CR-128985

20817-H011-R0-00

SPECTRAL ANALYSIS PROGRAM (SAP)

VOLUME II - PROGRAMMER'S MANUAL

NAS 9-11977 16 June 1972

Prepared for

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION
MANNED SPACECRAFT CENTER
HOUSTON, TEXAS

Prepared by

G. L. Houser
R. C. Thomas

Approved by: WBWarren
W. B. Warren
Project Manager

TRW
SYSTEMS GROUP

ABSTRACT

This document contains the Spectral Analysis Program (SAP) flow diagrams and listings. The Plot Generation Program (PLTGEN) flow diagrams and listings are also included.

CONTENTS

ILLUSTRATIONS

1. INTRODUCTION

The purpose of this manual is to present the listings and flow diagrams for the Spectral Analysis Program (SAP) and the Plot Generation Program (PLTGEN). SAP computes the power spectrum of an angle modulated signal and is useful in the analysis, design, and testing of angle modulation communication systems. PLTGEN accepts a SAP output tape and generates the appropriate commands required to drive the EAI Dataplotter. The PLTGEN output tape is mounted on the Dataplotter to obtain a power spectrum plot.

The detailed mathematical formulation of SAP is presented in Reference 1 (Final Project Report). A user's guide, providing the necessary information to 1) understand the general SAP computational approach and software structure, 2) set-up the input parameters in the appropriate tape or card format, and 3) execute the program, is presented in Reference 2 (The Spectral Analysis Program, Volume I - User's Guide). User information for PLTGEN is also included in Reference 2.

SAP and PLTGEN were written using the standard TRW self-documentation technique. With this technique, each subroutine contains:

- Programmer and date
- Purpose of subroutine
- Description of each input/output parameter on cards, tape, or through COMMON
- Remarks and restrictions
- Additional subroutines required
- Numerous functional descriptions of sections of programming logic.

2. SPECTRAL ANALYSIS PROGRAM

This section presents 1) a brief description, 2) flow diagrams, and 3) listing for each module or subroutine used in the Spectral Analysis Program. The presentations are in alphabetical order.

2.1 SAP MODULE/SUBROUTINE DESCRIPTIONS

The following list describes all of the modules/subroutines for the SAP program.

BITREV	Bit reversal routine
CLØCK	References computer clock
EFFT	Computes extended fast Fourier transform
FFT	Computes fast Fourier transform
FILTER	Performs filtering of the modulating or modulated signal
ISAR	Generates a signal tape compatible with SAP from a user supplied input tape
MAGTAP	Tape operations routine
MØD	Performs the exponentiation operation used to generate the modulated signal representation from the modulating signal
PLØT	Generates the SAP plot tape
SAP	Spectral Analysis Program driver program
TIMER	Computes timing differences
TRFN	Calculates the real and imaginary parts of the desired transfer function
TSGEN	Generates built-in modulating test signals
TTRANS	Large matrix transpose routine
TWRITE	Prints signal at specified stages of processing.

2.2 SAP FLOW DIAGRAMS

A complete flow diagram for each of the module/subroutines listed in Section 2.1 is given in the following diagrams.

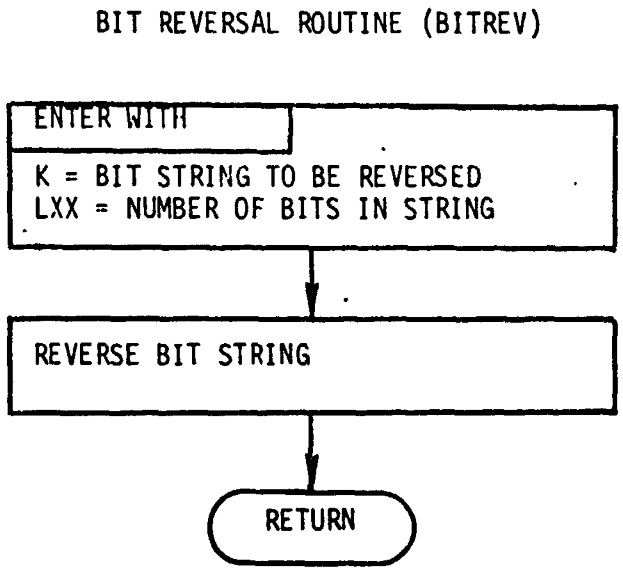
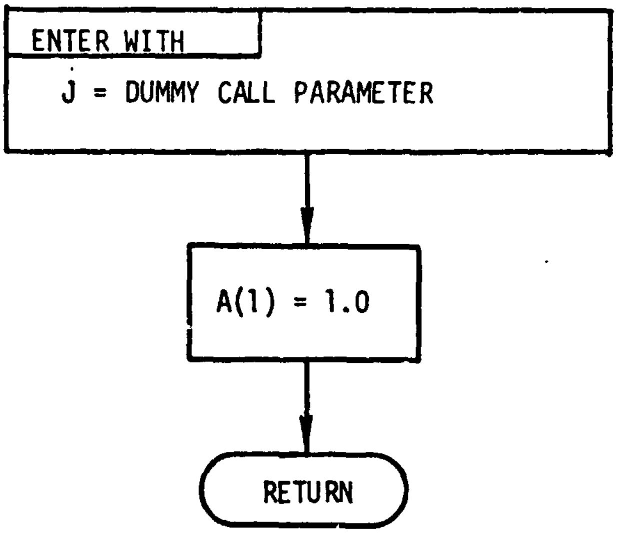

BIT REVERSAL ROUTINE (BITREV)
ENTER WITH
K = BIT STRING TO BE REVERSED
LXX = NUMBER OF BITS IN STRING
REVERSE BIT STRING
RETURN

Figure 2-1. Bit Reversal Routine (BITREV)

DUMMY TIMING ROUTINE* (CLOCK)



* The sole purpose of this dummy routine is to allocate enough core storage for a CS-1 routine named CLOCK that will be loaded later from paper tape.

Figure 2-2. Dummy Timing Routine* (Clock)

Extended FFT Subroutine

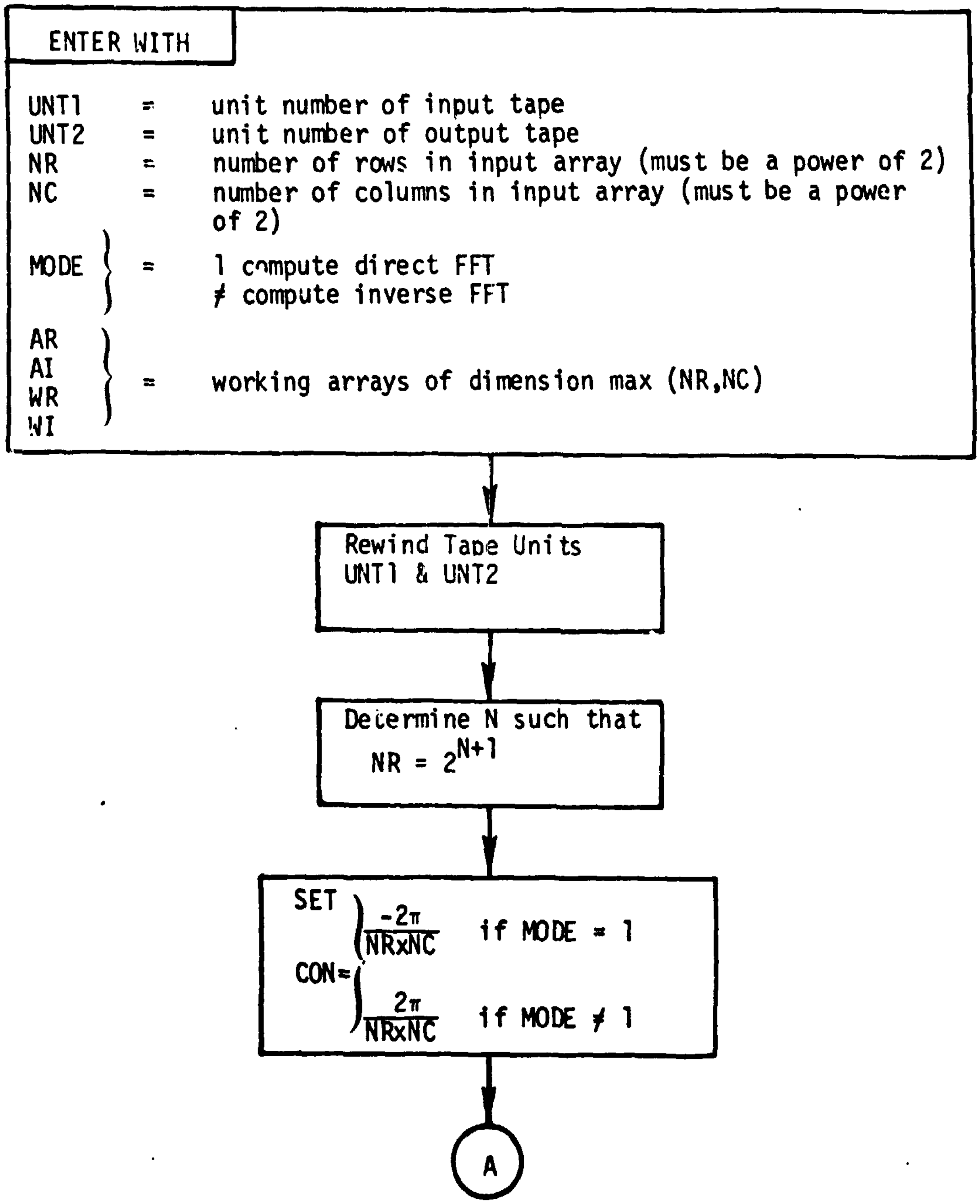
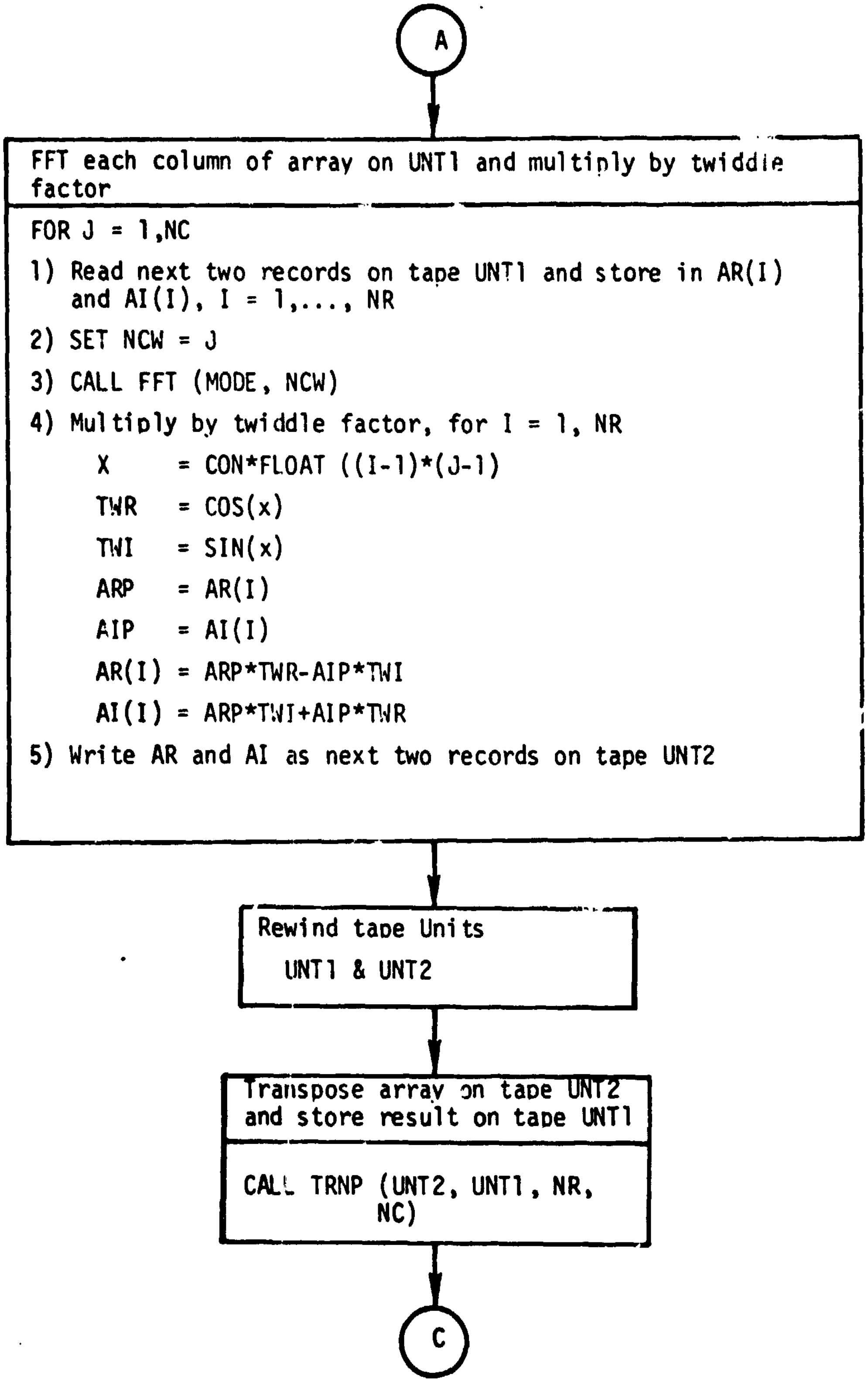

ENTER WITH
UNT1 = unit number of input tape
UNT2 = unit number of output tape
NR = number of rows in input array (must be a power of 2)
NC = number of columns in input array (must be a power of 2)
MODE = 1 compute direct FFT
≠ compute inverse FFT
AR
AI
WR
WI
= working arrays of dimension max (NR,NC)
Rewind Tape Units UNT1 & UNT2
Determine N such that
$NR = 2^{N+1}$
SET
CON = $\frac{-2\pi}{NR \times NC}$ if MODE = 1
$\frac{2\pi}{NR \times NC}$ if MODE ≠ 1
A

Figure 2-3. Extended FFT Subroutine



FFT each column of array on UNT1 and multiply by twiddle factor

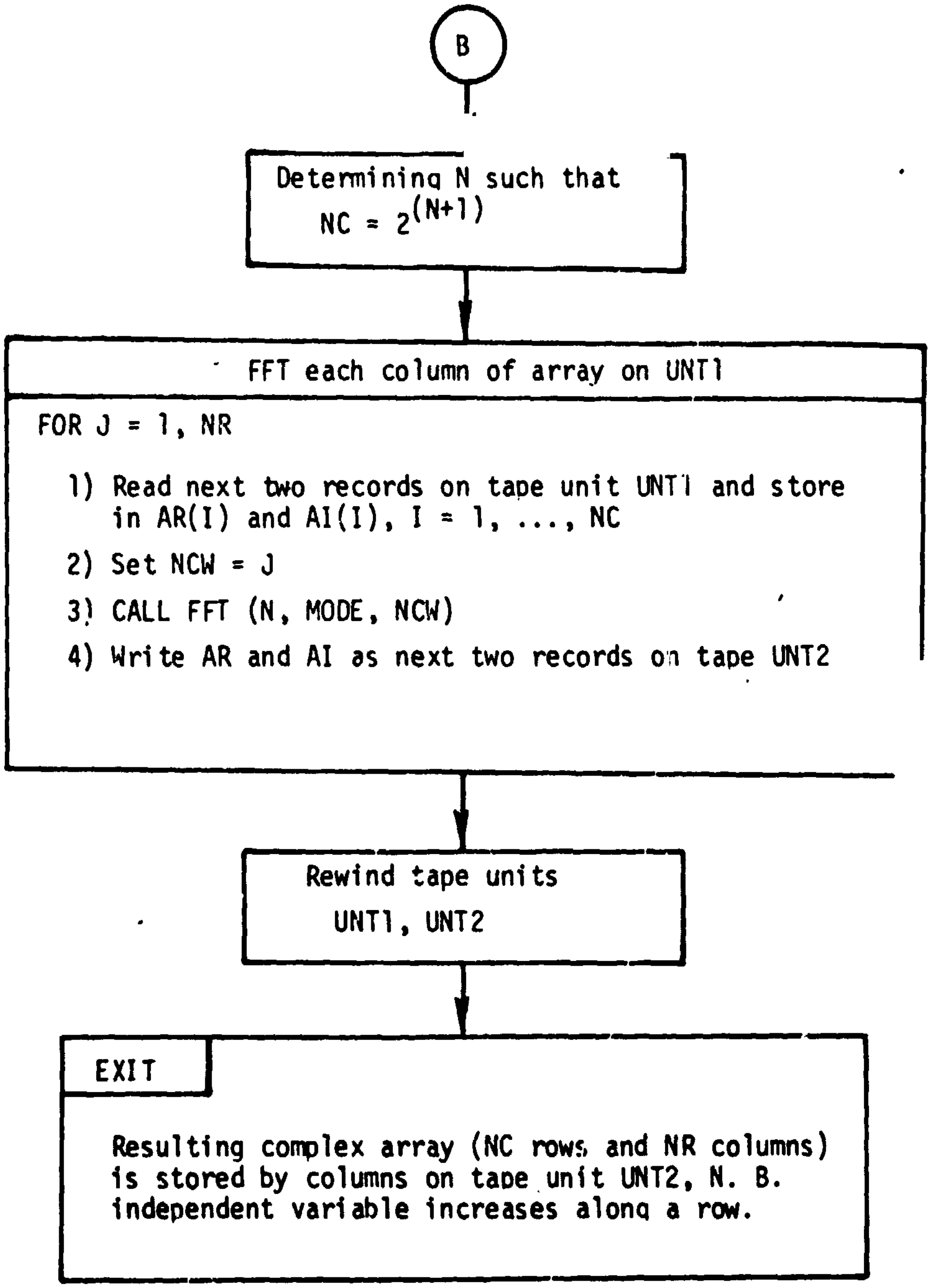
```
FOR J = 1,NC
1) Read next two records on tape UNT1 and store in AR(I)
   and AI(I), I = 1,..., NR
2) SET NCW = J
3) CALL FFT (MODE, NCW)
4) Multiply by twiddle factor, for I = 1, NR
      X     = CON*FLOAT ((I-1)*(J-1)
      TWR   = COS(x)
      TWI   = SIN(x)
      ARP   = AR(I)
      AIP   = AI(I)
      AR(I) = ARP*TWR-AIP*TWI
      AI(I) = ARP*TWI+AIP*TWR
5) Write AR and AI as next two records on tape UNT2
```

Rewind tape Units UNT1 & UNT2

Transpose array on tape UNT2 and store result on tape UNT1

```
CALL TRNP (UNT2, UNT1, NR,
        NC)
```

Figure 2-3. Extended FFT Subroutine (Cont'd)



Determining N such that $NC = 2^{(N+1)}$

FFT each column of array on UNT1

```
FOR J = 1, NR

  1) Read next two records on tape unit UNT1 and store
     in AR(I) and AI(I), I = 1, ..., NC
  2) Set NCW = J
  3) CALL FFT (N, MODE, NCW)
  4) Write AR and AI as next two records on tape UNT2
```

Rewind tape units
UNT1, UNT2

EXIT

Resulting complex array (NC rows and NR columns) is stored by columns on tape unit UNT2, N. B. independent variable increases along a row.

Figure 2-3. Extended FFT Subroutine (Cont'd)

Fast Fourier Transform Routine (FFT)

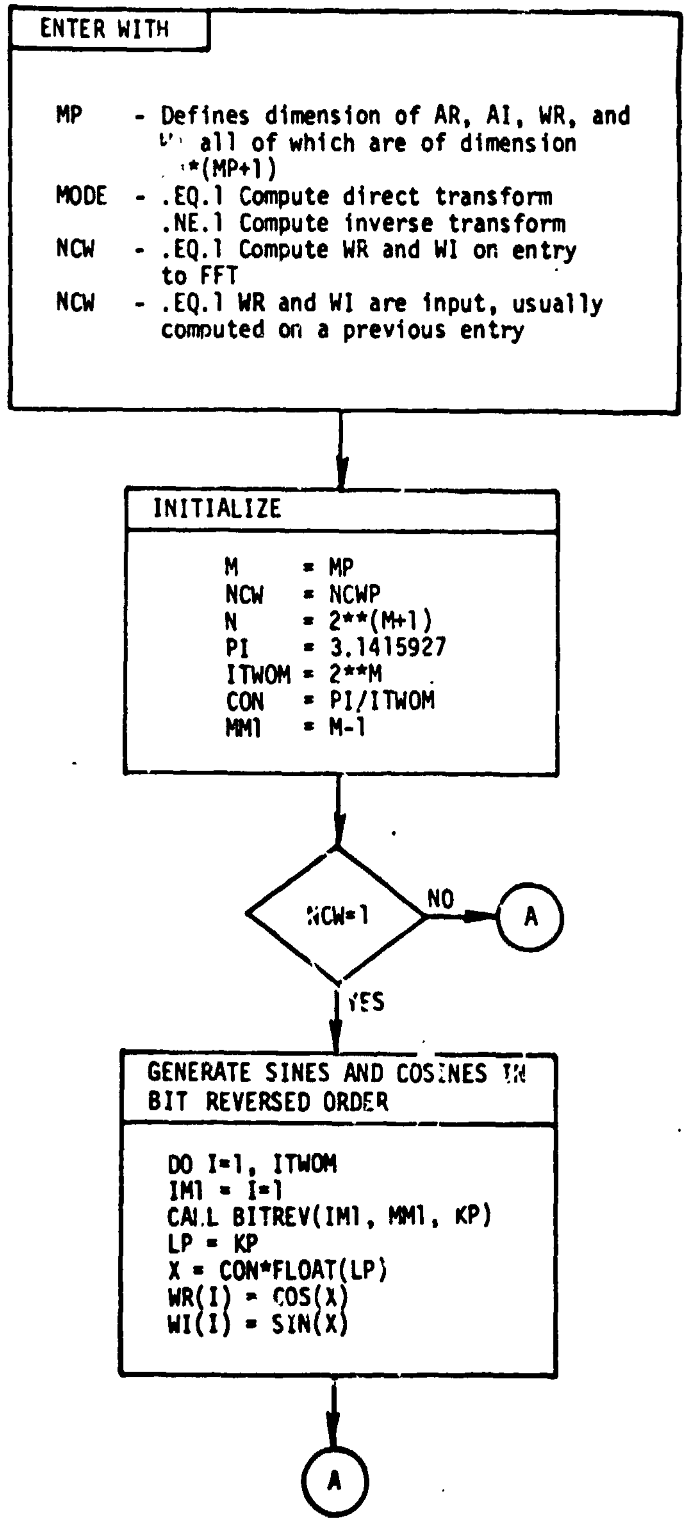
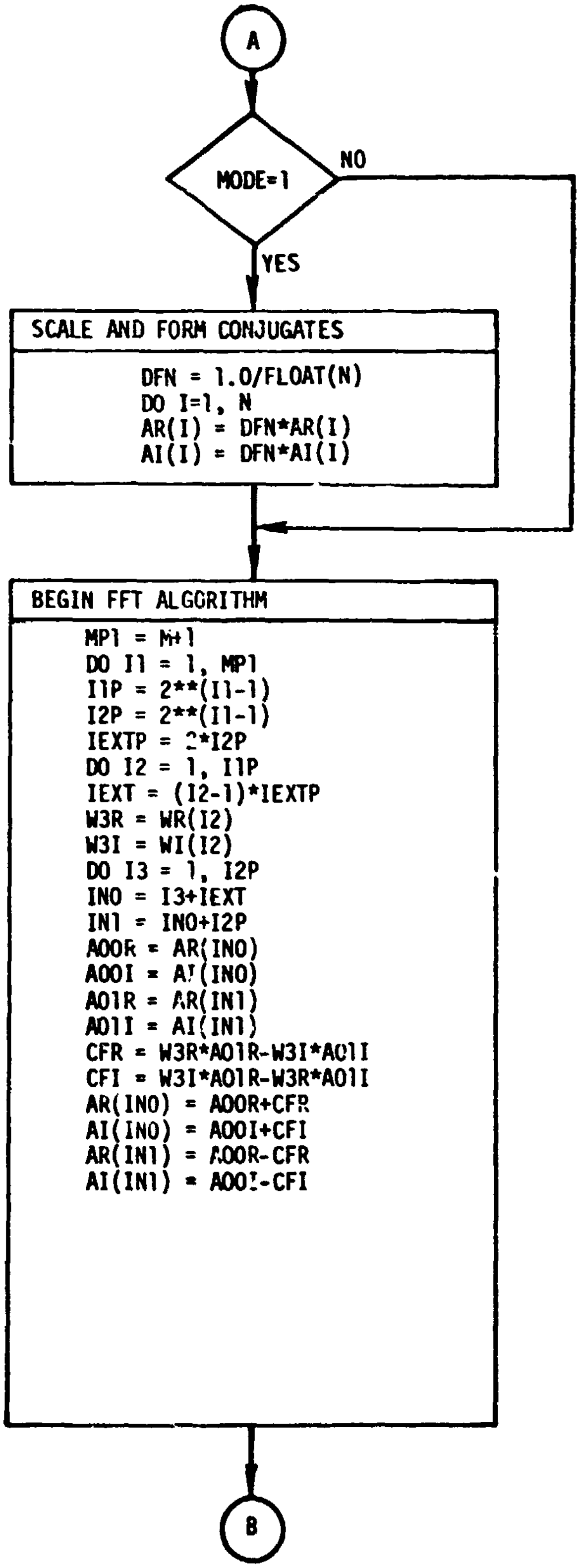

ENTER WITH
MP - Defines dimension of AR, AI, WR, and WI all of which are of dimension 2**(MP+1)
MODE - .EQ.1 Compute direct transform
.NE.1 Compute inverse transform
NCW - .EQ.1 Compute WR and WI on entry to FFT
NCW - .EQ.1 WR and WI are input, usually computed on a previous entry
INITIALIZE
M = MP
NCW = NCWP
N = 2**(M+1)
PI = 3.1415927
ITWOM = 2**M
CON = PI/ITWOM
MM1 = M-1
NCW=1
NO
A
YES
GENERATE SINES AND COSINES IN BIT REVERSED ORDER
DO I=1, ITWOM
IM1 = I=1
CALL BITREV(IM1, MM1, KP)
LP = KP
X = CON*FLOAT(LP)
WR(I) = COS(X)
WI(I) = SIN(X)
A

Figure 2-4. Fast Fourier Transform Routine (FFT)



```
DFN = 1.0/FLOAT(N)
DO I=1, N
AR(I) = DFN*AR(I)
AI(I) = DFN*AI(I)
```

```
MP1 = M+1
DO I1 = 1, MP1
I1P = 2**(I1-1)
I2P = 2**(I1-1)
IEXTP = 2*I2P
DO I2 = 1, I1P
IEXT = (I2-1)*IEXTP
W3R = WR(I2)
W3I = WI(I2)
DO I3 = 1, I2P
INO = I3+IEXT
IN1 = INO+I2P
AOOR = AR(INO)
AOOI = AI(INO)
AO1R = AR(IN1)
AO1I = AI(IN1)
CFR = W3R*AO1R-W3I*AO1I
CFI = W3I*AO1R-W3R*AO1I
AR(INO) = AOOR+CFR
AI(INO) = AOOI+CFI
AR(IN1) = AOOR-CFR
AI(IN1) = AOOI-CFI
```

Figure 2-4. Fast Fourier Transform Routine (FFT) (Cont'd)

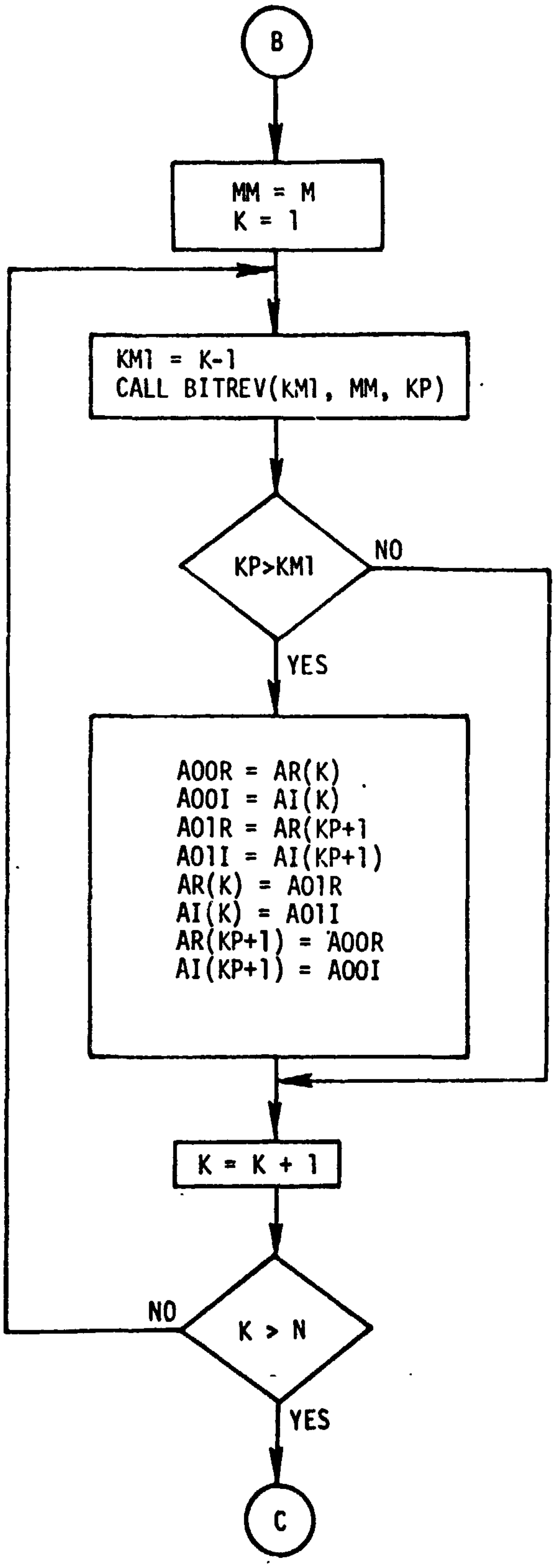


Figure 2-4. Fast Fourier Transform Routine (FFT) (Cont'd)

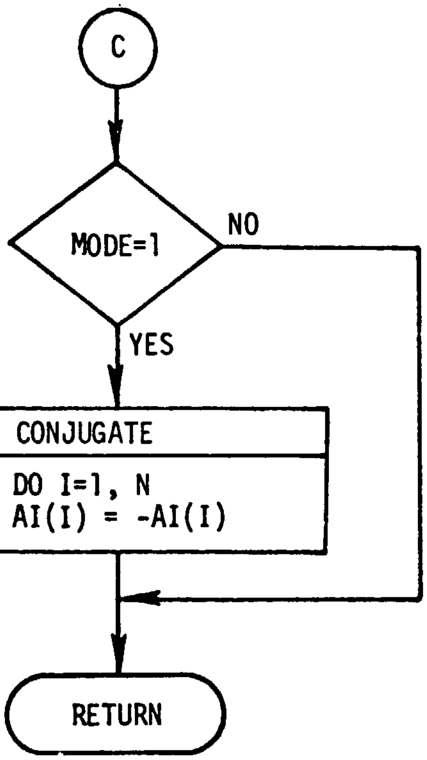
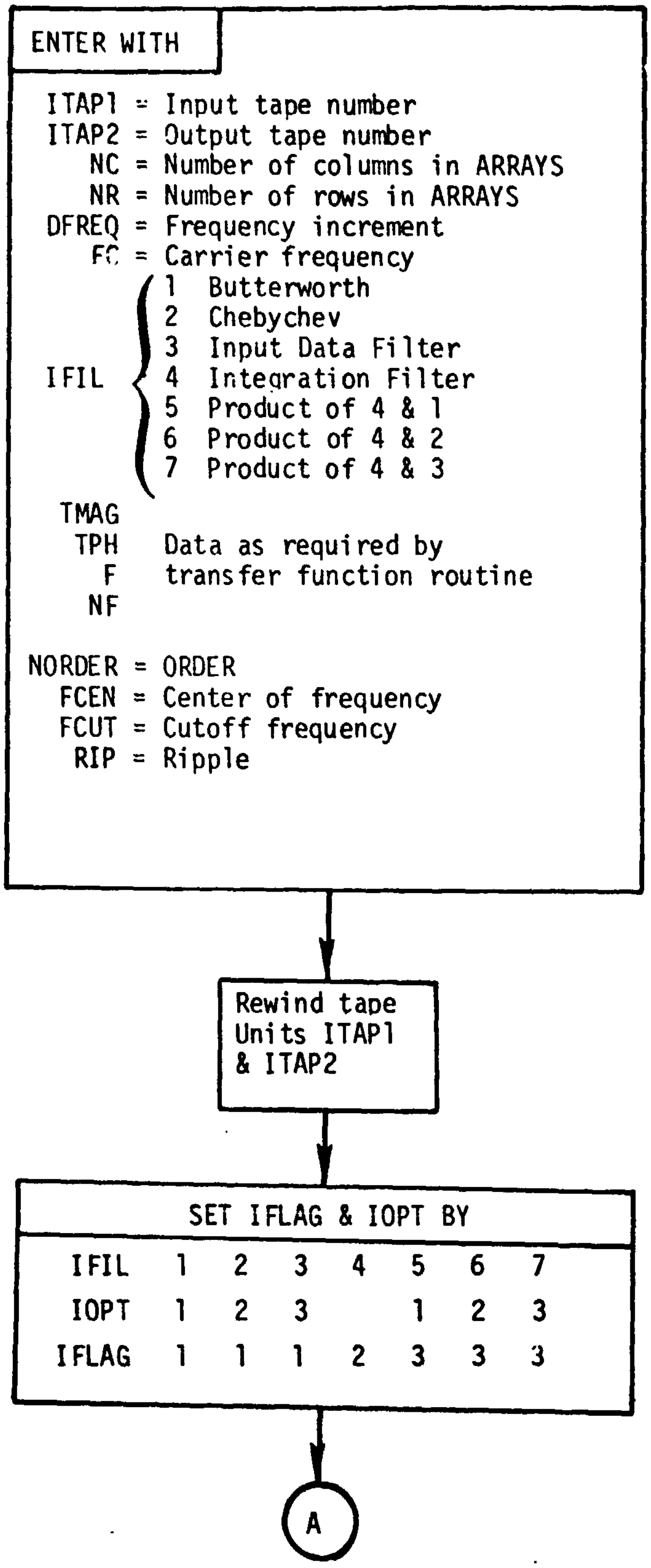


Figure 2-4. Fast Fourier Transform Routine (FFT) (Cont'd)

FILTER MODULE



ENTER WITH

ITAP1 = Input tape number
ITAP2 = Output tape number
NC = Number of columns in ARRAYS
NR = Number of rows in ARRAYS
DFREQ = Frequency increment
FC = Carrier frequency

IFIL
1 Butterworth
2 Chebychev
3 Input Data Filter
4 Integration Filter
5 Product of 4 & 1
6 Product of 4 & 2
7 Product of 4 & 3

TMAG
TPH
F
NF
Data as required by transfer function routine

NORDER = ORDER
FCEN = Center of frequency
FCUT = Cutoff frequency
RIP = Ripple

Rewind tape Units ITAP1 & ITAP2

SET IFLAG & IOPT BY

IFIL	1	2	3	4	5	6	7
IOPT	1	2	3		1	2	3
IFLAG	1	1	1	2	3	3	3

Figure 2-5. Filter Module

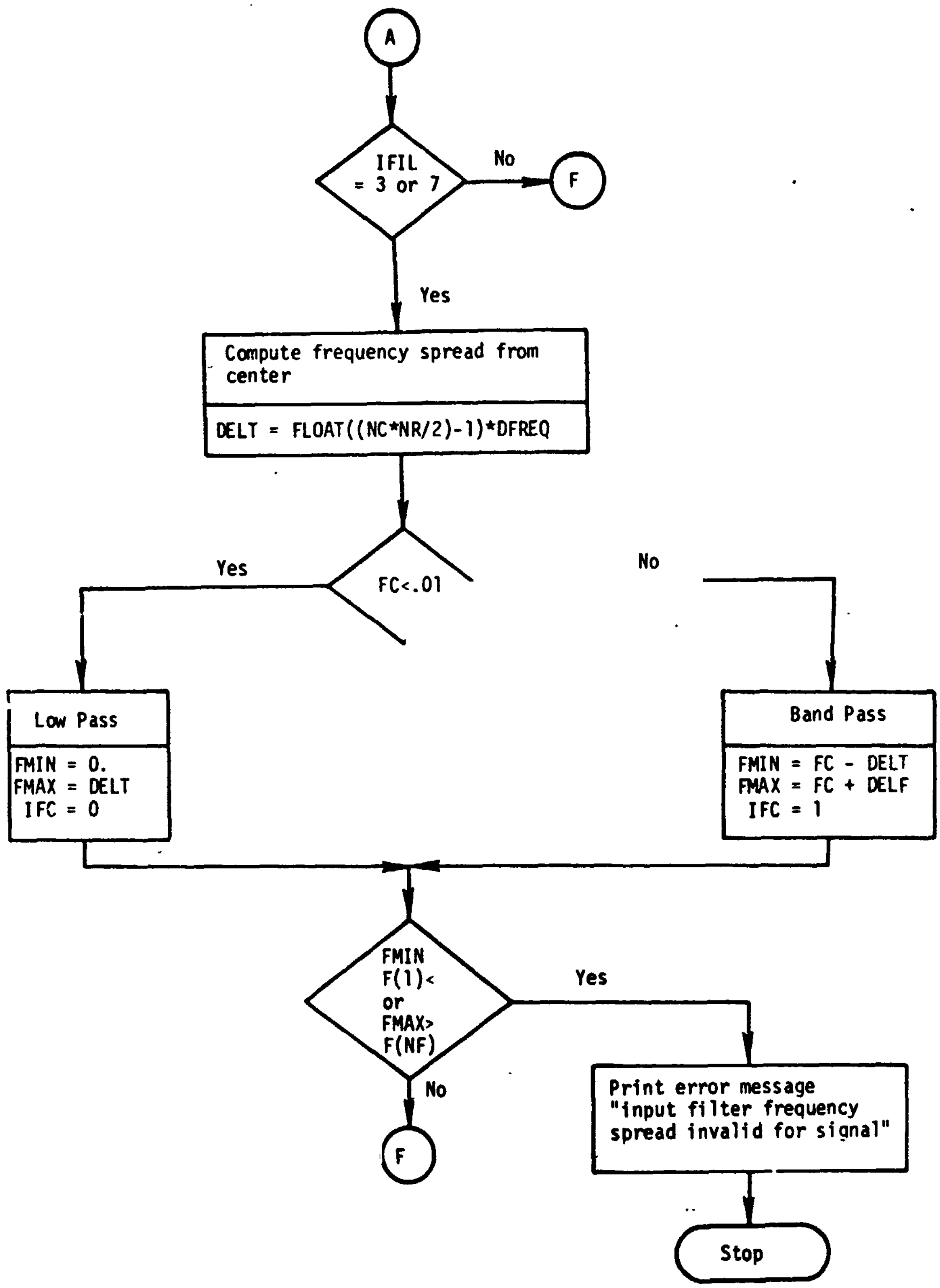
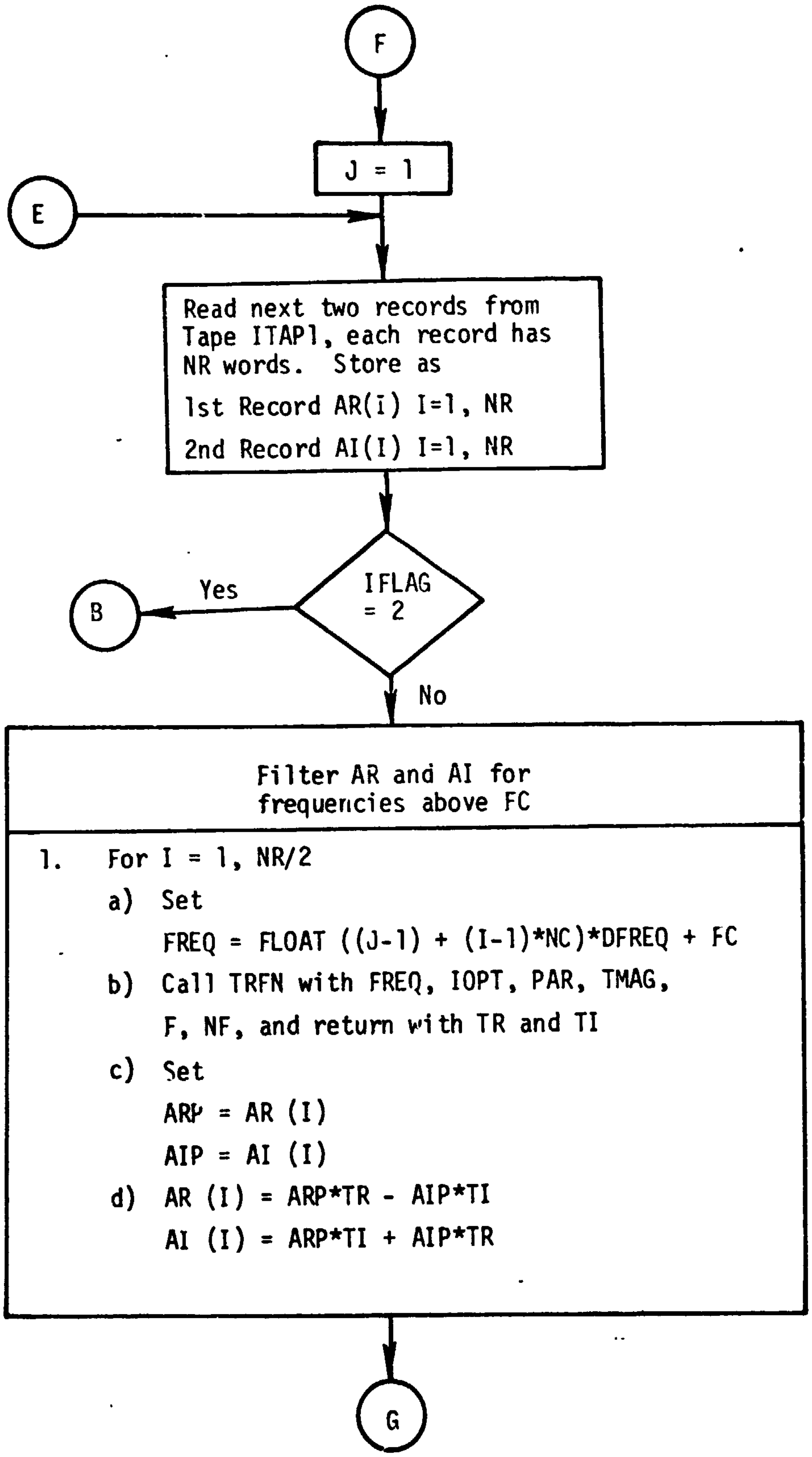

