Steve Spielman
Product [illegible]

digital

FAST FOURIER TRANSFORM AND DISPLAY

pdp12

DEC-12-FQEA-D

1st Printing February, 1971

The material in this handbook, including but not limited to instruction times and operating speeds, is for information purposes and is subject to change without notice.

TABLE OF CONTENTS

TABLE OF CONTENTS (Cont'd)

ACKNOWLEDGMENT

The PDP-12 Fast Fourier Transform + Display program is an adaptation of a program written by James Rothman, of Digital Equipment Corporation. The algorithm is described briefly in Section 7.0 of this manual and in detail in DECUSCOPE, Volume 72, Number 3, available from DECUS Library, Digital Equipment Corporation, Maynard, Massachusetts.

#####

1.0 INTRODUCTION

The FFTD (Fast Fourier Transform + Display) program can perform a Fast Fourier Transform or Inverse Fast Fourier Transform on 4 to 1Ø24 real or complex points which have been stored on a LAP6-DIAL[1] or data LINC-tape or disk. The real and imaginary parts of the input or output data and the magnitude of the output data may be displayed on the scope via a moving window. Transformed data may also be stored on a DIAL or data LINCtape or disk. In addition, the scale of the displayed data can be user-modified over twelve different ranges.

2.0 MINIMUM HARDWARE REQUIREMENTS

8K PDP-12B with EAE.

3.0 OPERATING PROCEDURE

3.1 Loading FFTD

FFTD is a "load and go" program and is called from tape or disk by the DIAL command:

```
     →LO FFTD, n)
```

where n is the tape (Ø-7) or disk (1Ø-17) containing the program. A DIAL system tape must be on unit Ø. (If a non-existent unit is addressed, NO is displayed on the scope. Press RETURN and issue the proper command.)

At any time during program operation, FFTD may be restarted by pressing the console keys; LINC mode, I/O PRESET, and START 2Ø.

3.2 FFTD Displays

The first display is:

```
DISPLAY 1          SINGLE PRECISION FFT
                   INPUT ON DIAL UNIT?  Y/N__
```

[1] LAP6-DIAL is hereafter referred to as DIAL.

Type Y if the data file is on a tape or disk containing DIAL; type N if the file is on a data tape or disk. (A file copied from paper tape via PIP must be referenced as a data tape or disk.)

The final user replies to all the scope displays are terminated by pressing LINE FEED.

If the input is on a DIAL tape or disk, the second display is:

```
DISPLAY 2          UNIT NUMBER__
                   FILE NAME________
```

Specify the unit number, Ø to 7 for tape, and 1Ø to 17 for disk, where the file is located and press RETURN. Then type the file name, which may be 1 to 8 characters long and must begin with a non-numeric character and not contain a ?, /,\ , or>. After typing the file name, press LINE FEED. Note that a file addressed by name on a DIAL tape or disk can not have a header block and must have been placed on the device only by the FFTD program. If a non-existent unit is requested, NO is displayed. To restart the program from LINCtape, press STOP, I/O PRESET, and START 2Ø. The program must be reloaded from an RK8 or RFØ8 disk.

The user is told if the file is not on the specified unit:

```
DISPLAY 3          CANNOT FIND
                   HIT RETURN TO CONT
```

Press RETURN to bring back display 2

If the input is on a data tape or disk, the second display is:

```
DISPLAY 4          UNIT NUMBER__
                   BLOCK NUMBER___
```

The unit may be any number from Ø to 7 for tape and 1Ø to 17 for disk. The block number must be an octal number from Ø to 777. If a data file with a header block is on a DIAL device, it may be accessed by this sequence (instead of the DIAL message). The correct block number is the value in the DIAL index plus one. After the file has been located, the calculation must be specified.

```
DISPLAY 5            HOW MANY PTS___
                       (4-1Ø24 BY POWERS OF 2)
                     REAL OR
                     COMPLEX?  R/C_
```

Powers of 2, from 2 to 1Ø, are acceptable, permitting 4 to 1Ø24 points. Type R if the data is real; type C if it is complex. (Refer to Section 4.0 for a description of data storage format.) If there is not enough room between the starting block number and the end of tape to hold the number of points specified, display 5 will reappear.

The calculation is further specified:

```
DISPLAY 6            FFT OR DISPLAY?  F/D_
                     TRANSFORM OR
                     INVERSE?  T/I_
```

If the data is just to be displayed, type D and press RETURN. Then type T if the data has most recently been transformed or I if it has not been manipulated at all or has been inversely transformed. Continue at display 7.

The next display is:

```
DISPLAY 7            OUTPUT ON DIAL UNIT?  Y/N_
```

Type Y if output is to a DIAL tape or disk; type N if output is to a data tape or disk.

A reply of Y to display 7 (DIAL tape or disk) causes the display:

```
DISPLAY 8            UNIT NUMBER__
                     FILE NAME________
```

These answers have the same restrictions as the input display, display 2. If there is not enough space on the DIAL tape/disk to hold the output data, the next display is:

```
DISPLAY 9            NO SPACE
                     HIT RETURN TO CONT
```

Press RETURN to bring back display 7.

If a file already exists with the specified name, the next display is:

```
DISPLAY 10          REPLACE?  Y/N_
```

Type Y or N to replace or not to replace the file. A reply of N will cause display 8 to reappear. If the file is to be replaced, but the new file is larger than the old file, display 9 will reappear.

If output is to a data tape or disk, the next display is:

```
DISPLAY 11          UNIT NUMBER__
                    BLK NUMBER___
```

The answers have the same restrictions as the input display, display 4. If there is not enough space from the starting block number to the end of the tape to hold the output data, display 9 will reappear.

The program will now read in the data, perform a Fast Fourier Transform or Inverse Fast Fourier Transform, and write the results as complex data pairs onto the specified tape or disk.

When the transform is completed or if just displays are desired, the following message is displayed:

```
DISPLAY 12          WHICH DISPLAY?
                    R(EAL)
                    I(MAGINARY)
                    M(AGNITUDE)
                    S(CALE FACTOR)
                    LINE FEED (RESTART)
```

Type R, I, M, or S and LINE FEED to obtain the desired display. The scale factor is displayed as a decimal number (Ø-12). (Refer to Section 6.0, Data Scaling, for an explanation of the scale factor.) (The magnitude, M, for a+b is $M = \sqrt{a^2+b^2}$.)

If the display is less than 512 points, it will be stationary and centered on the scope. If it contains 512 or more points, the display can be moved in either direction using A/D knob Ø.

A cursor which can be moved by rotating A/D knob 1 will ride along the curve. Associated with the cursor are four octal words displayed in the top left corner of the scope, one beneath the other. The first two words are the absolute 15-bit core address of the cursor point. The third word is the contents of the displayed core address, i.e., the actual 12-bit value in the data buffer of the data word that corresponds to the cursor point. The fourth word is the scope Y coordinate of the cursor point. The fourth word is a relative value and depends upon the Y scale factor and Y offset. Because the data is scaled to nine bits prior to display, the fourth word or Y coordinate will range from ØØØ1 to 1ØØØ$_8$, where ØØØ1 corresponds to the bottom of the scope and 1ØØØ to the top.

The curve can be expanded in the Y direction by typing a 1 or decreased by typing Q. Twelve different ranges are possible. As the display is enlarged, no check is made against losing significant digits of large values because the user may wish to expand small features of the display. Therefore, as the display is enlarged, large values may suddenly decrease in size as significant digits are lost.

The magnitude display is shown at half scale initially. If the values allow, the number 1 can be typed once to show the display at full scale.

Pressing RETURN will cause display 12 to reappear. As many displays as desired may be requested. Subsequent displays will be initially shown at the same range as the preceding display. Pressing LINE FEED without entering a character will cause display 1 to reappear.

4.0 EXAMPLE

This section provides examples of the displays which result from a transform performed on a square wave of 512 points and from an inverse transform performed on the resulting coefficients.

4.1 Input Display

Consider a square wave[1] of 512 real points which has the following format on tape or disk:

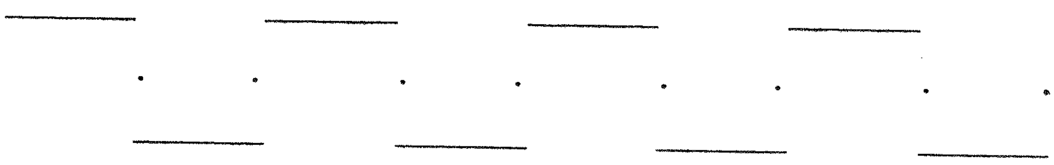
Address	Value	
Ø	2ØØØ	77 points
	⋮	
77	1ØØØ	
1ØØ	ØØØØ	77 points
	⋮	
177	1ØØØ	

[1] The displays shown on the following pages are adaptations and are for demonstration purposes only.

```
Address    Value
  2ØØ       2ØØØ
                     > 77 points
  277       1ØØØ
  300       ØØØØ
                     > 77 points
  377       1ØØØ
  4ØØ       2ØØØ
                     > 77 points
  477       1ØØØ
  5ØØ       ØØØØ
                     > 77 points
  577       1ØØØ
  6ØØ       2ØØØ
                     > 77 points
  677       1ØØØ
  7ØØ       ØØØØ
                     > 77 points
  777       1ØØØ
```

If the input is displayed, there will only be a REAL display. It will look as follows, assuming the cursor is to the extreme left and the display is not moving.

ØØØ1
ØØØØ
2ØØØ
Ø6Ø1

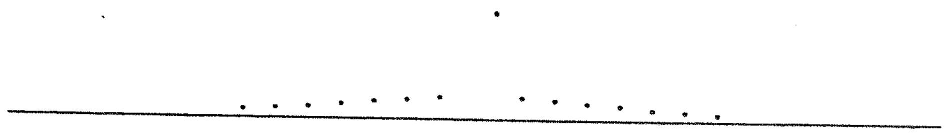


The first two values in the upper left hand corner are the address of the point on which the cursor is resting. When the cursor is at the extreme left, it indicates location ØØØØ of field 1. The third value is the contents of that memory location, in this case, 2ØØØ. The fourth value is the position of the cursor with respect to the bottom of the screen. [1 = bottom, 4Ø1 = X axis (middle), 1ØØØ = top.]

4.2 Transform Displays

4.2.1 Real Display

ØØØ1
2ØØØ
ØØØØ
Ø4Ø1



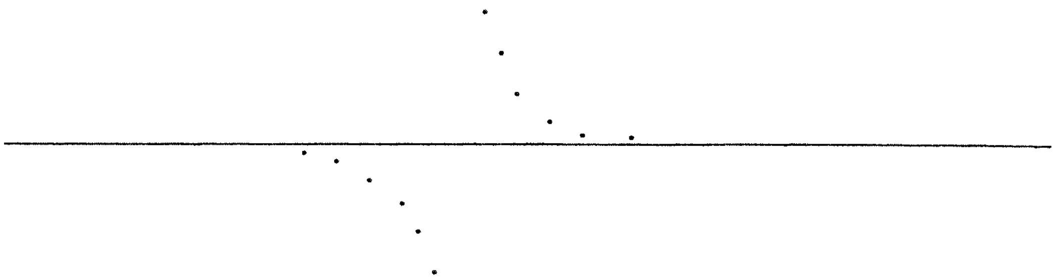
Moving the cursor to the highest point in the display will change the value display to:

```
ØØØ1
24ØØ
2ØØØ
Ø6Ø1
```

This is the DC component of the wave.

4.2.2 Imaginary Display

```
ØØØ1
2ØØØ
ØØØØ
Ø4Ø1
```



Moving the cursor to the lowest point produces the values:

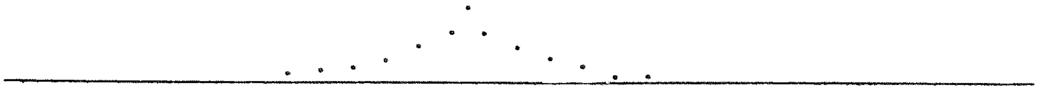
```
ØØØ1
2374
6567
Ø257
```

Moving the cursor to the highest point displays:

```
ØØØ1
24Ø4
1214
Ø522
```

4.2.3 Magnitude Display

```
ØØØ1
2ØØØ
ØØØØ
Ø4Ø1
```



Moving the cursor to the highest point gives the following display:

```
ØØØ1
24ØØ
1ØØØ
Ø5Ø1
```

Because the magnitude of maximum values causes overflow, a factor of 2 is removed during computation. Therefore, the values displayed are half scale; type the key "1" once to display the magnitude at full scale.

4.2.4 Scale Factor Display

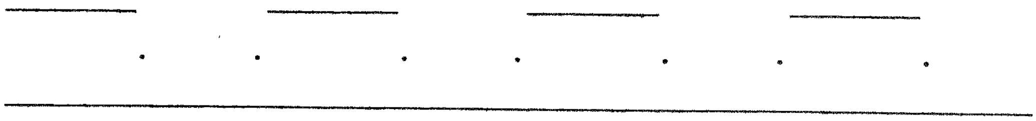
The scale factor has a value of 1. To obtain the actual coefficients, rest the cursor on the desired point and shift right the third value of the corner display the number of bits equal to the scale factor. In this example, the highest value of the real display is 2ØØØ. Shifting it right by the scale factor (=1) yields 1ØØØ, the actual value of the DC component, which in binary is 001 000 000 000. Because the binary point is to the right of the sign bit, the actual value is $+.01_2$.

4.3 Inverse Transform Displays

The output of the transfer was 512 complex points. The inverse yields the following displays:

4.3.1 Real Display

ØØØ1
ØØØØ
Ø764
Ø477



The third value, Ø764, is a deviation from 1ØØØ, the exact value. At this time there are 2 scale factors involved. The relationship between the computed results and the original data is:

$$\text{results} = [(\text{original data})*2^{\text{sum of scale factors}}]/\#\text{ of points}$$

Reducing the equation for the first point yields:

$$1\emptyset\emptyset\emptyset_8 = [(2\emptyset\emptyset\emptyset_8)*2^8]/1\emptyset\emptyset\emptyset_8$$

$$2^9 = 2^{1\emptyset}*2^8/2^9$$

$$=2^9$$

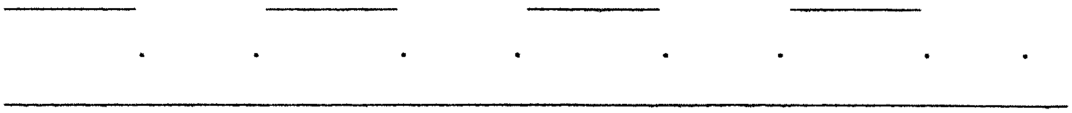
4.3.2 Imaginary Display

```
ØØØ1
1ØØØ
ØØØ7
Ø4Ø1
```

The values are very small and are the result of imprecision in the computations.

4.3.3 Magnitude Display

```
ØØØ1
2ØØØ
Ø372
Ø44Ø
```



As in the magnitude display of the transform, the values displayed are half scale. Because the imaginary components are essentially zero, the magnitude, when doubled, equals the real values.

4.3.4 Scale Factor Display

The scale factor has a value of 7.

5.0 DATA STORAGE

The data must be stored sequentially on tape or disk in a binary file starting at the beginning of a block. If the data is complex, the real parts are grouped together followed by the imaginary parts, if any. If there are none, the program will create imaginary parts of value zero. The input and output data are in the form of binary fractions. For output data, the location following the last imaginary part contains the scale factor (refer to Data Scaling, Section 6.0). A file of complex values are stored in the following format:

Ø	$REAL_1$	
	$REAL_2$	
	↓	
	$REAL_n$	
n	$IMAG_1$	
	$IMAG_2$	
	↓	
	$IMAG_n$	
2n	SCALE FACTOR	-only present if file is generated by the FFTD program.

6.0 DATA SCALING

All calculations in FFTD are done with single precision fixed point signed binary fractions. The binary point is located between bit Ø and bit 1, leaving an 11 bit signed mantissa. Bit Ø is used as a sign bit. Negative numbers are formed by taking the two's complement of the positive binary fraction, so all inputs must be scaled in magnitude to less than one. The outputs are also formatted as above.

In order to preserve precision, it is sometimes necessary to divide by 2 in a computation. As a result, a pseudo floating point format has been adopted in which a variable scale factor (or exponent) is imposed on all the Fourier coefficients. This scale factor or pseudo exponent is found in item SCAL after each transform has been completed. It is also stored after the last imaginary part on tape or disk. The values stored on tape or disk are the Fourier coefficients multiplied by 2^{SCAL}. Because in binary notation shifting a number right one bit is equivalent to dividing by two, to retrieve the coefficients themselves, shift each number right by the number of bits equal to the value of the scale factor. In the case of the inverse transform, the time samples are the values in memory multiplied by 2^{-SCAL}. If, however, the inverse transform was performed on normalized transform data, the results are equal to $([(\text{original data})*2^n]/\text{no. of points})$ where n equals the sum of both scale factors. To retrieve the time samples, shift left each number by the value of the scale factor.

7.0 SUBROUTINES USED

Manipulation of the DIAL and data LINCtapes and disk is done using the program MILDRED (DEC-12-FZDA). The question and answer displays are handled by QANDA (DEC-12-FISA). The data displays are handled by DISPLAY

(DEC-12-FLSA). A modification of FFTS-C (DECUS #8-144) is used to perform the Fourier Transforms.

8.0 ALGORITHM DESCRIPTION

The Fast Fourier Transformation enables computation of the power spectrum of a time series in a minimum of time. Specifically, it permits the discrete Fourier transformation

$$S_j=\frac{1}{N}\left[\sum_{k=\O}^{N-1} x_k e^{-2\pi ijk/N}\right] \qquad \begin{array}{l} j=\O,.,.,\ldots,N-1 \\ i=\sqrt{-1} \end{array}$$

of a series on N equally spaced time samples (where N is a power of 2). The time required is proportional to $N_2\log_2N$, whereas previous methods required times proportional to N. This gives a reduction in computation time of $1-\log_2N/N$ or over 99 percent for N=1Ø24. The algorithm makes use of the fact that

$$W^k=W^{(k\bmod N)} \text{ (where } W=e^{-2\pi i/N}\text{)}$$

to reduce the number of manipulations necessary for a transformation.

9.0 CORE CHART

```
     Field Ø

SEGMENT Ø
        PAGE Ø - IFFT
        *4ØØ   - FFT
        *14ØØ  - DISPLAY
SEGMENT 1      - MILDRED
SEGMENT 2      - MONITOR
                 QANDA
SEGMENT 3      - Data display code
                 FDV table
                 RWPARM table
                 Questions
                 Sine Table
     Field 1
        Ø      - Buffer - real parts
        2ØØØ   - Buffer - imaginary parts
```

10.0 PROGRAM REGION DESCRIPTION

10.1 Routines

IFFT - Take the Inverse Fourier Transformation of the data in field 1. The results are in bit inverted order (refer to the SORTX routine).

FFT - Take the Fourier Transformation of the data in field 1. The results are in bit inverted order (refer to the SORTX routine).

SORTX - Sort the data from bit inverted order to sequential order. Bit inversion means simply the process of re-ordering the bits in a binary number. For instance, the binary number 001 bit inverted is just 1ØØ (=4). For example, to locate S_5 in memory for a 16 point transformation (N=16, n=4), write 5 as a binary number of n=4 bits, $5_{1Ø}=Ø1Ø1_2$. Then reverse the order of these bits to $1Ø1Ø_2$. This means S_5 is stored in position 1Ø. Physically, then, S_5 of the real parts is to be found in location XRTAB+9.

MULTIP - Perform a rounded single precision signed multiply using EAE. The CAL+1 contains the address of the multiplicand. The AC contains the multiplier. Exit with the product in the AC.

INVRT - Reverse the bits of the number contained in the AC.

TRIGET - Fetch sine and cosine values. Specifically, if the AC=K on entry, the values of sin ($2\pi K/N$) and cos ($2\pi K/N$) are fetched from an internal trig table. K must be ≥N/2. A register COSINE contains the cosine value and the AC contains the sine value on exit.

ADDR - Perform a single precision add with rounding.

IDORA - This subroutine generates a moving window display with a cursor riding on the curve. For more information refer to the DISPLAY document, DEC-12-FLSA-D.

IFDIAL - Display the question: FROM DIAL UNIT? Y/N_ If the answer is Y, jump to UNTFIL; if N, jump to DATTAP; if neither, redisplay the question.

UNTFIL - Jump to the subroutine ASK2 to display:

```
UNIT NUMBER__
FILE NAME________
```

If the unit number is illegal, jump to ASK2 again to redisplay the question. If legal, jump to LOOKUP with the address of the File Description Vector (hereafter referred to as FDV) parameter list in the AC. If the file cannot be found, display the message:

```
CANNOT FIND
HIT RETURN TO CONT
```

When RETURN is hit, jump back to UNTFIL. If the file is found, jump to MOVINP.

DATTAP - Jump to the subroutine ASK3 to display:

```
UNIT NUMBER__
BLK NUMBER___
```

If an illegal value is entered, jump back to DATTAP. If all the input is legal, fall through to MOVINP.

MOVINP - Jump to FDV2RW to move the input information from the FDV to the read/write parameter list. Fall through to PTS.

PTS - Display:

```
NUMBER OF PTS____
   (4-1Ø24 BY POWERS OF 2)
REAL OR
COMPLEX?  R/C_
```

Set B1 to the address of the answer buffer, MPLIER to 12 and UPLEGL to -71 (-9) because the number of points is entered as a decimal value. Set the AC to the largest legal value, 2ØØØ, and jump to CONV. If the answer is an illegal value jump back to PTS; store the value in N and store its 1's complement in TEMP1. Since the number of points must be an integral power of 2, only one bit in TEMP1 may be set. Bit 11 is the exception to one bit being a power of 2. Check bit 11 first, then rotate the value adding up the number of bits set. If the total is not 1, jump back to PTS. Otherwise fall through to ROT1.

ROT1 - Compute the power of 2 by rotating right the value in TEMP1 and stepping B2 until the bit that is set is encountered in bit 11. Fall through to STAMU.

STAMU - Store the power of 2 in NU. If the power is less than 2, jump back to PTS. Otherwise load the AC with the number of points*2 and jump to NUMBKS to compute the number of blocks needed to hold the output. Store the value in FDV+7. Store it also in RWPARM+3 since, for complex data, the input and output data consist of the same number of blocks. If the answer to the second question is not R, jump to IFCOM. If it is R, the input consists of half as many words as the output. Load the AC with the value of N and jump to NUMBKS to compute the number of input blocks. Store the value in RWPARM+3. Set REALFG and jump to CKEND.

IFCOM - If the answer is C, clear REALFG and fall through to CKEND. Otherwise jump back to PTS to redisplay the question.

CKEND - If there is not enough room between the starting block number and the end of tape to hold the number of points specified, jump back to PTS. If the number of output words is 4ØØ or greater, another block will be needed to hold the scale factor. Increment FDV+7. Fall through to IFFFT.

IFFFT - Display: FFT OR DISPLAY? F/D_
TRANSFORM OR
INVERSE? T/I_

If the answer to the first question is D, set DISFLG to indicate that the data will only be displayed. If F, clear DISFLG to indicate that a Transform or Inverse Transform will be performed. If the answer to the second question is T, clear FTFLG; if I, set it. If DISFLG is set, jump to DISPLY to display the data. Otherwise, jump to OUTQES.

