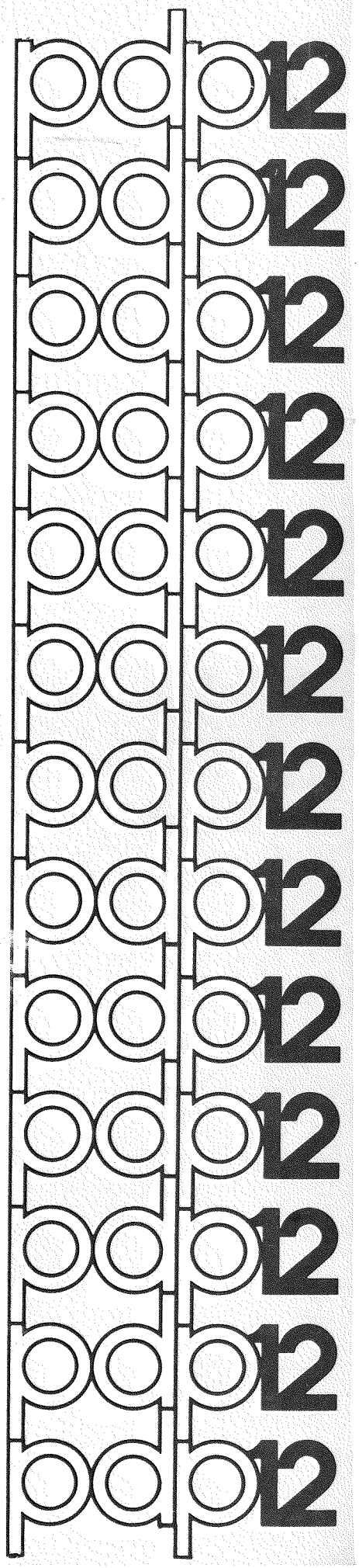
digital

FAST FOURIER TRANSFORM AND DISPLAY



FAST FOURIER TRANSFORM AND DISPLAY

PROGRAMMER'S REFERENCE MANUAL

SOFTWARE SUPPORT CATEGORY

The software described in this document is supported by Digital Equipment Corporation under Category I as defined on page iii of this document.

For additional copies of this document, order No. DEC-12-AMFTA-A-D from Software Distribution Center, Digital Equipment Corporation, Maynard, Mass. 01754. Price $5.00

DIGITAL EQUIPMENT CORPORATION • MAYNARD, MASSACHUSETTS

DEC-12-AMFTA-A-D
1st Printing February, 1971
2nd Printing August, 1972

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.

The following are trademarks of Digital Equipment Corporation, Maynard, Massachusetts:

DEC	PDP
FLIP CHIP	FOCAL
DIGITAL	COMPUTER LAB
OMNIBUS	UNIBUS

SOFTWARE SUPPORT CATEGORIES

Digital Equipment Corporation (DEC) makes available four categories of software. These categories reflect the types of support a customer may expect from DEC for a specified software product. DEC reserves the right to change the category of a software product at any time. The four categories are as follows:

CATEGORY I
Software Products Supported at no Charge

This classification includes current versions of monitors, programming languages, and support programs provided by DEC. DEC will provide installation (when applicable), advisory, and remedial support at no charge. These services are limited to original purchasers of DEC computer systems who have the requisite DEC equipment and software products.

At the option of DEC, a software product may be recategorized from Category I to Category II for a particular customer if the software product has been modified by the customer or a third party.

CATEGORY II
Software Products that Receive Support for a Fee

This category includes prior versions of Category I programs and all other programs available from DEC for which support is given. Programming assistance (additional support), as available, will be provided on these DEC programs and non-DEC programs when used in conjunction with these DEC programs and equipment supplied by DEC.

CATEGORY III
Pre-Release Software

DEC may elect to release certain software products to customers in order to facilitate final testing and/or customer familiarization. In this event, DEC will limit the use of such pre-release software to internal, non-competitive applications. Category III software is only supported by DEC where this support is consistent with evaluation of the software product. While DEC will be grateful for the reporting of any criticism and suggestions pertaining to a pre-release, there exists no commitment to respond to these reports.

CATEGORY IV
Non-Supported Software

This category includes all programs for which no support is given.

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 7, Number 3, available from DECUS Library, Digital Equipment Corporation, Maynard, Massachusetts.

*****

TABLE OF CONTENTS

TABLE OF CONTENTS (Cont'd)

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 7 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

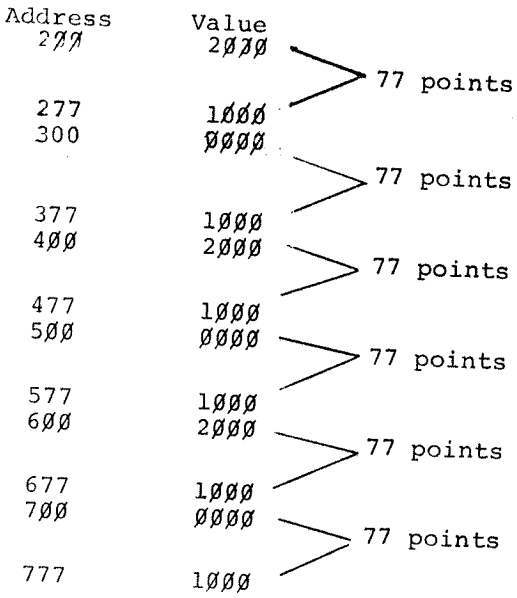
This section provides examples of the displays which result from a transform performed on a square wave of 512 points and from an inverse transform performed on the resulting coefficients.

4.1 Input Display

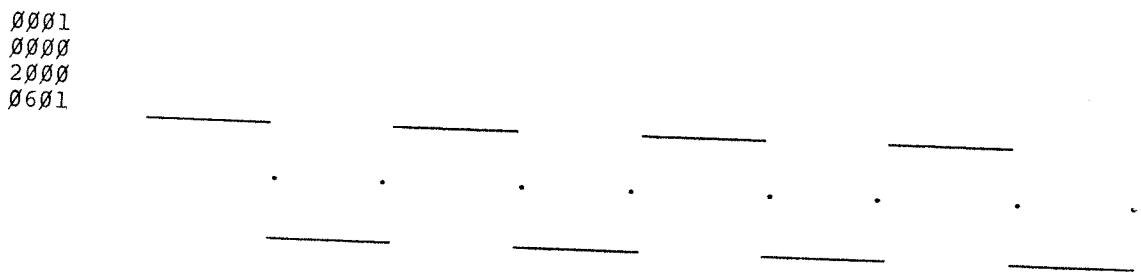
Consider a square wave[1] of 512 real points which has the following format on tape or disk:

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.



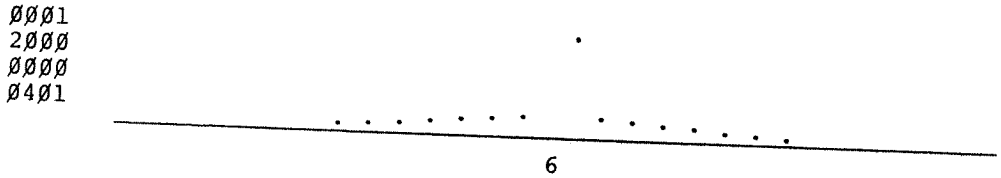
If the input is displayed, there will only be a REAL display. It will look as follows, assuming the cursor is to the extreme left and the display is not moving.



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

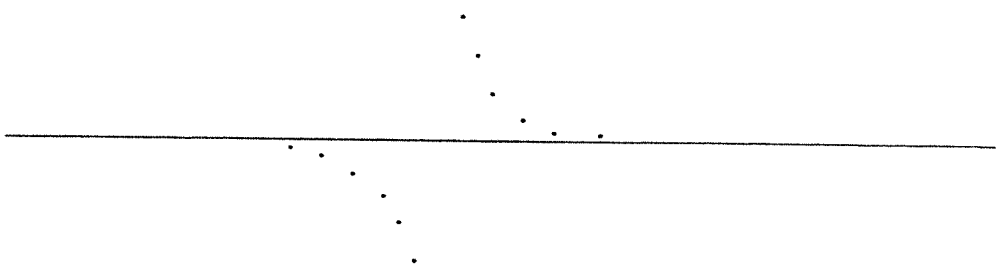


Moving the cursor to the highest point in the display will change the value display to: ØØØ1
24ØØ
2ØØØ
Ø6Ø1

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:

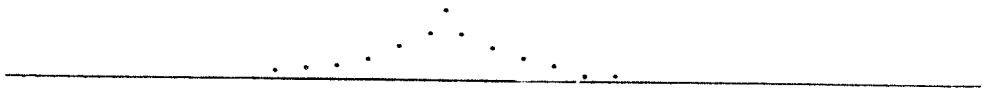
ØØØ1
2374
6567
Ø257

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.

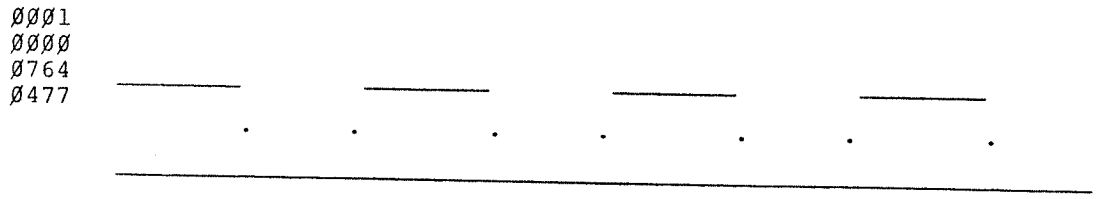
4.2.4 Scale Factor Display

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



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:

$$1ØØØ_8 = [(2ØØØ_8) * 2^8]/1ØØØ_8$$
$$2^9 = 2^{1Ø} * 2^8/2^9$$
$$= 2^9$$

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
ØØØØ
Ø372
Ø44Ø
```

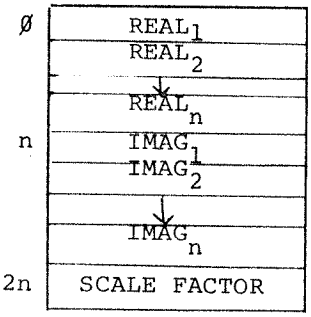
As in the magnitude display of the transform, the values displayed are half scale (displayed scale factor - 1). Because the imaginary components are essentially zero, the magnitude, when doubled, equals the real values.

4.3.4 Scale Factor Display

The scale factor has a value of 7. For the magnitude display, scale factor is 7-1 since display is already half scale.

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:



-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 ([(original data)*2^n]/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

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

of a series on N equally spaced time samples (where N is a power of 2). The time required is proportional to $N \log_2 N$, whereas previous methods required times proportional to N^2. This gives a reduction in computation time of $1-\log_2 N/N$ or over 99 percent for N=1Ø24. The algorithm makes use of the fact that

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

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

01417 IFFT - Take the Inverse Fourier Transformation of the data in field 1. The results are in bit inverted order (refer to the SORTX routine).

04000 FFT - Take the Fourier Transformation of the data in field 1. The results are in bit inverted order (refer to the SORTX routine).

0701 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 100 (=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, $5_{10}=0101_2$. Then reverse the order of these bits to 1010_2. This means S_5 is stored in position 10. Physically, then, S_5 of the real parts is to be found in location XRTAB+9.

1000 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.

1040 INVRT - Reverse the bits of the number contained in the AC.

1060 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 $\geq N/2$. A register COSINE contains the cosine value and the AC contains the sine value on exit.

ADDR - Perform a single precision add with rounding.

1200 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.

4026 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.

4044 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.

4061 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.

4063 MOVINP - Jump to FDV2RW to move the input information from the FDV to the read/write parameter list. Fall through to PTS.

4064 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.

4136 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.

4144 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.

4023 IFCOM - If the answer is C, clear REALFG and fall through to CKEND. Otherwise jump back to PTS to redisplay the question.

4211 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.

4231 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.

4273 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.

4310 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.

4320 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.

4357 PROC - If IFTFLG is Ø, jump to FT to do a Transform. Otherwise, fall through to do an Inverse Transform.

4363 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.

4365 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.

4367 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.

4377 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.

4423 NOSPAC - Jump to the subroutine ASK to display the message:

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

When RETURN is hit, jump to OUTQES.

4430 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.

4446 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.

4452 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

4466 FDV2RW - Transfer the unit number, starting block number, and number of blocks from the FDV parameter list to the READ/WRITE parameter list.

4503 NUMBKS - Enter with the number of words in the AC. Convert this value to blocks by counting the number of times 400 can be subtracted from it before the value becomes negative. Return with the number of blocks in the AC.

4523 ASK2 - Jump to OCTL to set MPLIER to 10 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.

4572 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.

4627 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.

4711 OCTL - OCTL sets MPLIER to 1Ø and UPLEGL to -67(-7) so that CONV will convert an octal number.

4720 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.

6001 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.

6035 WCHDIS - Jump to DPIMAG, DPMAG, DPREAL, or DPSCAL if the answer was I, M, R, or S,respectively. Otherwise redisplay the question.

6055 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.

6117 DPREAL - Check IFTFLG for the same reason as in DPIMAG. The only difference is that the signs of the real parts are not reversed.

6130 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.

6147 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.

6162 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.

6171 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.

6211 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.

6216 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.

6226 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.

6270 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.

6320 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.

7116 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.

7132 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.

6375 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.

6405 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.

7052 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:

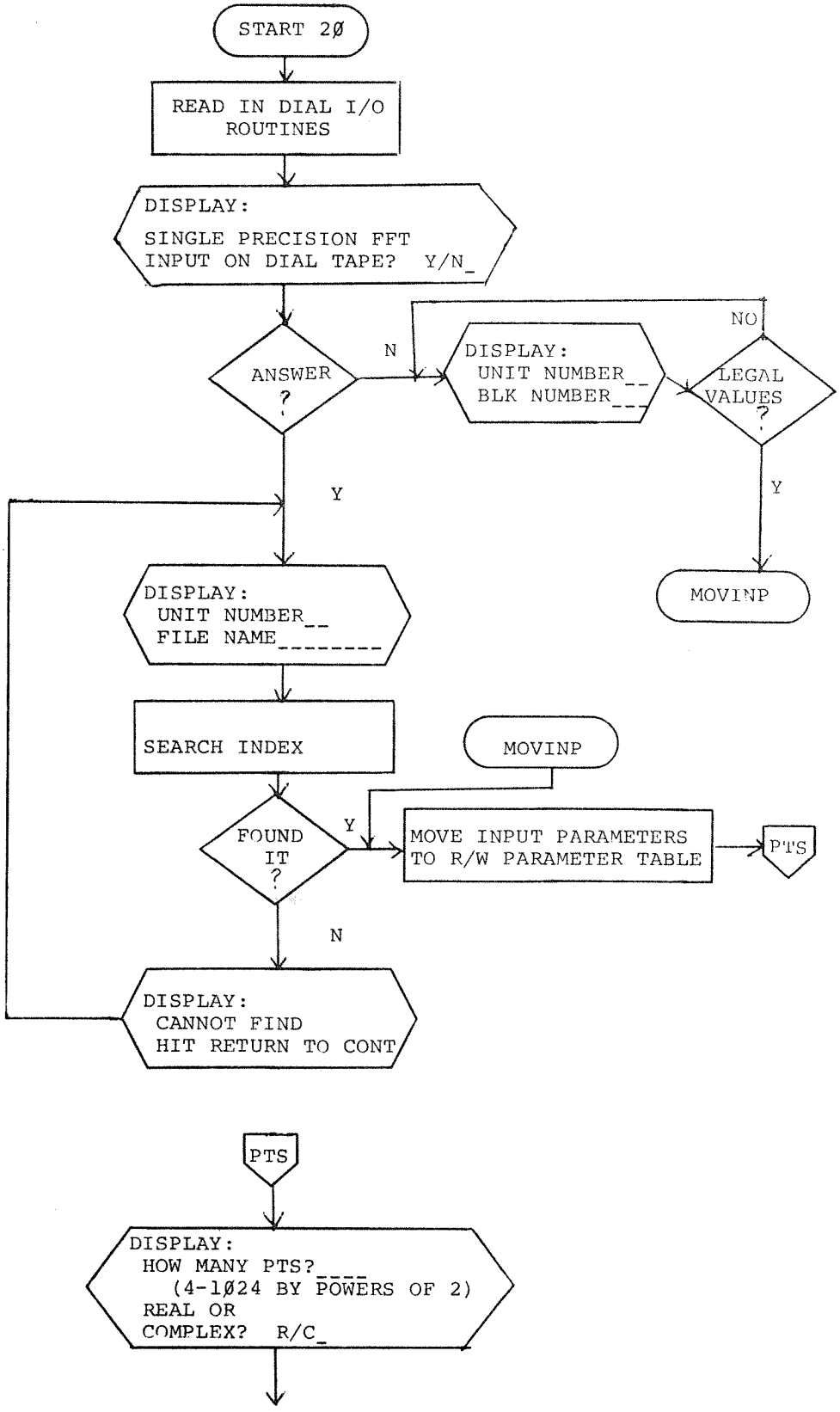
```
→AS MILQAN,n ↵        where n is the unit
→SB MILQAN,n ↵        containing the program
→AS SIN256,n ↵
→SB SIN256,n ↵
→AS FFTC-1 ↵          (FFTC-1 chains to FFTC-2)
→SB FFTC-1 ↵
→ZE ↵
→AB MILQAN,n ↵
→AB SIN256,n ↵
→AB FFTC-1,n ↵
→SB FFTD,n,L ↵        (saves the whole program)
```

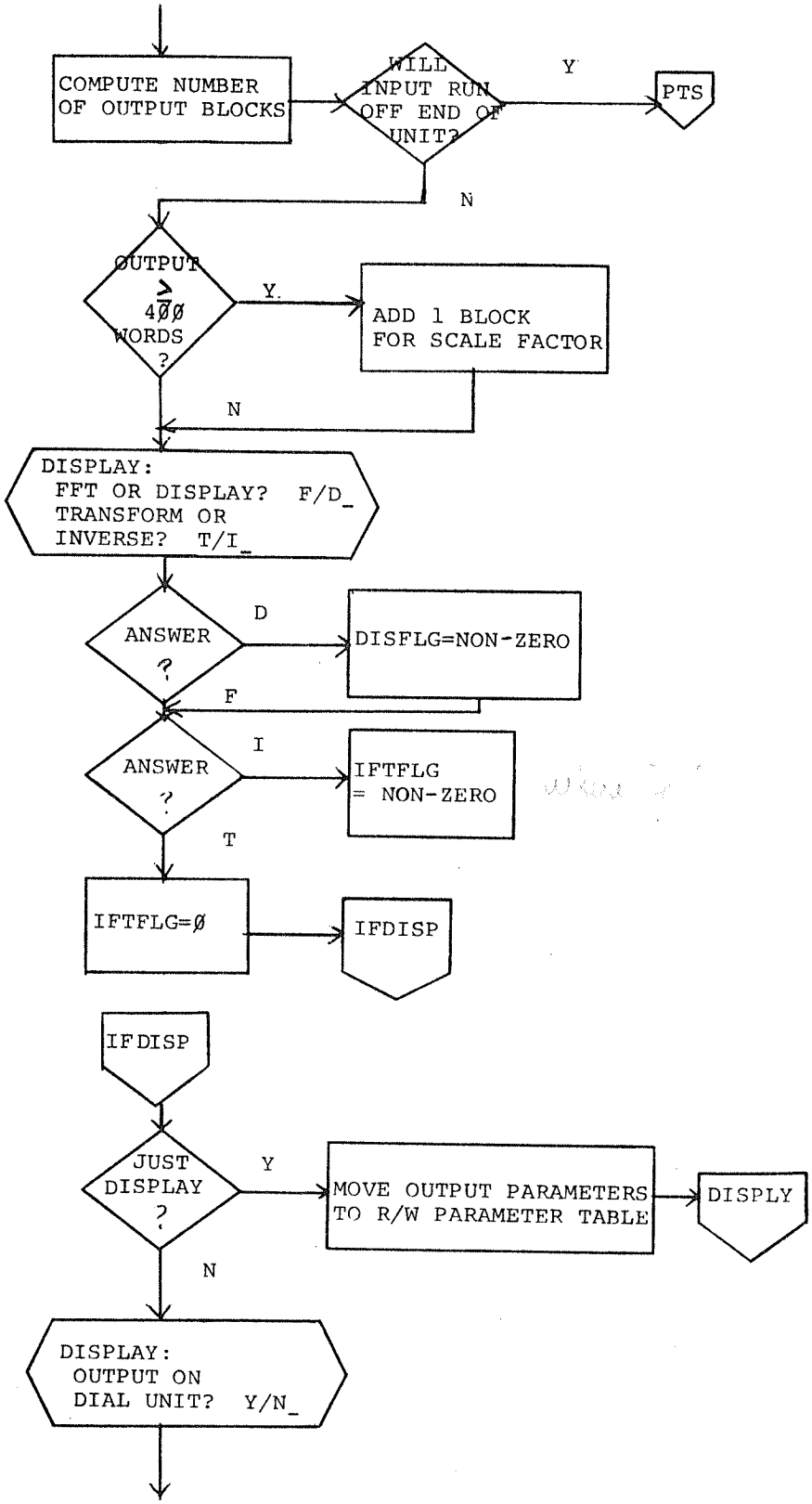
12.0 SYSTEM FLOWCHARTS

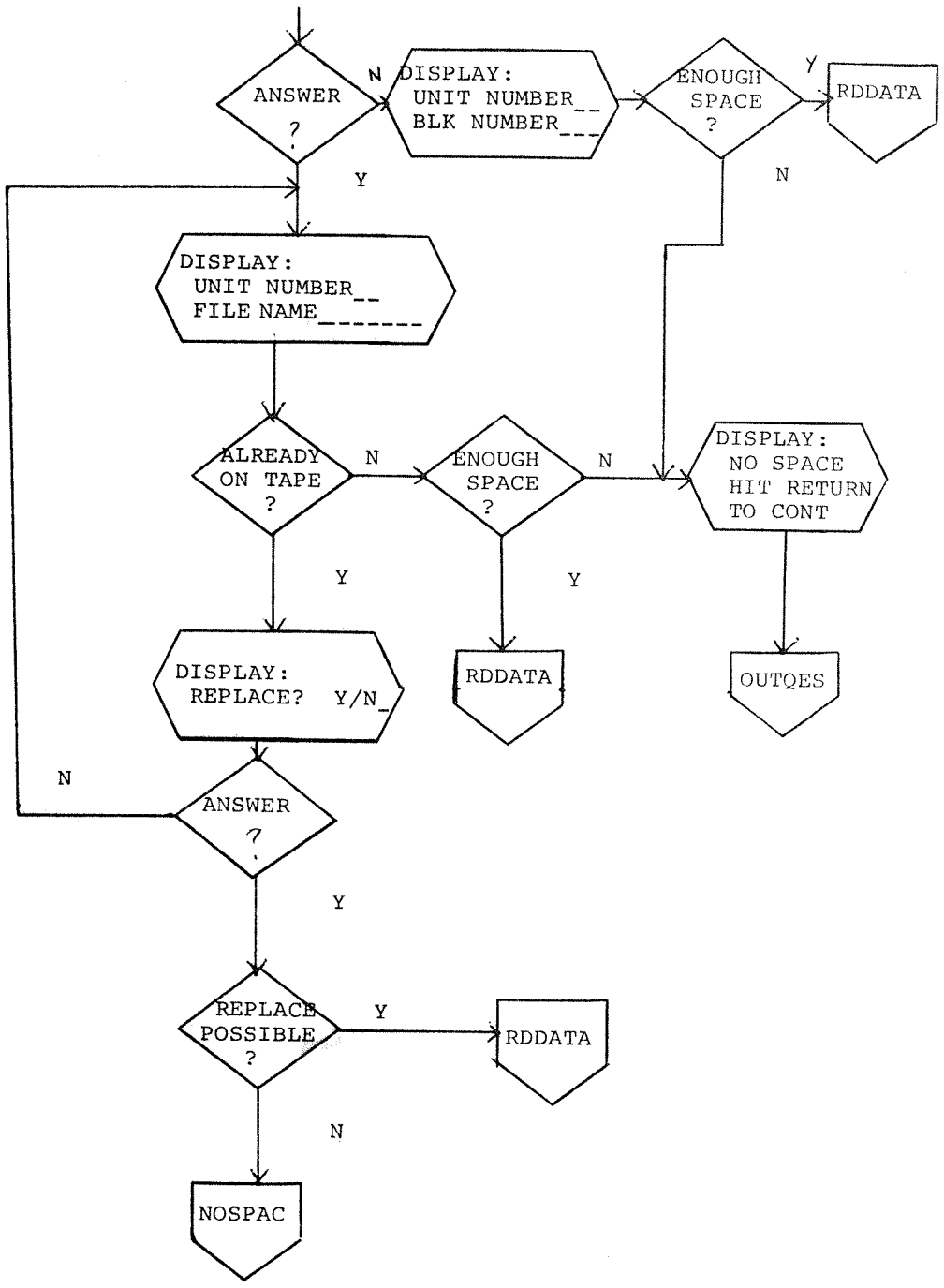
(Attached)

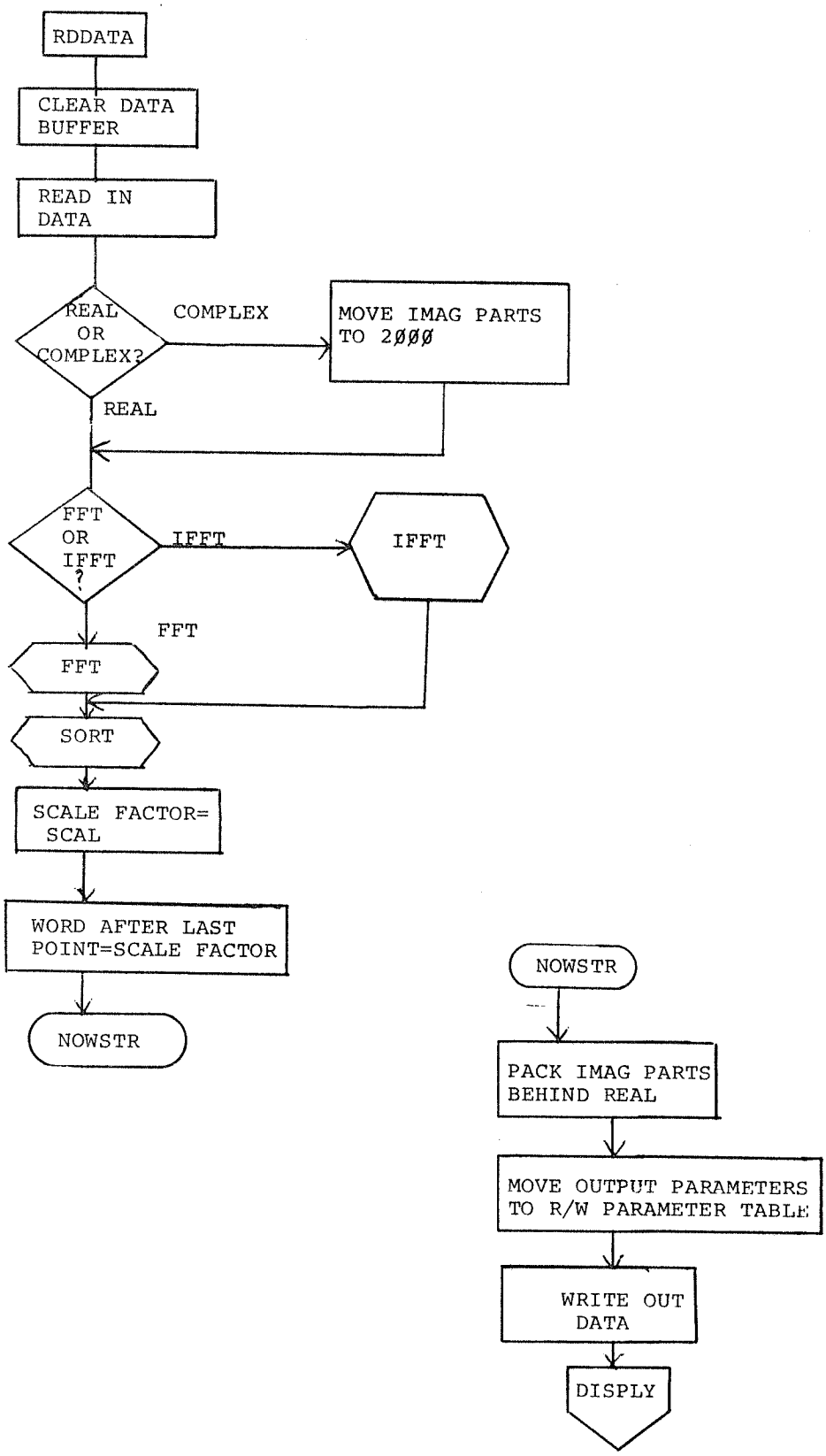
13.0 PROGRAM LISTING

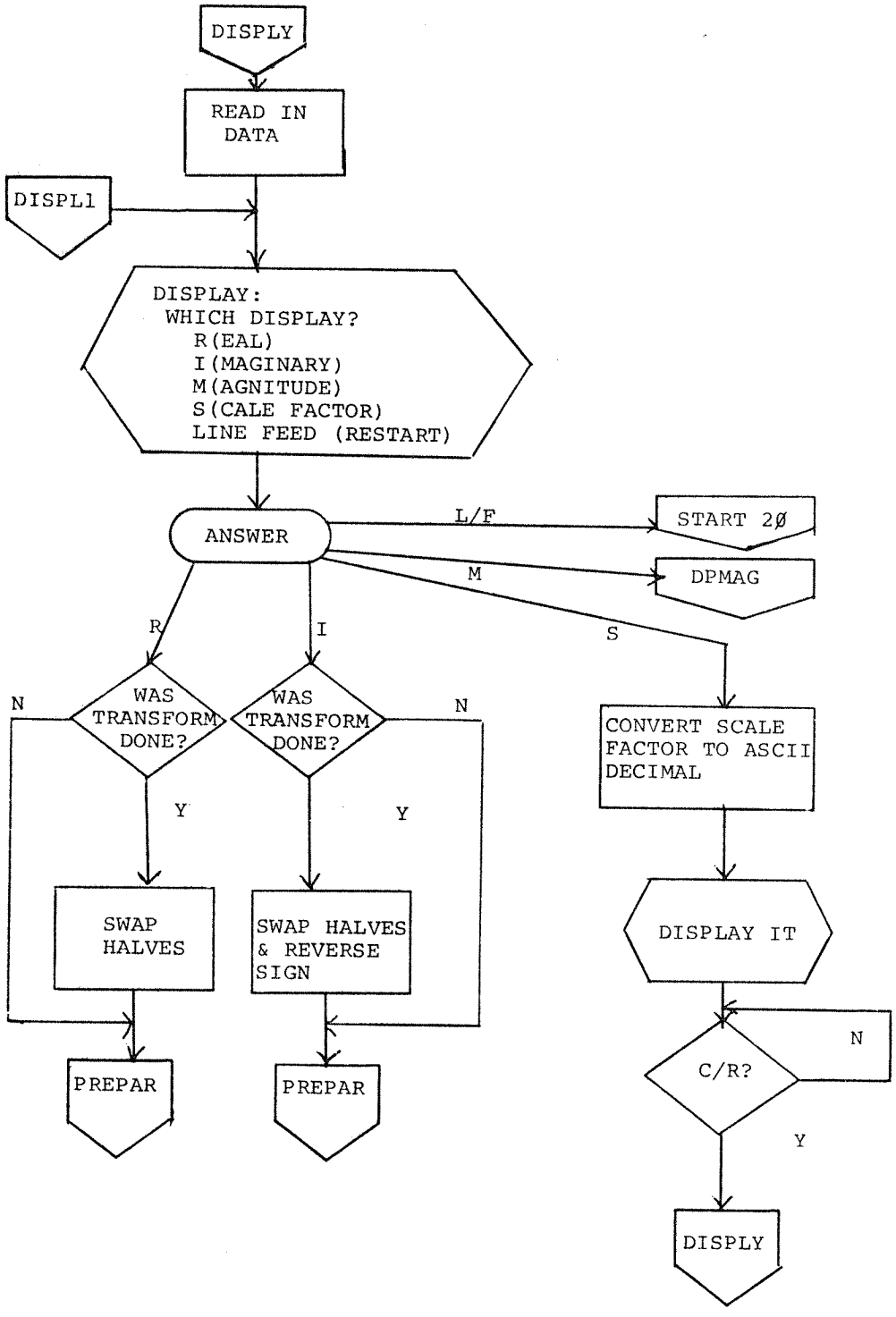
(Attached)

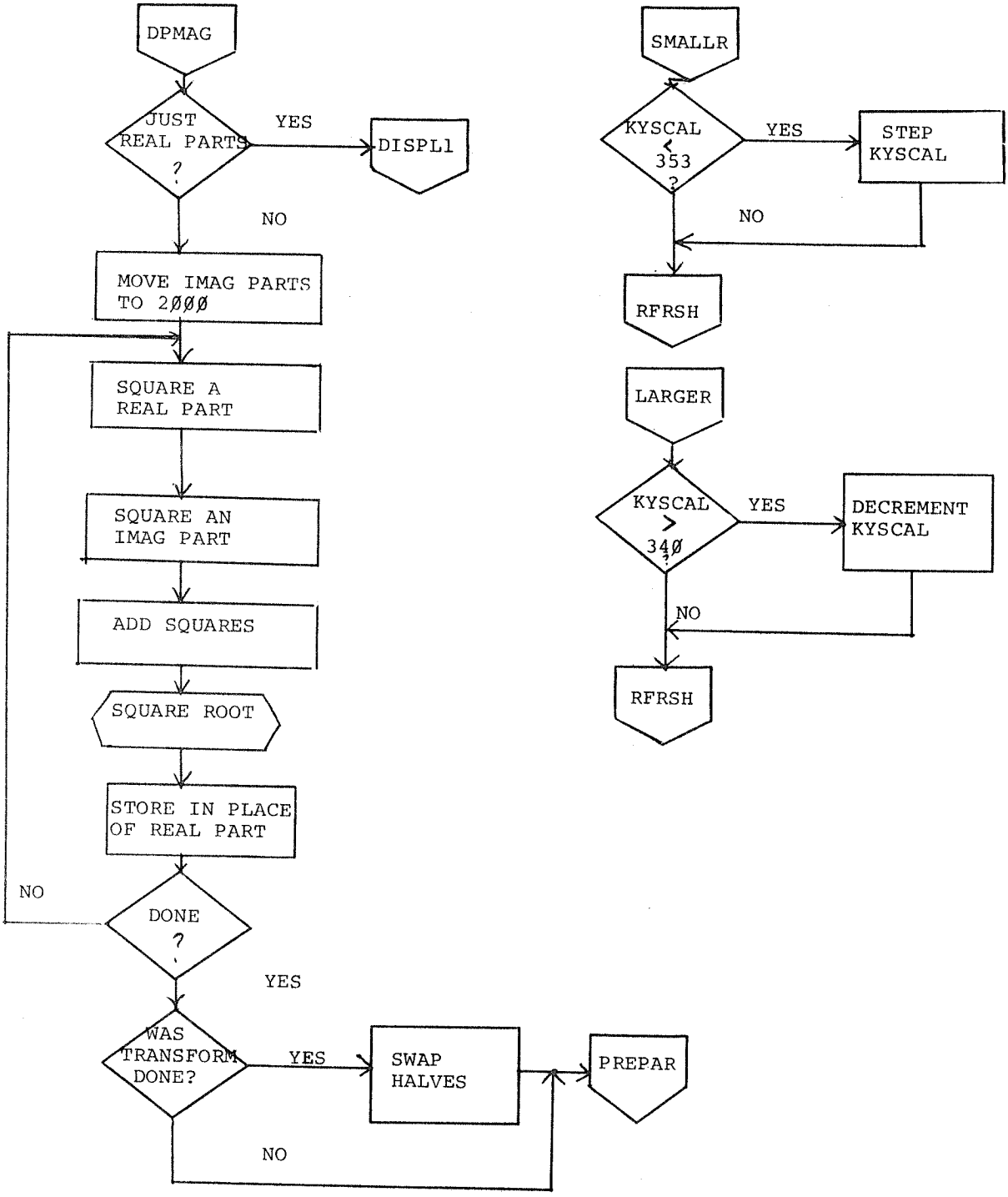

START 2Ø
READ IN DIAL I/O ROUTINES
DISPLAY:
SINGLE PRECISION FFT
INPUT ON DIAL TAPE? Y/N_
ANSWER ?
N
Y
DISPLAY:
UNIT NUMBER__
BLK NUMBER__
LEGAL VALUES ?
NO
Y
MOVINP
DISPLAY:
UNIT NUMBER__
FILE NAME_____
SEARCH INDEX
MOVINP
FOUND IT ?
Y
MOVE INPUT PARAMETERS TO R/W PARAMETER TABLE
PTS
N
DISPLAY:
CANNOT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

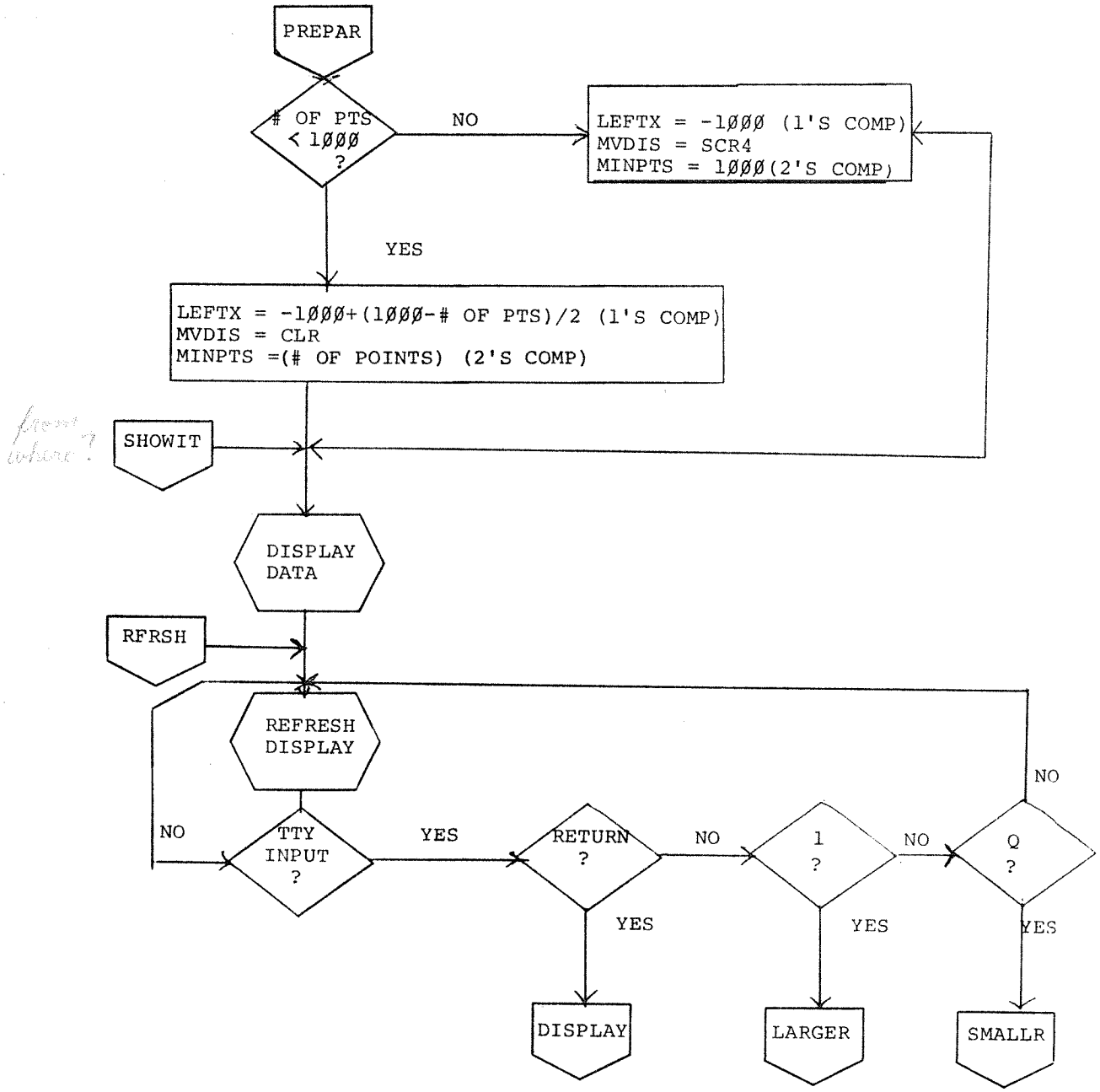

COMPUTE NUMBER OF OUTPUT BLOCKS
WILL INPUT RUN OFF END OF UNIT?
Y
PTS
N
OUTPUT ≥ 4ØØ WORDS ?
Y
ADD 1 BLOCK FOR SCALE FACTOR
N
DISPLAY: FFT OR DISPLAY? F/D_ TRANSFORM OR INVERSE? T/I_
ANSWER ?
D
DISFLG=NON-ZERO
F
ANSWER ?
I
IFTFLG = NON-ZERO
T
IFTFLG=Ø
IFDISP
IFDISP
JUST DISPLAY ?
Y
MOVE OUTPUT PARAMETERS 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
N
Y
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
R
I
S
N
WAS TRANSFORM DONE?
WAS TRANSFORM DONE?
N
Y
Y
SWAP HALVES
SWAP HALVES & REVERSE SIGN
PREPAR
PREPAR
CONVERT SCALE FACTOR TO ASCII DECIMAL
DISPLAY IT
C/R?
N
Y
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
DONE ?
NO
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 < 1000 ?
NO
LEFTX = -1000 (1'S COMP)
MVDIS = SCR4
MINPTS = 1000(2'S COMP)
YES
LEFTX = -1000+(1000-# OF PTS)/2 (1'S COMP)
MVDIS = CLR
MINPTS =(# OF POINTS) (2'S COMP)
SHOWIT
DISPLAY DATA
RFRSH
REFRESH DISPLAY
NO
TTY INPUT ?
YES
RETURN ?
NO
1 ?
NO
Q ?
NO
YES
YES
YES
DISPLAY
LARGER
SMALLR

