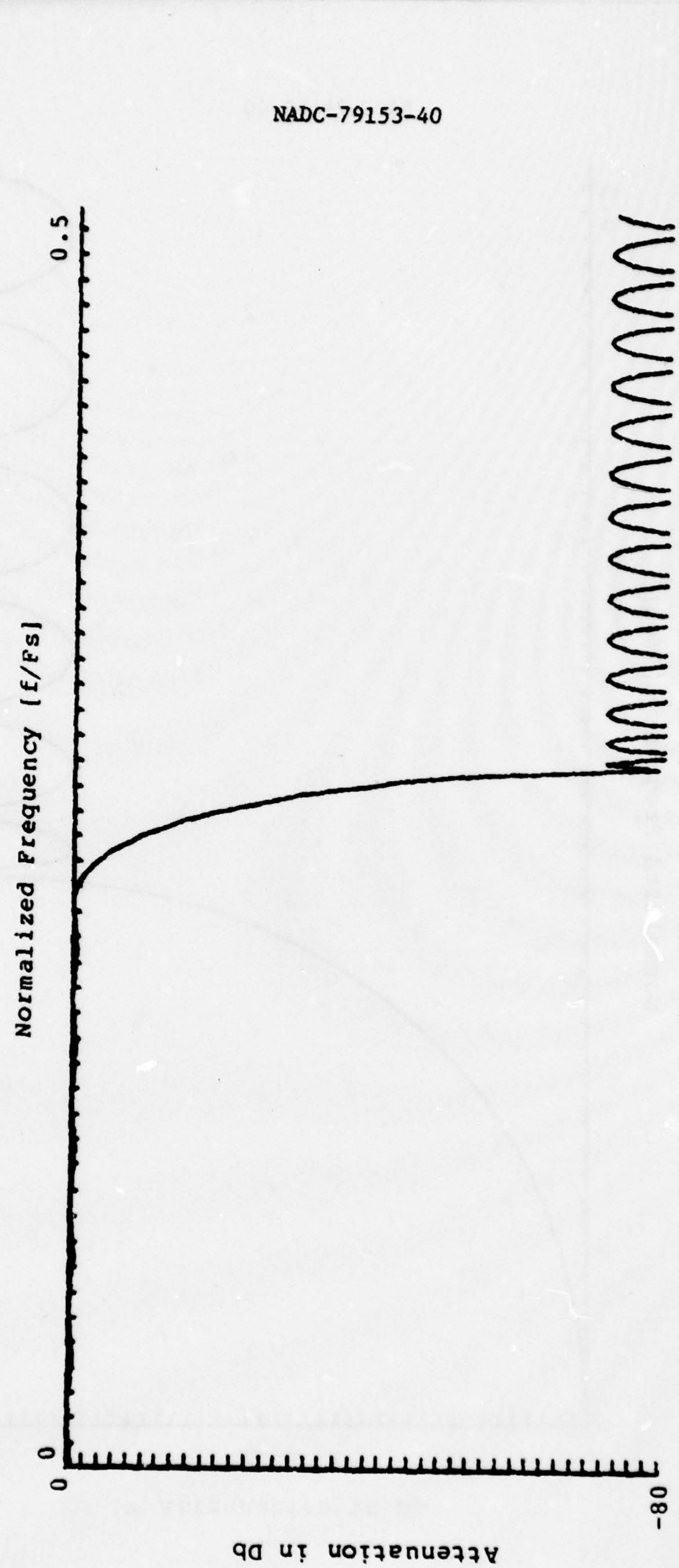


Figure 30 - Frequency Response for 55 Stage 1:1 Digital Filter

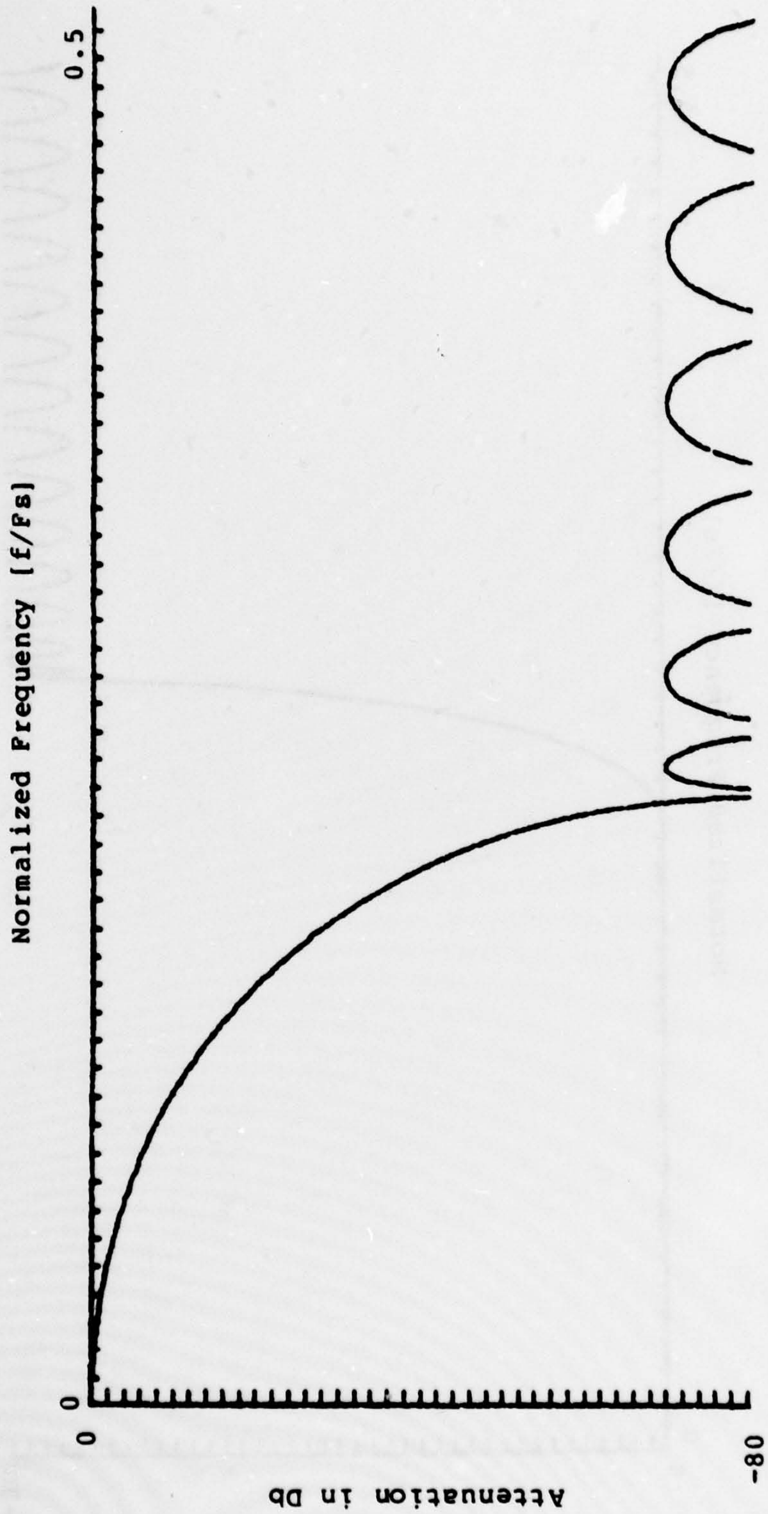


Figure 31 - Frequency Response for 16 Stage 1:4 (SSB) Digital Filter

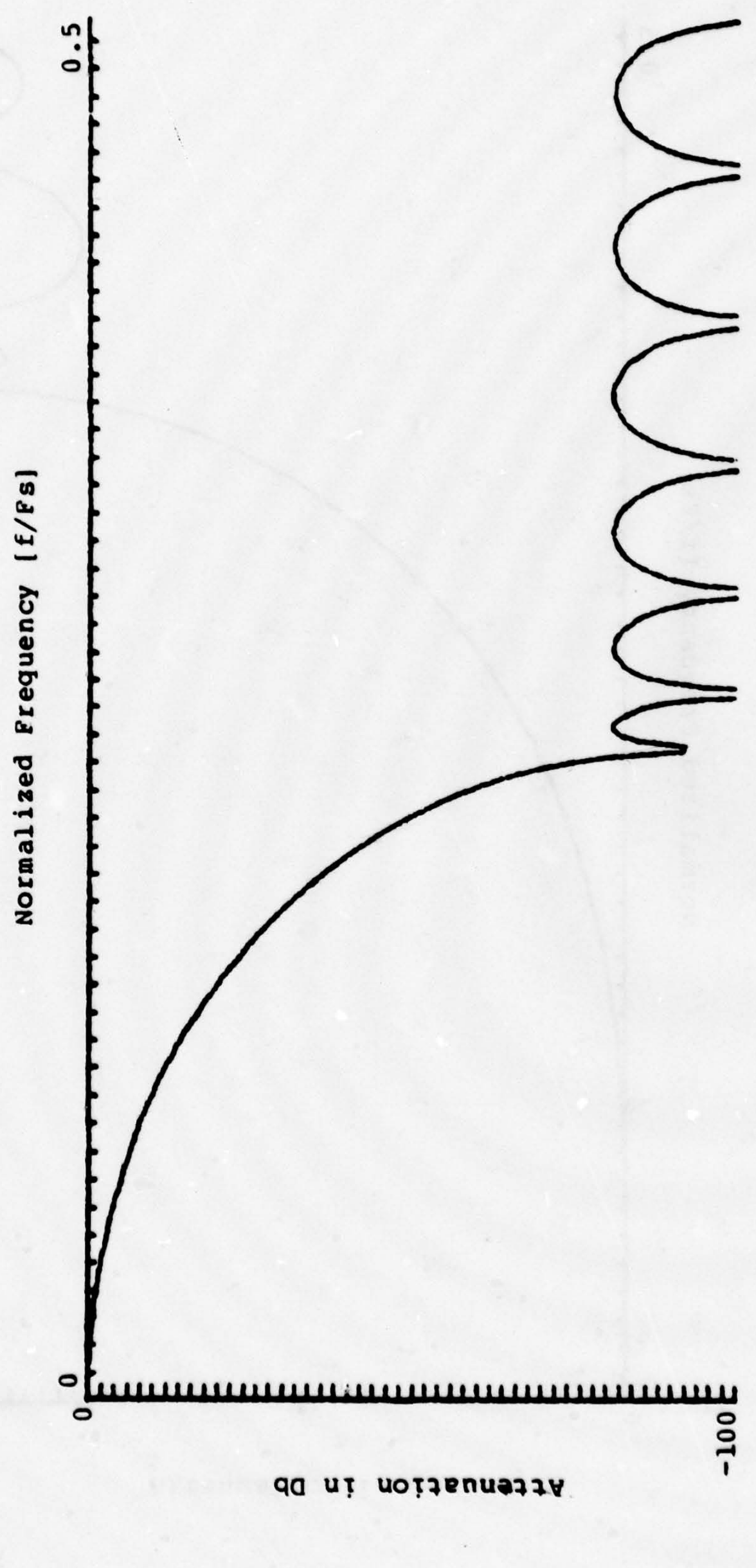


Figure 32 - Frequency Response for 16 Stage 1:4 Digital Filter

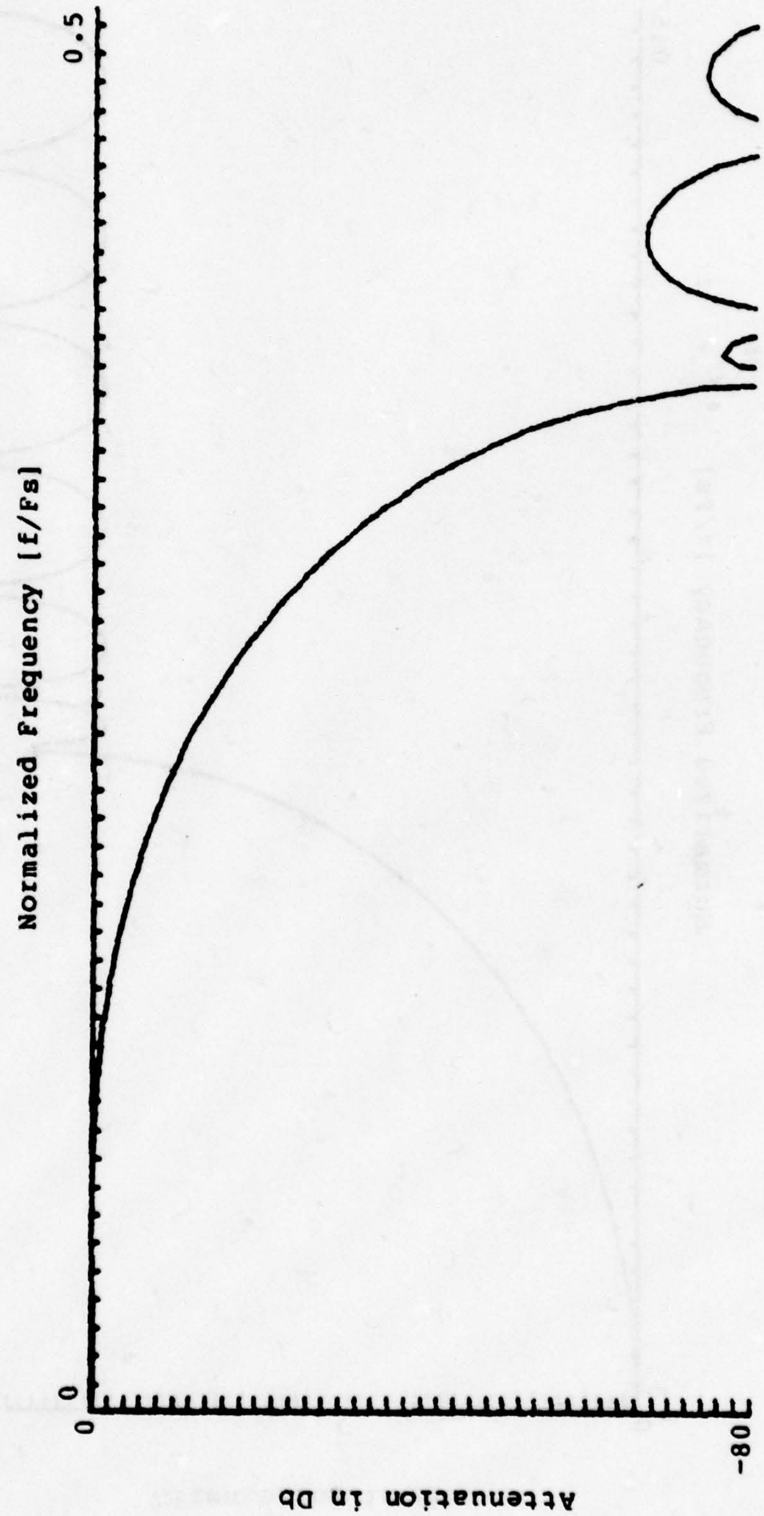


Figure 33 - Frequency Response for 10 Stage 1:2
Digital Filter Rounded off to 12 Bits

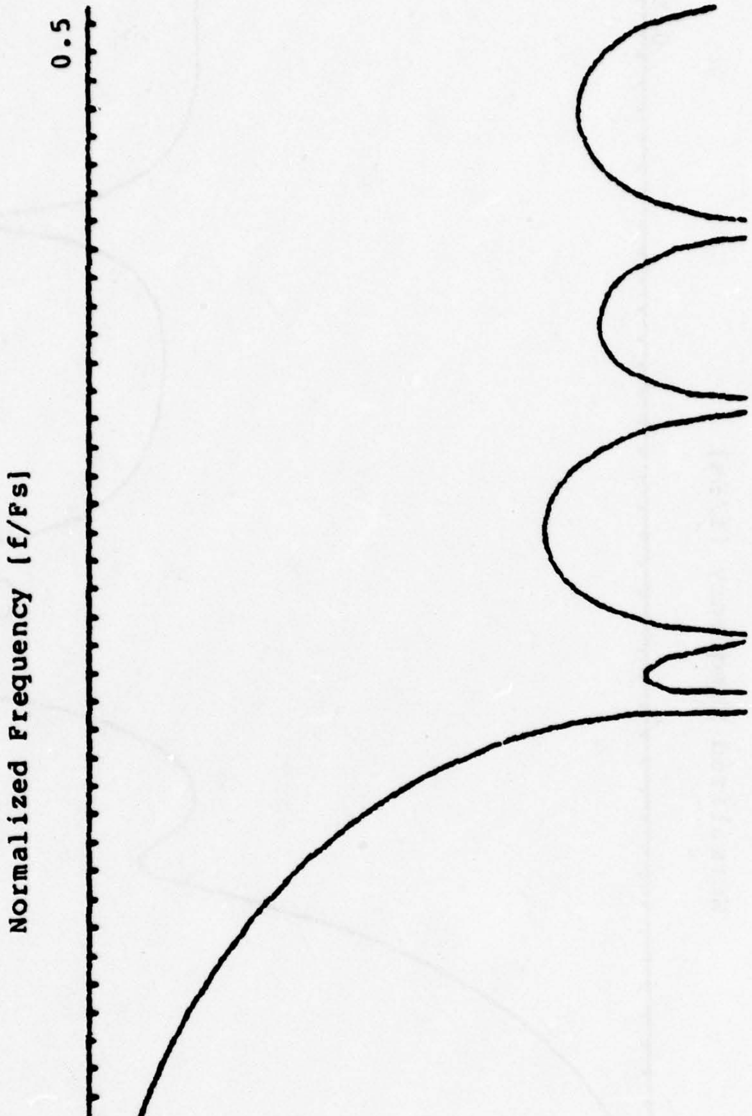


Figure 34 - Frequency Response for 10 Stage 1:4
Digital Filter Rounded off to 12 Bits

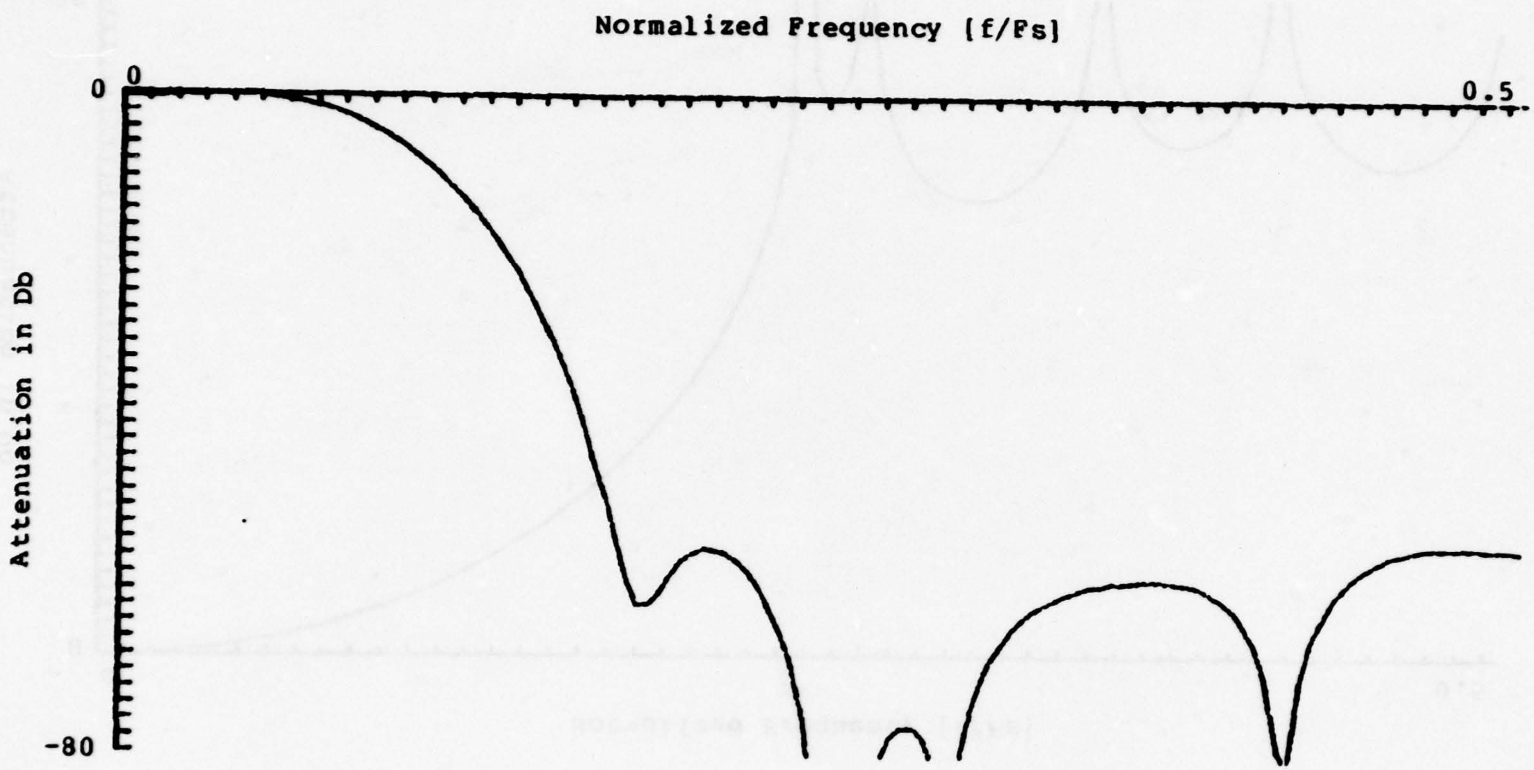


Figure 35 - Frequency Response for 25 Stage 1:4 Digital Filter Rounded off to 12 Bits