OUTQES - Display the question: OUTPUT ON DIAL UNIT? Y/N_
If the answer is Y jump to OUTUNT; if N jump to ONDAT; otherwise redisplay the question.

OUTUNT - Jump to the subroutine ASK2 to display:

UNIT NUMBER__
FILE NAME________

If an illegal value is input, redisplay the question. Otherwise jump to ENTER with the address of the parameter list in the AC. If a file with the specified name already exists, jump to SAMNAM. If there is not enough space to hold the output data, jump to NOSPAC. If it is a new file and there is enough space to hold it, fall through to RDDATA.

RDDATA - Clear 4ØØØ words of field 1 and read in the input data. If REALFG is Ø, the data is complex - move the imaginary parts to start at location 2ØØØ. If it is non-zero, the data is real and nothing need be done. Jump to PROC.

PROC - If IFTFLG is Ø, jump to FT to do a Transform. Otherwise, fall through to do an Inverse Transform.

IFT - Jump to the subroutine IFFT to do an Inverse Transform on the input data. Then jump to the subroutine SORTX to sort the coefficients into sequential order from bit inverted order. Jump to STSCAL to store the scale factor which is equal to NU-SCAL. The data should be shifted by this value.

FT - Jump to the subroutine FFT to transform the input data. Then jump to the subroutine SORTX to sort the coefficients into sequential order from bit inverted order. The scale factor is the value in SCAL and equals the number of bits by which the data should be shifted right. Fall through to STSCAL.

STSCAL - Store the scale factor in the word following the last imaginary part. Move the imaginary parts from 2ØØØ to immediately behind the real parts.

NOWSTR - Jump to the subroutine FDV2RW to move the output parameters from the FDV to the read/write parameter list. Write the data onto the output tape and jump to DISPLY.

NOSPAC - Jump to the subroutine ASK to display the message:

```
NO SPACE
HIT RETURN TO CONT
```

When RETURN is hit, jump to OUTQES.

SAMNAM - Jump to the subroutine ASK to display:

```
ALREADY EXISTS
REPLACE?  Y/N_
```

If the answer is Y, jump to REPL; if it is N, jump to OUTUNT. If it is neither, redisplay the question.

REPL - Try to replace the existing file with the new file. If the new file is longer, jump to NOSPAC. If the replacement is successful, jump to RDDATA.

ONDAT - Jump to the subroutine ASK3 to display:

```
UNIT NUMBER__
BLK NUMBER___
```

If an illegal value is entered, redisplay the question. If there is not enough space between the specified block number and the end of tape to hold the output data, jump to NOSPAC. Otherwise, jump to RDDATA.

10.2 Subroutines

FDV2RW - Transfer the unit number, starting block number, and number of blocks from the FDV parameter list to the READ/WRITE parameter list.

NUMBKS - Enter with the number of words in the AC. Convert this value to blocks by counting the number of times 4ØØ can be subtracted from it before the value becomes negative. Return with the number of blocks in the AC.

ASK2 - Jump to OCTL to set MPLIER to 1Ø and UPLEGL to -67(-7) because the unit number is input as an octal number.

```
Display:  UNIT NUMBER__
          FILE NAME________
```

by jumping to the subroutine ASK with the address of QUES2 in the AC. Set B1 to the address of the answer buffer and jump to the subroutine CONV with the largest legal unit number, 17, in the AC. If the value is illegal, return to CALL+1. If legal, store it and the file name in the FDV parameter list. Fill the file name out to 8 characters with 77's. Return to CALL+2.

ASK3 - Display: UNIT NUMBER__
BLK NUMBER________

by jumping to the subroutine ASK with the address of QUES3 in the AC. Set B1 to the address of the answer buffer and jump to OCTL to set MPLIER to 1Ø and UPLEGL to -67(7) because the unit and block numbers are input in octal. Jump to subroutine CONV with the largest legal unit number, 17, in the AC. If the value is illegal, return to CALL+1. Otherwise, store it in word Ø of the FDV parameter list. B1 is now pointing to the block number. Jump to CONV with the largest legal block number, 777, in the AC. If the value is illegal, return to CALL+1. If legal, store it in word 6 of the FDV parameter list. Return to CALL+2.

CONV - CONV is entered with the largest legal value in the AC and B1 pointing to the address - (1 half word) of the first character to be converted. Store the 1's complement of the largest legal value in TEMP2 and clear TEMP1. UPLEGL contains a -71(-9) or -67(-7) and MPLIER contains a 1Ø or 12 depending on whether the number to be converted is in decimal or octal. Extract a character and compare it against an ASCII Ø and the contents of UPLEGL. If it is a legal value, jump to MULPLY which will multiply the value in TEMP1 by the contents of MPLIER and add the digit being converted to it. Repeat the procedure until a character is found which is not between Ø and UPLEGL. If it is not a 34, 74, or Ø, it is an illegal character: return to CALL+1. A 34 or 74 indicates the end of the input field; a Ø indicates the end of the input. Compare the converted value in TEMP1 against the maximum legal value in TEMP2. If the value is legal return to CALL+2; otherwise return to CALL+1.

OCTL - OCTL sets MPLIER to 1Ø and UPLEGL to -67(-7) so that CONV will convert an octal number.

ASK - ASK is entered with the address of the display in the AC. Store it in the parameter list and jump to QAINIT to display the message. Refresh the display until the answer is input. Return to the calling routine.

DISPLY - This region is entered either after the Transform or Inverse Transform is completed or in response to a D in answer to the display: FFT OR DISPLAY? F/D_. Since the data is manipulated in preparation for each display it must be read in before each display. After reading in the data, display:

```
WHICH DISPLAY?
R(EAL)
I(MAGINARY)
M(AGNITUDE)
S(CALE FACTOR)
LINE FEED (RESTART)
```

If the answer buffer contained Ø, just LINE FEED was hit: jump to IFDIAL to restart the program. Otherwise jump to WCHDIS.

WCHDIS - Jump to DPIMAG, DPMAG, DPREAL, or DPSCAL if the answer was I, M, R, or S,respectively. Otherwise redisplay the question.

DPIMAG - If REALFG is non-zero, the input is real and no Transform was performed. Therefore, there are no imaginary parts to display; redisplay the question. If REALFG is zero, check IFTFLG. If it equals zero, either an Inverse Transform was performed or the original data is just being displayed. In either case the data is in the right order. If IFTFLG is non-zero, a transform was performed. The positive half of the curve is first followed by the negative half and the signs are reversed. Swap the halves and reverse signs before jumping to PREPAR.

DPREAL - Check IFTFLG for the same reason as in DPIMAG. The only difference is that the signs of the real parts are not reversed.

PREPAR - If less than 1ØØØ points are to be displayed, the display will not move and the points displayed will be centered on the scope. To achieve this, LEFTX is set to the 1's complement of -1ØØØ+(1ØØØ=# of points)/2,MINPTS to the 2's complement of the number of points, and MVDIS to the instruction CLR. Jump to SHOWIT.

GQ1ØØØ - If 1ØØØ or more points are to be displayed, the display will fill the scope and will move. To achieve this, LEFTX is set to the 1's complement of 1ØØØ, MINPTS to the 2's complement of 1ØØØ and MVDIS to the instruction SCR 4. Fall through to SHOWIT.

SHOWIT - Jump to the subroutine IDORA to display the data. The six parameters following the call to IDORA are in order: the memory field of the lower address, the lower address, the memory field of the higher address, the higher address, the Y offset of the display and the scale factor of the data. Both fields are always 1, the lower address is always Ø. The higher address is set in the region DISPLY. The Y offset is always Ø; therefore the baseline is half way up the scope. The scale factor is the instruction SCR plus the number of bits to scale the data right before displaying it. Since IDORA displays only the right nine bits, if the left three bits are significant, the data must be scaled right three before displaying it.

RFRSH - Jump to RDORA to refresh repeatedly the display until a key on the teletype is hit. If the RETURN is hit, jump to REDPLY which jumps to DISPLY to redisplay the question: WHICH DISPLAY? If a 1 is entered, jump to LARGER to blow up the display. If a Q is hit, jump to SMALLR to decrease its size. If anything else is entered, ignore it.

SMALLR - If the instruction at SIZE contains a shift of 11 bits, a bigger shift would be meaningless. Jump back to RFRSH. Otherwise, increment the value of the shift and jump to SHOWIT.

LARGER - If the instruction at SIZE contains a shift of Ø bits,jump back to RFRSH. Otherwise decrement the value of the shift and jump to SHOWIT.

DPSCAL - If REALFG is non-zero, only real parts are present, meaning this program did not create the file and therefore there is no scale factor. Return to DISPLY to redisplay the question. If REALFG is Ø, the scale factor is stored after the last imaginary part. Convert it to ASCII decimal and display it.

DPMAG - If REALFG is non-zero, the input data is real and no transform was performed; therefore the magnitude is the same as the real points. Redisplay the question: WHICH DISPLAY? Otherwise move the imaginary parts to location 2ØØØ. Set RELPTR and IMGPTR, which contain the effective address of the multipliers, to 6ØØØ since the data begins at location Ø of their respective segments and is fractional. Fall through to NXTMAG.

NXTMAG - Square a real part and store it. Square the imaginary part, add the square of the real part to it, jump to the subroutine SQRT to get the square root of the sum and store it in place of the real part. Repeat the process for each point. Then jump to SHOWIT to display the magnitude.

MOVPTS - The subroutine MOVPTS moves values from one buffer (address -1 in 1Ø) in field 1 to another (address -1 in 11). If CMPFLG equals 1, the values are complemented as they are moved. TEMPR contains the 2's complement of the number of values to move.

MVRLMG - The subroutine MVRLMG is used to swap the first and second halves of the real or magnitude values. In the process they are moved from the buffer starting at location Ø to the one starting at 2ØØØ.

FDV - The File Descriptor Vector parameter list is used by the LOOKUP, ENTER, and REPLACE sections of MILDRED. Word Ø contains the unit number, words 1-4 contain the file name, word 5 contains a 2 indicating the file is binary, word 6 is the starting block number, and word 7 is the number of blocks. Word 6 is filled by LOOKUP, ENTER and REPLACE. Word 7 is filled by LOOKUP but must be supplied for ENTER and REPLACE.

RWPARM - The Read/Write parameter list is used by the READ and WRITE sections of MILDRED. Bits Ø-2 of word Ø contain the field, bits 9-11 contain the unit. Word Ø contains the starting address, word 1 the starting tape block number and word 2 the number of blocks.

SQRT - The subroutine SQRT is entered with a value in the double precision location DPSQ. It returns with the square root in the AC.

10.3 Symbols

N	Number of words in computation
NU	Power of 2 of value of N
L	Index to show what array is being constructed
S	Gives spacing between node pairs in the Lth array
NOVER4	Storage for N/4
MAXNU	Power of 2 of largest table size (13)
MNOVR2	Storage for N/2
QR	Pointer to real part of X(Q)
QI	Pointer to imaginary part of X(Q)
PR	Pointer to real part of X(P)
PI	Pointer to imaginary part of X(P)
Q	Numerical index Q (=Ø,1,...,N-1)
P	Numerical index P (=Ø,.,...,N-1)
K	Number in the node being operated on
C	Interrupts computation of Lth array every S passes
ADD2	Used by subroutine ADDR as data (addend) Used by monitor as a temporary location
TEMPR	Temporary storage register for real parts Used by monitor as a temporary location
SINE	Temporary storage for sin (S*PI*K/N) Used by monitor as a temporary location
COSINE	Temporary storage for cos (2*PI*K/N) Used by monitor as a temporary location
GR	Real part of product (W^k*X(P)) - temporary storage Used by monitor as a temporary location
GI	Imaginary part of product (W^k*X(P)) - temporary storage
SCAL	Pseudo exponent of Fourier coefficients
SHFLAG	If =1, add with shift; if =Ø, add without shift
SHFCHK	Indicates if all X's in an iteration are <.5
DISFLG	If ≠Ø, the data will just be displayed
IFTFLG	If ≠Ø, an Inverse Transform was performed
REALFG	If ≠Ø, the data does not contain imaginary parts
DPSQ	Used to save the double precision squares of the real and imaginary parts during calculation of the magnitude.
CMPFLG	If =1, the subroutine MOVPTS will complement the values as it moves them

10.4 Beta Registers

Beta registers 1, 2, and 3 are used by the monitor in ASK2 and ASK3 as temporary pointers and counters. QANDA and MILDRED make more extensive use of the Beta registers.

11.0 ASSEMBLY INSTRUCTIONS

The FFTD program is assembled in three sections by assembling and saving each, then adding them together. The entire command sequence is:

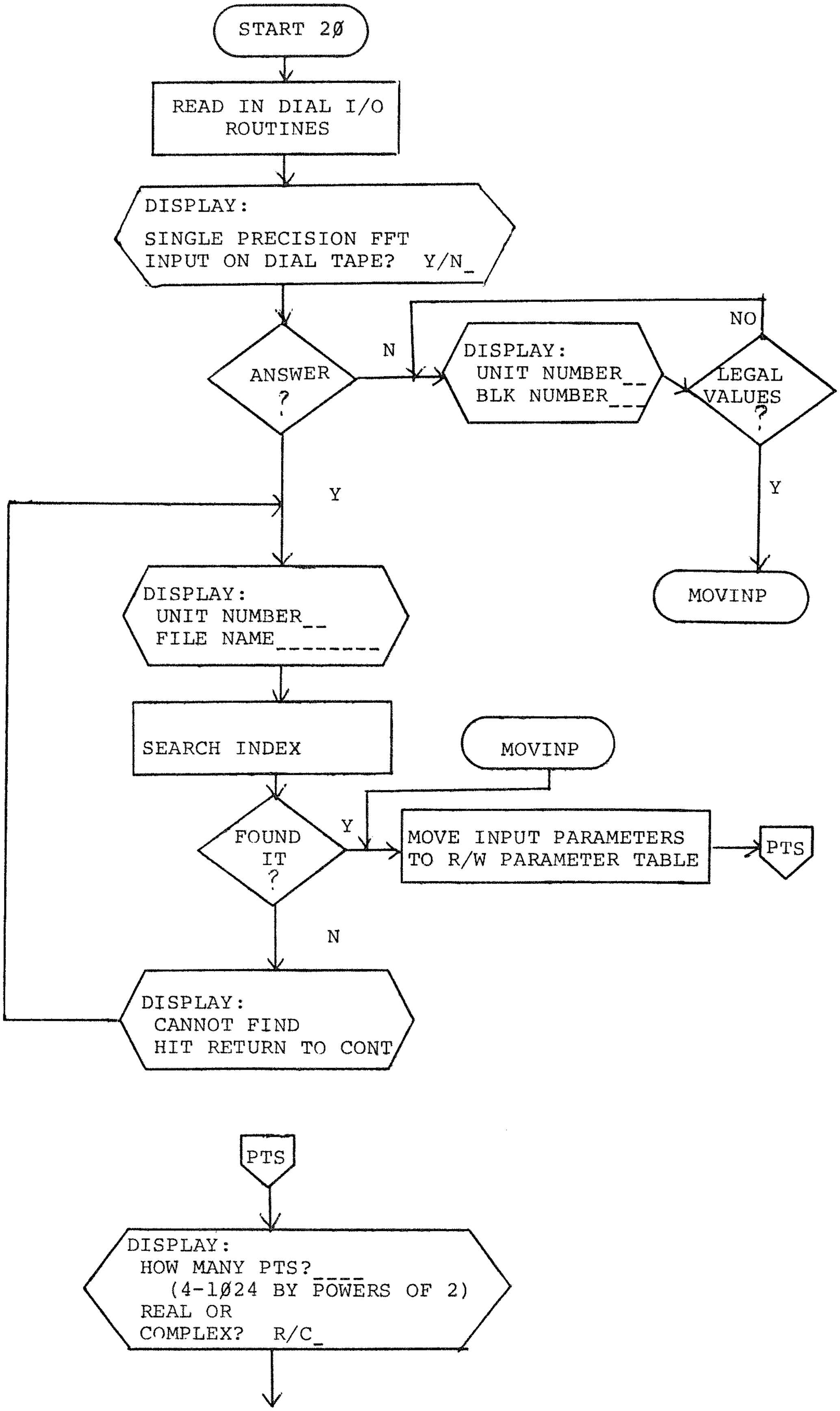
```
→AS MILQAN,n↵          where n is the unit
→SB MILQAN,n↵          containing the program
→AS SIN256,n↵
→SB SIN256,n↵
→AS FFTC-1↵            (FFTC-1 chains to FFTC-2)
→SB FFTC-1↵
→ZE↵
→AB MILQAN,n↵
→AB SIN256,n↵
→AB FFTC-1,n↵
→SB FFTD,n,L↵          (saves the whole program)
```

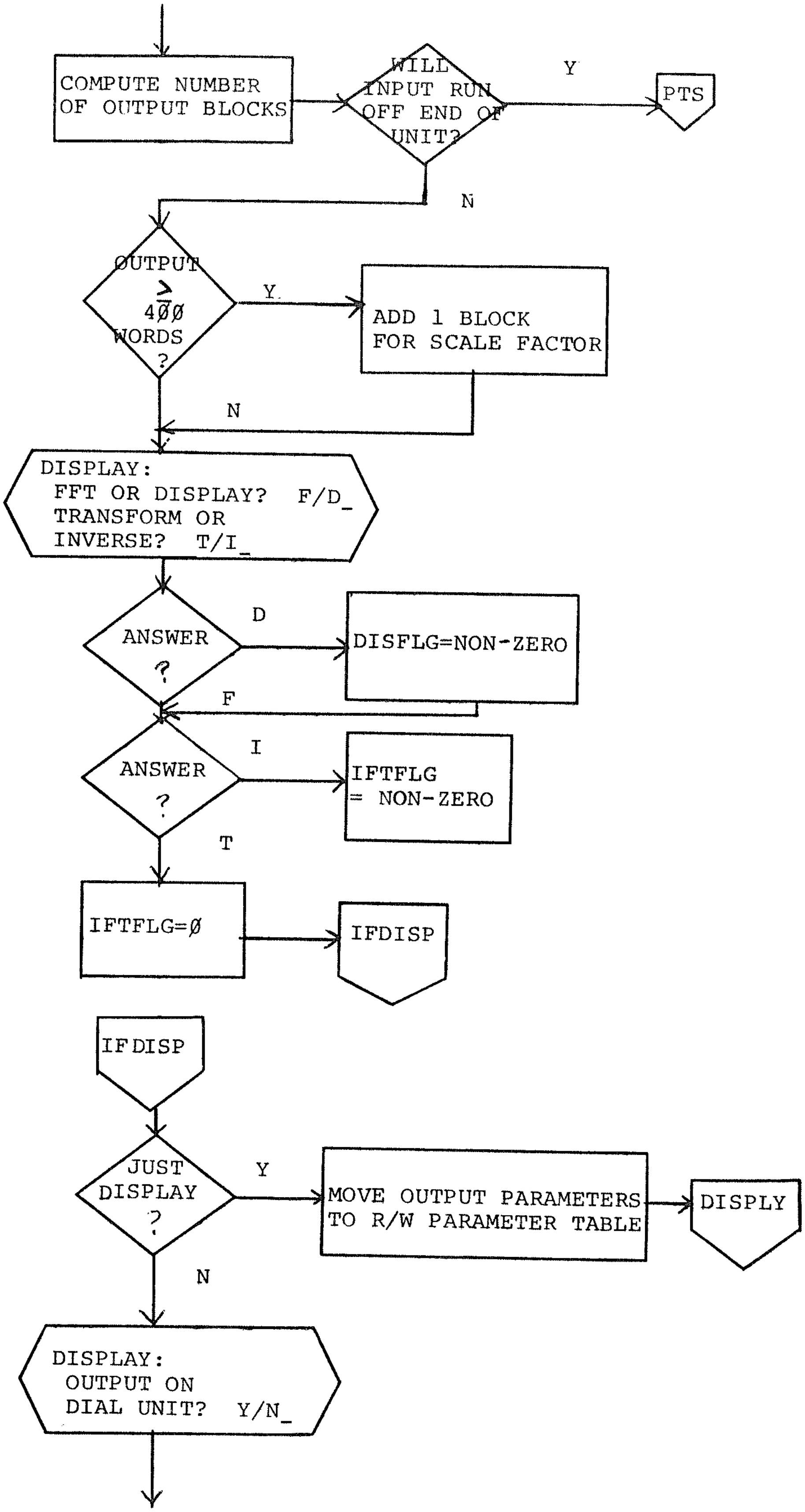
12.0 SYSTEM FLOWCHARTS

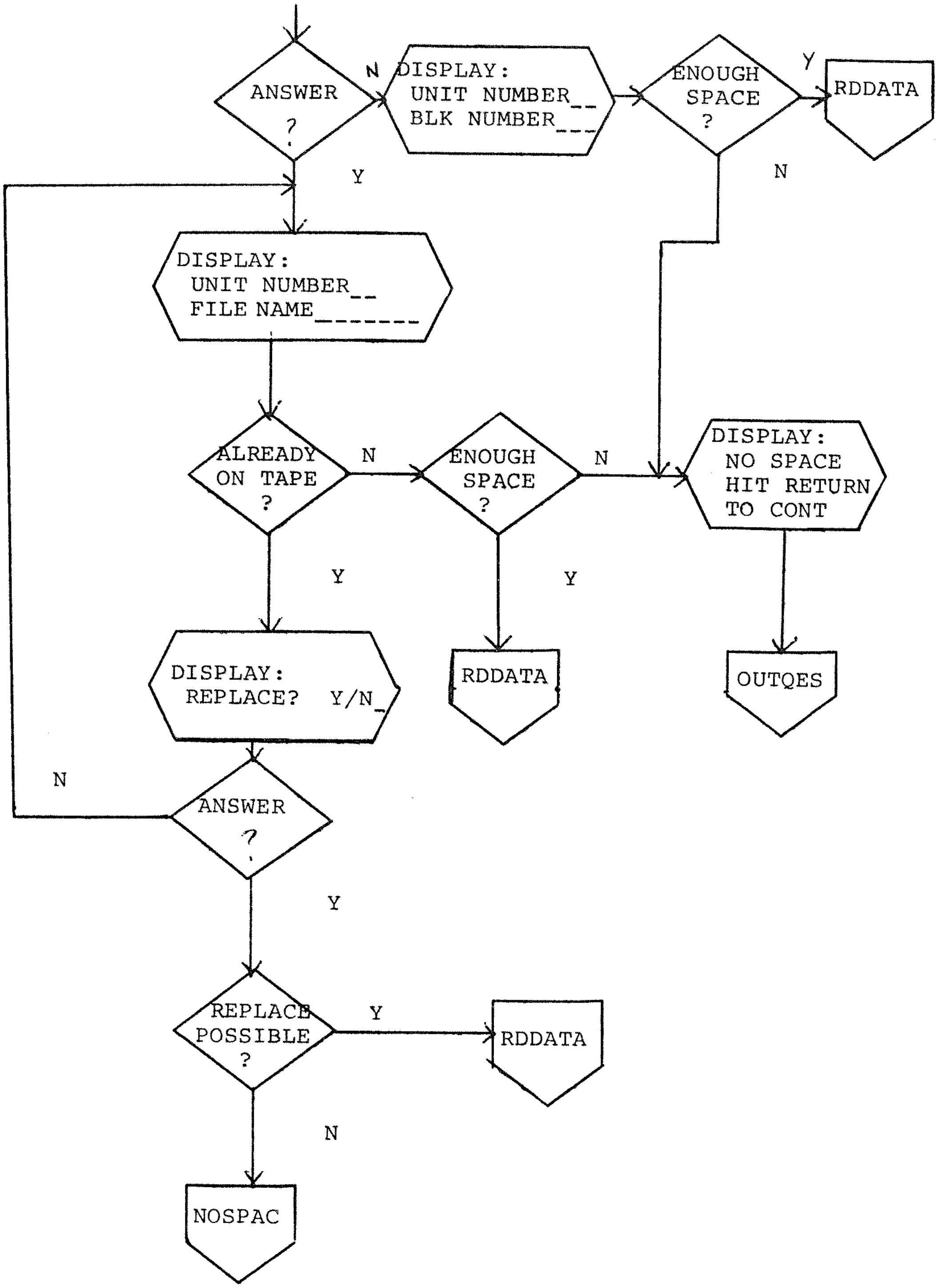
(Attached)