A
IFIL = 3 or 7
No
F
Yes
Compute frequency spread from center
DELT = FLOAT((NC*NR/2)-1)*DFREQ
FC<.01
Yes
No
Low Pass
FMIN = 0.
FMAX = DELT
IFC = 0
Band Pass
FMIN = FC - DELT
FMAX = FC + DELF
IFC = 1
FMIN F(1)< or FMAX> F(NF)
Yes
No
F
Print error message "input filter frequency spread invalid for signal"
Stop

Figure 2-5. Filter Module (Cont'd)



Read next two records from Tape ITAP1, each record has NR words. Store as

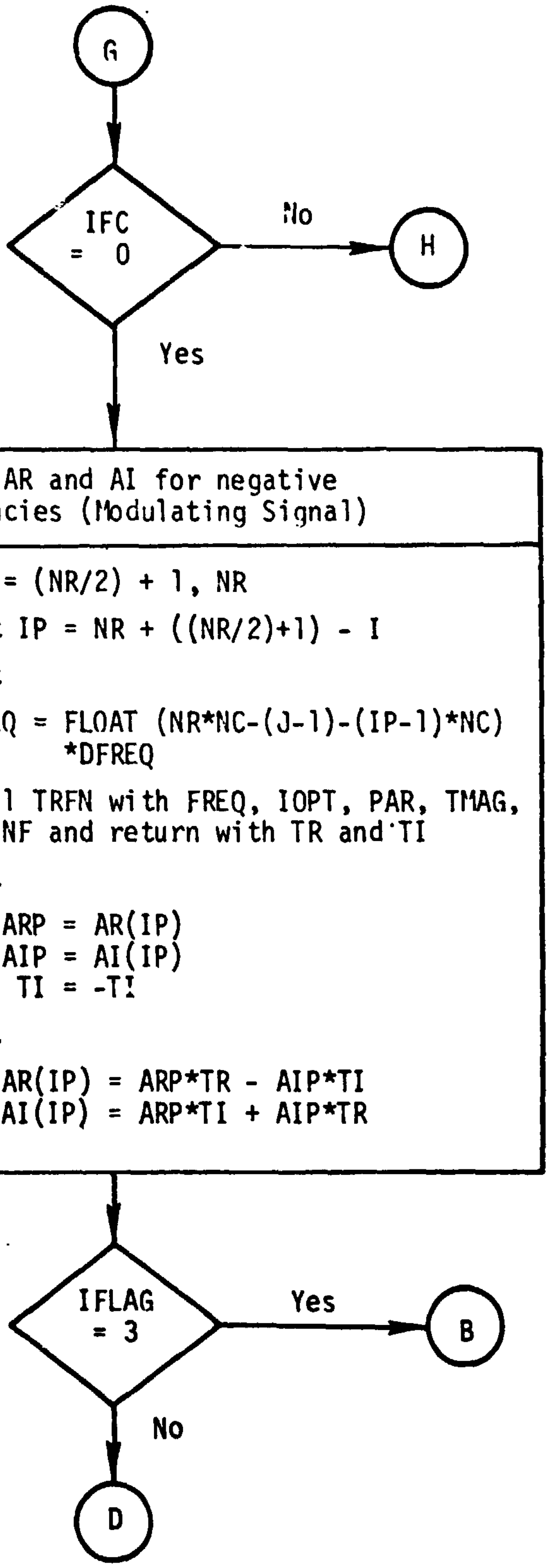
1st Record AR(I) I=1, NR

2nd Record AI(I) I=1, NR

Filter AR and AI for frequencies above FC

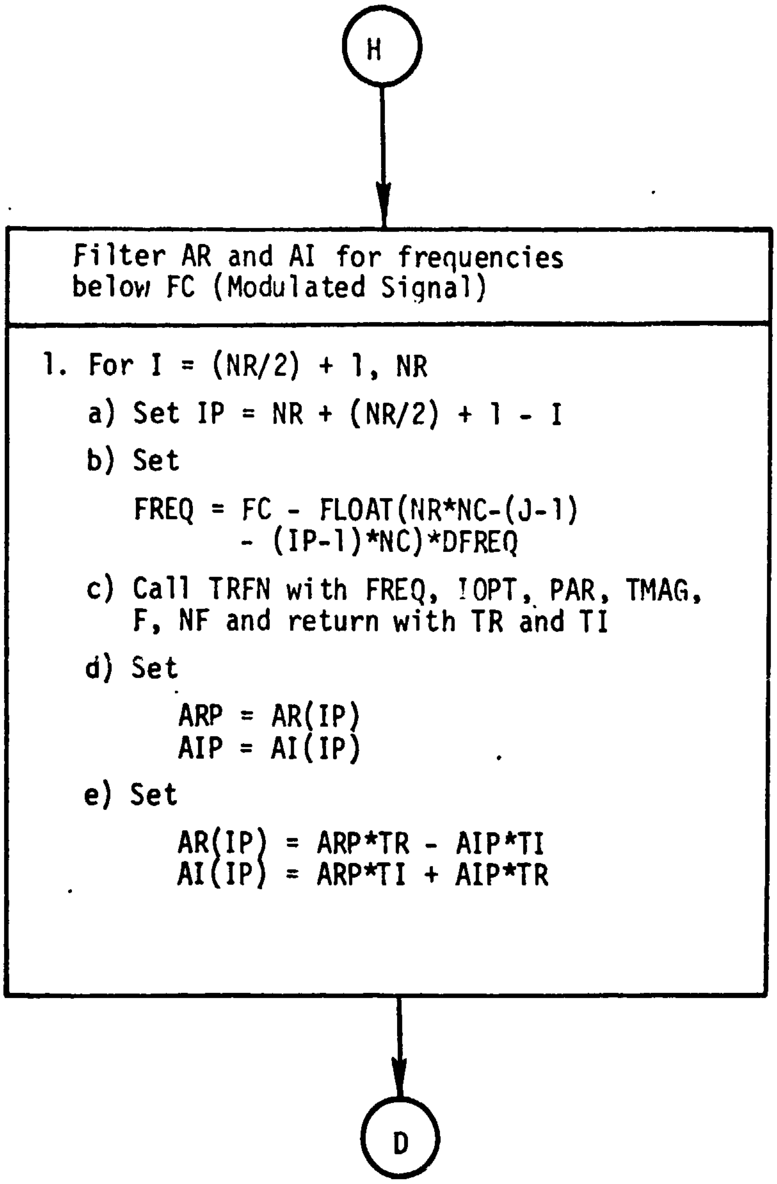
```
1.    For I = 1, NR/2
      a)  Set
          FREQ = FLOAT ((J-1) + (I-1)*NC)*DFREQ + FC
      b)  Call TRFN with FREQ, IOPT, PAR, TMAG,
          F, NF, and return with TR and TI
      c)  Set
          ARP = AR (I)
          AIP = AI (I)
      d)  AR (I) = ARP*TR - AIP*TI
          AI (I) = ARP*TI + AIP*TR
```

Figure 2-5. Filter Module (Cont'd)


G
IFC = 0
No
H
Yes
Filter AR and AI for negative frequencies (Modulating Signal)
IFLAG = 3
Yes
B
No
D

```
1.  For I = (NR/2) + 1, NR
    a) Set IP = NR + ((NR/2)+1) - I
    b) Set
       FREQ = FLOAT (NR*NC-(J-1)-(IP-1)*NC)
              *DFREQ
    c) Call TRFN with FREQ, IOPT, PAR, TMAG,
       F, NF and return with TR and TI
    d) Set
          ARP = AR(IP)
          AIP = AI(IP)
           TI = -TI
    e) Set
          AR(IP) = ARP*TR - AIP*TI
          AI(IP) = ARP*TI + AIP*TR
```

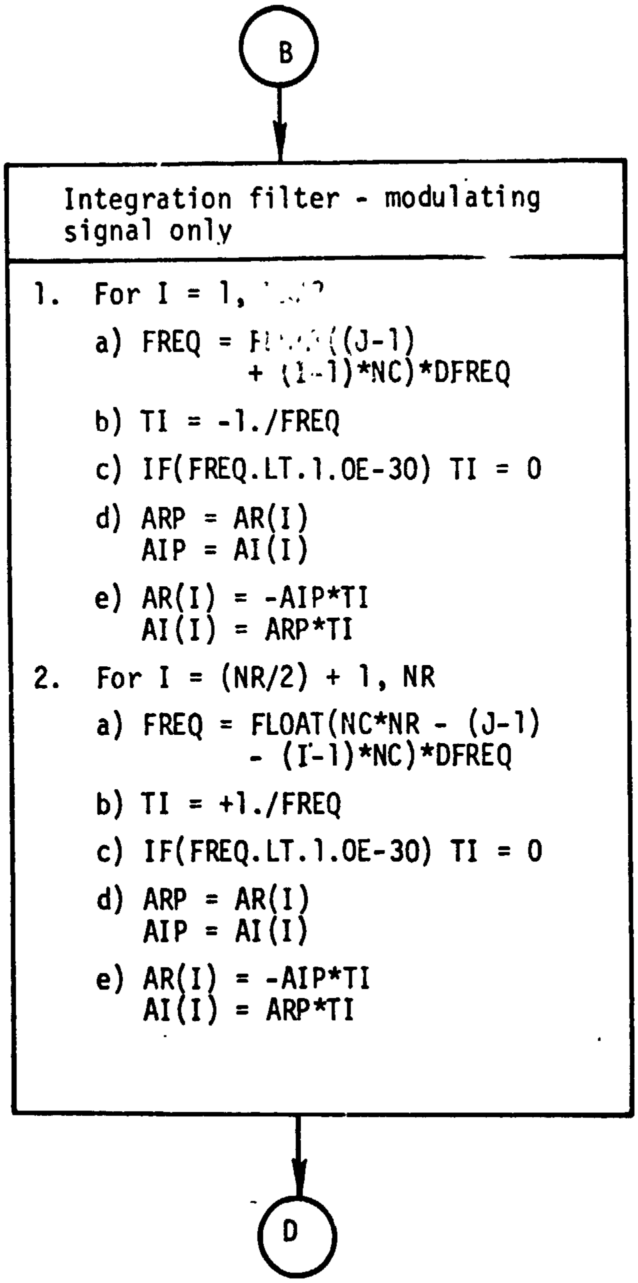
Figure 2-5. Filter Module (Cont'd)



Filter AR and AI for frequencies below FC (Modulated Signal)

```
1. For I = (NR/2) + 1, NR
   a) Set IP = NR + (NR/2) + 1 - I
   b) Set
      FREQ = FC - FLOAT(NR*NC-(J-1)
             - (IP-1)*NC)*DFREQ
   c) Call TRFN with FREQ, IOPT, PAR, TMAG,
      F, NF and return with TR and TI
   d) Set
         ARP = AR(IP)
         AIP = AI(IP)
   e) Set
         AR(IP) = ARP*TR - AIP*TI
         AI(IP) = ARP*TI + AIP*TR
```

Figure 2-5. Filter Module (Cont'd)



Integration filter - modulating signal only

```
1.  For I = 1, NR/2
    a) FREQ = FLOAT((J-1)
                + (I-1)*NC)*DFREQ
    b) TI = -1./FREQ
    c) IF(FREQ.LT.1.0E-30) TI = 0
    d) ARP = AR(I)
       AIP = AI(I)
    e) AR(I) = -AIP*TI
       AI(I) = ARP*TI
2.  For I = (NR/2) + 1, NR
    a) FREQ = FLOAT(NC*NR - (J-1)
                - (I-1)*NC)*DFREQ
    b) TI = +1./FREQ
    c) IF(FREQ.LT.1.0E-30) TI = 0
    d) ARP = AR(I)
       AIP = AI(I)
    e) AR(I) = -AIP*TI
       AI(I) = ARP*TI
```

Figure 2-5. Filter Module (Cont'd)

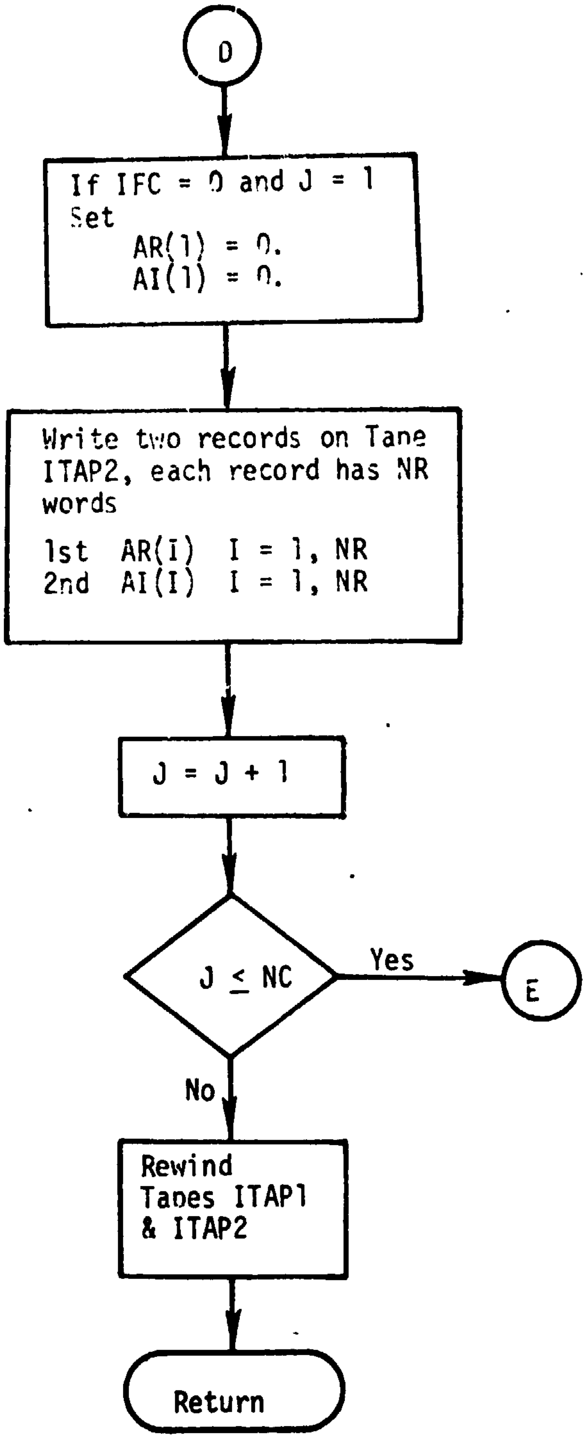


Figure 2-5. Filter Module (Cont'd)

INPUT SIGNAL MODULE

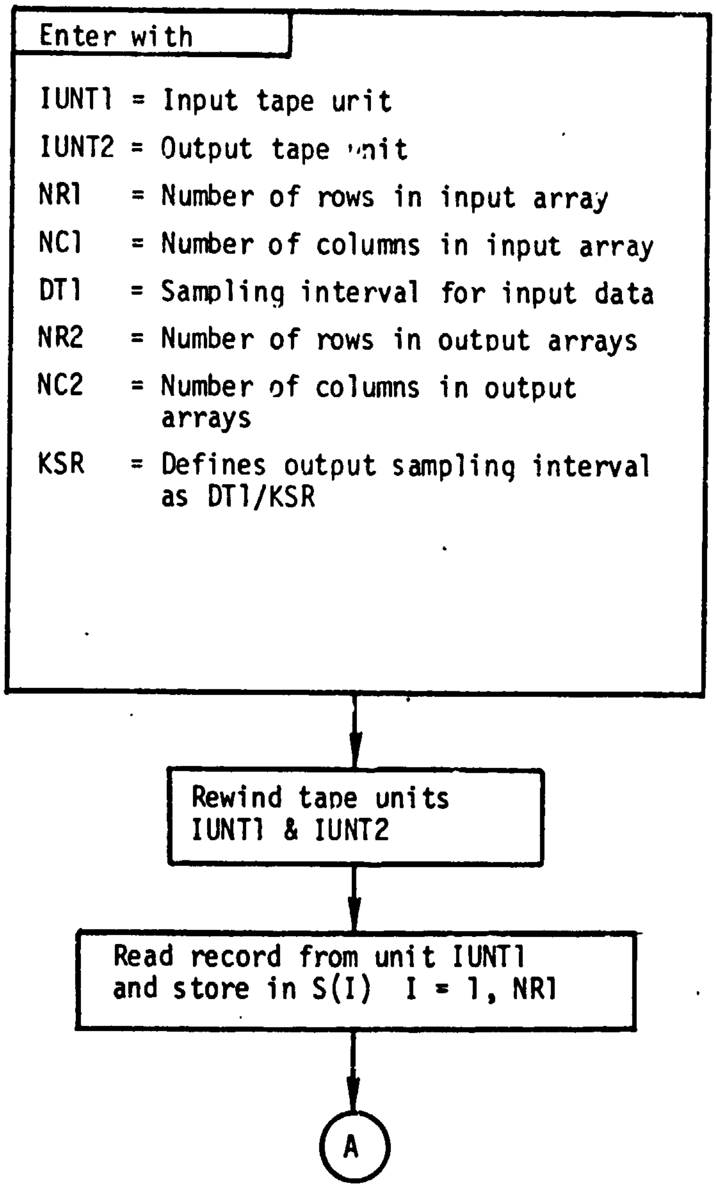

Enter with
IUNT1 = Input tape urit
IUNT2 = Output tape unit
NR1 = Number of rows in input array
NC1 = Number of columns in input array
DT1 = Sampling interval for input data
NR2 = Number of rows in outout arrays
NC2 = Number of columns in output arrays
KSR = Defines output sampling interval as DT1/KSR
Rewind tane units IUNT1 & IUNT2
Read record from unit IUNT1 and store in S(I) I = 1, NR1
A

Figure 2-6. Input Signal Module

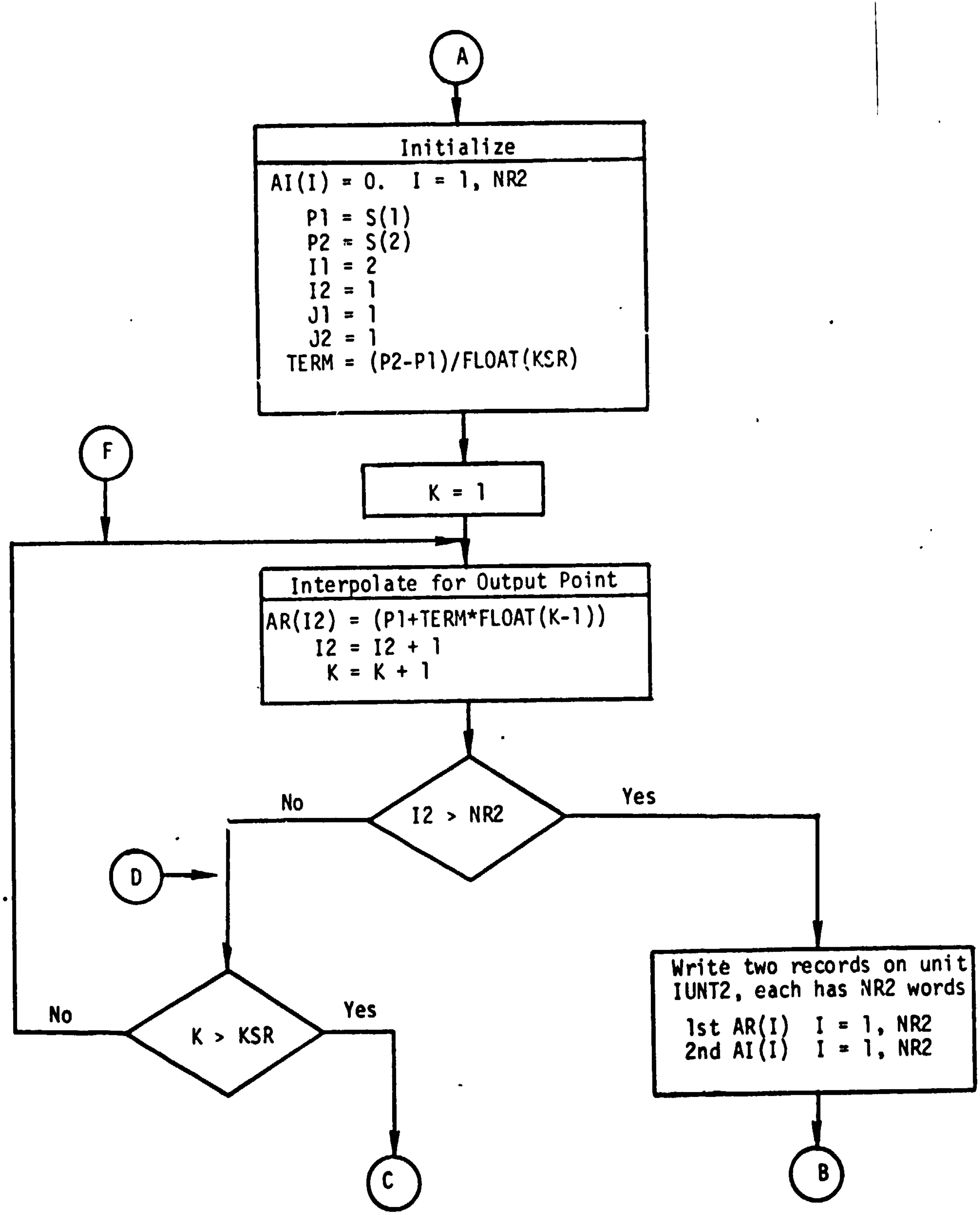


Figure 2-6. Input Signal Module (Cont'd)

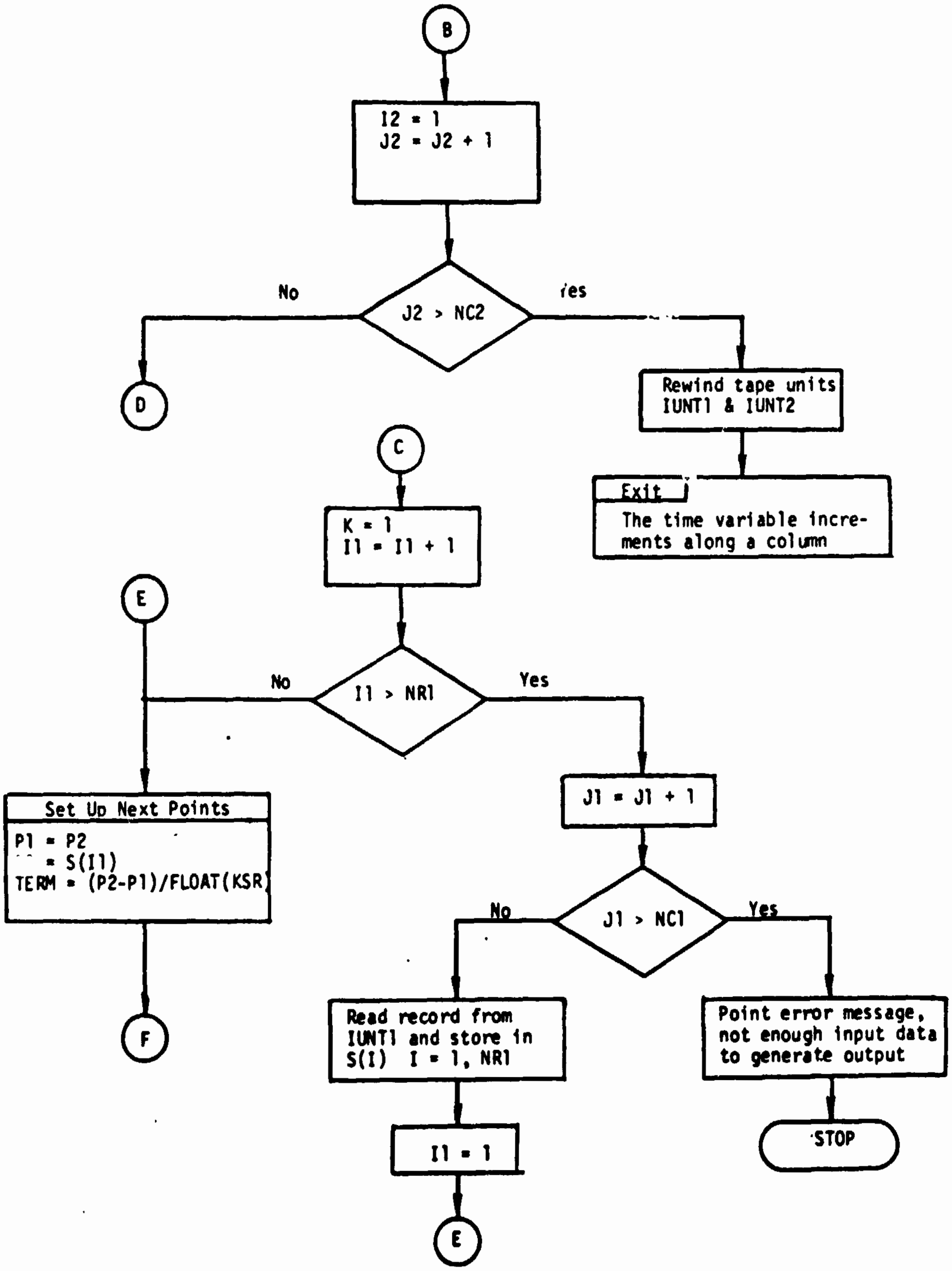
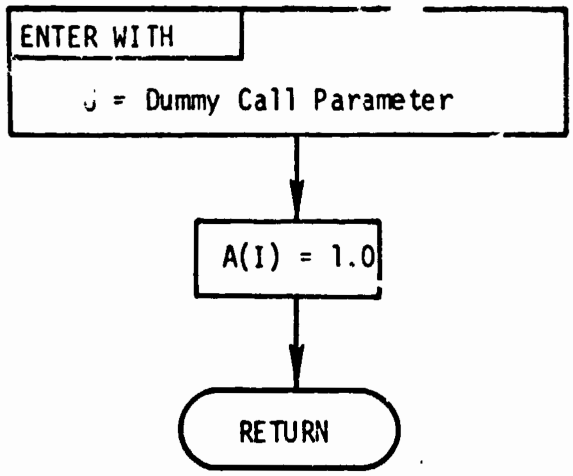


Figure 2-6. Input Signal Module (Cont'd)

Dummy Tape Control Routine* (MAGTAP)



* The sole purpose of this dummy routine is to allocate enough core storage for a CS-1 routine named MAGTAP that will be loaded later from paper tape.

Figure 2-7. Dummy Tape Control Routine* (MAGTAP)

Modulation Module (MOD)

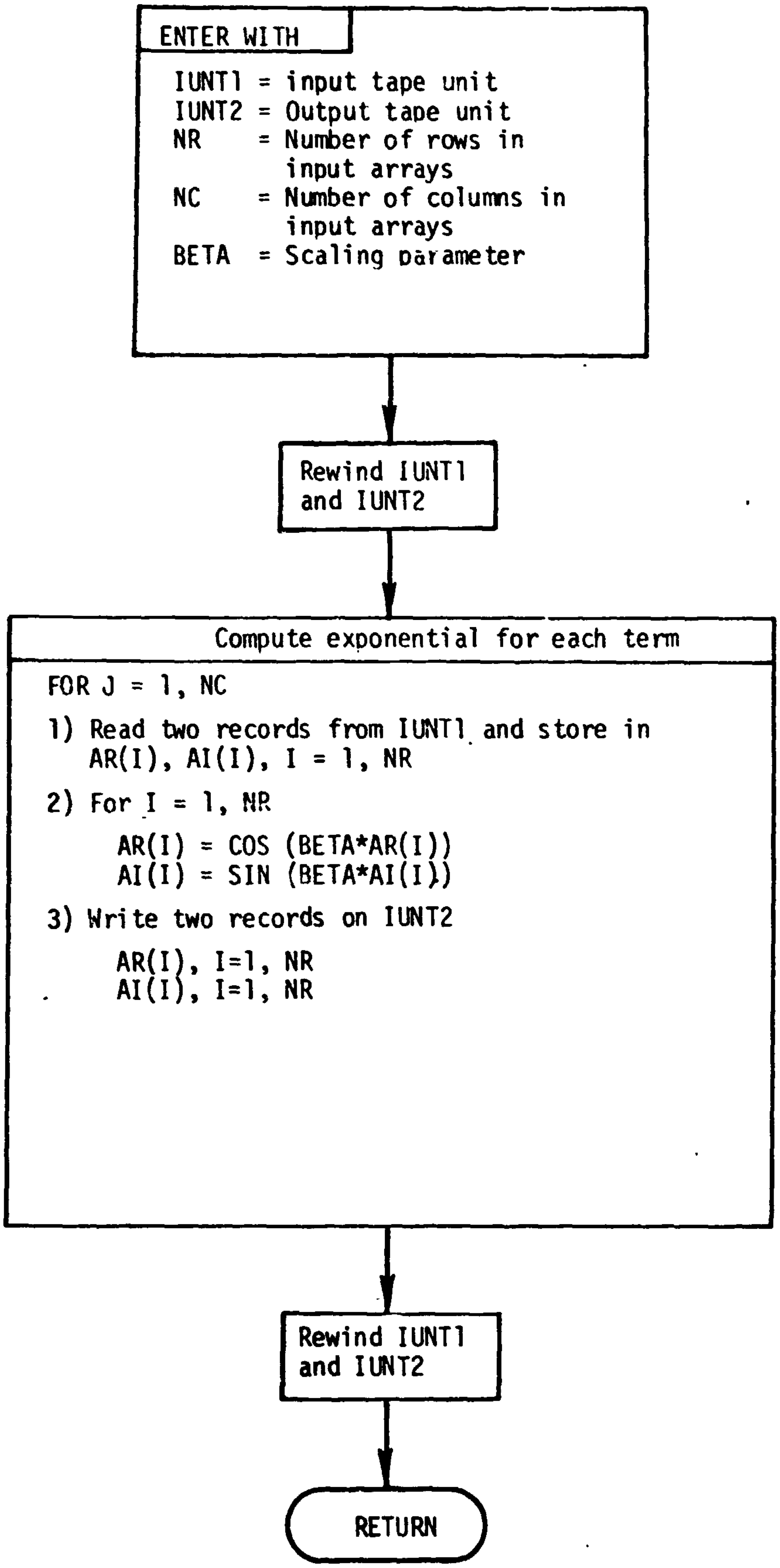
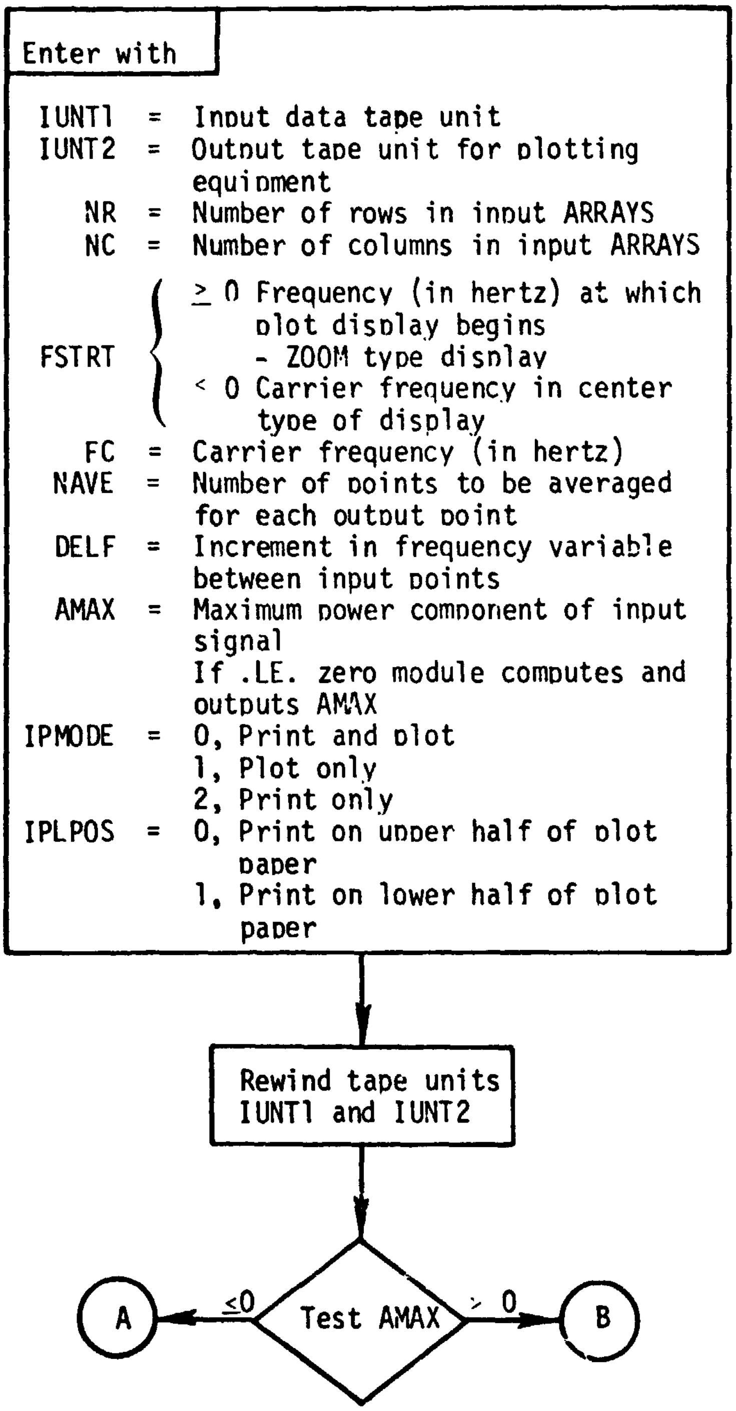


Figure 2-8. Modulation Module (MOD)

PLOT MODULE



Enter with

IUNT1 = Input data tape unit
IUNT2 = Output tape unit for plotting equipment
NR = Number of rows in input ARRAYS
NC = Number of columns in input ARRAYS
FSTRT { ≥ 0 Frequency (in hertz) at which plot display begins - ZOOM type display
< 0 Carrier frequency in center type of display
FC = Carrier frequency (in hertz)
NAVE = Number of points to be averaged for each output point
DELF = Increment in frequency variable between input points
AMAX = Maximum power component of input signal
If .LE. zero module computes and outputs AMAX
IPMODE = 0, Print and plot
1, Plot only
2, Print only
IPLPOS = 0, Print on upper half of plot paper
1, Print on lower half of plot paper

Figure 2-9. Plot Module

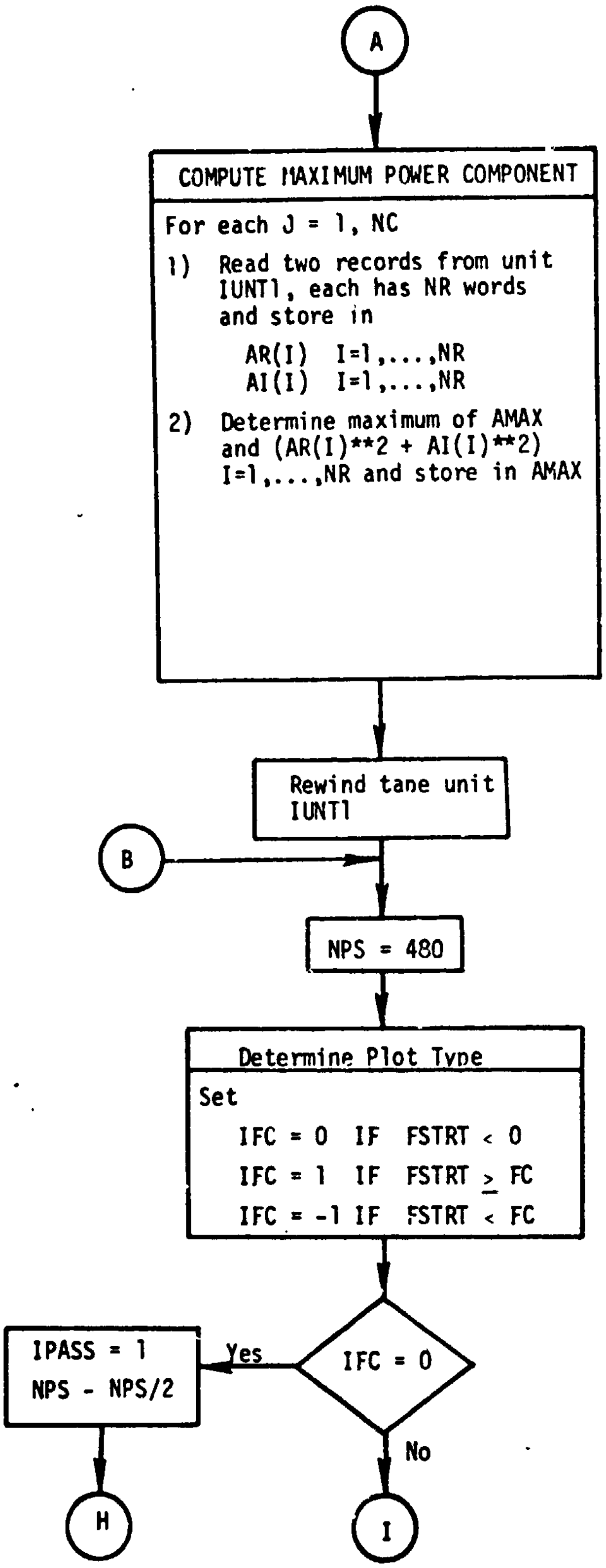


Figure 2-9. Plot Module (Cont'd)

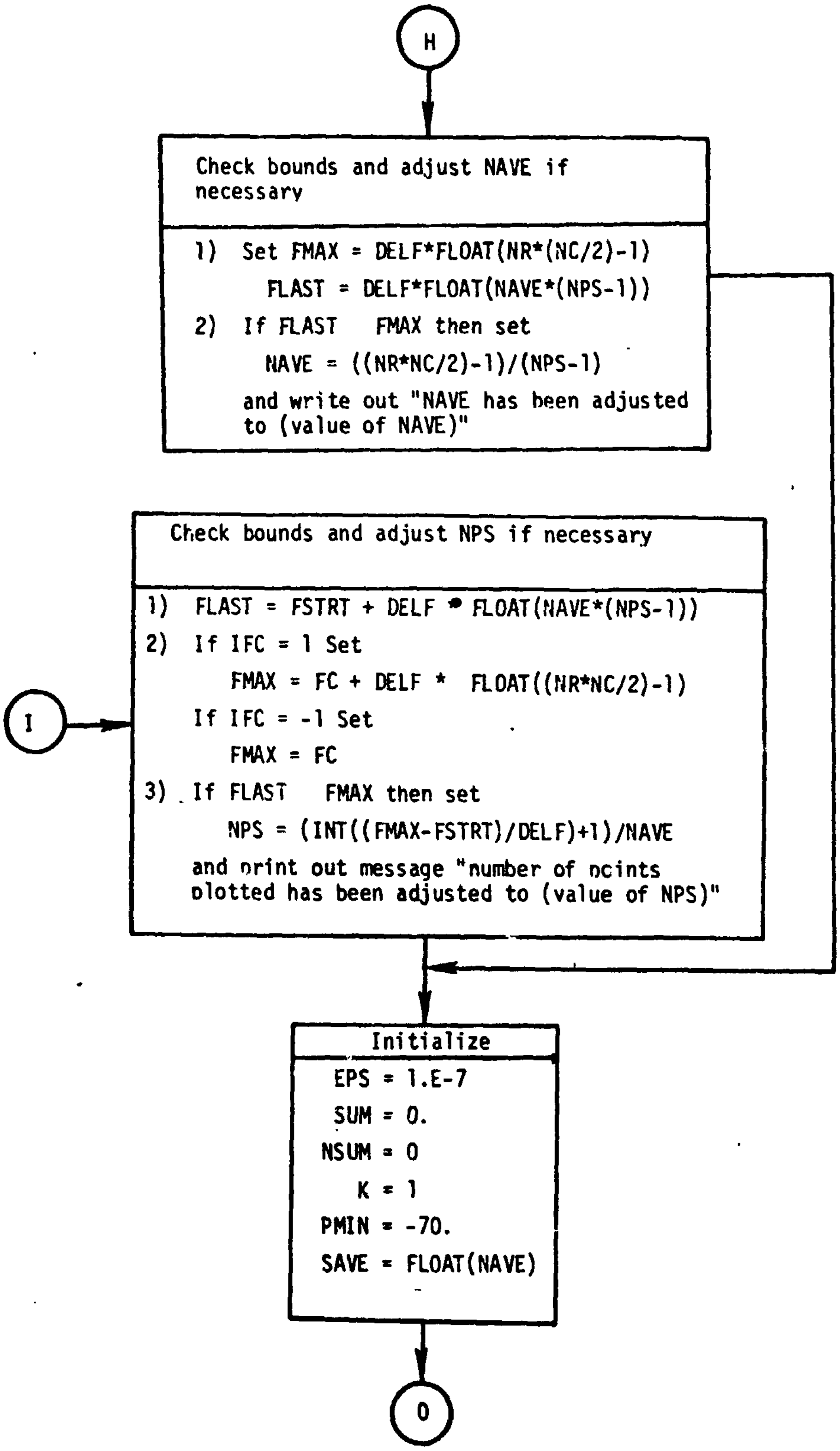

H
Check bounds and adjust NAVE if necessary
1) Set FMAX = DELF*FLOAT(NR*(NC/2)-1)
FLAST = DELF*FLOAT(NAVE*(NPS-1))
2) If FLAST FMAX then set
NAVE = ((NR*NC/2)-1)/(NPS-1)
and write out "NAVE has been adjusted to (value of NAVE)"
Check bounds and adjust NPS if necessary
1) FLAST = FSTRT + DELF * FLOAT(NAVE*(NPS-1))
2) If IFC = 1 Set
FMAX = FC + DELF * FLOAT((NR*NC/2)-1)
If IFC = -1 Set
FMAX = FC
3) If FLAST FMAX then set
NPS = (INT((FMAX-FSTRT)/DELF)+1)/NAVE
and print out message "number of points plotted has been adjusted to (value of NPS)"
I
Initialize
EPS = 1.E-7
SUM = 0.
NSUM = 0
K = 1
PMIN = -70.
SAVE = FLOAT(NAVE)
O