```
0000                    *20
0001                    /FFTS-REAL
0002                    /THIS IS A PROGRAM FOR CALCULATING THE
0003                    /FAST FOURIER TRANSFORMATION OF N REAL
0004                    /TIME SAMPLES WHICH ARE STORED ON DIAL
0005                    /OR DATA TAPE OR DISK
0006                    /TO BE RUN ON A PDP-12 COMPUTER EQUIPPED/WITH THE FOLLOWING MINIMUM HARDWARE:
0007                    /  1) ASR 33 OR ASR 35 TELETYPE
0010                    /  2) 8 K OF CORE MEMORY
0011                    /  3) VR12 CRT DISPLAY
0012                    /
0013                    /COPYRIGHT 1970, DIGITAL EQUIPMENT CORPORATION
0014                    /         MAYNARD, MASS, 01754
0015                    /TRANSFORM ALGORITHM
0016                    /WRITTEN BY JAMES ROTHMAN ** AUGUST, 1968
0017                            QARFSH=1053
0020                            QAINIT=1000
0021                            XRTAB=0
0022                            XITAB=2000
0023                            SINTAB=7347
0024                            CDF1=6211
0025                            CDF0=6201
0026                            PMODE
0027                    /PAGE ZERO
0030                    *3
0031                    /TABLE PARAMETERS
0032    0003    0000    N,      0               /NUMBER OF POINTS IN COMPUTAION DIVIDED BY 2
0033    0004    0000    NU,     0               /POWER OF TWO OF POINTS IN COMPUTATION (N=2↑NU) MINUS 1
0034    0005    0000    L,      0               /INDEX TO SHOW WHAT ARRAY IS BEING CONSTRUCTED
0035    0006    0000    S,      0               /GIVES SPACING BETWEEN NODE PAIRS IN THE LTH ARRAY.
0036    0007    0000    F,      0               /USED FOR SCALING NODE POSITION TO GET NUMBER IN NODES.
0037
0040                    *20
0041    0020    0000    NOVER4, 0               /STORAGE FOR N/4
0042    0021    0012    MAXNU,  BIGSNU          /LARGEST TABLE SIZE (POWER OF 2)
0043    0022    0000    MNOVR2, 0               /STORAGE FOR -N/2
0044                    /INDEXING VARIABLES
0045    0023    0000    QR,     0               /POINTER TO REAL PART OF X(Q)
0046    0024    0000    QI,     0               /POINTER TO IMAG, PART OF X(Q)
0047    0025    0000    PR,     0               /POINTER TO REAL PART OF X(P)
0050    0026    0000    PI,     0               /POINTER TO IMAG, PART OF X(P)
0051    0027    0000    Q,      0               /NUMERICAL INDEX Q(=0,1,,,,N-1)
0052    0030    0000    P,      0               /NUMBERICAL INDEX P(=0,1,,,,N-1)
0053    0031    0000    K,      0               /NUMBER IN THE NODE BEING OPERATED ON
0054                    /LOOP DELIMITERS
0055    0032    0000    C,      0               /INTERRUPTS COMPUTATION OF LTH ARRAY EVERY S PASSES
0056                    /DATA VARIABLES
0057    0033    0000    ADD2,   0               /USED BY SUBROUTINE ADDR AS DATA (ADDEND)
0060    0034    0000    TEMPR,  0               /TEMPORARY STORAGE REGISTER FOR REAL PARTS
0061    0035    0000    SINE,   0               /TEMP. STORAGE FOR SIN (S*PI*K/N)
0062    0036    0000    COSINE, 0               /TEMP. STORAGE FOR COS (2*PI*K/N)
0063    0037    0000    GR,     0               /REAL PART OF PRODUCT (W*K)*X(P), TEMP STORAGE
0064    0040    0000    GI,     0               /IMAG. PART OF (W*K)*X(P). TEMP STORAGE
0065                    /SUBROUTINE CALL LIST
0066    0041    1135    ADDER,  ADDR            /ADD C(AC) TO C(ADD2) AND SCALE RIGHT ONE IF NECESSARY.
0067    0042    0701    SORT,   SORTX           /BIT INVERTED BUFFER SORTED
0070    0043    1040    INVERT, INVRT           /WORD IN AC OF NU BITS IS BIT INVERTED
0071    0044    1000    MULT,   MULTIP
0072    0045    1060    GETRIG, TRIGET          /FETCH SIN AND COS OF 2*PI*C(AC)/N
0073    0046    0400    DOFFT,  FFT             /DO FFT OF THE INPUT BUFFER
0074    0047    0147    DOIFFT, IFFT            /DO INVERSE OF BUFFER
```