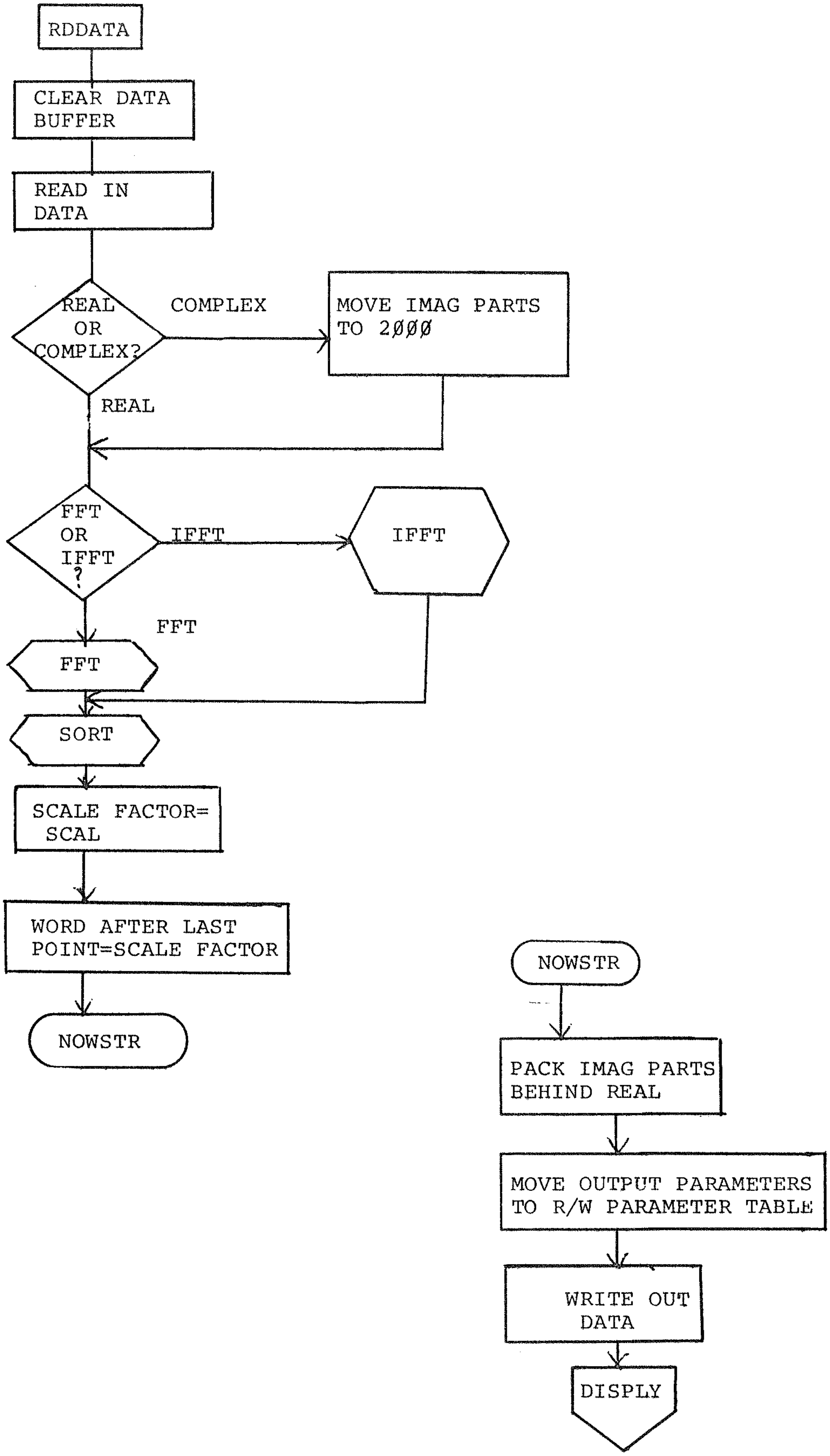
13.0 PROGRAM LISTING

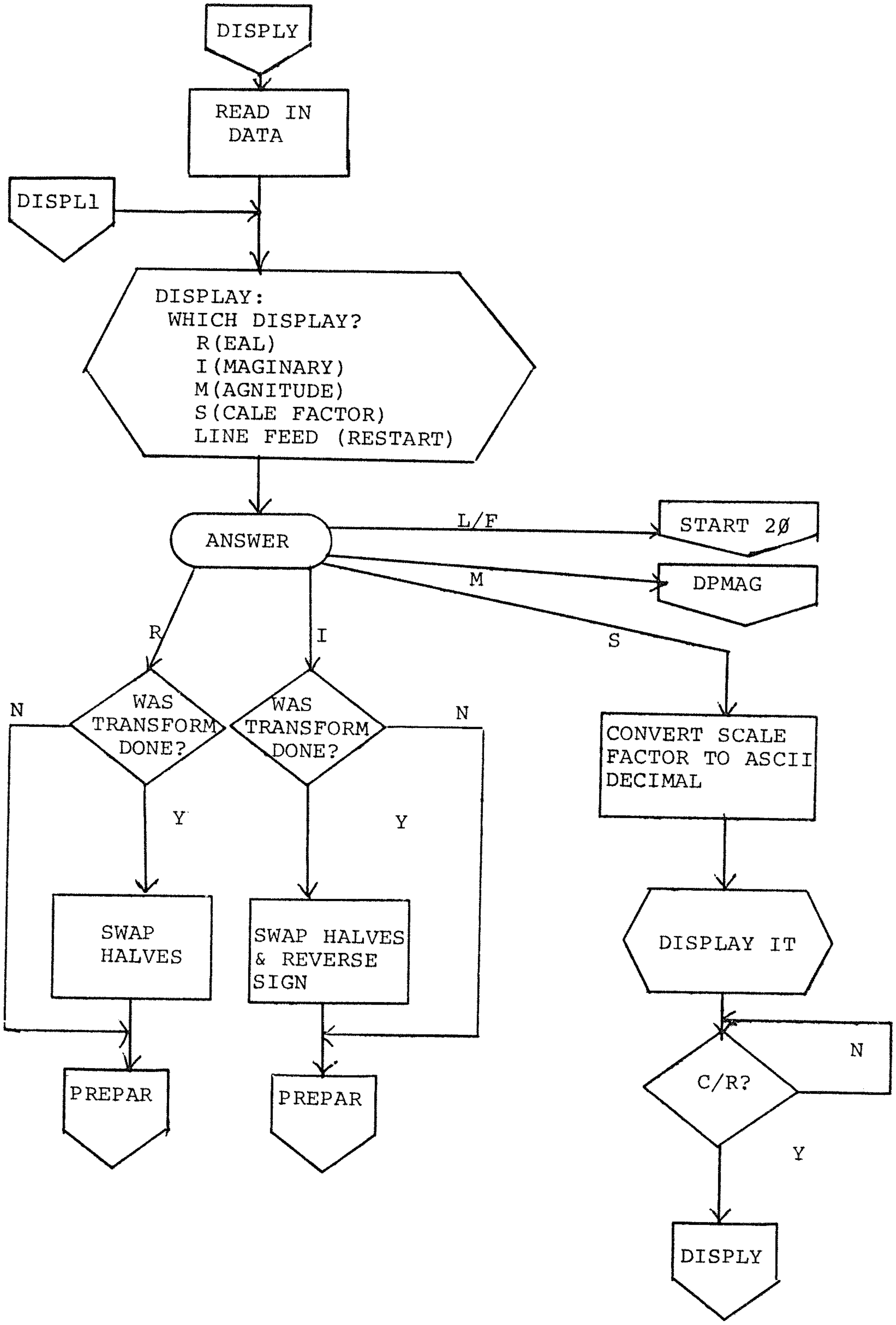
(Attached)

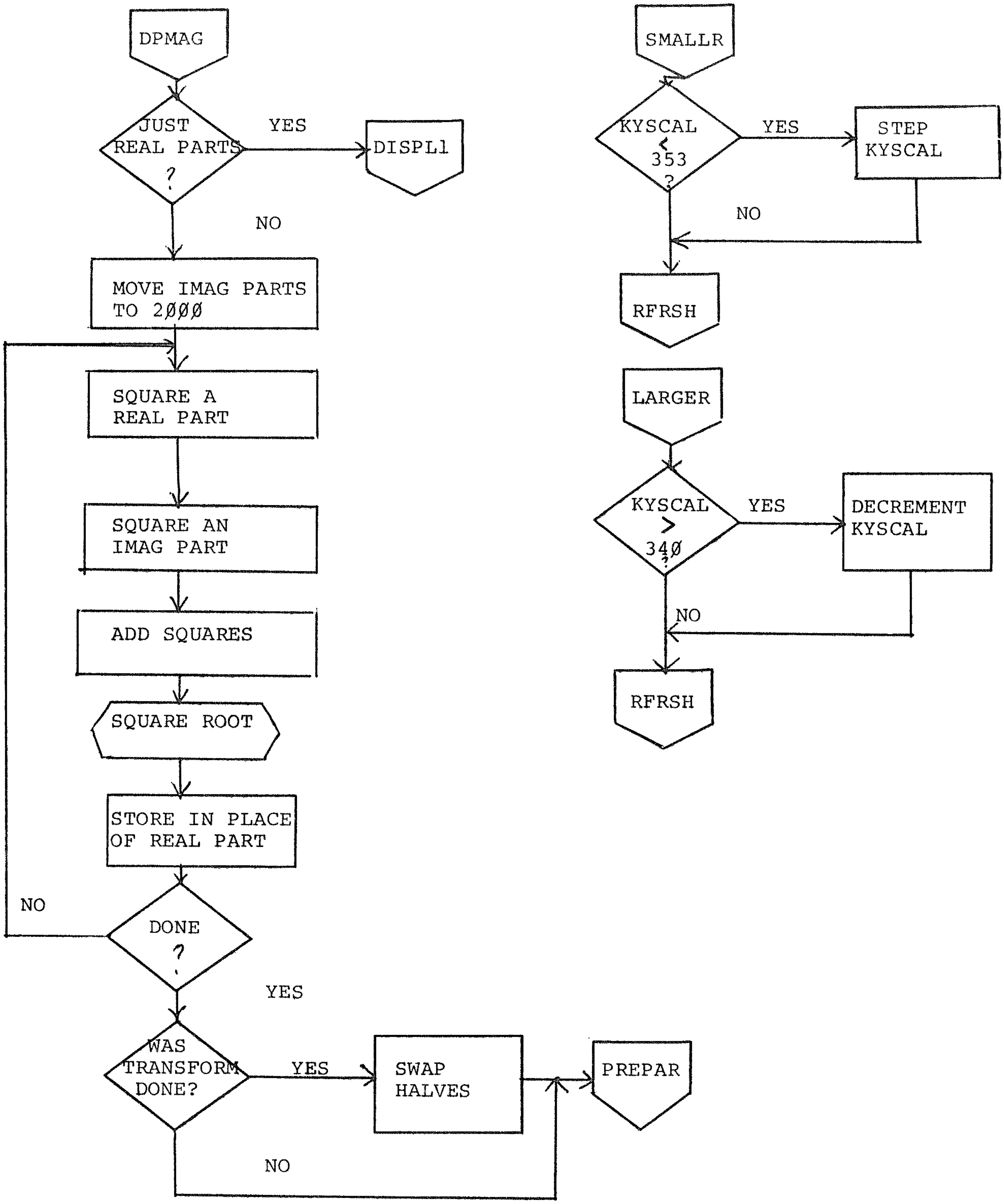

START 2Ø
READ IN DIAL I/O ROUTINES
DISPLAY:
SINGLE PRECISION FFT INPUT ON DIAL TAPE? Y/N_
ANSWER ?
N
DISPLAY:
UNIT NUMBER__
BLK NUMBER__
LEGAL VALUES ?
NO
Y
MOVINP
Y
DISPLAY:
UNIT NUMBER__
FILE NAME_________
SEARCH INDEX
MOVINP
FOUND IT ?
Y
MOVE INPUT PARAMETERS TO R/W PARAMETER TABLE
PTS
N
DISPLAY:
CANNOT FIND
HIT RETURN TO CONT
PTS
DISPLAY:
HOW MANY PTS?____
(4-1Ø24 BY POWERS OF 2)
REAL OR
COMPLEX? R/C_

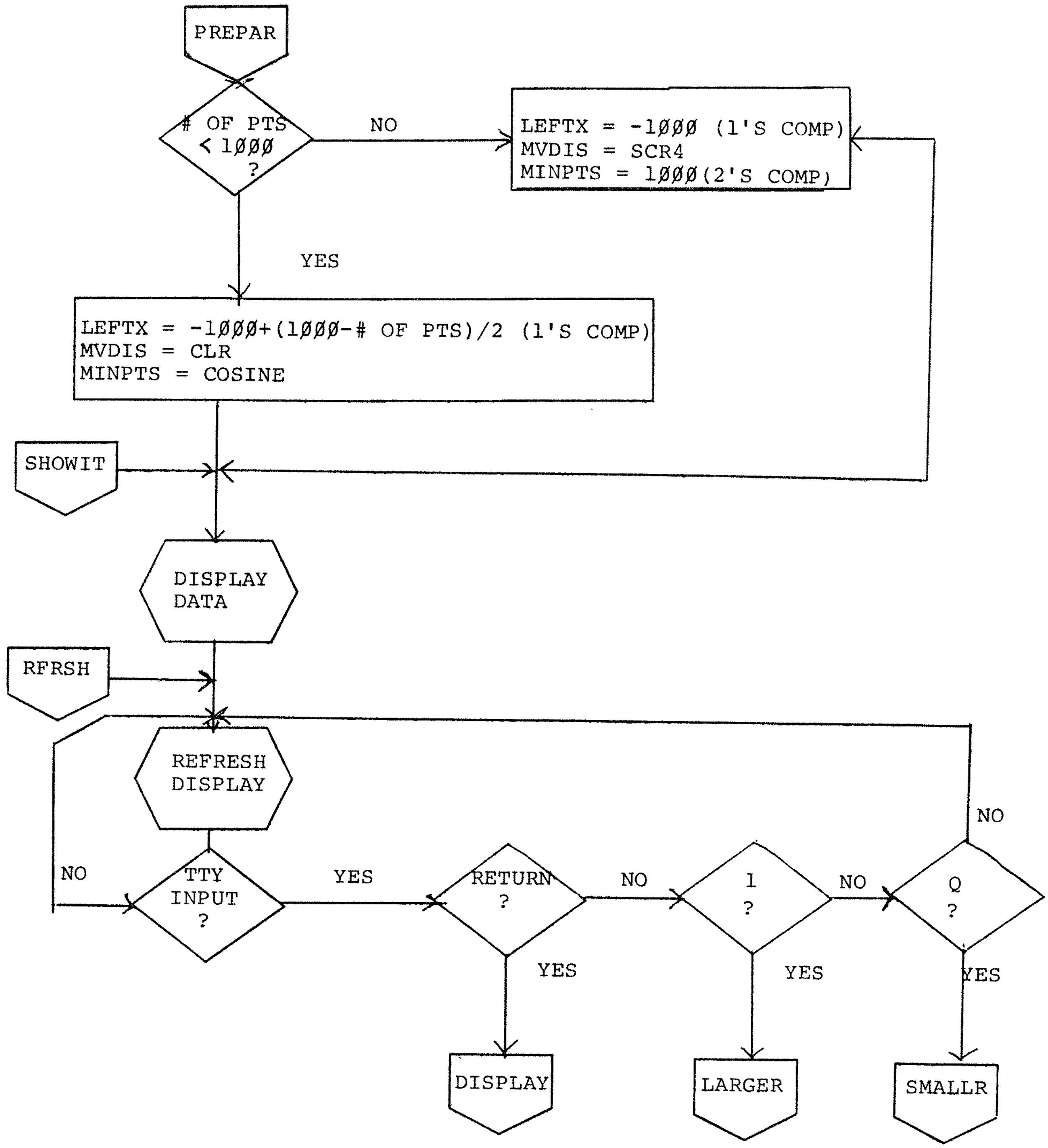

COMPUTE NUMBER OF OUTPUT BLOCKS
WILL INPUT RUN OFF END OF UNIT?
Y
PTS
N
OUTPUT ≥ 4ØØ WORDS ?
Y
ADD 1 BLOCK FOR SCALE FACTOR
N
DISPLAY:
FFT OR DISPLAY? F/D_
TRANSFORM OR INVERSE? T/I_
ANSWER ?
D
DISFLG=NON-ZERO
F
ANSWER ?
I
IFTFLG = NON-ZERO
T
IFTFLG=Ø
IFDISP
IFDISP
JUST DISPLAY ?
Y
MOVE OUTPUT PARAMETERS TO R/W PARAMETER TABLE
DISPLY
N
DISPLAY:
OUTPUT ON DIAL UNIT? Y/N_


ANSWER ?
N
DISPLAY: UNIT NUMBER__ BLK NUMBER___
ENOUGH SPACE ?
Y
RDDATA
Y
N
DISPLAY: UNIT NUMBER__ FILE NAME________
ALREADY ON TAPE ?
N
ENOUGH SPACE ?
N
DISPLAY: NO SPACE HIT RETURN TO CONT
Y
Y
DISPLAY: REPLACE? Y/N_
RDDATA
OUTQES
N
ANSWER ?
Y
REPLACE POSSIBLE ?
Y
RDDATA
N
NOSPAC


RDDATA
CLEAR DATA BUFFER
READ IN DATA
REAL OR COMPLEX?
COMPLEX
MOVE IMAG PARTS TO 2ØØØ
REAL
FFT OR IFFT ?
IFFT
IFFT
FFT
FFT
SORT
SCALE FACTOR= SCAL
WORD AFTER LAST POINT=SCALE FACTOR
NOWSTR
NOWSTR
PACK IMAG PARTS BEHIND REAL
MOVE OUTPUT PARAMETERS TO R/W PARAMETER TABLE
WRITE OUT DATA
DISPLY


DISPLY
READ IN DATA
DISPL1
DISPLAY:
WHICH DISPLAY?
R(EAL)
I(MAGINARY)
M(AGNITUDE)
S(CALE FACTOR)
LINE FEED (RESTART)
ANSWER
L/F
START 2Ø
M
DPMAG
S
R
I
N
WAS TRANSFORM DONE?
WAS TRANSFORM DONE?
N
Y
Y
SWAP HALVES
SWAP HALVES & REVERSE SIGN
CONVERT SCALE FACTOR TO ASCII DECIMAL
DISPLAY IT
C/R?
N
Y
PREPAR
PREPAR
DISPLY


DPMAG
JUST REAL PARTS ?
YES
DISPL1
NO
MOVE IMAG PARTS TO 2ØØØ
SQUARE A REAL PART
SQUARE AN IMAG PART
ADD SQUARES
SQUARE ROOT
STORE IN PLACE OF REAL PART
NO
DONE ?
YES
WAS TRANSFORM DONE?
YES
SWAP HALVES
PREPAR
NO
SMALLR
KYSCAL < 353 ?
YES
STEP KYSCAL
NO
RFRSH
LARGER
KYSCAL > 34Ø ?
YES
DECREMENT KYSCAL
NO
RFRSH


PREPAR
# OF PTS
< 1ØØØ
?
NO
LEFTX = -1ØØØ (1'S COMP)
MVDIS = SCR4
MINPTS = 1ØØØ(2'S COMP)
YES
LEFTX = -1ØØØ+(1ØØØ-# OF PTS)/2 (1'S COMP)
MVDIS = CLR
MINPTS = COSINE
SHOWIT
DISPLAY
DATA
RFRSH
REFRESH
DISPLAY
NO
TTY
INPUT
?
YES
RETURN
?
NO
1
?
NO
Q
?
NO
YES
YES
YES
DISPLAY
LARGER
SMALLR