Figure 2-9. Plot Module (Cont'd)

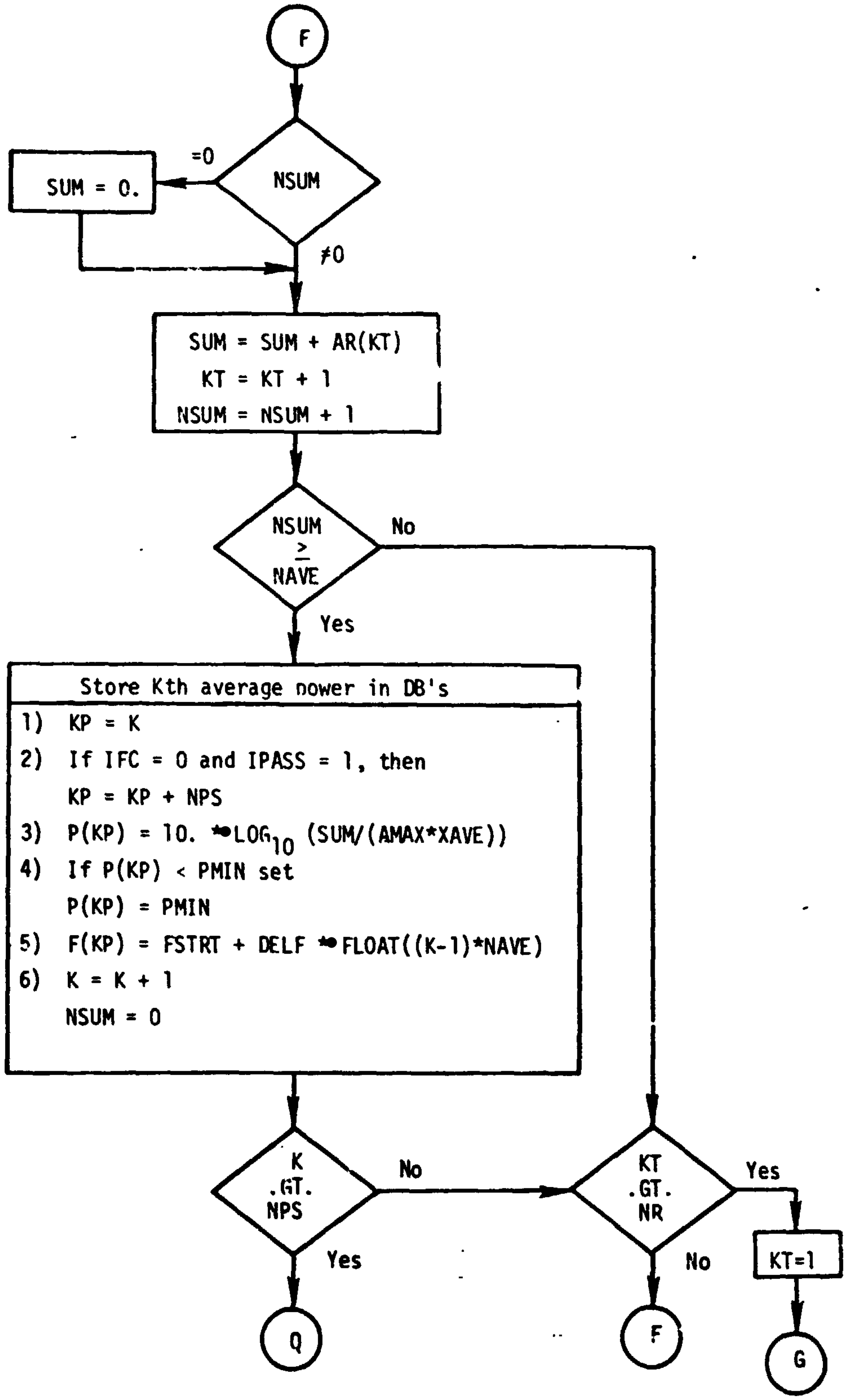


Figure 2-9. Plot Module (Cont'd)

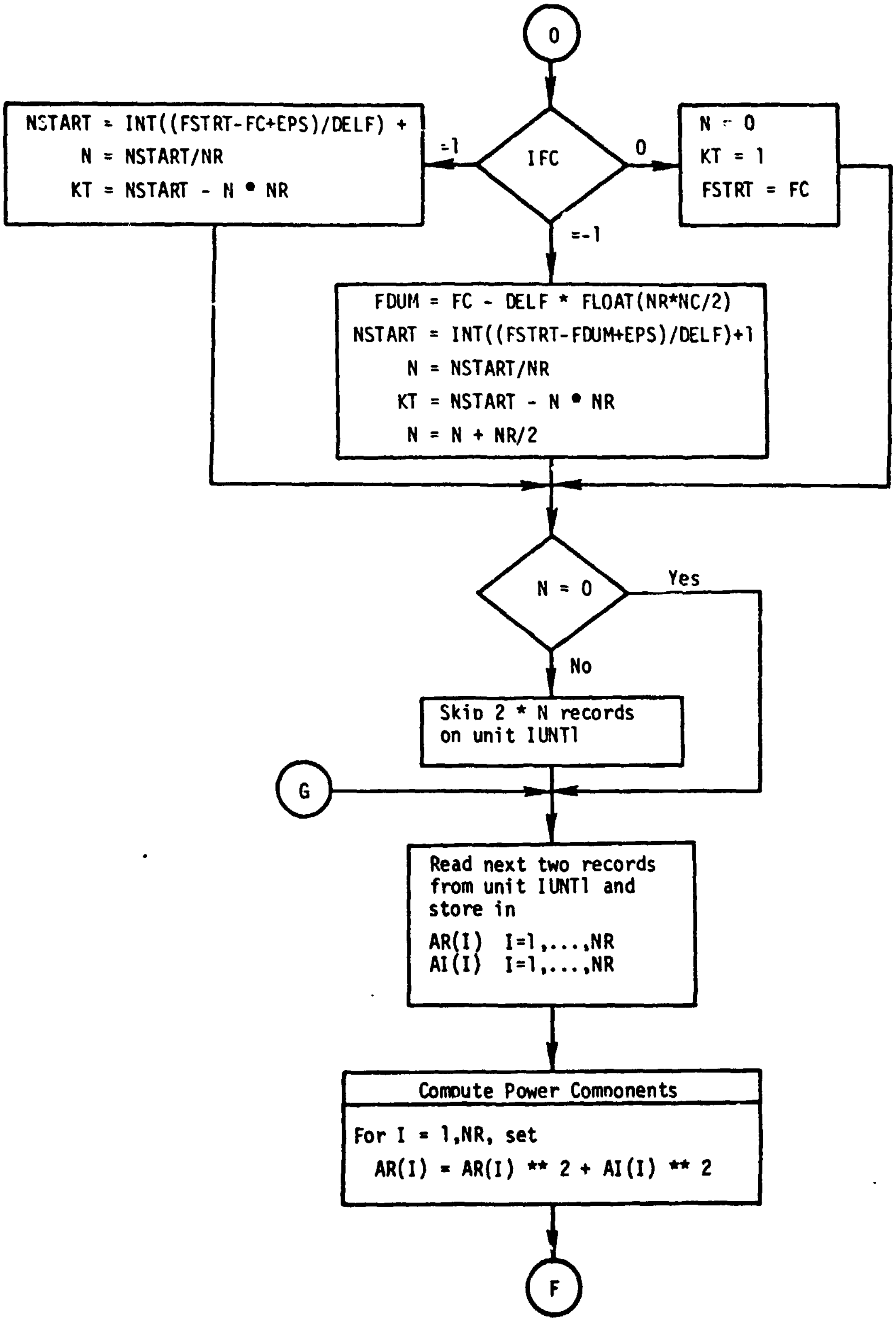

0
IFC
=1
0
=-1
NSTART = INT((FSTRT-FC+EPS)/DELF) +
N = NSTART/NR
KT = NSTART - N • NR
N = 0
KT = 1
FSTRT = FC
FDUM = FC - DELF * FLOAT(NR*NC/2)
NSTART = INT((FSTRT-FDUM+EPS)/DELF)+1
N = NSTART/NR
KT = NSTART - N • NR
N = N + NR/2
N = 0
Yes
No
Skip 2 * N records on unit IUNT1
G
Read next two records from unit IUNT1 and store in
AR(I) I=1,...,NR
AI(I) I=1,...,NR
Compute Power Components
For I = 1,NR, set
AR(I) = AR(I) ** 2 + AI(I) ** 2
F

Figure 2-9. Plot Module (Cont'd)

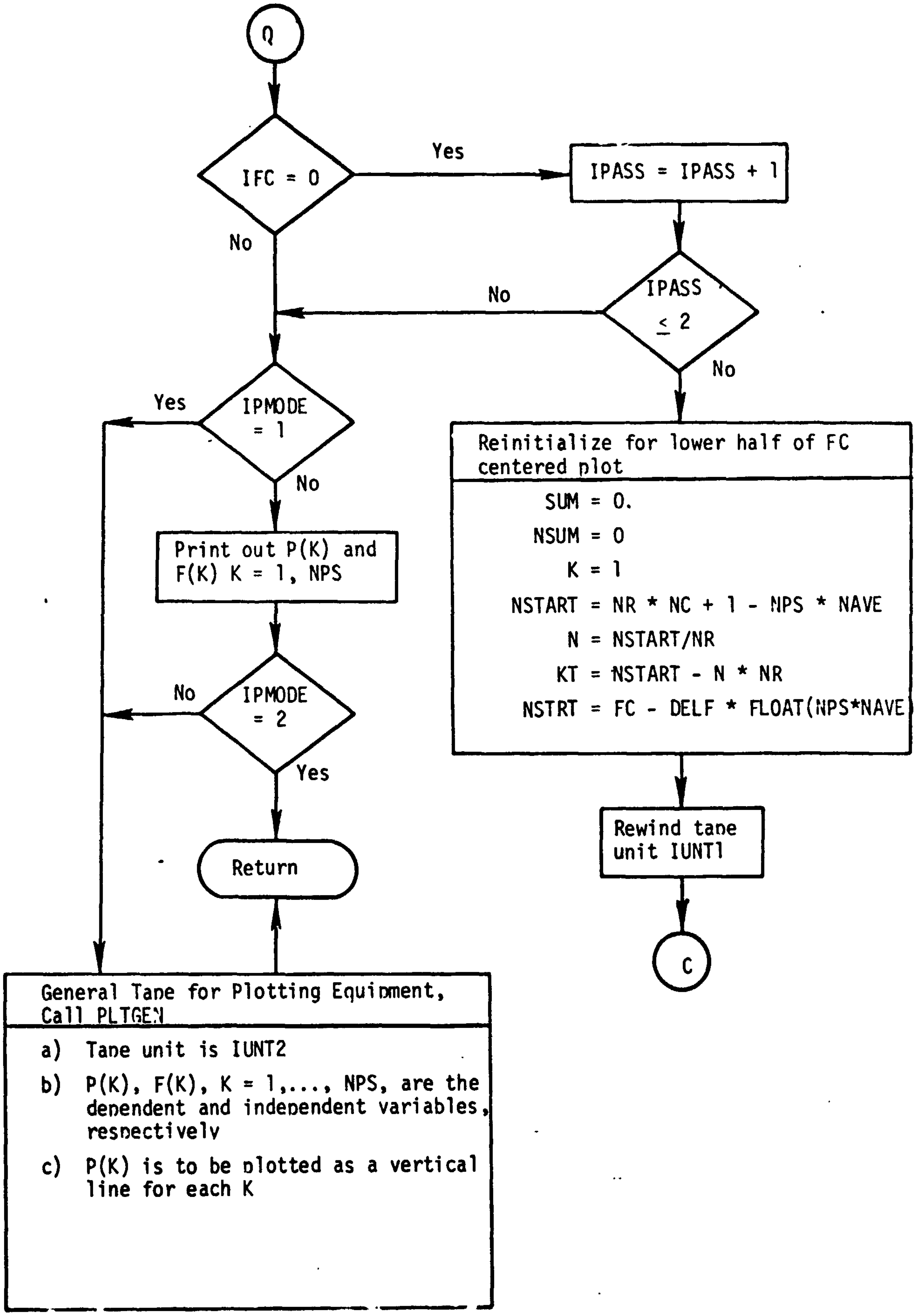

Q
IFC = 0
Yes
IPASS = IPASS + 1
No
IPASS ≤ 2
No
No
Yes
IPMODE = 1
No
Reinitialize for lower half of FC centered plot
SUM = 0.
NSUM = 0
K = 1
NSTART = NR * NC + 1 - NPS * NAVE
N = NSTART/NR
KT = NSTART - N * NR
NSTRT = FC - DELF * FLOAT(NPS*NAVE)
Print out P(K) and F(K) K = 1, NPS
No
IPMODE = 2
Yes
Rewind tape unit IUNT1
Return
C
General Tape for Plotting Equipment, Call PLTGEN
a) Tape unit is IUNT2
b) P(K), F(K), K = 1,..., NPS, are the dependent and independent variables, respectively
c) P(K) is to be plotted as a vertical line for each K

Figure 2-9. Plot Module (Cont'd)

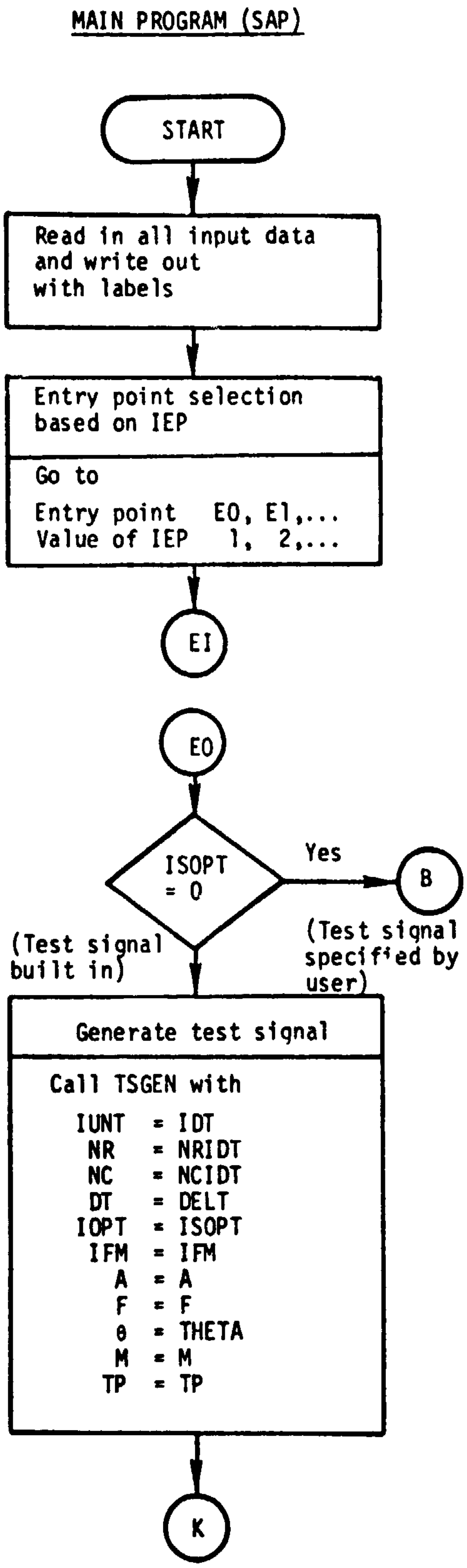
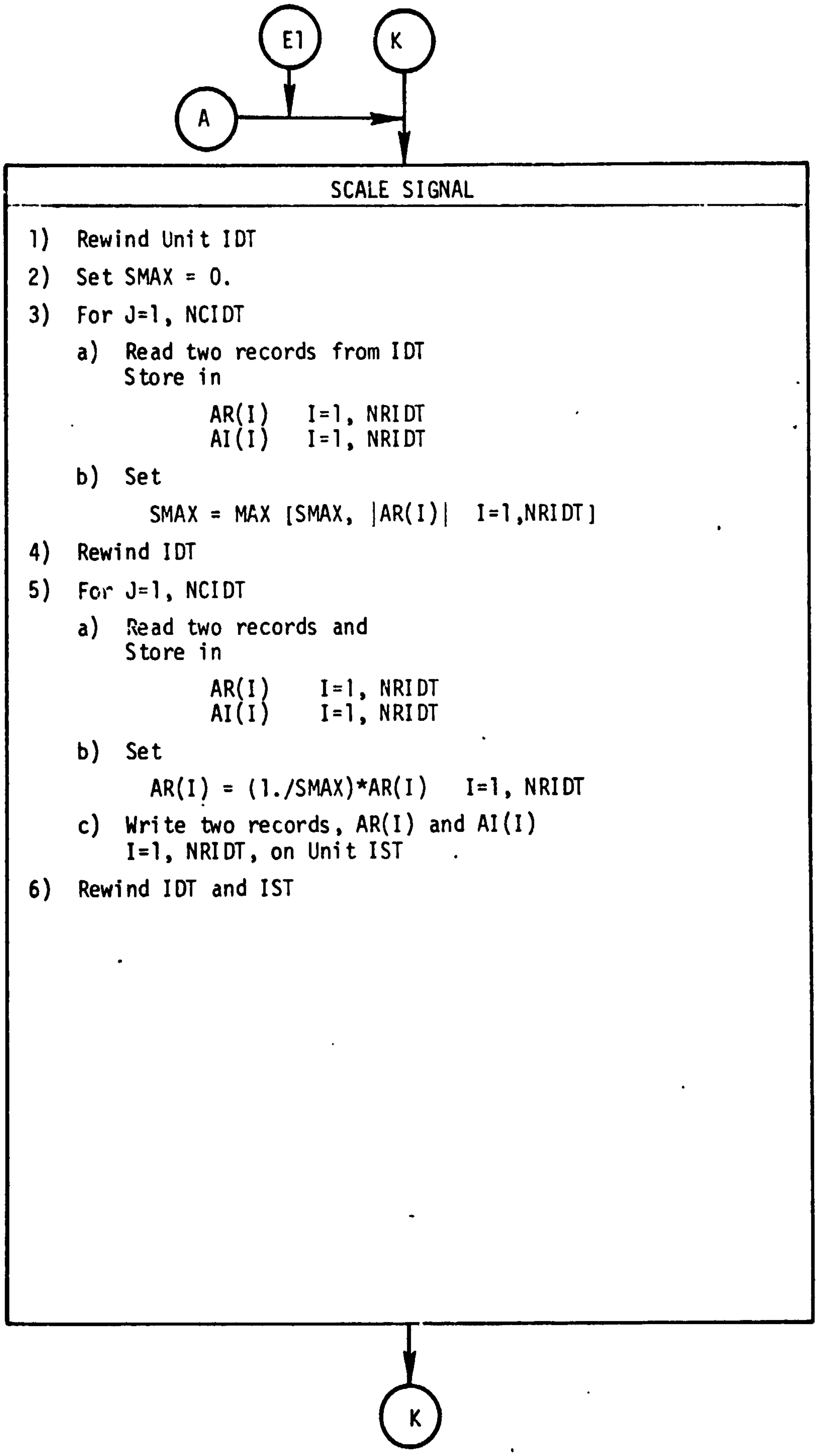


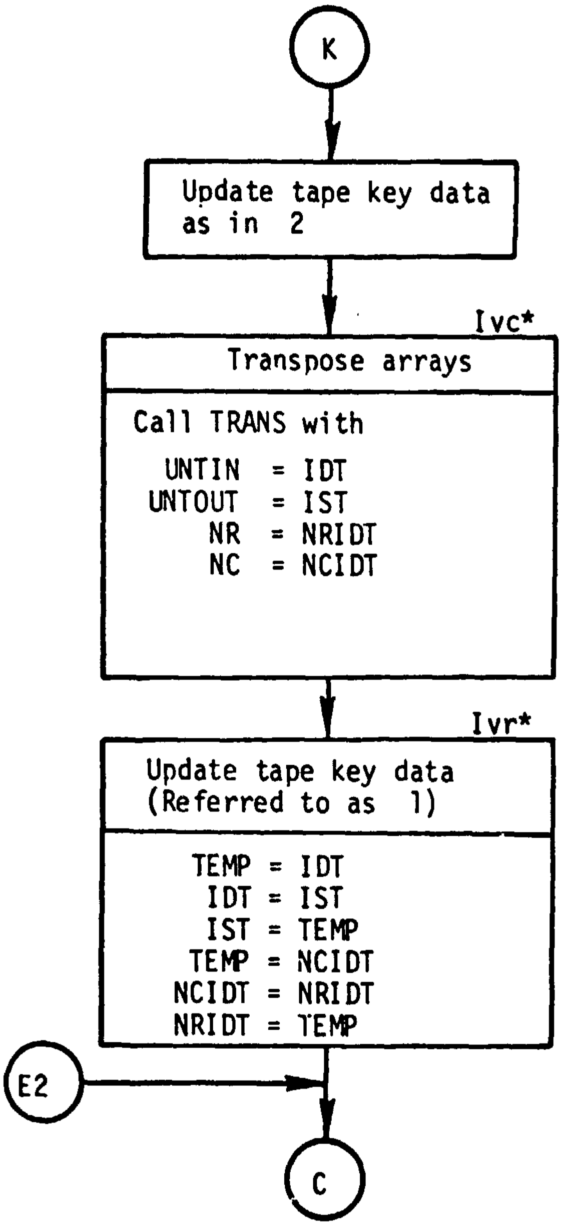
Figure 2-10. Main Program (SAP)



SCALE SIGNAL

```
1)  Rewind Unit IDT
2)  Set SMAX = 0.
3)  For J=1, NCIDT
    a)  Read two records from IDT
        Store in
                AR(I)   I=1, NRIDT
                AI(I)   I=1, NRIDT
    b)  Set
          SMAX = MAX [SMAX, |AR(I)|  I=1,NRIDT]
4)  Rewind IDT
5)  For J=1, NCIDT
    a)  Read two records and
        Store in
                AR(I)    I=1, NRIDT
                AI(I)    I=1, NRIDT
    b)  Set
          AR(I) = (1./SMAX)*AR(I)   I=1, NRIDT
    c)  Write two records, AR(I) and AI(I)
        I=1, NRIDT, on Unit IST
6)  Rewind IDT and IST
```

Figure 2-10. Main Program (SAP) (Cont'd)



*Ivc and Ivr are special notations, for convenience only, defining the relationship between the independent variable (time or frequency) and the rows and columns of the arrays at the point indicated as follows: Ivc - denotes that the independent variable increments along a column (or within a record), thus, the next point is generally in the same record while Ivr - denotes that the independent variable increments along a row (or from record to record), thus, the next point is generally in the next record.

Figure 2-10. Main Program (SAP) (Cont'd)

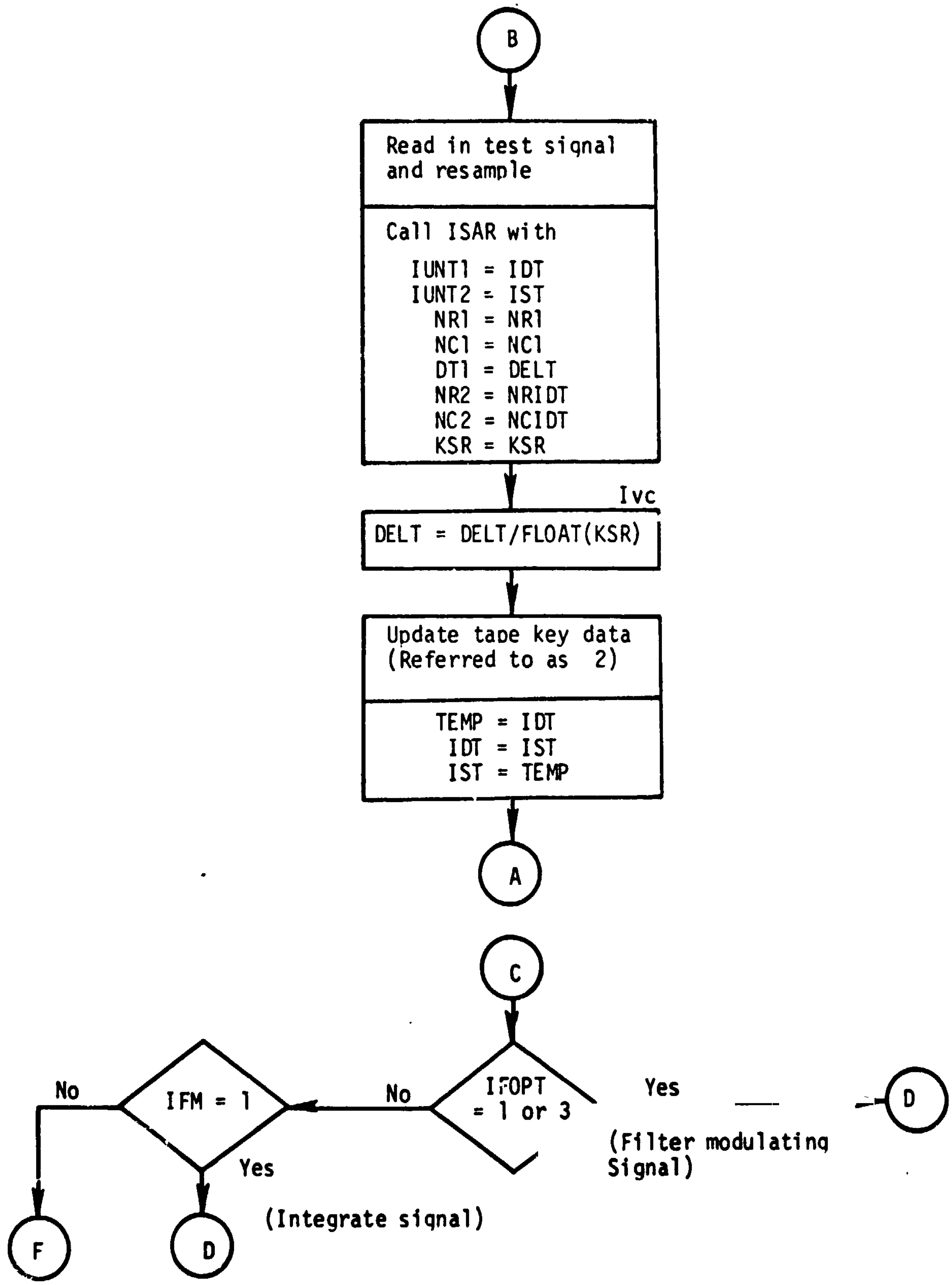


Figure 2-10. Main Program (SAP) (Cont'd)

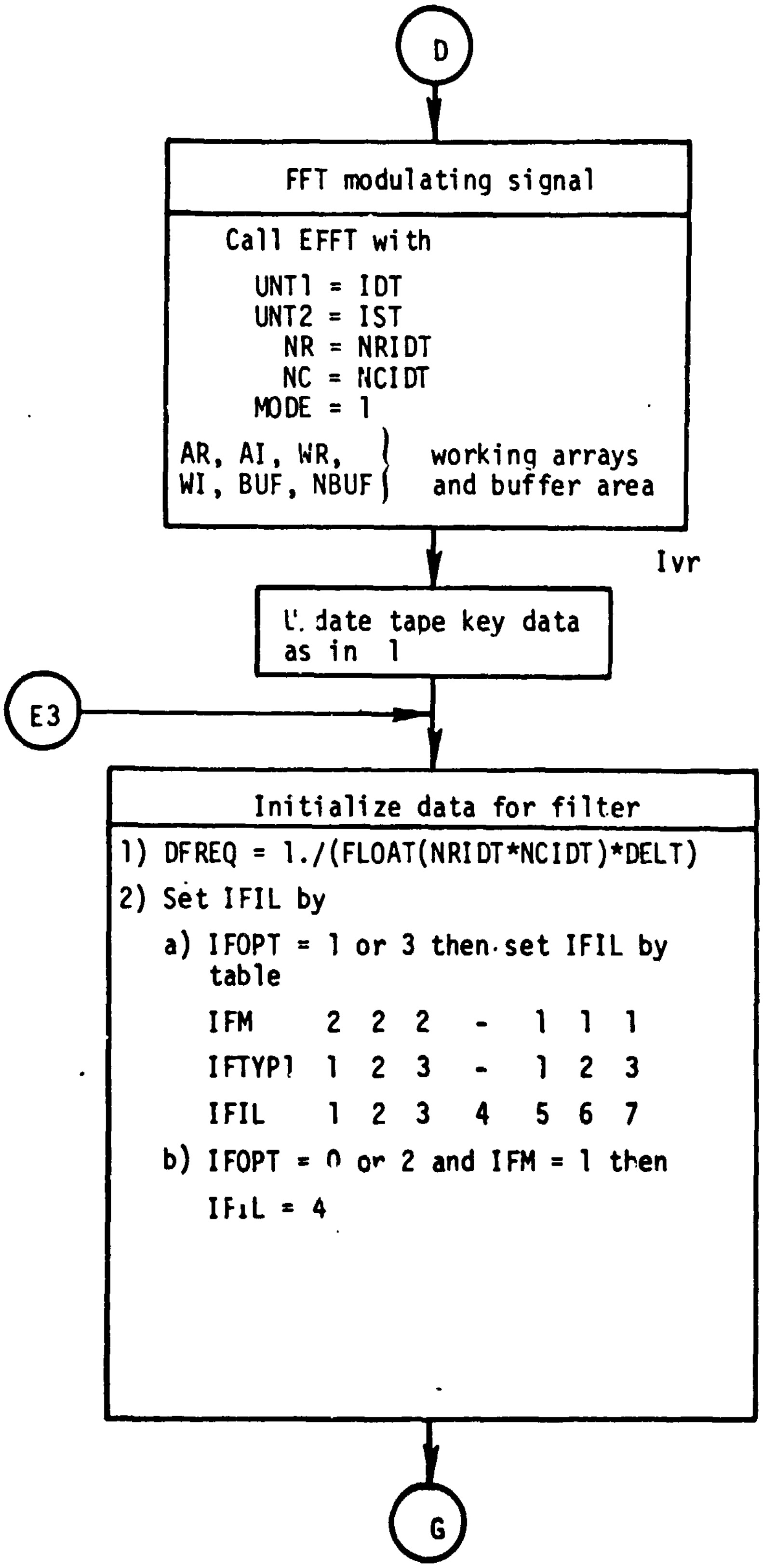

D
FFT modulating signal
Call EFFT with
UNT1 = IDT
UNT2 = IST
NR = NRIDT
NC = NCIDT
MODE = 1
AR, AI, WR, WI, BUF, NBUF
working arrays and buffer area
Ivr
Update tape key data as in 1
E3
Initialize data for filter
1) DFREQ = 1./(FLOAT(NRIDT*NCIDT)*DELT)
2) Set IFIL by
a) IFOPT = 1 or 3 then set IFIL by table
IFM 2 2 2 - 1 1 1
IFTYP1 1 2 3 - 1 2 3
IFIL 1 2 3 4 5 6 7
b) IFOPT = 0 or 2 and IFM = 1 then
IFIL = 4
G

Figure 2-10. Main Program (SAP) (Cont'd)

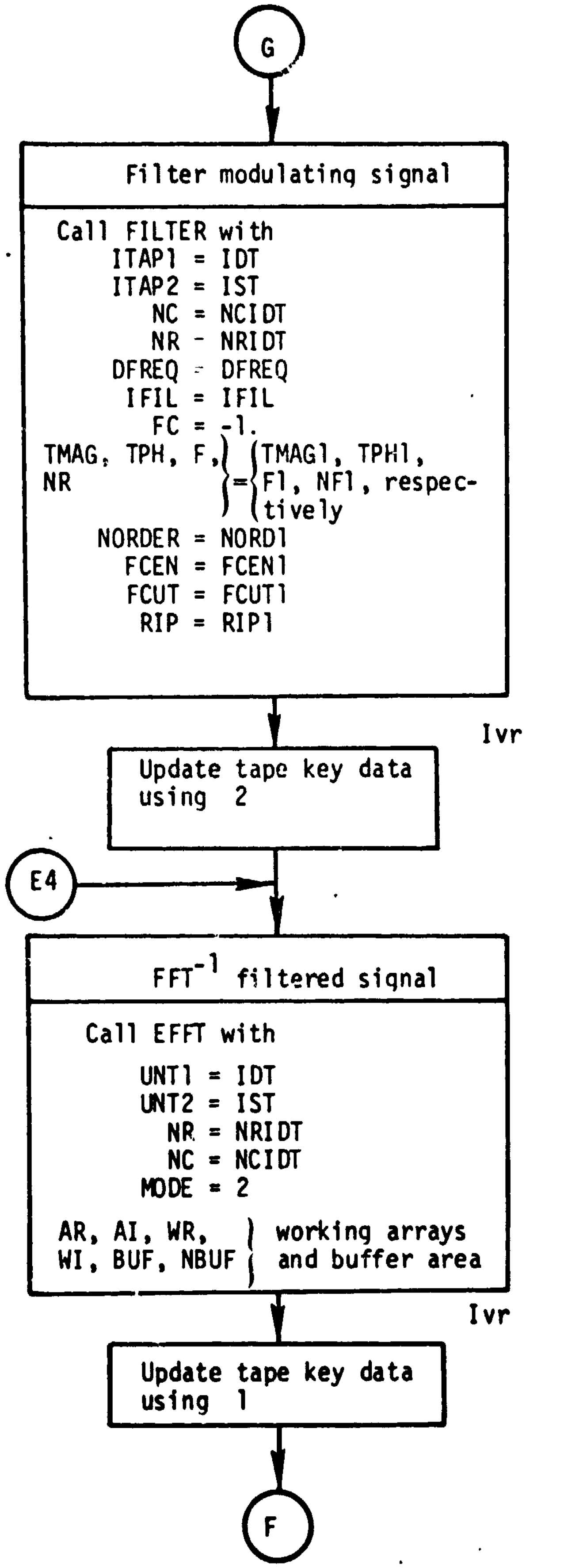


Figure 2-10. Main Program (SAP) (Cont'd)

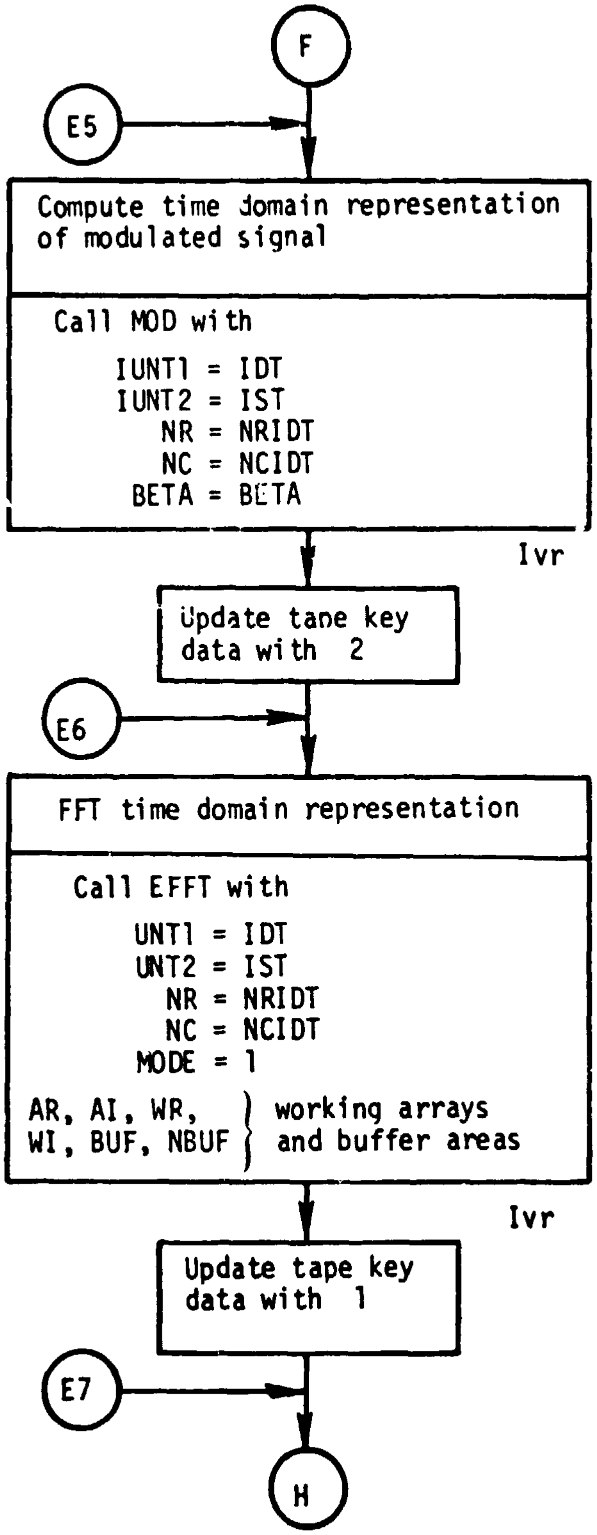

F
E5
Compute time domain representation of modulated signal
Call MOD with
IUNT1 = IDT
IUNT2 = IST
NR = NRIDT
NC = NCIDT
BETA = BETA
Ivr
Update tane key data with 2
E6
FFT time domain representation
Call EFFT with
UNT1 = IDT
UNT2 = IST
NR = NRIDT
NC = NCIDT
MODE = 1
AR, AI, WR, WI, BUF, NBUF
working arrays and buffer areas
Ivr
Update tape key data with 1
E7
H

Figure 2-10. Main Program (SAP) (Cont'd)

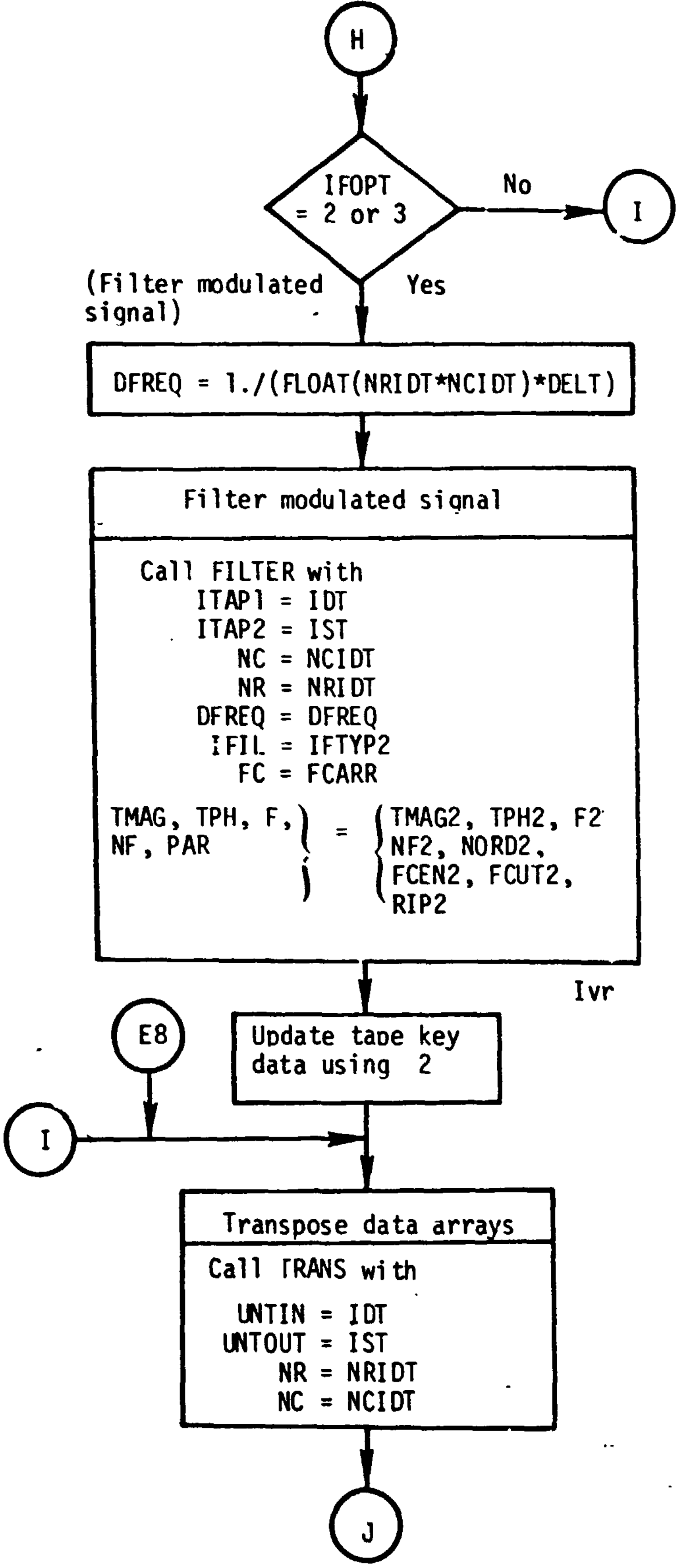


Figure 2-10. Main Program (SAP) (Cont'd)