```
0076  0051  0000  XRLOC,  XRTAB            /INPUT BUFFER AND [illegible]
0077  0052  2000  XLOCDF, XITAB=XRTAB      /DIFF IN ADDR OF REAL & IMAG PART TABLES
0100                /PSEUDO FLOATING POINT FORMAT FLAGS
0101  0053  0000  SCAL,   0                /PSEUDO EXPONENT OF FOURIER COEFFICIENTS
0102  0054  0001  SHFLAG, 1                /IF =1, ADD WITH SHIFT; IF=0, ADD WITHOUT SHIFT
0103  0055  0000  SHFCHK, 0                /INDICATES IF ALL XS IN AN ITERATION ARE <.5
0104                /POINTERS TO SINE TABLE LOOK-UP SHIFTS
0105  0056  1077  SHIFT1, SHFT1           /THE NUMBER 10-NU MUST BE PLACED
0106  0057  1114  SHIFT2, SHFT2           /IN EACH OF THESE LOCATIONS
0107  0060  1125  SHIFT3, SHFT3
0110                /POINTERS TO INSTRUCTION "FLAG" LOCATIONS
0111  0061  0000  WORD,   0
0112  0062  0000  WORDP,  0
0113  0063  0000  FLIPCT, 0
0114                /
0115  0064  0544  RBUILD, BUILD
0116  0065  0542  RESETC, SETC
0117  0066  0515  RECHK,  CHKPT
0120  0067  4000  M4000,  -4000
0121  0070  7777  M1,     -1
0122  0071  7766  M12,    -12
0123  0072  7770  M10,    -10
0124  0073  6160  GRET10, 6160
0125  0074  4060  LESS10, 4060
0126  0075  7774  M4,     -4
0127  0076  6270  PDPMAG, DPMAG
0130  0077  7767  M11,    -11
0131  0100  7773  M5,     -5
0132  0101  6000  C6000,  6000
0133  0102  7563  M215,   -215
0134  0103  7457  M321,   -321
0135  0104  7425  M353,   -353
0136  0105  7440  M340,   -340
0137  0106  7517  M261,   -261
0140  0107  7400  M400,   -400
0141  0110  1777  C1777,  1777
0142  0111  0000  YSHFT,  0
0143  0112  0000  XCURHI, 0
0144  0113  0000  XCURLO, 0
0145  0114  0000  CORVAL, 0
0146  0115  0000  YCUR,   0
0147  0116  0000  COUNT,  0
0150  0117  1200  KIDORA, IDORA
0151  0120  1343  KRDORA, RDORA
0152  0121  6162  PSHOWT, SHOWIT
0153  0122  6171  PRFRSH, RFRSH
0154  0123  6404  PFDV7,  FDV+7
0155  0124  1361  PMVDIS, MOVDIS
0156  0125  1363  PLEFTX, LEFTX
0157  0126  7132  PMRLMG, MVRLMG
0160  0127  7116  PMVPTS, MOVPTS
0161  0130  0000  CMPFLG, 0
0162  0131  0000  MINPTS, 0
0163  0132  6053  PRELFG, REALFG
0164  0133  4356  PIFTFG, IFTFLG
0165  0134  7774  PREAD,  7774
0166  0135  7775  PWRITE, 7775
0167  0136  1444  KYSCAL, YSCAL
0170  0137  1000  C1000,  1000
0171  0140  2000  C2000,  2000
0172  0141  6777  M1K,    6777           /-1000 1,S COMP
0173  0142  0000  DPSQ,   0              /D.P. SQUARE
```

```
0174      0143  0000                0
0175                                LMODE
0176      0144  0644  LDF4,         LDF 4
0177      0145  0344  SCR4,         SCR 4
0200      0146  0011  CCLR,         CLR
0201                                PMODE
0202                                EJECT
```

```
0203                    /THIS SUBROUTINE TAKES THE INVERSE FFT (IFFT) OF THE DATA IN THE BUFFER,
0204                    /IT IS ASSUMED THAT THIS DATA IS STORED SEQUENTIAL ORDER,
0205                    /THE RESULTS ARE STORED IN BIT INVERTED ORDER,
0206                    /THE ALGORITHM USED IS AS FOLLOWS:
0207                    /       THE NORMAL TRANSFORM IS PERFORMED, EXCEPT:
0210                    /       ON FETCHING THE VALUE FOR IM[W*K], WHICH IS
0211                    /       THE SIN(2*PI*K/N), THIS SIN VALUE IS NEGATED,
0212                    /
0213                    /THE REASONING FOR THIS IS AS FOLLOWS:
0214                    /       A WEIGHTING FACTOR OF W*8-K) IS USED IN THE IFFT
0215                    /       AND SINCE W*K AND W*(-K) ARE THE SAME EXCEPT THAT
0216                    /       THEIR IMAGINARY PARTS HAVE OPPOSITE SIGNS, IT FOLLOWS
0217                    /       THAT IM]W*K] SHOULD BE REPLACED BY -IM[W*K],
0220    0147  0000  IFFT,   0
0221    0150  7300          CLA CLL
0222    0151  1162          TAD     CCIA    /NEGATE IM[W*K], GET CIA INSTRUCTION
0223    0152  3561          DCA I   SGNADJ  /AND PUT AT LOCATION ADJSN
0224    0153  4446          JMS I   DOFFT   /DO FFT
0225    0154  6201          CDF0
0226    0155  1163          TAD     CNOP    /RE-INSTATE NOP AT ADJSGN FOR FFT,
0227    0156  3561          DCA I   SGNADJ
0230    0157  6211          CDF1
0231    0160  5547          JMP I   IFFT    /EXIT
0232    0161  0570  SGNADJ, ADJSGN          /POINTER TO SIGN ADJUST INSTRUCTION
0233    0162  7041  CCIA,   CIA
0234    0163  7000  CNOP,   NOP
0235                        EJECT
```