INDEX

```
                  *2Ø
                  /FFTS-REAL
                  /THIS IS A PROGRAM FOR CALCULATING THE
                  /FAST FOURIER TRANSFORMATION OF N REAL
                  /TIME SAMPLES WHICH ARE STORED ON DIAL
                  /OR DATA TAPE OR DISK
                  /TO BE RUN ON A PDP-12 COMPUTER EQUIPPED/WITH THE FOLLOWING MINIMUM HARDWARE:
                  /  1) ASR 33 OR ASR 35 TELETYPE
                  /  2) 8 K OF CORE MEMORY
                  /  3) VR12 CRT DISPLAY
                  /
                  /COPYRIGHT 197Ø, DIGITAL EQUIPMENT CORPORATION
                  /         MAYNARD, MASS, Ø1754
                  /TRANSFORM ALGORITHM
                  /WRITTEN BY JAMES ROTHMAN -- AUGUST, 1968
                          QARFSH=1Ø53
                          QAINIT=1ØØØ
                          XRTAB=Ø
                          XITAB=2ØØØ
                          SINTAB=7347
                          CDF1=6211
                          CDFØ=62Ø1
                          PMODE
                  /PAGE ZERO
                  *3
                  /TABLE PARAMETERS
ØØØ3   ØØØØ       N,      Ø               /NUMBER OF POINTS IN COMPUTAION DIVIDED BY 2
ØØØ4   ØØØØ       NU,     Ø               /POWER OF TWO OF POINTS IN COMPUTATION (N=2↑NU) MINUS 1
ØØØ5   ØØØØ       L,      Ø               /INDEX TO SHOW WHAT ARRAY IS BEING CONSTRUCTED
ØØØ6   ØØØØ       S,      Ø               /GIVES SPACING BETWEEN NODE PAIRS IN THE LTH ARRAY,
ØØØ7   ØØØØ       F,      Ø               /USED FOR SCALING NODE POSITION TO GET NUMBER IN NODES.

                          *2Ø
ØØ2Ø   ØØØØ       NOVER4, Ø               /STORAGE FOR N/4
ØØ21   ØØ12       MAXNU,  BIGSNU          /LARGEST TABLE SIZE (POWER OF 2)
ØØ22   ØØØØ       MNOVR2, Ø               /STORAGE FOR -N/2
                  /INDEXING VARIABLES
ØØ23   ØØØØ       QR,     Ø               /POINTER TO REAL PART OF X(Q)
ØØ24   ØØØØ       QI,     Ø               /POINTER TO IMAG, PART OF X(Q)
ØØ25   ØØØØ       PR,     Ø               /POINTER TO REAL PART OF X(P)
ØØ26   ØØØØ       PI,     Ø               /POINTER TO IMAG, PART OF X(P)
ØØ27   ØØØØ       Q,      Ø               /NUMERICAL INDEX Q(=Ø,1,..N-1)
ØØ3Ø   ØØØØ       P,      Ø               /NUMBERICAL INDEX P(=Ø,1,..,N-1)
ØØ31   ØØØØ       K,      Ø               /NUMBER IN THE NODE BEING OPERATED ON
                  /LOOP DELIMITERS
ØØ32   ØØØØ       C,      Ø               /INTERRUPTS COMPUTATION OF LTH ARRAY EVERY S PASSES
                  /DATA VARIABLES
ØØ33   ØØØØ       ADD2,   Ø               /USED BY SUBROUTINE ADDR AS DATA (ADDEND)
ØØ34   ØØØØ       TEMPR,  Ø               /TEMPORARY STORAGE REGISTER FOR REAL PARTS
ØØ35   ØØØØ       SINE,   Ø               /TEMP. STORAGE FOR SIN (S*PI*K/N)
ØØ36   ØØØØ       COSINE, Ø               /TEMP. STORAGE FOR COS (2*PI*K/N)
ØØ37   ØØØØ       GR,     Ø               /REAL PART OF PRODUCT (W↑K)*X(P), TEMP STORAGE
ØØ4Ø   ØØØØ       GI,     Ø               /IMAG. PART OF (W↑K)*X(P), TEMP STORAGE
                  /SUBROUTINE CALL LIST
ØØ41   1135       ADDER,  ADDR            /ADD C(AC) TO C(ADD2) AND SCALE RIGHT ONE IF NECESSARY,
ØØ42   Ø7Ø1       SORT,   SORTX           /BIT INVERTED BUFFER SORTED
ØØ43   1Ø4Ø       INVERT, INVRT           /WORD IN AC OF NU BITS IS BIT INVERTED
ØØ44   1ØØØ       MULT,   MULTIP
ØØ45   1Ø6Ø       GETRIG, TRIGET          /FETCH SIN AND COS OF 2*PI*C(AC)/N
ØØ46   Ø4ØØ       DOFFT,  FFT             /DO FFT OF THE INPUT BUFFER
```

```
0050  7547  SINLOC, SINTAB
0051  0000  XRLOC,  XRTAB           /INPUT BUFFER AND TABLE OF ARRAYS
0052  2000  XLOCDF, XITAB=XRTAB     /DIFF IN ADDR OF REAL & IMAG PART TABLES
            /PSEUDO FLOATING POINT FORMAT FLAGS
0053  0000  SCAL,   0               /PSEUDO EXPONENT OF FOURIER COEFFICIENTS
0054  0001  SHFLAG, 1               /IF =1, ADD WITH SHIFT; IF=0, ADD WITHOUT SHIFT
0055  0000  SHFCHK, 0               /INDICATES IF ALL XS IN AN ITERATION ARE <.5
            /POINTERS TO SINE TABLE LOOK-UP SHIFTS
0056  1077  SHIFT1, SHFT1           /THE NUMBER 10-NU MUST BE PLACED
0057  1114  SHIFT2, SHFT2           /IN EACH OF THESE LOCATIONS
0060  1125  SHIFT3, SHFT3
            /POINTERS TO INSTRUCTION "FLAG" LOCATIONS
0061  0000  WORD,   0
0062  0000  WORDP,  0
0063  0000  FLIPCT, 0
            /
0064  0544  RBUILD, BUILD
0065  0542  RESETC, SETC
0066  0515  RECHK,  CHKPT
0067  4000  M4000,  -4000
0070  7777  M1,     -1
0071  7766  M12,    -12
0072  7770  M10,    -10
0073  6160  GRET10, 6160
0074  4060  LESS10, 4060
0075  7774  M4,     -4
0076  6270  PDPMAG, DPMAG
0077  7767  M11,    -11
0100  7773  M5,     -5
0101  6000  C6000,  6000
0102  7563  M215,   -215
0103  7457  M321,   -321
0104  7425  M353,   -353
0105  7440  M340,   -340
0106  7517  M261,   -261
0107  7400  M400,   -400
0110  1777  C1777,  1777
0111  0000  YSHFT,  0
0112  0000  XCURHI, 0
0113  0000  XCURLO, 0
0114  0000  CORVAL, 0
0115  0000  YCUR,   0
0116  0000  COUNT,  0
0117  1200  KIDORA, IDORA
0120  1343  KRDORA, RDORA
0121  6162  PSHOWT, SHOWIT
0122  6171  PRFRSH, RFRSH
0123  6404  PFDV7,  FDV+7
0124  1361  PMVDIS, MOVDIS
0125  1363  PLEFTX, LEFTX
0126  7132  PMRLMG, MVRLMG
0127  7116  PMVPTS, MOVPTS
0130  0000  CMPFLG, 0
0131  0000  MINPTS, 0
0132  6053  PRELFG, REALFG
0133  4356  PIFTFG, IFTFLG
0134  7774  PREAD,  7774
0135  7775  PWRITE, 7775
0136  1444  KYSCAL, YSCAL
0137  1000  C1000,  1000
0140  2000  C2000,  2000
0141  6777  M1K,    6777            /-1000 1,S CL
0142  0000  DPSQ,   0               /D.P. SQUARE
```

```
Ø174     Ø143  ØØØØ         Ø
Ø175                        LMODE
Ø176     Ø144  Ø644  LDF4,  LDF 4
Ø177     Ø145  Ø344  SCR4,  SCR 4
Ø2ØØ     Ø146  ØØ11  CCLR,  CLR
Ø2Ø1                        PMODE
Ø2Ø2                        EJECT
```

```
                  /THIS SUBROUTINE TAKES THE INVERSE FFT (IFFT) OF THE DATA IN THE BUFFER.
                  /IT IS ASSUMED THAT THIS DATA IS STORED SEQUENTIAL ORDER.
                  /THE RESULTS ARE STORED IN BIT INVERTED ORDER.
                  /THE ALGORITHM USED IS AS FOLLOWS:
                  /       THE NORMAL TRANSFORM IS PERFORMED, EXCEPT:
                  /       ON FETCHING THE VALUE FOR IM[W↑K], WHICH IS
                  /       THE SIN(2*PI*K/N), THIS SIN VALUE IS NEGATED.
                  /
                  /THE REASONING FOR THIS IS AS FOLLOWS:
                  /       A WEIGHTING FACTOR OF W↑8-K) IS USED IN THE IFFT
                  /       AND SINCE W↑K AND W↑(-K) ARE THE SAME EXCEPT THAT
                  /       THEIR IMAGINARY PARTS HAVE OPPOSITE SIGNS, IT FOLLOWS
                  /       THAT IM]W↑K] SHOULD BE REPLACED BY -IM[W↑K].
Ø147   ØØØØ   IFFT,   Ø
Ø15Ø   73ØØ           CLA CLL
Ø151   1162           TAD     CCIA    /NEGATE IM[W↑K], GET CIA INSTRUCTION
Ø152   3561           DCA I   SGNADJ  /AND PUT AT LOCATION ADJSN
Ø153   4443           JMS I   DOFFT   /DU FFT
Ø154   62Ø1           CDFØ
Ø155   1163           TAD     CNOP    /RE-INSTATE NOP AT ADJSGN FOR FFT.
Ø156   3561           DCA I   SGNADJ
Ø157   6211           CDF1
Ø16Ø   5547           JMP I   IFFT    /EXIT
Ø161   Ø57Ø   SGNADJ, ADJSGN          /POINTER TO SIGN ADJUST INSTRUCTION
Ø162   7Ø41   CCIA,   CIA
Ø163   7ØØØ   CNOP,   NOP
                      EJECT
                  *400
                  /COMPUTATION OF FIRST COMPLEX ARRAY FROM INPUT DATA
                  /NUMBER OF INPUT POINTS IN "N" .LOG[illegible](N)IN"NU", FOR DETAILS OF ALGORITHM, SEE FLOWCHART
0400  0000  FFT,      0
0401  7301            CLA IAC CLL
0402  3005            DCA      L                   /L<=1
0403  3053            DCA      SCAL                /INITIALIZE FLOATING POINT FORMAT
0404  7001            IAC
0405  3054            DCA      SHFLAG
0406  3055            DCA      SHFCHK
0407  1003            TAD      N
0410  7112            CLL RTR                      /INITIALIZE PROGRAM CONSTANTS
0411  3020            DCA      NOVER4
0412  1004            TAD      NU
0413  7041            CIA
0414  1021            TAD      MAXNU
0415  3456            DCA I    SHIFT1
0416  1456            TAD I    SHIFT1
0417  3457            DCA I    SHIFT2
0420  1457            TAD I    SHIFT2
0421  3460            DCA I    SHIFT3
0422  1003            TAD      N
0423  7110            CLL RAR
0424  3006            DCA      S                   /S<=N/2 IS SPACING OF NODE PAIRS IN FIRST ARRAY
0425  1006            TAD      S
0426  7041            CIA
0427  3022            DCA      MNOVR2
0430  7040            CMA                          /AC<=-1
0431  1006            TAD      S                   /AC<=[N/2-1]*2
0432  1051            TAD      XRLOC               /BEGINNING OF TABLE OF REAL PARTS,
0433  3023            DCA      QR                  /Q<=N/2-1, QR POINTS TO WORD IN MEMORY, WHILE Q IS ACTUAL INDEX
0434  1004            TAD      NU
0435  7041            CIA
0436  7001            IAC
0437  3007            DCA      F                   /F<=1-NU (=L-NU SINCE L=1)
0440  1023  LOOP1,    TAD      QR                  /QR=XRLOC+Q AT ALL TIMES.
0441  1006            TAD      S
0442  3025            DCA      PR                  /P<=Q+N/2
0443  1023            TAD      QR                  /XLOCDF=XILOC-XRLOC (XILOC=BEGIN, OF IMAG PARTS TABLE)
0444  1052            TAD      XLOCDF              /QR+XLOCDF=(S+XRLOC)+(XILOC-XRLOC)=XILOC+S=QI
0445  3024            DCA      QI                  /QI=XILOC+Q AT ALL TIMES, QI POINTS TO IMAG. PART OF X(Q)
0446  1025            TAD      PR
0447  1052            TAD      XLOCDF              /COMPUTE COMPLEX OPERATIONS X(P)<=X(Q)-X(P) AND X(Q)<=X(Q)+X(P)
0450  3026            DCA      PI                  /BY REAL AND IMAGINARY PARTS.
0451  6211            CDF1
0452  1424            TAD I    QI                  /IM(X(Q)) (IM () MEANS IMAGINARY PART)
0453  3033            DCA      ADD2                /MAKE IT ADDEND, DO IMAG. PARTS FIRST
0454  1426            TAD I    PI                  /IM(X(P))
0455  4441            JMS I    ADDER               /FORM ADDITION IM[X(P)+X(Q)]=IM[X(P)]+IM[X(Q)] AND SCALE RIGHT
0456  3034            DCA      TEMPR               /FOR SCALING, THEN STORE.
0457  1424            TAD I    QI                  /FORM DIFFERENCE IM[X(Q)-X(P)]=IM[X(Q)]-IM[X(P)]
0460  3033            DCA      ADD2
0461  1426            TAD I    PI
0462  7041            CIA
0463  4441            JMS I    ADDER
0464  3426            DCA I    PI                  /PUT AWAY AT IM[X(P)]
0465  1034            TAD      TEMPR               /GET IM[X(P)+X(Q)]
0466  3424            DCA I    QI                  /PUT AT IM[X(Q)], IMAGINARY PARTS DONE.
0467  1423            TAD I    QR                  /ADD REAL PARTS NEXT
0470  3033            DCA      ADD2
0471  1425            TAD I    PR                  /RE=REAL PART
0472  4441            JMS I    ADDER               /FORM RE[X(P)+X(Q)]=RE[X(P)]+RE[X(Q)]
```

```
0474  1423          TAD I   QR             /GET RE[X(Q)]
0475  3033          DCA     ADD2
0476  1425          TAD I   PR             /RE=REAL PART
0477  7041          CIA
0500  4441          JMS I   ADDER          /FORM RE[X(Q)-(P)] (DIVIDED BY 2)
0501  3425          DCA I   PR             /PUT AT RE[X(P)]
0502  1034          TAD     TEMPR          /GET RE[X(Q)+X(P)]
0503  3423          DCA I   QR             /PUT AT RE[X(Q)],REAL PARTS DONE
0504  1051          TAD XRLOC              /Q=QR-XRLOC
0505  7041          CIA
0506  1023          TAD QR          /AC IS Q
0507  7750          SPA SNA CLA     /IS Q>0? (IE THE WHOLE ARRAY HAS NOT BEEN COVERED)
0510  5315          JMP CHKPT       /NO, Q=0, DONE WITH FIRST ARRAY, MOVE ON TO OTHERS
0511  7040          CMA             /YES, Q<=Q-1, MOVE UP THIS ARRAY
0512  1023          TAD QR          /OR EQUIVALENTLY, QR<=QR-1
0513  3023          DCA QR
0514  5240          JMP LOOP1       /DO NEXT NODE PAIR
0515  1005  CHKPT,  TAD L           /L GIVES THE NUMBER OF THE VERTICAL ARRAY JUST BUILT
0516  7041          CIA
0517  1004          TAD NU          /IS L=NU?  (IE HAS THE LAST ARRAY BEEN COMPUTED?)
0520  7650          SNA CLA
0521  5600          JMP I FFT    /YES, DONE, RESULTS STORED IN BIT REVERSED ORDER
0522  1055          TAD SHFCHK      /GET SCALE FACTOR AND ADJUST FOR PROPER
0523  3054          DCA SHFLAG      /ADDITION ON NEXT ITERATION
0524  1055          TAD     SHFCHK
0525  7650          SNA CLA
0526  2053          ISZ     SCAL
0527  3055          DCA     SHFCHK
0530  2005          ISZ     L              /L<=L+1, MOVE ON TO NEXT ARRAY
0531  1006          TAD     S              /S GIVES SPACING BETWEEN NODE PAIRS, WHICH IS N/2↑L
0532  7110          CLL RAR                /DIVIDE BY 2 AND PUT BACK, SO THAT ON THE LTH PASS THROUGH
0533  3006          DCA     S              /S WILL=N/2↑L, THE SPACING,
0534  2007          ISZ     F              /F<=F+1, ON LTH PASS, F WILL BE F=L-NU, THE SCALE FACTOR FOR K.
0535  7000          NOP                    /NOP FOR WHEN F=-1 TO PREVENT ERROR DUE TO SKIP
0536  7040          CMA                    /AC<=-1
0537  1003          TAD     N
0540  1051          TAD     XRLOC
0541  3025          DCA     PR             /P<=N-1, PR POINTS TO RE[X(P=N-1)]
0542  7201  SETC,   CLA IAC
0543  3032          DCA     C              /C<=1, C BREAKS BUILD LOOP EVERY S ITERATIONS
0544  1025  BUILD,  TAD     PR      /SO AS TO AVOID RECOMPUTATION
0545  1052          TAD     XLOCDF
0546  3026          DCA     PI
0547  1051          TAD     XRLOC   /PR=XRLOC+P
0550  7041          CIA
0551  1025          TAD     PR
0552  3030          DCA     P       /ACTUAL INDEX IS P:(0,1,,,,N-1)
0553  1007          TAD     F              /BUILD ARRAY, F=L-NU, SHIFT "P"-F PLACES RIGHT (=NU-L)
0554  7450          SNA                    /SHIFT ZERO PLACES?
0555  5364          JMP     NOROT          /YES, LEAVE ALONE
0556  7040          CMA                    /F COMPLEMENTED IS -F-(1)=-(F+1)=PLACES TO BE SHIFTED-1
0557  3362          DCA     SHIFCT         /CONTAINS-F-1
0560  1030          TAD     P              /GET NODE INDEX
0561  7417          LSR                    /SHIFT P RIGHT SHIFCT+1=-F-1+1=-F=NU-L PLACES
0562  7402  SHIFCT, HLT             /STORAGE FOR SHIFT COUNT.
0563  7410          SKP                    /AC<=INTEGER PART [P*2↑F]
0564  1030  NOROT,  TAD     P              /NO ROTATION, JUST GET P=P*2↑0
0565  4443          JMS I   INVERT         /INVERT BIT ORDER AND PUT IN K (NUMBER IN PTH NODE)
0566  1022          TAD     MNOVR2         /SUBTRACT N/2 TO GET NUMBER IN Q (=K) (PS NODE PAIR.)
0567  4445          JMS I   GETRIG         /GET REAL AND IMAGINARY PARTS OF W↑K.
0570  7000  ADJSGN, NOP                    /SET    CIA FOR DOING IFFT, NOP FOR FFT.
0571  3035          DCA     SINE           /SIN(2PI*K/N)=-IM[W↑K], COS IN REGISTER COSINE.
0572  1425          TAD I   PR             /FORM (W↑K)*X(P)-A COMPLEX MULTIPLICATION
```

```
Ø434   Ø573  4444      JMS I   MULT          /DO [illegible] PART FIRST=RE[X(P)]*COSINE+IM[X(P)]*SINE
Ø435   Ø574  ØØ36      COSINE                /AC= [illegible](P)]*COSINE=RE[X(P)]*RE[W↑K]
Ø436   Ø575  3Ø33      DCA     ADD2          /SAVE FOR ADDITION LATER
Ø437   Ø576  1426      TAD I   PI            /GET IM[X(P)]
Ø44Ø   Ø577  4444      JMS I   MULT
Ø441   Ø6ØØ  ØØ35      SINE                  /AC=IM[X(P)]*SINE=-IM[W↑K]*IM[X(P)]
Ø442   Ø6Ø1  1Ø33      TAD     ADD2          /AC=RE[W↑K]*RE[X(P)]-IM[W↑K]*IM[X(P)]=RE[X(P)=W↑K]
Ø443   Ø6Ø2  3Ø37      DCA     GR            /STORE AT GR
Ø444              /DO IMAG, PART NEXT=IM[X(P)]*COSINE-RE[X(P)]*SINE=IM[X(P)]*RE[W↑K]+RE[X(P)]*IM[W↑K]
Ø445   Ø6Ø3  1426      TAD I   PI
Ø446   Ø6Ø4  4444      JMS I   MULT          /AC=IM[X(P)]
Ø447   Ø6Ø5  ØØ36      COSINE                /AC=IM[X(P)]*COSINE=IM[(P)]*RE[W↑K]
Ø45Ø   Ø6Ø6  3Ø33      DCA ADD2              /STORE FOR LATER ADDITION
Ø451   Ø6Ø7  1425      TAD I PR              /AC=RE[X(P)]
Ø452   Ø61Ø  4444      JMS I MULT
Ø453   Ø611  ØØ35      SINE             /AC=RE[X(P)]*SINE=-RE[X(P)]*IM[W↑K]
Ø454   Ø612  7Ø41      CIA              /AC=RE[X(P)]M*IM[W↑K]
Ø455   Ø613  1Ø33      TAD ADD2         /AC=IM[X(P)]*RE[W↑K]+RE[X(P)]*IM[W↑K]=IM[X(P)*W↑K]
Ø456   Ø614  3Ø4Ø      DCA GI           /STORE AT GI, SO GI=IM[X(P)*W↑K] AND GR=RE[X(P)*W↑K] G=GR+I*GI
Ø457   Ø615  1ØØ6      TAD S            /LOCATE P NODE PAIR Q, LOCATED S=N/(2↑L) UP ARRAY
Ø46Ø   Ø616  7Ø41      CIA              /SO SET Q=P-S=INDEX OF NODE PAIR
Ø461   Ø617  1Ø25      TAD PR           /LOCATE X(Q) IN MEMORY BY FIXING POINTERS QR AND QI
Ø462   Ø62Ø  3Ø23      DCA QR           /TO QS REAL AND IMAG PARTS RESPECTIVELY
Ø463   Ø621  1Ø23      TAD     QR
Ø464   Ø622  1Ø52      TAD     XLOCDF
Ø465   Ø623  3Ø24      DCA     QI
Ø466   Ø624  1423      TAD I   QR            /DO THE COMPLEX OPERATIONS:  X(P)<=X(Q)-G;X(Q)<=X(Q)+G
Ø467   Ø625  3Ø33      DCA     ADD2          /FIRST DO REAL PART OF X(P), GET RE[X(Q)] AND STORE
Ø47Ø   Ø626  1Ø37      TAD     GR            /GET RE[G]
Ø471   Ø627  7Ø41      CIA
Ø472   Ø63Ø  4441      JMS I   ADDER         /SUBTRACT THEM,
Ø473   Ø631  3425      DCA I   PR            /RE[X(P)]<=RE[X(Q)]-RE[G]
Ø474   Ø632  1424      TAD I   QI            /COMPUTE IMAG, PART OF X(P), GET IM[X(Q)]
Ø475   Ø633  3Ø33      DCA     ADD2          /AND STORE
Ø476   Ø634  1Ø4Ø      TAD     GI            /GET IM[G]
Ø477   Ø635  7Ø41      CIA
Ø5ØØ   Ø636  4441      JMS I   ADDER         /AND SUBTRACT THEM.
Ø5Ø1   Ø637  3426      DCA I   PI            /IM[X(P)]<=IM[X(Q)]-IM[G],X(P) IS NOW DONE.
Ø5Ø2   Ø64Ø  1423      TAD I   QR            /NEXT COMPUTE X(Q), FIRST REAL PART
Ø5Ø3   Ø641  3Ø33      DCA     ADD2          /GET RE[G] AND STORE
Ø5Ø4   Ø642  1Ø37      TAD     GR            /GET RE[G] AND ADD TO FORM
Ø5Ø5   Ø643  4441      JMS I   ADDER         /RE[X(Q)]+RE[G].
Ø5Ø6   Ø644  3423      DCA I   QR            /RE[X(Q)]<=RE[X(Q)]+RE[G]
Ø5Ø7   Ø645  1424      TAD I   QI            /NOW COMPUTE IMAG PART OF X(Q), GET IM[X(Q)]
Ø51Ø   Ø646  3Ø33      DCA     ADD2          /AND STORE
Ø511   Ø647  1Ø4Ø      TAD     GI            /GET IM[G] AND ADD TO FORM
Ø512   Ø65Ø  4441      JMS I   ADDER         /IM[X(Q)]+IM[G]
Ø513   Ø651  3424      DCA I   QI            /IM[X(Q)]<=IM[X(Q)]+IM[G], THE NEW NODE PAIR IS COMPUTED.
Ø514   Ø652  7Ø4Ø      CMA                   /MOVE UP ARRAY TO NEXT NODE, SET AC=-1
Ø515   Ø653  1Ø3Ø      TAD     P             /TO FORM -1
Ø516   Ø654  3Ø3Ø      DCA     P             /P<=P-1
Ø517   Ø655  7Ø4Ø      CMA
Ø52Ø   Ø656  1Ø25      TAD     PR       /DO THE SAME FOR POINTER PR
Ø521   Ø657  3Ø25      DCA     PR
Ø522   Ø66Ø  1Ø32      TAD     C             /CHECK ON SPACING, IS A NODE WHICH HAS ALREADY BEEN COMPUTED
Ø523   Ø661  7Ø41      CIA                   /ABOUT TO BE RE-DONE, OR EQUIVALENTLY,
Ø524   Ø662  1ØØ6      TAD     S             /IS C=S?
Ø525   Ø663  764Ø      SZA CLA               /YES.
Ø526   Ø664  5277      JMP     CNOTS         /NO, DO NEXT NODE PAIR
Ø527   Ø665  1Ø3Ø      TAD     P             /YES, BUT ARE WE AT THE TOP OF THE ARRAY?
Ø53Ø   Ø666  7Ø4Ø      CMA                   /OR, IS S=P+1?  (P COMPLEMENTED=-P-1=-(P+1)
Ø531   Ø667  1ØØ6      TAD     S
```

```
0671 5466           JMP I   RECHK          /YES, DONE WITH THIS ARRAY, DO NEXT ONE.
Ø672 1ØØ6           TAD     S              /NO, MOVE PAST AREA THAT HAS AREADY BEEN DONE, OR SET P TO P-S.
Ø673 7Ø41           CIA                    /BY CHANGING THE POINTER TO RE[X(P)]
Ø674 1Ø25           TAD     PR
Ø675 3Ø25           DCA     PR
Ø676 5465           JMP I   RESETC         /REINITIALIZE C TO 1 SINCE AN UNUSED AREA HAS BEEN ENTERED.
Ø677 2Ø32 CNOTS,    ISZ     C              /C<=C+1, ANOTHER NODE PAIR HAS BEEN HANDLED.
Ø7ØØ 5464           JMP I   RBUILD         /DO NEXT NODE PAIR IN THIS AREA.
Ø7Ø1 ØØØØ SORTX,    Ø                      /SUBROUTINE THAT
Ø7Ø2 7Ø4Ø           CMA                    /SORTS OUT TRANSFORMS BY
Ø7Ø3 1ØØ3           TAD     N              /BIT INVERSION OF ADDRESS.
Ø7Ø4 3Ø27           DCA     Q              /Q<=N-1, START FROM BOTTOM OF BUFFER
Ø7Ø5 1Ø27 REVERS,   TAD     Q              /P<=BIT INVERTED Q
Ø7Ø6 4443           JMS I   INVERT         /BIT INVERSION ROUTINE
Ø7Ø7 3Ø3Ø           DCA     P
Ø71Ø 1Ø3Ø           TAD     P              /FORM Q-P
Ø711 7Ø41           CIA
Ø712 1Ø27           TAD     Q
Ø713 775Ø           SPA SNA CLA            /IS P<Q?
Ø714 5345           JMP     SWAPED         /NO, HAVE ALREADY DONE THIS PAIR
Ø715 1Ø3Ø           TAD     P              /YES, SWAP ORDER
Ø716 1Ø51           TAD     XRLOC          /FIRST SET UP SUBSCRIPT POINTERS FOR X(P) AND X(Q).
Ø717 3Ø25           DCA     PR
Ø72Ø 1Ø27           TAD     Q
Ø721 1Ø51           TAD     XRLOC
Ø722 3Ø23           DCA     QR
Ø723 1Ø25           TAD     PR
Ø724 1Ø52           TAD     XLOCDF
Ø725 3Ø26           DCA     PI
Ø726 1Ø23           TAD     QR
Ø727 1Ø52           TAD     XLOCDF
Ø73Ø 3Ø24           DCA     QI       /EXCHANGE: X(P)<=X(Q) AND X(Q)<=X(P)
Ø731 1425           TAD I   PR                     /EXCHANGE REAL PARTS, GET RE[X(P)]
Ø732 3Ø34           DCA     TEMPR                  /STORE IT.
Ø733 1423           TAD I   QR                     /GET RE[X(Q)]
Ø734 3425           DCA I   PR                     /MAKE IT RE[X(P)]
Ø735 1Ø34           TAD     TEMPR                  /GET RE[X(P)]
Ø736 3423           DCA I   QR                     /MAKE IT RE[X(Q)]
Ø737 1426           TAD I   PI                     /EXCHANGE IMAGINARY PARTS, GET IM[X(P)]
Ø74Ø 3Ø34           DCA     TEMPR                  /STORE IT.
Ø741 1424           TAD I   QI                     /GET IM[X(Q)]
Ø742 3426           DCA I   PI                     /MAKE IT IM[X(P)]
Ø743 1Ø34           TAD     TEMPR                  /GET IM[X(P)]
Ø744 3424           DCA I   QI                     /MAKE IT IM[X(Q)]