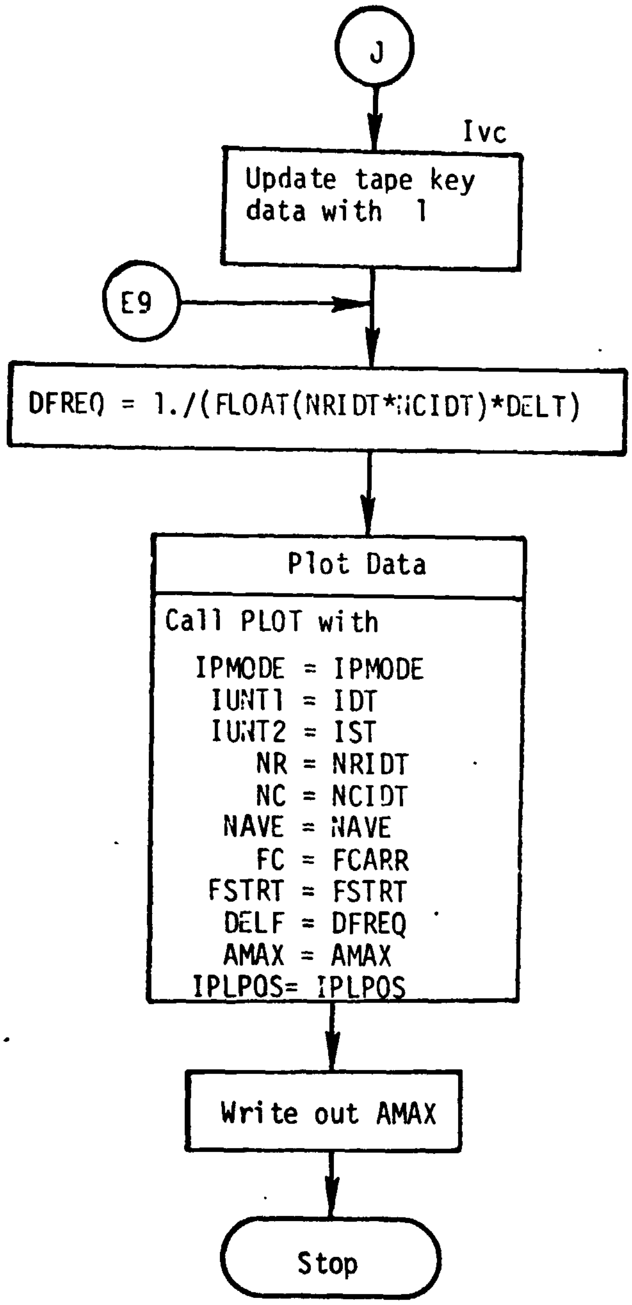


Figure 2-10. Main Program (SAP) (Cont'd)

Timing Difference Comoutation Routine (TIMER)

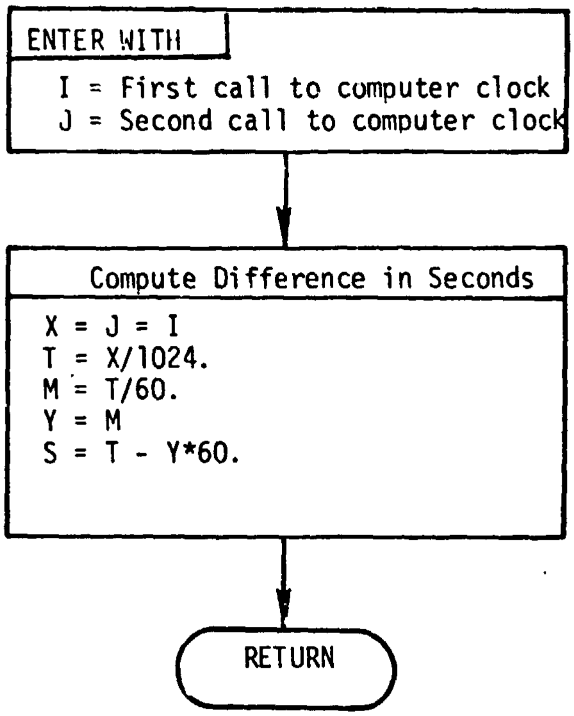

ENTER WITH
I = First call to computer clock
J = Second call to computer clock
Compute Difference in Seconds
X = J = I
T = X/1024.
M = T/60.
Y = M
S = T - Y*60.
RETURN

Figure 2-11. Timing Difference Computation Routine (TIMER)

TRANSFER FUNCTION COMPUTATION ROUTINE

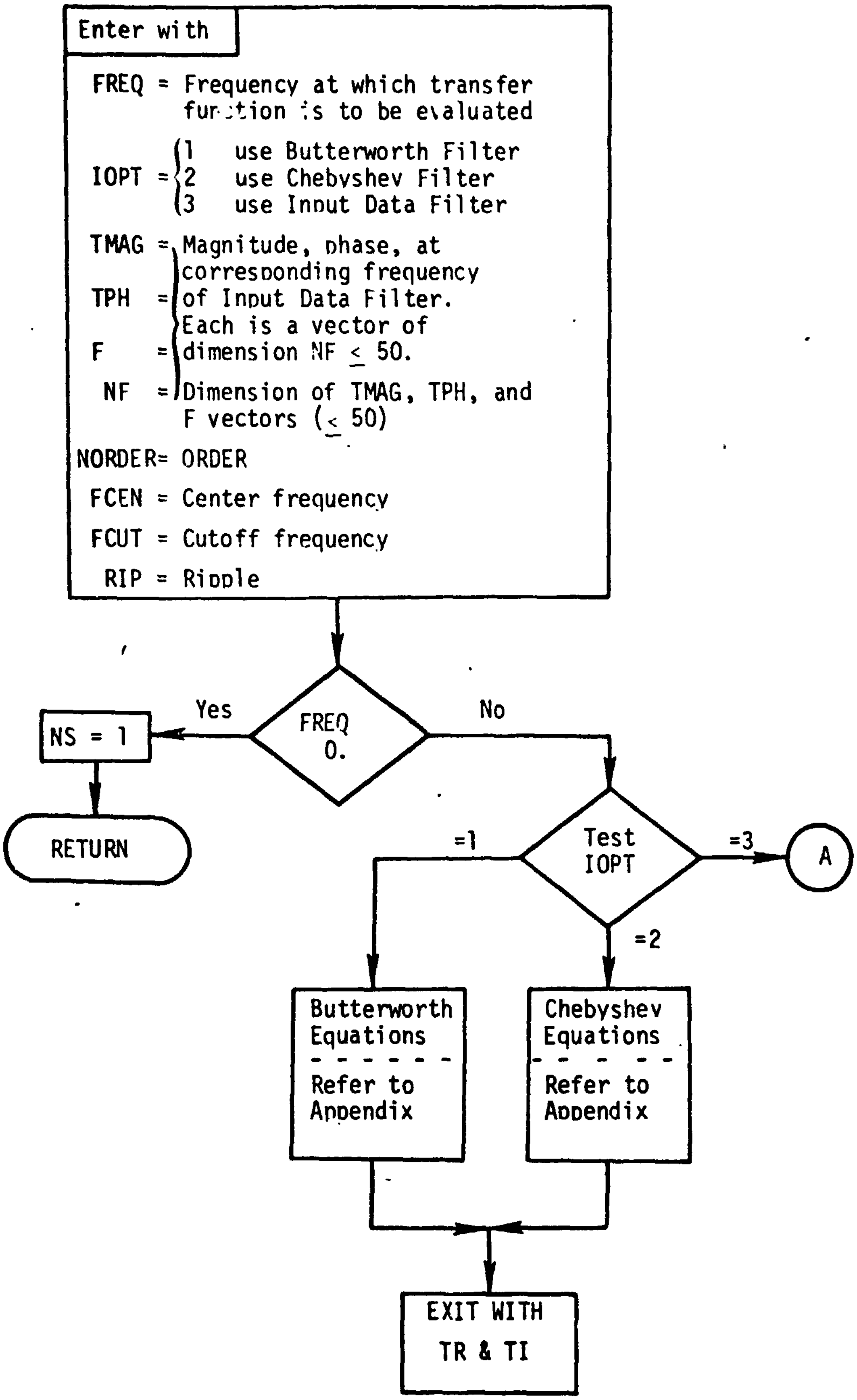

Enter with
FREQ = Frequency at which transfer function is to be evaluated
IOPT = 1 use Butterworth Filter
2 use Chebyshev Filter
3 use Input Data Filter
TMAG = Magnitude, phase, at corresponding frequency
TPH = of Input Data Filter.
F = Each is a vector of dimension NF ≤ 50.
NF = Dimension of TMAG, TPH, and F vectors (≤ 50)
NORDER= ORDER
FCEN = Center frequency
FCUT = Cutoff frequency
RIP = Ripple
FREQ 0.
Yes
No
NS = 1
RETURN
Test IOPT
=1
=2
=3
A
Butterworth Equations
Refer to Appendix
Chebyshev Equations
Refer to Appendix
EXIT WITH TR & TI

Figure 2-12. Transfer Function Computation Routine

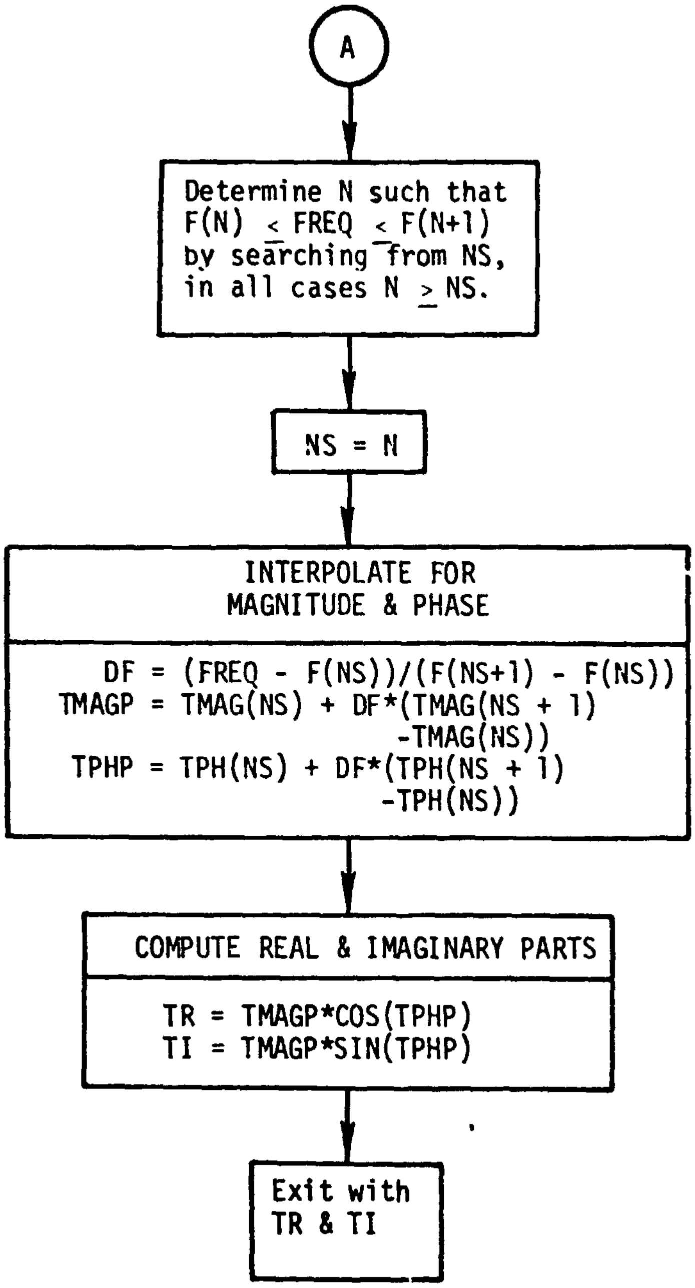
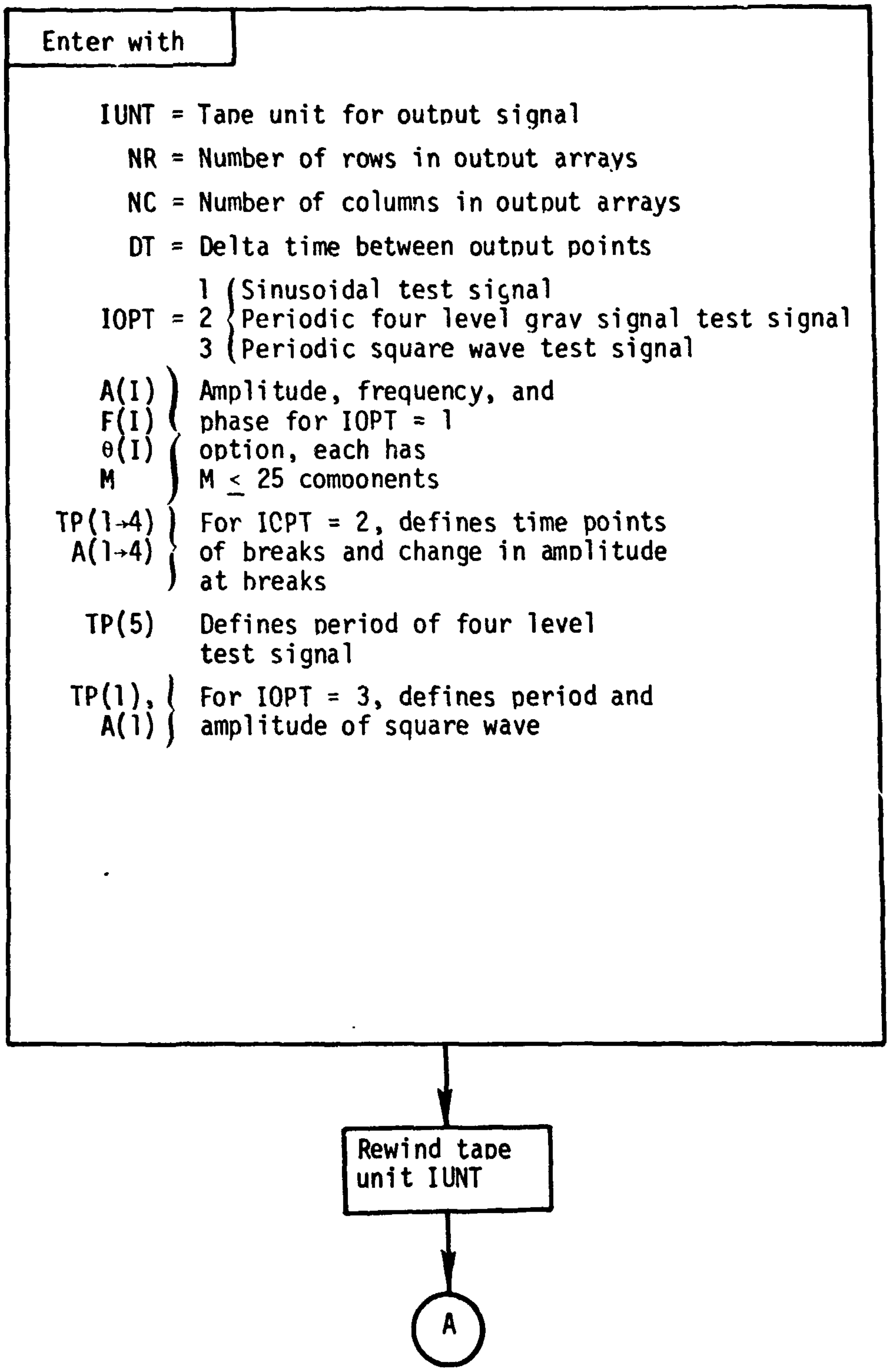

A
Determine N such that F(N) < FREQ < F(N+1) by searching from NS, in all cases N > NS.
NS = N
INTERPOLATE FOR MAGNITUDE & PHASE
DF = (FREQ - F(NS))/(F(NS+1) - F(NS))
TMAGP = TMAG(NS) + DF*(TMAG(NS + 1) -TMAG(NS))
TPHP = TPH(NS) + DF*(TPH(NS + 1) -TPH(NS))
COMPUTE REAL & IMAGINARY PARTS
TR = TMAGP*COS(TPHP)
TI = TMAGP*SIN(TPHP)
Exit with TR & TI

Figure 2-12. Transfer Function Computation Routine (Cont'd)

TEST SIGNAL GENERATION MODULE



Enter with

IUNT = Tape unit for output signal

NR = Number of rows in output arrays

NC = Number of columns in output arrays

DT = Delta time between output points

IOPT = 1 Sinusoidal test signal
IOPT = 2 Periodic four level grav signal test signal
IOPT = 3 Periodic square wave test signal

A(I), F(I), θ(I), M: Amplitude, frequency, and phase for IOPT = 1 option, each has M ≤ 25 components

TP(1→4), A(1→4): For ICPT = 2, defines time points of breaks and change in amplitude at breaks

TP(5): Defines period of four level test signal

TP(1), A(1): For IOPT = 3, defines period and amplitude of square wave

Figure 2-13. Test Signal Generation Module

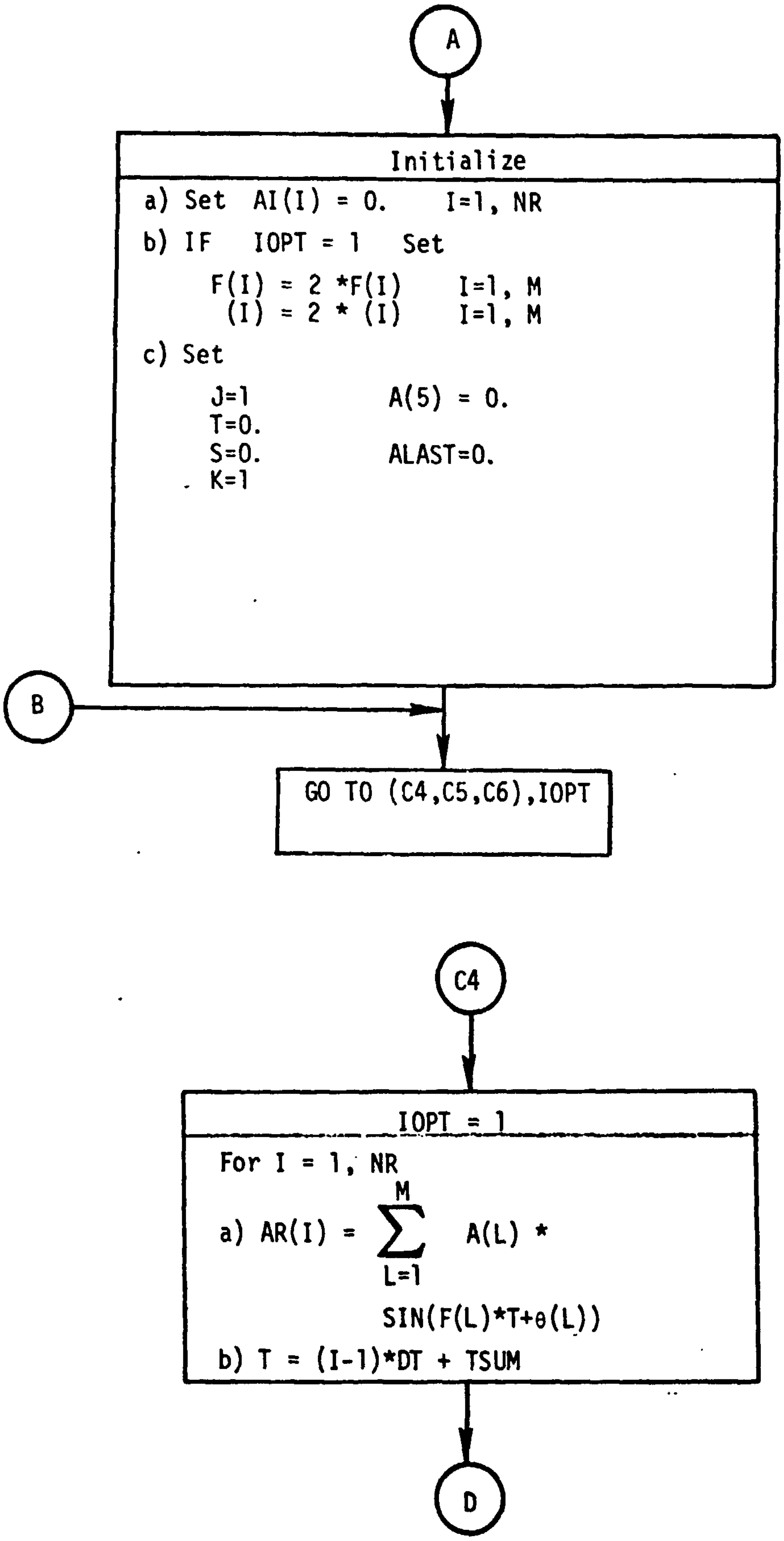
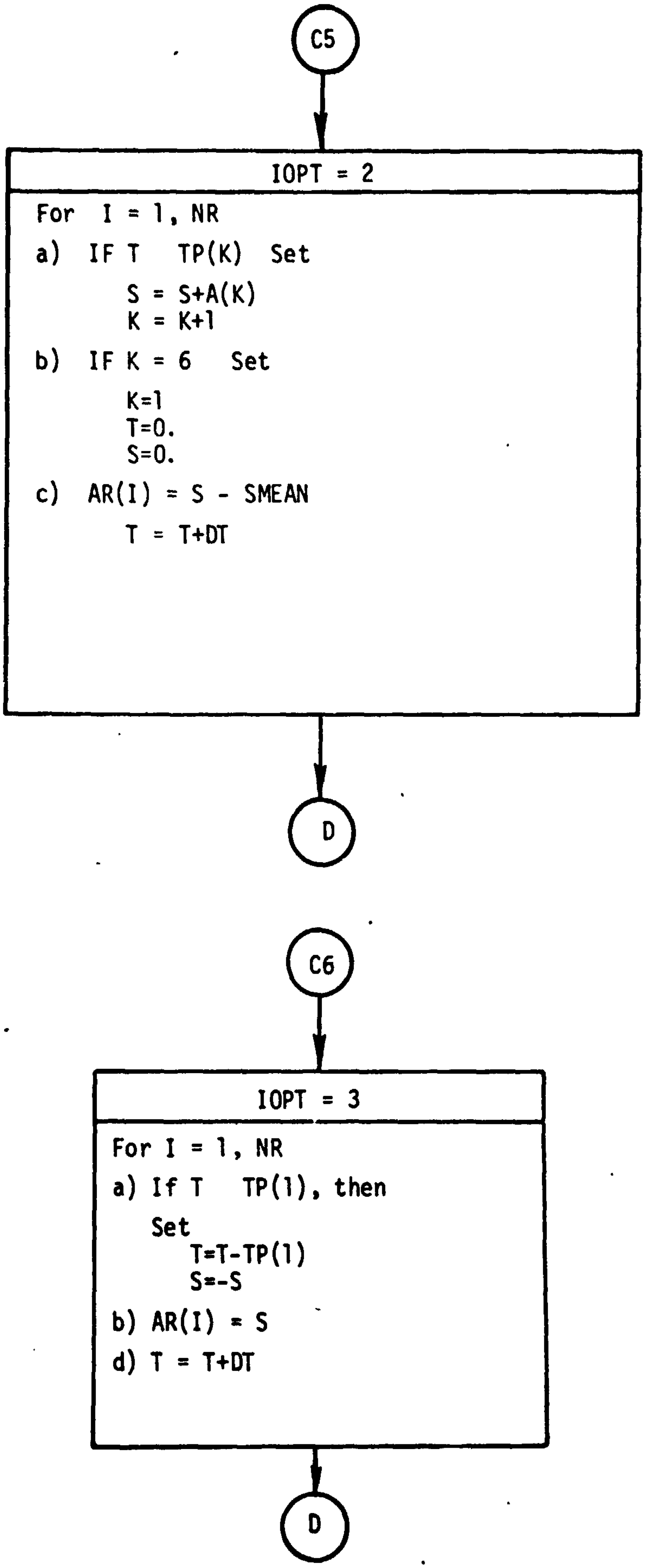


Figure 2-13. Test Signal Generation Module (Cont'd)



```
IOPT = 2
For  I = 1, NR
a)  IF T   TP(K)  Set
        S = S+A(K)
        K = K+1
b)  IF K = 6   Set
        K=1
        T=0.
        S=0.
c)  AR(I) = S - SMEAN
        T = T+DT
```

```
IOPT = 3
For I = 1, NR
a) If T   TP(1), then
   Set
      T=T-TP(1)
      S=-S
b) AR(I) = S
d) T = T+DT
```

Figure 2-13. Test Signal Generation Module (Cont'd)

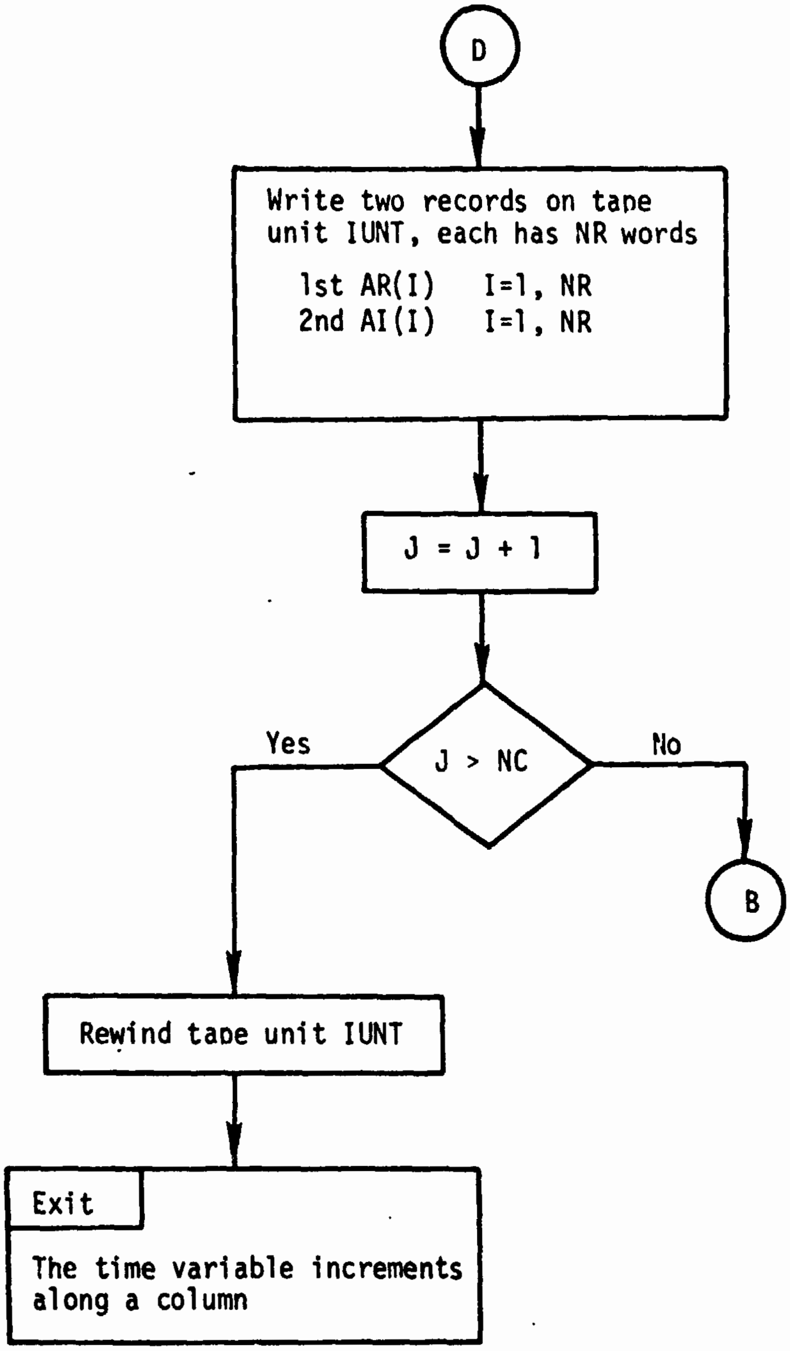
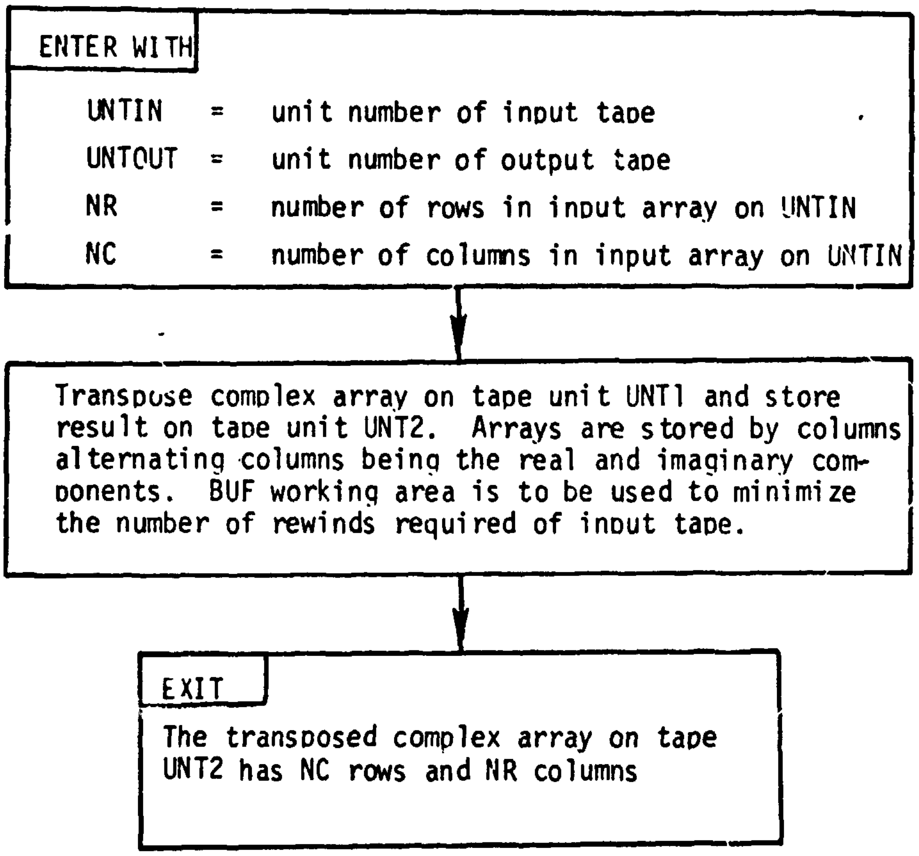

D
Write two records on tape unit IUNT, each has NR words
1st AR(I) I=1, NR
2nd AI(I) I=1, NR
J = J + 1
J > NC
Yes
No
B
Rewind tape unit IUNT
Exit
The time variable increments along a column

Figure 2-13. Test Signal Generation Module (Cont'd)

Transpose Subroutine



ENTER WITH

UNTIN = unit number of input tape

UNTOUT = unit number of output tape

NR = number of rows in input array on UNTIN

NC = number of columns in input array on UNTIN

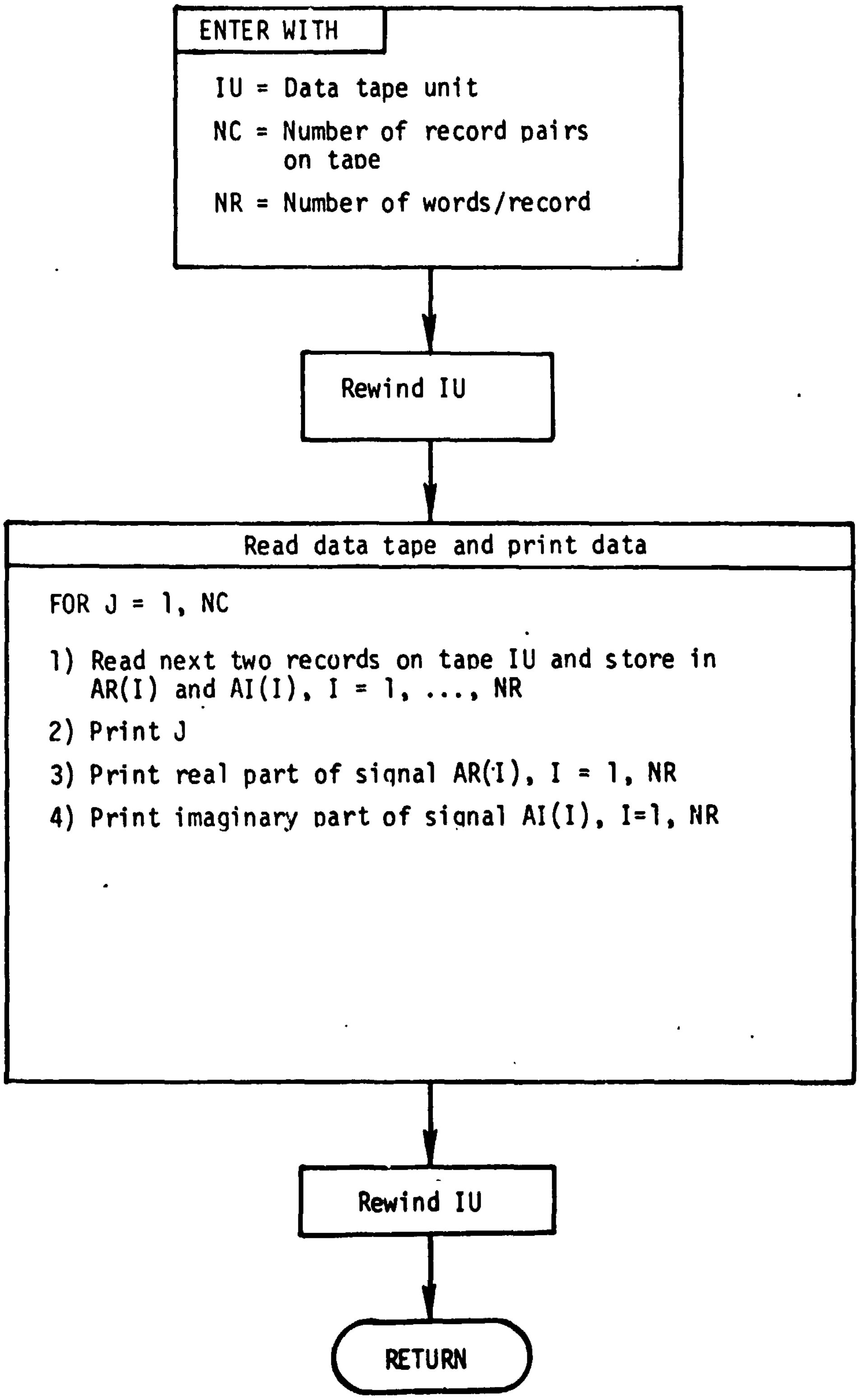
Transpose complex array on tape unit UNT1 and store result on tape unit UNT2. Arrays are stored by columns alternating columns being the real and imaginary components. BUF working area is to be used to minimize the number of rewinds required of input tape.

EXIT

The transposed complex array on tape UNT2 has NC rows and NR columns

Figure 2-14. Transpose Subroutine

Tape Signal Read and Print Routine (TWRITE)


ENTER WITH
IU = Data tape unit
NC = Number of record pairs on tape
NR = Number of words/record
Rewind IU
Read data tape and print data
Rewind IU
RETURN

```
FOR J = 1, NC
1) Read next two records on tape IU and store in
   AR(I) and AI(I), I = 1, ..., NR
2) Print J
3) Print real part of signal AR(I), I = 1, NR
4) Print imaginary part of signal AI(I), I=1, NR
```

Figure 2-15. Tape Signal Read and Print Routine (TWRITE)

2.3 SAP LISTINGS

This section presents a complete listing for the SAP program.