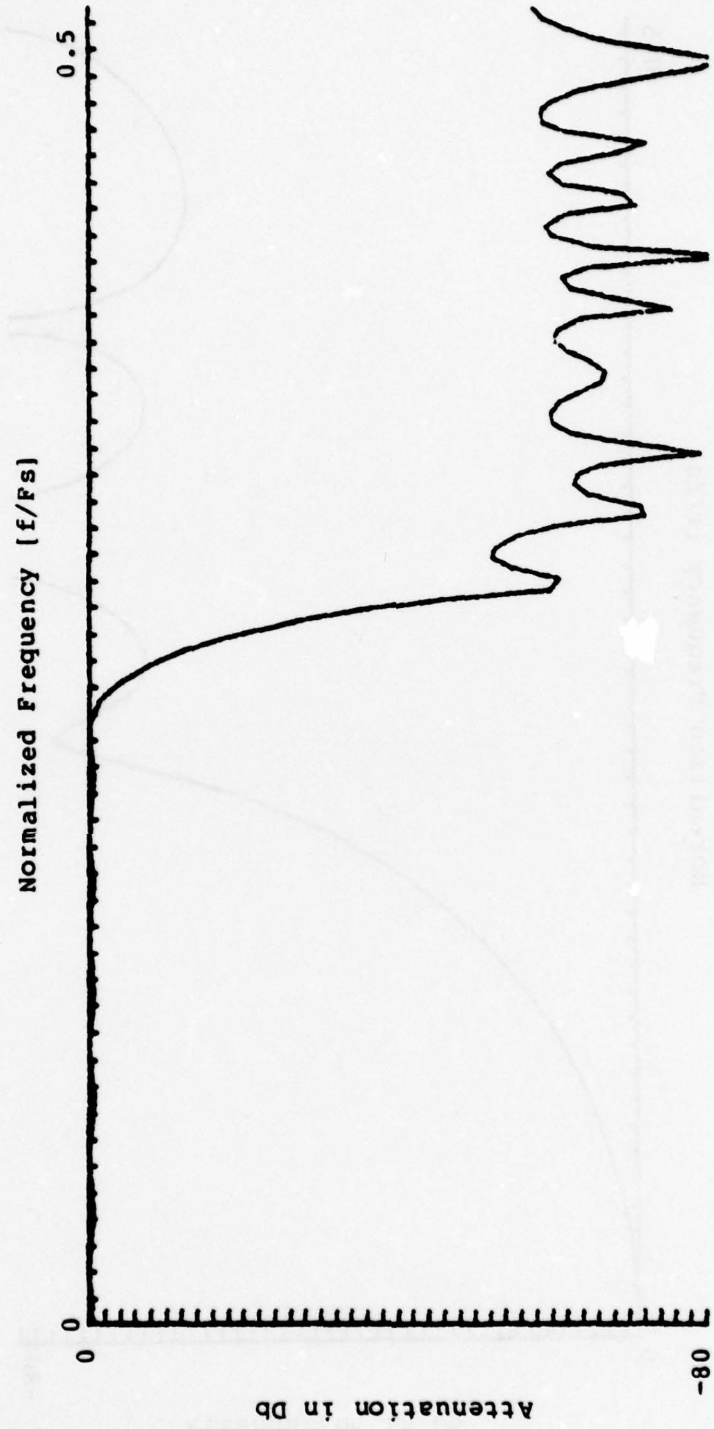


Figure 36 - Frequency Response for 55 Stage 1:1
Digital Filter Rounded off to 12 Bits

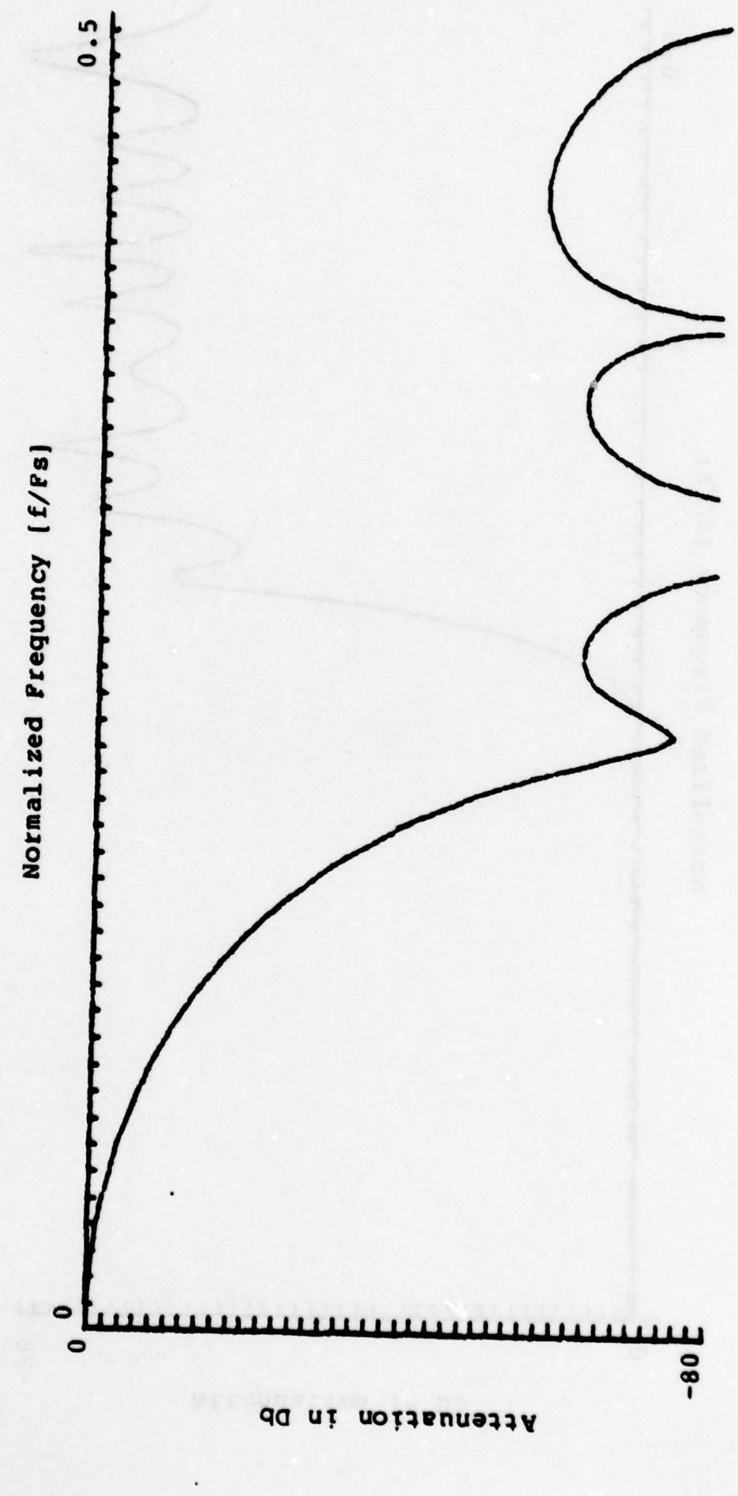


Figure 37 - Frequency Response for 16 Stage 1:4 (SSB)
Digital Filter Rounded off to 12 Bits

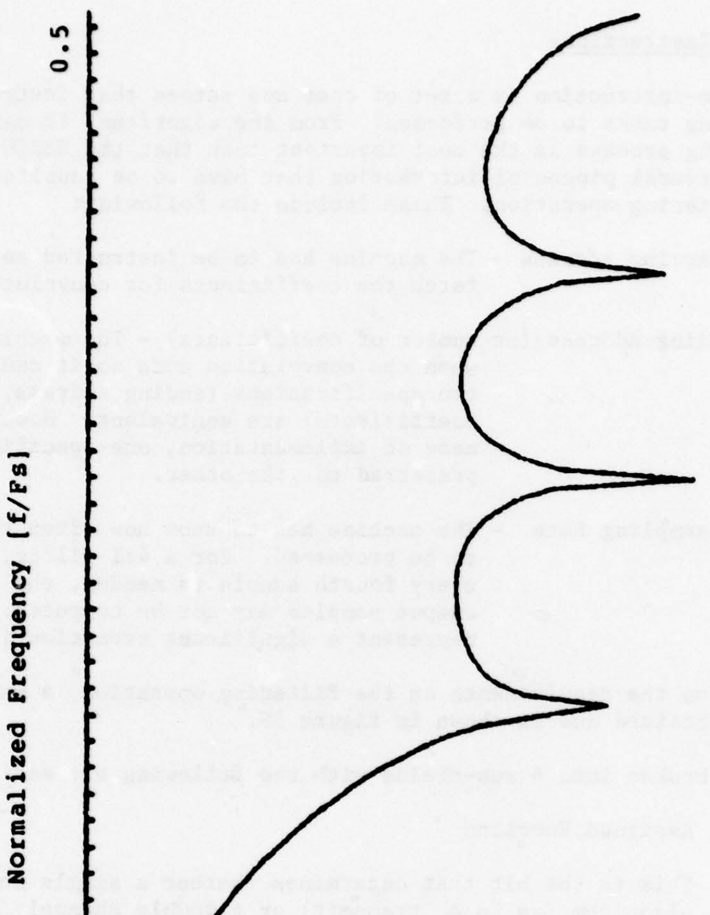


Figure 38 - Frequency Response for 16 Stage 1:4
Digital Filter Rounded off to 12 Bits

4.2 Macro Instructions

A macro-instruction is a set of ones and zeroes that instructs the machine on processing tasks to be performed. From the algorithm, it can be seen that the filtering process is the most important task that the NBSCU has to perform. There are several pieces of information that have to be supplied to the machine for the filtering operation. These include the following:

a. Starting Address - The machine has to be instructed as to where to fetch the coefficients for convolution.

b. Ending Address (or number of coefficients) - The machine has to know when the convolution ends so it can stop. The two specifications (ending address, or number of coefficients) are equivalent. However, due to ease of implementation, one specification may be preferred to the other.

c. Resampling Rate - The machine has to know how often the samples are to be processed. For a 4:1 filter, since only every fourth sample is needed, the other three output samples may not be computed. This will represent a significant reduction in loading.

Based on the requirements on the filtering operation, a macro-instruction format is obtained and is shown in figure 39.

It is broken into 6 sub-fields with the following bit assignments:

Bit	Assigned Function
0	This is the bit that determines whether a single channel (0) algorithm (as in AM transmit) or a double channel (1) algorithm (as in AM receive) is used.
1	This bit determines the rate at which the memory counters count. For a dual channel operation, the rate is half of that of a single channel operation because two multiplications are required.
2	This bit is reserved for single sideband transmit mode.
3	This bit represents a frequency translation process occurring after a filtering process.
4-13	These ten bits represent the starting address for the filter coefficients.
14-23	These ten bits specify the final location for the filter coefficient.

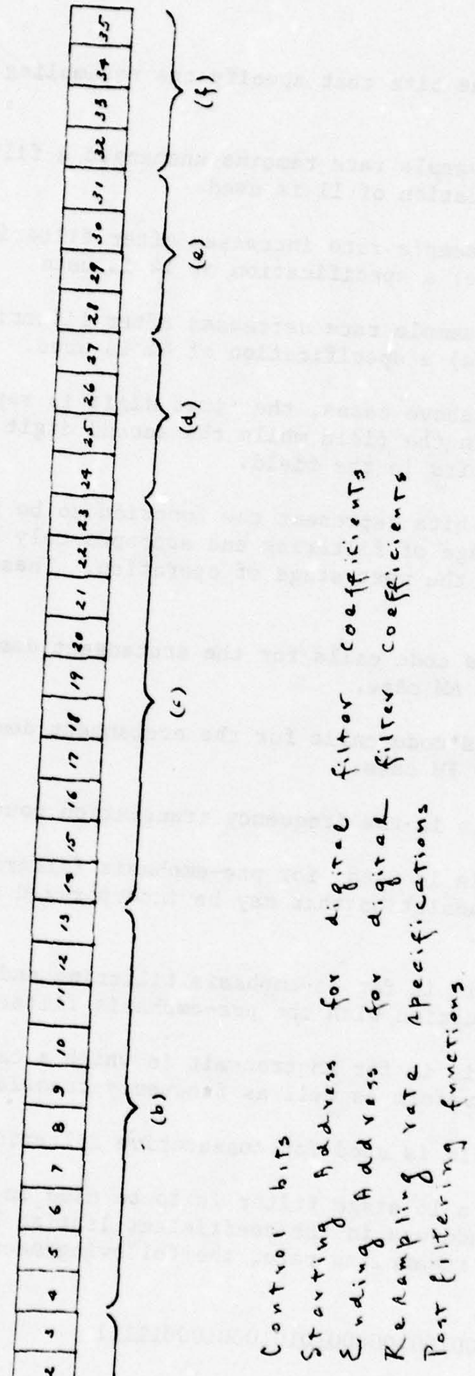


Figure 39 - Macroinstruction Subfield Definition

24-31 These are the bits that specify the resampling rate and are defined as follows:

(a) If the sample rate remains unchanged a filter resampling specification of 11 is used.

(b) If the sample rate increases after filtering (e.g., four times the rate) a specification of 14 is used.

(c) If the sample rate decreases after filtering (e.g., one fourth the rate) a specification of 41 is used.

In all the above cases, the first digit is represented by the first four bits in the field while the second digit is represented by the last four bits in the field.

32-35 These four bits represent the function to be performed after the present stage of filtering and appropriately conditions the bits for use in the next stage of operation. These bits are defined as follows:

0000 This code calls for the arctangent demodulation routine in the AM case.

0001 This code calls for the arctangent demodulation routine in the FM case.

0101 This is the frequency translation routine.

0100 This is used for pre-emphasis filtering and frequency translation that may be incorporated into the algorithm later on.

1000 This is for de-emphasis filtering and will be using in conjunction with the pre-emphasis filter.

1100 This is for AM transmit in which a carrier is added to the waveform as well as frequency translation.

1111 This is used for consecutive filtering process.

Therefore, if a 16 stage filter is to be used in a single channel mode with the starting address in the coefficient listing located at address 10, and assuming a 1:1 resampling rate, the following macrocode can be obtained for this process.

001100000010010000010101000100011111

4.3 The Arctangent Demodulator

The Arctangent algorithm is a fast and efficient algorithm to determine the phase and amplitude of a complex number. The process is essentially a rectangular to polar coordinate conversion process. Given a complex number, which can be represented by the vector diagram in figure 40, the amplitude can be obtained by using the equation

$$A = R \cos \theta + Q \sin \theta.$$

The angle θ is given by:

$$\theta = \tan^{-1} \frac{Q}{R}$$

Notice that regardless of the angle θ , the amplitude is always positive. This is equivalent to the diode detector output of an AM demodulator which is always positive. Only two multiplications and one addition are required to obtain the amplitude provided that the phase is known. It is, therefore, obvious that as far as the demodulation process goes, phase demodulation is the most efficient, if the Arctangent Demodulator is to be used.

Obtaining the absolute phase angle of the vector is a little difficult because it involves a division process as well as an arctangent process. Straight implementation of the division process and the series method of obtaining the arctangent of a number is very inefficient and involves many multiplications. These two methods are therefore abandoned. A more efficient way to implement the two functions is by means of table look up. By putting two tables (1/X, arctangent) on read only memories (ROMS), the phase angle can be obtained by merely one multiplication. Therefore, four tables, as seen from the equation, are required for the Arctangent Demodulator for both amplitude and phase computation. These are the 1/X, $\tan^{-1}$, sine and cosine tables. Since the sine and cosine can be obtained from one another, one table can be eliminated. Furthermore, since the value of the sine and cosine function are unique up to 90°, the required sine table will be one fourth of the full sine table. To reduce the size of the arctangent table, the magnitude of the two vector components can be obtained and compared. The larger of the two vectors can be used to address the 1/X table. Multiplication of the table output and the smaller vector will produce a number that is less than 1, corresponding to 45°. Therefore, a $\tan^{-1}$ table ranging from 0 to 45° is all that is required. The procedure for obtaining the phase, and amplitude is as follows:

1. Obtain magnitude of vectors and store original signs of vectors.

2. Compare the two numbers and use the larger of the two to address the 1/X table.

3. Multiply the output of the 1/X table with the smaller number to form the product $|Y|/|X|$.

4. Use the product to address the arctangent ROM to obtain the angle in the first octant.

For getting the magnitude of the vector, the angle can be put into the proper quadrant and the corresponding values of the sine and cosine of the

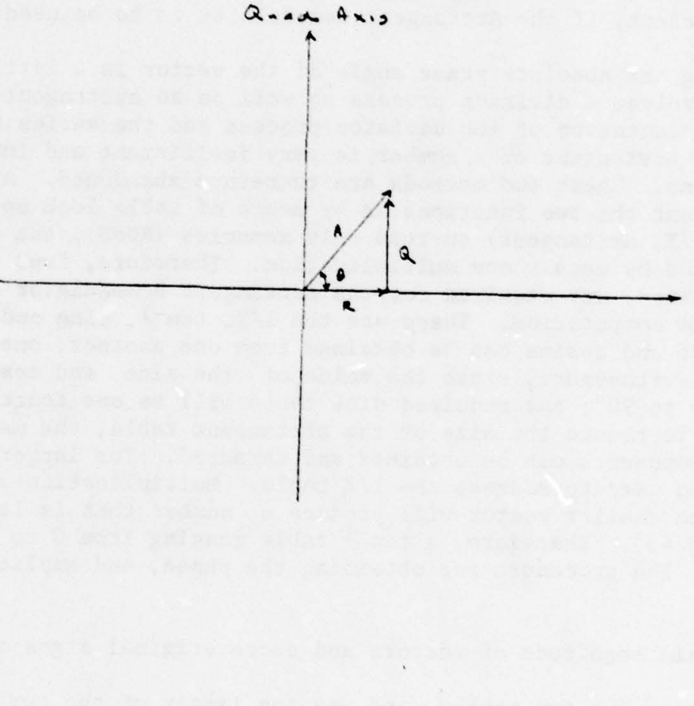


Figure 40 - Real and Quad Components of a Complex Number

angle in the first quadrant are obtained from the sine table. The amplitude is equal to R cos θ + Q sin θ. Since the amplitude is always positive, and also cos θ and sin θ is positive in the first quadrant, the following equation can be used without affecting the result.

$$/R/ \cos\theta + /Q/ \sin\theta$$

where $0° \leq \theta \leq 90°$ and is the angle that has the same absolute sine and cosine value as the original angle. The absolute phase angle can then be determined by using table 12. Notice that the angle ranges from +180° to -180° because bipolar 2's complement arithmetic is used.

TABLE 12 - PHASE DETERMINATION OF ARCTANGENT DEMODULATOR

R	Q	|R| - |Q|	Octant	Absolute Angle
+	+	+	1	$\mathrm{Tan}^{-1} \|Q\|/\|R\|$
+	+	-	2	$90° - \mathrm{Tan}^{-1} \|R\|/\|Q\|$
-	+	-	3	$90° + \mathrm{Tan}^{-1} \|R\|/\|Q\|$
-	+	+	4	$180° - \mathrm{Tan}^{-1} \|Q\|/\|R\|$
-	-	+	5	$-180° + \mathrm{Tan}^{-1} \|Q\|/\|R\|$
-	-	-	6	$-90° - \mathrm{Tan}^{-1} \|R\|/\|Q\|$
+	-	-	7	$-90° + \mathrm{Tan}^{-1} \|R\|/\|Q\|$
+	-	+	8	$-\mathrm{Tan}^{-1} \|Q\|/\|R\|$

NBSCU Design Requirements

Two of the most important functions that the narrowband signal conversion unit has to perform is that of filtering and demodulation. Since the unit has to handle FM signals which are very phase sensitive, non-recursive finite impulse response (FIR) filters were chosen over infinite impulse response (IIR) filters. In the FM and SSB mode, notice that the output of one filter is the input to another filter, and therefore, the filtering circuits must have the capability to loop the output back to the input of the filter to be reprocessed. Furthermore, since the filters are performing the resampling and desampling functions, this imposes a condition on the input and output buffer size. The maximum requirement occurs in the single sideband receive and transmit mode in which 32 input samples are required to produce one output sample. Thus, the buffers used must be at least 32 words in length. A clock generator is also required to supply the required clocks to the A/D and D/A converters. A multiplier, which is the most important element in digital filtering, must be capable of operating fast enough to do the required number of multiplications in real time. The various functions that are to be performed should be microprocessor controlled. In the event that the microprocessor fails, it is desirable that a standby function can be loaded manually by the operator. This is also one of the redundant features in the TIES program.

5. NBSCU CIRCUIT DESIGN

Circuits were designed based on the functions that the narrowband signal conversion unit has to perform and can be separated into two categories — analog and digital. A top level block diagram for the narrowband signal conversion unit is shown in figure 41.

Discussions on the analog and digital circuits are given in the following sections.

5.1 Analog Interface Circuits

The analog circuits serve as the interface between the analog world and the signal processor. The interface consists of a sample-and-hold circuit, an A/D converter at the input and D/A converter at the output. A circuit that generates the sampling rate suitable for conversion is also part of the analog interface circuit which is shown in figure 42. All circuit components are commercially available hardware.

Upon command from the sampling rate generator, the input signal to the sample-and-hold circuit is sampled. The sampled value is then held by a capacitor. The A/D convertor will produce a 10 bit word in two's complement notation. The sampling rate generator is simply a programmable divide-by-N circuit that takes in a frequency of 230.4 KHz and produce a sub-multiple of this frequency. Depending on whether the transmit or receive mode is used, the higher or lower frequency is routed appropriately to the A/D or D/A converter. Notice that no phase lock loop is required to generate the sampling frequencies because the resampling and desampling filters are multiples or reciprocals of the input sampling rate.

5.2 Digital Circuits

A detailed block diagram for the NBSCU is shown in figure 43.

The digital circuit can be partitioned into two parts — the loading circuits used to load the filter coefficients from mass storage memory into the memory of the signal processor and the circuits that process the input samples.