Ø745 1Ø27 SWAPED,   TAD     Q              /IS Q=Ø?, IE; ARE WE AT THE TOP OF THE ARRAY
Ø746 764Ø           SZA CLA
Ø747 5352           JMP .+3
Ø75Ø 62Ø1           CDFØ
Ø751 57Ø1           JMP I   SORTX          /YES, DONE EXIT
Ø752 7Ø4Ø           CMA                    /NO, Q<=Q-1, IE; MOVE UP THE ARRAY
Ø753 1Ø27           TAD     Q
Ø754 3Ø27           DCA     Q
Ø755 53Ø5           JMP     REVERS         /GO BACK AND CONTINUE
                    EJECT
```

```
                         *1000
                 /SIGNED S.P. MULTIPLY, USING THE EAE
                 /ENTRY: AC=MULTIPLIER, C(CALL+1)=ADDR OF MULTIPLICAND, EXIT*AC=PRODUCT,
                 /AN 11 BIT SIGNED BINARY FRAC
1000  0000       MULTIP, 0                  /AC=ARG1 (MULTIPLIER)
1001  7100               CLL                /ARG1>0?
1002  7510               SPA
1003  7061               CMA CML IAC        /NO-MAKE POS-SET L=1 TO SHOW IT WAS NEG
1004  7421               MQL                /LOAD INTO MQ
1005  6201               CDF0
1006  1600               TAD I   MULTIP     /GET ADDR OF MULTIPLICAND
1007  3220               DCA     ARG2               /STORE
1010  1620               TAD I   ARG2               /AND RETRIEVE MULTIPLICAND ITSELF.
1011  2200               ISZ     MULTIP             /(FOR EXIT AT CALL+2)

1012  7510               SPA                        /ARG2>0?
1013  7061               CMA CML IAC                /NO, MAKE POSITIVE, CHANGE LINK, SINCE-1+-1=1 AND -1+1=-1
1014  3220               DCA     ARG2               /PUT AWAY AT ARG2
1015  7010               RAR
                                            /SIGN IN LINK, PUT INTO AC11 AND
1016  3237               DCA     SIGN               /PUT AWAY AT SIGN (=1 IF -; =0 IF +)
1017  7405               MUY                        /DO MULTIPLICATION
1020  7402       ARG2,   HLT                        /ARGUMENT 2 (MULTIPLICAND)
1021  7413               SHL                        /NORMALIZE BINARY POINT.
1022  0000               0
1023  3220               DCA     ARG2               /SAVE HIGH ORDER, NOW ROUND OFF.
1024  1237               TAD     SIGN
1025  7413               SHL                        /SET AC11=MQ0,AC0-10=0
1026  0000               0
1027  1220               TAD     ARG2
1030  7510               SPA
1031  7350               CLA CLL CMA RAR
1032  7000               NOP
1033  7430               SZL                        /POSITIVE SIGN?
1034  7041               CMA IAC                    /NO, NEGATE
1035  6211               CDF1
1036  5600               JMP I   MULTIP             /EXIT, SIGNED RESULT IN AC.
1037  0000       SIGN,   0
                 /BIT INVERSION ROUTINE
                 /ENTRY:  AC=WORD TO BE INVERTED; EXIT:AC=RESULT
                 /NU CONTAINS THE NUME OF BITS IN THE WORD
1040  0000       INVRT,  0
1041  3061               DCA     WORD               /GET WORD TO BE INVERTED
1042  3062               DCA     WORDP              /ZERO OBJECT REGISTER
1043  1004               TAD     NU                 /GET NUMBER OF BITS TO BE
1044  7041               CIA                        /INVERTED AND USE TO LIMIT THE
1045  3063               DCA     FLIPCT             /EXTENT OF LOOP
1046  1061       FLIP,   TAD     WORD               /PULL OUT RIGHTMOST BIT OF WORD
1047  7110               CLL RAR            /RT MOST BIT NOW IN AC
1050  3061               DCA     WORD               /(PUT BACK SO A NEW BIT IS OPERATED ON EACH TIME)
1051  1062               TAD     WORDP              /AND PUSH INTO WORDP FROM LEFT
1052  7004               RAL
1053  3062               DCA     WORDP
1054  2063               ISZ     FLIPCT             /ALL BITS DONE?
1055  5246               JMP     FLIP               /NO, DO NEXT BIT
1056  1062               TAD     WORDP              /YES, PICK UP RESULT
1057  5640               JMP I   INVRT              /AND EXIT
                         EJECT
```

```
                /THIS SUBROUTINE FETCHES THE VALUES OF SIN(2*PI*C(AC)/N)
                /AND OF COS(2*PI*C(AC)/N) FOR C(AC) < N/2+1
                /ENTRY:  AC=INDEX OF LOOP UP
                /EXIT :  COS(2*PI*C(AC)/N) STORED AT "COSINE" AND
                /        AC=VALUE OF SIN(2*PI*C(AC)/N).
1060  0000      TRIGET, 0
1061  6201              CDFØ
1062  3031              DCA     K               /STORE C(AC) AT K.
1063  7421              MQL                     /CLEAR MQ
1064  1031              TAD     K               /FORM N/4-K,
1065  7141              CLL CIA
1066  1020              TAD     NOVER4
1067  3333              DCA     NO4MIK
1070  7430              SZL                     /IS N/4-K<Ø?
1071  5310              JMP     QUAD1           /NO, FIRST QUADRANT ANGLE.
1072  1333      QUAD2,  TAD     NO4MIK          /2ND QUADRANT, GET -COS AT K-N/4.
1073  7041              CIA
1074  7417              LSR                     /MAKE CORRECTIVE RIGHT SHIFT ON INDEX.
1075  0000              Ø
1076  7413              SHL                     /FIND ON SINE TABLE FOR 2*MAXNU BY MULTIPLYING
1077  7402      SHFT1,  HLT                     /INDEX BY 2*(MAXNU-NU), WHICH IS STORED HERE.
1100  1050              TAD     SINLOC          /LOCATE IT IN MEMORY.
1101  3334              DCA     INDEX
1102  1734              TAD I   INDEX
1103  7041              CIA                     /2ND QUADRANT COS IS NEGATIVE.
1104  3036              DCA     COSINE
1105  1333              TAD     NO4MIK          /GET SIN AT N/2-K
1106  1020              TAD     NOVER4
1107  5322              JMP     SINRET
1110  1333      QUAD1,  TAD     NO4MIK          /GET COS AT N/4-K,
1111  7417              LSR
1112  0000              Ø
1113  7413              SHL
1114  7402      SHFT2,  HLT
1115  1050              TAD     SINLOC
1116  3334              DCA     INDEX
1117  1734              TAD I   INDEX
1120  3036              DCA     COSINE
1121  1031              TAD     K                          /GET SIN AT K.
1122  7417      SINRET, LSR
1123  0000              Ø
1124  7413              SHL
1125  7402      SHFT3,  HLT
1126  1050              TAD     SINLOC
1127  3334              DCA     INDEX
1130  1734              TAD I   INDEX           /AC=SINVALUE.
1131  6211              CDF1
1132  5660              JMP I   TRIGET
1133  0000      NO4MIK, Ø                       /STORAGE FOR N/4-K
1134  0000      INDEX,  Ø                       /POINTER TO SINE TABLE
                /THIS ROUTINE PERFORMS A SINGLE PRECISION ADD WITH ROUNDING EACH ARGUMENT IS
                /SHIFTED RIGHT ONCE TO PREVENT OVERFLOW OF BINARY POINT (IF NECESSARY)
                /AND THEN CHECKED TO SEE IF IT CAN BE NORMALIZED AFTER ADDITION
                /ENTRY:  AC=ADDEND,C(ADD2)=AUGEND
                /EXIT  :  -AC=RESULT, DIVIDED BY TWO IF NECESSARY.
1135  0000      ADDR,   Ø
1136  3374              DCA     ADD1
1137  1054              TAD     SHFLAG          /SHOULD ADD BE DONE WITH SHIFT?
1140  7650              SNA CLA
1141  5357              JMP     ADDWOS          /NO, GO ADD WITH OUT SHIFT
1142  1374              TAD     ADD1            /YES GET ADDEND
1143  7415              ASR                     /DO 1 SIGNED RIGHT SHIFT
1144  0000              Ø
```

```
1145 3374          DCA     ADD1
1146 1Ø33          TAD     ADD2
1147 7415          ASR                     /MQØ=LO(ADD2)
115Ø ØØØØ          Ø                       /MQ(1)=LO(ADD(1))
1151 3Ø33          DCA     ADD2
1152 75Ø1          MQA                     /GET MQ
1153 7ØØ4          RAL                     /L<=LO(ADD2); ACØ<=LO(ADD1)
1154 7Ø6Ø          CMA CML                 /COMPLEMENT BOTH,

1155 772Ø          SMA SNL CLA             /IF BOTH WERE=1 (NEITHER=Ø), INTRODUCE A CARRY.
1156 7ØØ1          IAC
1157 1374 ADDWOS,  TAD     ADD1            /DO THE ADDITION,
116Ø 1Ø33          TAD     ADD2
1161 3375          DCA     XSUM            /STORE THE RESULT
1162 1375          TAD     XSUM            /CHECK TO SEE IF ALREADY NORMALIZED,
1163 751Ø          SPA                     /IS IT POSITIVE?
1164 7Ø41          CIA                     /MAKE IT POSITIVE,
1165 7ØØ4          RAL                     /GET BIT 1, WAS NORMALIZED IF =1
1166 77ØØ          SMA CLA
1167 5372          JMP     NOTNOR          /NOT NORMALIZED, LEAVE SHFCHK ALONE,
117Ø 7ØØ1          IAC
1171 3Ø55          DCA     SHFCHK          /SET SHFCHK=1
1172 1375 NOTNOR,  TAD     XSUM
1173 5735          JMP I   ADDR            /AND EXIT
1174 ØØØØ ADD1,    Ø                /ADDEND STORAGE
1175 ØØØØ XSUM,    Ø                /TEMP STORAGE FOR SUM
                   EJECT
```

```
1Ø45
1Ø46            /DEFINITIONS FOR EAE
1Ø47            DVI=74Ø7
1Ø5Ø            NMI=7411
1Ø51            SHL=7413
1Ø52            ASR=7415
1Ø53            LSR=7417
1Ø54            MQL=7421
1Ø55            MUY=74Ø5
1Ø56            MQA=75Ø1
1Ø57            CAM=7621
1Ø6Ø            SCA=7441
1Ø61            SCL=74Ø3
1Ø62            /ASSEMBLY PARAMETERS
1Ø63            BIGSNU=12       /LARGEST TRANSFORMATION HAS DIMENSION 2↑1Ø.
1Ø64                    EJECT
-
```

```
                  /MOVING WINDOW DISPLAY SUBROUTINE
                          PMODE
                          PAGE
1200   ØØØØ       IDORA,  Ø                   /GET BOUNDS
1201   73ØØ               CLA CLL
1202   62Ø1       ACDFØ,  CDF Ø
1203   16ØØ               TAD I IDORA         /DATA BUFFER
1204   3635               DCA I KMNFLD        /15 BIT
1205   22ØØ               ISZ IDORA           /LOWER BOUND
1206   16ØØ               TAD I IDORA         /AT P+1, P+2
1207   3636               DCA I KMNADR        /MINFLD,MINADR
1210   22ØØ               ISZ IDORA
1211   16ØØ               TAD I IDORA         /UPPER BOUND
1212   3637               DCA I KMXFLD        /AT P+3, P+4
1213   22ØØ               ISZ IDORA
1214   7ØØ1               IAC                 /RDORA USES
1215   16ØØ               TAD I IDORA         /MAX+1
1216   364Ø               DCA I KMXADR
1217   7ØØ4               RAL
1220   1637               TAD I KMXFLD
1221   3637               DCA I KMXFLD
1222   22ØØ               ISZ IDORA
1223   16ØØ               TAD I IDORA         /Y SHIFT
1224   3111               DCA YSHFT
1225   22ØØ               ISZ IDORA
1226   16ØØ               TAD I IDORA         /Y SCALE
1227   3536               DCA I KYSCAL
1230   1635               TAD I KMNFLD        /INITIALIZE
1231   3641               DCA I KBUFHI        /WINDOW
1232   1636               TAD I KMNADR        /STARTING ADDR
1233   3642               DCA I KBUFLO
1234   56ØØ               JMP I IDORA         /RTN TO SCR N
1235   1415       KMNFLD, MINFLD
1236   1416       KMNADR, MINADR
1237   1474       KMXFLD, MAXFLD
1240   1475       KMXADR, MAXADR
1241   1574       KBUFHI, BUFHI
1242   1575       KBUFLO, BUFLO
1243   Ø4Ø1       P4Ø1,   4Ø1
1244   1243       DSCLOC, TAD P4Ø1            /DSC X,Y COORD
1245   3274               DCA VCOORD
1246   1112               TAD XCURHI          /FIELD
1247   4261               JMS DSCWD
1250   1113               TAD XCURLO          /ADDRESS
1251   4261               JMS DSCWD
1252   1114               TAD CORVAL          /CONTENTS OF
1253   4261               JMS DSCWD           /CURSR CORE LOC
1254   1115               TAD YCUR            /Y COORD OF
1255   1243               TAD P4Ø1
1256   4261               JMS DSCWD           /CURSOR POINT
1257   ØØØØ       RTNCDF, Ø                   /RESTORE USER
                                              /DATA FLD
1260   5743               JMP I RDORA         /RTN
1261   ØØØØ       DSCWD,  Ø                   /DSC C(AC)
1262   6141               LINC
                          LMODE
1263   5276               STC TEMP            /SAVE VALUE
1264   4ØØ1               STC XCORD           /CHAN 1
1265   ØØ24               SFA                 /VC FOR FULL
1266   Ø265               ROL I 5             /SIZE IS -4Ø
1267   1Ø2Ø               LDA I               /-2Ø FOR HALF
1270   7757               -2Ø
```

```
                                ROL 1              /NO VC*4Ø
1165     1273  116Ø             ADM I              /UPDATE VC
1166     1274  ØØØØ   VCOORD,   Ø
1167     1275  1Ø2Ø   DSCLOP,   LDA I
117Ø     1276  ØØØØ   TEMP,     Ø
1171     1277  Ø243             ROL 3              /1 DIGIT
1172     13ØØ  1Ø4Ø             STA                /AT A TIME
1173     13Ø1  1276             TEMP               /UPDATE
1174     13Ø2  156Ø             BCL I              /LOW 3 BITS
1175     13Ø3  777Ø             777Ø               /ONLY
1176     13Ø4  Ø241             ROL 1              /*2 AND REL
1177     13Ø5  112Ø             ADA I              /TO GRID TAB
12ØØ     13Ø6  1323             TAB&1777
12Ø1     13Ø7  4ØØ2             STC 2
12Ø2     131Ø  3274             ADD VCOORD
12Ø3     1311  1742             DSC 2
12Ø4     1312  1762             DSC I 2
12Ø5     1313  Ø221             XSK I 1            /MAKE GAP
12Ø6     1314  Ø221             XSK I 1            /BETWEEN CHARS
12Ø7     1315  152Ø             SRO I              /DSC 4 CHARS ?
121Ø     1316  3567             3567
1211     1317  7275             JMP DSCLOP         /NO CONT
1212     132Ø  ØØØ2             PDP
1213                            PMODE
1214     1321  73ØØ             CLA CLL
1215     1322  5661             JMP I DSCWD        /RTN
1216     1323  4536   TAB,      4536     /6Ø,Ø
1217     1324  3651             3651
122Ø     1325  21Ø1             21Ø1     /61,1
1221     1326  Ø177             Ø177
1222     1327  4523             4523     /62,2
1223     133Ø  2151             2151
1224     1331  4122             4122     /63,3
1225     1332  2651             2651
1226     1333  2414             2414     /64,4
1227     1334  Ø477             Ø477
123Ø     1335  5172             5172     /65,5
1231     1336  Ø651             Ø651
1232     1337  15Ø6             15Ø6     /66,6
1233     134Ø  4225             4225
1234     1341  4443             4443     /67,7
1235     1342  6Ø5Ø             6Ø5Ø
1236     1343  ØØØØ   RDORA,    Ø
1237     1344  73ØØ             CLA CLL            /SAVE USER DF
124Ø     1345  6214             RDF
1241     1346  12Ø2             TAD ACDFØ
1242     1347  3257             DCA RTNCDF
1243     135Ø  6141             LINC
1244                            LMODE
1245     1351  Ø1Ø1   CSAM,     CURSAM             /CURSOR
1246     1352  Ø341             SCR 1              /9 BITS COVERS
1247     1353  ØØØ2             PDP                /SCOPE
125Ø                            PMODE              /MAKE RANGE
1251     1354  1243             TAD P4Ø1           /-1 TO -1ØØØ
1252     1355  7141             CIA CLL
1253     1356  6141             LINC
1254                            LMODE
1255     1357  5573             STC CURCNT&1777
1256     136Ø  Ø1ØØ   WSAM,     WINSAM             /WINDOW
1257     1361  ØØØØ   MOVDIS,   Ø                  /SCR 4 OR CLR
126Ø     1362  ØØ61             SET I XCORD
1261     1363  ØØØØ   LEFTX,    Ø                  /LEFT COORD (    ISPLAY
1262     1364  74ØØ             JMP CONT&1777
```

```
1365  0105  FREE,   FRESAM
1366  0341          SCR 1
1367  0002          PDP
                    PMODE
1370  3115          DCA YCUR
1371  1115          TAD YCUR
1372  6141          LINC
1373  6000          6000            /JMP Ø
                    PAGE
1400  0002  CONT,   2
1401  6201  CCDFØ,  CDF Ø
1402  3347          DCA DBLLO       /PUT KNOB VAL
1403  1347          TAD DBLLO       /IN DAC
1404  7710          SPA CLA         /PROPAGATE SIGN
1405  7040          CMA             /BIT HI ORD
1406  3341          DCA DBLHI
1407  4304          JMS DADD
1410  1347          TAD DBLLO       /UPDATE WIN ADDR
1411  3375          DCA BUFLO
1412  1341          TAD DBLHI
1413  3374          DCA BUFHI
                                    /MUST CHK
                                    /WINDOW SA
                                    /WITH BOUNDS
                                    /TO MAINTAIN
                                    /BUFFER RING

1414  4316          JMS BOUND       /LOWER BOUND
1415  0001  MINFLD, 1
1416  0000  MINADR, Ø
1417  7700          SMA CLA         /LOW END WRAP?
1420  5273          JMP CHKHI       /NO
1421  1274          TAD MAXFLD      /RESET TO
1422  3374          DCA BUFHI       /UPPER BOUND
1423  1275          TAD MAXADR
1424  3375  WRAP,   DCA BUFLO
1425  4304          JMS DADD        /CORRECT WRAP
1426  1347          TAD DBLLO       /CORRECTED
1427  3375          DCA BUFLO       /WINDOW SA
1430  1341          TAD DBLHI
1431  3374          DCA BUFHI
1432  1375  SETFLD, TAD BUFLO       /SET DISPLAY
1433  3304          DCA BUFPTR      /ARGS
1434  1131          TAD MINPTS
1435  3116          DCA COUNT
1436  1374          TAD BUFHI
1437  3316          DCA BOUND
1440  4341          JMS SETDF
1441  1704  NXTPNT, TAD I BUFPTR
1442  1111          TAD YSHFT       /OFF SET
1443  6141          LINC
                    LMODE
1444  0341  YSCAL,  SCR 1           /SCALE FACTOR
1445  0161          DIS I XCORD
1446  0002          PDP
                    PMODE
1447  2373          ISZ CURCNT      /READY TO DIS
                                    /CURSOR ?
1450  7610  CURRTN, SKP CLA         /NO
1451  5351          JMP CURDIS
1452  2376          ISZ ENDLO       /CHK FOR HI
1453  5263          JMP OKEND       /END WRAP
```

```
1455  5263            JMP OKEND
1456  1216            TAD MINADR      /RESET TO
1457  3304            DCA BUFPTR      /LOWER BOUND
1460  1215            TAD MINFLD
1461  3316            DCA BOUND
1462  5266            JMP NXTDF
1463  2304  OKEND,    ISZ BUFPTR      /CHK FOR FIELD
                                      /BOUNDARY
1464  5267            JMP OKFLD       /ITS OK
1465  2316            ISZ BOUND       /SET NXT FLD
1466  4341  NXTDF,    JMS SETDF
1467  2116  OKFLD,    ISZ COUNT       /512 PNTS ?
1470  5241            JMP NXTPNT      /NO
1471  5672            JMP I .+1       /DSC READ OUT
1472  1244            DSCLOC
1473  4316  CHKHI,    JMS BOUND       /CHK UPR BOUND
1474  0002  MAXFLD, 2
1475  0000  MAXADR, 0
1476  7710  M70,      SPA CLA         /HI WRAP ?
1477  5232            JMP SETFLD
1500  1215            TAD MINFLD      /YES
1501  3374            DCA BUFHI       /RESET TO
1502  1216            TAD MINADR      /LOWER BOUND
1503  5224            JMP WRAP
            /DOUBLE PRECISION ADD
            /(DBLHI,DBLLO)+(BUFHI,BUFLO)
            /RESULT IN (DBLHI,DBLLO)
            /(BUFHI,BUFLO)=INITIAL SCOPE ADDRESS

1504  0000  DADD,     0
1505  7300            CLA CLL
1506  1347            TAD DBLLO
1507  1375            TAD BUFLO
1510  3347            DCA DBLLO
1511  7004            RAL
1512  1341            TAD DBLHI
1513  1374            TAD BUFHI
1514  3341            DCA DBLHI
1515  5704            JMP I DADD

            /ADD -UPPER OR -LOWER BOUND
            /TO (BUFHI,BUFLO)
            /BOUND IS AT P+1,P+2 OF CALL

1516  0000  BOUND,    0
1517  1716            TAD I BOUND     /2S COM OF ARG
1520  7140            CMA CLL         /TO DAC
1521  3341            DCA DBLHI
1522  2316            ISZ BOUND
1523  1716            TAD I BOUND
1524  7041            CIA
1525  7430            SZL
1526  2341            ISZ DBLHI
1527  7000  M1000,    NOP
1530  3347            DCA DBLLO
1531  4304            JMS DADD
1532  1341            TAD DBLHI
1533  3377            DCA ENDHI       /DAC HOLDS -NUM
1534  1347            TAD DBLLO       /TO END OF BUF
1535  3376            DCA ENDLO       /NO MATTER FOR
                                      /LOW END WRA
1536  1341            TAD DBLHI       /TO CHK FOR
1537  2316            ISZ BOUND       /UPON RTN
```

```
1540  5716          JMP I BOUND
1541  0000  SETDF,  0                  /SET 8 FIELD
1542  1316          TAD BOUND          /REL TO BOUND
1543  7106          CLL RTL
1544  7004          RAL
1545  1201          TAD CCDF0
1546  3347          DCA .+1
1547  0000  DBLLO,  0
1550  5741          JMP I SETDF
1551  3115  CURDIS, DCA YCUR           /DISP CURSOR
1552  1316          TAD BOUND          /SAVE X,Y
1553  3112          DCA XCURHI         /COORDINATES
1554  1304          TAD BUFPTR
1555  3113          DCA XCURLO
1556  1704          TAD I BUFPTR
1557  3114          DCA CORVAL
1560  1276          TAD M70
1561  3347          DCA DBLLO
1562  1115          TAD YCUR
1563  6141  CURLOP, LINC
                    LMODE
1564  0465          SNS I 5
1565  7365          JMP FREE           /FREE CURSOR
1566  0141          DIS XCORD
1567  0002          PDP
                    PMODE
1570  2347          ISZ DBLLO
1571  5363          JMP CURLOP
1572  5250          JMP CURRTN
1573  0000  CURCNT, 0
            /THESE 5 GUYS MAY BE PAGE 0
1574  0001  BUFHI,  1
1575  0000  BUFLO,  0
1576  0000  ENDLO,  0
1577  0000  ENDHI,  0
                    DBLHI=SETDF
                    BUFPTR=DADD
                    XCORD=1
                    LMODE
                    CURSAM=SAM 1       /CURSOR KNOB
                    WINSAM=SAM 0       /WINDOW KNOB
                    FRESAM=SAM 5       /FREE CURSOR
                    SCALE=SCR
                    SC12BU=SCR 3       /SCALE FACTOR
                                       /12 BIT UNSIGNED
                    OF12BU=4000        /Y OFFSET FOR
                                       /12 BIT UNSIGNED
                    CHAIN "FFTC-2"
```

```
ØØØØ                    *2Ø
2ØØ1                          EJECT
```

```
                        LMODE
                        SEGMNT 2
                        *2Ø
ØØ2Ø  Ø647              LDF     7     /BOOTSTRAP IN DIAL MS I/O ROUTINES
ØØ21  Ø7ØØ              RDC
ØØ22  6322              6322
ØØ23  Ø7ØØ              RDC
ØØ24  7323              7323
ØØ25  Ø643              LDF     3
ØØ26  1Ø2Ø    IFDIAL,   LDA I               /INPUT FROM DIAL TAPE?
ØØ27  2411              QUES1+2ØØØ
ØØ3Ø  Ø6Ø2              LIF 2