SAP PROGRAM LISTING

```
      MAINPROGRAM  SAP
CD* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
CD
CD    PROGRAMMER AND DATE
CD            RICHARD C. THOMAS
CD            TRW SYSTEMS
CD            JUNE 1972
CD
CD    PURPOSE
CD            THIS IS THE MAIN ROUTINE. IT READS ALL USER INPUTS AND
CD            CONTROLS THE FLOW THROUGH THE SUBROUTINE
CD
CD    DESCRIPTION OF PARAMETERS
CD
CD       INPUT
CD         COMMON
CD         CARD
CD            IDT     - DATA TAPE UNIT
CD            IEP     - PROGRAM ENTRY POINT
CD            IFM     - .EQ.1 FM MODULATION
CD                    - .EQ.2 PHASE MODULATION
CD            IFOPT   - .EQ.0 NO FILTERING
CD                    - .EQ.1   FILTER MODULATING OF TEST SIGNAL ONLY
CD                    - .EQ.2   FILTER MODULATED SIGNAL ONLY
CD                    - .EQ.3   FILTER TEST AND MODULATED SIGNAL
CD            IFTYP1  - .EQ.1 BUTTERWORTH FILTER
CD                    - .EQ.2 CHEBYCHEV FILTER
CD                    - .EQ.3 INPUT FILTER
CD            IFTYP2  - .EQ.1 BUTTERWORTH FILTER
CD                    - .EQ.2 CHEBYCHEV FILTER
CD                    - .EQ.3 INPUT FILTER
CD            IPLPOS  - .EQ.0 PLOT WILL BE ON UPPER HALF OF PLOT PAPER
CD                      .EQ.1 PLOT WILL BE ON LOWER HALF OF PLOT PAPE
CD            IPMODE  - .EQ.0 PRINT POWER SPECTRUM AND GENERATE DATA
CD                      .EQ.1 ONLY GENERATE DATA TAPE
CD                      .EQ.2 ONLY PRINT POWER SPECTRUM
CD            ISOPT   - .EQ.0 USER INPUT SIGNAL ON TAPE IDT
CD                    - .EQ.1 SINUSOIDAL TEST SIGNAL
CD                    - .EQ.2 PERIODIC FOUR LEVEL GRAY SIGNAL TEST
CD                    - .EQ.3 PERIODIC SQUARE WAVE TEST SIGNAL
```

SAP PROGRAM LISTING

```
CD                IST     - SPARE TAPE UNIT
CD                KSR     - DEFINES SAMPLING RATE FOR RESAMPLED SIGNAL
CD                NBUF    - SIZE OF BUFFER FOR TAPE TRANSPOSE ROUTINE,TTRANS
CD                NCTOT   - NUMBER OF COLUMNS IN DATA MATRIX
CD                NF1     - DEFINES NUMBER OF POINTS FOR INPUT FILTER,
CD                          IFTYP1 = 3
CD                NF2     - DEFINES NUMBER OF POINTS FOR INPUT FILTER,
CD                          IFTYP2 = 3
CD                NRTOT   - NUMBER OF ROWS IN DATA MATRIX
CD                NR1     - NUMBER OF RECORDS ON INPUT DATA TAPE
CD                NAVE    - NUMBER OF POINTS TO BE AVERAGED IN OUTPUT
CD                I2PRT   - THIS ARRAY CONTROLS THE PRINTING OF THE SIGNAL
CD                          ON THE DATA TAPE. THE SIGNAL ON THE TAPE WILL BE
CD                          PRINTED BY A CALL TO TWRITE IF THE PARAMETER IS
CD                          EQUAL TO 1.
CD                I2PRT(1) REFERS  TO PRINT AFTER TSGEN OR ISAR
CD                I2PRT(2) REFERS  TO PRINT AFTER FIRST CALL TO EFFT
CD                I2PRT(3) REFERS  TO PRINT AFTER FIRST CALL TO FILTER
CD                I2PRT(4) REFERS  TO PRINT AFTER SECOND CALL TO EFFT
CD                I2PRT(5) REFERS  TO PRINT AFTER THIRD CALL TO EFFT
CD                I2PRT(6) REFERS  TO PRINT AFTER SECOND CALL TO FILTER
CD                AMAX    - MAXIMUM POWER COMPONENT OF INPUT SIGNAL
CD                BETA    - MODULATION INDEX, MULTIPLIES ENTIRE TEST SIGNAL
CD                DELT    - TIME SEPARATION(IN SECONDS) BETWEEN POINTS IN
CD                          TEST SIGNAL OR BETWEEN USER INPUT SIGNAL PRIOR
CD                          TO RESAMPLING
CD                ITITLE  - UP TO 30 CHARACTERS FOR RUN TITLE AND/OR DATE
CD                FC      - CARRIER FREQUENCY
CD                F1      - FREQUENCY(IN HERTZ) AT WHICH MAGNITUDE AND PHASE
CD                          IS SPECIFIED FOR MODULATED SIGNAL
CD                F2      - FREQUENCY(IN HERTZ) AT WHICH MAGNITUDE AND PHASE
CD                          IS SPECIFIED FOR MODULATING SIGNAL
CD                FCEN1   - CENTER FREQUENCY FOR BUTTERWORTH OR CHEBYCHEV
CD                FCEN2   - CENTER FREQUENCY FOR BUTTERWORTH OR CHEBYCHEV
CD                RIP1    - RIPPLE FOR BUTTERWORTH FILTER
CD                TMAG1   - MAGNITUDE OF INPUT FILTER TRANSFER FUNCTION
CD                TMAG2   - MAGNITUDE OF INPUT FILTER TRANSFER FUNCTION
CD                NORD1   - ORDER OF THE BUTTERWORTH OR CHEBYCHEV FILTER
CD                NORD2   - ORDER OF THE BUTTERWORTH OR CHEBYCHEV FILTER
CD                FCUT1   - CUTOFF FREQUENCY FOR BUTTERWORTH OR CHEBYCHEV
CD              FCUT2   - CUTOFF FREQUENCY FOR BUTTERWORTH OR CHEBYCHEV
CD              RIP2    - RIPPLE FOR BUTTERWORTH FILTER
CD              IPH1    - CORRESPONDING PHASE(IN RADIANS)
CD              IPH2    - CORRESPONDING PHASE(IN RADIANS)
CD              M       - NUMBER OF SINUSOIDS(LE.25)
CD              A(I)    - AMPLITUDE OF I SINUSOID
CD              F(I)    - FREQUENCY(IN HERTZ) OF I SINUSOID
CD              THETA   - PHASE ANGLE(IN DEGREES) OF I SINUSOID
CD              A       - .IOPT.EQ.2 CHANGE IN AMPLITUDE OF BREAKS
CD              A       - .IOPT.EQ.3 AMPLITUDE OF SQUARE WAVE
CD              IP      - .IOPT.EQ.3 PERIOD OF SQUARE WAVE
CD          TAPE
CD
CD        OUTPUT
CD          COMMON
CD          PRINT
CD              LOCAT   - LAST ENTRY POINT PASSED
CD              IDT     - DATA TAPE UNIT
CD              IST     - SPARE TAPE UNIT
CD              NRIDT   - NUMBER OF RECORD PAIRS CURRENTLY ON DATA TAPE
CD              NCIDT   - NUMBER OF WORDS/RECORD CURRENTLY ON 41T1 T175
CD              MIN     - TOGETHER MIN AND SECS GIVE THE ELAPSED TIME OF A
CD              SECS      SEGMENT OF THE PROGRAM.
CD          TAPE
CD
CD      REMARKS AND RESTRICTIONS
CD      SUBROUTINES REQUIRED
CD              BITREV
CD              CLOCK
CD              EFFT
CD              FFT
CD              FILTER
CD              ISAR
CD              MAGTAP
CD              MOD
CD              PLOT
CD              TIMER
CD              TRFN
CD              TSGEN
CD              TTRANS
```

SAP PROGRAM LISTING

```
CD                TWRITE
CD
CD* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
      DIMENSION Z1[2048],Z2[2048)
      DIMENSION AR[256],AI[256],WR[256),WI[256)
      DIMENSION F1[50),F2[50),PAR1[50),PAR2[50),TMAG1[50),TMAG2[50)
     1,TPH1[50),TPH2[50),A[25),F[25),THETA[25),TP[6),ITITLE[30)
      DIMENSION IZPRT[6)
      DIMENSION TMAG[25),TPH[25)
      COMMON/STORGE/AR,AI,Z1,Z2
      COMMON/TITL/ITITLE
      EQUIVALENCE[WR[1),Z1[1))
      EQUIVALENCE[WI[1),Z2[1))
      CALL CLOCK[ITIME1)
      I1 = 1
      I2 = 2
      I4 = 4
      I7 = 7
      CALL MAGTAP[I4,I4,AR,I1)
      CALL MAGTAP[I4,I4,AR,I2)
      READ[5,1002) [ITITLE[I),I=1,30)
 1002 FORMAT[30A1)
      READ[5,1001) IDT,    IEP    ,IFM    ,IFOPT ,IFTYP1,IFTYP2,IPLPOS,
     1        IPMODE,ISOPT ,IST    ,KSR    ,NCIDT ,NC1    ,NF1    ,NF2    ,
     2        NRIDT ,NR1    ,NAVE   ,[IZPRT[I),I=1,6)
 1001 FORMAT[16I5)
      READ[5,1005) AMAX,BETA,DELT,FC,FSTRT
      WRITE[6,1998)
 1998 FORMAT[1H1)
      WRITE[6,1999)
 1999 FORMAT[1H ,132H*************************************************
     1*****************************************************
     2***************)
      WRITE[6,2000) [ITITLE[I),I=1,30)
 2000 FORMAT[2H *48X2H* 30A1,2H *47X1H*)
      WRITE[6,1999)
      WRITE[6,2005)IDT    ,IEP    ,IFM    ,IFOPT ,IFTYP1,IFTYP2,IPLPOS,
     1        IPMODE,ISOPT ,IST    ,KSR    ,NCIDT ,NC1    ,NF1    ,NF2    ,
     2        NRIDT ,NR1    ,NAVE   ,[IZPRT[I),I=1,6)
 2005 FORMAT[1H0,16X98HIDT IEP IFM IFOPT IFTYP1 IFTYP2 IPLPOS IPMODE ISO
```

```
     1APT IST NSR NCIDT NCJ NF1 NF2 NRIDT NK1 NAVE IZPRT/18XI1,2XI2,3XI1,
     24XI1,6XI1,6XI1,6XI1,6XI1,5XI1,4XI1,3XI1,3XI3,2XI3,2XI1,3XI1,2XI3,
     32XI3,2XI3,4XI1,1XI1,1XI1,1XI1,1XI1,1XI1)
      WRITE[6,2016) AMAX,BETA,DELT,FSTART,FC
 2016 FORMAT[1H ,6HAMAX =,E16.8,1H,4X6HBETA =,E16.8,1H,4X6HDELT =,E16.8,
     11H,4X8HFSTART =,E16.8,1H,4X4HFC =,E16.8)
      IF[IFOPT.EQ.0) GO TO 13
      IF[IFOPT.EQ.2) GO TO 9
      IF[IFTYP1.LT.3) GO TO 8
      READ[5,1005) [F1[I),I=1,NF1)
 1005 FORMAT[8F10.0)
      READ[5,1005) [TMAG1[K),K=1,NF1)
      READ[5,1005) [TPH1[L),L=1,NF1)
      WRITE[6,2011)[F1[I),TMAG1[I),TPH1[I),I=1,NF1)
 2011 FORMAT[1H ,43X2HF111X5HTMAG111X4HTPH1/[35X3E15.8))
      GO TO 9
    8 CONTINUE
      READ[5,1006) NORD1,FCUT1,RIP1
      WRITE[6,2010) NORD1,FCUT1,RIP1
 2010 FORMAT[1H ,47X5HNORD15X5HFCUT111X4HRIP1/50XI1,2E16.8)
    9 CONTINUE
   11 CONTINUE
      IF[IFTYP2.LT.3) GO TO 12
      IF[IFOPT.EQ.1) GO TO 13
      READ[5,1005) [F2[I),I=1,NF2)
      READ[5,1005) [TMAG2[K),K=1,NF2)
      READ[5,1005) [TPH2[L),L=1,NF2)
      WRITE[6,2012)[F2[I),TMAG2[I),TPH2[I),I=1,NF2)
 2012 FORMAT[1H ,43X2HF211X5HTMAG211X4HTPH2/[35X3E15.8))
      GO TO 13
   12 CONTINUE
      READ[5,1006) NORD2,FCEN2,FCUT2,RIP2
 1006 FORMAT[I5,3F10.0)
      WRITE[6,2015) NORD2,FCEN2,FCUT2,RIP2
 2015 FORMAT[1H ,35X5HNORD2,6X5HFCEN211X5HFCUT212X4HRIP2/36XI5,3E16.8)
   13 CONTINUE
      IF[IFTYP1.EQ.2) GO TO [14,16,14,16,14,16),NORD1
   14 CONTINUE
      IF[IFTYP2.EQ.2) GO TO [17,16,17,16,17,16),NORD2
      GO TO 17
```

SAP PROGRAM LISTING

```
   16 CONTINUE
      WRITE(6,1998)
      WRITE(6,1999)
      WRITE(6,2045)
 2045 FORMAT(1H ,45X40HEVEN ORDER CHEBYCHEV FILTER NOT ALLOWED.)
      WRITE(6,1999)
      STOP
   17 CONTINUE
      WRITE(6,1999)
      IF(ISOPT.EQ.0) GO TO 20
      GO TO (5,10,15),ISOPT
    5 CONTINUE
      READ(5,1010) M
 1010 FORMAT(I2)
      READ(5,1015) (A(I),I=1,M)
      READ(5,1015) (F(I),I=1,M)
      READ(5,1015) (THETA(I),I=1,M)
 1015 FORMAT(8F10.0)
      WRITE(6,2001) (A(L),F(L),THETA(L),L,L=1,M)
 2001 FORMAT(1H ,48X4HA(M)12X4HF(M)11X8HTHETA(M)3X1HM/(42X3E16.8,I3))
      WRITE(6,1999)
      GO TO 20
   10 CONTINUE
      READ(5,1020) (TP(I),I=1,5)
      READ(5,1020) (A(I),I=1,4)
 1020 FORMAT(8F10.0)
      WRITE(6,2002) (TP(L),A(L),L=1,4),TP(5)
 2002 FORMAT(1H ,63X2HTP15X1HA/(56X2E16.8))
      WRITE(6,1999)
      GO TO 20
   15 CONTINUE
      READ(5,1025) TP(1),A(1)
 1025 FORMAT(2F10.0)
      WRITE(6,2002) TP(1),A(1)
      TP(1) = TP(1)/2.0
      WRITE(6,1999)
   20 CONTINUE
      GO TO (100,200,300,400,500,600,700,800,900,1000),IEP
  100 CONTINUE
      LOCAT = 100
      WRITE(6,2020) LOCAT,IDT,IST,NRIDT,NCIDT
 2020 FORMAT(1H0,41X50H**************************************************
     1*/42X2H**46X2H**/42X2H**5X30HPROGRAM HAS PASSED ENTRY POINT,I5,1H.
     25X2H**/42X2H**5X26H DATA IS CURRENTLY ON UNIT,I2,1H.9X2H**/42X2H**
     35X    32H SPARE TAPE IS CURRENTLY ON UNIT,I2,1H.6X2H**/42X3H** 10H
     4 THERE ARE,I4,1X8HROWS AND,I4,1X19HCOLUMNS OF DATA. **)
       CALL CLOCK(MTIME)
       ITIMEX = MTIME
       CALL TIMER(ITIME1,MTIME,MIN,SECS)
       WRITE(6,1403) MIN,SECS
 1403 FORMAT(1H ,41X19H** EXECUTION TIME =,I3,1X8HMINUTES,2XF6.3,1X7HSEC
     1ONDS,3H **/42X2H**46X2H**/42X50H*********************************
     2*****************)
       IF(ISOPT.EQ.0) GO TO 150
       CALL TSGEN(IDT,NRIDT,NCIDT,DELT,ISOPT,IFM,A,F,THETA,M,TP)
       IF(IZPRT(1).EQ.1) CALL TWRITE(IDT,NCIDT,NRIDT)
       GO TO 200
  150  CONTINUE
       CALL ISAR(IDT,IST,NRIDT,NCIDT,DELT,NR1,NC1,KSR)
       X = FLOAT(KSR)
       DELT = DELT/X
       ITEMP = IDT
       IDT = IST
       IST = ITEMP
       IF(IZPRT(17).EQ.1) CALL TWRITE(IDT,NCIDT,NRIDT)
  200  CONTINUE
       LOCAT = 200
       WRITE(6,2020) LOCAT,IDT,IST,NRIDT,NCIDT
       CALL CLOCK(MTIME)
       CALL TIMER(ITIMEX,MTIME,MIN,SECS)
       ITIMEX = MTIME
       WRITE(6,1403) MIN,SECS
C*** DETERMINE MAXIMUM SIGNAL COMPONENT
       IUNT1=IDT
       IUNT2=IST
       NR=NRIDT
       NC=NCIDT
       SMAX=0.0
       DO 2 J=1,NC
       CALL MAGTAPE(I1,NR,AR,IUNT1)
      CALL MAGTAPE(11,NR,AI,IUNT1)
      DO 1 I=1,NR
      X=ABS(AR(I))
    1 IF(X.GT.SMAX) SMAX=X
    2 CONTINUE
      CALL MAGTAPE(14,NR,AR,IUNT1)
C*** PERFORM NORMALIZATION
      SMAX=1.0/SMAX
      DO 4 J=1,NC
      CALL MAGTAPE(11,NR,AR,IUNT1)
      CALL MAGTAPE(11,NR,AI,IUNT1)
      DO 95 I=1,NR
      AR(I)=SMAX*AR(I)
   95 CONTINUE
      CALL MAGTAPE(12,NR,AR,IUNT2)
      CALL MAGTAPE(12,NR,AI,IUNT2)
    4 CONTINUE
      CALL MAGTAPE(14,NR,AR,IUNT1)
      CALL MAGTAPE(14,NR,AR,IUNT2)
      ITEMP=IDT
      IDT=IST
      IST=ITEMP
      CALL TTRANS(NRIDT,NCIDT,IDT,IST)
      ITEMP = IDT
      IDT = IST
      IST = ITEMP
      ITEMP = NCIDT
      NCIDT = NRIDT
      NRIDT = ITEMP
  300 CONTINUE
      LOCAT = 300
      WRITE(6,2020) LOCAT,IDT,IST,NRIDT,NCIDT
      CALL CLOCK(MTIME)
      CALL TIMER(ITIMEX,MTIME,MIN,SECS)
      ITIMEX = MTIME
      WRITE(6,1403) MIN,SECS
      IF((IFOPT.EQ.1).OR.(IFOPT.EQ.3)) GO TO 310
      IF(IFM.NE.1) GO TO 600
  310 CONTINUE
      CALL EFFT(AR,AI,WR,WI,IDT,IST,NRIDT,NCIDT,I1)
      ITEMP = IDT
      IDT = IST
      IST = ITEMP
      ITEMP = NCIDT
      NCIDT = NRIDT
      NRIDT = ITEMP
      IF(IZPRT(2).EQ.1) CALL TWRITE(IDT,NCIDT,NRIDT)
  400 CONTINUE
      LOCAT = 400
      WRITE(6,2020) LOCAT,IDT,IST,NRIDT,NCIDT
      CALL CLOCK(MTIME)
      CALL TIMER(ITIMEX,MTIME,MIN,SECS)
      ITIMEX = MTIME
      WRITE(6,1403) MIN,SECS
C *** INITIALIZE DATA FOR FILTER
      DFREQ = 1.0/(DELT*FLOAT(NRIDT*NCIDT))
      IF((IFOPT.EQ.1).OR.(IFOPT.EQ.3)) GO TO 408
      IFIL = 4
      GO TO 416
  408 CONTINUE
      IF(IFM.EQ.1) GO TO 412
      IFIL = IFTYP1
      GO TO 416
  412 CONTINUE
      IFIL = IFTYP1 + 4
  416 CONTINUE
      FCEN1 = 0.0
      X9 = -1.0
      CALL FILTER(IDT,IST,NCIDT,NRIDT,DFREQ,X9,IFIL,TMAG1,TPH1,F1,NF1
     *,NORD1,FCEN1,FCUT1,RIP1)
      ITEMP = IDT
      IDT = IST
      IST = ITEMP
      IF(IZPRT(3).EQ.1) CALL TWRITE(IDT,NCIDT,NRIDT)
  500 CONTINUE
      LOCAT = 500
      WRITE(6,2020) LOCAT,IDT,IST,NRIDT,NCIDT
      CALL CLOCK(MTIME)
      CALL TIMER(ITIMEX,MTIME,MIN,SECS)
      ITIMEX = MTIME
      WRITE(6,1403) MIN,SECS
      MODE = 2
      CALL EFFT(AR,AI,WR,WI,IDT,IST,NRIDT,NCIDT,MODE)
      ITEMP = IDT
      IDT = IST
      IST = ITEMP
      ITEMP = NCIDT
      NCIDT = NRIDT
      NRIDT = ITEMP
      IF(IZPRT(4).EQ.1) CALL TWRITE(IDT,NCIDT,NRIDT)
  600 CONTINUE
      LOCAT = 600
      WRITE(6,2020) LOCAT,IDT,IST,NRIDT,NCIDT
      CALL CLOCK(MTIME)
      CALL TIMER(ITIMEX,MTIME,MIN,SECS)
      ITIMEX = MTIME
      WRITE(6,1403) MIN,SECS
      CALL MOD(IDT,IST,NRIDT,NCIDT,BETA)
      ITEMP=IDT
      IDT=IST
      IST=ITEMP
  700 CONTINUE
      LOCAT = 700
      WRITE(6,2020) LOCAT,IDT,IST,NRIDT,NCIDT
      CALL CLOCK(MTIME)
      CALL TIMER(ITIMEX,MTIME,MIN,SECS)
      ITIMEX = MTIME
      WRITE(6,1403) MIN,SECS
      MODE = 1
      CALL EFFT(AR,AI,WR,WI,IDT,IST,NRIDT,NCIDT,MODE)
      ITEMP = IDT
      IDT = IST
      IST = ITEMP
      ITEMP = NCIDT
      NCIDT = NRIDT
      NRIDT = ITEMP
      IF(IZPRT(5).EQ.1) CALL TWRITE(IDT,NCIDT,NRIDT)
  800 CONTINUE
      LOCAT = 800
      WRITE(6,2020) LOCAT,IDT,IST,NRIDT,NCIDT
      CALL CLOCK(MTIME)
      CALL TIMER(ITIMEX,MTIME,MIN,SECS)
      ITIMEX = MTIME
      WRITE(6,1403) MIN,SECS
      IF(IFOPT.LT.2) GO TO 900
      DFREQ = 1./(FLOAT(NRIDT*NCIDT)*DELT)
      IFIL = IFTYP2
      CALL FILTER(IDT,IST,NCIDT,NRIDT,DFREQ,FC,IFIL,TMAG2,TPH2,F2,NF2
     *,NORD2,FCEN2,FCUT2,RIP2)
      ITEMP = IDT
      IDT = IST
      IST = ITEMP
      IF(IZPRT(6).EQ.1) CALL TWRITE(IDT,NCIDT,NRIDT)
  900 CONTINUE
      LOCAT = 900
      WRITE(6,2020) LOCAT,IDT,IST,NRIDT,NCIDT
      CALL CLOCK(MTIME)
      CALL TIMER(ITIMEX,MTIME,MIN,SECS)
      ITIMEX = MTIME
      WRITE(6,1403) MIN,SECS
      CALL TTRANS(NRIDT,NCIDT,IDT,IST)
      ITEMP = IDT
      IDT = IST
      IST = ITEMP
      ITEMP = NCIDT
      NCIDT = NRIDT
      NRIDT = ITEMP
 1000 CONTINUE
      LOCAT = 1000
      WRITE(6,2020) LOCAT,IDT,IST,NRIDT,NCIDT
      CALL CLOCK(MTIME)
      CALL TIMER(ITIMEX,MTIME,MIN,SECS)
      ITIMEX = MTIME
      WRITE(6,1403) MIN,SECS
      DFREQ = 1./(DELT*FLOAT(NRIDT*NCIDT))
      X = (FSTRT+1.E-7)/DFREQ
      NSTART = X + 1
      DELF = DFREQ
      CALL PLOT(IDT,IST,NRIDT,NCIDT,IPMODE,        NAVE,FC,DELF,FSTRT
     1,AMAX,IPLPOS)
```

```
      WRITE(6,1999)
      WRITE(6,2025) AMAX
 2025 FORMAT(1H ,6HAMAX =,E15.8)
      WRITE(6,1999)
      IF(IPMODE.EQ.2) GO TO 1050
      WRITE(6,2035)
 2035 FORMAT(1H ,40X50HRUN TERMINATED. PLOT DATA TAPE GENERATED ON UNIT
     12)
      GO TO 1100
 1050 CONTINUE
      WRITE(6,2040)
 2040 FORMAT(1H ,43X44HRUN TERMINATED. NO PLOT DATA TAPE GENERATED.)
 1100 CONTINUE
      WRITE(6,1999)
      CALL CLOCK(ITIME2)
      CALL TIMER(ITIME1,ITIME2,MIN,SECS)
      WRITE(6,1403) MIN,SECS
      STOP
      END
```

SAP PROGRAM LISTING

```
      SUBROUTINE PLOT(IUNT1,IUNT2,NR,NC,IPMODE,        NAVE,FC,DELF,FSTRT
     1,AMAX,IPLPOS)
CD* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
CD
CD    PROGRAMMER AND DATE
CD            RICHARD C. THOMAS
CD            TRW SYSTEMS
CD            JUNE 1972
CD
CD    PURPOSE
CD            GENERATES THE DATA TAPE REQUIRED TO DRIVE THE EAI PLOTTER
CD            ALSO PRINTS THE DATA TO BE PLOTTED.
CD
CD    USAGE
CD            CALL PLOT(IUNT1,IUNT2,NR,NC,IPMODE,NAVE,FC,DELF,FSTRT,
CD                     AMAX,IPLPOS)
CD
CD    DESCRIPTION OF PARAMETERS
CD
CD       INPUT
CD         CALLING SEQUENCE
CD            IUNT1   - INPUT TAPE UNIT
CD            IUNT2   - SPARE TAPE UNIT
CD            NR      - NUMBER OF ROWS IN INPUT ARRAYS
CD            NC      - NUMBER OF COLUMNS IN INPUT ARRAYS
CD            NAVE    - NUMBER OF INPUT POINTS TO BE AVERAGED IN OUTPUT
CD            FC      - CENTER FREQUENCY OF PLOT
CD            DELF    - INCREMENT BETWEEN INPUT POINTS(FREQUENCY)
CD            FSTRT   - STARTING FREQUENCY OF PLOT
CD            IPMODE  - .EQ.0 PRINT POWER SPECTRUM AND GENERATE DATA
CD                      .EQ.2 ONLY PRINT POWER SPECTRUM
CD                      .EQ.1 ONLY GENERATE DATA TAPE
CD            AMAX    - MAXIMUM POWER COMPONENT OF INPUT SIGNAL
CD         COMMON
CD            ITITLE  - UP TO 30 CHARACTERS FOR RUN TITLE AND/OR DATE
CD         CARD
CD         TAPE
CD            AR      - REAL PART OF SIGNAL
CD            AI      - IMAGINARY PART OF SIGNAL
CD
```

SAP PROGRAM LISTING

```
CD        OUTPUT
CD          CALL SEQUENCE
CD          COMMON
CD          PRINT
CD              P       - POWER SPECTRUM ARRAY, DEPENDENT PLOT VARIABLE
CD              F       - FREQUENCY ARRAY, INDEPENDENT PLOT VARIABLE
CD          TAPE
CD
CD     REMARKS AND RESTRICTIONS
CD     SUBROUTINES REQUIRED
CD              MAGTAP
CD
CD* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
      DIMENSION AR[256),AI[256),P[2048),F[2048)
      DIMENSION ITITLE[30)
      COMMON/STORGE/AR,AI,P,F
      COMMON/TITL/ITITLE
      I1 = 1
      I2 = 2
      I4 = 4
      I5 = 5
      I512 = 480
      CALL MAGTAP[I4,I4,AR,IUNT1)
      CALL MAGTAP[I4,I4,AR,IUNT2)
      IF[AMAX.GT.0.0) GO TO 5
      DO 4 J=1,NC
      CALL MAGTAP[I1,NR,AR,IUNT1)
      CALL MAGTAP[I1,NR,AI,IUNT1)
      DO 3 I=1,NR
      X = AR[I)**2 + AI[I)**2
      IF[X.GT.AMAX) AMAX = X
    3 CONTINUE
    4 CONTINUE
      CALL MAGTAP[I4,I4,AR,IUNT1)
    5 CONTINUE
      NPS = 480
      IF[IPMODE.EQ.2) NPS = NR*NC/2
      IF[FSTRT) 6,7,7
    6 CONTINUE
      IFC = 0
      GO TO 9
    7 CONTINUE
      IF[FSTRT.GE.FC) GO TO 8
      IFC = -1
      GO TO 9
    8 CONTINUE
      IFC = 1
    9 CONTINUE
      IF[IFC.NE.0) GO TO 20
      IPASS = 1
      NPS = NPS/2
      FMAX = DELF*FLOAT[NR*NC/2-1)
      FLAST = DELF*FLOAT[NAVE*[NPS-1))
      IF[FLAST.LE.FMAX) GO TO 30
      NAVE = [[NR*NC/2-1)/[NPS-1))
      WRITE[6,2000) NAVE
 2000 FORMAT[1H ,25HNAVE HAS BEEN ADJUSTED TO,I4)
      GO TO 30
   20 CONTINUE
      FLAST = FSTRT + DELF*FLOAT[NAVE*[NPS-1))
      IF[IFC.EQ.1) GO TO 22
      FMAX = FC
      GO TO 24
   22 CONTINUE
      FMAX = FC + DELF*FLOAT[NR*NC/2-1)
   24 CONTINUE
      IF[FLAST.LE.FMAX) GO TO 30
      NPS =[INT[[FMAX-FSTRT)/DELF)+1)/NAVE
      WRITE[6,2005) NPS
 2005 FORMAT[1H ,45HNUMBER OF POINTS PLOTTED HAS BEEN ADJUSTED TO,I4)
      IF[NPS.GE.480) GO TO 30
      IBASE = NPS + 1
      DO 26 I=IBASE,480
      P[I) = -70.0
   26 CONTINUE
   30 CONTINUE
      EPS = 1.E-7
      SUM = 0.
      NSUM = 0
      K = 1
```

SAP PROGRAM LISTING

```
      PMIN = -70.
      XAVE = NAVE
      IF(IFC) 36,32,34
   32 CONTINUE
      N = 0
      KT=1
      FSTRT = FC
      GO TO 38
   34 CONTINUE
      NSTART =INT((FSTRT-FC+EPS)/DELF) + 1
      N = NSTART/NR
      KT = NSTART - N*NR
      GO TO 38
   36 CONTINUE
      FDUM = FC - DELF*FLOAT(NR*NC/2)
      NSTART = INT((FSTRT-FDUM+EPS)/DELF) + 1
      N = NSTART/NR
      KT = NSTART - N*NR
      N = N + NR/2
   38 CONTINUE
   40 CONTINUE
      IF(N.EQ.0) GO TO 50
      DO 45 I=1,N
      CALL MAGTAP(I1,NR,AR,IUNT1)
      CALL MAGTAP(I1,NR,AI,IUNT1)
   45 CONTINUE
   50 CONTINUE
      CALL MAGTAP(I1,NR,AR,IUNT1)
      CALL MAGTAP(I1,NR,AI,IUNT1)
      DO 55 I=1,NR
      AR(I) = AR(I)**2 + AI(I)**2
   55 CONTINUE
   60 CONTINUE
      IF(NSUM.EQ.0) SUM = 0.
      SUM = SUM + AR(KT)
      KT = KT + 1
      NSUM = NSUM + 1
      IF(NSUM.LT.NAVE) GO TO 70
      KP = K
      IF((IFC.EQ.0).AND.(IPASS.EQ.1)) KP = KP + NPS
      ARG = SUM/[AMAX*XAVE)
      IF[ARG.LE.0.0) ARG = 1.0E-08
      P[KP) = 10.0*ALOG10[ARG)
      IF[P[KP).LT.PMIN) P[KP) = PMIN
      F[KP) = FSTRT + DELF*FLOAT[[K-1)*NAVE)
      K = K + 1
      NSUM = 0
      IF[K.GT.NPS) GO TO 80
   70 CONTINUE
      IF[KT.LE.NR) GO TO 60
      KT = 1
      GO TO 50
   80 CONTINUE
      IF[IFC.NE.0) GO TO 90
      IPASS = IPASS + 1
      IF[IPASS.GT.2) GO TO 90
      SUM = 0.0
      NSUM = 0
      K = 1
      NSTART = NR*NC + 1 - NPS*NAVE
      N = NSTART/NR
      KT = NSTART - N*NR
      FSTRT = FC - DELF*FLOAT[NPS*NAVE)
      CALL MAGTAP[I4,14,AR,IUNT1)
      GO TO 40
   90 CONTINUE
      IF[IPMODE.EQ.1) GO TO 100
      IF[IFC.EQ.0) NPS=480
      NPS1 = NPS/4
      NPS2 = 2*NPS1
      NPS3 = 3*NPS1
      WRITE[6,1999)
 1999 FORMAT[1H ,132H***************************************************
     1******************************************************************
     2***************)
      WRITE[6,2010) [F[K),P[K),F[K+NPS1).P[K+NPS1),F[K+NPS2),P[K+NPS2),
     1 F[K+NPS3),P[K+NPS3),K=1,NPS1)
 2010 FORMAT[1H1,5X4HF[K)12X4HP[K)13X4HF[K)12X4HP[K)13X4HF[K)12X4HP[K)13
     1X4HF[K)12X4HP[K)/[2E16.8,1X2E16.8,1X2E16.8,1X2E16.8))
      WRITE[6,1999)
      IF(IPMODE.EQ.2) GO TO 110
  100 CONTINUE
      F(1) = IPLPOS
      DO 105 I=1,30
      F(I+1) = ITITLE(I)
  105 CONTINUE
      F(32) = DELF
      F(33) = FC
      F(34) = FSTRT
      F(35) = NPS
      F(36) = IFC
      I36 = 36
      CALL MAGTAPE(I4,14,P,I2)
      CALL MAGTAPE(I2,I36,F,I2)
      CALL MAGTAPE(I2,1512,P,I2)
      CALL MAGTAPE(I5,14,P,I2)
      CALL MAGTAPE(I4,14,P,I2)
  110 CONTINUE
      RETURN
      END
```

SAP PROGRAM LISTING

```
      SUBROUTINE BITREV(K,LXX,KP)
      L = LXX
CD* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
CD
CD    PROGRAMMER AND DATE
CD             DAVID M. DETCHMENDY
CD             TRW SYSTEMS
CD             MAY 1970
CD
CD    PURPOSE
CD             THIS SUBROUTINE SOLVES THE BIT REVERSAL PROBLEM, I.E.
CD             GIVEN K AND L WHERE
CD
CD                  K = I0 + I1*(2**1) + ... + IL*(2**L)
CD
CD             FIND KP WHERE
CD
CD                  KP = I0*(2**L) + I1*(2**(L-1)) + ... + IL
CD
CD             IN THE ABOVE I0, I1, ... , IL ARE ZERO OR ONE
CD
CD    USAGE
CD             CALL BITREV(K,L,KP)
CD
CD    DESCRIPTION OF PARAMETERS
CD
CD       INPUT
CD         CALLING SEQUENCE
CD             K - INTEGER
CD             L - DEFINES NUMBER OF BINARY DIGITS IN K
CD         COMMON
CD             NONE
CD         CARD
CD             NONE
CD         TAPE
CD             NONE
CD
CD       OUTPUT
CD         CALLING SEQUENCE
CD             KP - INTEGER RESULTING FROM THE BIT REVERSAL
```

SAP PROGRAM LISTING

```
CD          COMMON
CD              NONE
CD          CARD
CD              NONE
CD          PRINT
CD              NONE
CD          TAPE
CD              NONE
CD
CD     REMARKS AND RESTRICTIONS
CD              NONE
CD
CD     SUBROUTINES REQUIRED
CD              NONE
CD
CD     METHOD
CD              TESTS EACH BIT IN K BY SUBTRACTION, THEN CHECKS THE SIGN
CD              OF THE RESULT TO GENERATE A 1 OR A 0 IN THE BIT REVERSED
CD              BINARY DIGIT OF KP
CD
CD* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
      ID = 2
      KP = 0
      KO = K
      J1 = 1
      J2 = 2**L
      LP = L+1
      DO 10 I=1,LP
      KN = KO - J2
      IF( KN .LT. 0 ) GO TO 5
      KP = KP + J1
      KO = KN
    5 J1 = 2*J1
   10 J2 = J2/2
      RETURN
      END
      SUBROUTINE CLOCK[J)
CD* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
CD
CD    PROGRAMMER AND DATE
CD             RICHARD C. THOMAS
CD             TRW SYSTEMS
CD             JUNE 1972
CD
CD    PURPOSE
CD             CLOCK IS A DUMMY ROUTINE THAT ALLOCATES CORE STORAGE FOR
CD             A CS-1 ASSEMBLY LANGUAGE ROUTINE NAMED CLOCK. THE CS-1
CD             PROGRAM REFERENCES THE COMPUTER CLOCK. THE COMPUTER CLOCK
CD             GIVES AN ABSOLUTE TIME WHICH HAS NO MEANINGFUL REFERENCE
CD             TIME BASE. TWO SEPERATE CALLS TO THIS ROUTINE MAY BE USED
CD             WITH SUBROUTINE TIMER TO COMPUTE THE RUN TIME OF A SEG-
CD             MENT OF THE PROGRAM.
CD
CD    USAGE
CD             CALL CLOCK[I)
CD
CD    DESCRIPTION OF PARAMETERS
CD
CD       INPUT
CD    NONE
CD
CD       OUTPUT
CD         CALL SEQUENCE
CD             I      - TIME FROM COMPUTER CLOCK.
CD
CD    REMARKS AND RESTRICTIONS
CD             THIS ROUTINE ALLOCATES THE NECESSARY CORE REQUIRED FOR
CD             THE CS-1 PROGRAM WHICH MUST BE LOADED BY PAPER TAPE. SEE
CD             PROGRAM OPERATING INSTRUCTIONS FOR CORRECT PROCEDURES.
CD    SUBROUTINES REQUIRED
CD    NONE
CD
CD* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
      DIMENSION A[300)
      A[1)=1.0
      RETURN
```

SAP PROGRAM LISTING

END

SAP PROGRAM LISTING

```
      SUBROUTINE EFFT[AR,AI,WR,WI,II,IO,NROW,NCOL,MODE)
CD* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
CD    PROGRAMMER AND DATE
CD        RICHARD C. THOMAS
CD        TRW SYSTEMS
CD        DECEMBER 1971
CD
CD    PURPOSE
CD        COMPUTES THE DISCRETE FOURIER TRANSFORM USING THE COOLEY-
CD        TUKEY ALGORITHM
CD
CD    USAGE
CD        CALL EFFT[AR,AI,WR,WI,II,IO,NROW,NROW,NCOL,MODE,NBUF)
CD
CD    DESCRIPTION OF PARAMETERS
CD
CD     INPUT
CD        AR - REAL PORTION OF FUNCTION TO BE TRANSFORMED.
CD        AI - IMAGINARY PORTION OF FUNCTION TO BE TRANSFORMED.
CD        WR - REAL PORTION OF EXPONENTIAL TERMS.
CD        WI - IMAGINARY PORTION OF EXPONENTIAL TERMS.
CD        II - INPUT TAPE UNIT.
CD        IO - OUTPUT TAPE UNIT.
CD        NROW - NUMBER OF ROWS IN INPUT ARRAY.
CD        NCOL - NUMBER OF COLUMNS IN INPUT ARRAY.
CD        MODE - .EQ.1 COMPUTE DIRECT FFT.
CD                .NE.1 COMPUTE INVERSE FFT.
CD
CD     OUTPUT
CD        AR - REAL PORTION OF TRANSFORM.
CD        AI - IMAGINARY PORTION OF TRANSFORM.
CD        WR - REAL PORTION OF EXPONENTIAL TERMS.
CL        WI - IMAGINARY PORTION OF EXPONENTIAL TERMS.
CD
CD    SUBROUTINES REQUIRED
CD        BITREV
CD        FFT
CD* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
      COMMON/STORGE/AR,AI,WR,WI
      DIMENSION AR[256),AI[256),WR[2048),WI[2048)
      I1 = 1
      I2 = 2
      I4 = 4
      NR = NROW
      NC = NCOL
      CON = 6.2831854/[NR*NC)
      CALL MAGTAP[I4,I4,AR,II)
      CALL MAGTAP[I4,I4,AR,IO)
      J = NR
      N = 1
   10 CONTINUE
      IF[J.EQ.4) GO TO 20
      N = N + 1
      J = J/2
      GO TO 10
   20 CONTINUE
      IF[MODE.EQ.1) CON = - CON
      DO 40 J=1,NC
      CALL MAGTAP[I1,NR,AR,II)
      CALL MAGTAP[I1,NR,AI,II)
      CALL FFT[N,MODE,J)
      DO 30 I=1,NR
      X = CON*FLOAT[[I-1)*[J-1))
      TWR = COS[X)
      TWI = SIN[X)
      ARP = AR[I)
      AIP = AI[I)
      AR[I) = ARP*TWR-AIP*TWI
      AI[I) = ARP*TWI+AIP*TWR
   30 CONTINUE
      CALL MAGTAP[I2,NR,AR,IO)
      CALL MAGTAP[I2,NR,AI,IO)
   40 CONTINUE
      CALL MAGTAP[I4,I4,AR,II)
      CALL MAGTAP[I4,I4,AR,IO)
      CALL TTRANS[NR,NC,IO,II)
      L = NC
      N = 1
   50 CONTINUE
      IF[L.EQ.4) GO TO 60
      N = N + 1
      L = L/2
      GO TO 50
   60 CONTINUE
      DO 70 J=1,NR
      CALL MAGTAPE(I1,NC,AR,II)
      CALL MAGTAPE(I1,NC,AI,II)
      CALL FFT(N,MODE,J)
      CALL MAGTAPE(I2,NC,AR,IO)
      CALL MAGTAPE(I2,NC,AI,IO)
   70 CONTINUE
      CALL MAGTAPE(I4,I4,AR,II)
      CALL MAGTAPE(I4,I4,AR,IO)
      RETURN
      END
```

SAP PROGRAM LISTING

```
      SUBROUTINE FFT(M,P,MODE,NCWP)
CD* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
CD
CD    PROGRAMMER AND DATE
CD            DAVID M. DETCHMENDY
CD            TRW SYSTEMS
CD            MAY 1970
CD
CD    PURPOSE
CD            COMPUTES THE DISCRETE FOURIER TRANSFORM USING THE COOLEY-
CD            TUKEY ALGORITHM
CD
CD    USAGE
CD            CALL FFT(AR,AI,WR,WI,M,MODE,NCW)
CD
CD    DESCRIPTION OF PARAMETERS
CD
CD       INPUT
CD         CALLING SEQUENCE
CD            AR - REAL PORTION OF FUNCTION TO BE TRANSFORMED
CD            AI - IMAGINARY PORTION OF FUNCTION TO BE TRANSFORMED
CD            WR - REAL PORTION OF EXPONENTIAL TERMS
CD            WI - IMAGINARY PORTION OF EXPONENTIAL TERMS
CD            M  - DEFINES DIMENSION OF AR, AI, WR, AND WI ALL ARE OF
CD                 DIMENSION 2**(M+1)
CD            MODE - .EQ.1 COMPUTE DIRECT TRANSFORM
CD                   .NE.1 COMPUTE INVERSE TRANSFORM
CD            NCW  - .EQ.1 COMPUTE WR AND WI ON ENTRY TO FFT
CD                   .NE.1 WR AND WI ARE INPUT, USUALLY COMPUTED ON A
CD                   PREVIOUS ENTRY
CD         COMMON
CD            NONE
CD         CARD
CD            NONE
CD         TAPE
CD            NONE
CD
CD       OUTPUT
CD         CALLING SEQUENCE
CD            AR - REAL PORTION OF TRANSFORM
CD             AI - IMAGINARY PORTION OF TRANSFORM
CD             WR - REAL PORTION OF EXPONENTIAL TERMS
CD             WI - IMAGINARY PORTION OF EXPONENTIAL TERMS
CD          COMMON
CD             NONE
CD          CARD
CD             NONE
CD          PRINT
CD             NONE
CD          TAPE
CD             NONE
CD
CD     REMARKS AND RESTRICTIONS
CD             1. OUTPUT AR AND AI ARE OVER WRITTEN ON INPUT AR AND AI
CD             2. THE OUTPUT WR AND WI ARE STORED IN BIT REVERSED ORDER
CD             3. AR AND AI ARE IN NATURAL ORDER
CD             4. AR,AI,WR, AND WI ARE EXTERNALLY DIMENSIONED
CD
CD     SUBROUTINES REQUIRED
CD             BITREV
CD
CD     METHOD
CD             REFERENCE - D.M.DETCHMENDY, ENGINEERING DESCRIPTION OF A
CD                         FAST FOURIER TRANSFORM ALGORITHM, TRW IOC
CD                         5522.7-70-113, JULY 1970.
CD
CD* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
       DIMENSION AR(256),AI(256),WR(2048),WI(2048)
       COMMON/STORGE/AR,AI,WR,WI
       M = MP
       NCW = NCWP
       N = 2**(M+1)
       PI = 3.1415927
       ITWOM = 2**M
       CON = PI/ITWOM
       MM1 = M-1
C
C              GENERATE COSINES AND SINES IN BIT REVERSED ORDER, IF
C              NCW = 1
C
```

```
      IF( NEW .NE. 1 ) GO TO 12
      DO 10 I=1,ITWOM
      IM1 = I-1
      CALL BITREV(IM1,MM1,KP)
      LP = KP
      X = CON*FLOAT(LP)
      WR(I) = COS(X)
   10 WI(I) = SIN(X)
   12 CONTINUE
C
C              SCALE AND FORM CONJUGATE, IF MODE = 1
C
      IF( MODE .NE. 1 ) GO TO 22
      DFN = 1.0/FLOAT(N)
      DO 20 I=1,N
      AR(I) = DFN*AR(I)
   20 AI(I) = -DFN*AI(I)
   22 CONTINUE
C
C              BEGIN FFT ALGORITHM
C
C                   THE INDICES KEEP TRACK OF
C
C                        I1 - THE A BEING CALCULATED IS A SUB I1
C                        I2 - THE GROUP OF A SUB I1 BEING CALCULATED
C                             IS I2
C                        I3 - THE INDEX ON THE A PAIR BEING CALCULATED
C
      MP1 = M+1
      DO 100 I1 = 1,MP1
      I1P = 2**(I1-1)
      I2P = 2**(MP1-I1)
      IEXTP = 2*I2P
      DO 60 I2 = 1,I1P
      IEXT = (I2-1)*IEXTP
      W3R = WR(I2)
      W3I = WI(I2)
      DO 60 I3=1,I2P
      IN0 = I3 + IEXT
      IN1 = IN0 + I2P
      A00R = AR[IN0)
      A00I = AI[IN0)
      A01R = AR[IN1)
      A01I = AI[IN1)
      CFR = W3R*A01R - W3I*A01I
      CFI = W3I*A01R + W3R*A01I
      AR[IN0) = A00R + CFR
      AI[IN0) = A00I + CFI
      AR[IN1) = A00R - CFR
   60 AI[IN1) = A00I - CFI
   80 CONTINUE
  100 CONTINUE
C
C               INTERCHANGE THE A-S USING BIT REVERSAL
C
      MM = M
      DO 120 K = 1,N
      KM1 = K-1
      CALL BITREV[KM1,MM,KP)
      IF [KP - KM1) 120,120,114
  114 A00R = AR[K)
      A00I = AI[K)
      A01R = AR[KP+1)
      A01I = AI[KP+1)
      AR[K) = A01R
      AI[K) = A01I
      AR[KP+1) = A00R
      AI[KP+1) = A00I
  120 CONTINUE
C
C               FORM CONJUGATE, IF MODE =1
C
      IF[ MODE .NE. 1 ) GO TO 124
      DO 122 I=1,N
  122 AI[I) = - AI[I)
  124 CONTINUE
      RETURN
      END
```