5.2.1 Loading Circuits

The loading circuit is shown in figure 44. The operation of this circuit is described as follows:

Toggle switch, S01, in the open position, enables EPROMS 1 and 2 and the B inputs to MU1. By depressing the preset switch S02, the appropriate flip-flops and counters are preset. Depressing switch S03 causes D0101 to change state, thereby turning the master clock on at A020103. The clock then goes through a counter which has outputs that are sub-multiples of the basic clock rate. This is necessary because the EPROMS have a cycle time that is much longer than the 200 nsec master clock period. Using the divide by 16 output

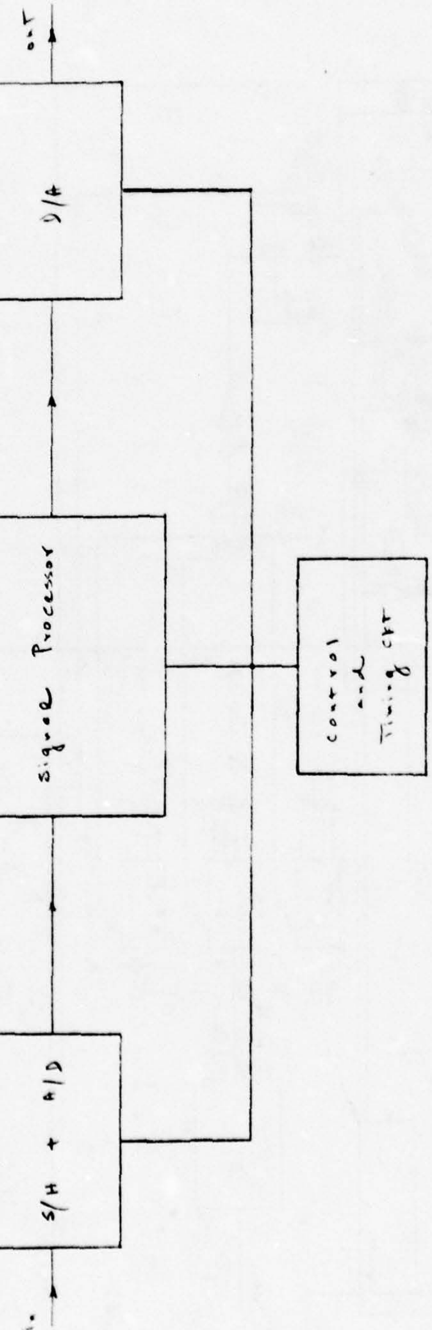


Figure 41 - Simplified Diagram for TIES NBSCU

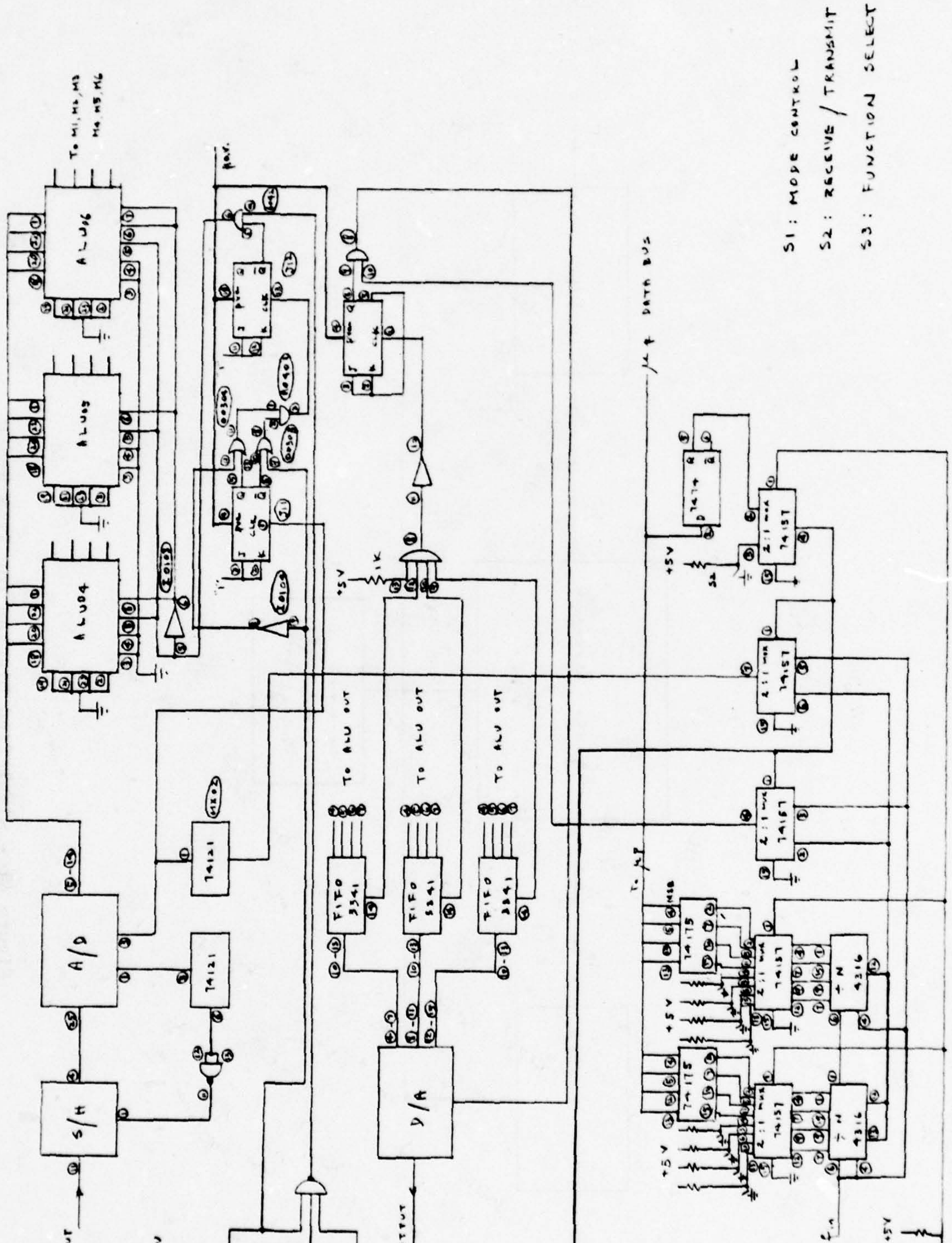


Figure 42 - NBSCU Analog Interface and Clock Generator Circuits

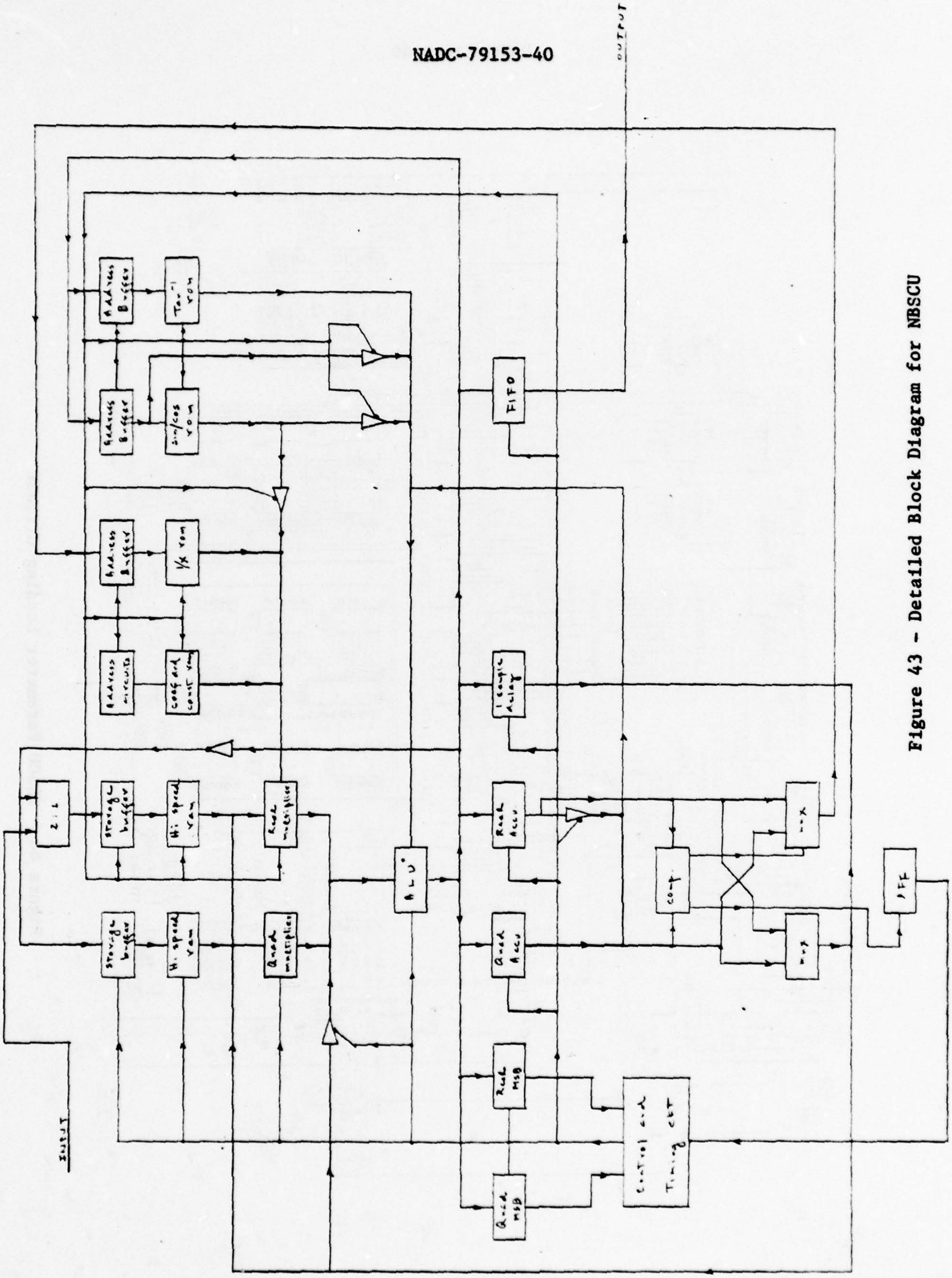


Figure 43 - Detailed Block Diagram for NBSCU

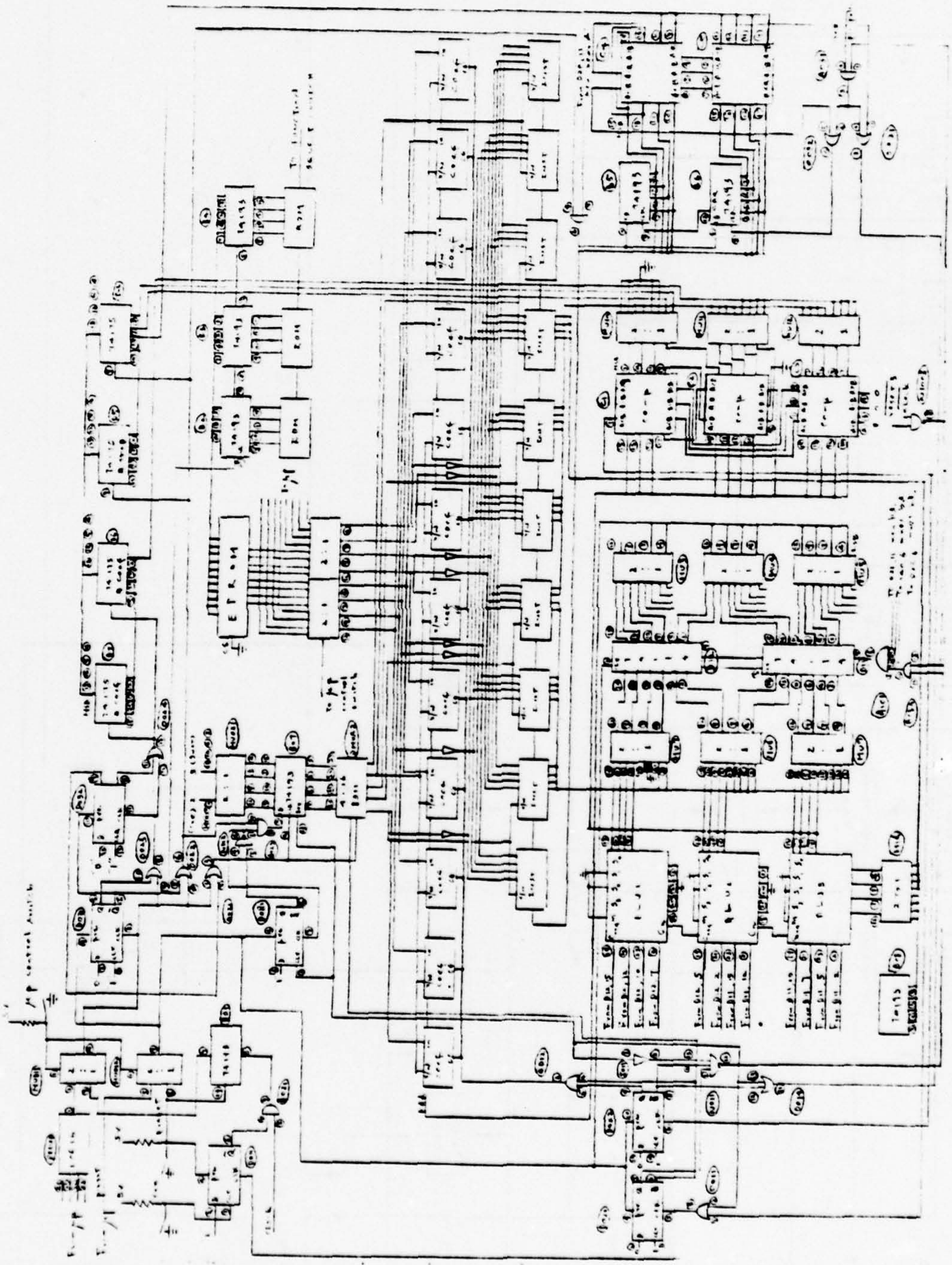


Figure 44 - NBSCU Parameter Loading Circuit

as the clock to the EPROM address counter, the contents of the first location (which consists of the number of stages of filtering involved in the mod/demo process) is loaded into flip-flops D04, D05 while the content of location 2 (which contains the required number of constants and coefficients) is loaded into flip-flops D06, D07. The 36 bit macro instructions, each of which occupy 5 locations in the EPROM are loaded into the instruction RAMs which are addressed by counters B05, B06. Comparators C04, C05, which compare the number of stages required for the mod/demod process and the counter outputs, produce a signal when all the macro instructions are loaded into the instruction RAMs. The coefficient RAMs then begin to fill up. The addressing device in this case is a set of arithmetic logic units because of the resampling requirement which will be seen later on. Upon completion, comparators C01, C02, C03 produce a signal that changes the state of flip-flop D0201, which turns a LED on indicating that the processor is loaded.

5.2.2 Signal Processor Circuits

Upon completion of the loading process, the signal processor is left in the idle state. It will remain in the idle state until enough samples required for processing are gathered. The number of samples required can be determined by the overall resampling rate. The circuit for buffer loading is shown in figure 45.

When the required samples are acquired, a signal is produced at 0240308 and shaped by the one shot circuit MX02 which produces a 30 nsec pulse that changes the state of D0802 which turns 00401 on. The signal processor now becomes active with the master clock turned on at 0040103. Three reset pulses are produced and designated RESET 1, RESET 2, RESET 3. RESET 1 is used to reset devices that need to be reset after all the input samples are processed. RESET 2 is used to reset devices that need to be reset after each filtering process. RESET 3 is used to reset devices that need to be reset after each convolution.

After the proper devices are reset by the three reset pulses, the input data, which are being held by the input buffer begins the transfer to the high speed RAMs. The data transfer is complicated due to the resampling process. The circuit is shown in figure 46.

Inputs to comparators C01 and C02 will be coming from the macro instruction RAMs representing the resampling rate used for the particular filter. Since a resampling filter of 4:1 requires the processing of 1 sample out of 4 input samples, four samples are loaded into the high speed RAMs. However, the convolution process only takes place after the fourth sample is loaded. On the other hand, if a 1:4 filtering function is to be performed all input samples are used (including the zeroes used for zero filling). Therefore, the inputs to comparator C01 determine the overall samples required to produce one sample. However, the input to the comparator C02 does not represent the number of zeroes to be filled. Instead, the number of zeroes needed is one less than the input to comparator C02. Provision is made so that the state of the counter B01 can be stored in memory R05 in case not

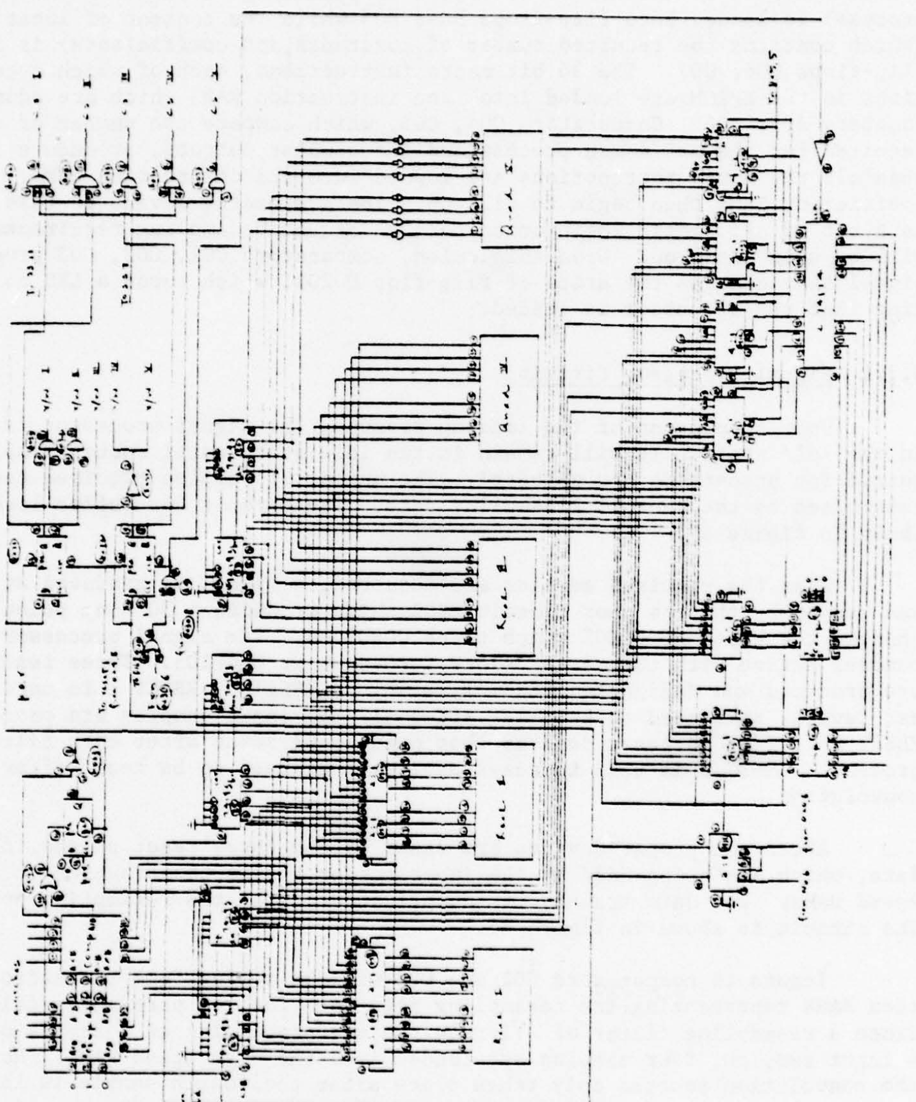


Figure 45 - NBSCU Buffer Loading Circuit

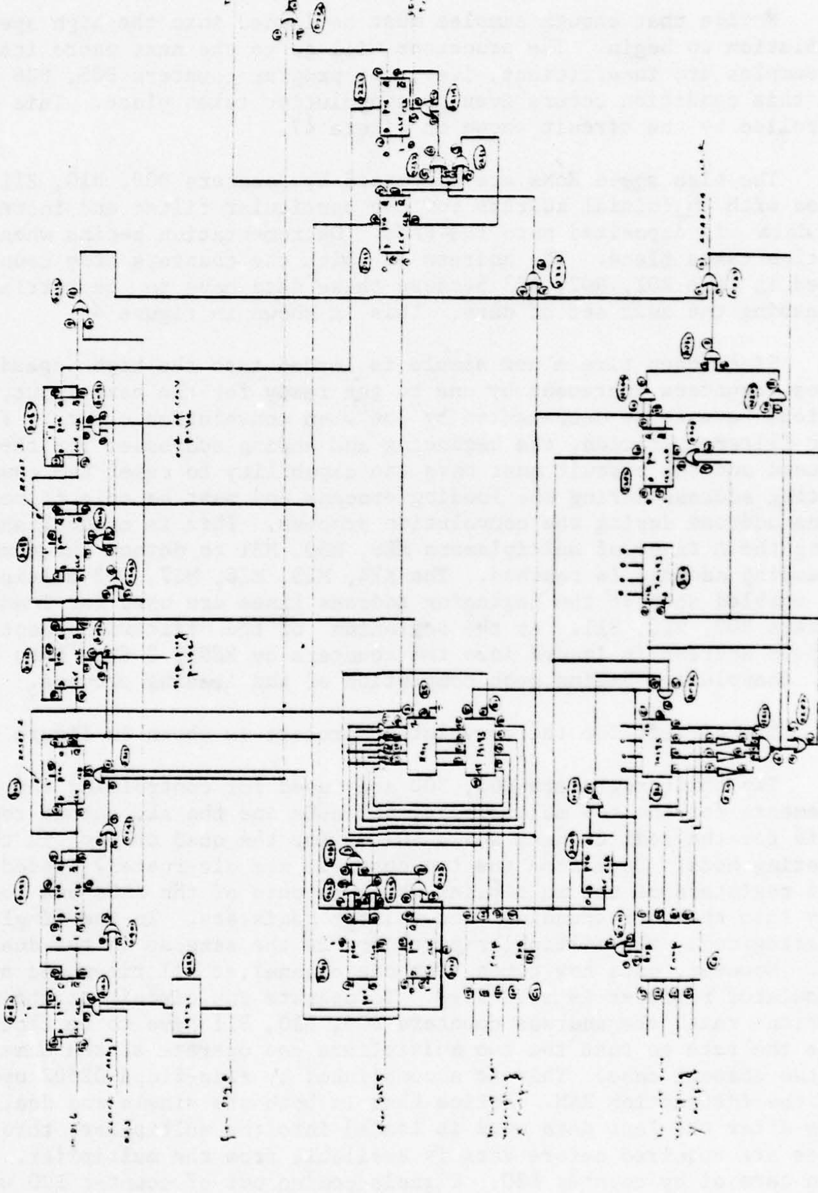


Figure 46 - Reset and Control Circuit for Transferring Data into High Speed RAM

enough samples are available for processing. When enough samples are loaded in the high speed RAMs for one convolution, a signal from comparator C01 will turn on the clock to the convolutional filter at 0220102.

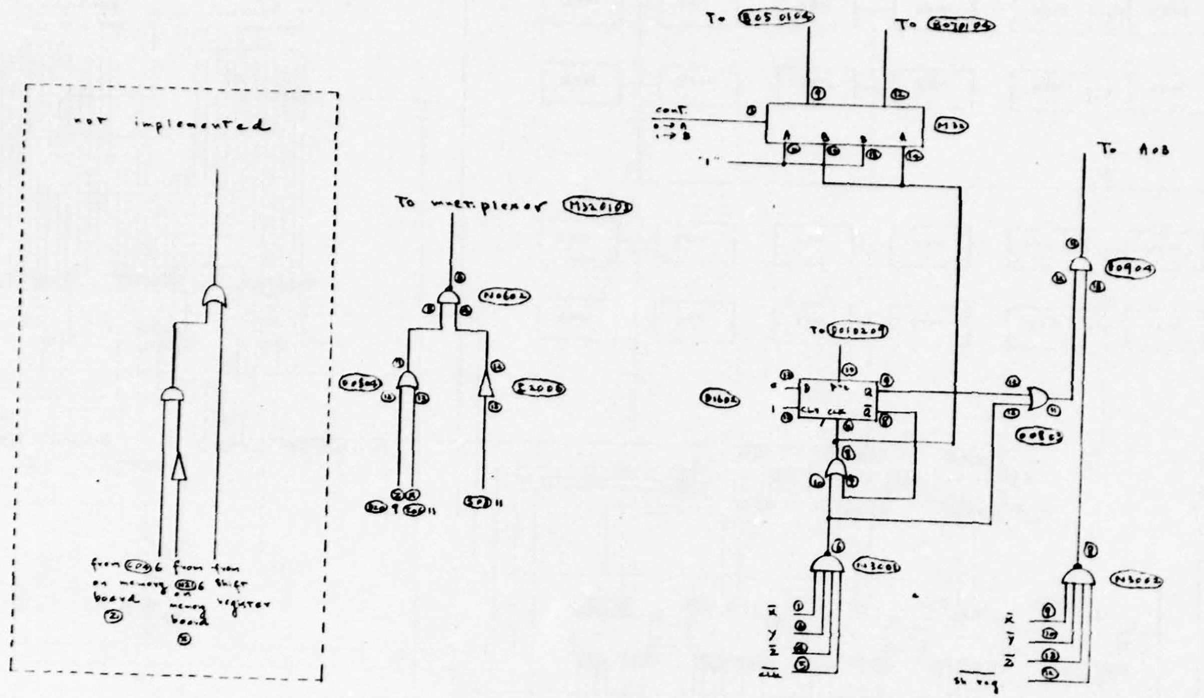
Notice that enough samples must be loaded into the high speed RAMs for convolution to begin. The processor will go to the next macro instruction if the samples are insufficient, i.e., the program counters B05, B06 will increment when this condition occurs even no convolution takes place. This operation is controlled by the circuit shown on figure 47.