ØØ31  672Ø              JMP     ASK
ØØ32  13ØØ              LDH
ØØ33  7Ø43              ANSWER+6ØØØ
ØØ34  146Ø              SAE I
ØØ35  ØØ31              31
ØØ36  Ø456              SKP
ØØ37  6Ø44              JMP     UNTFIL   /DIAL
ØØ4Ø  146Ø              SAE I
ØØ41  ØØ16              16
ØØ42  5Ø26              JMP     IFDIAL  /ERROR
ØØ43  6Ø61              JMP     DATTAP            /DATA
              /ASK FOR UNIT NO + FILE NAME
ØØ44  6523    UNTFIL,   JMP     ASK2
ØØ45  6Ø44              JMP     .-1               /ASK AGAIN
ØØ46  Ø6Ø1              LIF     1
ØØ47  1Ø2Ø              LDA I
ØØ5Ø  2375              FDV+2ØØØ
ØØ51  6Ø2Ø              JMP     2Ø                /SEARCH INDEX
ØØ52  Ø456              SKP                       /CANT FIND IT
ØØ53  6Ø63              JMP     MOVINP
ØØ54  Ø6Ø2              LIF     2
ØØ55  1Ø2Ø              LDA I                     /DISPLAY ERROR MSG
ØØ56  276Ø              MSG1+2ØØØ
ØØ57  672Ø              JMP     ASK
ØØ6Ø  6Ø44              JMP     UNTFIL            /ASK AGAIN
ØØ61  6572    DATTAP,   JMP ASK3
ØØ62  6Ø61              JMP     DATTAP            /ERROR-ASK AGAIN
ØØ63  6166    MOVINP,   JMP FDV2RW          /MOVE INPUT PARAMETERS TO R/W LIST
ØØ64  Ø643    PTS,      LDF     3
ØØ65  Ø6Ø2              LIF     2
ØØ66  1Ø2Ø              LDA I                     /ASK FOR NO OF PTS
ØØ67  2521              QUES4+2ØØØ
ØØ7Ø  672Ø              JMP     ASK
ØØ71  ØØ61              SET I   1
ØØ72  3Ø43              ANSWER+2ØØØ
ØØ73  1Ø2Ø              LDA I               /INPUT IS DECIMAL
ØØ74  ØØ12              12
ØØ75  47Ø1              STC MPLIER
ØØ76  1Ø2Ø              LDA I
ØØ77  77Ø6              -71
Ø1ØØ  4Ø45              STC UPLEGL
Ø1Ø1  1Ø2Ø              LDA I                     /1Ø24 PTS MAX
Ø1Ø2  2ØØØ              2ØØØ
Ø1Ø3  6627              JMP     CONV
Ø1Ø4  6Ø64              JMP     PTS               /ERROR
Ø1Ø5  Ø64Ø              LDF     Ø
Ø1Ø6  1Ø4Ø              STA
Ø1Ø7  2ØØ3              N+2ØØØ
Ø11Ø  Ø643              LDF 3
Ø111  ØØ17              COM
```

```
Ø1Ø2   Ø114  ØØØØ  TEMP1,  Ø
Ø1Ø3   Ø115  Ø456          SKP
Ø1Ø4   Ø116  6Ø64          JMP PTS          /BIT 11=1 NOT POWER OF 2
Ø1Ø5   Ø117  ØØ62          SET I 2
Ø1Ø6   Ø12Ø  7764          -13
Ø1Ø7   Ø121  15ØØ  ROTAT,  SRO              /COUNT NO OF BITS SET
Ø11Ø   Ø122  Ø114          TEMP1
Ø111   Ø123  6126          JMP .+3
Ø112   Ø124  112Ø          ADA I
Ø113   Ø125  ØØØ1          1
Ø114   Ø126  Ø222          XSK I 2
Ø115   Ø127  6121          JMP ROTAT
Ø116   Ø13Ø  112Ø          ADA I
Ø117   Ø131  7776          -1
Ø12Ø   Ø132  ØØ17          COM
Ø121   Ø133  Ø45Ø          AZE
Ø122   Ø134  6Ø64          JMP PTS          />1-NOT POWER OF 2
Ø123   Ø135  4ØØ2          STC 2            /CLEAR
Ø124   Ø136  15ØØ  ROT1,   SRO              /DETERMINE POWER OF 2
Ø125   Ø137  Ø114          TEMP1
Ø126   Ø14Ø  Ø456          SKP
Ø127   Ø141  6144          JMP STAMU
Ø13Ø   Ø142  Ø222          XSK I 2
Ø131   Ø143  6136          JMP RØT1
Ø132   Ø144  1ØØØ  STAMU,  LDA
Ø133   Ø145  ØØØ2          2
Ø134   Ø146  Ø64Ø          LDF     Ø
Ø135   Ø147  1Ø4Ø          STA
Ø136   Ø15Ø  2ØØ4          NU+2ØØØ
Ø137   Ø151  112Ø          ADA I
Ø14Ø   Ø152  7776          -1
Ø141   Ø153  Ø451          APO
Ø142   Ø154  6Ø64          JMP     PTS      /POWER<2
Ø143   Ø155  1ØØØ          LDA                       /COMPUTE NO OF OUTPUT BLKS
Ø144   Ø156  2ØØ3          N+2ØØØ           /NO OF PTS
Ø145   Ø157  Ø241          ROL 1            /*2
Ø146   Ø16Ø  65Ø3          JMP NUMBKS       /CONVERT TO BLKS
Ø147   Ø161  1Ø4Ø          STA
Ø15Ø   Ø162  24Ø4          FDV+2ØØ7         /NO OF BLKS FOR REAL & IMAG
Ø151   Ø163  1Ø4Ø          STA
Ø152   Ø164  241Ø          RWPARM+2ØØ3
Ø153   Ø165  13ØØ          LDH
Ø154   Ø166  3Ø46          ANSWER+2ØØ3
Ø155   Ø167  146Ø          SAE I
Ø156   Ø17Ø  ØØ22          22
Ø157   Ø171  62Ø3          JMP IFCOM
Ø16Ø   Ø172  1Ø4Ø          STA              /REAL-SET FLAG
Ø161   Ø173  2Ø53          REALFG+2ØØØ
Ø162   Ø174  Ø64Ø          LDF Ø
Ø163   Ø175  1ØØØ          LDA
Ø164   Ø176  2ØØ3          N+2ØØØ           /COMPUTE INPUT BLKS
Ø165   Ø177  65Ø3          JMP NUMBKS
Ø166   Ø2ØØ  1Ø4Ø          STA
Ø167   Ø2Ø1  241Ø          RWPARM+2ØØ3
Ø17Ø   Ø2Ø2  6211          JMP CKEND
Ø171   Ø2Ø3  146Ø  IFCOM,  SAE I
Ø172   Ø2Ø4  ØØØ3          3
Ø173   Ø2Ø5  6Ø64          JMP PTS /ERROR
Ø174   Ø2Ø6  ØØ11          CLR              /COMPLEX-CLEAR FLAG
Ø175   Ø2Ø7  1Ø4Ø          STA
Ø176   Ø21Ø  2Ø53          REALFG+2ØØØ
Ø177   Ø211  1ØØØ  CKEND,  LDA              /WILL INPUT RUN OFF END OF TAPE
```

```
0212  2407          RWPARM+2002
0213  1100          ADA
0214  2410          RWPARM+2003
0215  1120          ADA I
0216  6777          -1000
0217  0471          APO I
0220  6064          JMP     PTS     /YES
0221  0002          PDP
                    PMODE
4222  7200          CLA
4223  1003          TAD N  /ADD 1 BLK FOR SCALE FACTOR IF 400 WORDS OR MORE
4224  7104          CLL RAL         /NO OF OUTPUT WRDS = NO OF PTS*2
4225  1107          TAD M400
4226  7700          SMA CLA
4227  2523          ISZ I PFDV7
4230  6141          LINC
                    LMODE
0231  1020  IFFFT,  LDA I           /DO FFT OR JUST DISPLAY?
0232  2625          QUES11+2000
0233  6720          JMP ASK
0234  1300          LDH
0235  7043          ANSWER+6000
0236  1460          SAE I
0237  0004          4
0240  6244          JMP .+4
0241  1060          STA I           /NOT=0 JUST DISPLAY
0242  0000  DISFLG, 0
0243  6251          JMP     FIF
0244  1460          SAE I
0245  0006          6
0246  6231          JMP IFFFT       /ERROR
0247  0011          CLR
0250  4242          STC     DISFLG  /=0 WILL DO TRANSFORM OR INVERSE
0251  1300  FIF,    LDH
0252  7044          ANSWER+6001
0253  1460          SAE I
0254  0024          24
0255  6261          JMP IFI
0256  0011          CLR
0257  4356          STC IFTFLG      /DO FFT
0260  6265          JMP IFDISP
0261  1460  IFI,    SAE I
0262  0011          11
0263  6231          JMP     IFFFT
0264  4356          STC IFTFLG      /DO IFFT
0265  2242  IFDISP, ADD     DISFLG
0266  0470          AZE I
0267  6273          JMP     OUTQES
0270  6466          JMP     FDV2RW  /MOVE OUTPUT PARAMETERS TO R/W
0271  0603          LIF     3       /JUST DISPLAY
0272  6001          JMP     DISPLY
            /GET OUTPUT INFO
0273  1020  OUTQES, LDA I
0274  2571          QUES5+2000
0275  6720          JMP     ASK             /PUT ON DIAL TAPE?
0276  1300          LDH
0277  7043          ANSWER+6000
0300  1460          SAE I
0301  0031          31
0302  0456          SKP
0303  6310          JMP     OUTUNT
0304  1460          SAE I
```

```
0306 6273          JMP OUTQES
0307 6452          JMP     ONDAT           /NO
0310 6523 OUTUNT,  JMP     ASK2    /ASK FOR UNIT NO & FILE NAME
0311 6310          JMP     OUTUNT  /ERROR
0312 0601          LIF     1
0313 1020          LDA I                   /ENTER IN INDEX
0314 2375          FDV+2000
0315 6022          JMP     22
0316 6430          JMP     SAMNAM          /NAME ALREADY USED
0317 6423          JMP     NOSPAC          /NO SPACE
0320 0002 RDDATA,  PDP     /CLEAR DATA BUFFER
                   PMODE
4321 7240          CLA CMA
4322 1051          TAD XRLOC
4323 3010          DCA 10
4324 1067          TAD M4000
4325 3011          DCA 11
4326 6211          CDF1
4327 3410          DCA I 10
4330 2011          ISZ 11
4331 5327          JMP .-2
4332 6201          CDFØ
4333 6212          CIF     10      /READ IN DATA
4334 4534          JMS I   PREAD
4335 6405          RWPARM
4336 6201          CDFØ
4337 7200          CLA
4340 1532          TAD I PRELFG    /REAL OR COMPLEX
4341 7640          SZA CLA
4342 5357          JMP   PROC      /REAL
4343 7040          CMA             /MOVE IMAG PARTS TO 2000
4344 1003          TAD     N       /OLD ADDR = NO OF PTS
4345 3010          DCA     10
4346 1110          TAD     C1777   /NEW ADDR = 2000
4347 3011          DCA     11
4350 1003          TAD     N
4351 7041          CIA
4352 3034          DCA     TEMPR   /CTR
4353 3130          DCA     CMPFLG  /DONT COMPLEMENT
4354 4527          JMS I   PMVPTS  /MOVE THEM
4355 5357          JMP     PROC

4356 0000 IFTFLG, Ø                /Ø=FFT NONØ=IFFT

4357 3532 PROC,    DCA I   PRELFG  /OUTPUT WILL BE COMPLEX REGARDLESS OF INPUT
4360 1356          TAD IFTFLG      /DO IFFT?
4361 7650          SNA CLA
4362 5365          JMP     FT      /NO
4363 4447 IFT,     JMS I DOIFFT
4364 7410          SKP
4365 4446 FT,      JMS I DOFFT
4366 4442          JMS I SORT      /PUT IN SEQUENTIAL ORDER
4367 1053 STSCAL,  TAD SCAL
4370 6211          CDF1
4371 3034          DCA     TEMPR    /SAVE
4372 1003          TAD     N
4373 7104          CLL RAL
4374 3036          DCA     COSINE  /NO OF PTS*2
4375 1034          TAD     TEMPR
4376 3436          DCA I   COSINE    /STORE SCALE FACTOR AFTER DATA
4377 6201 NOWSTR,  CDFØ
4400 1110          TAD     C1777   /OLD ADDR =
4401 3010          DCA     10
```

```
4402  7040          CMA             /NEW ADDR =    OF PTS
4403  1003          TAD     N
4404  3011          DCA     11
4405  1003          TAD     N
4406  7041          CIA
4407  3034          DCA     TEMPR   /CTR
4410  3130          DCA     CMPFLG  /DONT COMPLEMENT
4411  4527          JMS I   PMVPTS  /PACK IMAG PARTS BEHIND REAL
4412  6141          LINC
                    LMODE
0413  6466          JMP FDV2RW
0414  0002          PDP
                    PMODE
4415  6212          CIF     10      /WRITE OUT DATA
4416  4535          JMS I   PWRITE
4417  6405          RWPARM
4420  6141          LINC
                    LMODE
0421  0603          LIF 3
0422  6001          JMP DISPLY
0423  0602  NOSPAC, LIF     2
0424  1020          LDA I
0425  3013          MSG2+2000
0426  6720          JMP     ASK
0427  6273          JMP     OUTQES          /ASK OUTPUT QUESTIONS AGAIN
0430  0602  SAMNAM, LIF     2
0431  1020          LDA I                   /NAME ALREADY EXISTS
0432  2612          QUES6+2000
                                    /REPLACE WITH NEW FILE?
0433  6720          JMP     ASK
0434  1300          LDH
0435  7043          ANSWER+6000
0436  1460          SAE I
0437  0031          31
0440  0456          SKP
0441  6446          JMP     REPL
0442  1460          SAE I
0443  0016          16
0444  6430          JMP     SAMNAM
0445  6310          JMP     OUTUNT  /NO-ASK FOR NAME AGAIN
0446  0601  REPL,   LIF 1
0447  6024          JMP     24
0450  6423          JMP     NOSPAC
0451  6320          JMP     RDDATA
0452  0602  ONDAT,  LIF     2
0453  6572          JMP ASK3        /ASK FOR UNIT/BLK NO
0454  6452          JMP ONDAT       /ERROR
0455  1000          LDA
0456  2403          FDV+2006        /BLK NO
0457  1100          ADA
0460  2404          FDV+2007        /NO OF BLKS
0461  1120          ADA I
0462  6777          -1000
0463  0471          APO I
0464  6423          JMP NOSPAC      /NOT ENOUGH BLKS LEFT
0465  6320          JMP     RDDATA
                    /MOVE FDV PARAMETERS TO R-W LIST
0466  1000  FDV2RW, LDA
0467  2375          FDV+2000
0470  1040          STA
0471  2405          RWPARM+2000
0472  1000          LDA
```

```
                         STA
Ø475  24Ø7               RWPARM+2ØØ2
Ø476  1ØØØ               LDA
Ø477  24Ø4               FDV+2ØØ7
Ø5ØØ  1Ø4Ø               STA
Ø5Ø1  241Ø               RWPARM+2ØØ3
Ø5Ø2  6ØØØ               JMP Ø
            /CONVERT WORDS TO BLOCKS
Ø5Ø3  4114  NUMBKS, STC TEMP1
Ø5Ø4  2ØØØ               ADD Ø
Ø5Ø5  4522               STC NUMBKX
Ø5Ø6  2114               ADD TEMP1
Ø5Ø7  Ø643               LDF     3
Ø51Ø  ØØ61               SET I   1
Ø511  ØØØ1               1
Ø512  112Ø               ADA I
Ø513  7377               -4ØØ
Ø514  Ø451               APO
Ø515  652Ø               JMP .+3
Ø516  Ø221               XSK I   1
Ø517  6512               JMP      .-5
Ø52Ø  1ØØØ               LDA
Ø521  ØØØ1               1
Ø522  ØØØØ  NUMBKX, Ø
            /ASK FOR UNIT NUMBER & FILE NAME
            /CONV & STORE UNIT NUMBER
            /MOVE FILE NAME TO ENTER, LOOKUP PARAMETER LIST
            /STORE UNIT THRU B3
Ø523  1ØØØ  ASK2,   LDA
Ø524  ØØØØ          Ø
Ø525  4571          STC     ASK2X
Ø526  Ø6Ø2          LIF     2
Ø527  6711          JMP     OCTL     /CHANGE PARAMETERS TO HANDLE OCTAL NUMBERS
Ø53Ø  1Ø2Ø          LDA I
Ø531  2453          QUES2+2ØØØ
Ø532  672Ø          JMP     ASK      /PT TO UNIT NO-1H
Ø533  ØØ61          SET I   1
Ø534  3Ø43          ANSWER+2ØØØ
Ø535  1Ø2Ø          LDA I            /MAX VALUE
Ø536  ØØ17          17
Ø537  6627          JMP CONV
Ø54Ø  6571          JMP     ASK2X    /ERROR
Ø541  1Ø4Ø          STA              /STORE UNIT
Ø542  2375          FDV+2ØØØ
            /MOVE FILE NAME FROM ANSWER BUFFER TO LOOKUP, ENTER PARAMETER LIST
Ø543  ØØ61          SET I 1
Ø544  7Ø44          ANSWER+6ØØ1
Ø545  ØØ62          SET I   2
Ø546  6375          FDV+6ØØØ                /LEFT HALF 1ST OF FDV+1
Ø547  ØØ63          SET I   3
Ø55Ø  7767          -1Ø              /8 CHARS
Ø551  1321          LDH I   1        /IF 1ST CHAR OF NAME
Ø552  Ø47Ø          AZE I            /=ØØ, NO NAME WAS
Ø553  6571          JMP     ASK2X    /ENTERED-ERROR
Ø554  Ø456          SKP
Ø555  1321  INFILE, LDH I   1
Ø556  Ø45Ø          AZE                     /FILL TO 8 CHARS WITH 77
Ø557  6562          JMP     .+3
Ø56Ø  132Ø          LDH I
Ø561  77ØØ          77ØØ
Ø562  1362          STH I   2
Ø563  Ø223          XSK I   3
Ø564  6555          JMP     INFILE
```

```
0565  1020            LDA I                       /RETURN CALL+2
0566  0001            1
0567  1140            ADM
0570  0571            .+1
0571  0000  ASK2X,    0
                /ASK FOR UNIT NUMBER + BLK NO AND CONVERT
                /STORE UNIT THRU B7
                /"  BLK NO " B10
0572  1000  ASK3,     LDA
0573  0000            0
0574  4626            STC     ASK3X
0575  0602            LIF     2
0576  1020            LDA I
0577  2475            QUES3+2000
0600  6720            JMP     ASK
0601  0061            SET I   1                   /ADDR-1H OF 1ST CHAR - UNIT
0602  3043            ANSWER+2000
0603  6711            JMP OCTL          /CHANGE PARAMETERS TO HANDLE OCTAL
0604  1020            LDA I                       /MAX VALUE
0605  0017            17
0606  6627            JMP CONV
0607  6626            JMP ASK3X         /ERROR
0610  1040            STA
0611  2375            FDV+2000
0612  0061            SET I 1
0613  7044            ANSWER+6001
0614  1020            LDA I             /MAX VALUE FOR BLK NO
0615  0777            777
0616  6627            JMP     CONV
0617  6626            JMP     ASK3X
0620  1040            STA
0621  2403            FDV+2006
0622  1020            LDA I                       /EXIT CALL+2
0623  0001            1
0624  1140            ADM
0625  0626            .+1
0626  0000  ASK3X,    0
                /CONVERT NUMBER IN ANSWER BUFFER TO BINARY
                /ENTER WITH MAX LEGAL VALUE IN AC
                /IF LEGAL - EXIT CALL+2 WITH VALUE IN AC
0627  0017  CONV,     COM
0630  4675            STC     TEMP2 /COMPLEMENT MAX VALUE
0631  4114            STC     TEMP1
0632  2000            ADD     0                   /RETURN ADDR
0633  4674            STC     CONVER
0634  1321  NXTCHR,   LDH I   1                   /GET A CHAR
0635  0470            AZE I
0636  6660            JMP ERRCHK
0637  1120            ADA I
0640  7720            7720                        /-60 -2
                /S  COMP
0641  0451            APO
0642  6650            JMP     CHKEND              /<60 MAYBE TERMINATING CHAR
0643  1301            LDH     1                   /CK UPPER LIMIT
0644  1120            ADA I
0645  7710  UPLEGL,   -67               /UPPER LIM=7 OR 9
0646  0451            APO
0647  6676            JMP     MULPLY              /0-9 -0K
0650  1301  CHKEND,   LDH     1                   /<9 MAYBE TERMINATING CHAR
0651  1460            SAE I
0652  0034            34
0653  0456            SKP
```

```
0655  1460            SAE I
0656  0074            74
0657  6674            JMP     CONVER            /ILLEGAL CHAR
0660  1000  ERRCHK,   LDA                       /=34 OR 74 - NUMBER COMPLETED
0661  0114            TEMP1
0662  1100            ADA                       /ERROR CHECK SIZE
0663  0675            TEMP2
0664  0471            APO I
0665  6674            JMP     CONVER            /TOO LARGE
0666  1020            LDA I                     /OK STEP EXIT
0667  0001            1
0670  1140            ADM
0671  0674            .+3
0672  1000            LDA                       /EXIT WITH VALUE IN AC
0673  0114            TEMP1
0674  0000  CONVER,   Ø
0675  0000  TEMP2,    Ø
            /
0676  1000  MULPLY,   LDA             /VALUE SO FAR
0677  0114            TEMP1
0700  1260            MUL I
0701  0010  MPLIER,   1Ø
0702  4114            STC     TEMP1
0703  1301            LDH     1                 /+ THIS VALUE
0704  1560            BCL I
0705  7760            776Ø
0706  1140            ADM
0707  0114            TEMP1
0710  6634            JMP     NXTCHR
            /
                      /CHANGE PARAMETERS SO CONV & MULPLY WILL HANDLE OCTAL NUMBERS
0711  1020  OCTL,     LDA I
0712  0010            1Ø
0713  4701            STC     MPLIER
0714  1020            LDA I
0715  7710            -67
0716  4645            STC     UPLEGL
0717  6000            JMP     Ø
            /
            /DISPLAY QUESTIONS
0720  4734  ASK,      STC     QUESNO            /ADDR OF TEXT
0721  2000            ADD     Ø
0722  4740            STC     ASKX
0723  0500            IOB
                      PMODE
4724  6234            RIB
                      LMODE
0725  0343            SCR 3
0726  1560            BCL I
0727  7740            774Ø
0730  1120            ADA I
0731  0600            LIF Ø
0732  4737            STC ASKX-1
0733  7000            JMP     QAINIT            /DISPLAY
0734  0000  QUESNO,   Ø
0735  3043            ANSWER+2ØØØ
0736  7053            JMP     QARFSH            /WAIT FOR ANSWERS
0737  0000            Ø
0740  0000  ASKX,     Ø

                      EJECT
```

```
                        SEGMNT 3
                        LMODE
                        *1
ØØØ1  ØØØ2  DISPLY,     PDP
                        PMODE
6ØØ2  72ØØ              CLA
6ØØ3  1ØØ3              TAD     N
6ØØ4  711Ø              CLL RAR
6ØØ5  3Ø37              DCA     GR      /NO OF PTS/2
6ØØ6  1Ø37              TAD     GR
6ØØ7  7Ø41              CIA
6Ø1Ø  3Ø33              DCA     ADD2    /-NO OF PTS/2
6Ø11  114Ø              TAD     C2ØØØ
6Ø12  3364              DCA     LOADDR  /LOWER ADDR OF DISPLAY
6Ø13  111Ø              TAD     C1777
6Ø14  1ØØ3              TAD     N
6Ø15  3366              DCA     UPADDR  /UPPER ADDR OF DISPLAY
6Ø16  6212              CIF     1Ø      /READ IN DATA
6Ø17  4534              JMS I   PREAD
6Ø2Ø  64Ø5              RWPARM
6Ø21  6141              LINC
                        LMODE
ØØ22  1Ø2Ø  DISPL1,     LDA I           /WHICH DISPLAY
ØØ23  2665              QUES13+2ØØØ
ØØ24  Ø6Ø2              LIF 2
ØØ25  672Ø              JMP ASK
ØØ26  13ØØ              LDH
ØØ27  7Ø43              ANSWER+6ØØØ
ØØ3Ø  Ø45Ø              AZE
ØØ31  6Ø34              JMP .+3
ØØ32  Ø6Ø2              LIF 2
ØØ33  6Ø26              JMP IFDIAL      /LINE FEED
ØØ34  ØØØ2              PDP
                        PMODE
6Ø35  1Ø77  WCHDIS,     TAD M11
6Ø36  745Ø              SNA
6Ø37  5255              JMP DPIMAG      /IMAG
6Ø4Ø  1Ø75              TAD M4
6Ø41  745Ø              SNA
6Ø42  5476              JMP I PDPMAG    /MAGNITUDE
6Ø43  11ØØ              TAD M5
6Ø44  745Ø              SNA
6Ø45  5317              JMP DPREAL      /REAL
6Ø46  1Ø7Ø              TAD M1
6Ø47  765Ø              SNA CLA
6Ø5Ø  5654              JMP I PDPSCL    /SCALE FACTOR
6Ø51  6141  DISPER,     LINC            /ERROR
                        LMODE
ØØ52  6Ø22              JMP DISPL1
ØØ53  ØØØØ  REALFG,     Ø
                        PMODE
6Ø54  6226  PDPSCL,     DPSCAL
6Ø55  1253  DPIMAG,     TAD REALFG
6Ø56  764Ø              SZA CLA
6Ø57  5251              JMP DISPER      /NO IMAG PARTS TO DISPLAY
6Ø6Ø  1533              TAD I   PIFTFG  /IF TRANSFORM WAS DONE, SWAP HALVES
6Ø61  764Ø              SZA CLA
6Ø62  531Ø              JMP     NOSWPI  /INVERSE WAS DONE
6Ø63  7Ø4Ø              CMA
6Ø64  1ØØ3              TAD     N
6Ø65  3Ø1Ø              DCA     1Ø      /OLD LOW ADDR OF 1ST 1/2 = NO OF PTS
6Ø66  1Ø37              TAD     GR      /NEW LOW ADDR OF 1ST 1/2 = 2ØØØ + NO OF PTS/2
6Ø67  111Ø
```

```
6Ø7Ø  3Ø11           DCA     11
6Ø71  1Ø33           TAD     ADD2
6Ø72  3Ø34           DCA     TEMPR    /MOVE 1/2 OF PTS
6Ø73  72Ø1           CLA IAC
6Ø74  313Ø           DCA     CMPFLG   /COMPLEMENT VALUES
6Ø75  4527           JMS I   PMVPTS   /MOVE THEM
6Ø76  7Ø4Ø           CMA              /OLD ADDR OF 2ND 1/2 = 3/2 NO OF PTS
6Ø77  1Ø37           TAD     GR
61ØØ  1ØØ3           TAD     N
61Ø1  3Ø1Ø           DCA     1Ø
61Ø2  111Ø           TAD     C1777    /NEW ADDR OF 2ND 1/2 = 2ØØØ
61Ø3  3Ø11           DCA     11
61Ø4  1Ø33           TAD     ADD2
61Ø5  3Ø34           DCA     TEMPR    /1/2 OF PTS
61Ø6  4527           JMS I   PMVPTS   /MOVE THEM - 1ST 1/2 IS NOW 2ND 1/2; 2ND 1/2 IS NOW 1ST 1/2
61Ø7  533Ø           JMP     PREPAR
611Ø  1ØØ3  NOSWPI,  TAD     N        /LOW ADDR OF IMAG = NO OF PTS
6111  3364           DCA     LOADDR
6112  7Ø4Ø           CMA              /HIGH ADDR = 2*NO OF PTS-1
6113  1ØØ3           TAD     N
6114  1ØØ3           TAD     N
6115  3366           DCA     UPADDR
6116  533Ø           JMP     PREPAR

6117  1533  DPREAL,  TAD I   PIFTFG   /IF TRANSFORM WAS DONE, SWAP HALVES
612Ø  764Ø           SZA CLA
6121  5324           JMP     NOSWPR
6122  4526           JMS I   PMRLMG   /SWAP