SAP PROGRAM LISTING

```
      SUBROUTINE FILTER(ITAP1,ITAP2,NC,NR,DFREQ,FC,IFIL,TMAG,TPH,F,NF,
     1NORDER,FCEN,FCUT,RIP)
CD* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
CD
CD    PROGRAMMER AND DATE
CD             RICHARD C. THOMAS
CD             TRW SYSTEMS
CD             JUNE 1972
CD
CD    PURPOSE
CD             PERFORMS THE FILTERING OPERATION BY MULTIPLICATION OF THE
CD             TRANSFORM OF THE SIGNAL BY THE FILTER TRANSFER FUNCTION.
CD             THE FILTER TRANSFER FUNCTIONS CAN BE INPUT BY THE USER OR
CD             IT CAN BE SPECIFIED AS ONE OF THE BUILT-IN BUTTERWORTH OR
CD             CHEBYCHEV FILTERS. THIS ROUTINE IS ALSO USED TO 9NT57R1T5
CD             THE SIGNAL, BY THE USE OF THE TRANSFER FUNCTION 1/(JW),
CD             IN THE CASE OF FM SIGNALS.
CD
CD    USAGE
CD             CALL FILTER(ITAP1,ITAP2,NC,NR,DFREQ,FC,IFIL,TMAG,TPH,F,
CD                         NF,NORDER,FCEN,FCUT,RIP)
CD
CD    DESCRIPTION OF PARAMETERS
CD
CD       INPUT
CD         CALLING SEQUENCE
CD             ITAP1  - INPUT TAPE UNIT
CD             ITAP2  - OUTPUT TAPE UNIT
CD             NC     - NUMBER OF COLUMNS IN ARRAYS
CD             NR     - NUMBER OF ROWS IN ARRAYS
CD             DFREQ  - FREQUENCY INCREMENT
CD             FC     - CENTER FREQUENCY
CD             IFIL   - .EQ.1 BUTTERWORTH FILTER
CD                    - .EQ.2 CHEBYCHEV
CD                    - .EQ.3 INPUT DATA FILTER
CD                    - .EQ.4 INTEGRATION FILTER
CD                    - .EQ.5 PRODUCT OF 1 AND 4
CD                    - .EQ.6 PRODUCT OF 2 AND 4
CD                    - .EQ.7 PRODUCT OF 3 AND 4
CD             TMAG   - MAGNITUDE OF INPUT FILTER TRANSFER FUNCTION
CD              TPH     - CORRESPONDING PHASE[IN RADIANS)
CD              F       - FREQUENCY[IN HERTZ) AT WHICH MAGNITUDE AND PHASE
CD              NF      - DIMENSION OF TMAG,TPH AND F
CD              NORD    - ORDER OF THE BUTTERWORTH OR CHEBYCHEV FILTER
CD              FCEN    - CENTER FREQUENCY FOR BUTTERWORTH OR CHEBYCHEV
CD              FCUT    - CUTOFF FREQUENCY FOR BUTTERWORTH OR CHEBYCHEV
CD              RIP     - RIPPLE FOR BUTTERWORTH FILTER
CD          COMMON
CD              AR      - REAL PART OF FUNCTION BEING FILTERED
CD              AI      - IMAGINARY PART OF FUNCTION BEING FILTERED
CD              WR      - DUMMY COMMON VARIABLE
CD              WI      - DUMMY COMMON VARIABLE
CD          CARD
CD          TAPE
CD
CD        OUTPUT
CD          CALL SEQUENCE
CD          COMMON
CD          CARD
CD          PRINT
CD          TAPE
CD
CD     SUBROUTINES REQUIRED
CD              MAGTAP
CD              TRFN
CD     NONE
CD
CD* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
      DIMENSION F[1),TMAG[1),TPH[1)
      DIMENSION AR[256),AI[256),WR[2048),WI[2048)
      COMMON/STORGE/AR,AI,WR,WI
      I1=1
      I2=2
      I4=4
      IFC=0
      IF[FC.GT.0.01) IFC=1
      CALL MAGTAP[I4,NR,AR,ITAP1)
      CALL MAGTAP[I4,NR,AR,ITAP2)
C**** SET IFLAG AND IOPT
      IF[IFIL.GT.3) GO TO 1
      IFLAG=1
      IOPT=IFIL
      GO TO 2
    1 IFLAG=3
      IOPT=IFIL-4
      IF[IFIL.EQ.4)IFLAG=2
    2 CONTINUE
      IF[[IFIL.EQ.3).OR.[IFIL.EQ.7)) GO TO 4
      GO TO 10
    4 DELF=FLOAT[[NC*NR/2)-1)*DFREQ
      IF[FC.LT.0.01) GO TO 6
      FMIN=FC-DELF
      FMAX=FC+DELF
      IFC=1
      GO TO 7
    6 FMIN=0.0
      FMAX=DELF
      IFC=0
    7 IF[[FMIN.LT.F[1)).OR.[FMAX.GT.F[NF))) GO TO 997
   10 J=1
    3 CONTINUE
      CALL MAGTAP[I1,NR,AR,ITAP1)
      CALL MAGTAP[I1,NR,AI,ITAP1)
      NS=1
      IF[IFLAG.EQ.2) GO TO 400
C***  FILTER AR AND AI FOR FREQUENCIES ABOVE FC
      IUP=NR/2
      DO 101 I=1,IUP
      FREQ=FLOAT[[J-1)+[I-1)*NC)*DFREQ + FC
      CALL TRFN[FREQ,IOPT,NORDER,FCEN,FCUT,RIP,TMAG,F,NF,NS,TR,TI,TPH)
      ARP=AR[I)
      AIP=AI[I)
      AR[I)=ARP*TR - AIP*TI
      AI[I)=ARP*TI + AIP*TR
  101 CONTINUE
      NS=1
      IF[IFC.NE.0) GO TO 300
C***  FILTER AR AND AI FOR NEGATIVE FREQUENCIES [MODULATING SIGNAL)
      IS= NR/2 + 1
      DO 201 I=IS,NR
      IP= NR + [[NR/2)+1)-I
      FREQ=FLOAT[NR*NC-[J-1)-[IP-1)*NC)*DFREQ
      CALL TRFN[FREQ,IOPT,NORDER,FCEN,FCUT,RIP,TMAG,F,NF,NS,TR,TI,TPH)
      ARP=AR[IP)
      AIP=AI[IP)
      TI=-TI
      AR[IP)=ARP*TR -AIP*TI
      AI[IP)=ARP*TI +AIP*TR
  201 CONTINUE
      IF[IFLAG.EQ.3) GO TO 400
      GO TO 500
C***  FILTER AR AND AI FOR FREQUENCIES BELOW FC [MODULATED SIGNAL)
  300 CONTINUE
      IS= NR/2 + 1
      DO 301 I=IS,NR
      IP=NR + NR/2 + 1 - I
      FREQ = FC-FLOAT[NR*NC-[J-1)-[IP-1)*NC)*DFREQ
      CALL TRFN[FREQ,IOPT,NORDER,FCEN,FCUT,RIP,TMAG,F,NF,NS,TR,TI,TPH)
      ARP=AR[IP)
      AIP=AI[IP)
      AR[IP)=ARP*TR - AIP*TI
  301 AI[IP)=ARP*TI + AIP*TR
      GO TO 500
C***  INTEGRATION FILTER - MODULATING SIGNAL ONLY
  400 CONTINUE
      IUP=NR/2
      DO 415 I=1,IUP
      FREQ = FLOAT[[J-1)+[I-1)*NC)*DFREQ
      IF[FREQ.LT.1.0E-30) GO TO 405
      TI=-1./FREQ
      GO TO 410
  405 CONTINUE
      TI = 0.0
  410 CONTINUE
      ARP=AR[I)
      AIP=AI[I)
      AR[I)=-AIP*TI
      AI[I) = ARP*TI
  415 CONTINUE
      IS= NR/2 + 1
      DO 430 I=IS,NR
      FREQ=FLOAT(NC*NR-(J-1)-(I-1)*NC)*DFREQ
      IF(FREQ.LT.1.0E-30) GO TO 420
      TI= 1./FREQ
      GO TO 425
  420 CONTINUE
      TI = 0.0
  425 CONTINUE
      ARP=AR(I)
      AIP=AI(I)
      AR(I)=-AIP*TI
      AI(I) = ARP*TI
  430 CONTINUE
C***  WRITE TAPE
      IF((IFC.NE.0).OR.(J.NE.1)) GO TO 501
      AR(I) = 0.0
      AI(I) = 0.0
  501 CONTINUE
  500 CONTINUE
      CALL MAGTAP(I2,NR,AR,ITAP2)
      CALL MAGTAP(I2,NR,AI,ITAP2)
      J=J+1
      IF(J.LE.NC) GO TO 3
      CALL MAGTAP(I4,NR,AR,ITAP1)
      CALL MAGTAP(I4,NR,AR,ITAP2)
      RETURN
  997 WRITE(6,998)
  998 FORMAT(2X,49H INPUT FILTER FREQUENCY SPREAD INVALID FOR SIGNAL)
      STOP
      END
```

SAP PROGRAM LISTING

```
      SUBROUTINE MOD(IUNT1,IUNT2,NR,NC,BETA)
CD* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
CD
CD    PROGRAMMER AND DATE
CD             RICHARD C. THOMAS
CD             TRW SYSTEMS
CD             JUNE 1972
CD
CD    PURPOSE
CD             PERFORMS THE EXPONENTIATION OPERATION USED TO GENERATE
CD             THE MODULATED SIGNAL REPRESENTATION FROM THE MODULATING
CD             SIGNAL
CD
CD    USAGE
CD             CALL MOD(IUNT1,IUNT2,NR,NC,BETA)
CD
CD    DESCRIPTION OF PARAMETERS
CD
CD       INPUT
CD         CALLING SEQUENCE
CD             IUNT1  - INPUT DATA TAPE UNIT
CD             IUNT2  - OUTPUT DATA TAPE UNIT
CD             NR     - NUMBER OF WORDS/RECORD
CD             NC     - NUMBER OF REAL RECORDS
CD             BETA   - MODULATION INDEX, MULTIPLIES ENTIRE TEST SIGNAL
CD         COMMON
CD             AR     - REAL PART OF MODULATED FUNCTION
CD             AI     - IMAGINARY PART OF MODULATED FUNCTION
CD             WR     - DUMMY COMMON ARGUMENT
CD             WI     - DUMMY COMMON ARGUMENT
CD
CD    SUBROUTINES REQUIRED
CD             MAGTAP
CD
CD* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
      DIMENSION AR(256),AI(256),WR(2048),WI(2048)
      COMMON/STORGE/AR,AI,WR,WI
      I1 = 1
      I2 = 2
      I4 = 4
      CALL MAGTAPE(I4,NR,AR,IUNT1)
      CALL MAGTAPE(I4,NR,AR,IUNT2)
C*** COMPUTE EXPONENTIAL FOR EACH COMPONENT
      DO 4 J=1,NC
      CALL MAGTAPE(I1,NR,AR,IUNT1)
      CALL MAGTAPE(I1,NR,AI,IUNT1)
      DO 3 I=1,NR
      X=BETA*AR(I)
      AR(I)=COS(X)
      AI(I)=SIN(X)
    3 CONTINUE
      CALL MAGTAPE(I2,NR,AR,IUNT2)
      CALL MAGTAPE(I2,NR,AI,IUNT2)
    4 CONTINUE
      CALL MAGTAPE(I4,NR,AR,IUNT1)
      CALL MAGTAPE(I4,NR,AR,IUNT2)
      RETURN
      END
```

SAP PROGRAM LISTING

```
      SUBROUTINE TSGEN(IUNT,MR,MC,DT,IOPT,IFM,A,F,THETA,KT,TP)
CD* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
CD
CD    PROGRAMMER AND DATE
CD             RICHARD C. THOMAS
CD             TRW SYSTEMS
CD             JUNE 1972
CD
CD    PURPOSE
CD             GENERATES THE BUILT IN TEST SIGNALS FOR FREQUENCY MODULA-
CD             TION(FM) OR PHASE MODULATION. THE TEST SIGNALS ARE SUM OF
CD             SINUSOIDS,PERIODIC FOUR LEVEL TEST SIGNAL, AND A SQUARE
CD             WAVE.
CD
CD    USAGE
CD             CALL TSGEN(IUNT,MR,MC,DT,IOPT,IFM,A;F,THETA,K,TP,BETA)
CD
CD    DESCRIPTION OF PARAMETERS
CD
CD       INPUT
CD         CALLING SEQUENCE
CD             IUNT    - TAPE UNIT FOR GENERATED OUTPUT SIGNAL
CD             MR      - NUMBER OF ROWS IN OUTPUT ARRAYS
CD             MC      - NUMBER OF COLUMNS IN OUTPUT ARRAYS
CD             DT      - DELTA TIME BETWEEN OUTPUT POINTS
CD             IOPT    - .EQ.1 SINUSOIDAL TEST SIGNAL
CD                     - .EQ.2 THREE LEVEL GRAY TEST SIGNAL
CD                     - .EQ.3 PERIODIC SQUARE WAVE TEST SIGNAL
CD             IFM     - .EQ.1 FM MODULATION
CD                     - .EQ.2 PHASE MODULATION
CD             A       - .IOPT.EQ.1 AMPLITUDE
CD             F       - .IOPT.EQ.1 FREQUENCY
CD             THETA   - .IOPT.EQ.1 PHASE
CD             K       - .IOPT.EQ.1 NUMBER OF POINTS
CD             TP      - .IOPT.EQ.2 DEFINES TIME POINTS OF BREAKS
CD         COMMON
CD             AR      - REAL PART OF SIGNAL BEING GENERATED
CD             AI      - IMAGINARY PART OF SIGNAL BEING GENERATED
CD             WR      - DUMMY COMMON ARGUMENT
CD             WI      - DUMMY COMMON ARGUMENT
```

SAP PROGRAM LISTING

```
CD
CD        OUTPUT
CD          CALL SEQUENCE
CD          COMMON
CD          PRINT
CD          TAPE
CD
CD     SUBROUTINES REQUIRED
CD              MAGTAP
CD
CD* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
      DIMENSION A(25),F(25),THETA(25),TP(3),AR(256),AI(256)
      DIMENSION WR(2048),WI(2048)
      COMMON/STORGE/AR,AI,WR,WI
      NR = MR
      NC = MC
      M = K
      TWOPI=6.28318531
      I2 = 2
      I4 = 4
      CALL MAGTAP(I4,NR,AR,IUNT)
C**** SET IFLAG
      IFLAG=IOPT+3
C**** INITIALIZE
      CON=0.0
      DO 1 I=1,NR
    1 AI(I)=0.0
      IF(IOPT.NE.1) GO TO 3
      DO 2 I=1,M
      F(I)=TWOPI*F(I)
      THETA(I)=0.0174532925*THETA(I)
    2 CONTINUE
    3 J=1
      K=1
      T = 0.0
      S=0.0
      A(5)=0.0
      ALAST=0.0
      TSUM=0.0
      SMEAN=0.0
    5 CONTINUE
      IF[IFLAG.EQ.6) S=A[1)
      IF[IFLAG.NE.5) GO TO 90
   70 DO 71 I=1,4
   71 SMEAN=SMEAN+A[I)*[TP[5)-TP[I))
      SMEAN = SMEAN/TP[5)
   90 CONTINUE
C**** GO TO IFLAG OPTION
   10 CONTINUE
      GO TO [400,500,600),IOPT
C**** SINUSOIDAL TEST SIGNAL - PHASE MODULATION
  400 DO 402 I=1,NR
      T=[I-1)*DT + TSUM
      TEMP=0.0
      DO 401 L=1,M
  401 TEMP=TEMP + A[L)*SIN[F[L)*T + THETA[L))
      AR[I)=TEMP
  402 CONTINUE
      GO TO 999
C**** THREE GRAY LEVEL TEST SIGNAL - PHASE MODULATION
  500 DO 503 I=1,NR
      IF[T.LT.TP[K)) GO TO 501
      S=S + A[K)
      K=K + 1
  501 IF[K.NE.6) GO TO 502
      K=1
      T=0.0
      S=0.0
  502 AR[I)= S - SMEAN
      T= T + DT
  503 CONTINUE
      GO TO 999
C**** PERIODIC SQUARE WAVE TEST SIGNAL - PHASE MODULATION
  600 DO 602 I=1,NR
      IF[T.LT.TP[1)) GO TO 601
      T= T -TP[1)
      S= -S
  601 AR[I)=S
      T=T+DT
  602 CONTINUE
```

SAP PROGRAM LISTING

```
C**** WRITE TAPE
  999 CALL MAGTAPE(2,NR,AR,IUNT)
      CALL MAGTAPE(2,NR,AI,IUNT)
      TSUM=T + DT
      J= J + 1
      IF(J.LE.NC) GO TO 10
      CALL MAGTAPE(4,NR,AR,IUNT)
      RETURN
      END
```

SAP PROGRAM LISTING

```
      SUBROUTINE TTRANS(IA,IB,IUNITA,IUNITB)
CD* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
CD
CD    PROGRAMMER AND DATE
CD            RICHARD C. THOMAS
CD            TRW SYSTEMS
CD            JUNE 1972
CD
CD    PURPOSE
CD
CD    USAGE
CD            CALL TTRANS(IA,IB,IUNITA,IUNITB,MBUFF)
CD
CD    DESCRIPTION OF PARAMETERS
CD
CD       INPUT
CD         CALLING SEQUENCE
CD            IA     - NUMBER OF WORDS/RECORD ON INPUT RECORD
CD            IB     - NUMBER OF REAL DATA INPUT RECORDS
CD            IUNITA - INPUT DATA TAPE UNIT
CD            IUNITB - OUTPUT DATA TAPE
CD            MBUFF  - SIZE OF WORKING BUFFER
CD         COMMON
CD            BUF    - WORK AREA
CD            BUFI   - WORK AREA
CD            BUFF   - WORK AREA
CD            BUFFI  - WORK AREA
CD         TAPE
CD
CD       OUTPUT
CD         CALL SEQUENCE
CD         COMMON
CD         TAPE
CD
CD    REMARKS AND RESTRICTIONS
CD    SUBROUTINES REQUIRED
CD            MAGTAP
CD
CD* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
      DIMENSION BUF( 256),BUFF(2048),BUFFI(2048),BUFI( 256)
```

SAP PROGRAM LISTING

```
      COMMON/STORGE/BUF,BUFI,BUFF,BUFFI
      I1 = 1
      I2 = 2
      I4 = 4
      MBUFF = 2048
      IBF = IA*IB
      IF[IBF.LT.MBUFF) MBUFF = IBF
C *** COMPUT MAXIMUM NUMBER OF OUTPUT RECORDS TO BE LOADED PER COMPLETE
C *** PASS OF INPUT TAPE.
      IX = MBUFF/IB
C *** COMPUTE NUMBER OF COMPLETE PASSES OF INPUT TAPE.
      MAXRD = IA/IX
      IZZ = IA/IX
      IY = IA/MAXRD
      IZ = IB/MAXRD
      DO 200 IMX=1,IZZ
      DO 100 IXX=1,MAXRD
      DO 50 L=1,IZ
      CALL MAGTAP[I1,IA,BUF,IUNITA)
      CALL MAGTAP[I1,IA,BUFI,IUNITA)
      DO 50 I=1,IY
      JK = IY*[IMX-1) + I
      LL = L + IZ*[IXX-1) + IZ*MAXRD*[I-1)
      BUFF[LL) = BUF[JK)
      BUFFI[LL) = BUFI[JK)
   50 CONTINUE
  100 CONTINUE
      DO 75 I=1,IX
      MX = IB*[I-1) + 1
      CALL MAGTAP[I2,IB,BUFF[MX),IUNITB)
      CALL MAGTAP[I2,IB,BUFFI[MX),IUNITB)
   75 CONTINUE
      CALL MAGTAP[I4,I4,BUF,IUNITA)
  200 CONTINUE
      CALL MAGTAP[I4,I4,BUF,IUNITB)
  900 RETURN
      END
      SUBROUTINE TWRITE(IU,NC,NR)
CD* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
CD
CD    PROGRAMMER AND DATE
CD            RICHARD C. THOMAS
CD            TRW SYSTEMS
CD            JUNE 1972
CD
CD    PURPOSE
CD            THIS SUBROUTINE WILL READ THE REAL AND IMAGINARY PARTS OF
CD            A SIGNAL ON TAPE AND WILL OUTPUT THE SIGNAL ON THE
CD            PRINTER.
CD
CD    USAGE
CD            CALL TWRITE(IU,NC,NR)
CD
CD    DESCRIPTION OF PARAMETERS
CD
CD       INPUT
CD         CALLING SEQUENCE
CD            IU      - DATA TAPE UNIT
CD            NC      - NUMBER OF COLUMNS IN DATA MATRIX
CD            NR      - NUMBER OF ROWS IN DATA MATRIX
CD         COMMON
CD            AR      - INPUT/OUTPUT BUFFER
CD            AI      - INPUT/OUTPUT BUFFER
CD            WR      - DUMMY COMMON VARIABLE
CD            WI      - DUMMY COMMON VARIABLE
CD       OUTPUT
CD         PRINT
CD
CD    SUBROUTINES REQUIRED
CD    NONE
CD
CD* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
      DIMENSION AR(256),AI(256),WR(2048),WI(2048)
      COMMON/STORGE/AR,AI,WR,WI
      I1=1
      I4=4
      CALL MAGTAPE(I4,I4,AR,IU)
```

SAP PROGRAM LISTING

```
      DO 50 J=1,NC
      CALL MAGTAP(I1,NR,AR,IU)
      CALL MAGTAP(I1,NR,AI,IU)
      WRITE(6,2005) J
 2005 FORMAT(1H ,8HCOLUMN =,I4)
      WRITE(6,2010) (AR(I),I=1,NR)
 2010 FORMAT(1H ,4HREAL/(8E16.8))
      WRITE(6,2015) (AI(I),I=1,NR)
 2015 FORMAT(1H ,9HIMAGINARY/(8E16.8))
   50 CONTINUE
      CALL MAGTAP(I4,I4,AR,IU)
      RETURN
      END
```

SAP PROGRAM LISTING

```
      SUBROUTINE TIMER(I,J,M,S)
CD* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
CD
CD    PROGRAMMER AND DATE
CD             RICHARD C. THOMAS
CD             TRW SYSTEMS
CD             JUNE 1972
CD
CD    PURPOSE
CD             USING TWO CALLS TO SUBROUTINE CLOCK AS INPUTS TIME CALCU-
CD             LATES THE RUN TIME IN MINUTES AND SECONDS OF A SEGMENT OF
CD             THE PROGRAM.
CD
CD    USAGE
CD
CD    DESCRIPTION OF PARAMETERS
CD
CD       INPUT
CD         CALLING SEQUENCE
CD             I       - INITIAL TIME
CD             J       - FINAL TIME
CD             M       - MINUTES
CD             S       - SECONDS
CD
CD       OUTPUT
CD         CALL SEQUENCE
CD    REMARKS AND RESTRICTIONS
CD    SUBROUTINES REQUIRED
CD    NONE
CD
CD* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
      X = J - I
      T = X/1024.
      M = T/60.
      Y = M
      S = T - Y*60.
      RETURN
      END
```

SAP PROGRAM LISTING

```
      SUBROUTINE ISAR(IUNIT1,IUNIT2,NR2,NC2,DT1,NR1,NC1,KSR)
CD* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
CD
CD    PROGRAMMER AND DATE
CD              RICHARD C. THOMAS
CD              TRW SYSTEMS
CD              JUNE 1972
CD
CD    PURPOSE
CD              ACCEPTS THE USER SUPPLIED TEST SIGNAL FROM TAPE, RESAM-
CD              PLES IT AT A HIGHER DATA RATE AND GENERATES AN OUTPUT
CD              SIGNAL TAPE COMPATIBLE WITH SAP.
CD
CD    USAGE
CD              CALL ISAR(IUNIT1,IUNIT2,NR2,NC2,DT1,NR1,NC1,KSR,BETA)
CD
CD    DESCRIPTION OF PARAMETERS
CD
CD       INPUT
CD         CALLING SEQUENCE
CD              IUNIT1 - INPUT TAPE UNIT
CD              IUNIT2 - OUTPUT TAPE UNIT
CD              NR2    - NUMBER OF ROWS IN OUTPUT ARRAY
CD              NC2    - NUMBER OF COLUMNS IN OUTPUT ARRAY
CD              DT1    - SAMPLING INTERVAL FOR INPUT DATA
CD              NR1    - NUMBER OF ROWS IN INPUT ARRAY
CD              NC1    - NUMBER OF COLUMNS IN INPUT ARRAY
CD              KSR    - DEFINES SAMPLING RATE FOR RESAMPLED SIGNAL
CD              BETA   - MODULATION INDEX, MULTIPLIES ENTIRE TEST SIGNAL
CD         COMMON
CD              AR     - WORK AREA
CD              AI     - WORK AREA
CD              S      - INPUT BUFFER
CD              SX     - INPUT BUFFER
CD         TAPE
CD
CD       OUTPUT
CD         CALL SEQUENCE
CD         COMMON
CD         CARD
CD          TAPE
CD
CD     SUBROUTINES REQUIRED
CD              MAGTAP
CD
CD     NONE
CD
CD* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
      DIMENSION AR[256),AI[256),S[2048),SX[2048)
      COMMON/STORGE/AR,AI,S,SX
      CALL MAGTAP[K4,NR1,S,IUNIT1)
      CALL MAGTAP[K4,NR1,S,IUNIT2)
      K1 = 1
      K2 = 2
      K4 = 4
      FLTKSR = KSR
      CALL MAGTAP[K1,NR1,S,IUNIT1)
      DO 5 I=1,NR2
      AI[I) = 0.0
    5 CONTINUE
      P1 = S[1)
      P2 = S[2)
      I1 = 2
      I2 = 1
      J1 = 1
      J2 = 1
      TERM = [P2-P1)/FLTKSR
      K = 1
   20 CONTINUE
      AR[I2) = P1 + TERM*FLOAT[K-1)
      I2 = I2 + 1
      K = K + 1
      IF[I2.LE.NR2) GO TO 40
      CALL MAGTAP[K2,NR2,AR,IUNIT2)
      CALL MAGTAP[K2,NR2,AI,IUNIT2)
      I2 = 1
      J2 = J2 + 1
      IF[J2.LE.NC2) GO TO 40
      CALL MAGTAP[K4,NR1,AR,IUNIT1)
      CALL MAGTAP[K4,NR1,AR,IUNIT2)
```

SAP PROGRAM LISTING

```
      RETURN
   40 CONTINUE
      IF(K.LE.KSR) GO TO 20
      K = 1
      I1 = I1 + 1
      IF(I1.GT.NR1) GO TO 80
   60 CONTINUE
      P1 = P2
      P2 = S(I1)
      TERM = (P2-P1)/FLTKSR
      GO TO 20
   80 CONTINUE
      J1 = J1 + 1
      IF(J1.GT.NC1) GO TO 100
      CALL MAGTAP(K1,NR1,S,IUNIT1)
      I1 =  1
      GO TO 60
  100 CONTINUE
      WRITE(6,2000)
 2000 FORMAT(53H * * * NOT ENOUGH INPUT DATA TO GENERATE OUTPUT * * *)
      STOP
      END
```

SAP PROGRAM LISTING

```
      SUBROUTINE MAGTAP[J)
CD* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
CD
CD    PROGRAMMER AND DATE
CD             RICHARD C. THOMAS
CD             TRW SYSTEMS
CD             JUNE 1972
CD
CD    PURPOSE
CD             MAGTAP IS A DUMMY ROUTINE THAT ALLOCATES CORE STORAGE FOR
CD             A CS-1 ASSEMBLY LANGUAGE ROUTINE NAMED MAGTAP. THE CS-1
CD             PROGRAM PEFORMS ALL TAPE OPERATIONS REQUIRED BY THE PRO-
CD             GRAM. FOR TAPE PARITY ERRORS THE MAGTAP ROUTINE ATTEMPTS
CD             RECOVERY A MAXIMUM OF TEN TIMES
CD             FOR UNRECOVERABLE PARITY ERRORS, FRAME COUNT ERRORS AND
CD             TIMING ERRORS THE COMPUTER COMES TO A HALT WITH NO PRINT-
CD             ED ERROR MESSAGES. THE OPERATOR IS ABLE TO DETERMINE THE
CD             NATURE OF THE ERROR BY INDICATOR LIGHTS ON THE TAPE DRIVE
CD
CD    USAGE
CD             CALL MAGTAP[IOPT,NUM,ARRAY,IUNT)
CD
CD    DESCRIPTION OF PARAMETERS
CD
CD       INPUT
CD         CALLING SEQUENCE
CD             IOPT   - .EQ.1 READ TAPE RECORD
CD             IOPT   - .EQ.2 WRITE TAPE RECORD
CD             IOPT   - .EQ.3 BACKSPACE RECORD
CD             IOPT   - .EQ.4 REWIND TAPE
CD             IOPT   - .EQ.5 WRITE FILE MARK
CD             IOPT   - .EQ.6 SPACE FILE FORWARD
CD             IOPT   - .EQ.7 SPACE FILE BACKWARD
CD             NUM    - NUMBER OF WORDS IN DATA RECORD
CD             ARRAY  - DATA RECORD
CD             IUNT   - TAPE UNIT
CD         TAPE
CD
CD       OUTPUT
CD         TAPE
```

SAP PROGRAM LISTING

```
CD
CD    REMARKS AND RESTRICTIONS
CD              WHEN TAPE OPERATION IS NEITHER A READ OR WRITE, NUM AND
CD              ARRAY ARE DUMMY PARAMETERS.
CD              THIS ROUTINE ALLOCATES THE NECESSARY CORE REQUIRED FOR
CD              THE CS-1 PROGRAM WHICH MUST BE LOADED BY PAPER TAPE. SEE
CD              PROGRAM OPERATING INSTRUCTIONS FOR CORRECT PROCEDURES.
CD    SUBROUTINES REQUIRED
CD    NONE
CD
CD* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
      DIMENSION A(300)
      A(1) = 1.0
      RETURN
      END
```

SAP PROGRAM LISTING

```
      SUBROUTINE TRFN(FREQ,IOPT,NORDER,FCEN,FCUT,RIP,TMAG,F,NF,NS,TR,TI
     1,TPH)
CD* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
CD
CD    PROGRAMMER AND DATE
CD             RICHARD C. THOMAS
CD             TRW SYSTEMS
CD             JUNE 1972
CD
CD    PURPOSE
CD             CALCULATES THE REAL AND IMAGINARY PARTS OF THE DESIRED
CD             TRANSFER FUNCTION
CD
CD    USAGE
CD             CALL TRFN(FREQ,IOPT,NORDER,FCEN,FCUT,RIP,TMAG,F,NF,NS,
CD                       TR,TI,TPH)
CD
CD    DESCRIPTION OF PARAMETERS
CD
CD       INPUT
CD         CALLING SEQUENCE
CD             FREQ   - FREQ AT WHICH TRANSFER FCN IS TO BE EVALUATED
CD             IOPT   - .EQ.1 BUTTERWORTH FILTER
CD                    - .EQ.2 CHEBYCHEV
CD                    - .EQ.3 INPUT DATA FILTER
CD             NORD   - ORDER OF THE BUTTERWORTH OR CHEBYCHEV FILTER
CD             FCEN   - CENTER FREQUENCY FOR BUTTERWORTH OR CHEBYCHEV
CD             FCUT   - CUTOFF FREQUENCY FOR BUTTERWORTH OR CHEBYCHEV
CD             RIP    - RIPPLE FOR BUTTERWORTH FILTER
CD             TMAG   - MAGNITUDE OF INPUT FILTER TRANSFER FUNCTION
CD             F      - FREQUENCY(IN HERTZ) AT WHICH MAGNITUDE AND PHASE
CD             NF     - DIMENSION OF TMAG,TPH AND F
CD             TPH    - CORRESPONDING PHASE(IN RADIANS)
CD
CD* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
      DIMENSION F(1),TMAG(1),TPH(1)
      DIMENSION C(6,6),CC(6)
      DATA ((C(I,J),I=1,6),J=1,6)/1.0,0.0,0.0,0.0,0.0,0.0,
     11.414213562,1.0,0.0,0.0,0.0,0.0,
     22.0,2.0,1.0,0.0,0.0,0.0,
     32.613125929,3.414213562,2.613125929,1.0,0.0,0.0,
     43.236067977,5.236067977,5.236067977,3.236067977,1.0,0.0,
     53.863703305,7.464101615,9.141620172,7.464101615,
     63.863703305,1.0/
      IF(FREQ.LT.0.0) GO TO 800
      GO TO (100,200,300), IOPT
C***  BUTTERWORTH FILTER
  100 CONTINUE
      IRSW=-1
      IISW= 1
      FSAVE=1.0
      ISW=-1
      CR=1.0
      CI=0.0
      FNORM= ABS(FREQ-FCEN)/(FCUT-FCEN)
      DO 10 I=1,NORDER
      FPROD=FSAVE*FNORM
      X= C(I,NORDER)*FPROD
      IF(ISW.LT.0) GO TO 3
      CR=CR + IRSW*X
      ISW=-1
      IRSW=-IRSW
      GO TO 10
    3 CI= CI + IISW*X
      ISW=1
      IISW=-IISW
   10 FSAVE = FPROD
      D=CR**2 + CI**2
      TR= CR/D
      TI=-CI/D
      RETURN
C***  CHEBYSHEV FILTER
  200 CONTINUE
      IF(NS.GT.1) GO TO 201
      X=1.0/RIP
      SINHI=ALOG(X+SQRT(X**2+1))
      XORDER=NORDER
      A=SINHI/XORDER
      TEMP1=EXP(A)
      TEMP2=EXP(-A)
      COSH=(TEMP1+TEMP2)/2.0
      SINH=(TEMP1-TEMP2)/2.0
      TANH=SINH/COSH
  201 CONTINUE
      FNORM= ABS(FREQ-FCEN)/(FCUT-FCEN)
      GO TO (210,220,230,220,250,220),NORDER
  210 CC(1)=1.0/TANH
      GO TO 269
  220 CONTINUE
      WRITE(6,2000) NORDER
 2000 FORMAT(1H1,20X90HYOU ARE IN A HEAP OF TROUBLE, BOY. NORDER HAS BEE
     1N ALTERED BY PROGRAM TO AN INVALID VALUE,I2)
      STOP
  230 CONTINUE
      TEMP = TANH**3 + 3.0*TANH
      CC(1) = (5.0*TANH**2 + 3.0)/TEMP
      CC(2) = 8.0*TANH/TEMP
      CC(3) = 4.0/TEMP
      GO TO 269
  250 CONTINUE
      TEMP1 = TANH*0.309017
      TEMP12 = TEMP1**2
      TEMP2 = TANH*0.809017
      TEMP22 = TEMP2**2
      TEMP3 = TEMP12 + 0.9045085
      TEMP4 = TEMP22 + 0.3454915
      TEMP5 = TEMP3*TEMP4
      CC(1) = 1.0/TANH + 2.0*TEMP1/TEMP3 + 2.0*TEMP2/TEMP4
      CC(2) =(2.0*TEMP1/TANH)/TEMP3 + (2.0*TEMP2/TANH)/TEMP4 +
     1(TEMP3 + TEMP4 + TANH**2)/TEMP5
      CC(3)=1.0/(TANH*TEMP3) + 1.0/(TANH*TEMP4) + ((TANH + 2.0*TEMP1
     1  + 2.0*TEMP2)/TEMP5)
      CC(4) = (2.0*(TEMP1 + TEMP2) + TANH)/(TANH*TEMP5)
      CC(5) =  1.0/(TANH*TEMP5)
      GO TO 269
  269 CONTINUE
      IRSW=-1
      IISW= 1
      FSAVE=1.0
      ISW=-1
      CR=1.0
      CI=0.0
      DO 271 I=1,NORDER
      FPROD=FSAVE*FNORM
      X=CC[I)*FPROD
      IF[ISW.LT.0) GO TO 272
      CR=CR + IRSW*X
      ISW=-1
      IRSW=-IRSW
      GO TO 271
  272 CI=CI+IISW*X
      ISW=1
      IISW=-IISW
  271 FSAVE = FPROD
      D=CR**2 + CI**2
      TR= CR/D
      TI=-CI/D
      RETURN
C***  INPUT DATA FILTER
  300 CONTINUE
      DO 301 N=NS,50
      IF[FREQ.LT.F[N+1)) GO TO 302
  301 CONTINUE
      NS=1
      GO TO 300
  302 NS=N
      DF=[FREQ - F[NS))/[F[NS+1) - F[NS))
      TMAGP=TMAG[NS) + DF*[TMAG[NS+1) - TMAG[NS))
      TPHP=TPH[NS) + DF*[TPH[NS+1) - TPH[NS))
      TR=TMAGP*COS[TPHP)
      TI=TMAGP*SIN[TPHP)
      RETURN
  800 NS=1
      RETURN
      END
END
```

3. PLOT GENERATION PROGRAM (PLTGEN)

3.1 MODULE/SUBROUTINE DESCRIPTIONS

The following list describes all the PLTGEN program modules and subroutines.

ANNØTE	Scaling and labeling routine
CØMAND	Constructs all plotter commands
GRID	Constructs plot grid
INITAL	Initializing routine
MAGTAP	Tape operations routine
NUMBER	Converts a floating point number to a Hollerith string
PEN	Pen select routine
PLØTER	Draws plot lines
PLØTPT	Repositions plot pen
PLTGEN	Main program for the plot software package
PRINT	Converts a string of Hollerith characters into the EAI plotter character set.

3.2 PLTGEN FLOW DIAGRAMS

The flow diagrams for PLTGEN showing both the main program and all subroutines are shown in the following figures.