The high speed RAMs are addressed by counters B09, B10, B11, which are loaded with an initial address for the particular filter and increment each time new data is deposited onto the RAMs. Decrementation begins when actual convolution takes place. The address at which the counters stop counting is stored in RAMs R01, R02, R03 because these data have to be retrieved for processing the next set of data. This is shown in figure 48.

Since each time a new sample is loaded into the high speed RAM, the address counters increment by one to get ready for the next input, these counters have to be decremented by one when convolution occurs. For a particular filter operation, the beginning and ending addresses for the coefficients are used and the circuit must have the capability to reset the counter to the starting address during the loading process and must be able to reset to the ending address during the convolution process. This is accomplished by enabling the A input of multiplexers M29, M30, M31 to detect the condition that the ending address is reached. The M24, M25, M26, M27, M28 multiplexers are then enabled so that the beginning address lines are used for loading into counters B09, B10, B11. At the beginning of the filtering function, the previous address is loaded into the counters by RESET 3 from RAMs R01, R02, R03. Convolution begins upon completion of the loading process.

The circuit for the convolution process is shown in figure 49.

Two shift registers S01, S02 are used for controlling the timing requirements for the two multipliers, the ALUs and the ALU output registers. S01 is for the real channel while S02 is for the quad channel in the two channel operating mode. Data from the two channels are alternately loaded into the input registers of the multiplier while outputs of the ALUs are loaded appropriately into the corresponding accumulator registers. In the single channel operating mode, the multiplier circuitry is the same as in the dual channel mode. However, data now comes from one channel at all times and only one accumulator register is activated. To operate the circuit at the most efficient rate, the address counters B09, B10, B11 have to be clocked at twice the rate so that the two multipliers can operate at the same rate as in the two channel case. This is accomplished by flip-flops D2502 upon command from the instruction RAM. Notice that in both the single and dual channel cases after the last data word is loaded into the multiplier, three clock pulses are required before data is available from the multiplier. This is taken care of by counter B20. Signals coming out of counter B20 will trigger the state of flip-flop D1101 to change indicating that convolution is completed.



Y : X   Z : Total sample available

data ram counter has to be decremented before it can be increment

Z : D20 9	$\bar{Z}$: D20 8	$\overline{\text{clock}}$: I20 10	I20 11 connects O24 6
X : I06 11	$\bar{X}$: I06 10	$\overline{\text{SL reg}}$: I20 12	
Y : C01 5	$\bar{Y}$: C01 6		

Figure 47 - Program Counter Control Circuit

the phase and amplitude of a complex number. The process is essentially a

- 62 -

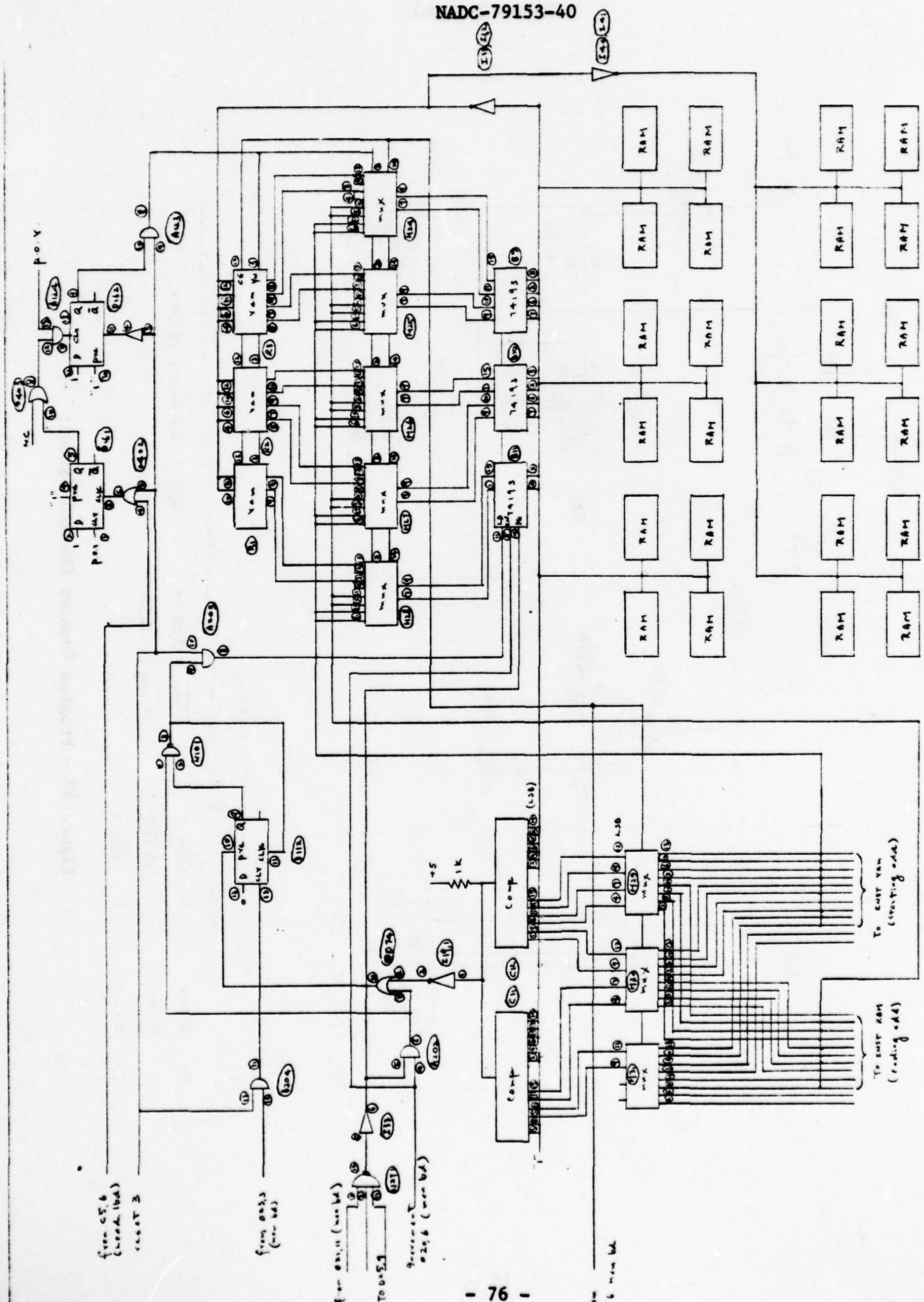


Figure 48 - Addressing Circuit for High Speed RAMs

proper quadrant and the corresponding values of the sine and cosine of the

- 63 -

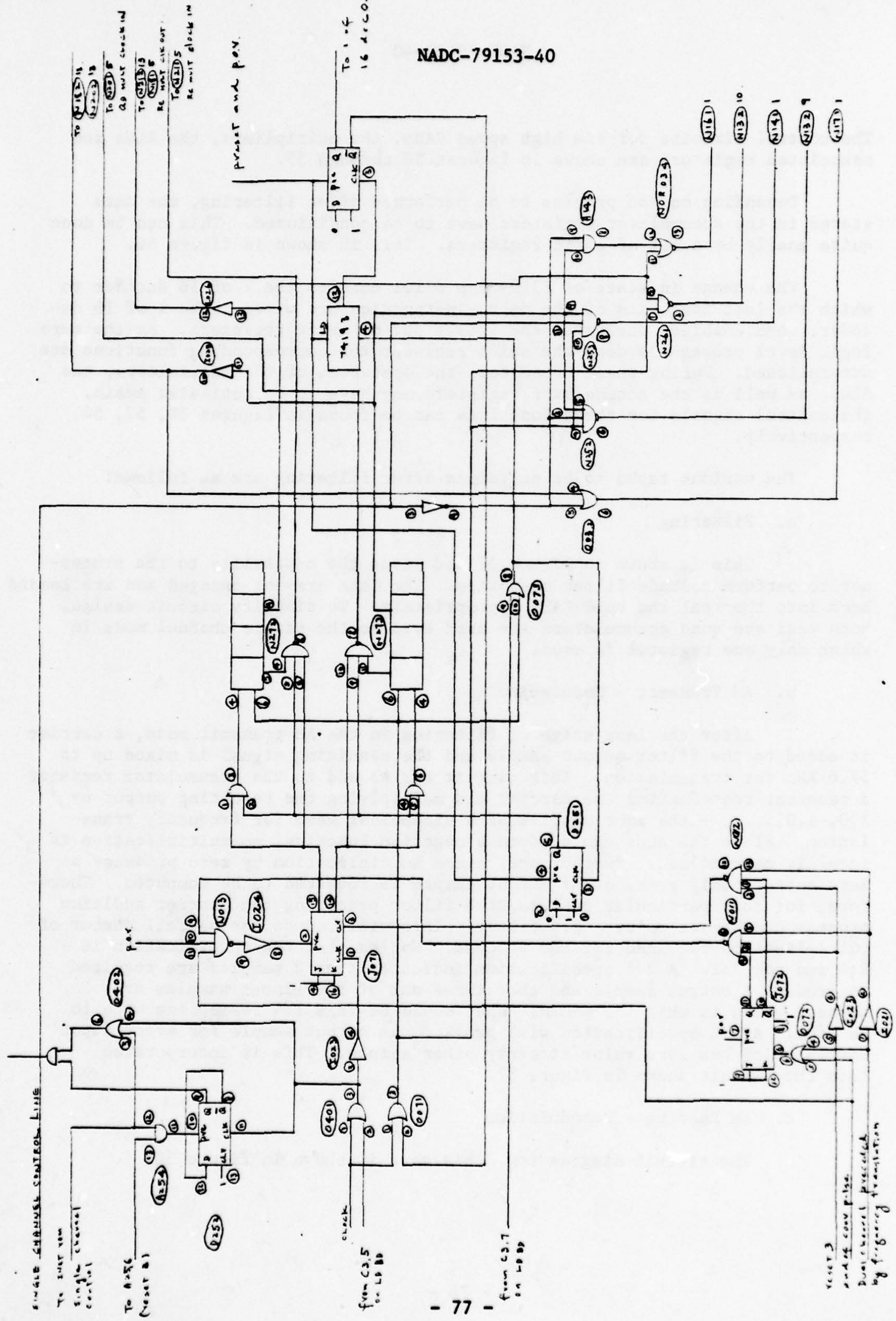


Figure 49 - Control Circuit for Convolutional Filtering

Figure 40 - Real and Quad Components of a Complex Number

- 64 -

The control circuits for the high speed RAMs, the multipliers, the ALUs and associated registers are shown in figures 50 through 55.

Depending on the process to be performed after filtering, the data stored in the accumulator registers have to be conditioned. This can be done quite easily by a set of shift registers. This is shown in figure 56.

The change in state of flip-flop D1101 enables the 1 of 16 decoder to which the last four bits of the macro-instruction are wired. The 1 of 16 decoder, when enabled, activates the proper set of shift registers. As the zero logic level propagates down the shift register the corresponding functions are accomplished. During these processes, the operation of the multipliers, the ALUs, as well as the accumulator registers may have to be activated again. The control signals for these functions can be found in figures 50, 52, 54, respectively.

The various tasks to be performed after filtering are as follows:

a. Filtering

This is shown in figure 57 and gives the capability to the processor to perform cascade filter operation. The data are not changed and are loaded back into the real and quad RAMs appropriately. To simplify circuit design, both real and quad accumulators are used even in the single channel mode in which only one register is used.

b. AM Transmit - Modulation

After the last stage of filtering in the AM transmit mode, a carrier is added to the filter output sample and the resulting signal is mixed up to 57.6 KHz for transmission. This circuit has to add to the accumulator register a constant representing the carrier and multiplying the resulting output by 1,0,-1,0,...., — the four samples of a sinusoidal wave for frequency translation. Since the ALUs can perform a negation function, no multiplication is actually carried out. Furthermore, since multiplication by zero produces a zero output, only every other output sample is required to be computed. Therefore, for this particular process, the filter preceding the carrier addition process can be set to be a 2:4 filter. This will reduce the overall number of multiplication required for the process. Notice that the specification is 2:4 and not 1:2. A 2:4 specification indicates that 2 samples are required to produce 1 output sample and that three out of four input samples are zeroes, which is what the normal input would be if a 1:4 resampling were to be done. A 1:2 specification will produce one output sample for every input sample which has zero value at every other sample. This is incorporated into the circuit shown in figure 57.

c. AM Receive - Demodulation

The circuit diagram for this case is shown in figure 58.

can be loaded manually by the operator. This is also one of the redundant features in the TIES program.