```
0236                    *400
0                       /COMPUTATION OF FIRST COMPLEX ARF    FROM INPUT DATA
0                       /NUMBER OF INPUT POINTS IN "N" .L    2)(N)IN"NU", FOR DETAILS OF ALGORITHM, SEE FLOWCHA
0241    0400  0000      FFT,    0
0242    0401  7301              CLA IAC CLL
0243    0402  3005              DCA     L               /L<=1
0244    0403  3053              DCA     SCAL            /INITIALIZE FLOATING POINT FORMAT
0245    0404  7001              IAC
0246    0405  3054              DCA     SHFLAG
0247    0406  3055              DCA     SHFCHK
0250    0407  1003              TAD     N
0251    0410  7112              CLL RTR                 /INITIALIZE PROGRAM CONSTANTS
0252    0411  3020              DCA     NOVER4
0253    0412  1004              TAD     NU
0254    0413  7041              CIA
0255    0414  1021              TAD     MAXNU
0256    0415  3456              DCA I   SHIFT1
0257    0416  1456              TAD I   SHIFT1
0260    0417  3457              DCA I   SHIFT2
0261    0420  1457              TAD I   SHIFT2
0262    0421  3460              DCA I   SHIFT3
0263    0422  1003              TAD     N
0264    0423  7110              CLL RAR
0265    0424  3006              DCA     S               /S<=N/2 IS SPACING OF NODE PAIRS IN FIRST ARRAY
0266    0425  1006              TAD     S
0267    0426  7041              CIA
0270    0427  3022              DCA     MNOVR2
0271    0430  7040              CMA                     /AC<=-1
0272    0431  1006              TAD     S               /AC<=[N/2-1]*2
0273    0432  1051              TAD     XRLOC           /BEGINNING OF TABLE OF REAL PARTS,
0274    0433  3023              DCA     QR              /Q<=N/2-1, QR POINTS TO WORD IN MEMORY, WHILE Q IS ACTUAL INDEX
0275    0434  1004              TAD     NU
0276    0435  7041              CIA
0277    0436  7001              IAC
0300    0437  3007              DCA     F               /F<=1-NU (=L-NU SINCE L=1)
0301    0440  1023    LOOP1,    TAD     QR              /QR=XRLOC+Q AT ALL TIMES,
0302    0441  1006              TAD     S
0303    0442  3025              DCA     PR              /P<=Q+N/2
0304    0443  1023              TAD     QR              /XLOCDF=XILOC-XRLOC (XILOC=BEGIN, OF IMAG PARTS TABLE)
0305    0444  1052              TAD     XLOCDF          /QR+XLOCDF=(S+XRLOC)+(XILOC-XRLOC)=XILOC+S=QI
0306    0445  3024              DCA     QI              /QI=XILOC+Q AT ALL TIMES, QI POINTS TO IMAG. PART OF X(Q)
0307    0446  1025              TAD     PR
0310    0447  1052              TAD     XLOCDF          /COMPUTE COMPLEX OPERATIONS X(P)<=X(Q)-X(P) AND X(Q)<=X(Q)+X(P)
0311    0450  3026              DCA     PI              /BY REAL AND IMAGINARY PARTS,
0312    0451  6211              CDF1
0313    0452  1424              TAD I   QI              /IM(X(Q)) (IM () MEANS IMAGINARY PART)
0314    0453  3033              DCA     ADD2            /MAKE IT ADDEND, DO IMAG. PARTS FIRST
0315    0454  1426              TAD I   PI              /IM(X(P))
0316    0455  4441              JMS I   ADDER           /FORM ADDITION IM[X(P)+X(Q)]=IM[X(P)]+IM[X(Q)] AND SCALE RIGHT
0317    0456  3034              DCA     TEMPR           /FOR SCALING, THEN STORE,
0320    0457  1424              TAD I   QI              /FORM DIFFERENCE IM[X(Q)-X(P)]=IM[X(Q)]-IM[X(P)]
0321    0460  3033              DCA     ADD2
0322    0461  1426              TAD I   PI
0323    0462  7041              CIA
0324    0463  4441              JMS I   ADDER
0325    0464  3426              DCA I   PI              /PUT AWAY AT IM[X(P)]
0326    0465  1034              TAD     TEMPR           /GET IM[X(P)+X(Q)]
0327    0466  3424              DCA I   QI              /PUT AT IM[X(Q)], IMAGINARY PARTS DONE,
0330    0467  1423              TAD I   QR              /ADD REAL PARTS NEXT
0331    0470  3033              DCA     ADD2
0332    0471  1425              TAD I   PR              /RE=REAL PART
0333    0472  4441              JMS I   ADDER           /FORM RE[X(P)+X(Q)]=RE[X(P)]+RE[X(Q)] (DIVIDED BY 2)
```

```
0336  0475  3033          TAD I   QR              /GET RE[X(Q)]
0337  0476  1425          DCA     ADD2
0340  0477  7041          TAD I   PR              /RE=REAL PART
0341  0500  4441          CIA
0342  0501  3425          JMS I   ADDER           /FORM RE[X(Q)-(P)] (DIVIDED BY 2)
0343  0502  1034          DCA I   PR              /PUT AT RE[X(P)]
0344  0503  3423          TAD     TEMPR           /GET RE[X(Q)+X(P)]
0345  0504  1051          DCA I   QR              /PUT AT RE[X(Q)],REAL PARTS DONE
0346  0505  7041          TAD XRLOC               /Q=QR-XRLOC
0347  0506  1023          CIA
0350  0507  7750          TAD QR          /AC IS Q
0351  0510  5315          SPA SNA CLA     /IS Q>0? (IE THE WHOLE ARRAY HAS NOT BEEN COVERED)
0352  0511  7040          JMP CHKPT       /NO, Q=0, DONE WITH FIRST ARRAY, MOVE ON TO OTHERS
0353  0512  1023          CMA             /YES, Q<=Q-1, MOVE UP THIS ARRAY
0354  0513  3023          TAD QR          /OR EQUIVALENTLY, QR<=QR-1
0355  0514  5240          DCA QR
0356  0515  1005  CHKPT,   JMP LOOP1       /DO NEXT NODE PAIR
0357  0516  7041          TAD L           /L GIVES THE NUMBER OF THE VERTICAL ARRAY JUST BUILT
0360  0517  1004          CIA
0361  0520  7650          TAD NU          /IS L=NU?  (IE HAS THE LAST ARRAY BEEN COMPUTED?)
0362  0521  5600          SNA CLA
0363  0522  1055          JMP I FFT     /YES, DONE, RESULTS STORED IN BIT REVERSED ORDER
0364  0523  3054          TAD SHFCHK      /GET SCALE FACTOR AND ADJUST FOR PROPER
0365  0524  1055          DCA SHFLAG      /ADDITION ON NEXT ITERATION
0366  0525  7650          TAD     SHFCHK
0367  0526  2053          SNA CLA
0370  0527  3055          ISZ     SCAL
0371  0530  2005          DCA     SHFCHK
0372  0531  1006          ISZ     L               /L<=L+1, MOVE ON TO NEXT ARRAY
0373  0532  7110          TAD     S               /S GIVES SPACING BETWEEN NODE PAIRS, WHICH IS N/2*L
0374  0533  3006          CLL RAR                 /DIVIDE BY 2 AND PUT BACK, SO THAT ON THE LTH PASS THROUGH
0375  0534  2007          DCA     S               /S WILL=N/2*L, THE SPACING,
0376  0535  7000          ISZ     F               /F<=F+1, ON LTH PASS, F WILL BE F=L-NU, THE SCALE FACTOR FOR K.
0377  0536  7040          NOP                     /NOP FOR WHEN F=-1 TO PREVENT ERROR DUE TO SKIP
0400  0537  1003          CMA                     /AC<=-1
0401  0540  1051          TAD     N
0402  0541  3025          TAD     XRLOC
0403  0542  7201  SETC,    DCA     PR              /P<=N-1, PR POINTS TO RE[X(P=N-1)]
0404  0543  3032          CLA IAC
0405  0544  1025  BUILD,   DCA     C               /C<=1, C BREAKS BUILD LOOP EVERY S ITERATIONS
0406  0545  1052          TAD     PR      /SO AS TO AVOID RECOMPUTATION
0407  0546  3026          TAD     XLOCDF
0410  0547  1051          DCA     PI
0411  0550  7041          TAD     XRLOC   /PR=XRLOC+P
0412  0551  1025          CIA
0413  0552  3030          TAD     PR
0414  0553  1007          DCA     P       /ACTUAL INDEX IS P:(0,1,,,,,N-1)
0415  0554  7450          TAD     F               /BUILD ARRAY, F=L-NU, SHIFT "P"-F PLACES RIGHT (=NU-L)
0416  0555  5364          SNA                     /SHIFT ZERO PLACES?
0417  0556  7040          JMP     NOROT           /YES, LEAVE ALONE
0420  0557  3362          CMA                     /F COMPLEMENTED IS -F-(1)=-(F+1)=PLACES TO BE SHIFTED-1
0421  0560  1030          DCA     SHIFCT          /CONTAINS-F-1
0422  0561  7417          TAD     P               /GET NODE INDEX
0423  0562  7402  SHIFCT,  LSR                     /SHIFT P RIGHT SHIFCT+1=-F-1+1=-F=NU-L PLACES
0424  0563  7410          HLT             /STORAGE FOR SHIFT COUNT.
0425  0564  1030  NOROT,   SKP                     /AC<=INTEGER PART [P*2+F]
0426  0565  4443          TAD     P               /NO ROTATION, JUST GET P=P*2+0
0427  0566  1022          JMS I   INVERT          /INVERT BIT ORDER AND PUT IN K (NUMBER IN PTH NODE)
0430  0567  4445          TAD     MNOVR2          /SUBTRACT N/2 TO GET NUMBER IN Q (=K) (PS NODE PAIR,)
0431  0570  7000  ADJSGN,  JMS I   GETRIG          /GET REAL AND IMAGINARY PARTS OF W*K,
0432  0571  3035          NOP                     /SET    CIA FOR DOING IFFT, NOP FOR FFT,
0433  0572  1425          DCA     SINE            /SIN(2*PI*K/N)=-IM[W*K], COS IN REGISTER COSINE,
                          TAD I   PR              /FORM [(RE[K)-X(P)] (DIVIDED BY 2)
```

```
0434  0573  4444          JMS I   MULT            /DO REAL PART FIRST=RE[X(P)]*COSINE+IM[X(P)]*SINE
04    0574  0036          COSINE                  /   RE[X(P)]*COSINE=RE[X(P)]*RE[W*K]
0     0575  3033          DCA     ADD2            /   E FOR ADDITION LATER
0437  0576  1426          TAD I   PI              /GET IM[X(P)]
0440  0577  4444          JMS I   MULT
0441  0600  0035          SINE                    /AC=IM[X(P)]*SINE=-IM[W*K]*IM[X(P)]
0442  0601  1033          TAD     ADD2            /AC=RE[W*K]*RE[X(P)]-IM[W*K]*IM[X(P)]=RE[X(P)*W*K]
0443  0602  3037          DCA     GR              /STORE AT GR
0444                /DO IMAG, PART NEXT=IM[X(P)]*COSINE-RE[X(P)]*SINE=IM[X(P)]*RE[W*K]+RE[X(P)]*IM[W*K]
0445  0603  1426          TAD I   PI
0446  0604  4444          JMS I   MULT            /AC=IM[X(P)]
0447  0605  0036          COSINE                  /AC=IM[X(P)]*COSINE=IM[(P)]*RE[W*K]
0450  0606  3033          DCA ADD2                /STORE FOR LATER ADDITION
0451  0607  1425          TAD I PR                /AC=RE[X(P)]
0452  0610  4444          JMS I MULT
0453  0611  0035          SINE            /AC=RE[X(P)]*SINE=-RE[X(P)]*IM[W*K]
0454  0612  7041          CIA             /AC=RE[X(P)]M*IM[W*K]
0455  0613  1033          TAD ADD2        /AC=IM[X(P)]*RE[W*K]+RE[X(P)]*IM[W*K]=IM[X(P)*W*K]
0456  0614  3040          DCA GI          /STORE AT GI, SO GI=IM[X(P)*W*K] AND GR=RE[X(P)*W*K] G=GR+I*GI
0457  0615  1006          TAD S           /LOCATE P NODE PAIR Q, LOCATED S=N/(2*L) UP ARRAY
0460  0616  7041          CIA             /SO SET Q=P-S=INDEX OF NODE PAIR
0461  0617  1025          TAD PR          /LOCATE X(Q) IN MEMORY BY FIXING POINTERS QR AND QI
0462  0620  3023          DCA QR          /TO QS REAL AND IMAG PARTS RESPECTIVELY
0463  0621  1023          TAD     QR
0464  0622  1052          TAD     XLOCDF
0465  0623  3024          DCA     QI
0466  0624  1423          TAD I   QR              /DO THE COMPLEX OPERATIONS:  X(P)<=X(Q)-G; X(Q)<=X(Q)+G
0467  0625  3033          DCA     ADD2            /FIRST DO REAL PART OF X(P), GET RE[X(Q)] AND STORE
0470  0626  1037          TAD     GR              /GET RE[G]
0471  0627  7041          CIA
0472  0630  4441          JMS I   ADDER           /SUBTRACT THEM,
0473  0631  3425          DCA I   PR              /RE[X(P)]<=RE[X(Q)]-RE[G]
0474  0632  1424          TAD I   QI              /COMPUTE IMAG, PART OF X(P), GET IM[X(Q)]
0475  0633  3033          DCA     ADD2            /AND STORE
0476  0634  1040          TAD     GI              /GET IM[G]
0477  0635  7041          CIA
0500  0636  4441          JMS I   ADDER           /AND SUBTRACT THEM,
0501  0637  3426          DCA I   PI              /IM[X(P)]<=IM[X(Q)]-IM[G],X(P) IS NOW DONE,
0502  0640  1423          TAD I   QR              /NEXT COMPUTE X(Q), FIRST REAL PART
0503  0641  3033          DCA     ADD2            /GET RE[G] AND STORE
0504  0642  1037          TAD     GR              /GET RE[G] AND ADD TO FORM
0505  0643  4441          JMS I   ADDER           /RE[X(Q)]+RE[G],
0506  0644  3423          DCA I   QR              /RE[X(Q)]<=RE[X(Q)]+RE[G]
0507  0645  1424          TAD I   QI              /NOW COMPUTE IMAG PART OF X(Q), GET IM[X(Q)]
0510  0646  3033          DCA     ADD2            /AND STORE
0511  0647  1040          TAD     GI              /GET IM[G] AND ADD TO FORM
0512  0650  4441          JMS I   ADDER           /IM[X(Q)]+IM[G]
0513  0651  3424          DCA I   QI              /IM[X(Q)]<=IM[X(Q)]+IM[G], THE NEW NODE PAIR IS COMPUTED,
0514  0652  7040          CMA                     /MOVE UP ARRAY TO NEXT NODE, SET AC=-1
0515  0653  1030          TAD     P               /TO FORM -1
0516  0654  3030          DCA     P               /P<=P-1
0517  0655  7040          CMA
0520  0656  1025          TAD     PR      /DO THE SAME FOR POINTER PR
0521  0657  3025          DCA     PR
0522  0660  1032          TAD     C               /CHECK ON SPACING, IS A NODE WHICH HAS ALREADY BEEN COMPUTED
0523  0661  7041          CIA                     /ABOUT TO BE RE-DONE, OR EQUIVALENTLY,
0524  0662  1006          TAD     S               /IS C=S?
0525  0663  7640          SZA CLA                 /YES.
0526  0664  5277          JMP     CNOTS           /NO, DO NEXT NODE PAIR
0527  0665  1030          TAD     P               /YES, BUT ARE WE AT THE TOP OF THE ARRAY?
0530  0666  7040          CMA                     /OR, IS S=P+1?  (P COMPLEMENTED=-P-1=-(P+1)
0531  0667  1006          TAD     S
```