FLOW FOR MAIN PROGRAM PLTGEN

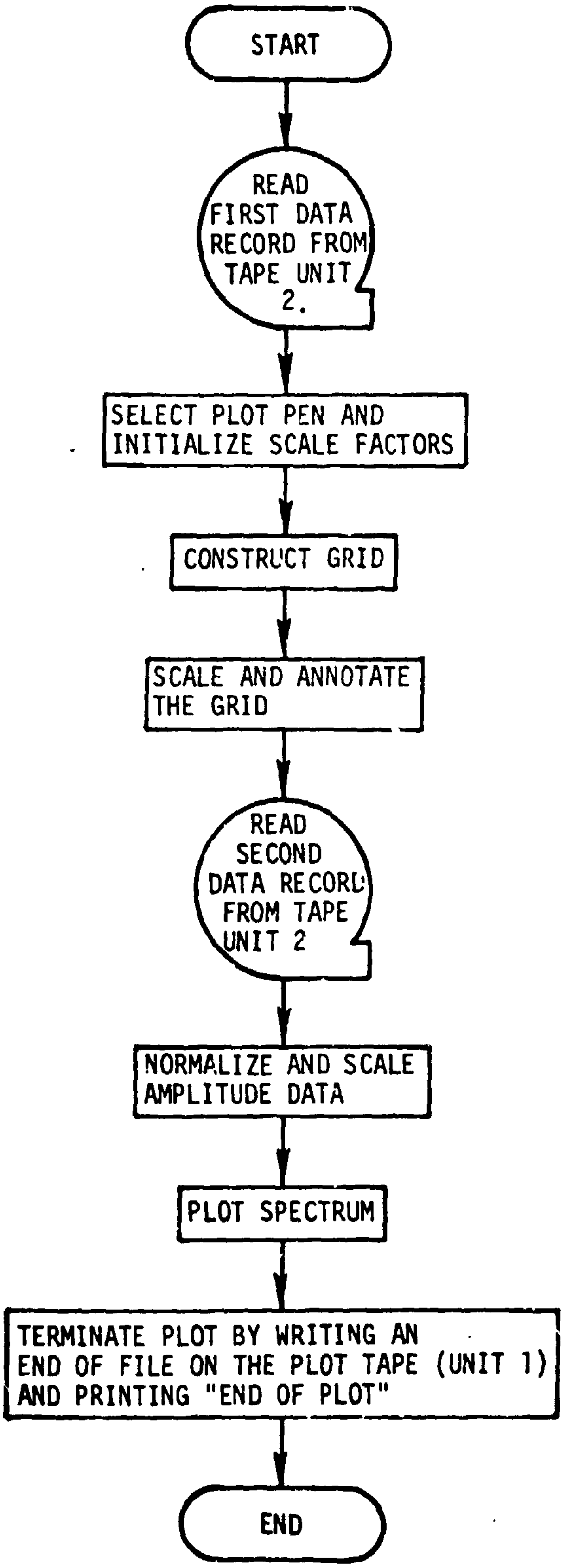

START
READ FIRST DATA RECORD FROM TAPE UNIT 2.
SELECT PLOT PEN AND INITIALIZE SCALE FACTORS
CONSTRUCT GRID
SCALE AND ANNOTATE THE GRID
READ SECOND DATA RECORD FROM TAPE UNIT 2
NORMALIZE AND SCALE AMPLITUDE DATA
PLOT SPECTRUM
TERMINATE PLOT BY WRITING AN END OF FILE ON THE PLOT TAPE (UNIT 1) AND PRINTING "END OF PLOT"
END

Figure 3-1. Flow for Main Program PLTGEN

FLOW FOR SUBROUTINE ANNOTE

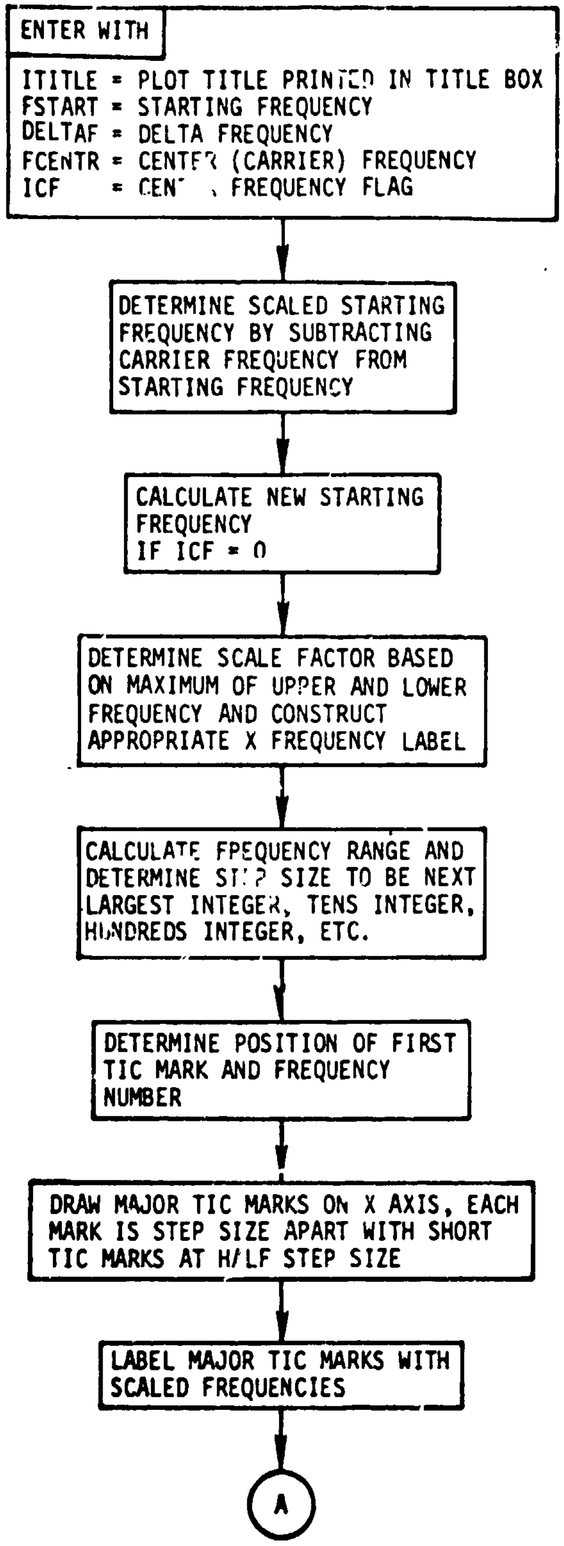

ENTER WITH
ITITLE = PLOT TITLE PRINTED IN TITLE BOX
FSTART = STARTING FREQUENCY
DELTAF = DELTA FREQUENCY
FCENTR = CENTER (CARRIER) FREQUENCY
ICF = CENTER FREQUENCY FLAG
DETERMINE SCALED STARTING FREQUENCY BY SUBTRACTING CARRIER FREQUENCY FROM STARTING FREQUENCY
CALCULATE NEW STARTING FREQUENCY IF ICF = 0
DETERMINE SCALE FACTOR BASED ON MAXIMUM OF UPPER AND LOWER FREQUENCY AND CONSTRUCT APPROPRIATE X FREQUENCY LABEL
CALCULATE FREQUENCY RANGE AND DETERMINE STEP SIZE TO BE NEXT LARGEST INTEGER, TENS INTEGER, HUNDREDS INTEGER, ETC.
DETERMINE POSITION OF FIRST TIC MARK AND FREQUENCY NUMBER
DRAW MAJOR TIC MARKS ON X AXIS, EACH MARK IS STEP SIZE APART WITH SHORT TIC MARKS AT HALF STEP SIZE
LABEL MAJOR TIC MARKS WITH SCALED FREQUENCIES
A

Figure 3-2. Flow for Subroutine ANNOTE

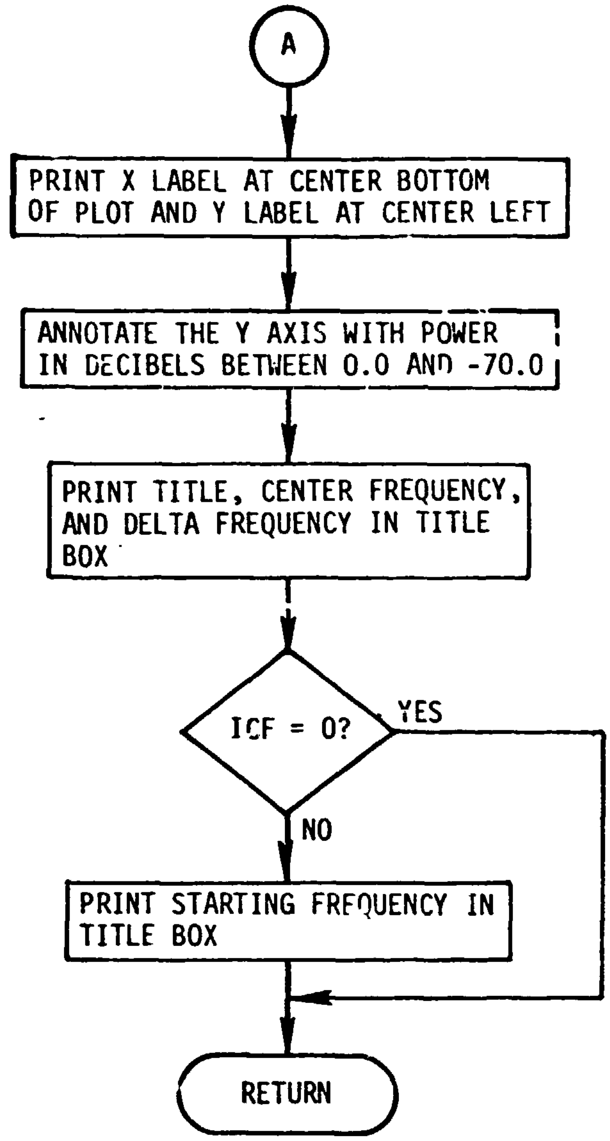

A
PRINT X LABEL AT CENTER BOTTOM OF PLOT AND Y LABEL AT CENTER LEFT
ANNOTATE THE Y AXIS WITH POWER IN DECIBELS BETWEEN 0.0 AND -70.0
PRINT TITLE, CENTER FREQUENCY, AND DELTA FREQUENCY IN TITLE BOX
ICF = 0?
YES
NO
PRINT STARTING FREQUENCY IN TITLE BOX
RETURN

Figure 3-2. Flow for Subroutine ANNOTE (Continued)

FLOW FOR SUBROUTINE GRID

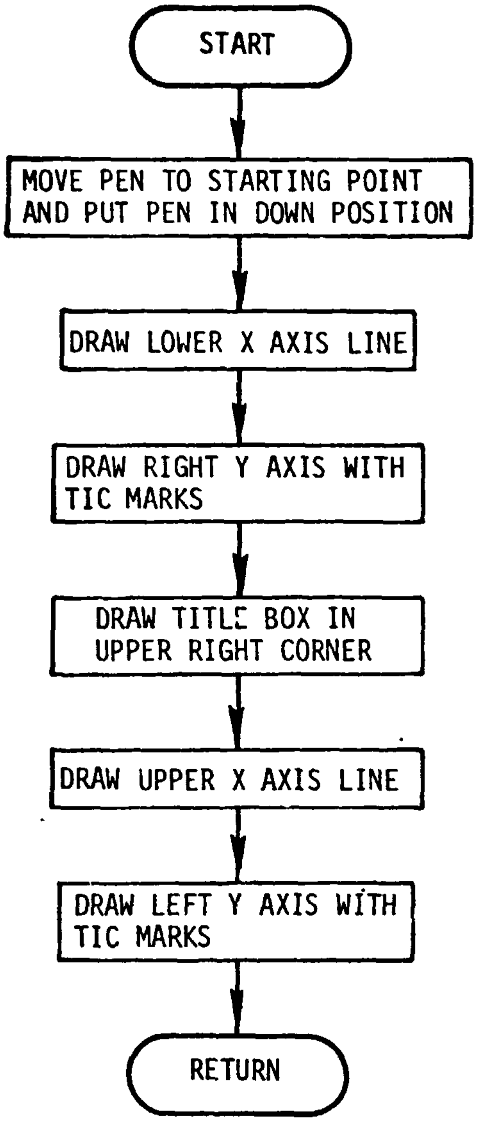

START
MOVE PEN TO STARTING POINT AND PUT PEN IN DOWN POSITION
DRAW LOWER X AXIS LINE
DRAW RIGHT Y AXIS WITH TIC MARKS
DRAW TITLE BOX IN UPPER RIGHT CORNER
DRAW UPPER X AXIS LINE
DRAW LEFT Y AXIS WITH TIC MARKS
RETURN

Figure 3-3. Flow for Subroutine GRID

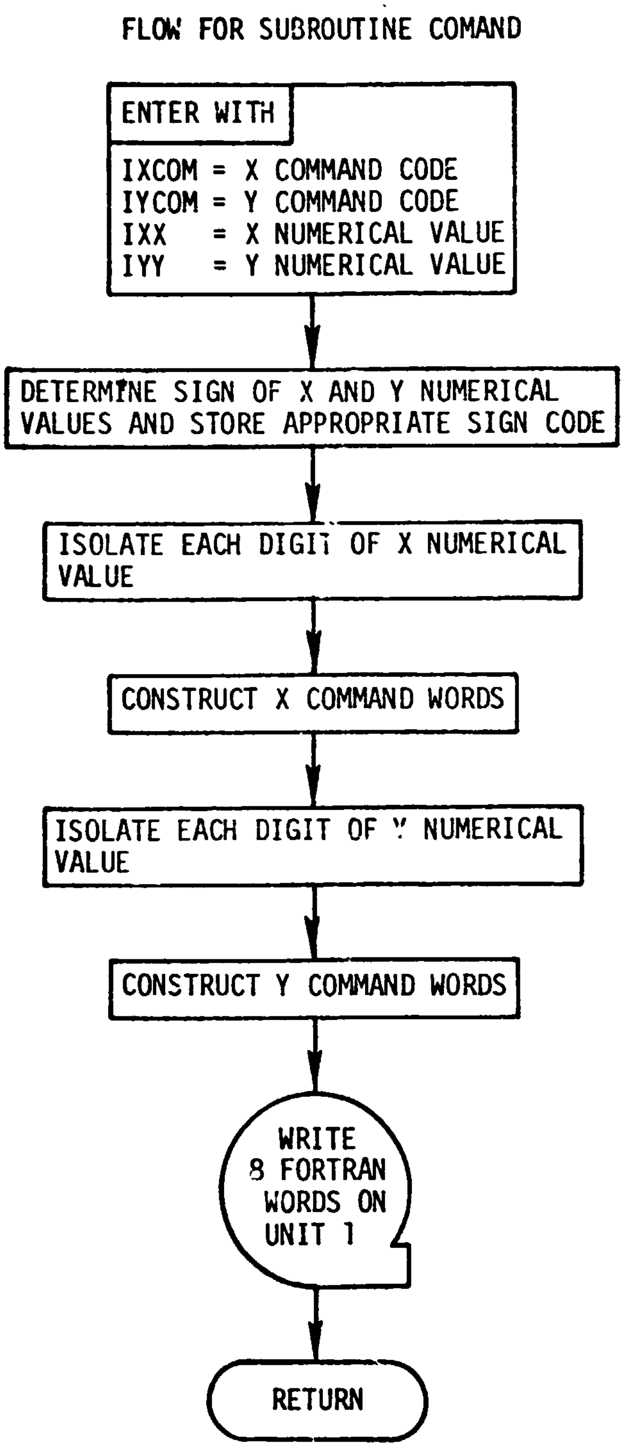

FLOW FOR SUBROUTINE COMAND
ENTER WITH
IXCOM = X COMMAND CODE
IYCOM = Y COMMAND CODE
IXX = X NUMERICAL VALUE
IYY = Y NUMERICAL VALUE
DETERMINE SIGN OF X AND Y NUMERICAL VALUES AND STORE APPROPRIATE SIGN CODE
ISOLATE EACH DIGIT OF X NUMERICAL VALUE
CONSTRUCT X COMMAND WORDS
ISOLATE EACH DIGIT OF Y NUMERICAL VALUE
CONSTRUCT Y COMMAND WORDS
WRITE 8 FORTRAN WORDS ON UNIT 1
RETURN

Figure 3-4. Flow for Subroutine COMAND

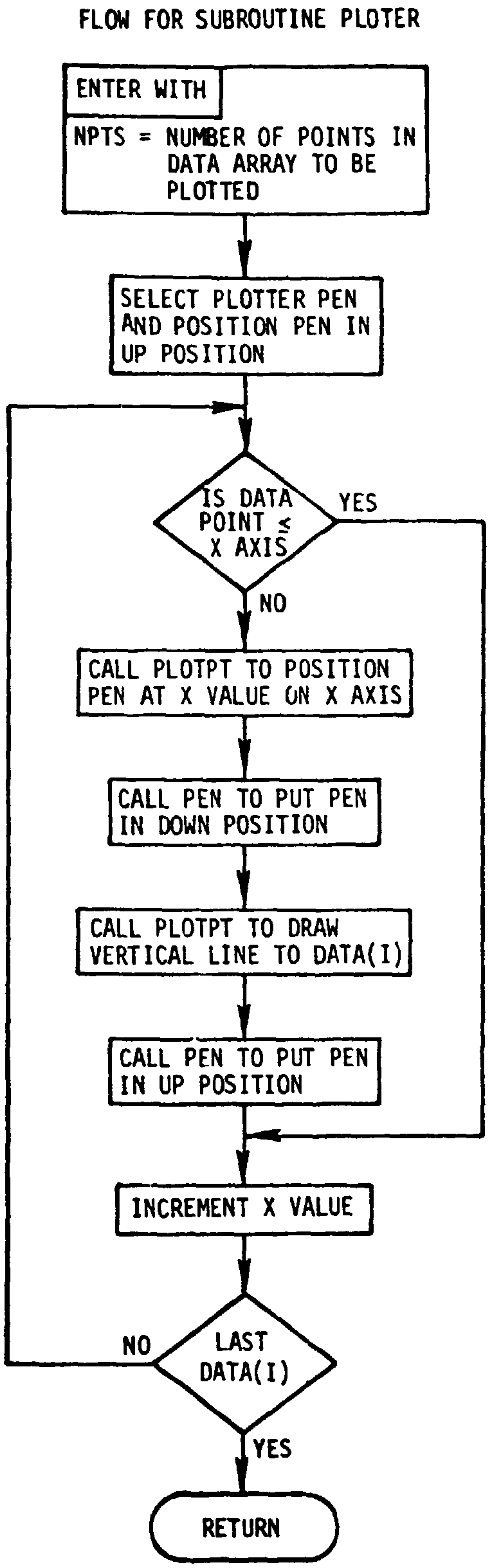

FLOW FOR SUBROUTINE PLOTER
ENTER WITH
NPTS = NUMBER OF POINTS IN DATA ARRAY TO BE PLOTTED
SELECT PLOTTER PEN AND POSITION PEN IN UP POSITION
IS DATA POINT ≤ X AXIS
YES
NO
CALL PLOTPT TO POSITION PEN AT X VALUE ON X AXIS
CALL PEN TO PUT PEN IN DOWN POSITION
CALL PLOTPT TO DRAW VERTICAL LINE TO DATA(I)
CALL PEN TO PUT PEN IN UP POSITION
INCREMENT X VALUE
NO
LAST DATA(I)
YES
RETURN

Figure 3-5. Flow for Subroutine PLOTER

FLOW FOR SUBROUTINE NUMBER

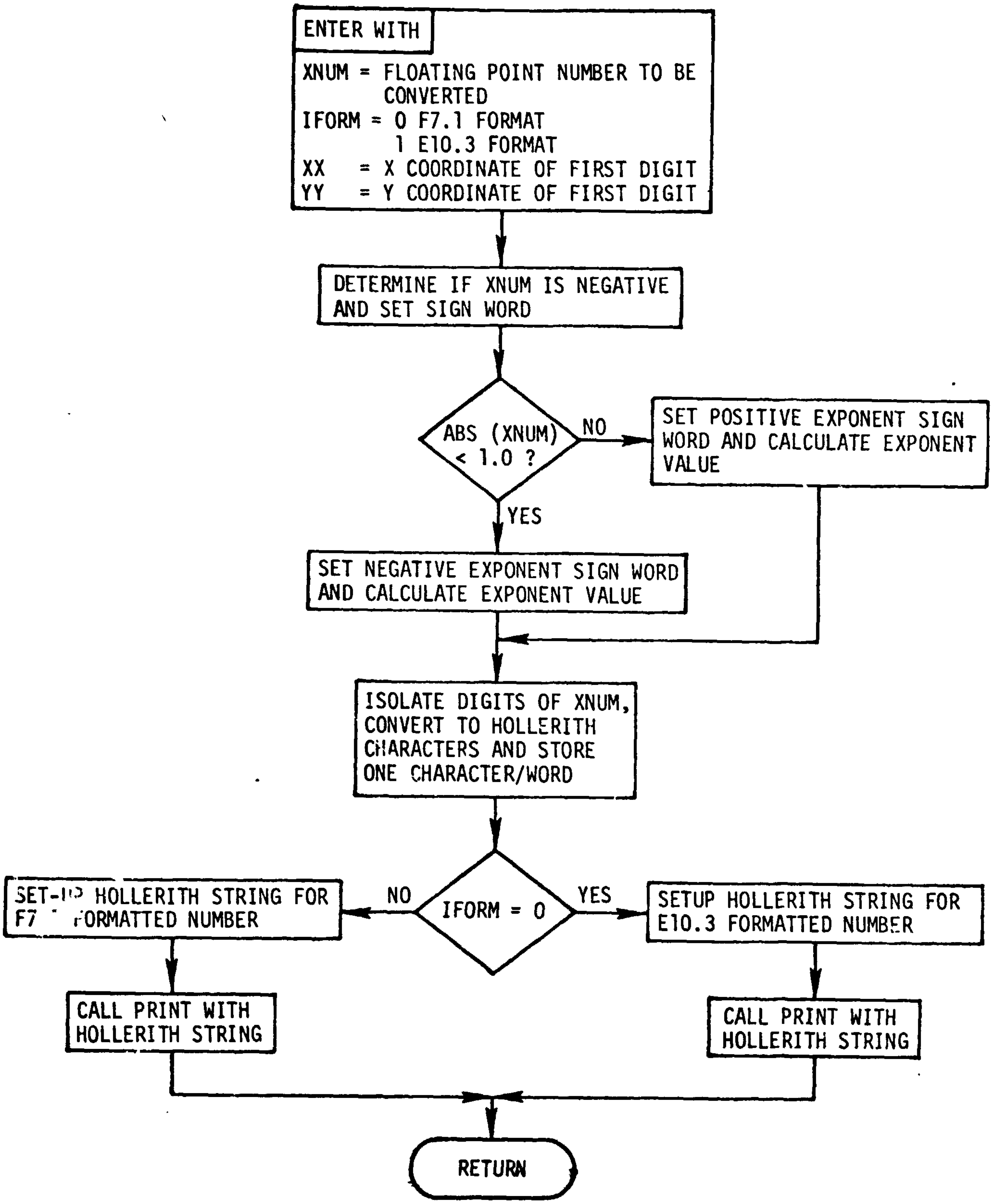

ENTER WITH
XNUM = FLOATING POINT NUMBER TO BE CONVERTED
IFORM = 0 F7.1 FORMAT
1 E10.3 FORMAT
XX = X COORDINATE OF FIRST DIGIT
YY = Y COORDINATE OF FIRST DIGIT
DETERMINE IF XNUM IS NEGATIVE AND SET SIGN WORD
ABS (XNUM) < 1.0 ?
NO
SET POSITIVE EXPONENT SIGN WORD AND CALCULATE EXPONENT VALUE
YES
SET NEGATIVE EXPONENT SIGN WORD AND CALCULATE EXPONENT VALUE
ISOLATE DIGITS OF XNUM, CONVERT TO HOLLERITH CHARACTERS AND STORE ONE CHARACTER/WORD
IFORM = 0
NO
SET-UP HOLLERITH STRING FOR F7.1 FORMATTED NUMBER
YES
SETUP HOLLERITH STRING FOR E10.3 FORMATTED NUMBER
CALL PRINT WITH HOLLERITH STRING
CALL PRINT WITH HOLLERITH STRING
RETURN

Figure 3-6. Flow for Subroutine NUMBER

FLOW FOR SUBROUTINE PRINT

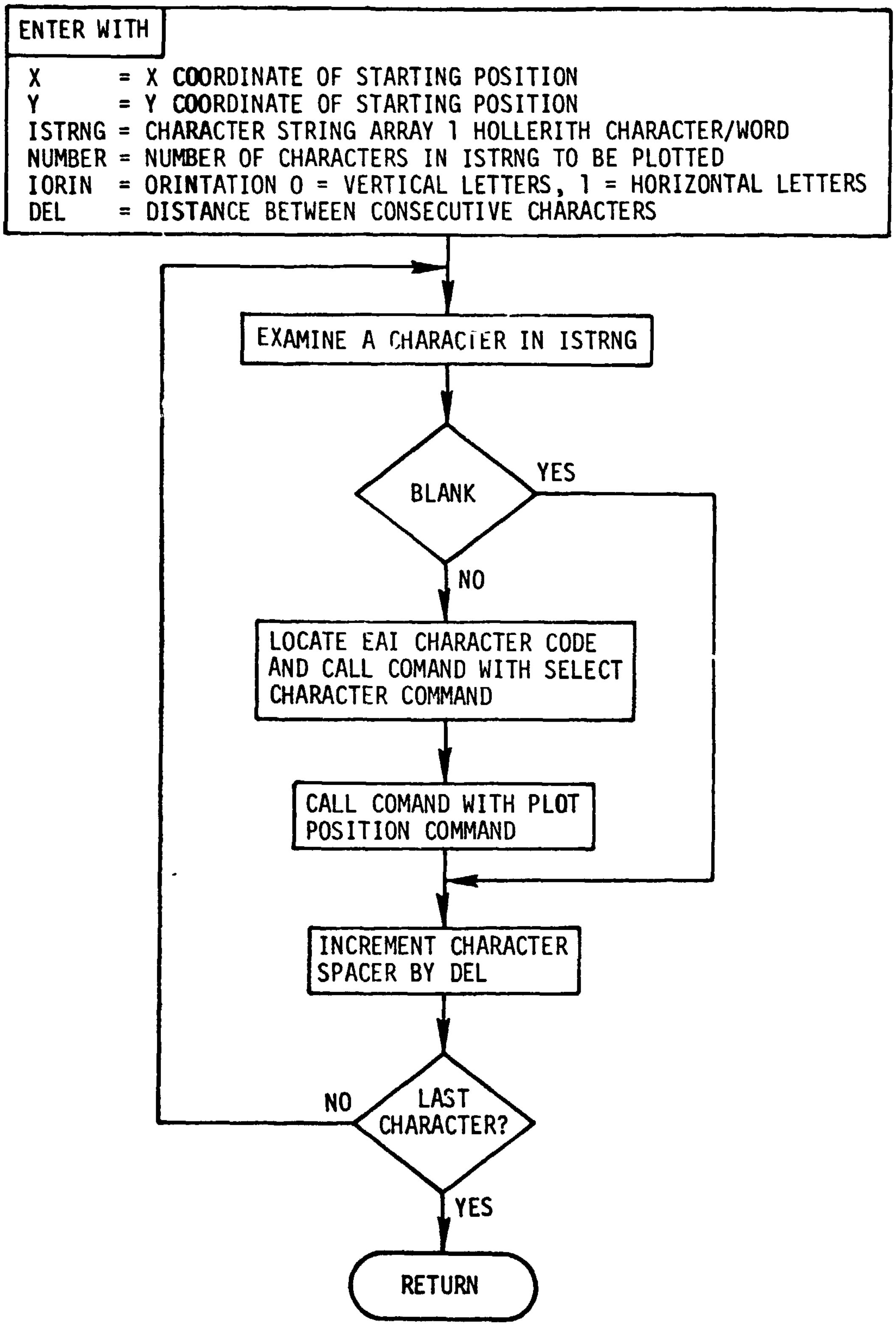

ENTER WITH
X = X COORDINATE OF STARTING POSITION
Y = Y COORDINATE OF STARTING POSITION
ISTRNG = CHARACTER STRING ARRAY 1 HOLLERITH CHARACTER/WORD
NUMBER = NUMBER OF CHARACTERS IN ISTRNG TO BE PLOTTED
IORIN = ORINTATION 0 = VERTICAL LETTERS, 1 = HORIZONTAL LETTERS
DEL = DISTANCE BETWEEN CONSECUTIVE CHARACTERS
EXAMINE A CHARACTER IN ISTRNG
BLANK
YES
NO
LOCATE EAI CHARACTER CODE AND CALL COMAND WITH SELECT CHARACTER COMMAND
CALL COMAND WITH PLOT POSITION COMMAND
INCREMENT CHARACTER SPACER BY DEL
LAST CHARACTER?
NO
YES
RETURN

Figure 3-7. Flow for Subroutine PRINT

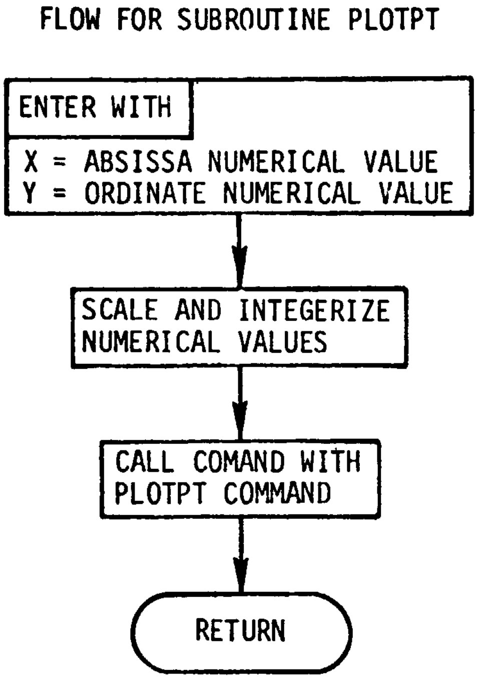

FLOW FOR SUBROUTINE PLOTPT
ENTER WITH
X = ABSISSA NUMERICAL VALUE
Y = ORDINATE NUMERICAL VALUE
SCALE AND INTEGERIZE NUMERICAL VALUES
CALL COMAND WITH PLOTPT COMMAND
RETURN

Figure 3-8. Flow for Subroutine PLOTPT

FLOW FOR SUBROUTINE INITAL

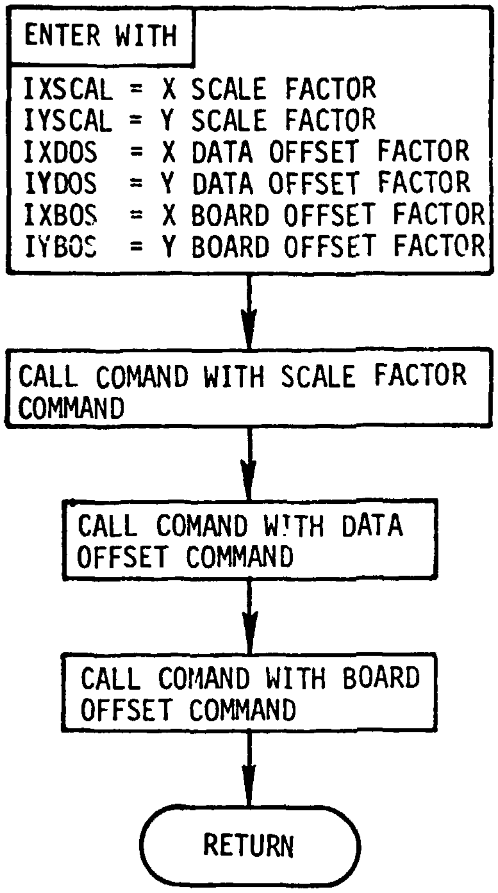

ENTER WITH
IXSCAL = X SCALE FACTOR
IYSCAL = Y SCALE FACTOR
IXDOS = X DATA OFFSET FACTOR
IYDOS = Y DATA OFFSET FACTOR
IXBOS = X BOARD OFFSET FACTOR
IYBOS = Y BOARD OFFSET FACTOR
CALL COMAND WITH SCALE FACTOR COMMAND
CALL COMAND WITH DATA OFFSET COMMAND
CALL COMAND WITH BOARD OFFSET COMMAND
RETURN

Figure 3-9. Flow for Subroutine INITAL

FLOW FOR SUBROUTINE PEN

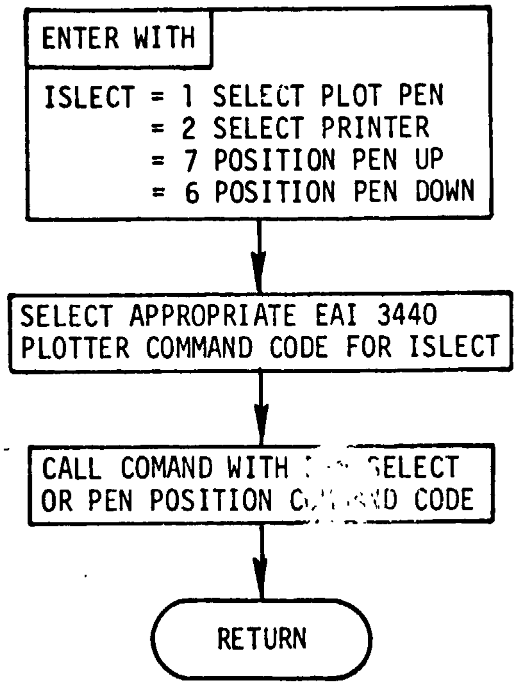

ENTER WITH
ISLECT = 1 SELECT PLOT PEN
= 2 SELECT PRINTER
= 7 POSITION PEN UP
= 6 POSITION PEN DOWN
SELECT APPROPRIATE EAI 3440 PLOTTER COMMAND CODE FOR ISLECT
CALL COMAND WITH PEN SELECT OR PEN POSITION COMMAND CODE
RETURN

Figure 3-10. Flow for Subroutine PEN

3.3 PLTGEN LISTINGS

This section presents a complete listing of the PLTGEN program.

PLTGEN PROGRAM LISTING

```
      MAINPROGRAM PLTGEN
CD* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
CD
CD    PROGRAMMER AND DATE
CD            G. L. HOUSER
CD            TRW SYSTEMS
CD            JUNE 1972
CD
CD    PURPOSE
CD            THE PLTGEN PROGRAM IS DESIGNED TO READ A DATA TAPE
CD            CONTAINING SPECTRUM DATA, GENERATED BY THE SAPDRI
CD            PROGRAM.  THE SPECTRUM DATA IS SCALED AND CONVERTED INTO
CD            PLOT COMMANDS FOR THE EAI PLOTTER.  A MAGNETIC TAPE
CD            CONTAINING THE FORMATTED PLOT COMMANDS IS GENERATED.
CD
CD    DESCRIPTION OF PARAMETERS
CD
CD       INPUT
CD         TAPE
CD            UNIT 2 [INPUT DATA TAPE)
CD               RECORD 1  [36 WORDS)
CD                 WORD  1       IPLPOS - PLOT POSITION FLAG
CD                       2 - 31  ITITLE - TITLE [1 HOLLERITH CHAR./WORD)
CD                      32       DELTAF - DELTA FREQUENCY
CD                      33       FCENTR - CARRIER FREGUENCY
CD                      34       FSTART - STARTING FREQUENCY
CD                      35       NPTS   - NUMBER OF SPECTRUM POINTS
CD                      36       IFC    - FLAG FOR CENTERING CARRIER
CD               RECORD 2  [480 WORDS)
CD                 WORD 1 - 480 DATA    - SPECTRUM VALUES [0.0 TO -70.0)
CD
CD       OUTPUT
CD         COMMON
CD            LABELED COMMON/A/  DATA [480)  -  SPECTRUM VALUES
CD         CARD
CD            NONE
CD         PRINT
CD            PRINTS ,END OF PLOT, WHEN ALL CALCULATIONS ARE COMPLETE
CD         TAPE
CD            UNIT 1 [OUTPUT PLOT TAPE)
```

PLTGEN PROGRAM LISTING

```
CD                RECORD 1 (ALL RECORDS CONTAIN 16 WORDS)
CD                  WORD 1
CD                    BYTE 1   -   X COMMAND CODE
CD                         2   -   THOUSANDS DIGIT
CD                         3   -   HUNDREDS DIGIT
CD                         4   -   TENS DIGIT
CD                         5   -   UNITS DIGIT
CD                  WORD 2         BLANK
CD                  WORD 3
CD                    BYTE 1   -   SIGN
CD                         2   -   NOT USED
CD                         3   -   BLANK
CD                         4   -   BLANK
CD                         5   -   BLANK
CD                  WORD 4 - 12    BLANK
CD                  WORD 13
CD                    BYTE 1   -   Y COMMAND CODE
CD                         2   -   THOUSANDS DIGIT
CD                         3   -   HUNDREDS DIGIT
CD                         4   -   TENS DIGIT
CD                         5   -   UNITS DIGIT
CD                  WORD 14        BLANK
CD                  WORD 15
CD                    BYTE 1   -   SIGN
CD                         2   -   NOT USED
CD                         3   -   BLANK
CD                         4   -   BLANK
CD                         5   -   BLANK
CD                  WORD 16        BLANK
CD
CD     REMARKS AND RESTRICTIONS
CD     NONE
CD
CD     SUBROUTINES REQUIRED
CD              ANNOTE
CD              GRID
CD              INITAL
CD              MAGTAP
CD              PEN
CD              PLOTER
CD              PLOTPT
CD
CD* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
      DIMENSION RECORD(36),      NWORD(2),      ITITLE(30)
      COMMON /A/  DATA(480)
      DATA IBLANK/5H     /
      I0=0
      I1=1
      I2=2
      I4=4
      I5=5
      I6=6
      I7=7
      I36=36
      X9=-9.999
C
C             REWIND TAPES - PLOT TAPE = UNIT 1, DATA TAPE = UNIT 2
C
      CALL MAGTAPE(I4,I2,DATA,I2)
C
C             READ 1ST DATA RECORD(36 WORDS) FROM UNIT 2
C
      CALL MAGTAPE(I1,I36,RECORD,I2)
C
      ISX=2222
      ISY=3333
      IBX=0
      IBY=0
      IDX=0
      IDY=0
C
C             IF IPLPOS = 0 UPPER PLOT
C                         1 LOWER PLOT
C
      IPLPOS=RECORD(1)
      IF(IPLPOS.EQ.1) IDY=9999
C
      DO 5 I=1,30
      ITITLE(I)=RECORD(I+1)
    5 CONTINUE
```

```
      DELTAF=RECORD(32)
      FCENTR=RECORD(33)
      FSTART=RECORD(34)
      NPTS=RECORD(35)
      IFC=RECORD(36)
C
C                 INITIALIZE PLOT ROUTINES
C
      CALL PEN(I1)
      CALL INITAL(ISX,ISY,IDX,IDY,IBX,IBY)
C
C                 CONSTRUCT GRID
C
      CALL GRID
C
C                 SCALE, LABEL, AND ANNOTATE THE GRID
C
      CALL ANNOTE(ITITLE,FSTART,DELTAF,FCENTR,IFC)
C
C                 READ SPECTRUM DATA FROM UNIT 2
C
      I480=480
      CALL MAGTAP(I1,I480,DATA,I2)
C
C                 NORMALIZE AND SCALE DATA FOR PLOTTING
C
      DO 10 I=1,NPTS
      DATA(I)=((DATA(I)+70.)/70.)*7.+1.
   10 CONTINUE
C
C                 PLOT SPECTRUM
C
      CALL PLOTER(NPTS)
C
C                 WRAP-UP PLOT BY POSITIONING PEN AND REWINDING TAPES
C
      CALL PEN (I7)
      CALL PLOTPT(X9,X9)
      CALL MAGTAP(I5,I2,DATA,I1)
      CALL MAGTAP(I4,I2,DATA,I2)
```

```
  WRITE(6,2)
2 FORMAT(12H END OF PLOT)
  STOP
  END
```

PLTGEN PROGRAM LISTING

```
      SUBROUTINE ANNOTE(ITITLE,FSTART,DELTAF,FCENTR,ICF)
CD* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
CD
CD    PROGRAMMER AND DATE
CD             G. L. HOUSER
CD             TRW SYSTEMS
CD             JUNE 1972
CD
CD    PURPOSE
CD             SUBROUTINE ANNOTE IS USED TO SCALE AND LABEL THE X AXIS,
CD             LABEL THE Y AXIS, AND PUT THE DESIRED INPUT VARIABLES IN
CD             THE TITLE BOX.
CD
CD    USAGE
CD             CALL ANNOTE (ITITLE,FSTART,DELTAF,FCENTR,ICF)
CD
CD    DESCRIPTION OF PARAMETERS
CD
CD       INPUT
CD         CALLING SEQUENCE
CD             ITITLE - TITLE OF PLOT, TO BE PLACED ON THE TOP LINE IN
CD                      THE TITLE BOX. 30 CHARACTERS MAXIMUM.
CD             FSTART - STARTING FREQUENCY IN HZ.
CD             DELTAF - DELTA FREQUENCY, INCREMENT BETWEEN CONSECUTIVE
CD                      FREQUENCY LINES IN HZ.
CD             FCENTR - CENTER FREQUENCY OR CARRIER FREQUENCY IN HZ.
CD             ICF    - FLAG FOR DETERMINING IF CENTER FREQUENCY IS
CD                      CENTERED IN CENTER OF GRID. (IFC = 0 - CENTERED)
CD
CD       OUTPUT
CD             NONE
CD
CD    REMARKS AND RESTRICTIONS
CD             ITITLE MUST BE AN ARRAY HAVING ONE HOLLERITH CHARACTER
CD             PER WORD.  THERE IS A MAXIMUM OF 30 CHARACTERS IN ITITLE.
CD
CD    SUBROUTINES REQUIRED
CD             PEN
CD             PLOTPT
CD             PRINT
```