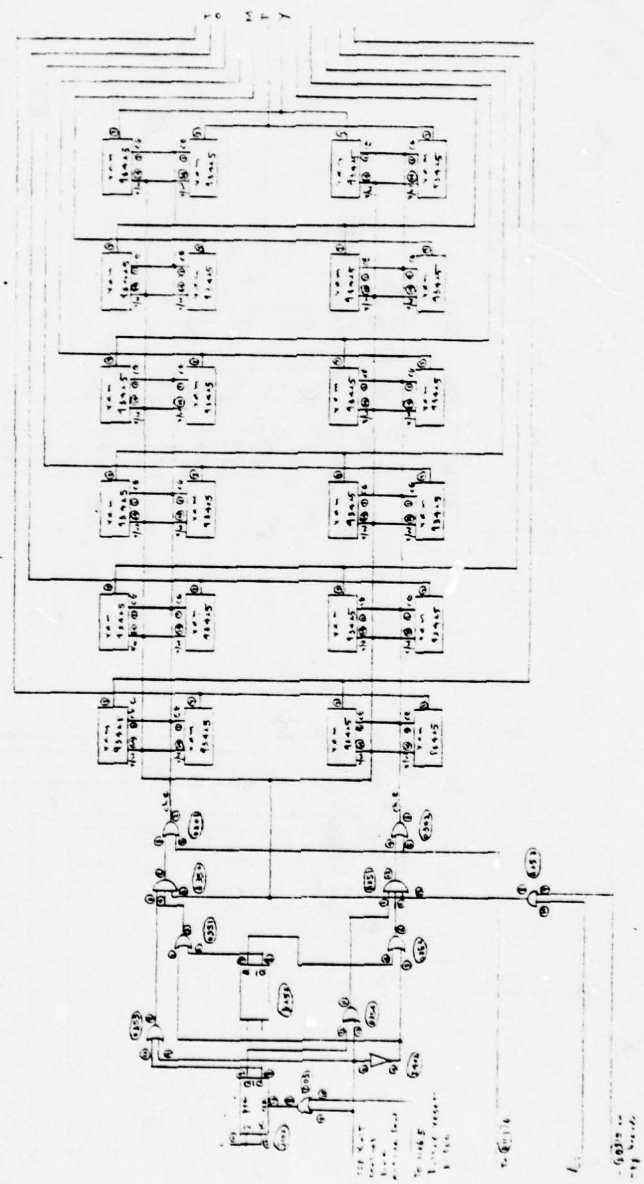


Figure 50 - High Speed RAMs Control Circuit

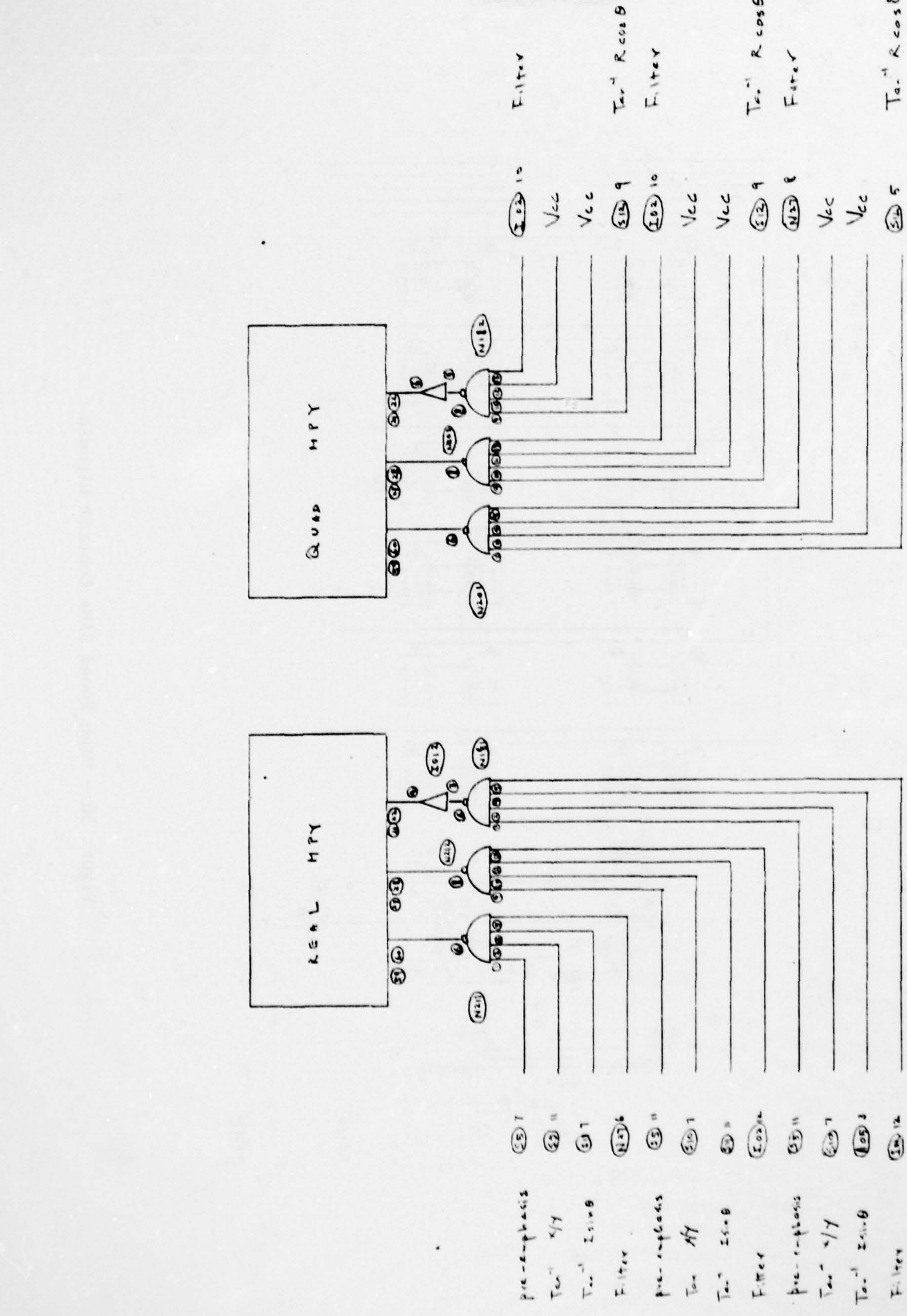


Figure 51 - Control Circuit for Multipliers

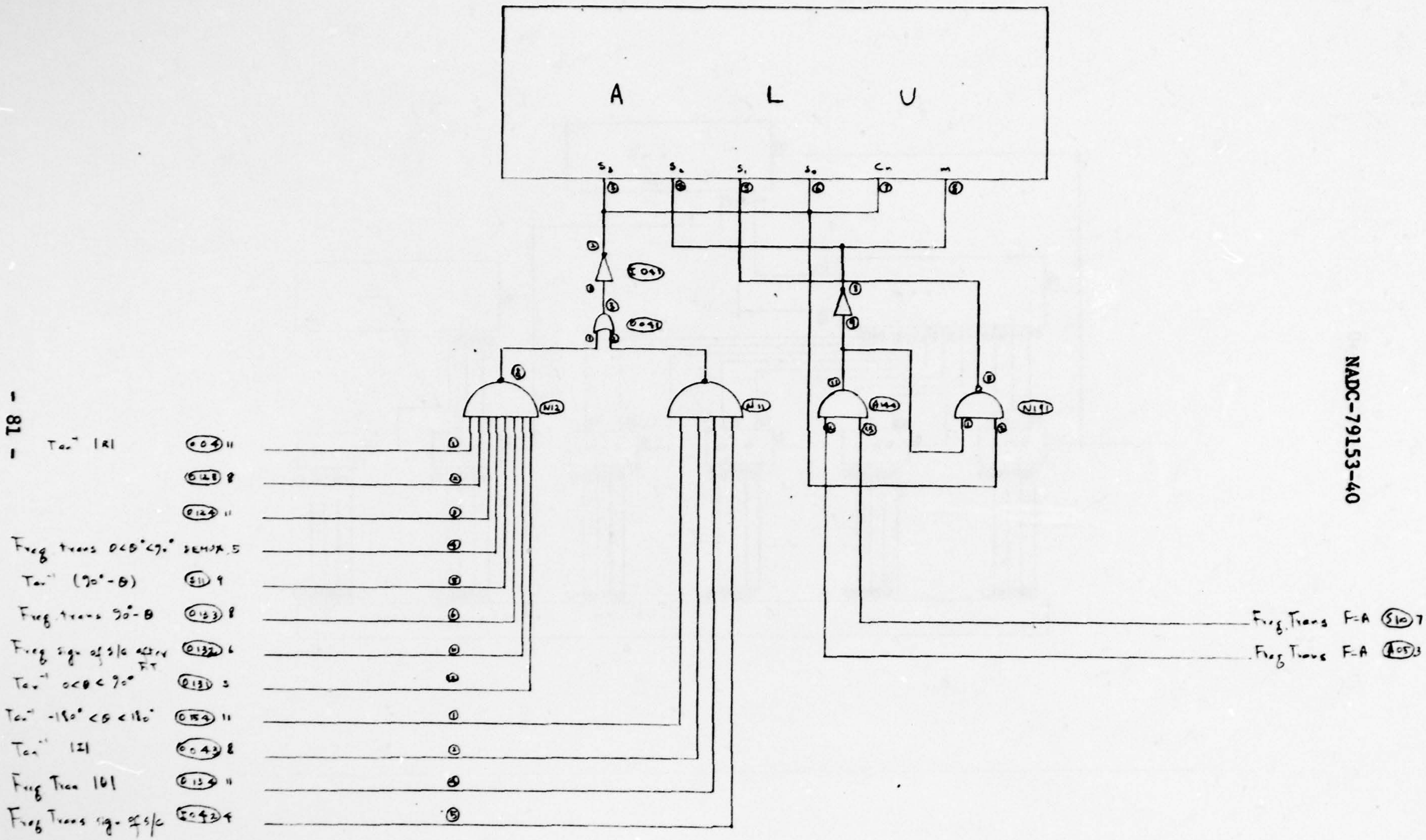


Figure 52 - Mode Control Circuit for Arithmatic Logic Units

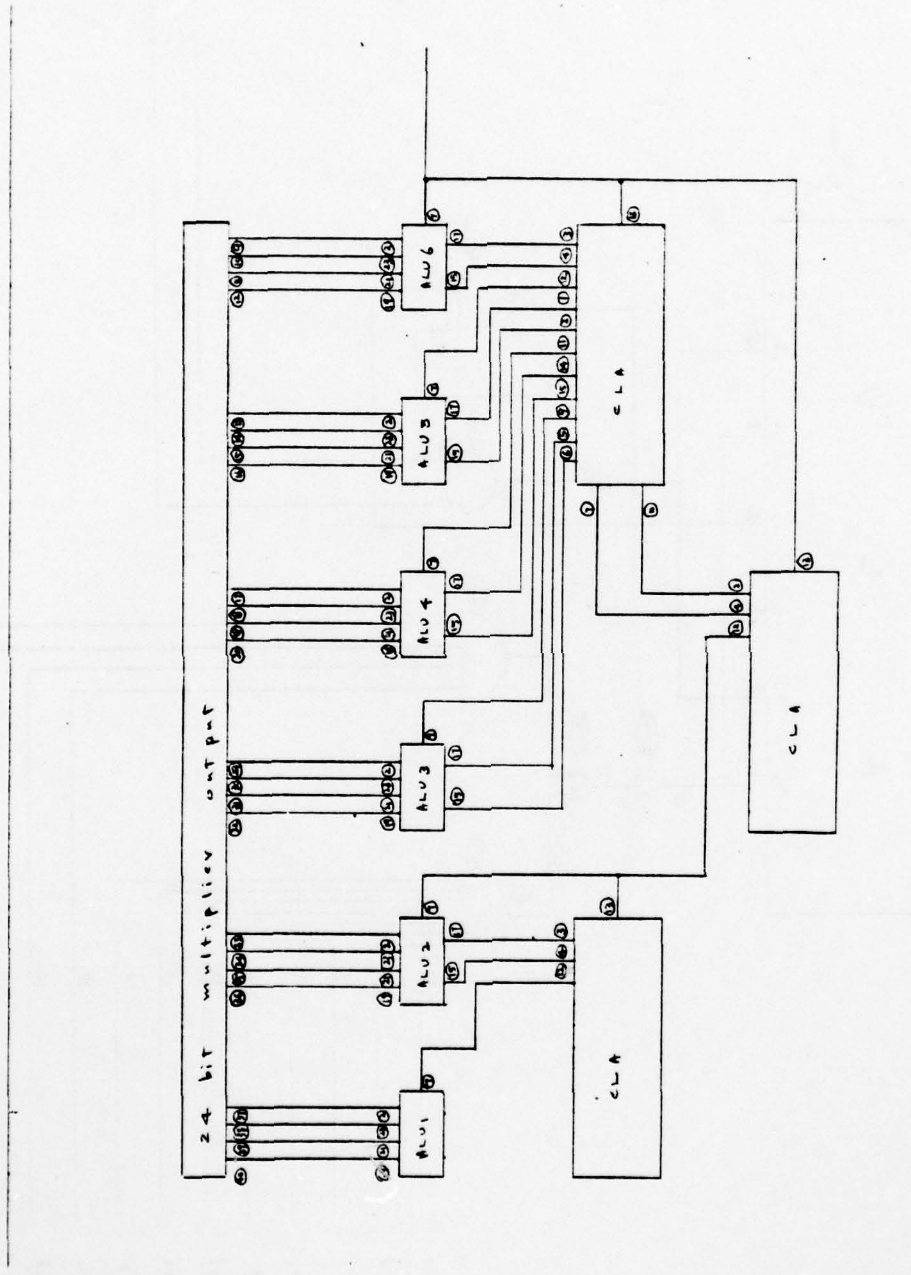


Figure 53 - Carry Control Circuit for Arithmatic Logic Units

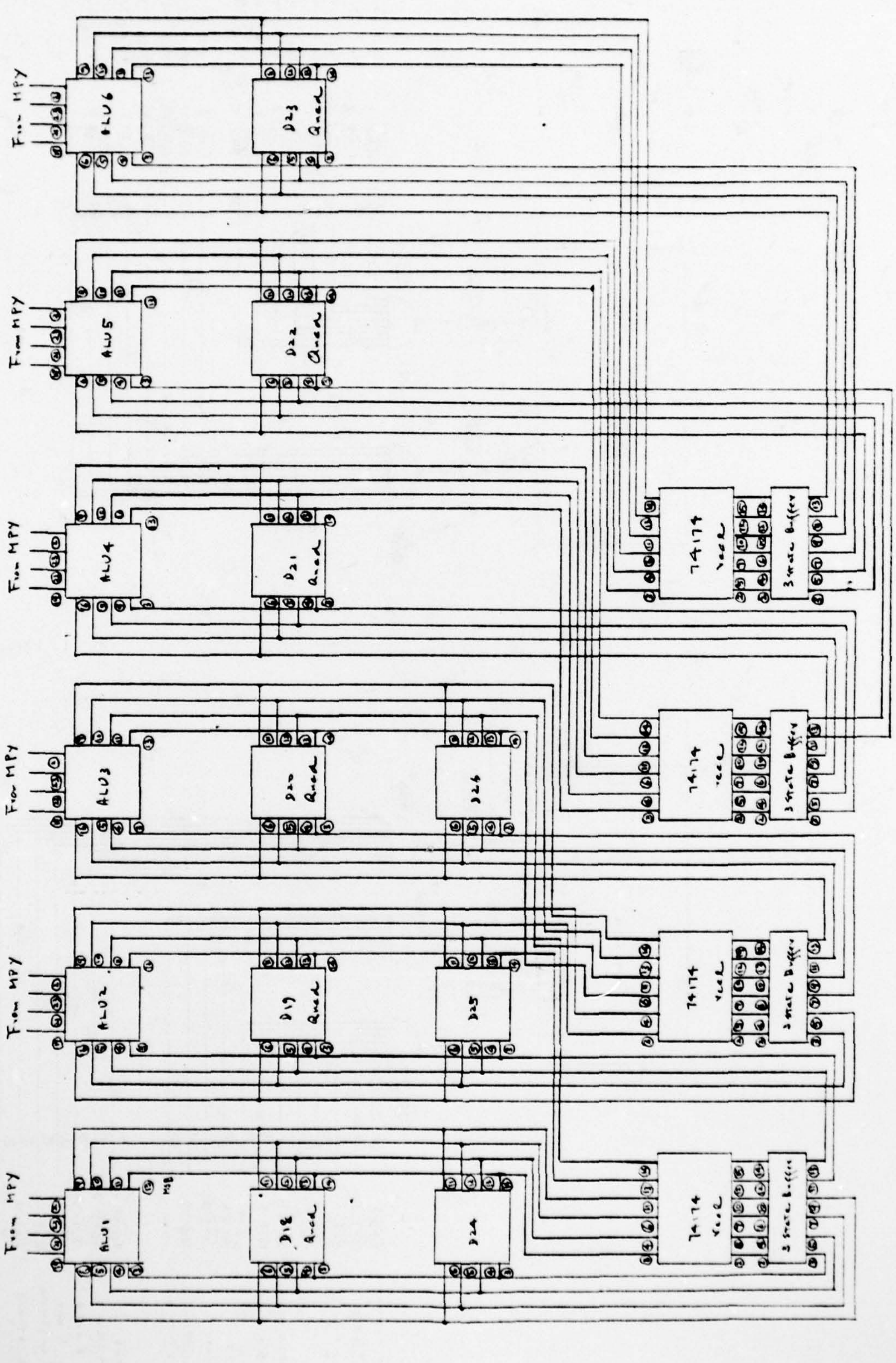


Figure 54 - I/O Circuit for ALUs and Associated Registers

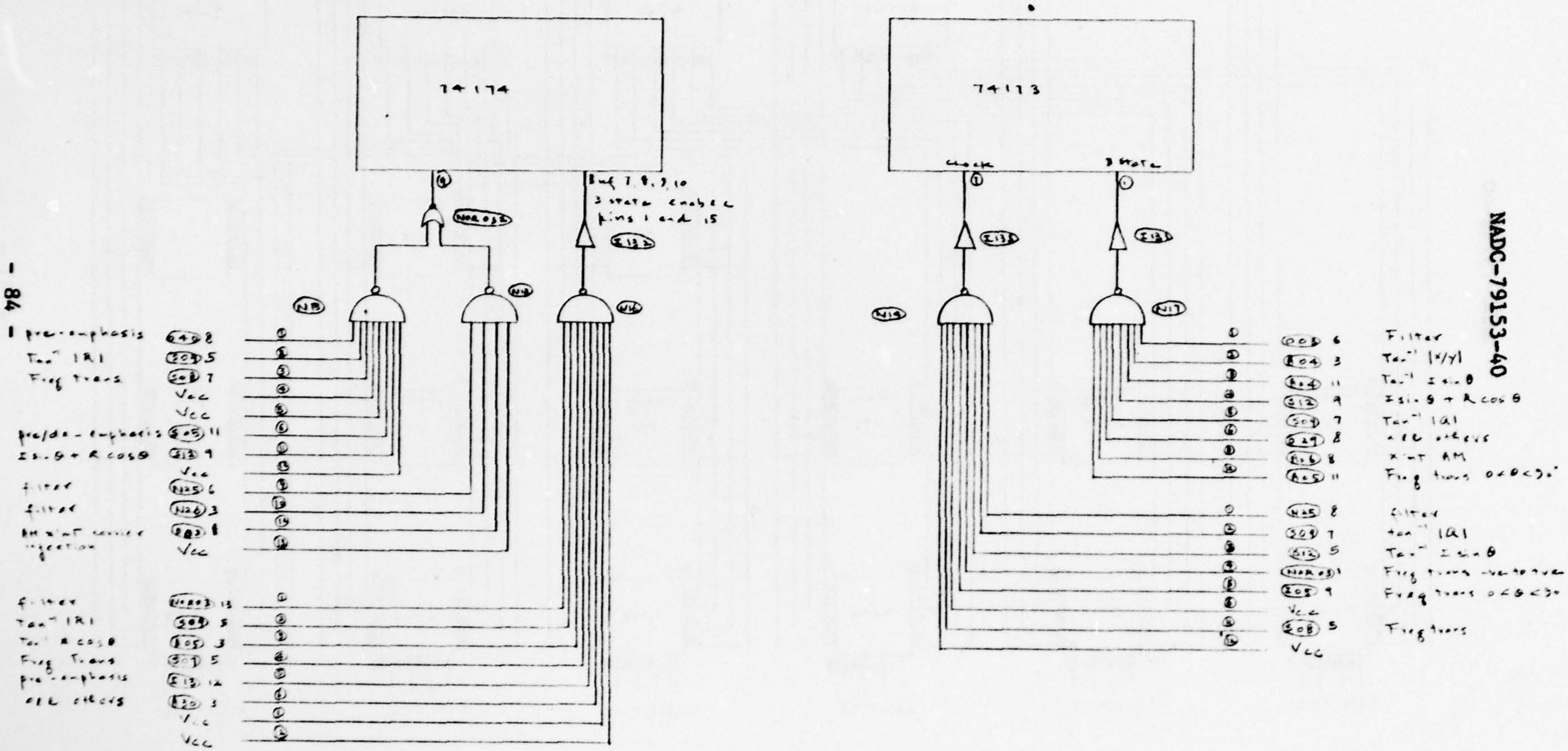


Figure 55 - Control Circuit for Registers

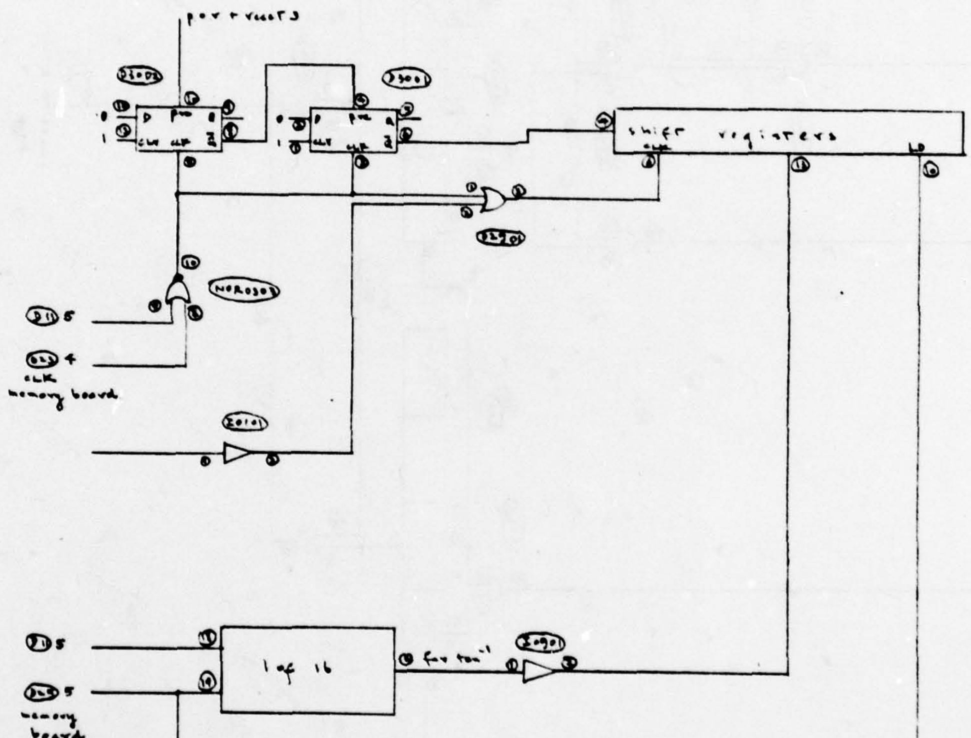


Figure 56 - Reprocessing Scheme for Filtered Data

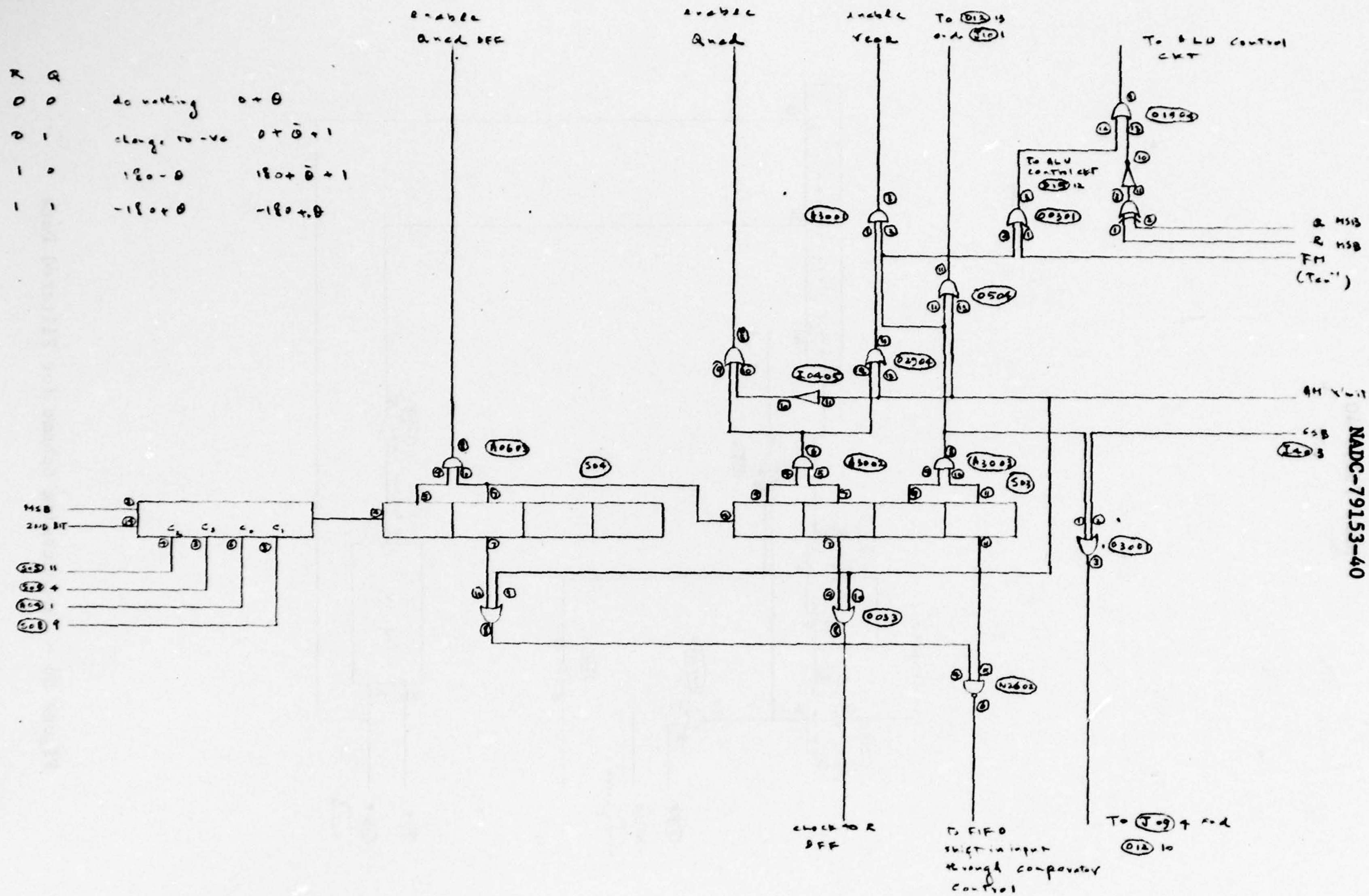


Figure 57 - Reprocessing of Filtered Data for AM Transmit, SSB Transmit/Receive and Arctangent Demodulation for FM Receive Modes

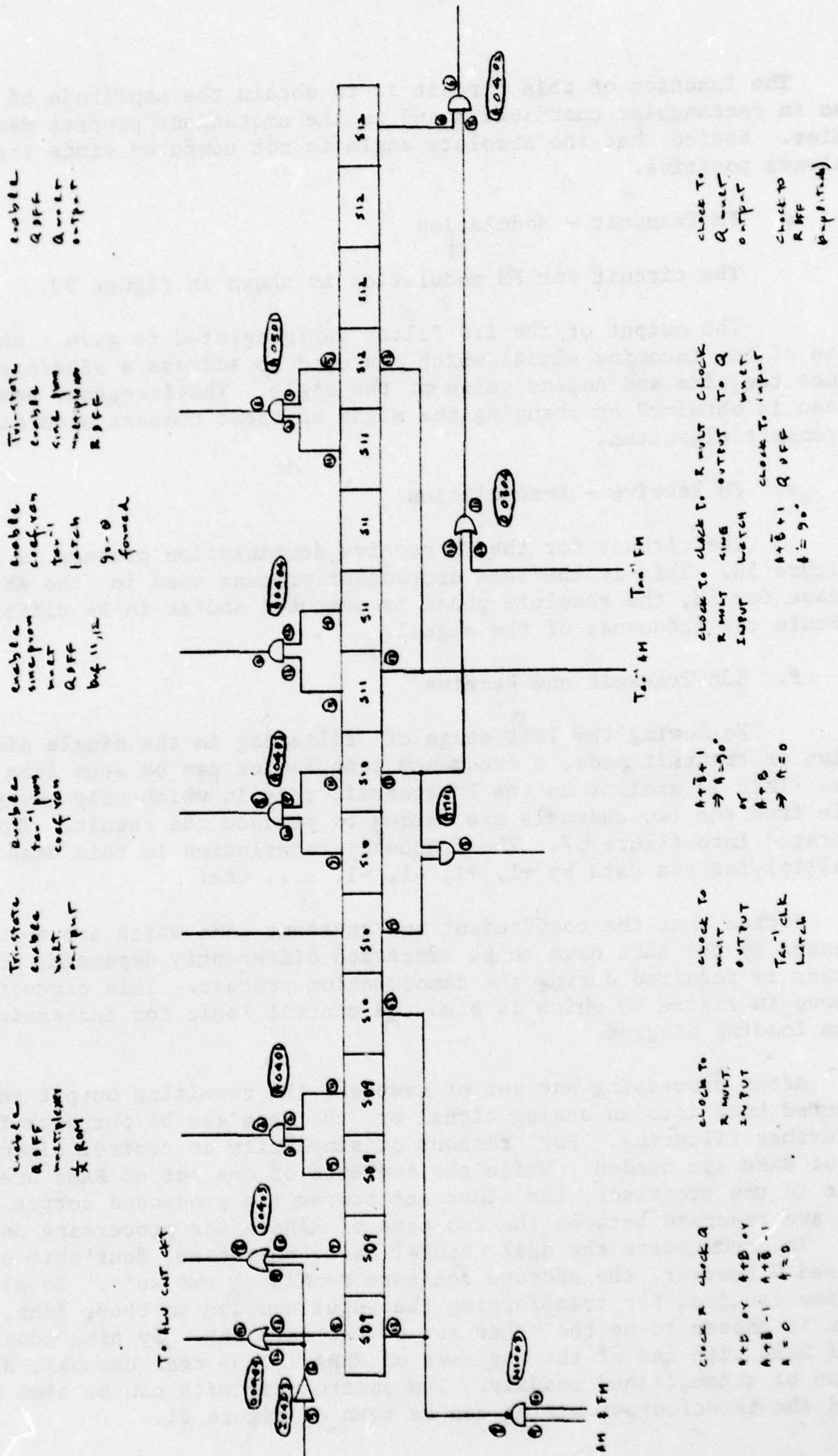


Figure 58 - Arctangent Demodulator Algorithm

The function of this circuit is to obtain the amplitude of a vector given in rectangular coordinates and in the arctangent process described earlier. Notice that the absolute angle is not computed since the amplitude is always positive.

d. FM Transmit - Modulation

The circuit for FM modulation is shown in figure 59.