6123  533Ø           JMP     PREPAR
6124  3364  NOSWPR,  DCA     LOADDR   /LOW ADDR OF REAL
6125  7Ø4Ø           CMA
6126  1ØØ3           TAD     N
6127  3366           DCA     UPADDR   /HIGH ADDR = NO OF PTS-1

613Ø  1Ø33  PREPAR,  TAD ADD2         /NO OF PTS <1ØØØ?
6131  71Ø4           CLL RAL
6132  1137           TAD C1ØØØ
6133  755Ø           SPA SNA
6134  5347           JMP GQ1ØØØ
6135  711Ø           CLL RAR          /YES
6136  7ØØ1           IAC              /CENTER DISPLAY
6137  1141           TAD M1K
614Ø  3525           DCA I PLEFTX     /1ØØØ-(1ØØØ-NO OF PTS/2) 1,S COMP
6141  1146           TAD CCLR
6142  3524           DCA I PMVDIS
6143  1Ø33           TAD ADD2         /WIDTH OF DISPLAY
6144  71Ø4           CLL RAL          /NO OF PTS
6145  3131           DCA MINPTS
6146  5362           JMP SHOWIT
6147  72ØØ  GQ1ØØØ,  CLA
615Ø  1141           TAD M1K          /LEFT JUSTIFY DISPLAY
6151  3525           DCA I PLEFTX     /-1ØØØ 1,S COMP
6152  1525           TAD I PLEFTX     /WIDTH OF DISPLAY
6153  7ØØ1           IAC
6154  3131           DCA MINPTS
6155  1145           TAD SCR4         /MOVE DISPLAY
6156  3524           DCA I PMVDIS
6157  5362           JMP SHOWIT
            /DISPLAY DATA
616Ø  6141  REDPLY,  LINC
                     LMODE
6161  6ØØ1           JMP DISPLY
                     PMODE
```

```
6162  4517  SHOWPT, JMS I KIDORA
6163  ØØØ1          1               /LOW ADDR FIELD
6164  ØØØØ  LOADDR, Ø               / "    "     "
6165  ØØØ1          1               /HIGH   "     "
6166  ØØØØ  UPADDR, Ø               / "     "
6167  ØØØØ          Ø               /Y OFFSET
                    LMODE
Ø17Ø  Ø343  SIZE,   SCR 3           /SCALE
                    PMODE
6171  452Ø  RFRSH,  JMS I KRDORA    /REFRESH UNTIL LF IS HIT
6172  6Ø31          KSF
6173  5371          JMP .-2
6174  6Ø36          KRB
6175  11Ø2          TAD M215
6176  765Ø          SNA CLA
6177  536Ø          JMP REDPLY
62ØØ  6Ø36          KRB
62Ø1  11Ø6          TAD M261
62Ø2  765Ø          SNA CLA
62Ø3  5216          JMP LARGER
62Ø4  6Ø36          KRB
62Ø5  11Ø3          TAD M321
62Ø6  765Ø          SNA CLA
62Ø7  5211          JMP SMALLR
621Ø  5522          JMP I PRFRSH
6211  1536  SMALLR, TAD I KYSCAL
6212  11Ø4          TAD M353
6213  771Ø          SPA CLA
6214  2536          ISZ I KYSCAL
6215  5522          JMP I PRFRSH
6216  1536  LARGER, TAD I KYSCAL
6217  11Ø5          TAD M34Ø
622Ø  775Ø          SPA SNA CLA
6221  5522          JMP I PRFRSH
6222  7Ø4Ø          CMA
6223  1536          TAD I KYSCAL
6224  3536          DCA I KYSCAL
6225  5522          JMP I PRFRSH

                    /DISPLAY SCALE FACTOR

6226  1532  DPSCAL, TAD I   PRELFG
6227  764Ø          SZA CLA
623Ø  5772          JMP I   PDSPER  /JUST REAL MEANS I DIDNT MAKE FILE - NO SCALE FACTOR
6231  1ØØ3          TAD     N
6232  71Ø4          CLL RAL
6233  3Ø34          DCA     TEMPR   /ADDR = NO OF PTS*2
6234  6211          CDF1
6235  1434          TAD I   TEMPR
6236  1Ø77          TAD     M11
6237  774Ø          SMA SZA CLA
624Ø  5244          JMP     GR9     />9
6241  1Ø74          TAD     LESS1Ø
6242  1434          TAD I   TEMPR   /SPACE + ASCII SCALE FACTOR
6243  5247          JMP     SHOSCL
6244  1Ø71  GR9,    TAD     M12
6245  1434          TAD I   TEMPR
6246  1Ø73          TAD     GRET1Ø  /1Ø+SCALE FACTOR-1Ø
6247  62Ø1  SHOSCL, CDFØ
625Ø  3266          DCA     DPMAG-2 /STORE IN DISPLAY PARAMETERS
6251  6141          LINC
```

```
0252  1020           LDA I
0253  2257           SCLFAC+2000
0254  0602           LIF 2
0255  6720           JMP      ASK        /DISPLAY IT
0256  6001           JMP      DISPLY
           SCLFAC,   TEXT Z
0257  4347

0260  4043

0261  4043
0262  0640
0263  4040
0264  4040
0265  4040
0266  4040
0267  3400
           F             \Z

                     PMODE

           /COMPUTE MAGNITUDE
6270  1532 DPMAG,    TAD I PRELFG
6271  7640           SZA CLA
6272  5772           JMP I PDSPER
6273  7040           CMA
6274  1003           TAD      N
6275  3010           DCA      10         /LOW ADDR OF IMAG = NO OF PTS
6276  1110           TAD      C1777
6277  3011           DCA      11         /MOVE TO 2000
6300  1003           TAD      N
6301  7041           CIA
6302  3034           DCA      TEMPR      /CTR
6303  3130           DCA      CMPFLG     /DONT COMPLEMENT VALUES
6304  4527           JMS I    PMVPTS     /MOVE IMAG TO 2000
6305  3036           DCA      COSINE     /ADDR OF 1ST REAL
6306  1110           TAD      C1777      /ADDR-1 OF IMAG PARTS
6307  3011           DCA      11
6310  1101           TAD      C6000      /FRAC MULT
6311  3324           DCA      RELPTR
6312  1101           TAD      C6000
6313  3336           DCA      IMGPTR
6314  1033           TAD      ADD2
6315  7104           CLL RAL
6316  3034           DCA      TEMPR      /-NO OF PTS
6317  6211           CDF1
6320  1436 NXTMAG,   TAD I    COSINE     /REAL PART
6321  6141           LINC
                     LMODE
0322  0644           LDF      4          /FIELD OF REAL
0323  1240           MUL                 /MULT BY ITSELF
0324  0000 RELPTR,   0
0325  0363           SCR I    3          /1 BECAUSE PROD IS SHIFTED LEFT 1, 2 BECAUSE MAX VALUE WILL OVERFLOW - TAKE OUT 2
           SQUARED
0326  0002           PDP
                     PMODE
6327  3143           DCA      DPSQ+1     /SAVE D.P. SQ
6330  7501           MQA
6331  3142           DCA      DPSQ
6332  1411           TAD I    11         /IMAG PART
6333  6141           LINC
                     LMODE
0334  0645           LDF      5          /FIELD OF IMAG
```

```
Ø335  124Ø          MUL
Ø336  ØØØØ  IMGPTR, Ø
Ø337  Ø363          SCR I   3       /SAME REASON AS REAL
Ø34Ø  ØØØ2          PDP
                    PMODE
6341  3Ø35          DCA     SINE    /SAVE H.O.
6342  71ØØ          CLL
6343  75Ø1          MQA             /L.O. OF IMAG
6344  1142          TAD     DPSQ    /L.O. OF REAL
6345  3142          DCA     DPSQ
6346  7ØØ4          RAL             /OVERFLOW IF ANY
6347  1Ø35          TAD     SINE    /H.O. OF IMAG
635Ø  1143          TAD     DPSQ+1  /H.O. OF REAL
6351  3143          DCA     DPSQ+1
6352  4773          JMS I   PSQRT   /TAKE SQ RT
6353  3436          DCA I   COSINE  /STORE IN PLACE OF REAL
6354  2Ø36          ISZ     COSINE  /STEP REAL PTR
6355  2336          ISZ     IMGPTR  / "   IMAG ADDR - ON LAST PT OF 1Ø24 PTS WILL SKIP
6356  2324          ISZ     RELPTR  /STEP REAL ADDR
6357  2Ø34          ISZ     TEMPR   /STEP CTR
636Ø  532Ø          JMP     NXTMAG
6361  6141          LINC
                    LMODE
Ø362  Ø643          LDF     3
Ø363  ØØØ2          PDP
                    PMODE
6364  1533          TAD I   PIFTFG  /IF TRANSFORM WAS DONE, SWAP HALVES
6365  764Ø          SZA CLA
6366  5774          JMP I   PNSWPR
6367  4526          JMS I   PMRLMG
637Ø  5771          JMP I   PPREPR  /DISPLAY MAG
            /
6371  613Ø  PPREPR, PREPAR
6372  6Ø51  PDSPER, DISPER
6373  7Ø52  PSQRT,  SQRT
6374  6124  PNSWPR, NOSWPR
                    EJECT
```

```
                      LMODE
               /LOOKUP, ENTER PARAMETER LIST
ØØ375  ØØØØ    FDV,   Ø             /UNIT Ø
Ø376   ØØØØ           Ø             /FILE NAME - 8 CHAR
Ø377   ØØØØ           Ø
Ø4ØØ   ØØØØ           Ø
Ø4Ø1   ØØØØ           Ø
Ø4Ø2   ØØØ2           2             /BINARY
Ø4Ø3   ØØØØ           Ø             /BLK NO
Ø4Ø4   ØØØØ           Ø             /NO OF BLKS
               /
Ø4Ø5   ØØØØ    RWPARM, ØØØØ         /UNIT
Ø4Ø6   ØØ2Ø           ØØ2Ø          /BUFFER ADDR
Ø4Ø7   ØØØØ           Ø             /BLK NO
Ø41Ø   ØØØØ           Ø             /NO OF BLKS
               /
                      LMODE
               /QUESTIONS
Ø411   4Ø4Ø
Ø412   4Ø4Ø
Ø413   4Ø23
Ø414   1116
Ø415   Ø714
Ø416   Ø54Ø
Ø417   2Ø22
Ø42Ø   Ø5Ø3
Ø421   1123
Ø422   1117
Ø423   164Ø
Ø424   Ø6Ø6
               QUES1,  TEXT Z     SINGLE PRECISION FFT
Ø425   2443

Ø426   474Ø
Ø427   4347

Ø43Ø   4Ø43
Ø431   Ø64Ø
Ø432   4Ø11
Ø433   162Ø
Ø434   2524
Ø435   4Ø17
               F  INPUT ON
Ø436   1643

Ø437   474Ø
Ø44Ø   43Ø6
Ø441   4Ø4Ø
Ø442   Ø411
Ø443   Ø114
Ø444   4Ø25
Ø445   1611
Ø446   2477
Ø447   4Ø31
Ø45Ø   5716
Ø451   7461
Ø452   34ØØ
               F  DIAL UNIT? Y/N<1\Z
               QUES2,  TEXT Z
Ø453   4Ø43
Ø454   Ø625
Ø455   1611
```

```
1450    Ø457  1625
1450    Ø46Ø  15Ø2
1450    Ø461  Ø522
1450                FUNIT NUMBER<2
1451    Ø462  7462
1451    Ø463  4347
1451
1452    Ø464  4Ø43
1452    Ø465  Ø6Ø6
1452    Ø466  1114
1452    Ø467  Ø54Ø
1452    Ø47Ø  16Ø1
1452    Ø471  15Ø5
1452    Ø472  4Ø4Ø
1452    Ø473  747Ø
1452    Ø474  34ØØ
1452                FFILE NAME  <8\Z
1453                QUES3,   TEXT Z
1454    Ø475  4Ø43
1454    Ø476  Ø64Ø
1454    Ø477  4Ø25
1454    Ø5ØØ  1611
1454    Ø5Ø1  244Ø
1454    Ø5Ø2  1625
1454    Ø5Ø3  15Ø2
1454    Ø5Ø4  Ø522
1454                F  UNIT NUMBER<2
1455    Ø5Ø5  7462
1455    Ø5Ø6  4347
1455
1456    Ø5Ø7  4Ø43
1456    Ø51Ø  Ø64Ø
1456    Ø511  4ØØ2
1456    Ø512  1413
1456    Ø513  4Ø16
1456    Ø514  2515
1456    Ø515  Ø2Ø5
1456    Ø516  224Ø
1456    Ø517  7463
1456    Ø52Ø  34ØØ
1456                F  BLK NUMBER <3\Z
1457                QUES4,   TEXT Z
146Ø    Ø521  4Ø43
146Ø    Ø522  Ø61Ø
146Ø    Ø523  1727
146Ø    Ø524  4Ø15
146Ø    Ø525  Ø116
146Ø    Ø526  314Ø
146Ø    Ø527  2Ø24
146Ø    Ø53Ø  2377
146Ø                FHOW MANY PTS? <4
1461    Ø531  7464
1461    Ø532  4347
1461
1462    Ø533  4Ø43
1462    Ø534  474Ø
1462    Ø535  5Ø64
1462    Ø536  5561
1462    Ø537  6Ø62
1462    Ø54Ø  644Ø
1462    Ø541  Ø231
```

```
1462    Ø545  234Ø
1462    Ø546  17Ø6
1462    Ø547  4Ø62
1462                          (4=1Ø24 BY POWERS OF 2)
1463    Ø55Ø  5143
1463
1464    Ø551  474Ø
1464    Ø552  43Ø6
1464    Ø553  22Ø5
1464    Ø554  Ø114
1464    Ø555  4Ø17
1464              FREAL OR
1465    Ø556  2243
1465
1466    Ø557  474Ø
1466    Ø56Ø  43Ø6
1466    Ø561  Ø317
1466    Ø562  152Ø
1466    Ø563  14Ø5
1466    Ø564  3Ø77
1466    Ø565  4Ø22
1466    Ø566  57Ø3
1466    Ø567  7461
1466    Ø57Ø  34ØØ
1466              FCOMPLEX? R/C<1\Z
1467              QUES5,   TEXT Z
1470    Ø571  43Ø6
1470    Ø572  4Ø4Ø
1470    Ø573  1725
1470    Ø574  242Ø
1470    Ø575  2524
1470    Ø576  4Ø17
1470              F  OUTPUT ON
1471    Ø577  1643
1471    Ø6ØØ  Ø64Ø
1471    Ø6Ø1  4ØØ4
1471    Ø6Ø2  11Ø1
1471    Ø6Ø3  144Ø
1471    Ø6Ø4  2516
1471    Ø6Ø5  1124
1471    Ø6Ø6  774Ø
1471    Ø6Ø7  3157
1471    Ø61Ø  1674
1471    Ø611  6134
1471              F  DIAL UNIT? Y/N<1\Z
1472              QUES6,   TEXT Z
1473    Ø612  4Ø43
1473
1474    Ø613  474Ø
1474    Ø614  43Ø6
1474    Ø615  4Ø22
1474    Ø616  Ø52Ø
1474    Ø617  14Ø1
1474    Ø62Ø  Ø3Ø5
1474    Ø621  774Ø
1474    Ø622  3157
1474    Ø623  1674
1474    Ø624  6134
1474              F REPLACE? Y/N<1\Z
1475              QUES11, TEXT Z
1476    Ø625  43Ø6
1476    Ø626  Ø6Ø6
```

```
1476       Ø627  244Ø
147        Ø63Ø  1722
147        Ø631  4ØØ4
1476       Ø632  1123
1476       Ø633  2Ø14
1476       Ø634  Ø131
1476       Ø635  774Ø
1476       Ø636  Ø657
1476       Ø637  Ø474
1476                    FFFT OR DISPLAY? F/D<1
1477       Ø64Ø  6143
1477
15ØØ       Ø641  474Ø
15ØØ       Ø642  4347
15ØØ
15Ø1       Ø643  4Ø43
15Ø1       Ø644  Ø624
15Ø1       Ø645  22Ø1
15Ø1       Ø646  1623
15Ø1       Ø647  Ø617
15Ø1       Ø65Ø  2215
15Ø1       Ø651  4Ø17
15Ø1                    FTRANSFORM OR
15Ø2       Ø652  2243
15Ø2
15Ø3       Ø653  474Ø
15Ø3       Ø654  43Ø6
15Ø3       Ø655  1116
15Ø3       Ø656  26Ø5
15Ø3       Ø657  2223
15Ø3       Ø66Ø  Ø577
15Ø3       Ø661  4Ø24
15Ø3       Ø662  5711
15Ø3       Ø663  7461
15Ø3       Ø664  34ØØ
15Ø3                    FINVERSE? T/I<1\Z
15Ø4                    QUES13, TEXT Z
15Ø5       Ø665  43Ø6
15Ø5       Ø666  271Ø
15Ø5       Ø667  11Ø3
15Ø5       Ø67Ø  1Ø4Ø
15Ø5       Ø671  Ø411
15Ø5       Ø672  232Ø
15Ø5       Ø673  14Ø1
15Ø5       Ø674  3177
15Ø5                    FWHICH DISPLAY?<1
15Ø6       Ø675  7461
15Ø6       Ø676  4347
15Ø6
15Ø7       Ø677  4Ø43
15Ø7       Ø7ØØ  474Ø
15Ø7       Ø7Ø1  4Ø4Ø
15Ø7       Ø7Ø2  4Ø4Ø
15Ø7       Ø7Ø3  225Ø
15Ø7       Ø7Ø4  Ø5Ø1
15Ø7                             R(EAL)
151Ø       Ø7Ø5  1451
151Ø       Ø7Ø6  4347
151Ø       Ø7Ø7  4Ø4Ø
151Ø       Ø71Ø  4Ø4Ø
151Ø       Ø711  4Ø11
151Ø       Ø712  5Ø15
```

```
1510   Ø714  1116
1510   Ø715  Ø122
1510                             I(MAGINARY)
1511   Ø716  3151
1511   Ø717  4347
1511   Ø72Ø  4Ø4Ø
1511   Ø721  4Ø4Ø
1511   Ø722  4Ø15
1511   Ø723  5ØØ1
1511   Ø724  Ø716
1511   Ø725  1124
1511   Ø726  25Ø4
1511                             M(AGNITUDE)
1512   Ø727  Ø551
1512   Ø73Ø  4347
1512   Ø731  4Ø4Ø
1512   Ø732  4Ø4Ø
1512   Ø733  4Ø23
1512   Ø734  5ØØ3
1512   Ø735  Ø114
1512   Ø736  Ø54Ø
1512   Ø737  Ø6Ø1
1512   Ø74Ø  Ø324
1512   Ø741  1722
1512                             S(CALE FACTOR)
1513   Ø742  5143
1513   Ø743  474Ø
1513   Ø744  4Ø4Ø
1513   Ø745  4Ø4Ø
1513   Ø746  1411
1513   Ø747  16Ø5
1513   Ø75Ø  4ØØ6
1513   Ø751  Ø5Ø5
1513   Ø752  Ø45Ø
1513   Ø753  22Ø5
1513   Ø754  2324
1513   Ø755  Ø122
1513   Ø756  2451
1513   Ø757  34ØØ
1513                             LINE FEED(RESTART)\Z
1514               /MESSAGES
1515               MSG1,   TEXT Z
1516   Ø76Ø  4347
1516
1517   Ø761  4Ø43
1517   Ø762  Ø64Ø
1517   Ø763  4Ø4Ø
1517   Ø764  4ØØ3
1517   Ø765  Ø116
1517   Ø766  1617
1517   Ø767  244Ø
1517   Ø77Ø  Ø611
1517               F     CANNOT FIND
152Ø   Ø771  16Ø4
152Ø   Ø772  4347
152Ø
1521   Ø773  4Ø43
1521   Ø774  474Ø
1521   Ø775  4Ø4Ø
1521   Ø776  4Ø4Ø
1521   Ø777  4Ø4Ø
1521   1ØØØ  4Ø4Ø
1521   1ØØ1  4Ø1Ø
```

```
1521      1ØØ2  1124
1521      1ØØ3  4Ø22
1521      1ØØ4  Ø524
1521      1ØØ5  2522
1521      1ØØ6  164Ø
1521      1ØØ7  2417
1521      1Ø1Ø  4ØØ3
1521      1Ø11  1716
1521      1Ø12  2434
1521                                 HIT RETURN TO CONT\Z
1522                  MSG2,    TEXT Z
1523      1Ø13  4347
1523
1524      1Ø14  4Ø43
1524      1Ø15  Ø64Ø
1524      1Ø16  4Ø4Ø
1524      1Ø17  4Ø16
1524      1Ø2Ø  174Ø
1524      1Ø21  232Ø
1524      1Ø22  Ø1Ø3
1524                  F    NO SPACE
1525      1Ø23  Ø543
1525
1526      1Ø24  474Ø
1526      1Ø25  4347
1526      1Ø26  4Ø4Ø
1526      1Ø27  4Ø4Ø
1526      1Ø3Ø  4Ø4Ø
1526      1Ø31  4Ø1Ø
1526      1Ø32  1124
1526      1Ø33  4Ø22
1526      1Ø34  Ø524
1526      1Ø35  2522
1526      1Ø36  164Ø
1526      1Ø37  2417
1526      1Ø4Ø  4ØØ3
1526      1Ø41  1716
1526      1Ø42  2434
1526                              HIT RETURN TO CONT\Z
1527                  /
153Ø      1Ø43  ØØØØ  ANSWER, Ø
1531                          *.+6
1532
1533                          EJECT
-
```

```
                          PMODE
                /SQUARE   ROOT
7Ø52   ØØØØ     SQRT,     Ø
7Ø53   3273               DCA      ROOT      /CLR ROOT
7Ø54   1142               TAD      DPSQ      /IF SQ IS Ø, EXIT
7Ø55   764Ø               SZA CLA
7Ø56   5262               JMP      NOTØ
7Ø57   1143               TAD      DPSQ+1
7Ø6Ø   765Ø               SNA CLA
7Ø61   5652               JMP I    SQRT
7Ø62   1143     NOTØ,     TAD      DPSQ+1    /1ST APPROX OF ROOT
7Ø63   711Ø               CLL RAR            /DIVIDE BY 2
7Ø64   114Ø               TAD      C2ØØØ     /+1/2
7Ø65   3273               DCA      ROOT

7Ø66   1142     SQRT2,    TAD      DPSQ      /SUM OF SQUARES
7Ø67   7421               MQL
7Ø7Ø   1143               TAD      DPSQ+1
7Ø71   71ØØ               CLL
7Ø72   74Ø7               DVI                /DIVIDE BY ROOT APPROX
7Ø73   ØØØØ     ROOT,     Ø
7Ø74   743Ø               SZL
7Ø75   5314               JMP      SQRT1     /OVERFLOW
7Ø76   72ØØ               CLA
7Ø77   75Ø1               MQA                /QUOTIENT TO AC
71ØØ   1273               TAD      ROOT      /ADD APPROX TO QUOTIENT
71Ø1   7ØØ1               IAC
71Ø2   711Ø               CLL RAR            /DIVIDE BY 2
71Ø3   3Ø35               DCA      SINE      /SAVE
71Ø4   1273               TAD      ROOT      /SUBTRACT OLD ROOT FROM NEW ONE
71Ø5   7Ø41               CIA
71Ø6   1Ø35               TAD      SINE
71Ø7   765Ø               SNA CLA
711Ø   5314               JMP      SQRT1     /=Ø
7111   1Ø35               TAD      SINE
7112   3273               DCA      ROOT      /NEW ROOT
7113   5266               JMP      SQRT2

7114   1273     SQRT1,    TAD      ROOT
7115   5652               JMP I    SQRT


                          EJECT
```

```
                        PMODE
                /MOVE PTS FROM ONE AREA TO ANOTHER
                /10 = OLD BUFFER
                /11 = NEW    "
                /IF CMPFLG=1, COMPLEMENT VALUE
7116  0000      MOVPTS, 0
7117  6211              CDF1
7120  1130      NXTPT,  TAD     CMPFLG
7121  7110              CLL RAR         /TO LK
7122  1410              TAD I   10
7123  7430              SZL
7124  7041              CIA
7125  3411              DCA I   11
7126  2034              ISZ     TEMPR
7127  5320              JMP     NXTPT
7130  6201              CDF0
7131  5716              JMP I   MOVPTS

                /MOVE REAL OR MAGNITUDE VALUES
                /FROM 0 TO 2000
                /AND SWAP HALVES
                /DO NOT COMPLEMENT
7132  0000      MVRLMG, 0
7133  7040              CMA             /OLD ADDR OF 1ST 1/2 = 0
7134  3010              DCA     10
7135  1037              TAD     GR      /NEW ADDR OF 1ST 1/2 = 2000 + 1/2 NO OF PTS
7136  1110              TAD     C1777
7137  3011              DCA     11
7140  1033              TAD     ADD2
7141  3034              DCA     TEMPR   /MOVE 1/2 NO OF PTS
7142  3130              DCA     CMPFLG  /DONT COMPLEMENT
7143  4527              JMS I   PMVPTS  /MOVE THEM
7144  7040              CMA             /OLD ADDR OF 2ND 1/2 = 1/2 NO OF PTS
7145  1037              TAD     GR
7146  3010              DCA     10
7147  1110              TAD     C1777   /NEW ADDR OF 2ND 1/2 = 2000
7150  3011              DCA     11
7151  1033              TAD     ADD2
7152  3034              DCA     TEMPR   /1/2 NO OF PTS
7153  4527              JMS I   PMVPTS  /MOVE THEM
7154  5732              JMP I   MVRLMG

                        EJECT
```

```
ACDFØ  1202
A[illegible]R  ØØ41
[illegible]    1135
ADDWOS 1157
ADD1   1174
ADD2   ØØ33
ADJSGN Ø57Ø
ANSWER 7Ø43
ARG2   1Ø2Ø
ASK    472Ø
ASKX   474Ø
ASK2   4523
ASK2X  4571
ASK3   4572
ASK3X  4626
ASR    7415
BIGSNU ØØ12
BOUND  1516
BUFHI  1574
BUFLO  1575
BUFPTR 15Ø4
BUILD  Ø544
C      ØØ32
CAM    7621
CCDFØ  14Ø1
CCIA   Ø162
CCLR   Ø146
CDFØ   62Ø1
CDF1   6211
CHKEND 465Ø
CHKHI  1473
CHKPT  Ø515
CKEND  4211
CMPFLG Ø13Ø
CNOP   Ø163
CNOTS  Ø677
CONT   14ØØ
CONV   4627
CONVER 4674
CORVAL Ø114
COSINE ØØ36
COUNT  Ø116
CSAM   1351
CURCNT 1573
CURDIS 1551
CURLOP 1563
CURRTN 145Ø
CURSAM Ø1Ø1
C1ØØØ  Ø137
C1777  Ø11Ø
C2ØØØ  Ø14Ø
C6ØØØ  Ø1Ø1
DADD   15Ø4
DATTAP 4Ø61
DBLHI  1541
DBLLO  1547
DISFLG 4242
DISPER 6Ø51
DISPLY 6ØØ1
DISPL1 6Ø22
```

```
DOIFFT 0047
DPIMAG 6055
DPMAG  6270
DPREAL 6117
DPSCAL 6226
DPSQ   0142
DSCLOC 1244
DSCLOP 1275
DSCWD  1261
DVI    7407
ENDHI  1577
ENDLO  1576
ERRCHK 4660
F      0007
FDV    6375
FDV2RW 4466
FFT    0400
FIF    4251
FLIP   1046
FLIPCT 0063
FREE   1365
FRESAM 0105
FT     4365
GETRIG 0045
GI     0040
GQ1000 6147
GR     0037
GRET10 0073
GR9    6244
IDORA  1200
IFCOM  4203
IFDIAL 4026
IFDISP 4265
IFFFT  4231
IFFT   0147
IFI    4261
IFT    4363
IFTFLG 4356
IMGPTR 6336
INDEX  1134
INFILE 4555
INVERT 0043
INVRT  1040
K      0031
KBUFHI 1241
KBUFLO 1242
KIDORA 0117
KMNADR 1236
KMNFLD 1235
KMXADR 1240
KMXFLD 1237
KRDORA 0120
KYSCAL 0136
L      0005
LARGER 6216
LDF4   0144
LEFTX  1363
LESS10 0074
LO     6164
LO     0440
LSR    7417
MAXADR 1475
```

```
MAXFLU 1474
MAXNU  0021
MI[illegible]R 1416
MI[illegible]D 1415
MINPTS 0131
MNOVR2 0022
MOVDIS 1361
MOVINP 4063
MOVPTS 7116
MPLIER 4701
MQA    7501
MQL    7421
MSG1   6760
MSG2   7013
MULPLY 4676
MULT   0044
MULTIP 1000
MUY    7405
MVRLMG 7132
M1     0070
M1K    0141
M10    0072
M1000  1527
M11    0077
M12    0071
M215   0102
M261   0106
M321   0103
M340   0105
M353   0104
M4     0075
M400   0107
M4000  0067
M5     0100
M70    1476
N      0003
NMI    7411
NOROT  0564
NOSPAC 4423
NOSWPI 6110
NOSWPR 6124
NOTNOR 1172
NOT0   7062
NOVER4 0020
NOWSTR 4377
NO4MIK 1133
NU     0004
NUMBKS 4503
NUMBKX 4522
NXTCHR 4634
NXTDF  1466
NXTMAG 6320
NXTPNT 1441
NXTPT  7120
OCTL   4711
OF12BU 4000
OKEND  1463
OKFLD  1467
ONDAT  4452
OUTQES 4273
OUTUNT 4310
P      0030
```

```
PDSPER 6372
PFDV7  Ø123
PI     ØØ26
PIFTFG Ø133
PLEFTX Ø125
PMRLMG Ø126
PMVDIS Ø124
PMVPTS Ø127
PNSWPR 6374
PPREPR 6371
PR     ØØ25
PREAD  Ø134
PRELFG Ø132
PREPAR 613Ø
PRFRSH Ø122
PROC   4357
PSHOWT Ø121
PSQRT  6373
PTS    4Ø64
PWRITE Ø135
P4Ø1   1243
Q      ØØ27
QAINIT 1ØØØ
QARFSH 1Ø53
QI     ØØ24
QR     ØØ23
QUAD1  111Ø
QUAD2  1Ø72
QUESNO 4734
QUES1  6411
QUES11 6625
QUES13 6665
QUES2  6453
QUES3  6475
QUES4  6521
QUES5  6571
QUES6  6612
RBUILD ØØ64
RDDATA 432Ø
RDORA  1343
REALFG 6Ø53
RECHK  ØØ66
REDPLY 616Ø
RELPTR 6324
REPL   4446
RESETC ØØ65
REVERS Ø7Ø5
RFRSH  6171
ROOT   7Ø73
ROTAT  4121
ROT1   4136
RTNCDF 1257
RWPARM 64Ø5
S      ØØØ6
SAMNAM 443Ø
SCA    7441
SCAL   ØØ53
SCALE  Ø34Ø
SCL     4Ø3
SCLFA   257
SCR4   Ø145
SC12BU Ø343
```

```
SETC   Ø5
SETDF  1
SETFLD 14
SGNADJ Ø161
SHFCHK ØØ55
SHFLAG ØØ54
SHFT1  1Ø77
SHFT2  1114
SHFT3  1125
SHIFCT Ø562
SHIFT1 ØØ56
SHIFT2 ØØ57
SHIFT3 ØØ6Ø
SHL    7413
SHOSCL 6247
SHOWIT 6162
SIGN   1Ø37
SINE   ØØ35
SINLOC ØØ5Ø
SINRET 1122
SINTAB 7347
SIZE   617Ø
SMALLR 6211
SORT   ØØ42
SORTX  Ø7Ø1
SQRT   7Ø52
SQRT1  7114
SQRT2  7Ø66
STAMU  4144
STSCAL 4367
SWAPED Ø745
TAB    1323
TEMP   1276
TEMPR  ØØ34
TEMP1  4114
TEMP2  4675
TRIGET 1Ø6Ø
UNTFIL 4Ø44
UPADDR 6166
UPLEGL 4645
VCOORD 1274
WCHDIS 6Ø35
WINSAM Ø1ØØ
WORD   ØØ61
WORDP  ØØ62
WRAP   1424
WSAM   136Ø
XCORD  ØØØ1
XCURHI Ø112
XCURLO Ø113
XITAB  2ØØØ
XLOCDF ØØ52
XRLOC  ØØ51
XRTAB  ØØØØ
XSUM   1175
YCUR   Ø115
YSCAL  1444
YSHFT  Ø111
```