PLTGEN PROGRAM LISTING

```
CD              NUMBER
CD
CD    METHOD
CD              THE CENTER FREQUENCY BIAS IS SUBTRACTED FROM THE STARTING
CD              FREQUENCY TO OBTAIN THE ACTUAL PLOTTED STARTING FREQ.
CD              THE MAXIMUM FREQ. IS EXAMINED AND A SCALE FACTOR IS
CD              DETERMINED.  THE POSITION AND VALUE OF THE FIRST TIC MARK
CD              IS CALCULATED AND THE DISTANCE BETWEEN TIC MARKS IS
CD              FOUND.  THE TIC MARKS ARE PLACED ON THE X AXIS.  THE TIC
CD              MARKS ALONG THE X AND Y AXIS ARE THEN LABELED.  THE TITLE
CD              AND OTHER DESIRED PARAMETERS ARE PRINTED IN THE TITLE
CD              BOX  THE STARTING FREQ. IS OMITTED FROM THE TITLE BOX
CD              WHEN IFC = 0.
CD
CD* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
      DIMENSION LABX[30),LABY[30),ITITLE[30),ITLE[30),IFST[30),IDELF[30)
      DIMENSION IHERTZ[4)
      DIMENSION ICFT[30)
      DATA[IFST[I),I=1,20)/5HS     ,5HT     ,5HA     ,5HR     ,5HT     ,
     *     5HI     ,5HN     ,5HG     ,5H      ,5HF     ,5HR     ,5HE     ,
     *     5HQ     ,5HU     ,5HE     ,5HN     ,5HC     ,5HY     ,5H      ,
     *     5H=     /
      DATA[ICFT[I),I=1,20)/5HC     ,5HA     ,5HR     ,5HR     ,5HI     ,
     *     5HE     ,5HR     ,5H      ,5HF     ,5HR     ,5HE     ,5HQ     ,
     *     5HU     ,5HE     ,5HN     ,5HC     ,5HY     ,5H      ,5H      ,
     *     5H=     /
      DATA[IDELF[I),I=1,20)/5HD     ,5HE     ,5HL     ,5HT     ,5HA     ,
     *      5H     ,5HF     ,5HR     ,5HE     ,5HQ     ,5HU     ,5HE     ,
     *      5HN    ,5HC     ,5HY     ,5H      ,5H      ,5H      ,5H      ,
     *      5H=    /
      DATA [LABY[I),I=1,15)/5HP     ,5HO     ,5HW     ,5HE     ,5HR     ,
     *                      5H,     ,5H      ,5HD     ,5HE     ,5HC     ,
     *                      5HI     ,5HB     ,5HE     ,5HL     ,5HS     /
      DATA [LABX[I),I=1,11)/5HF     ,5HR     ,5HE     ,5HQ     ,5HU     ,
     *                      5HE     ,5HN     ,5HC     ,5HY     ,5H,     ,
     *                      5H      /
      DATA [IHERTZ[I),I=1,4)/5HK     ,5HM     ,5HH     ,5HZ     /
      DATA IBLANK/5H     /
      INTEGER UP,DOWN
      DELF=DELTAF
      FSTR=FSTART
      FCNTR=FCENTR
      FCNT=FCNTR
C
C             DETERMINE SCALED (BIASED) STARTING FREQUENCY BY
C             SUBTRACTING OFF CARRIER FREQUENCY
C
      FSTRT=FSTART - FCNTR
      FCNTR=0.0
      IFC=ICF
      UP=7
      DOWN=6
      DELX=.0066
      I0=0
      I1=1
      I2=2
      I14=14
      I15=15
      I20=20
      I30=30
C
C             CALCULATE STARTING FREQUENCY IF IFC=0
C
      IF(IFC.EQ.0) FSTRT=FCNTR-240.*DELF
C
C             CHOOSE SCALE FACTOR FOR FREQUENCY LABELS
C
      FMAX=460.*DELF+FSTRT
      FMAX=AMAX1(ABS(FMAX),ABS(FSTRT))
      IF(FMAX-1.E+4) 20,20,10
   10 IF(FMAX-1.E+7) 30,30,40
   20 CONTINUE
      LABX(12)=IHERTZ(3)
      LABX(13)=IHERTZ(4)
      LABX(14)=IBLANK
      SCALE=1.E+00
      GO TO 50
   30 CONTINUE
      LABX(12)=IHERTZ(1)
      LABX(13)=IHERTZ(3)
      LABX(14)=IHERTZ(4)
      SCALE=1.E+3
      GO TO 50
   40 CONTINUE
      LABX(12)=IHERTZ(2)
      LABX(13)=IHERTZ(3)
      LABX(14)=IHERTZ(4)
      SCALE=1.E+6
   50 CONTINUE
C
C               DETERMINE FREQUENCY RANGE AND STEP SIZE
C
      FSTRTS=FSTRT/SCALE
      DELFS=DELF/SCALE
      FRANGE=480.*DELFS
      STEP=FRANGE/24.
      IF(STEP-1.0) 60,60,70
   60 CONTINUE
      STEP=.5
      GO TO 100
   70 CONTINUE
      IF(STEP-10.) 74,100,77
   74 CONTINUE
      ISTEP=STEP
      SSTEP=ISTEP
      ISTEP=ISTEP+1
      IF(SSTEP-STEP) 76,75,76
   75 ISTEP=ISTEP-1
   76 STEP=ISTEP
      GO TO 100
   77 CONTINUE
      I=0
   80 CONTINUE
      I=I+1
      IREM=STEP/10.**I
      IF(IREM.LT.10) GO TO 90
      GO TO 80
   90 CONTINUE
      STEP=(IREM+1)*10**I
C
```

```
C                DETERMINE POSITION OF FIRST TIC MARK AND FIRST FREQUENCY
C                NUMBER
C
  100 IFIRST=FSTRTS/STEP
      STARTY=1.
      STARTX=-8.222
      XSTOP=7.777
      DX=.03333333/DELFS
      FIRST=IFIRST*STEP+STEP
      FIRSTX=STARTX+[FIRST-FSTRTS)*DX
      DELTAX=DX*STEP
C
C                PUT TIC MARKS ON X GRID
C
      TIC=.06
      TICY=STARTY-TIC
      TIC2=STARTY-2.*TIC
      CALL PEN[UP)
      CALL PLOTPT[STARTX,STARTY)
      CALL PEN[DOWN)
      CALL PLOTPT[STARTX,TIC2)
      CALL PEN[UP)
      FIRST2=FIRSTX-DELTAX/2.
      X=FIRSTX
      X2=FIRST2
  110 CONTINUE
      IF[X-XSTOP) 120,120,130
  120 CONTINUE
      CALL PLOTPT[X2,STARTY)
      CALL PEN[DOWN)
      CALL PLOTPT[X2,TICY)
      CALL PEN[UP)
      CALL PLOTPT[X,STARTY)
      CALL PEN[DOWN)
      CALL PLOTPT[X,TIC2)
      CALL PEN[UP)
      X=X+DELTAX
      X2=X2+DELTAX
      GO TO 110
  130 CONTINUE
```

```
C
C                LABEL TIC MARKS WITH SCALED FREQUENCIES
C
      CALL PEN(I2)
      Y=.75
      X=STARTX-.25
      CALL NUMBER(FSTRTS,I0,X,Y)
      FRACT=.0001
      X=FIRSTX-.2
      COUNT=FIRST
  140 CONTINUE
      IF(X-XSTOP) 145,145,170
  145 CONTINUE
      IF(STEP-1.0) 160,160,150
  150 ICOUNT=COUNT+FRACT
      IF(ICOUNT.LT.0) ICOUNT=COUNT-FRACT
      COUNT=ICOUNT
  160 CONTINUE
      CALL NUMBER(COUNT,I0,X,Y)
      X=X+DELTAX
      COUNT=COUNT+STEP
      GO TO 140
C
C                PRINT X LABEL AT CENTER BOTTOM OF PLOT
C
  170 CONTINUE
      DO 180 I=1,30
      ITLE(I)=ITITLE(I)
  180 CONTINUE
      DY=1.
      X=-.5
      Y=.1
      CALL PRINT(X,Y,LABX,I14,I1,DELX)
      X=-9.200
      Y=7.5
      DELY=.1
      CALL PRINT(X,Y,LABY,I15,I0,DELY)
C
C                LABEL Y AXIS WITH DECIBEL SCALE VALUES (0.0  -  -70.0)
C
```

```
      Y=8.0
      X=-9.06
      XN=10.
      DO 190 I=1,8
      XN=XN-10.
      CALL NUMBER(XN,I0,X,Y)
      Y=Y-DY
  190 CONTINUE
C
C               PRINT TITLE, CENTER FREQ., AND STARTING FREQ IN TITLE BOX
C
      XT=5.866
      YT=9.333
      CALL PRINT(XT,YT,ITLE,I30,I1,DELX)
      YT=YT-.3
      CALL PRINT(XT,YT,IDELF,I20,I1,DELX)
      X=XT+21.*DELX
      CALL NUMBER(DELF,I1,X,YT)
      YT=YT-.3
      CALL PRINT(XT,YT,ICFT,I20,I1,DELX)
      CALL NUMBER(FCNT,I1,X,YT)
C
C               OMIT STARTING FREQ IF IFC=0
C
      IF(IFC.EQ.0) GO TO 200
      YT=YT-.3
      CALL PRINT(XT,YT,IFST,I20,I1,DELX)
      CALL NUMBER(FSTR,I1,X,YT)
  200 CONTINUE
      RETURN
      END
      SUBROUTINE GRID
CD* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
CD
CD    PROGRAMMER AND DATE
CD              G. L. HOUSER
CD              TRW SYSTEMS
CD              JUNE 1972
CD
CD    PURPOSE
CD              THIS SUBROUTINE IS USED TO DRAW A RECTANGULAR BOX 25 IN.
CD              BY 7 IN. WITH TIC MARKS 1 INCH APART ALONG THE Y AXES.
CD              A SMALL TITLE BOX IS ALSO DRAWN IN THE UPPER RIGHT
CD              CORNER.  THESE DIMENSIONS ARE VALID ONLY WHEN USING A
CD              1.5 TO 1. X SCALE FACTOR AND A 1. TO 1. Y SCALE FACTOR.
CD
CD    USAGE
CD              CALL GRID
CD
CD    DESCRIPTION OF PARAMETERS
CD
CD       INPUT
CD         CALLING SEQUENCE
CD              NONE
CD
CD    REMARKS AND RESTRICTIONS
CD              NONE
CD
CD    SUBROUTINES REQUIRED
CD              PEN
CD              PLOTPT
CD
CD    METHOD
CD              THE RECTANGULAR BOX IS DRAWN COUNTER CLOCKWISE STARTING
CD              IN THE LOWER LEFT CORNER.  THE TIC MARKS AND TITLE BOX
CD              ARE DRAWN AS THEY ARE ENCOUNTERED.
CD
CD* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
      INTEGER UP,DOWN
      I0=0
      I1=1
```

PLTGEN PROGRAM LISTING

```
      IE=2
      DOWN=6
      UP=7
      TIC=.1
      STARTX=-8.555
      STARTY=1.
      CALL PEN(UP)
      CALL PLOTPT(STARTX,STARTY)
      CALL PEN(DOWN)
      X=STARTX+.3333
      DELTA=1.
C
C                DRAW LOWER X AXIS LINE
C
      DO 10 I=1,18
      CALL PLOTPT(X,STARTY)
      IF(I.EQ.18) GO TO 10
      IF(I.EQ.17) DELTA=.3333
      X=X+DELTA
   10 CONTINUE
C
C                DRAW RIGHT Y AXIS WITH TIC MARKS
C
      TICX=X+TIC
      Y=STARTY
      DO 20 I=1,7
      Y=Y+1.
      CALL PLOTPT(X,Y)
      IF(I.EQ.7) GO TO 20
      CALL PLOTPT(TICX,Y)
      CALL PLOTPT(X,Y)
   20 CONTINUE
C
C                DRAW TITLE BOX IN UPPER RIGHT CORNER
C
      YBOX=Y+1.5
      CALL PLOTPT(X,YBOX)
      XBOX=X
      DO 30 I=1,2
      XBOX=XBOX-1.2
      CALL PLOTPT(XBOX,YBOX)
   30 CONTINUE
      CALL PLOTPT(XBOX,Y)
      CALL PEN(UP)
      CALL PLOTPT(X,Y)
      CALL PEN(DOWN)
C
C               DRAW UPPER X AXIS
C
      TICY=Y+TIC
      X=X-.3333
      DELTA=1.
      DO 40 I=1,18
      CALL PLOTPT(X,Y)
      IF(I.EQ.18) GO TO 40
      IF(I.EQ.17) DELTA=.3333
      X=X-DELTA
   40 CONTINUE
C
C               DRAW LEFT Y AXIS WITH TIC MARKS
C
      TICX=X-TIC
      DO 50 I=1,7
      Y=Y-1.
      CALL PLOTPT(X,Y)
      IF(I.EQ.7) GO TO 50
      CALL PLOTPT(TICX,Y)
      CALL PLOTPT(X,Y)
   50 CONTINUE
C
      RETURN
      END
```

PLTGEN PROGRAM LISTING

```
      SUBROUTINE COMAND (IXCOM,IYCOM,IXX,IYY)
CD* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
CD
CD    PROGRAMMER AND DATE
CD             G. L. HOUSER
CD             TRW SYSTEMS
CD             JUNE 1972
CD
CD    PURPOSE
CD             TO CONSTRUCT AND WRITE ON MAGNETIC TAPE, ALL EAI PLOTTER
CD             COMMANDS.
CD
CD    USAGE
CD             CALL COMAND (IXCOM,IYCOM,IXX,IYY)
CD
CD    DESCRIPTION OF PARAMETERS
CD
CD       INPUT
CD         CALLING SEQUENCE
CD             IXCOM   -  X COMMAND CODE
CD             IYCOM   -  Y COMMAND CODE
CD             IXX     -  X NUMERICAL VALUE TO BE PLOTTED.
CD             IYY     -  Y NUMERICAL VALUE TO BE PLOTTED.
CD
CD       OUTPUT
CD         TAPE
CD             MAGNETIC TAPE UNIT 1 (SEE REMARKS BELOW)
CD
CD    REMARKS AND RESTRICTIONS
CD             THE UNIVAC 1230 COMPUTER WORD CONSISTS OF 5, SIX 29T
CD             BYTES.  EACH DECIMAL DIGIT (CHARACTER) IS REPRESENTED BY
CD             ONE BYTE.  THE FORTRAN MAGTAP ROUTINE MUST WRITE TEN
CD             BYTES FOR EACH 5 BYTE FORTRAN WORD SPECIFIED.  THE FIRST
CD             5 BYTES CONTAIN THE COMPUTER WORD SPECIFIED AND THE
CD             SECOND 5 BYTES ARE ALWAYS BLANK.
CD
CD             THE EAI PLOTTER CAN BE MANUALLY SET TO READ RECORDS AND
CD             WORD SIZES OF VARIOUS LENGTHS.  EACH PLOT COMMAND MUST
CD             HOWEVER CONTAIN 6 BYTES OF INFORMATION.  THE PLOT CONSOLE
CD             IS THEREFORE SET TO READ A 6 WORD RECORD, CONSISTING OF
CD                 12 BYTE (CHARACTER) WORDS.  THE FIRST 12 CHARACTER WORD
CD                 IS THE X COMMAND WORD AND THE SIXTH WORD IS THE Y COMMAND
CD                 WORD.
CD
CD                 BOTH THE X AND Y COMMAND WORDS ON THE PLOT TAPE HAVE THE
CD                 SAME STRUCTURE WHICH IS AS FOLLOWS0  THE X COMMAND IS THE
CD                 FIRST BYTE.  THE SECOND THROUGH THE FIFTH BYTES ARE THE
CD                 THOUSANDTHS, HUNDREDTHS, TENTHS, AND UNITS POSITIONS OF
CD                 THE VALUE TO BE PLOTTED.  THE SIXTH THROUGH THE TENTH
CD                 BYTES ARE NOT USED.  THE ELEVENTH BYTE IS THE SIGN AND
CD                 THE TWELVTH BYTE IS NOT USED.
CD
CD                 THE IWORD ARRAY IS CONSTRUCTED IN THIS ROUTINE.  IT IS
CD                 STRUCTURED TO SATISFY THE ABOVE REQUIREMENTS, KEEPING IN
CD                 MIND THAT THE MAGTAP ROUTINE WRITES A BLANK WORD AFTER
CD                 EACH 5 CHARACTER WORD SPECIFIED TO BE WRITTEN ON THE MAG.
CD                 TAPE.
CD
CD       SUBROUTINES REQUIRED
CD                 MAGTAP
CD
CD       METHOD
CD                 THE X COMMAND CODE AND NUMERICAL VALUE ARE DETERMINED AND
CD                 THE X PLOT COMMAND IS FORMED IN IWORD(1) AND IWORD(2).
CD                 IWORD(7) AND IWORD(8) CONTAIN THE PROPERLY FORMATTED
CD                 Y COMMAND.  THE 8 WORD RECORD IS THEN WRITTEN ON TAPE
CD                 UNIT 1.   TAPE UNIT 1 THEN ACTUALLY CONTAINS A 16 WORD
CD                 RECORD, DUE TO THE INTELACED BLANK WORDS. (SEE REMARKS)
CD
CD* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
      DIMENSION IWORD(8),ICHAR(5)
      I1=1
      I2=2
      I6=6
      IXSIGN=0
      IYSIGN=0
      IX=IXX
      IY=IYY
      IXC=IXCOM
      IYC=IYCOM
```

```
C
C                 DETERMINE SIGN OF X AND Y NUMERICAL VALUES AND SET SIGN
C                 WORDS
C
      IF(IX.LT.0) IXSIGN=40
      IF(IY.LT.0) IYSIGN=40
      IX=IABS(IX)
      IY=IABS(IY)
      DO 10 I=1,15
      IWORD(I)=0
   10 CONTINUE
C
C                 LOAD X COMMAND CODE
C
      ICHAR(1)=IXC
C
C                 ISOLATE EACH DIGIT OF X NUMERICAL VALUE
C
      DO 20 I=1,4
      K=6-I
      ITEMP=IX/10
      ICHAR(K)=IX-ITEMP*10
      IX=ITEMP
   20 CONTINUE
C
C                 BUILD X COMMAND WORDS
C
      IWORD(1)=ICHAR(1)*2**24+ICHAR(2)*2**18+ICHAR(3)*2**12+
     *            ICHAR(4)*2**6+ICHAR(5)
      IWORD(2)=IXSIGN*2**24
C
C                 LOAD Y COMMAND CODE
C
      ICHAR(1)=IYC
C
C                 ISOLATE EACH DIGIT OF Y NUMERICAL VALUE
C
      DO 30 I=1,4
      K=6-I
      ITEMP=IY/10
      ICHAR[K)=IY-ITEMP*10
      IY=ITEMP
   30 CONTINUE
C
C                 BUILD Y COMMAND WORDS
C
      IWORD[7)=ICHAR[1)*2**24+ICHAR[2)*2**18+ICHAR[3)*2**12+
     *          ICHAR[4)*2**6+ICHAR[5)
      IWORD[8)=IYSIGN*2**24
C
C                 WRITE EIGHT FORTRAN WORDS ON UNIT 1 [SEE REMARKS ABOVE)
C
      CALL MAGTAPE[I2,18,IWORD,I1)
C
      RETURN
      END
```

PLTGEN PROGRAM LISTING

```
      SUBROUTINE PLOTER (NPTS)
CD* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
CD
CD    PROGRAMMER AND DATE
CD             G. L. HOUSER
CD             TRW SYSTEMS
CD             JUNE 1972
CD
CD    PURPOSE
CD             SUBROUTINE PLOTER IS USED TO DRAW STRAIGHT LINES FROM A
CD             BASE POSITION TO EACH SCALED Y VALUE IN THE INPUT DATA
CD             ARRAY.
CD
CD    USAGE
CD             CALL PLOTER (NPTS)
CD
CD    DESCRIPTION OF PARAMETERS
CD
CD       INPUT
CD         CALLING SEQUENCE
CD             NPTS   - NUMBER OF DATA POINTS (STRAIGHT LINES) TO BE
CD                      PLOTTED
CD
CD         COMMON
CD             COMMON  /A/  DATA(480) - ARRAY OF Y DATA POINTS
CD                                      TO BE PLOTTED.
CD
CD       OUTPUT
CD             NONE
CD
CD    REMARKS AND RESTRICTIONS
CD             DATA POINTS IN DATA ARRAY MUST HAVE VALUE BETWEEN
CD             1. AND 8.  NO ERROR MESSAGE IS GIVEN.
CD
CD    SUBROUTINES REQUIRED
CD             PEN
CD             PLOTPT
CD
CD    METHOD
CD             EACH DATA VALUE IS COMPARED TO THE BASE POSITION, IF IT
```

PLTGEN PROGRAM LISTING

```
CD              IS EQUAL, THE X POSITION COUNTER IS INCREMENTED, IF THE
CD              VALUE IS GREATER, A LINE IS DRAWN FROM THE BASE POSITION
CD              TO THE Y DATA VALUE, AND THEN THE POSITION COUNTER IS
CD              INCREMENTED.
CD
CD* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
      COMMON/A/ DATA(480)
      INTEGER UP,DOWN
      NPT=NPTS
      II=1
      DOWN=6
      UP=7
C
C        DATA POINT VALUES MUST BE BETWEEN 1. AND 8.
C
      YBIAS=1.0
      XSTART=-6.222
      DX=.0333
C
C             SELECT PEN 1, PLACE PEN IN UP POSITION
C
      CALL PEN(II)
      CALL PEN(UP)
C
C             MOVE PEN TO START POSITION
C
      CALL PLOTPT(XSTART,YBIAS)
C
C            EXAMINE AND PLOT EACH INPUT POINT
C
      X=XSTART
      DO 50 I=1,NPT
      Y=DATA(I)
      IF(Y-YBIAS) 30,30,20
   20 CONTINUE
      CALL PLOTPT(X,YBIAS)
      CALL PEN(DOWN)
      CALL PLOTPT(X,Y)
      CALL PEN(UP)
   30 CONTINUE
```

```
      X=X+DX
   50 CONTINUE
C
C                   PLACE PEN IN UP POSITION AND RETURN
C
      CALL PEN(UP)
      RETURN
      END
      SUBROUTINE NUMBER(XNUM,IFORM,XX,YY)
CD* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
CD
CD    PROGRAMMER AND DATE
CD             G. L. HOUSER
CD             TRW SYSTEMS
CD             JUNE 1972
CD
CD    PURPOSE
CD             SUBROUTINE NUMBER CONVERTS A FLOATING POINT NUMBER INTO
CD             A FORMATTED HOLLERITH STRING. AVAILABLE SPECIFIED FORMATS
CD             ARE F7.1 AND E10.3.
CD
CD    USAGE
CD             CALL NUMBER (XNUM,IFORM,XX,YY)
CD
CD    DESCRIPTION OF PARAMETERS
CD
CD       INPUT
CD         CALLING SEQUENCE
CD             XNUM    - FLOATING POINT NUMBER TO BE CONVERTED
CD             IFORM   - FORMAT SELECTOR  0 = F7.1,  1 = E10.3
CD             X       - X COORDINATE OF FIRST DIGIT
CD             Y       - Y COORDINATE OF FIRST DIGIT
CD
CD       OUTPUT
CD             NONE
CD
CD    REMARKS AND RESTRICTIONS
CD             SUBROUTINE NUMBER DOES NOT CHECK FOR, OR PROTECT AGAINST
CD             THE INPUT NUMBER EXCEEDING THE SPECIFIED FORMAT LIMITS.
CD
CD    SUBROUTINES REQUIRED
CD             PRINT
CD
CD    METHOD
CD             THE SIGN AND EXPONENTIAL MAGNITUDE OF THE INPUT NUMBER
CD             IS DETERMINED.  EACH DIGIT OF THE NUMBER IS THEN ISOLATED
CD             AND CONVERTED TO HOLLERITH DATA.  THE PROPERLY FORMATTED
CD             NUMBER CONVERTED TO A HOLLERITH STRING IS THEN SENT TO
CD              SUBROUTINE PRINT.
CD
CD* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
      DIMENSION  ICODE(14),ICHAR(10),IFIELD(80)
      DATA (ICODE(I),I=1,14)/5H0    ,5H1    ,5H2    ,5H3    ,5H4
     * ,5H5    ,5H6    ,5H7    ,5H8    ,5H9    ,5H+    ,5H-    ,5H.
     * ,5HE    /
      DATA IBLANK/5H     /
      I0=0
      I1=1
      I2=2
      I7=7
      I10=10
      I=0
      NPOWER=0
      IPOWER=0
      DELX=.0666
      X=XX
      Y=YY
      IORIN=1
      IFRMAT=IFORM
      XN=XNUM
      DO 5 K=1,80
      IFIELD(K)=IBLANK
    5 CONTINUE
C
C               DETERMINE IF NUMBER IS NEGATIVE AND SET SIGN WORD
C
      IS=IBLANK
      IF(XN-0.0) 30,100,50
   30 CONTINUE
      IS=ICODE(12)
   50 CONTINUE
      XN=ABS(XN)
C
C               DETERMINE IF NUMBER HAS A NEGATIVE EXPONENT AND SET
C               EXPONENT SIGN WORD
C
      IF(XN-1.0) 60,80,80
   60 CONTINUE
```

```
      ISP=ICODE(12)
C
C                   CALCULATE EXPONENT IF IT IS NEGATIVE
C
   65 I=I+1
      XPOWER=10.**I
      POWER=1./XPOWER
      IF(XN+1.E-20-POWER) 65,70,70
   70 CONTINUE
      IPOWER=I
      NPOWER=IPOWER+4
      GO TO 100
C
C                   SET POSITIVE EXPONENT SIGN WORD AND DETERMINE EXPONENT
C                   VALUE
C
   80 CONTINUE
      ISP=ICODE(11)
   85 I=I+1
      POWER=10.**I
      IF(XN+1.E-6-POWER) 90,85,85
   90 CONTINUE
      IPOWER=I-1
      NPOWER=4-IPOWER
  100 CONTINUE
      IX=XN*10.**NPOWER+1.E-3
C
C                   ISOLATE DIGITS OF NUMBER BEING CONVERTED
C
      DO 110 J=1,5
      K=6-J
      ITEMP=IX/10
      N=IX-ITEMP*10+1
      ICHAR(K)=ICODE(N)
      IX=ITEMP
  110 CONTINUE
C
C                   IF F7.1 FORMAT GO TO 120
C
      IF(IFRMAT.EQ.0) GO TO 120
```

```
C
C                 SET UP HOLLERITH STRING FOR E10.3 FORMATTED NUMBER
C
      ITEMP=IPOWER/10
      N=IPOWER-ITEMP*10+1
      ICHAR(6)=ICODE(N)
      IX=ITEMP
      ITEMP=IX/10
      N=IX-ITEMP*10+1
      ICHAR(5)=ICODE(N)
      IFIELD(1)=IS
      IFIELD(2)=ICHAR(1)
      IFIELD(3)=ICODE(13)
      IFIELD(4)=ICHAR(2)
      IFIELD(5)=ICHAR(3)
      IFIELD(6)=ICHAR(4)
      IFIELD(7)=ICODE(14)
      IFIELD(8)=ISP
      IFIELD(9)=ICHAR(5)
      IFIELD(10)=ICHAR(6)
      CALL PRINT(X,Y,IFIELD,I10,IORIN,DELX)
      RETURN
C
C                 SET UP HOLLERITH STRING FOR F7.1 FORMATTED NUMBER
C
  120 CONTINUE
      IST=4-IPOWER
      J=0
      DO 160 I=1,7
      IF(I-IST)160,130,140
  130 CONTINUE
      IFIELD(I)=IS
      GO TO 160
  140 CONTINUE
      IF(I.NE.6) GO TO 150
      IFIELD(6)=ICODE(13)
      GO TO 160
  150 CONTINUE
      J=J+1
      IFIELD(I)=ICHAR(J)
  160 CONTINUE
      CALL PRINT(X,Y,IFIELD,I7,IORIN,DELX)
C
      RETURN
      END
```

PLTGEN PROGRAM LISTING

```
      SUBROUTINE PLOTPT (X,Y)
CD* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
CD
CD    PROGRAMMER AND DATE
CD             G. L. HOUSER
CD             TRW SYSTEMS
CD             JUNE 1972
CD
CD    PURPOSE
CD             SUBROUTINE PLOTPT IS USED TO GIVE THE PEN A COMMAND TO
CD             MOVE FROM ITS PRESENT POSITION TO A NEW POSITION (X,Y).
CD             THE PEN MAY BE IN EITHER THE UP OR DOWN POSITION.
CD
CD    USAGE
CD             CALL PLOTPT (X,Y)
CD
CD    DESCRIPTION OF PARAMETERS
CD
CD       INPUT
CD         CALLING SEQUENCE
CD             X      - ABSISSA VALUE BETWEEN -9.999 AND +9.999
CD             Y      - ORDINATE VALUE BETWEEN -9.999 AND +9.999
CD
CD       OUTPUT
CD             NONE
CD
CD    REMARKS AND RESTRICTIONS
CD             IF THE X OR Y VALUES EXCEED THE ABOVE LIMITATIONS THE
CD             PLOTTER WILL RETURN TO THE CENTER OF THE PLOT BOARD AND
CD             CONTINUE PLOTTING.
CD
CD    SUBROUTINES REQUIRED
CD             COMAND
CD
CD    METHOD
CD             THE X AND Y COORDINATES ARE SCALED AND SENT TO SUBROUTINE
CD             COMAND.
CD
CD* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
      II=1
```

```
XX=X
YY=Y
IX=XX*1000.
IY=YY*1000.
CALL COMAND(11,11,IX,IY)
RETURN
END
```

PLTGEN PROGRAM LISTING

```
      SUBROUTINE INITAL (IXSCAL,IYSCAL,IXDOS,IYDOS,IXBOS,IYBOS)
CD* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
CD
CD    PROGRAMMER AND DATE
CD            G. L. HOUSER
CD            TRW SYSTEMS
CD            JUNE 1972
CD
CD    PURPOSE
CD            THE PURPOSE OF SUBROUTINE INITAL IS TO INITIALIZE
CD            THE PLOT PROGRAM BY SETTING UP SCALE FACTORS, BOARD
CD            OFFSET, AND DATA OFFSET COMMANDS.
CD
CD    USAGE
CD            CALL INITAL (IXSCAL,IYSCAL,IXDOS,IYDOS,IXBOS,IYBOS)
CD
CD    DESCRIPTION OF PARAMETERS
CD
CD       INPUT
CD         CALLING SEQUENCE
CD            IXSCAL - X SCALE FACTOR (3333 IS NOMINAL ONE TO ONE
CD                     CORRESPONDENCE)
CD            IYSCAL - Y SCALE FACTOR (SAME AS ABOVE)
CD            IXDOS  - X DATA OFFSET (0000 IS NO OFFSET)
CD            IYDOS  - Y DATA OFFSET (0000 IS NO OFFSET)
CD            IXBOS  - X BOARD OFFSET (SAME AS ABOVE)
CD            IYBOS  - Y BOARD OFFSET (SAME AS ABOVE)
CD
CD
CD       OUTPUT
CD            NONE
CD
CD    REMARKS AND RESTRICTIONS
CD            NONE
CD
CD    SUBROUTINES REQUIRED
CD            COMAND
CD
CD    METHOD
CD            THE SCALE FACTORS, BOARD OFFSET,AND DATA OFFSET
CD                COMMANDS ARE FORMATTED AND SENT TO SUBROUTINE COMAND.
CD
CD* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
      DIMENSION NWORD(2)
      IXS=IXSCAL
      IYS=IYSCAL
      IXD=IXDOS
      IYD=IYDOS
      IXB=IXBOS
      IYB=IYBOS
      I1=1
      I2=2
      I4=4
C
C                SET UP SCALE FACTOR
C
      ISC=2
      CALL COMAND(ISC,ISC,IXS,IYS)
C
C                SET UP DATA OFFSET
C
      IDOC=3
      CALL COMAND(IDOC,IDOC,IXD,IYD)
C
C                SET UP BOARD OFFSET
C
      IBOC=4
      CALL COMAND(IBOC,IBOC,IXB,IYB)
C
      RETURN
      END
      SUBROUTINE PEN(ISLECT)
CD* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
CD
CD    PROGRAMMER AND DATE
CD             G. L. HOUSER
CD             TRW SYSTEMS
CD             JUNE 1972
CD
CD    PURPOSE
CD             SUBROUTINE PEN IS USED TO SELECT EITHER THE PLOT PEN OR
CD             THE PRINTER, AND TO GIVE PEN UP OR PEN DOWN COMMANDS.
CD
CD    USAGE
CD             CALL PEN (ISLECT)
CD
CD    DESCRIPTION OF PARAMETERS
CD
CD       INPUT
CD         CALLING SEQUENCE
CD             ISLECT - 1  PEN 1 (PLOTTER)
CD                      2  PEN 2 (PRINTER)
CD                      6  PEN DOWN
CD                      7  PEN UP
CD
CD       OUTPUT
CD             NONE
CD
CD    REMARKS AND RESTRICTIONS
CD             NONE
CD
CD    SUBROUTINES REQUIRED
CD             COMAND
CD
CD    METHOD
CD             ISLECT IS CHECKED TO DETERMINE IF THE REQUEST IS A SELECT
CD             PEN OR PRINTER COMMAND, THE APPROPRIATE CODE IS SENT
CD             TO SUBROUTINE COMAND, ISLECT = 6 OR 7 AUTOMATICALLY GIVES
CD             PEN UP OR PEN DOWN CODE
CD
CD
```

```
CD* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
      IS=ISELECT
      IX=0
      IY=IS
C
C                IS = 1, SELECT PEN 1 WITH CODE 11
C
      IF(IS.EQ.1) IY=11
C
C                IS = 2, SELECT PRINTER WITH CODE 12
C
      IF(IS.EQ.2) IY=12
C
      CALL COMAND(IX,IY,IX,IX)
C
      RETURN
      END
      SUBROUTINE PRINT(X,Y,ISTRNG,NUMBER,IORIN,DEL)
CD* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
CD
CD    PROGRAMMER AND DATE
CD            G. L. HOUSER
CD            TRW SYSTEMS
CD            JUNE 1972
CD
CD    PURPOSE
CD            SUBROUTINE PRINT IS USED TO CONVERT A STRING OF HOLLERITH
CD            CHARACTERS INTO THE EAI PLOTTER CHARACTER SET.  THE
CD            CONVERTED CHARACTERS ARE THEN SENT TO SUBROUTINE COMAND
CD            TO BE FORMED INTO PLOT COMMANDS.
CD
CD    USAGE
CD            CALL PRINT (X,Y,ISTRNG,NUMBER,IORIN,DEL)
CD
CD    DESCRIPTION OF PARAMETERS
CD
CD       INPUT
CD         CALLING SEQUENCE
CD            X      - X COORDINATE VALUE, STARTING POSITION BETWEEN
CD                     -9.999 AND +9.999
CD            Y      - Y COORDINATE VALUE, STARTING POSITION BETWEEN
CD                     -9.999 AND +9.999
CD            ISTRNG - ARRAY CONTAINING CHARACTER STRING
CD            NUMBER - NUMBER OF CHARACTERS IN CHARACTER STRING
CD            IORIN  - ORINTATION OF CHARACTERS  0 - VERTICAL,
CD                     1 - HORIZONTAL
CD            DEL    - DISTANCE BETWEEN CHARACTERS IN INCHES
CD
CD       OUTPUT
CD            NONE
CD
CD    REMARKS AND RESTRICTIONS
CD            THE MAXIMUM NUMBER OF CHARACTERS TO BE CONVERTED IN ONE
CD            STRING IS 80.
CD
CD            THE CHARACTER STRING MUST BE CONTAINED IN AN ARRAY HAVING
CD            ONE CHARACTER PER WORD
```

PLTGEN PROGRAM LISTING

```
C
C     SUBROUTINES REQUIRED
C              COMAND
C
C     METHOD
C              EACH CHARACTER IN THE STRING IS COMPARED WITH EACH
C              CHARACTER IN THE EAI CHARACTER SET.  WHEN A MATCH IS MADE
C              THE CODE IS SENT TO SUBROUTINE COMAND.  THE SPACING
C              VARIABLE IS INCREMENTED AND THE NEXT CHARACTER IN THE
C              STRING IS PROCESSED.
C
C* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
      DIMENSION ISTRNG[80],ICHAR[45],ICODE[45],LABEL[80]
C
C              EAI CHARACTER TABLE
C
      DATA [ICHAR[I),I=1,44)/5H0     ,5H1     ,5H2     ,5H3     ,5H4     ,
     *        5H5     ,5H6     ,5H7     ,5H8     ,5H9     ,5HA     ,5HB     ,
     *        5HC     ,5HD     ,5HE     ,5HF     ,5HG     ,5HH     ,5HI     ,
     *        5HJ     ,5HK     ,5HL     ,5HM     ,5HN     ,5HO     ,5HP     ,
     *        5HQ     ,5HR     ,5HS     ,5HT     ,5HU     ,5HV     ,5HW     ,
     *        5HX     ,5HY     ,5HZ     ,5H-     ,5H+     ,5H[     ,5H)     ,
     *        5H=     ,5H*     ,5H,     ,5H.     /
      DATA [ICODE[I),I=1,44)/ 46, 1, 2, 3, 4, 5, 6, 7,10,11,61,62,63
     *   ,64,65,66,67,70,71,41,42,43,44,45,46,47,50,51,22,23,24,25,26
     *   ,27,30,31,40,60,34,74,13,54,33,73/
C
      DATA IBLANK/5H     /
      NUM=NUMBER
      ID=IORIG
      DELTA=DEL
      I0=0
      I1=1
      ISLECT=5
      XX=X
      YY=Y
      DO 10 I=1,NUM
      LABEL[I)=ISTRNG[I)
   10 CONTINUE
C
```

```
C                 IDENTIFY EACH CHARACTER IN THE STRING
C
      DO 60 I=1,NUM
      IX=XX*1000.
      IY=YY*1000.
C
C                 IF CHARACTER IS A BLANK, INCREMENT CHARACTER SPACER
C
      IF(LABEL(I).EQ.IBLANK) GO TO 50
C
C                 SEARCH EAI CHARACTER TABLE
C
      DO 30 J=1,44
      N=J
      IF(LABEL(I).EQ.ICHAR(N)) GO TO 40
   30 CONTINUE
   40 CONTINUE
C
C                 LOAD CHARACTER AND ORIENTATION CODE
C
      ICH=ICODE(N)
      IF(ID.EQ.1) ICH=-ICH
C
C                 SEND SELECT CHARACTER COMMAND
C
      CALL COMAND(IO,ISLECT,IO,ICH)
C
C                 SEND PLOT CHARACTER COMMAND
C
      CALL COMAND(I1,I1,IX,IY)
   50 CONTINUE
      IF(ID.EQ.1) GO TO 55
C
C                 INCREMENT Y SPACER IF HORIZONTAL LETTERS (VERTICAL STRING)
C
      YY=YY+DELTA
      GO TO 60
   55 CONTINUE
C
C                 INCREMENT X SPACER IF VERTICAL LETTERS (HORIZONTAL STRING)
```

```
C
      XX=XX+DELTA
   60 CONTINUE
C
      RETURN
      END
```

PLTGEN PROGRAM LISTING

```
      SUBROUTINE MAGTAP(J)
CD* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
CD
CD    PROGRAMMER AND DATE
CD            RICHARD C. THOMAS
CD            TRW SYSTEMS
CD            JUNE 1972
CD
CD    PURPOSE
CD            MAGTAP IS A DUMMY ROUTINE THAT ALLOCATES CORE STORAGE FOR
CD            A CS-1 ASSEMBLY LANGUAGE ROUTINE NAMED MAGTAP. THE CS-1
CD            PROGRAM PEFORMS ALL TAPE OPERATIONS REQUIRED BY THE PRO-
CD            GRAM. FOR TAPE PARITY ERRORS THE MAGTAP ROUTINE ATTEMPTS
CD            RECOVERY A MAXIMUM OF TEN TIMES
CD            FOR UNRECOVERABLE PARITY ERRORS, FRAME COUNT ERRORS AND
CD            TIMING ERRORS THE COMPUTER COMES TO A HALT WITH NO PRINT-
CD            ED ERROR MESSAGES. THE OPERATOR IS ABLE TO DETERMINE THE
CD            NATURE OF THE ERROR BY INDICATOR LIGHTS ON THE TAPE DRIVE
CD
CD    USAGE
CD            CALL MAGTAP(IOPT,NUM,ARRAY,IUNT)
CD
CD    DESCRIPTION OF PARAMETERS
CD
CD       INPUT
CD         CALLING SEQUENCE
CD            IOPT   - .EQ.1 READ TAPE RECORD
CD            IOPT   - .EQ.2 WRITE TAPE RECORD
CD            IOPT   - .EQ.3 BACKSPACE RECORD
CD            IOPT   - .EQ.4 REWIND TAPE
CD            IOPT   - .EQ.5 WRITE FILE MARK
CD            IOPT   - .EQ.6 SPACE FILE FORWARD
CD            IOPT   - .EQ.7 SPACE FILE BACKWARD
CD            NUM    - NUMBER OF WORDS IN DATA RECORD
CD            ARRAY  - DATA RECORD
CD            IUNT   - TAPE UNIT
CD         TAPE
CD
CD       OUTPUT
CD         TAPE
```

PLTGEN PROGRAM LISTING

```
CD
CD    REMARKS AND RESTRICTIONS
CD             WHEN TAPE OPERATION IS NEITHER A READ OR WRITE, NUM AND
CD             ARRAY ARE DUMMY PARAMETERS.
CD             THIS ROUTINE ALLOCATES THE NECESSARY CORE REQUIRED FOR
CD             THE CS-1 PROGRAM WHICH MUST BE LOADED BY PAPER TAPE. SEE
CD             PROGRAM OPERATING INSTRUCTIONS FOR CORRECT PROCEDURES.
CD    SUBROUTINES REQUIRED
CD    NONE
CD
CD* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
      DIMENSION A(300)
      A(1) = 1.0
      RETURN
      END
```

REFERENCES

1. Final Project Report, "Development of Spectral Analysis Math Models and Software Program and Spectral Analyzer - Digital Converter Interface Equipment Design," TRW Technical Report No. 20817-H012-R0-00, dated June 1972.

2. "Spectral Analysis Program, Volume I - User's Guide," TRW Technical Report No. 20817-H010-R0-00, dated June 1972.