The output of the 1:4 filter is integrated to give a phase representation of the incoming signal which is used to address a sine/cosine table to produce the sine and cosine value of the angle. The frequency translation process is obtained by changing the signs of these numbers according to the FM transmit algorithm.

e. FM Receive - Demodulation

The circuit for the FM receive demodulation process is incorporated in figure 58. This is the same arctangent process used in the AM case. In the case for FM, the absolute phase is computed and is to be differentiated to obtain the frequency of the signal.

f. SSB Transmit and Receive

Following the last stage of filtering in the single sideband receive or transmit mode, a frequency translation can be seen from the algorithm. This is similar to the FM transmit case in which only every other sample from the two channels are needed to produce the result. This is incorporated into figure 57. The frequency translation in this case is equivalent to multiplying the data by +1, +1, -1, -1, ... etc.

Notice that the coefficient and constant RAMs which are normally addressed by the ALUs have to be addressed differently depending on which constant is required during the demodulation process. This circuit diagram is shown in figure 60 which is also the control logic for addressing the RAM in the loading diagram.

After processing one set of samples, the resulting output can be D/A converted back into an analog signal or the data can be put back into memory for further filtering. For reasons of simplicity in control circuitry, two sets of RAMs are needed. While the contents of one set of RAMs are used as inputs to the processor, the other set stores the processed output. The roles are reversed between the two sets of RAMs after processing one set of data. To accommodate the dual channel case, therefore, four sets of RAMs are used. However, the address counters remain as two sets. To eliminate the time required for transferring the input samples to these RAMs, the input buffer is chosen to be the other set of the same RAM. By ping ponging this set of RAMs with one of the two sets of RAMs in the real channel, data transfer can be accomplished readily. The control circuits can be seen on figure 45 and the input/output wiring can be seen on figure 61.

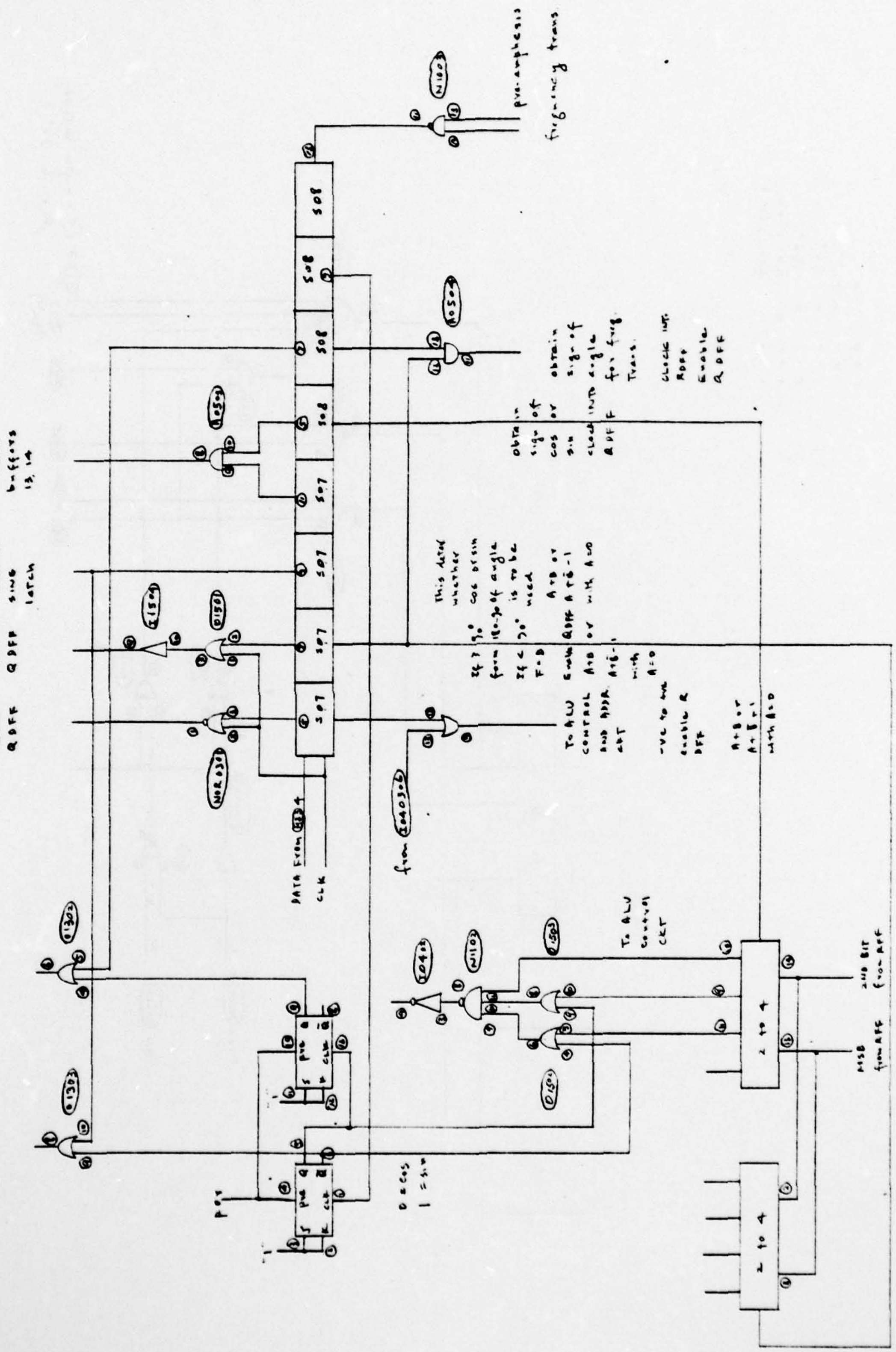


Figure 59 - FM Modulation Algorithm

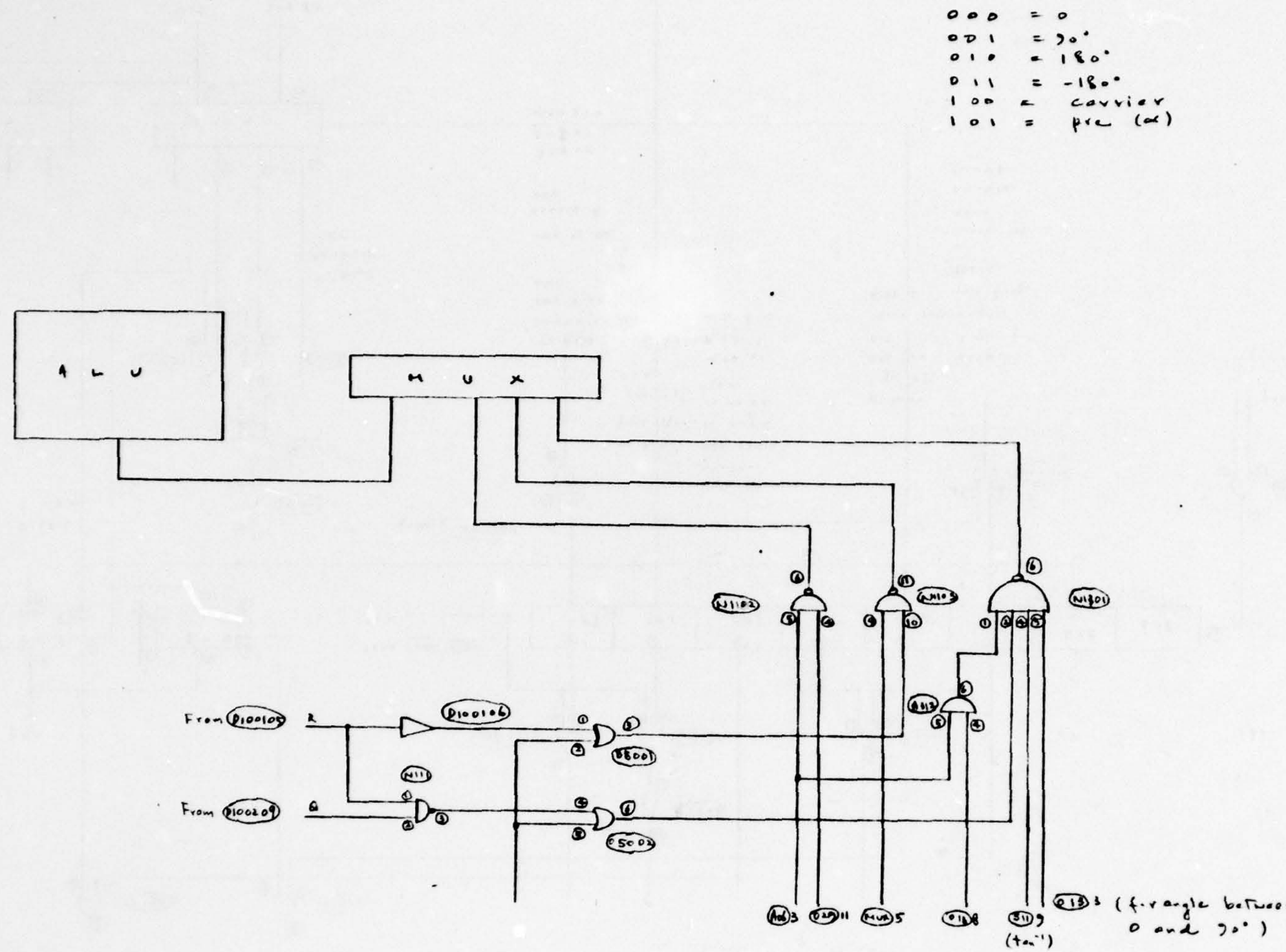


Figure 60 - Addressing Circuit for Constants and Coefficients RAMs

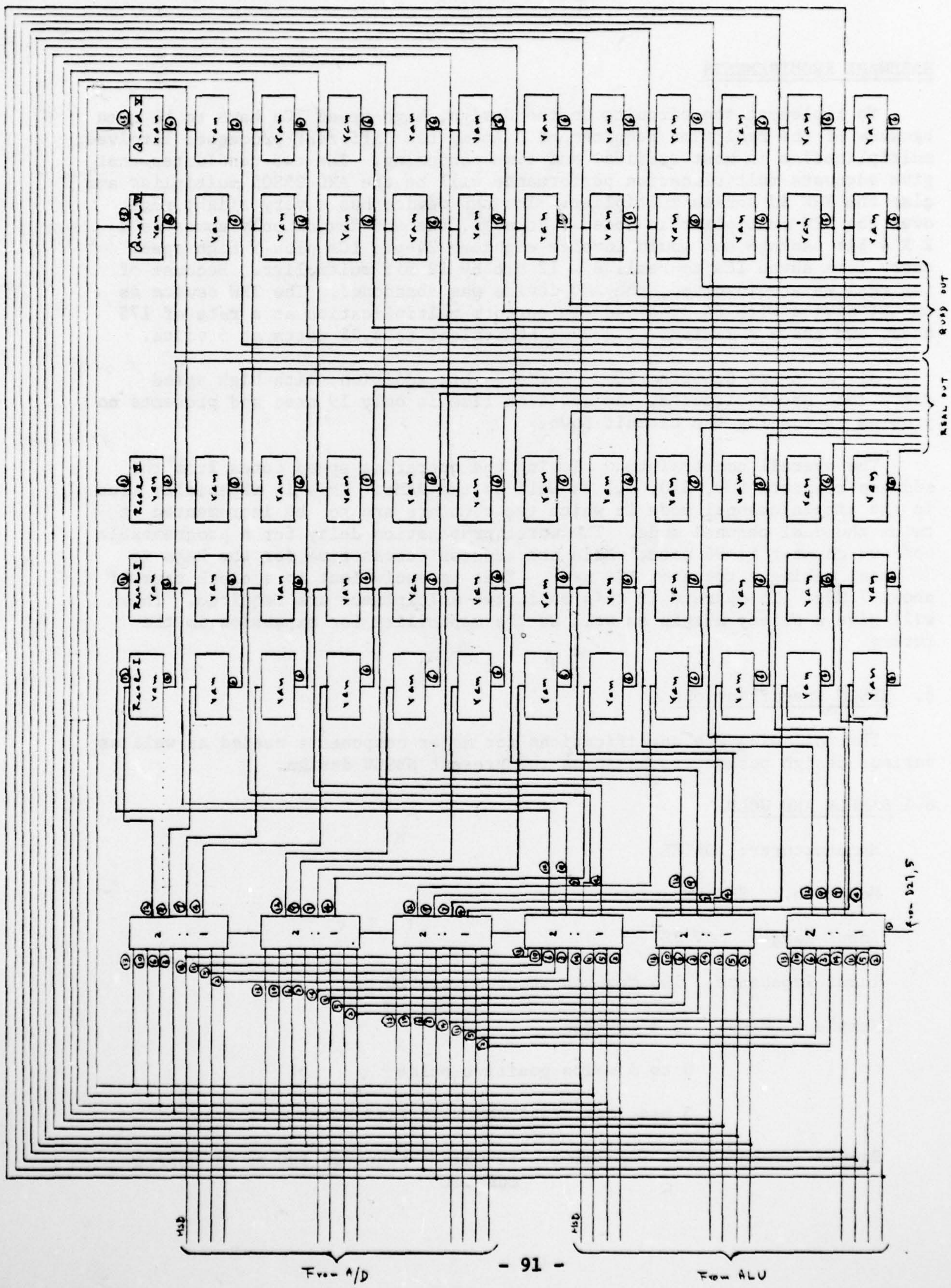


Figure 61 - I/O Circuit for Data Storage RAMs

HARDWARE REQUIREMENTS

To implement the circuits in the design, high speed ICs have to be used because of the real time requirement. Among the different processes involved, multiplication is most critical and time consuming. The two candidates that give adequate multiplication performance will be the AMD 25S05 multiplier and also the TRW SJ series multiplier. The AMD device has a very slight edge over the TRW multiplier in speed. However, the AMD device only comes as a 2 X 4 bit package and would involve eighteen 24-pin ICs plus 5 high speed carry look ahead ICs to realize a 12 bit by 12 bit multiplier. Because of the real estate involved, the AMD device was abandoned. The TRW device is an LSI device with 64 pins and can perform multiplication at a rate of 175 nsec, and draws a maximum of 850 ma equivalent to 4.25 watts at 5 volts.

All ALUs are Schottky ICs. For a 24 bit addition, with high speed carry look ahead circuits, the addition time is only 19 nsec and presents no problem of loading the circuit down.

The overall constraint on slowing the operating speed comes from the address counters B09, B10, B11 as well as the RAMs. This is especially true in the single channel mode in which the counters are to be incremented at twice the dual channel mode. The worst propagation delay for a programmable up/down counter is 36 nsec, while the address access time for the RAMs is 30 nsec, making a total of 138 nsec. This is equivalent to a clock rate of about 7 MHz. To operate at this rate, two multipliers are required. This will give a safety margin as well as the capability for expansion in the future.

6. NBSCU Specifications

The following are specifications for major components needed as well as various design parameters based on the present NBSCU design.

6.1 SAMPLE AND HOLD

Manufacturer: DATEL

Model No.: SHM-UH

Input Range: ±5V FS

Input Impedance: 100 Megohms shunted with 20 pf.

Sample command: 35 ± 10 nsec

0 to 5 volts positive pulse

3 nsec max. rise and fall time

Sample command Input Impedance: 50 ohms - requires 100 ma of source current

Output Voltage Range: ±5V FS

Output Current: ±30 ma max.

Output Impedance: 10 ohms, max.

Acquisition Time: 35 nsec for full scale input

Aperture Time: 200 psec.

Output Slewing Rate: 500V/μsec

Maximum Sample Rate: 10 MHz

Hold Decay Rate: 50 μV/μsec

6.2 <u>A/D Converter</u>

Manufacturer: DATEL

Model Number: ADC-G3C

Output Coding: 2's complement (reverse coding)

No. of Bits: 10

Analog Input Range: ±5 volts

Input Impedance: 2KΩ

Start Conversion: 2V min. to 7V max. positive pulse.
30 nsec min. width
"1" = Resets
"0" = Starts Conversion

End of Conversion Output: 0 = Conversion complete
1 = Reset and Conversion Period

Conversion Rate: 1 MHz

6.3 <u>D/A Converter</u>

Manufacturer: DATEL

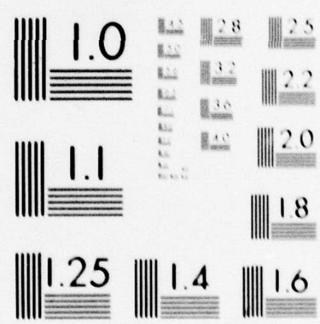
Model No.: DAC-VR 12B3C

Input Coding: 2's complement

AD-A073 401 NAVAL AIR DEVELOPMENT CENTER WARMINSTER PA COMMUNICA--ETC F/G 17/2
TIES NARROWBAND SIGNAL CONVERSION UNIT DESIGN REPORT.(U)
JUL 79 Y LUI
UNCLASSIFIED NADC-79153-40 NL

2 OF 2
AD A073401

END
DATE FILMED
10-79
DDC



MICROCOPY RESOLUTION TEST CHART

NATIONAL BUREAU OF STANDARDS-1963-A

Input Loading: 1 TTL load

Data Strobe: Information must be present at the register inputs of the D/A converter prior to strobing. Transfer of information to the output pins occurs when the strobe input goes from 1 to 0.

Update Rate: 10 MHz max.

Output Voltage: ±5 volts

Output Settling Time: <2 μsec to ± 0.025% of FS

6.4 Processor

Input Buffer Size: 256, 12 bit words

Scratch pad memory size: 1,024, 12 bit words

Macroinstruction memory size: Sixteen 36 bit words

Multiplication Rate: 10 MHz

Addition Rate: 50 MHz

Maximum Sampling Rate: 5 KHz - 500 KHz

No. of IC's used: 360

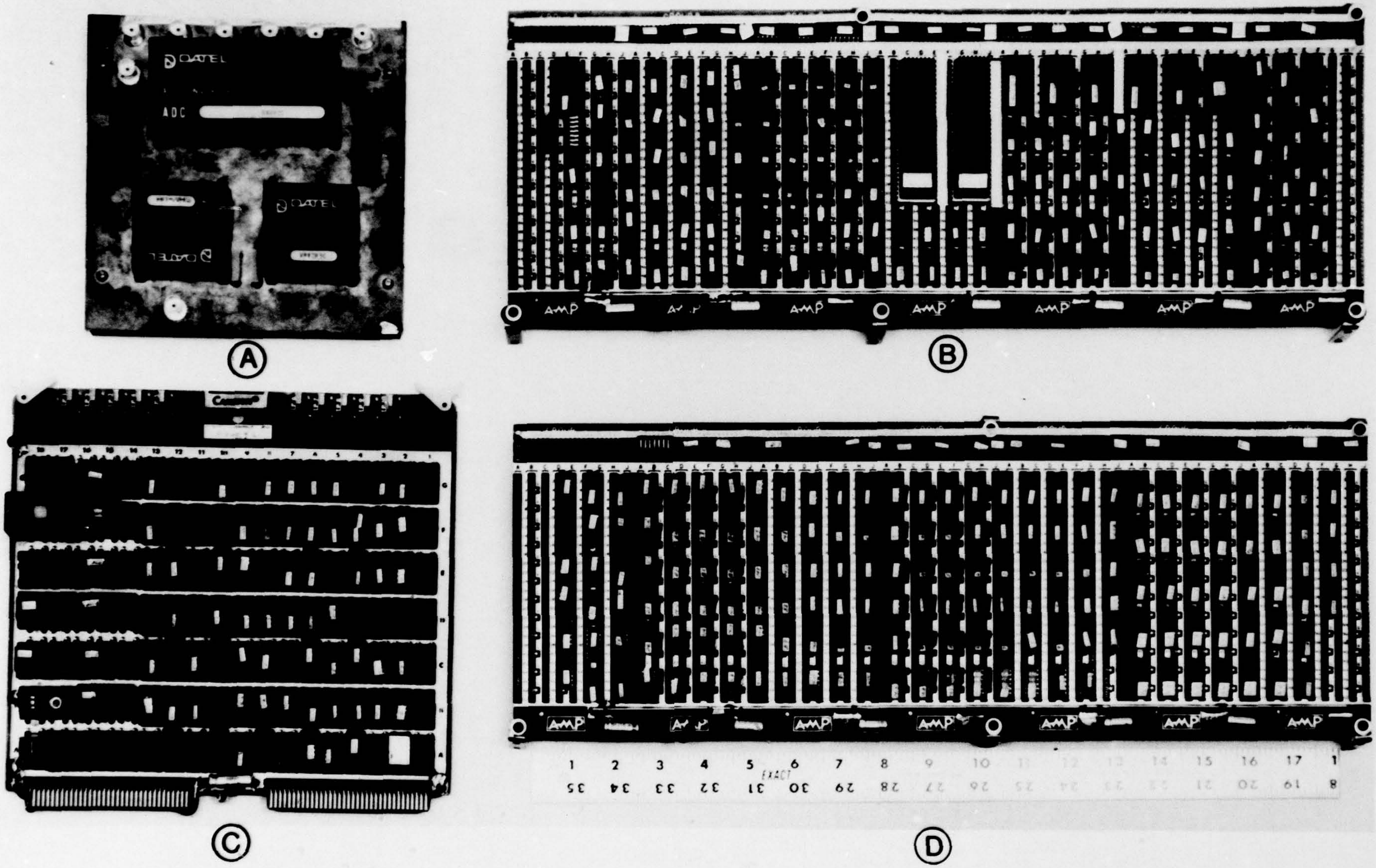
Power Supply: 5 volt, -12 volts

Current drawn: 20 amperes at 5 volt
70 ma at -12 volts

Output Buffer Size: 64, 12 bit words (FIFOs)

No. of Circuit Boards: One 19" x 8" Multiplier - ALU Board
One 19" x 8" Memory Board
One 10" x 10" I/O and Control Board
One 8" x 8" Analog-Digital Conversion Board
These boards are shown in figure 62.

TIES NBSCU (NADC)



(A) ANALOG-DIGITAL CONVERSION BOARD
(B) MULTIPLER-ALU BOARD
(C) I/O AND CONTROL BOARD
(D) MEMORY BOARD

Figure 62 - Circuit Boards for TIES NBSCU

APPENDIX A

NOISE SOURCES

In the course of digital processing of signals, there are many places that noise may be introduced that may affect the quality of the signal. In the following paragraphs, some of these noise sources are investigated.

A.1 Noise Sources from a Sampling Process

The quality of the signal begins to deteriorate the moment it is sampled — assuming sampling is high enough in frequency so that there is no aliasing effect. The first group of noise sources come from the sample-and-hold circuit which freezes the fast moving signal for the A/D converter to make the conversion. Those noise sources are listed below:

1) OFFSET - This is the extent to which the output deviates from zero with zero input and is usually a function of time and temperature and can be adjusted.

2) NONLINEARITY - This is the amount by which the output versus input deviates from a straight line.

3) SCALE FACTOR ERROR - This is the amount by which the output deviates from specified gain (usually unity). This parameter can also be adjusted.

4) APERTURE UNCERTAINTY - This is the time elapsed between the command to hold and the actual opening of the hold switch.

5) DROOP - This is the drift of the output caused by the discharge of the storage capacitor.