FFTD
DEC-12-FQEA-D

READER'S COMMENTS

Digital Equipment Corporation maintains a continuous effort to improve the quality and usefulness of its publications. To do this effectively we need user feedback -- your critical evaluation of this manual.

Please comment on this manual's completeness, accuracy, organization, usability, and readability.

____
____
____
____

Did you find errors in this manual? If so, specify by page.

____
____
____
____
____
____

How can this manual be improved?

____
____
____
____
____
____

Other comments?

____
____
____
____
____
____

Please state your position. ____________________ Date: ______________

Name: ____________________ Organization: ____________________

Street: ____________________ Department: ____________________

City: ______________ State: ______________ Zip or Country ______________

INDEX

HOW TO OBTAIN SOFTWARE INFORMATION

Announcements for new and revised software, as well as programming notes, software problems, and documentation corrections are published by Software Information Service in the following newsletters.

Digital Software News for the PDP-8 & PDP-12

Digital Software News for the PDP-11

Digital Software News for the PDP-9/15 Family

These newsletters contain information applicable to software available from Digital's Program Library. Articles in Digital Software News update the cumulative Software Performance Summary which is contained in each basic kit of system software for new computers. To assure that the monthly Digital Software News is sent to the appropriate software contact at your installation, please check with the Software Specialist or Sales Engineer at your nearest Digital office.

Questions or problems concerning DEC software should be reported to the Software Specialist. In cases where no Software Specialist is available, please send a Software Performance Report form with details of the problem to:

Software Information Service
Digital Equipment Corporation
146 Main Street, Bldg. 3-5
Maynard, Massachusetts 01754

These forms which are available without charge from the Program Library, should be fully filled out and accompanied by Teletype output as well as listings or tapes of the user program to facilitate a complete investigation. An answer will be sent to the individual and appropriate topics of general interest will be printed in the newsletter.

New and revised software and manuals, Software Performance Report forms, and software price lists are available from the Program Library. When ordering, include the document number and a brief description of the program or manual requested. Revisions of programs and documents will be announced in the newsletters. Direct all inquiries and requests to:

Program Library
Digital Equipment Corporation
146 Main Street, Bldg. 1-2
Maynard, Massachusetts 01754

Digital Equipment Computer Users Society (DECUS) maintains a user library and publishes a catalog of programs as well as the DECUSCOPE magazine for its members and non-members who request it. For further information please write to:

DECUS
Digital Equipment Corporation
146 Main Street, Bldg. 3-5
Maynard, Massachusetts 01754

- - - - - - - - - - - - - - - - - Fold Here -

- - - - - - - - - - - - - - Do Not Tear - Fold Here and Staple - - - - - - - - - - - - - -

digital

Digital Equipment Corporation
Software Information Services
146 Main Street, Bldg. 3-5
Maynard, Massachusetts 01754

Digital Equipment Corporation
Maynard, Massachusetts

digital