```
0533  0671  5466          JMP I   RECHK          /YES, DONE WITH THIS ARRAY, DO NEXT ONE.
0534  0672  1006          TAD     S              /NO, MOVE PAST ARE[ THAT HAS AREADY BEEN DONE, OR SET P TO P-S.
0535  0673  7041          CIA                    /BY CHANGING THE POINTER TO RE[X(P)]
0536  0674  1025          TAD     PR
0537  0675  3025          DCA     PR
0540  0676  5465          JMP I   RESETC         /REINITIALIZE C TO 1 SINCE AN UNUSED AREA HAS BEEN ENTERED.
0541  0677  2032  CNOTS,  ISZ     C              /C<=C+1, ANOTHER NODE PAIR HAS BEEN HANDLED.
0542  0700  5464          JMP I   RBUILD         /DO NEXT NODE PAIR IN THIS AREA.
0543  0701  0000  SORTX,  0                      /SUBROUTINE THAT
0544  0702  7040          CMA                    /SORTS OUT TRANSFORMS BY
0545  0703  1003          TAD     N              /BIT INVERSION OF ADDRESS.
0546  0704  3027          DCA     Q              /Q<=N-1, START FROM BOTTOM OF BUFFER
0547  0705  1027  REVERS,  TAD     Q              /P<=BIT INVERTED Q
0550  0706  4443          JMS I   INVERT         /BIT INVERSION ROUTINE
0551  0707  3030          DCA     P
0552  0710  1030          TAD     P              /FORM Q-P
0553  0711  7041          CIA
0554  0712  1027          TAD     Q
0555  0713  7750          SPA SNA CLA            /IS P<Q?
0556  0714  5345          JMP     SWAPED         /NO, HAVE ALREADY DONE THIS PAIR
0557  0715  1030          TAD     P              /YES, SWAP ORDER
0560  0716  1051          TAD     XRLOC          /FIRST SET UP SUBSCRIPT POINTERS FOR X(P) AND X(Q).
0561  0717  3025          DCA     PR
0562  0720  1027          TAD     Q
0563  0721  1051          TAD     XRLOC
0564  0722  3023          DCA     QR
0565  0723  1025          TAD     PR
0566  0724  1052          TAD     XLOCDF
0567  0725  3026          DCA     PI
0570  0726  1023          TAD     QR
0571  0727  1052          TAD     XLOCDF
0572  0730  3024          DCA     QI     /EXCHANGE: X(P)<=X(Q) AND X(Q)<=X(P)
0573  0731  1425          TAD I   PR             /EXCHANGE REAL PARTS, GET RE[X(P)]
0574  0732  3034          DCA     TEMPR          /STORE IT.
0575  0733  1423          TAD I   QR             /GET RE[X(Q)]
0576  0734  3425          DCA I   PR             /MAKE IT RE[X(P)]
0577  0735  1034          TAD     TEMPR          /GET RE[X(P)]
0600  0736  3423          DCA I   QR             /MAKE IT RE[X(Q)]
0601  0737  1426          TAD I   PI             /EXCHANGE IMAGINARY PARTS, GET IM[X(P)]
0602  0740  3034          DCA     TEMPR          /STORE IT.
0603  0741  1424          TAD I   QI             /GET IM[X(Q)]
0604  0742  3426          DCA I   PI             /MAKE IT IM[X(P)]
0605  0743  1034          TAD     TEMPR          /GET IM[X(P)]
0606  0744  3424          DCA I   QI             /MAKE IT IM[X(Q)]
0607  0745  1027  SWAPED,  TAD     Q              /IS Q=0?, IE; ARE WE AT THE TOP OF THE ARRAY
0610  0746  7640          SZA CLA
0611  0747  5352          JMP .+3
0612  0750  6201          CDF0
0613  0751  5701          JMP I   SORTX          /YES, DONE EXIT
0614  0752  7040          CMA                    /NO, Q<=Q-1, IE; MOVE UP THE ARRAY
0615  0753  1027          TAD     Q
0616  0754  3027          DCA     Q
0617  0755  5305          JMP     REVERS         /GO BACK AND CONTINUE
0620                      EJECT
-
```

```
                        *1000
                        /SIGNED S.P. MULTIPLY, USING THE
                        /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
```

```
0713                    /THIS SUBROUTINE FETCHES THE VALUES OF SIN(2*PI*C(AC)/N)
0714                    /AND OF COS(2*PI*C(AC)/N) FOR C(AC) < N/2+1
0715                    /ENTRY:  AC=INDEX OF LOOP UP
0716                    /EXIT :  COS(2*PI*C(AC)/N) STORED AT "COSINE" AND
0717                    /        AC=VALUE OF SIN(2*PI*C(AC)/N).
0720    1060  0000  TRIGET, 0
0721    1061  6201          CDF0
0722    1062  3031          DCA     K               /STORE C(AC) AT K.
0723    1063  7421          MQL                     /CLEAR MQ
0724    1064  1031          TAD     K               /FORM N/4-K.
0725    1065  7141          CLL CIA
0726    1066  1020          TAD     NOVER4
0727    1067  3333          DCA     NO4MIK
0730    1070  7430          SZL                     /IS N/4-K<0?
0731    1071  5310          JMP     QUAD1           /NO, FIRST QUADRANT ANGLE.
0732    1072  1333  QUAD2,  TAD     NO4MIK          /2ND QUADRANT, GET -COS AT K-N/4.
0733    1073  7041          CIA
0734    1074  7417          LSR                     /MAKE CORRECTIVE RIGHT SHIFT ON INDEX.
0735    1075  0000          0
0736    1076  7413          SHL                     /FIND ON SINE TABLE FOR 2*MAXNU BY MULTIPLYING
0737    1077  7402  SHFT1,  HLT                     /INDEX BY 2*(MAXNU-NU), WHICH IS STORED HERE.
0740    1100  1050          TAD     SINLOC          /LOCATE IT IN MEMORY.
0741    1101  3334          DCA     INDEX
0742    1102  1734          TAD I   INDEX
0743    1103  7041          CIA                     /2ND QUADRANT COS IS NEGATIVE.
0744    1104  3036          DCA     COSINE
0745    1105  1333          TAD     NO4MIK          /GET SIN AT N/2-K
0746    1106  1020          TAD     NOVER4
0747    1107  5322          JMP     SINRET
0750    1110  1333  QUAD1,  TAD     NO4MIK          /GET COS AT N/4-K,
0751    1111  7417          LSR
0752    1112  0000          0
0753    1113  7413          SHL
0754    1114  7402  SHFT2,  HLT
0755    1115  1050          TAD     SINLOC
0756    1116  3334          DCA     INDEX
0757    1117  1734          TAD I   INDEX
0760    1120  3036          DCA     COSINE
0761    1121  1031          TAD     K                       /GET SIN AT K.
0762    1122  7417  SINRET, LSR
0763    1123  0000          0
0764    1124  7413          SHL
0765    1125  7402  SHFT3,  HLT
0766    1126  1050          TAD     SINLOC
0767    1127  3334          DCA     INDEX
0770    1130  1734          TAD I   INDEX           /AC=SINVALUE.
0771    1131  6211          CDF1
0772    1132  5660          JMP I   TRIGET
0773    1133  0000  NO4MIK, 0                       /STORAGE FOR N/4-K
0774    1134  0000  INDEX,  0                       /POINTER TO SINE TABLE
0775                    /THIS ROUTINE PERFORMS A SINGLE PRECISION ADD WITH ROUNDING EACH ARGUMENT IS
0776                    /SHIFTED RIGHT ONCE TO PREVENT OVERFLOW OF BINARY POINT (IF NECESSARY)
0777                    /AND THEN CHECKED TO SEE IF IT CAN BE NORMALIZED AFTER ADDITION
1000                    /ENTRY:  AC=ADDEND,C(ADD2)=AUGEND
1001                    /EXIT  :  -AC=RESULT, DIVIDED BY TWO IF NECESSARY.
1002    1135  0000  ADDR,   0
1003    1136  3374          DCA     ADD1
1004    1137  1054          TAD     SHFLAG          /SHOULD ADD BE DONE WITH SHIFT?
1005    1140  7650          SNA CLA
1006    1141  5357          JMP     ADDWOS          /NO,    ADD WITH OUT SHIFT
1007    1142  1374          TAD     ADD1            /YES    T ADDEND
1010    1143  7415          ASR                     /DO 1 SIGNED RIGHT SHIFT
```

```
1012    1145  3374          DCA      ADD1
10      1146  1033          TAD      ADD2
10      1147  7415          ASR                        /  =LO(ADD2)
1015    1150  0000          0                          /MQ(1)=LO(ADD(1))
1016    1151  3033          DCA      ADD2
1017    1152  7501          MQA                        /GET MQ
1020    1153  7004          RAL                        /L<=LO(ADD2); AC0<=LO(ADD1)
1021    1154  7060          CMA CML                    /COMPLEMENT BOTH,
1022
1023    1155  7720          SMA SNL CLA                /IF BOTH WERE=1 (NEITHER=0), INTRODUCE A CARRY.
1024    1156  7001          IAC
1025    1157  1374  ADDWOS, TAD      ADD1               /DO THE ADDITION,
1026    1160  1033          TAD      ADD2
1027    1161  3375          DCA      XSUM               /STORE THE RESULT
1030    1162  1375          TAD      XSUM               /CHECK TO SEE IF ALREADY NORMALIZED,
1031    1163  7510          SPA                         /IS IT POSITIVE?
1032    1164  7041          CIA                         /MAKE IT POSITIVE,
1033    1165  7004          RAL                         /GET BIT 1, WAS NORMALIZED IF =1
1034    1166  7700          SMA CLA
1035    1167  5372          JMP      NOTNOR             /NOT NORMALIZED, LEAVE SHFCHK ALONE,
1036    1170  7001          IAC
1037    1171  3055          DCA      SHFCHK             /SET SHFCHK=1
1040    1172  1375  NOTNOR, TAD      XSUM
1041    1173  5735          JMP I    ADDR               /AND EXIT
1042    1174  0000  ADD1,   0                  /ADDEND STORAGE
1043    1175  0000  XSUM,   0                  /TEMP STORAGE FOR SUM
1044                EJECT
```

```
1046                    /DEFINITIONS FOR EAE
1047                    DVI=7407
1050                    NMI=7411
1051                    SHL=7413
1052                    ASR=7415
1053                    LSR=7417
1054                    MQL=7421
1055                    MUY=7405
1056                    MQA=7501
1057                    CAM=7621
1060                    SCA=7441
1061                    SCL=7403
1062                    /ASSEMBLY PARAMETERS
1063                    BIGSNU=12       /LARGEST TRANSFORMATION HAS DIMENSION 2*10,
1064                            EJECT
```

```
1065                        /MOVING WINDOW DISPLAY SUBROUTINE
1066                        PMODE
1067                        PAGE
1070    1200  0000  IDORA,  0                 /GET BOUNDS
1071    1201  7300          CLA CLL
1072    1202  6201  ACDF0,  CDF 0
1073    1203  1600          TAD I IDORA       /DATA BUFFER
1074    1204  3635          DCA I KMNFLD      /15 BIT
1075    1205  2200          ISZ IDORA         /LOWER BOUND
1076    1206  1600          TAD I IDORA       /AT P+1, P+2
1077    1207  3636          DCA I KMNADR      /MINFLD,MINADR
1100    1210  2200          ISZ IDORA
1101    1211  1600          TAD I IDORA       /UPPER BOUND
1102    1212  3637          DCA I KMXFLD      /AT P+3, P+4
1103    1213  2200          ISZ IDORA
1104    1214  7001          IAC               /RDORA USES
1105    1215  1600          TAD I IDORA       /MAX+1
1106    1216  3640          DCA I KMXADR
1107    1217  7004          RAL
1110    1220  1637          TAD I KMXFLD
1111    1221  3637          DCA I KMXFLD
1112    1222  2200          ISZ IDORA
1113    1223  1600          TAD I IDORA       /Y SHIFT
1114    1224  3111          DCA YSHFT
1115    1225  2200          ISZ IDORA
1116    1226  1600          TAD I IDORA       /Y SCALE
1117    1227  3536          DCA I KYSCAL
1120    1230  1635          TAD I KMNFLD      /INITIALIZE
1121    1231  3641          DCA I KBUFHI      /WINDOW
1122    1232  1636          TAD I KMNADR      /STARTING ADDR
1123    1233  3642          DCA I KBUFLO
1124    1234  5600          JMP I IDORA       /RTN TO SCR N
1125    1235  1415  KMNFLD,  MINFLD
1126    1236  1416  KMNADR,  MINADR
1127    1237  1474  KMXFLD,  MAXFLD
1130    1240  1475  KMXADR,  MAXADR
1131    1241  1574  KBUFHI,  BUFHI
1132    1242  1575  KBUFLO,  BUFLO
1133    1243  0401  P401,    401
1134    1244  1243  DSCLOC,  TAD P401         /DSC X,Y COORD
1135    1245  3274          DCA VCOORD
1136    1246  1112          TAD XCURHI        /FIELD
1137    1247  4261          JMS DSCWD
1140    1250  1113          TAD XCURLO        /ADDRESS
1141    1251  4261          JMS DSCWD
1142    1252  1114          TAD CORVAL        /CONTENTS OF
1143    1253  4261          JMS DSCWD         /CURSR CORE LOC
1144    1254  1115          TAD YCUR          /Y COORD OF
1145    1255  1243          TAD P401
1146    1256  4261          JMS DSCWD         /CURSOR POINT
1147    1257  0000  RTNCDF,  0                /RESTORE USER
1150                                          /DATA FLD
1151    1260  5743          JMP I RDORA       /RTN
1152    1261  0000  DSCWD,   0                /DSC C(AC)
1153    1262  6141          LINC
1154                        LMODE
1155    1263  5276          STC TEMP          /SAVE VALUE
1156    1264  4001          STC XCORD         /CHAN 1
1157    1265  0024          SFA               /VC FOR FULL
1160    1266  0265          ROL I 5           /SIZE IS -40
1161    1267  1020          LDA I             /-20 FOR HALF
1162    1270  7757          -20
```