Among the above noise sources from the sample-and-hold, the most important ones are the aperture uncertainty and the droop.

The aperture uncertainty has two components — a nominal time delay, and an uncertainty caused by jitter or variation from time to time and unit to unit and is shown in figure A1.

The aperture uncertainty of a sample-and-hold unit dictates the rate that the input signal can vary given that the signal is to be resolved to a certain desired minimum value as in the case of an A/D converter. Assuming that a signal of the form A sin ωt is sampled by a sample-and-hold circuit. The rate at which the signal varies with time will be the time derivative of the signal and is given by the following equation:

$$\frac{d}{dt}A \sin \omega t = \omega A \cos \omega t \quad \text{volts/sec}$$

If an A/D converter is to be used to put the sampled value into 10 binary bits, the aperture error should not cause the output to vary more than 1/2 LSB. Assuming the A/D converter has 1 volt as its full scale, therefore 1/2 LSB = 2^{-11} volts. The maximum rate at which the input signal can vary is ωA volts/sec, and therefore, the maximum aperture error is $2^{-11}/\omega A$ seconds. Therefore, to specify the aperture error, one must know the weight of the LSB of the A/D converter and the strength of the incoming signal. Notice that if ωA becomes larger indicating that the frequency is larger (A is a constant), a smaller aperture uncertainty is required.

As aperture time is important to fast moving signals, the droop is important to slow moving waveforms. A typical diagram for the droop is shown in figure A2 and is the effect caused by the discharge of the capacitor in the hold circuit. Since slow moving signals imply a slower sampling rate, the capacitor will have more time to discharge than in higher rate sampling circuits. Therefore, given the LSB for the A/D converter using in conjunction with the sample-and-hold with a certain droop, the minimum sampling rate can be determined. For example, assume a sample and hold has a droop rate of 50 μv/μsec and a 1 volt full scale 10 bit A/D converter has 2^{-11} volt as the weight of half its LSB. The minimum sampling rate should be:

$$f = [(2^{-11}\ \text{volt})(1\ \text{sec}/50\ \text{volt})]^{-1} = 102.4\ \text{KHz}$$

Sampling less than 102.4 KHz may yield an erroneous result at the output of the A/D converter.

The source of error coming from the A/D converter is known as quantization noise. The noise arises because of the fact that only a finite number of bits are used to represent an analog voltage. This is known as truncation error, or roundoff error depending on the scheme used to obtain the fixed binary number.

For two's complement representation, the error E_T for truncation was found to be $0 \geq E_T > -2^{-b}$. Rounding off a two's complement number will produce an error given by the following expression:

$$-(1/2)(2^{-b}) < E_R \leq (1/2)(2^{-b})$$

The mean and variance of the quantization noise are shown to be

$$m_e = 0$$

$$\sigma_e^2 = \Delta^2/12 = 2^{-2b}/12$$

for rounding and

$$m_e = -2^{-b}/2$$

$$\sigma_e^2 = 2^{-2b}/12$$

for two's complement truncation.

If we assume that the signal power to the A/D converter is σ_x^2, then the signal to noise ratio produced by the quantization will be $\sigma_x^2/\sigma_e^2 = (12/2^{-2b})\sigma_x^2$.

Expressing in terms of db, we have the following:

$$SNR = 10 \log_{10} (\sigma_x^2/\sigma_e^2) = 6b + 10.8 + 10 \log_{10} (\sigma_x^2)$$

It can be seen that the signal to noise ratio increases 6 db for each bit added to the register length. To prevent the signal from exceeding the dynamic range of the A/D converter, the incoming signal can be scaled. Let A be the scaling factor, the resulting signal to noise ratio will then be:

$$10 \log_{10}\left(\frac{A^2\sigma_x^2}{\sigma_e^2}\right) = 6b + 10.8 + 10 \log_{10} (\sigma_x^2) + 20 \log_{10} (A)$$

Since A is less than 1, 20 $\log_{10}$ (A) is negative and so by reducing the input amplitude, the signal to noise ratio is reduced. If A is assumed to be $(1/4)\sigma_x$, the signal to noise ratio is (6b-1.24)db. To obtain a signal to noise ratio of 60 db, 10 bits are required.

The effect of quantization for the coefficients is slightly different from that of the A/D conversion process. In the A/D conversion process, noise is introduced into the signal. In the quantization of the filter coefficients, the effect is a change from the ideal frequency response.

A.2 FIR Filter Noise

The filtering process of a signal is given by the following equation:

$$y(n) = \sum_{k=0}^{N-1} h(k)\, X(n-k)$$

which is a convolution sum process. The direct realization of such a system is a direct realization of the convolution sum process. The flow graph of the realization is shown in figure A3.

Since the multiplication of two N bit numbers produces a number with 2N bits, the product may be truncated to reduce to N bits. This corresponds to the addition of noise to the filter. Assuming that each product is rounded off before summation, the output noise is

$$f(n) = \sum_{k=0}^{N-1} e_k(n)$$

and the variance of the output noise is

$$\sigma_f^2 = M \frac{2^{-2b}}{12}$$

where M = number of coefficients

Notice that the noise is cumulative and depends on the number of coefficients used for the filter. Given the signal power, the signal to noise ratio for a fixed number of bits used can be computed. Notice also that if all 2N bits are kept for addition with the accumulator output truncated to N bit, the variance is reduced to $2^{-2b}/12$ which is identical to the noise produced by the A/D conversion process.

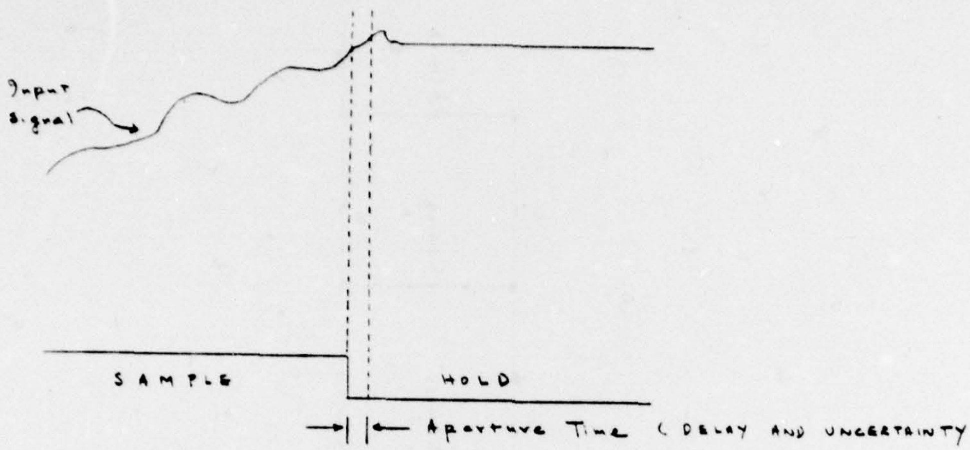


Figure A1 - Aperture Error in Sample-and-Holds

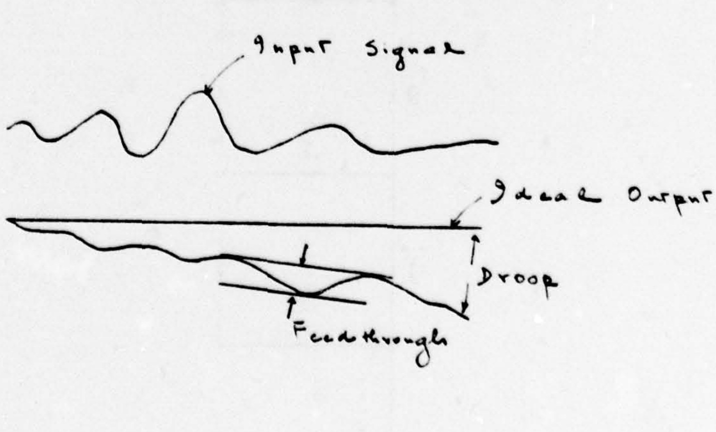


Figure A2 - Droop in Sample-and-Holds

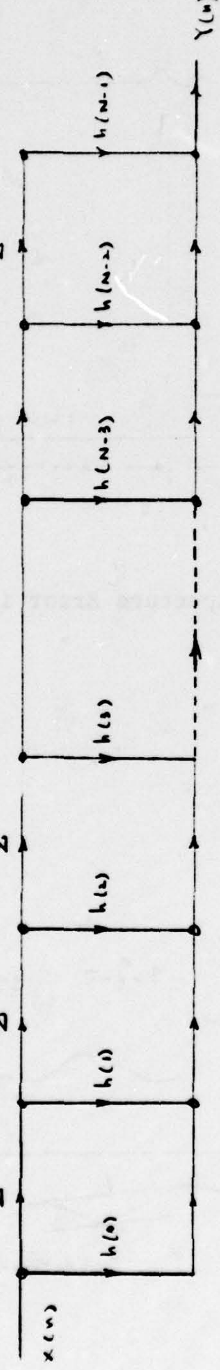


Figure A3 - Flow Graph for FIR Filter

APPENDIX B

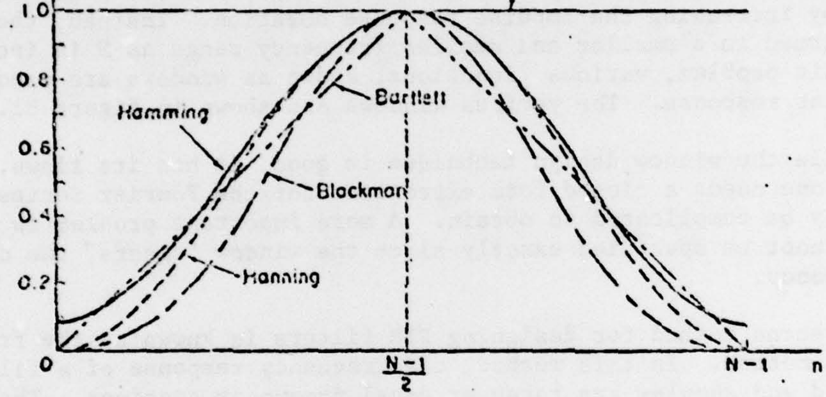
FIR FILTER DESIGN

There are many ways that finite impulse response filters can be designed. One of the design techniques is by the Fourier Series method in which the Fourier Series representation of the frequency response of the filter is computed and used as coefficients for the filter. Since the Fourier series have infinite representations, these coefficients have to be truncated. However, direct truncation leads to the Gibbs phenomenon which manifests itself as a fixed percentage overshoot and ripple before and after an approximated discontinuity in the frequency response. The size of these overshoots does not reduce by increasing the impulse response duration. Instead, the overshoot is confirmed to a smaller and smaller frequency range as N is increased. To solve this problem, various functions, known as windows are used to smooth the filter response. The various windows are shown in figure B1.

While the window design technique is good, it has its flaws. One problem is that one needs a closed form expression for the Fourier Series coefficient which may be complicated to obtain. A more important problem is that the band edges cannot be specified exactly since the window "smears" the discontinuity in frequency.

A second method for designing FIR filters is known as the frequency sampling method. In this method, the frequency response of a filter is specified and samples are taken at equal frequency spacings. The impulse response is obtained by using the interpolation equation. In the frequency sampling method, in order to obtain the best result it is important that the value for the transition region be an optimum value. To do the optimization problem, a series of equations have to be written and solved mathematically for the solution. This involves linear programming and a computer is needed as well as a simplex algorithm for minimization. The disadvantage to this design technique is that the selection of N as well as the selection of transition samples are based on experience and requires iterations of the computer program to find the optimal transition samples. Notice also that the ripples cannot be specified.

A third, and recommended technique for designing digital FIR filters is by means of the Remez Exchange Algorithm. The filters designed by using this technique have proved to be optimal filters. Optimal in the sense that the peak approximation error over the entire interval of approximation is minimized. This program was run on the NADC CDC 6600 computer and used for the design of the various filters and differentiator in the algorithms. The program is listed in the following pages.



Window	Peak Amplitude of Side Lobe (dB)	Transition Width of Main Lobe	Minimum Stopband Attenuation (dB)
Rectangular	−13	$4\pi/N$	−21
Bartlett	−25	$8\pi/N$	−25
Hanning	−31	$8\pi/N$	−44
Hamming	−41	$8\pi/N$	−53
Blackman	−57	$12\pi/N$	−74