```
1362    1455  5263          JMP OKEND
1363    1456  1216          TAD MINADR      /RESET TO
1364    1457  3304          DCA BUFPTR      /LOWER BOUND
1365    1460  1215          TAD MINFLD
1366    1461  3316          DCA BOUND
1367    1462  5266          JMP NXTDF
1370    1463  2304  OKEND,  ISZ BUFPTR      /CHK FOR FIELD
1371                                        /BOUNDARY
1372    1464  5267          JMP OKFLD       /ITS OK
1373    1465  2316          ISZ BOUND       /SET NXT FLD
1374    1466  4341  NXTDF,  JMS SETDF
1375    1467  2116  OKFLD,  ISZ COUNT       /512 PNTS ?
1376    1470  5241          JMP NXTPNT      /NO
1377    1471  5672          JMP I .+1       /DSC READ OUT
1400    1472  1244          DSCLOC
1401    1473  4316  CHKHI,  JMS BOUND       /CHK UPR BOUND
1402    1474  0002  MAXFLD, 2
1403    1475  0000  MAXADR, 0
1404    1476  7710  M70,    SPA CLA         /HI WRAP ?
1405    1477  5232          JMP SETFLD
1406    1500  1215          TAD MINFLD      /YES
1407    1501  3374          DCA BUFHI       /RESET TO
1410    1502  1216          TAD MINADR      /LOWER BOUND
1411    1503  5224          JMP WRAP
1412                /DOUBLE PRECISION ADD
1413                /(DBLHI,DBLLO)+(BUFHI,BUFLO)
1414                /RESULT IN (DBLHI,DBLLO)
1415                /(BUFHI,BUFLO)=INITIAL SCOPE ADDRESS
1416
1417    1504  0000  DADD,   0
1420    1505  7300          CLA CLL
1421    1506  1347          TAD DBLLO
1422    1507  1375          TAD BUFLO
1423    1510  3347          DCA DBLLO
1424    1511  7004          RAL
1425    1512  1341          TAD DBLHI
1426    1513  1374          TAD BUFHI
1427    1514  3341          DCA DBLHI
1430    1515  5704          JMP I DADD
1431
1432                /ADD -UPPER OR -LOWER BOUND
1433                /TO (BUFHI,BUFLO)
1434                /BOUND IS AT P+1,P+2 OF CALL
1435
1436    1516  0000  BOUND,  0
1437    1517  1716          TAD I BOUND     /2S COM OF ARG
1440    1520  7140          CMA CLL         /TO DAC
1441    1521  3341          DCA DBLHI
1442    1522  2316          ISZ BOUND
1443    1523  1716          TAD I BOUND
1444    1524  7041          CIA
1445    1525  7430          SZL
1446    1526  2341          ISZ DBLHI
1447    1527  7000  M1000,  NOP
1450    1530  3347          DCA DBLLO
1451    1531  4304          JMS DADD
1452    1532  1341          TAD DBLHI
1453    1533  3377          DCA ENDHI       /DAC HOLDS -NUM
1454    1534  1347          TAD DBLLO       /TO END OF BUF
1455    1535  3376          DCA ENDLO       /NO MATTER F
1456                                        /LOW END WRA
1457    1536  1341          TAD DBLHI       /TO CHK FOR
```

```
1461    1540  5716                JMP I BOUND
1462    1541  0000  'SETDF,'       0                   /SET 8 F
1463    1542  1316                TAD BOUND           /REL TO
1464    1543  7106                CLL RTL
1465    1544  7004                RAL
1466    1545  1201                TAD CCDF0
1467    1546  3347                DCA .+1
1470    1547  0000  DBLLO,         0
1471    1550  5741                JMP I SETDF
1472    1551  3115  CURDIS,       DCA YCUR            /DISP CURSOR
1473    1552  1316                TAD BOUND           /SAVE X,Y
1474    1553  3112                DCA XCURHI          /COORDINATES
1475    1554  1304                TAD BUFPTR
1476    1555  3113                DCA XCURLO
1477    1556  1704                TAD I BUFPTR
1500    1557  3114                DCA CORVAL
1501    1560  1276                TAD M70
1502    1561  3347                DCA DBLLO
1503    1562  1115                TAD YCUR
1504    1563  6141  CURLOP,       LINC
1505                                LMODE
1506    1564  0465                SNS I 5
1507    1565  7365                JMP FREE            /FREE CURSOR
1510    1566  0141                DIS XCORD
1511    1567  0002                PDP
1512                                PMODE
1513    1570  2347                ISZ DBLLO
1514    1571  5363                JMP CURLOP
1515    1572  5250                JMP CURRTN
1516    1573  0000  CURCNT,       0
1517                /THESE 5 GUYS MAY BE PAGE 0
1520    1574  0001  BUFHI,        1
1521    1575  0000  BUFLO,        0
1522    1576  0000  ENDLO,        0
1523    1577  0000  ENDHI,        0
1524                                DBLHI=SETDF
1525                                BUFPTR=DADD
1526                                XCORD=1
1527                                LMODE
1530                                CURSAM=SAM 1        /CURSOR KNOB
1531                                WINSAM=SAM 0        /WINDOW KNOB
1532                                FRESAM=SAM 5        /FREE CURSOR
1533                                SCALE=SCR
1534                                SC12BU=SCR 3        /SCALE FACTOR
1535                                                    /12 BIT UNSIGNED
1536                                OF12BU=4000         /Y OFFSET FOR
1537                                                    /12 BIT UNSIGNED
1540                                CHAIN "FFTC-2"
```

```
0000                    *20
0001                            EJECT
-
```

```
0002                        *       LMODE
0                                   SEGMNT 2
0                                   *20
0005        0020  0647              LDF     7    /BOOTSTRAP IN DIAL MS I/O ROUTINES
0006        0021  0700              RDC
0007        0022  6322              6322
0010        0023  0700              RDC
0011        0024  7323              7323
0012        0025  0643              LDF     3
0013        0026  1020  IFDIAL,     LDA I          /INPUT FROM DIAL TAPE?
0014        0027  2411              QUES1+2000
0015        0030  0602              LIF 2
0016        0031  6720              JMP     ASK
0017        0032  1300              LDA
0020        0033  7043              ANSWER+6000
0021        0034  1460              SAE I
0022        0035  0031              31
0023        0036  0456              SKP
0024        0037  6044              JMP     UNTFIL   /DIAL
0025        0040  1460              SAE I
0026        0041  0016              16
0027        0042  6026              JMP     IFDIAL   /ERROR
0030        0043  6061              JMP     DATTAP           /DATA
0031                    /ASK FOR UNIT NO + FILE NAME
0032        0044  6523  UNTFIL,     JMP     ASK2
0033        0045  6044              JMP     .-1              /ASK AGAIN
0034        0046  0601              LIF     1
0035        0047  1020              LDA I
0036        0050  2375              FDV+2000
0037        0051  6020              JMP     20               /SEARCH INDEX
0040        0052  0456              SKP                      /CANT FIND IT
0041        0053  6063              JMP     MOVINP
0042        0054  0602              LIF     2
0043        0055  1020              LDA I                    /DISPLAY ERROR MSG
0044        0056  2760              MSG1+2000
0045        0057  6720              JMP     ASK
0046        0060  6044              JMP     UNTFIL           /ASK AGAIN
0047        0061  6572  DATTAP,     JMP ASK3
0050        0062  6061              JMP     DATTAP           /ERROR-ASK AGAIN
0051        0063  6466  MOVINP,     JMP FDV2RW       /MOVE INPUT PARAMETERS TO R/W LIST
0052        0064  0643  PTS,        LDF     3
0053        0065  0602              LIF     2
0054        0066  1020              LDA I                    /ASK FOR NO OF PTS
0055        0067  2521              QUES4+2000
0056        0070  6720              JMP     ASK
0057        0071  0061              SET I   1
0060        0072  3043              ANSWER+2000
0061        0073  1020              LDA I            /INPUT IS DECIMAL
0062        0074  0012              12
0063        0075  4701              STC MPLIER
0064        0076  1020              LDA I
0065        0077  7706              -71
0066        0100  4645              STC UPLEGL
0067        0101  1020              LDA I                    /1024 PTS MAX
0070        0102  2000              2000
0071        0103  6627              JMP     CONV
0072        0104  6064              JMP     PTS              /ERROR
0073        0105  0640              LDF     0
0074        0106  1040              STA
0075        0107  2003              N+2000
0076        0110  0643              LDF 3
0077        0111  0017              COM
```

```
0101  0113  1520          SRO 1
0102  0114  0000  TEMP1,  0
0103  0115  0456          SKP
0104  0116  6064          JMP PTS          /BIT 11=1 NOT POWER OF 2
0105  0117  0062          SET I 2
0106  0120  7764          -13
0107  0121  1500  ROTAT,  SRO              /COUNT NO OF BITS SET
0110  0122  0114          TEMP1
0111  0123  6126          JMP ,+3
0112  0124  1120          ADA I
0113  0125  0001          1
0114  0126  0222          XSK I 2
0115  0127  6121          JMP ROTAT
0116  0130  1120          ADA I
0117  0131  7776          -1
0120  0132  0017          COM
0121  0133  0450          AZE
0122  0134  6064          JMP PTS          />1-NOT POWER OF 2
0123  0135  4002          STC 2            /CLEAR
0124  0136  1500  ROT1,   SRO              /DETERMINE POWER OF 2
0125  0137  0114          TEMP1
0126  0140  0456          SKP
0127  0141  6144          JMP STAMU
0130  0142  0222          XSK I 2
0131  0143  6136          JMP ROT1
0132  0144  1000  STAMU,  LDA
0133  0145  0002          2
0134  0146  0640          LDF     0
0135  0147  1040          STA
0136  0150  2004          NU+2000
0137  0151  1120          ADA I
0140  0152  7776          -1
0141  0153  0451          APO
0142  0154  6064          JMP     PTS      /POWER<2
0143  0155  1000          LDA                      /COMPUTE NO OF OUTPUT BLKS
0144  0156  2003          N+2000           /NO OF PTS
0145  0157  0241          ROL 1            /*2
0146  0160  6503          JMP NUMBKS       /CONVERT TO BLKS
0147  0161  1040          STA
0150  0162  2404          FDV+2007         /NO OF BLKS FOR REAL & IMAG
0151  0163  1040          STA
0152  0164  2410          RWPARM+2003
0153  0165  1300          LDH
0154  0166  3046          ANSWER+2003
0155  0167  1460          SAE I
0156  0170  0022          22
0157  0171  6203          JMP IFCOM
0160  0172  1040          STA              /REAL-SET FLAG
0161  0173  2053          REALFG+2000
0162  0174  0640          LDF 0
0163  0175  1000          LDA
0164  0176  2003          N+2000           /COMPUTE INPUT BLKS
0165  0177  6503          JMP NUMBKS
0166  0200  1040          STA
0167  0201  2410          RWPARM+2003
0170  0202  6211          JMP CKEND
0171  0203  1460  IFCOM,  SAE I
0172  0204  0003          3
0173  0205  6064          JMP PTS /ERROR
0174  0206  0011          CLR              /COMPLEX-CLEAR FLAG
0175  0207  1040          STA
0176  0210  2053          REALFG+2000
0177  0211  1000  CKEND,  LDA              /WILL INPUT RUN OFF END OF TAPE
```

```
0200    0212  2407          RWPARM+2002
0201    0213  1100          ADA
0202    0214  2410          RWPARM+2003
0203    0215  1120          ADA I
0204    0216  6777          -1000
0205    0217  0471          APO I
0206    0220  6064          JMP     PTS     /YES
0207    0221  0002          PDP
0210                        PMODE
0211    4222  7200          CLA
0212    4223  1003          TAD N  /ADD 1 BLK FOR SCALE FACTOR IF 400 WORDS OR MORE
0213    4224  7104          CLL RAL         /NO OF OUTPUT WRDS = NO OF PTS*2
0214    4225  1107          TAD M400
0215    4226  7700          SMA CLA
0216    4227  2523          ISZ I PFDV7
0217    4230  6141          LINC
0220                        LMODE
0221    0231  1020  IFFFT,  LDA I           /DO FFT OR JUST DISPLAY?
0222    0232  2625          QUES11+2000
0223    0233  6720          JMP ASK
0224    0234  1300          LDH
0225    0235  7043          ANSWER+6000
0226    0236  1460          SAE I
0227    0237  0004          4
0230    0240  6244          JMP .+4
0231    0241  1060          STA I           /NOT=0 JUST DISPLAY
0232    0242  0000  DISFLG, 0
0233    0243  6251          JMP     FIF
0234    0244  1460          SAE I
0235    0245  0006          6
0236    0246  6231          JMP IFFFT       /ERROR
0237    0247  0011          CLR
0240    0250  4242          STC     DISFLG  /=0 WILL DO TRANSFORM OR INVERSE
0241    0251  1300  FIF,     LDH
0242    0252  7044          ANSWER+6001
0243    0253  1460          SAE I
0244    0254  0024          24
0245    0255  6261          JMP IFI
0246    0256  0011          CLR
0247    0257  4356          STC IFTFLG      /DO FFT
0250    0260  6265          JMP IFDISP
0251    0261  1460  IFI,     SAE I
0252    0262  0011          11
0253    0263  6231          JMP     IFFFT
0254    0264  4356          STC IFTFLG      /DO IFFT
0255    0265  2242  IFDISP, ADD     DISFLG
0256    0266  0470          AZE I
0257    0267  6273          JMP     OUTQES
0260    0270  6466          JMP     FDV2RW  /MOVE OUTPUT PARAMETERS TO R/W
0261    0271  0603          LIF     3       /JUST DISPLAY
0262    0272  6001          JMP     DISPLY
0263                /GET OUTPUT INFO
0264    0273  1020  OUTQES, LDA I
0265    0274  2571          QUES5+2000
0266    0275  6720          JMP     ASK             /PUT ON DIAL TAPE?
0267    0276  1300          LDH
0270    0277  7043          ANSWER+6000
0271    0300  1460          SAE I
0272    0301  0031          31
0273    0302  0456          SKP
0274    0303  6310          JMP     OUTUNT
0275    0304  1460          SAE I
```