Figure B1 - Window Functions

```
      PROGRAM REMEXC (INPUT,OUTPUT)
C
C
C
C     PROGRAM FOR THE DESIGN OF LINEAR PHASE FINITE IMPULSE
C     RESPONSE (FIR) FILTER USING THE REMEZ EXCHANGE ALGORITHM
C     THREE TYPES OF FILTERS ARE INCLUDED--BANDPASS FILTERS
C     DIFFERENTIATORS, AND HILBERT TRANSFORM FILTERS.
C
C     THE INPUT DATA CONSISTS OF 5 CARDS.
C
C     CARD 1--FILTER LENGTH, TYPE OF FILTER (1-MULTIPLE
C     PASSBAND/STOPBAND, 2-DIFFERENTIATOR, 3-HILBERT TRANSFORM
C     FILTER), NUMBER OF BANDS, AND GRID
C     DENSITY.  (FORMAT - 4I3)
C
C     CARD 2--BANDEDGES, LOWER AND UPPER EDGES FOR EACH BAND
C     WITH A MAXIMUM OF 8 BANDS.  (FORMAT - 16F5.4)
C
C     CARD 3--DESIRED FUNCTION (OR DESIRED SLOPE IF A
C     DIFFERENTIATOR) FOR EACH BAND.  (FORMAT - 8F5.2)
C
C     CARD 4--WEIGHT FUNCTION IN EACH BAND.  FOR A
C     DIFFERENTIATOR, THE WEIGHT FUNCTION IS INVERSELY
C     PROPORTIONAL TO F.  (FORMAT - 8F5.2)
C
C     CARD 5--RIPPLE AND ATTENUATION IN PASSBAND AND STOPBAND.
C     THIS CARD IS OPTIONAL AND CAN BE USED TO SPECIFY LOWPASS
C     FILTERS DIRECTLY IN TERMS OF PASSBAND RIPPLE AND STOPBAND
C     ATTENUATION IN DB.  THE FILTER LENGTH IS DETERMINED FROM
C     THE APPROXIMATION RELATIONSHIPS GIVEN IN :
C     L. R. RABINER, APPROXIMATE DESIGN RELATIONSHIP FOR
C     LOWPASS DIGITAL FILTERS, IEEE TRANS. ON AUDIO AND
C     ELECTROACUSTICS, VOL. AU-21, NO. 5, OCTOBER 73.
C     ****WHEN THIS OPTION IS USED THE FILTER LENGTH ON
C     CARD 1 SHOULD BE SET TO 0.****  (FORMAT - 2D15.8)
C
C     SAMPLE INPUT DATA SETUP
C
C     THE FOLLOWING INPUT DATA SPECIFIES A LENGTH 32 BANDPASS
C     FILTER WITH STOPBANDS 0 TO 0.1 AND 0.425 TO 0.5, AND
C     PASSBAND FROM 0.2 TO 0.35 WITH WEIGHTING OF 10 IN THE
C     STOPBANDS AND 1 IN THE PASSBAND.
C     THE GRID DENSITY IS 32.
C     032001003032
C     .0000.1000.2000.3500.4250.5000
C     .00001.000.0000
C     10.001.00010.00
C
C     THE FOLLOWING INPUT DATA SPECIFIES A LENGTH 32 WIDEBAND
C     DIFFERENTIATOR WITH SLOPE 1 AND WEIGHTING OF 1/F.  THE GRID
C     DENSITY IS ASSUMED TO BE 16.
C     032002001000
C     .0000.5000
C     1.000
```

```
PROGRAM     REMEXC    CDC 6600 FTN V3.0-P380 OPT=1  79/03/29. 13.28.54.   PAGE       2
          C      1.000
          C
          C      THE FOLLOWING INPUT DATA SPECIFIES A LOWPASS FILTER
          C      BY GIVING DIRECTLY THE RIPPLE AND ATTENUATION.
          C      000001002032
          C      .0000.2000.3000.5000
          C      1.000.0000
          C      1.0001.000
          C      0.20000000000000040.000000000000000
          C
                 COMMON DES,WT,ALPHA,IEXT,NFCNS,NGRID,PI2,AD,DEV,X,Y,GRID
                 DIMENSION IEXT(66),AD(66),ALPHA(66),X(66),Y(66)
                 DIMENSION H(66)
                 DIMENSION DES(1045),GRID(1045),WT(1045)
                 DIMENSION EDGE(20),FX(10),WTX(10),DEVIAT(10)
                 DIMENSION OMEGA(50),RESPA(50)
                 DOUBLE PRECISION OMEGA,SUMAR,SUMAC,RESPA,ATTN,RIPPLE,SVRIP,DELTAF
                 DOUBLE PRECISION PI2,PI
                 DOUBLE PRECISION AD,DEV,X,Y
                 PI2=6.283185307179586
                 PI=3.141592653589793
          C
          C      THE PROGRAM IS SET UP FOR A MAXIMUM LENGTH OF 128, BUT
          C      THIS UPPER LIMIT CAN BE CHANGED BY REDIMENSIONING THE
          C      ARRAYS IEXT, AD, ALPHA, X, Y, H TO BE NFMAX/2 + 2.
          C      THE ARRAYS DES, GRID, AND WT MUST BE DIMENSIONED
          C      16(NFMAX/2 + 2).
          C
                 NFMAX=128
             100 CONTINUE
                 JTYPE=0
          C
          C      PROGRAM INPUT SECTION
          C
                 READ1,NFILT,JTYPE,NBANDS,LGRID
               1 FORMAT (4I3)
                 IF(NFILT.GT.NFMAX.OR.NFILT.EQ.3) CALL ERROR
                 IF(NBANDS.LE.0) NBANDS=1
          C
          C      GRID DENSITY IS ASSUMED TO BE 16 UNLESS SPECIFIED
          C      OTHERWISE
          C
                 IF(LGRID.LE.0) LGRID=16
                 JB=2*NBANDS
                 READ2, (EDGE(J),J=1,16)
               2 FORMAT (16(F5.4))
                 READ3, (FX(J),J=1,8)
               3 FORMAT (8(F5.2))
                 READ4, (WTX(J),J=1,8)
               4 FORMAT (8(F5.2))
                 IF(JTYPE.EQ.0) CALL ERROR
                 IF(NFILT.EQ.0) GO TO 10
                 NROX=0
                 GO TO  11
              10 READ5,RIPPLE,ATTN
```

```
    5 FORMAT (2D15.8)
      RIPPLE=10**(RIPPLE/20)
      ATTN=10**(-ATTN/20)
      RIPPLE=(RIPPLE-1)/(RIPPLE+1)
      RIPPLE=DLOG10(RIPPLE)
      ATTN=DLOG10(ATTN)
      SVRIP=RIPPLE
      RIPPLE=(0.005309*(RIPPLE**2)+0.07114*RIPPLE-0.4761)*ATTN
     1+(-0.00266*(RIPPLE**2)-0.5941*RIPPLE-0.4278)
      ATTN=11.01217+0.51244*SVRIP-0.51244*ATTN
      DELTAF=EDGE(3)-EDGE(2)
      NFILT=RIPPLE/DELTAF-ATTN*(DELTAF**2)+1
      NROX=1
   11 NEG=1
      IF(JTYPE.EQ.1) NEG=0
      NODD=NFILT/2
      NODD=NFILT-2*NODD
      NFCNS=NFILT/2
      IF(NODD.EQ.1.AND.NEG.EQ.0) NFCNS=NFCNS+1
C
C     SET UP THE GRID DENSITY.  THE NUMBER OF POINTS IN THE GRID
C     IS (FILTER LENGTH + 1)*GRID DENSITY/2
C
      GRID(1)=EDGE(1)
      DELF=LGRID*NFCNS
      DELF=0.5/DELF
      IF(NEG.EQ.0) GO TO 135
      IF(EDGE(1).LT.DELF) GRID(1)=DELF
  135 CONTINUE
      J=1
      L=1
      LBAND=1
  140 FUP=EDGE(L+1)
  145 TEMP=GRID(J)
C
C     CALCULATE THE DESIRED MAGNITUDE RESPONSE AND THE WEIGHT
C     FUNCTION ON THE GRID
C
      DES(J)=EFF(TEMP,FX,WTX,LBAND,JTYPE)
      WT(J)=WATE(TEMP,FX,WTX,LBAND,JTYPE)
      J=J+1
      GRID(J)=TEMP+DELF
      IF(GRID(J).GT.FUP) GO TO 150
      GO TO 145
  150 GRID(J-1)=FUP
      DES(J-1)=EFF(FUP,FX,WTX,LBAND,JTYPE)
      WT(J-1)=WATE(FUP,FX,WTX,LBAND,JTYPE)
      LBAND=LBAND+1
      L=L+2
      IF(LBAND.GT.NBANDS) GO TO 160
      GRID(J)=EDGE(L)
      GO TO 140
  160 NGRID=J-1
      IF(NEG.NE.NODD) GO TO 165
      IF(GRID(NGRID).GT.(0.5-DELF)) NGRID=NGRID-1
```

```
PROGRAM     REMEXC CDC 6600 FTN V3.0-P380 OPT=1  79/03/29. 13.28.54.PAGE     4
      165 CONTINUE
C
C         SET UP A NEW APPROXIMATION PROBLEM WHICH IS EQUIVALENT
C         TO THE ORIGINAL PROBLEM
C
          IF(NEG) 170,170,180
      170 IF(NODD.EQ.1) GO TO 200
          DO 175 J=1,NGRID
          CHANGE=DCOS(PI*GRID(J))
          DES(J)=DES(J)/CHANGE
      175 WT(J)=WT(J)*CHANGE
          GO TO 200
      180 IF(NODD.EQ.1) GO TO 190
          DO 185 J=1,NGRID
          CHANGE=DSIN(PI*GRID(J))
          DES(J)=DES(J)/CHANGE
      185 WT(J)=WT(J)*CHANGE
          GOTO 200
      190 DO 195 J=1,NGRID
          CHANGE=DSIN(PI2*GRID(J))
          DES(J)=DES(J)/CHANGE
      195 WT(J)=WT(J)*CHANGE
C
C         INITIAL GUESS FOR THE EXTREMAL FREQUENCIES--EQUALLY
C         SPACED ALONG THE GRID
C
      200 TEMP=FLOAT(NGRID-1)/FLOAT(NFCNS)
          DO 210 J=1,NFCNS
      210 IEXT(J)=(J-1)*TEMP+1
          IEXT(NFCNS+1)=NGRID
          NM1=NFCNS-1
          NZ=NFCNS+1
C
C         CALL THE REMEZ EXCHANGE ALGORITHM TO DO THE APPROXIMATION
C         PROBLEM
C
          CALL REMEZ(EDGE,NBANDS)
C
C         CALCULATE THE IMPULSE RESPONSE.
C
          IF(NEG) 300,300,320
      300 IF(NODD.EQ.0) GO TO 310
          DO 305 J=1,NM1
      305 H(J)=0.5*ALPHA(NZ-J)
          H(NFCNS)=ALPHA(1)
          GO TO 350
      310 H(1)=0.25*ALPHA(NFCNS)
          DO 315 J=2,NM1
      315 H(J)=0.25*(ALPHA(NZ-J)+ALPHA(NFCNS+2-J))
          H(NFCNS)=0.5*ALPHA(1)+0.25*ALPHA(2)
          GO TO 350
      320 IF(NODD.EQ.0) GO TO 330
          H(1)=0.25*ALPHA(NFCNS)
          H(2)=0.25*ALPHA(NM1)
          DO 325 J=3,NM1
```

```
  325 H(J)=0.25*(ALPHA(NZ-J)-ALPHA(NFCNS+3-J))
      H(NFCNS)=0.5*ALPHA(1)-0.25*ALPHA(3)
      H(NZ)=0.0
      GO TO 350
  330 H(1)=0.25*ALPHA(NFCNS)
      DO 335 J=2,NM1
  335 H(J)=0.25*(ALPHA(NZ-J)-ALPHA(NFCNS+2-J))
      H(NFCNS)=0.5*ALPHA(1)-0.25*ALPHA(2)
C
C     PROGRAM OUTPUT SECTION
C
      PRINT 340
  340 FORMAT(1H1)
  350 PRINT 360
  360 FORMAT(25X,*FINITE IMPULSE RESPONSE (FIR)*/
     125X,*LINEAR PHASE DIGITAL FILTER DESIGN*/
     225X,*REMEZ EXCHANGE ALGORITHM*/)
      IF(JTYPE.EQ.1)PRINT 365
  365 FORMAT(25X,*BANDPASS FILTER*/)
      IF(JTYPE.EQ.2) PRINT 370
  370 FORMAT(25X,*DIFFERENTIATOR*/)
      IF(JTYPE.EQ.3) PRINT 375
  375 FORMAT(25X,*HILBERT TRANSFORMER*/)
      PRINT 378,NFILT
  378 FORMAT(25X,*FILTER LENGTH = *,I3/)
      IF(NROX.EQ.1) PRINT 379
  379 FORMAT(25X,*FILTER LENGTH DETERMINED BY APPROXIMATION*/)
      PRINT 380
  380 FORMAT(23X,*..... IMPULSE RESPONSE .....*)
      DO 381 J=1,NFCNS
      K=NFILT+1-J
      IF(NEG.EQ.0) PRINT 382,J,H(J),K
      IF(NEG.EQ.1) PRINT 383,J,H(J),K
  381 CONTINUE
  382 FORMAT(20X,*H(*,I3,*) = *,E15.8,* = H(*,I4,*)*)
  383 FORMAT(20X,*H(*,I3,*) = *,E15.8,* = -H(*,I4,*)*)
      IF(NEG.EQ.1.AND.NODD.EQ.1) PRINT 384,NZ
  384 FORMAT(20X,*H(*,I3,*) = 0.0*)
      DO 450 K=1,NBANDS,4
      KUP=K+3
      IF(KUP.GT.NBANDS) KUP=NBANDS
      PRINT 385,(J,J=K, KUP)
  385 FORMAT(/24X,4(*BAND*,I3,8X))
      PRINT 390,(EDGE(2*J-1),J=K,KUP)
  390 FORMAT(2X,*LOWER BAND EDGE*,5F15.9)
      PRINT 395,(EDGE(2*J),J=K,KUP)
  395 FORMAT(2X,*UPPER BAND EDGE*,5F15.9)
      IF(JTYPE.NE.2) PRINT 400,(FX(J),J=K,KUP)
  400 FORMAT(2X,*DESIRED VALUE*,2X,5F15.9)
      IF(JTYPE.EQ.2) PRINT 405,(FX(J),J=K,KUP)
  405 FORMAT(2X,*DESIRED SLOPE*,2X,5F15.9)
      PRINT 410,(WTX(J),J=K,KUP)
  410 FORMAT(2X,*WEIGHTING*,6X,5F15.9)
      DO 420 J=K,KUP
  420 DEVIAT(J)=DEV/WTX(J)
```

```
      PRINT 425,(DEVIAT(J),J=K,KUP)
  425 FORMAT(2X,*DEVIATION*,6X,5F15.9)
      IF(JTYPE.NE.1) GO TO 450
      DO 430 J=K,KUP
      IF (FX(J).EQ.1.0) DEVIAT(J)=(1.0+DEVIAT(J))/(1.0-DEVIAT(J))
  430 DEVIAT(J)=20.0*ALOG10(DEVIAT(J))
      PRINT 435,(DEVIAT(J),J=K,KUP)
  435 FORMAT(2X,*DEVIATION IN DB*,5F15.9)
  450 CONTINUE
      PRINT 455,(GRID(IEXT(J)),J=1,NZ)
  455 FORMAT(/2X,*EXTREMAL FREQUENCIES*/(2X,5F12.7))
      PRINT 460
  460 FORMAT(1H1)
C     CALCULATE FREQUENCY RESPONSE
      PRINT 710
  710 FORMAT(18X*........ FREQUENCY RESPONSE ........*)
      DO 610 IKA = 1,50
      OMEGA(IKA) = 0.010000000*(IKA - 1)
      SUMAR = 0.0
      SUMAC = 0.0
      DO 620 NIK = 1, NFCNS
      SUMAR=SUMAR+H(NIK)*(DCOS(PI2*OMEGA(IKA)*(NIK-1)) +
     1DCOS(PI2*OMEGA(IKA)*(NFILT-NIK)))
      SUMAC = SUMAC + H(NIK)*(DSIN(PI2*OMEGA(IKA)*(NIK-1)) +
     1DSIN(PI2*OMEGA(IKA)*(NFILT-NIK)))
  620 CONTINUE
      IF (NODD.EQ.1) SUMAR=SUMAR-H(NFCNS)*DCOS(PI2*OMEGA(IKA)*(NFCNS
     1-1))
      IF (NODD.EQ.1) SUMAC=SUMAC-H(NFCNS)*DSIN(PI2*OMEGA(IKA)*(NFCNS
     1-1))
      RESPA(IKA) = DSQRT(SUMAR**2 + SUMAC**2)
      RESPA(IKA) = 20.0*DLOG10(RESPA(IKA))
      PRINT 630, OMEGA(IKA),RESPA(IKA)
  630 FORMAT(15X,D12.4,15X,D12.4)
  610 CONTINUE
      IF(NFILT.NE.0) GO TO 100
      CONTINUE
      END
```

```
FUNCTION    EFF CDC 6600 FTN V3.0-P380 OPT=1  79/03/29. 13.28.54. PAGE     1

                  FUNCTION EFF(TEMP,FX,WTX,LBAND,JTYPE)
            C
            C     FUNCTION TO CALCULATE THE DESIRED MAGNITUDE RESPONSE
            C     AS A FUNCTION OF FREQUENCY
            C
                  DIMENSION FX(5),WTX(5)
                  IF(JTYPE.EQ.2) GO TO 1
                  EFF=FX(LBAND)
                  RETURN
                1 EFF=FX(LBAND)*TEMP
                  RETURN
                  END
```

```
FUNCTION    WATE   CDC 6600 FTN V3.0-P380 OPT=1  79/03/29. 13.28.54. PAGE    1
                      FUNCTION WATE(TEMP,FX,WTX,LBAND,JTYPE)
                C
                C     FUNCTION TO CALCULATE THE WEIGHT FUNCTION AS A FUNCTION
                C     OF FREQUENCY
                      DIMENSION FX(5),WTX(5)
                      IF(JTYPE.EQ.2) GO TO 1
                      WATE=WTX(LBAND)
                      RETURN
                    1 IF(FX(LBAND).LT.0.001) GO TO 2
                      WATE=WTX(LBAND)/TEMP
                      RETURN
                    2 WATE=WTX(LBAND)
                      RETURN
                      END
```

```
SUBROUTINE  ERROR  CDC 6600 FTN V3.0-P380 OPT=1  79/03/29. 13.28.54. PAGE    1

          SUBROUTINE ERROR
          PRINT 1
        1 FORMAT(* ........... ERROR IN INPUT DATA ...........*)
          STOP
          END
```

```
SUBROUTINE  REMEZ  CDC 6600 FTN V3.0-P380 OPT=1  79/03/29. 13.28.54. PAGE      1

              SUBROUTINE REMEZ(EDGE,NBANDS)
        C
        C     THIS SUBROUTINE IMPLEMENTS THE REMEZ EXCHANGE ALGORITHM
        C     FOR THE WEIGHTED CHEBYCHEV APPROXIMATION OF A CONTINUOUS
        C     FUNCTION WITH A SUM OF COSINES.  INPUTS TO THE SUBROUTINE
        C     ARE A DENSE GRID WHICH REPLACES THE FREQUENCY AXIS, THE
        C     DESIRED FUNCTION ON THIS GRID, THE WEIGHT FUNCTION ON THE
        C     GRID, THE NUMBER OF COSINES, AND AN INITIAL GUESS OF THE
        C     EXTREMAL FREQUENCIES.  THE PROGRAM MINIMIZES THE CHEBYCHEV
        C     ERROR BY DETERMINING THE BEST LOCATION OF THE EXTREMAL
        C     FREQUENCIES (POINTS OF MAXIMUM ERROR) AND THEN CALCULATES
        C     THE COEFFICIENTS OF THE BEST APPROXIMATION.
        C
              COMMON DES,WT,ALPHA,IEXT,NFCNS,NGRID,PI2,AD,DEV,X,Y,GRID
              DIMENSION EDGE(20)
              DIMENSION IEXT(66),AD(66),ALPHA(66),X(66),Y(66)
              DIMENSION DES(1045),GRID(1045),WT(1045)
              DIMENSION A(66),P(65),Q(65)
              DOUBLE PRECISION PI2,DNUM,DDEN,DTEMP,A,P,Q
              DOUBLE PRECISION AD,DEV,X,Y
        C
        C     THE PROGRAM ALLOWS A MAXIMUM NUMBER OF ITERATIONS OF 25
        C
              ITRMAX=25
              DEVL=-1.0
              NZ=NFCNS+1
              NZZ=NFCNS+2
              NITER=0
          100 CONTINUE
              IEXT(NZZ)=NGRID+1
              NITER=NITER+1
              IF(NITER.GT.ITRMAX) GO TO 400
              DO 110 J=1,NZ
              DTEMP=GRID(IEXT(J))
              DTEMP=DCOS(DTEMP*PI2)
          110 X(J)=DTEMP
              JET=(NFCNS-1)/15+1
              DO 120 J=1,NZ
          120 AD(J)=D(J,NZ,JET)
              DNUM=0.0
              DDEN=0.0
              K=1
              DO 130 J=1,NZ
              L=IEXT(J)
              DTEMP=AD(J)*DES(L)
              DNUM=DNUM+DTEMP
              DTEMP=K*AD(J)/WT(L)
              DDEN=DDEN+DTEMP
          130 K=-K
              DEV=DNUM/DDEN
              NU=1
              IF(DEV.GT.0.0) NU=-1
              DEV=-NU*DEV
              K=NU
              DO 140 J=1,NZ
```

```
SUBROUTINE  REMEZ  CDC 6600 FTN V3.0-P380 OPT=1  79/03/29. 13.28.54.
                                                              PAGE      2
          L=IEXT(J)
          DTEMP=K*DEV/WT(L)
          Y(J)=DES(L)+DTEMP
      140 K=-K
          IF(DEV.GE.DEVL) GO TO 150
          CALL OUCH
          GO TO 400
      150 DEVL=DEV
          JCHNGE=0
          K1=IEXT(1)
          KNZ=IEXT(NZ)
          KLOW=0
          NUT=-NU
          J=1
C
C         SEARCH FOR THE EXTREMAL FREQUENCIES OF THE BEST
C         APPROXIMATION
C
      200 IF(J.EQ.NZZ) YNZ=COMP
          IF(J.GE.NZZ) GO TO 300
          KUP=IEXT(J+1)
          L=IEXT(J)+1
          NUT=-NUT
          IF(J.EQ.2) Y1=COMP
          COMP=DEV
          IF(L.GE.KUP) GO TO 220
          ERR=GEE(L,NZ)
          ERR=(ERR-DES(L))*WT(L)
          DTEMP=NUT*ERR-COMP
          IF(DTEMP.LE.0.0) GO TO 220
          COMP=NUT*ERR
      210 L=L+1
          IF(L.GE.KUP) GO TO 215
          ERR=GEE(L,NZ)
          ERR=(ERR-DES(L))*WT(L)
          DTEMP=NUT*ERR-COMP
          IF(DTEMP.LE.0.0) GO TO 215
          COMP=NUT*ERR
          GO TO 210
      215 IEXT(J)=L-1
          J=J+1
          KLOW=L-1
          JCHNGE=JCHNGE+1
          GO TO 200
      220 L=L-1
      225 L=L-1
          IF(L.LE.KLOW) GO TO 250
          ERR=GEE(L,NZ)
          ERR=(ERR-DES(L))*WT(L)
          DTEMP=NUT*ERR-COMP
          IF(DTEMP.GT.0.0) GO TO 230
          IF(JCHNGE.LE.0) GO TO 225
          GO TO 260
      230 COMP=NUT*ERR
      235 L=L-1
```

```
      IF(L.LE.KLOW) GO TO 240
      ERR=GEE(L,NZ)
      ERR=(ERR-DES(L))*WT(L)
      DTEMP=NUT*ERR-COMP
      IF(DTEMP.LE.0.0) GO TO 240
      COMP=NUT*ERR
      GO TO 235
  240 KLOW=IEXT(J)
      IEXT(J)=L+1
      J=J+1
      JCHNGE=JCHNGE+1
      GO TO 200
  250 L=IEXT(J)+1
      IF(JCHNGE.GT.0) GO TO 215
  255 L=L+1
      IF(L.GE.KUP) GO TO 260
      ERR=GEE(L,NZ)
      ERR=(ERR-DES(L))*WT(L)
      DTEMP=NUT*ERR-COMP
      IF(DTEMP.LE.0.0) GO TO 255
      COMP=NUT*ERR
      GO TO 210
  260 KLOW=IEXT(J)
      J=J+1
      GO TO 200
  300 IF(J.GT.NZZ) GO TO 320
      IF(K1.GT.IEXT(1)) K1=IEXT(1)
      IF(KNZ.LT.IEXT(NZ)) KNZ=IEXT(NZ)
      NUT1=NUT
      NUT=-NU
      L=0
      KUP=K1
      COMP=YNZ*(1.00001)
      LUCK=1
  310 L=L+1
      IF(L.GE.KUP) GO TO 315
      ERR=GEE(L,NZ)
      ERR=(ERR-DES(L))*WT(L)
      DTEMP=NUT*ERR-COMP
      IF(DTEMP.LE.0.0) GO TO 310
      COMP=NUT*ERR
      J=NZZ
      GO TO 210
  315 LUCK=6
      GO TO 325
  320 IF(LUCK.GT.9) GO TO 350
      IF(COMP.GT.Y1) Y1=COMP
      K1=IEXT(NZZ)
  325 L=NGRID+1
      KLOW=KNZ
      NUT=-NUT1
      COMP=Y1*(0.00001)
  330 L=L-1
      IF(L.LE.KLOW) GO TO 340
      ERR=GEE(L,NZ)
```

```
          ERR=(ERR-DES(L))*WT(L)
          DTEMP=NUT*ERR-COMP
          IF(DTEMP.LE.0.0) GO TO 330
          J=NZZ
          COMP=NUT*ERR
          LUCK=LUCK+10
          GO TO 235
      340 IF(LUCK.EQ.6) GO TO 370
          DO 345 J=1,NFCNS
      345 IEXT(NZZ-J)=IEXT(NZ-J)
          IEXT(1)=K1
          GO TO 100
      350 KN=IEXT(NZZ)
          DO 360 J=1,NFCNS
      360 IEXT(J)=IEXT(J+1)
          IEXT(NZ)=KN
          GO TO 100
      370 IF(JCHNGE.GT.0) GO TO 100
C         CALCULATION OF THE COEFFICIENTS OF THE BEST APPROXIMATION
C         USING THE INVERSE DISCRETE FOURIER TRANSFORM
      400 CONTINUE
          NM1=NFCNS-1
          FSH=1.0E-06
          GTEMP=GRID(1)
          X(NZZ)=-2.0
          CN=2*NFCNS-1
          DELF=1.0/CN
          L=1
          KKK=0
          IF(EDGE(1).EQ.0.0.AND.EDGE(2*NBANDS).EQ.0.5) KKK=1
          IF(NFCNS.LE.3) KKK=1
          IF(KKK.EQ.1) GO TO 405
          DTEMP=DCOS(PI2*GRID(1))
          DNUM=DCOS(PI2*GRID(NGRID))
          AA=2.0/(DTEMP-DNUM)
          BB=-(DTEMP+DNUM)/(DTEMP-DNUM)
      405 CONTINUE
          DO 430 J=1,NFCNS
          FT=(J-1)*DELF
          XT=DCOS(PI2*FT)
          IF(KKK.EQ.1) GO TO 410
          XT=(XT-BB)/AA
          FT=ACOS(XT)/PI2
      410 XE=X(L)
          IF(XT.GT.XE) GO TO 420
          IF((XE-XT).LT.FSH) GO TO 415
          L=L+1
          GO TO 410
      415 A(J)=Y(L)
          GO TO 425
      420 IF((XT-XE).LT.FSH) GO TO 415
          GRID(1)=FT
          A(J)=GEE(1,NZ)
      425 CONTINUE
          IF(L.GT.1) L=L-1
```

```
  430 CONTINUE
      GRID(1)=GTEMP
      CDEN=PI2/CN
      DO 510 J=1,NFCNS
      DTEMP=0.0
      DNUM=(J-1)*DDEN
      IF(NM1.LT.1) GO TO 505
      DO 500 K=1,NM1
  500 DTEMP=DTEMP+A(K+1)*DCOS(DNUM*K)
  505 DTEMP=2.0*DTEMP+A(1)
  510 ALPHA(J)=DTEMP
      DO 550 J=2,NFCNS
  550 ALPHA(J)=2*ALPHA(J)/CN
      ALPHA(1)=ALPHA(1)/CN
      IF(KKK.EQ.1) GO TO 545
      P(1)=2.0*ALPHA(NFCNS)*BB+ALPHA(NM1)
      P(2)=2.0*AA*ALPHA(NFCNS)
      Q(1)=ALPHA(NFCNS-2)-ALPHA(NFCNS)
      DO 540 J=2,NM1
      IF(J.LT.NM1) GO TO 515
      AA=0.5*AA
      BB=0.5*BB
  515 CONTINUE
      P(J+1)=0.0
      DO 520 K=1,J
      A(K)=P(K)
  520 P(K)=2.0*BB*A(K)
      P(2)=P(2)+A(1)*2.0*AA
      JM1=J-1
      DO 525 K=1,JM1
  525 P(K)=P(K)+Q(K)+AA*A(K+2)
      JP1=J+1
      DO 530 K=3,JP1
  530 P(K)=P(K)+AA*A(K-1)
      IF(J.EQ.NM1) GO TO 540
      DO 535 K=1,J
  535 Q(K)=-A(K)
      Q(1)=Q(1)+ALPHA(NFCNS-1-J)
  540 CONTINUE
      DO 543 J=1,NFCNS
  543 ALPHA(J)=P(J)
  545 CONTINUE
      IF(NFCNS.GT.3) RETURN
      ALPHA(NFCNS+1)=0.0
      ALPHA(NFCNS+2)=0.0
      RETURN
      END
```

```
FUNCTION    D   CDC 6600 FTN V3.0-P380 CPT=1  79/03/29. 13.28.54. PAGE        1

              DOUBLE PRECISION FUNCTION D(K,N,M)
C             FUNCTION TO CALCULATE THE LAGRANGE INTERPOLATION
C             COEFFICIENT FOR USE IN THE FUNCTION GEE
              COMMON DES,WT,ALPHA,IEXT,NFCNS,NGRID,PI2,AD,DEV,X,Y,GRID
              DIMENSION IEXT(66),AD(66),ALPHA(66),X(66),Y(66)
              DIMENSION DES(1045),GRID(1045),WT(1045)
              DOUBLE PRECISION AD,DEV,X,Y
              DOUBLE PRECISION Q
              DOUBLE PRECISION PI2
              D=1.0
              Q=X(K)
              DO 3 L=1,M
              DO 2 J=L,N,M
              IF(J-K)1,2,1
            1 D=2.0*D*(Q-X(J))
            2 CONTINUE
            3 CONTINUE
              D=1.0/D
              RETURN
              END
```

```
FUNCTION     GEE  CDC 6600 FTN V3.0-P380 OPT=1  79/03/29. 13.28.54. PAGE       1

                 DOUBLE PRECISION FUNCTION GEE(K,N)
C                 FUNCTION TO EVALUATE THE FREQUENCY RESPONSE USING THE
C                LAGRANGE INTERPOLATION FORMULA IN THE BARYCENTRIC FORM
                 COMMON DES,WT,ALPHA,IEXT,NFCNS,NGRID,PI2,AD,DEV,X,Y,GRID
                 DIMENSION IEXT(66),AD(66),ALPHA(66),X(66),Y(66)
                 DIMENSION DES(1045),GRID(1045),WT(1045)
                 DOUBLE PRECISION P,C,D,XF
                 DOUBLE PRECISION PI2
                 DOUBLE PRECISION AD,DEV,X,Y
                 P=0.0
                 XF=GRID(K)
                 XF=DCOS(PI2*XF)
                 D=0.0
                 DO 1 J=1,N
                 C=XF-X(J)
                 C=AD(J)/C
                 D=D+C
               1 P=P+C*Y(J)
                 GEE=P/D
                 RETURN
                 END
```

```
SUBROUTINE  OUCH CDC 6600 FTN V3.0-P380 OPT=1  79/03/29. 13.28.54. PAGE        1
            SUBROUTINE OUCH
            PRINT 1
          1 FORMAT(*............. FAILURE TO CONVERGE .............*/
           1*0PROBABLE CAUSE IS MACHINE ROUNDING ERROR*/
           2*0THE IMPULSE RESPONSE MAY BE CORRECT*/
           3*0CHECK WITH A FREQUENCY RESPONSE*)
            RETURN
            END
```

D I S T R I B U T I O N   L I S T

REPORT NO. NADC-79153-40

AIRTASK NO. WF21-200-000
Work Unit No. ZC902

	No. of Copies
NAVAIRSYSCOM, AIR-950D	5
(2 for retention)	
(1 for AIR-370)	
(1 for AIR-5330)	
(1 for AIR-53301)	
CNO	2
(1 for OP-951)	
(1 for OP-952)	
COMOPTEVFOR	1
DDC	12