```
0277    0306  6273          JMP OUTQES
0300    0307  6452          JMP     ONDAT           /NO
0301    0310  6523  OUTUNT, JMP     ASK2    /ASK FOR UNIT NO & FILE NAME
0302    0311  6310          JMP     OUTUNT  /ERROR
0303    0312  0601          LIF     1
0304    0313  1020          LDA I                   /ENTER IN INDEX
0305    0314  2375          FDV+2000
0306    0315  6022          JMP     22
0307    0316  6430          JMP     SAMNAM          /NAME ALREADY USED
0310    0317  6423          JMP     NOSPAC          /NO SPACE
0311    0320  0002  RDDATA, PDP     /CLEAR DATA BUFFER
0312                        PMODE
0313    4321  7240          CLA CMA
0314    4322  1051          TAD XRLOC
0315    4323  3010          DCA 10
0316    4324  1067          TAD M4000
0317    4325  3011          DCA 11
0320    4326  6211          CDF1
0321    4327  3410          DCA I 10
0322    4330  2011          ISZ 11
0323    4331  5327          JMP .-2
0324    4332  6201          CDF0
0325    4333  6212          CIF     10      /READ IN DATA
0326    4334  4534          JMS I   PREAD
0327    4335  6405          RWPARM
0330    4336  6201          CDF0
0331    4337  7200          CLA
0332    4340  1532          TAD I PRELFG    /REAL OR COMPLEX
0333    4341  7640          SZA CLA
0334    4342  5357          JMP   PROC      /REAL
0335    4343  7040          CMA             /MOVE IMAG PARTS TO 2000
0336    4344  1003          TAD     N       /OLD ADDR = NO OF PTS
0337    4345  3010          DCA     10
0340    4346  1110          TAD     C1777   /NEW ADDR = 2000
0341    4347  3011          DCA     11
0342    4350  1003          TAD     N
0343    4351  7041          CIA
0344    4352  3034          DCA     TEMPR   /CTR
0345    4353  3130          DCA     CMPFLG  /DONT COMPLEMENT
0346    4354  4527          JMS I   PMVPTS  /MOVE THEM
0347    4355  5357          JMP     PROC
0350
0351    4356  0000  IFTFLG, 0               /0=FFT NON0=IFFT
0352
0353    4357  3532  PROC,   DCA I   PRELFG  /OUTPUT WILL BE COMPLEX REGARDLESS OF INPUT
0354    4360  1356          TAD IFTFLG      /DO IFFT?
0355    4361  7650          SNA CLA
0356    4362  5365          JMP     FT      /NO
0357    4363  4447  IFT,    JMS I DOIFFT
0360    4364  7410          SKP
0361    4365  4446  FT,     JMS I DOFFT
0362    4366  4442          JMS I SORT      /PUT IN SEQUENTIAL ORDER
0363    4367  1053  STSCAL, TAD SCAL
0364    4370  6211          CDF1
0365    4371  3034          DCA     TEMPR    /SAVE
0366    4372  1003          TAD     N
0367    4373  7104          CLL RAL
0370    4374  3036          DCA     COSINE  /NO OF PTS*2
0371    4375  1034          TAD     TEMPR
0372    4376  3436          DCA I   COSINE    /STORE SCALE FACTOR AFTER DATA
0373    4377  6201  NOWSTR, CDF0
0374    4400  1110          TAD     C1777   /OLD ADDR = 2000
0375    4401  3010          DCA     10
```

```
0376    4402  7040          CMA             /NEW ADDR = NO OF PTS
0       4403  1003          TAD     N
0       4404  3011          DCA     11
0401    4405  1003          TAD     N
0402    4406  7041          CIA
0403    4407  3034          DCA     TEMPR   /CTR
0404    4410  3130          DCA     CMPFLG  /DONT COMPLEMENT
0405    4411  4527          JMS I   PMVPTS  /PACK IMAG PARTS BEHIND REAL
0406    4412  6141          LINC
0407                        LMODE
0410    0413  6466          JMP FDV2RW
0411    0414  0002          PDP
0412                        PMODE
0413    4415  6212          CIF     10      /WRITE OUT DATA
0414    4416  4535          JMS I   PWRITE
0415    4417  6405          RWPARM
0416    4420  6141          LINC
0417                        LMODE
0420    0421  0603          LIF 3
0421    0422  6001          JMP DISPLY
0422    0423  0602  NOSPAC, LIF     2
0423    0424  1020          LDA I
0424    0425  3013          MSG2+2000
0425    0426  6720          JMP     ASK
0426    0427  6273          JMP     OUTQES          /ASK OUTPUT QUESTIONS AGAIN
0427    0430  0602  SAMNAM, LIF     2
0430    0431  1020          LDA I                   /NAME ALREADY EXISTS
0431    0432  2612          QUES6+2000
0432                                                /REPLACE WITH NEW FILE?
0433    0433  6720          JMP     ASK
0434    0434  1300          LDH
0435    0435  7043          ANSWER+6000
0436    0436  1460          SAE I
0437    0437  0031          31
0440    0440  0456          SKP
0441    0441  6446          JMP     REPL
0442    0442  1460          SAE I
0443    0443  0016          16
0444    0444  6430          JMP     SAMNAM
0445    0445  6310          JMP     OUTUNT  /NO-ASK FOR NAME AGAIN
0446    0446  0601  REPL,   LIF 1
0447    0447  6024          JMP     24
0450    0450  6423          JMP     NOSPAC
0451    0451  6320          JMP     RDDATA
0452    0452  0602  ONDAT,  LIF     2
0453    0453  6572          JMP ASK3        /ASK FOR UNIT/BLK NO
0454    0454  6452          JMP ONDAT       /ERROR
0455    0455  1000          LDA
0456    0456  2403          FDV+2006        /BLK NO
0457    0457  1100          ADA
0460    0460  2404          FDV+2007        /NO OF BLKS
0461    0461  1120          ADA I
0462    0462  6777          -1000
0463    0463  0471          APO I
0464    0464  6423          JMP NOSPAC      /NOT ENOUGH BLKS LEFT
0465    0465  6320          JMP     RDDATA
0466                        /MOVE FDV PARAMETERS TO R-W LIST
0467    0466  1000  FDV2RW, LDA
0470    0467  2375          FDV+2000
0471    0470  1040          STA
0472    0471  2405          RWPARM+2000
```

```
0475    0474  1040          STA
0476    0475  2407          RWPARM+2002
0477    0476  1000          LDA
0500    0477  2404          FDV+2007
0501    0500  1040          STA
0502    0501  2410          RWPARM+2003
0503    0502  6000          JMP 0
0504                  /CONVERT WORDS TO BLOCKS
0505    0503  4114  NUMBKS, STC TEMP1
0506    0504  2000          ADD 0
0507    0505  4522          STC NUMBKX
0510    0506  2114          ADD TEMP1
0511    0507  0643          LDF     3
0512    0510  0061          SET I   1
0513    0511  0001          1
0514    0512  1120          ADA I
0515    0513  7377          -400
0516    0514  0451          APO
0517    0515  6520          JMP .+3
0520    0516  0221          XSK I   1
0521    0517  6512          JMP     .-5
0522    0520  1000          LDA
0523    0521  0001          1
0524    0522  0000  NUMBKX, 0
0525                  /ASK FOR UNIT NUMBER & FILE NAME
0526                  /CONV & STORE UNIT NUMBER
0527                  /MOVE FILE NAME TO ENTER, LOOKUP PARAMETER LIST
0530                  /STORE UNIT THRU B3
0531    0523  1000  ASK2,   LDA
0532    0524  0000          0
0533    0525  4571          STC     ASK2X
0534    0526  0602          LIF     2
0535    0527  6711          JMP     OCTL    /CHANGE PARAMETERS TO HANDLE OCTAL NUMBERS
0536    0530  1020          LDA I
0537    0531  2453          QUES2+2000
0540    0532  6720          JMP     ASK     /PT TO UNIT NO-1H
0541    0533  0061          SET I   1
0542    0534  3043          ANSWER+2000
0543    0535  1020          LDA I           /MAX VALUE
0544    0536  0017          17
0545    0537  5627          JMP CONV
0546    0540  6571          JMP     ASK2X   /ERROR
0547    0541  1040          STA             /STORE UNIT
0550    0542  2375          FDV+2000
0551                  /MOVE FILE NAME FROM ANSWER BUFFER TO LOOKUP, ENTER PARAMETER LIST
0552    0543  0061          SET I 1
0553    0544  7044          ANSWER+6001
0554    0545  0062          SET I   2
0555    0546  6375          FDV+6000                /LEFT HALF 1ST OF FDV+1
0556    0547  0063          SET I   3
0557    0550  7767          -10             /8 CHARS
0560    0551  1321          LDH I   1       /IF 1ST CHAR OF NAME
0561    0552  0470          AZE I           /=00, NO NAME WAS
0562    0553  6571          JMP     ASK2X   /ENTERED-ERROR
0563    0554  0456          SKP
0564    0555  1321  INFILE, LDH I   1
0565    0556  0450          AZE                     /FILL TO 8 CHARS WITH 77
0566    0557  6562          JMP     .+3
0567    0560  1320          LDH I
0570    0561  7700          7700
0571    0562  1362          STH I   2
0572    0563  0223          XSK I   3
0573    0564  6555          JMP     INFILE
```

```
05      0565  1020          LDA I                   /    IRN CALL+2
05      0566  0001          1
0576    0567  1140          ADM
0577    0570  0571          .+1
0600    0571  0000  ASK2X,   0
0601                /ASK FOR UNIT NUMBER + BLK NO AND CONVERT
0602                /STORE UNIT THRU 87
0603                /"  BLK NO " 810
0604    0572  1000  ASK3,    LDA
0605    0573  0000          0
0606    0574  4626          STC     ASK3X
0607    0575  0002          LIF     2
0610    0576  1020          LDA I
0611    0577  2475          QUES3+2000
0612    0600  6720          JMP     ASK
0613    0601  0061          SET I   1               /ADDR-1H OF 1ST CHAR - UNIT
0614    0602  3043          ANSWER+2000
0615    0603  6711          JMP OCTL        /CHANGE PARAMETERS TO HANDLE OCTAL
0616    0604  1020          LDA I                   /MAX VALUE
0617    0605  0017          17
0620    0606  6627          JMP CONV
0621    0607  6626          JMP ASK3X       /ERROR
0622    0610  1040          STA
0623    0611  2375          FDV+2000
0624    0612  0061          SET I 1
0625    0613  7044          ANSWER+6001
0626    0614  1020          LDA I           /MAX VALUE FOR BLK NO
0627    0615  0777          777
0630    0616  6627          JMP     CONV
0631    0617  6626          JMP     ASK3X
0632    0620  1040          STA
0633    0621  2403          FDV+2006
0634    0622  1020          LDA I                   /EXIT CALL+2
0635    0623  0001          1
0636    0624  1140          ADM
0637    0625  0626          .+1
0640    0626  0000  ASK3X,   0
0641                /CONVERT NUMBER IN ANSWER BUFFER TO BINARY
0642                /ENTER WITH MAX LEGAL VALUE IN AC
0643                /IF LEGAL - EXIT CALL+2 WITH VALUE IN AC
0644    0627  0017  CONV,    COM
0645    0630  4675          STC     TEMP2 /COMPLEMENT MAX VALUE
0646    0631  4114          STC     TEMP1
0647    0632  2000          ADD     0               /RETURN ADDR
0650    0633  4674          STC     CONVER
0651    0634  1321  NXTCHR, LDH I   1               /GET A CHAR
0652    0635  0470          AZE I
0653    0636  6660          JMP ERRCHK
0654    0637  1120          ADA I
0655    0640  7720          7720                    /-60 -2
0656                /S COMP
0657    0641  0451          APO
0660    0642  6650          JMP     CHKEND          /<60 MAYBE TERM  ATING CHAR
0661    0643  1301          LDH     1               /CK UPPER LIMIT
0662    0644  1120          ADA I
0663    0645  7710  UPLEGL, -67             /UPPER LIM=7 OR 9
0664    0646  0451          APO
0665    0647  6676          JMP     MULPLY          /0-9 -0K
0666    0650  1301  CHKEND, LDH     1               /<9 MAYBE TERMINATING C
0667    0651  1460          SAE I
0673    0652  0034          34
```

```
0674    0656  0074          74
0675    0657  6674          JMP     CONVER          /ILLEGAL CHAR
0676    0660  1000  ERRCHK,  LDA                     /=34 OR 74 - NUMBER COMPLETED
0677    0661  0114          TEMP1
0700    0662  1100          ADA                     /ERROR CHECK SIZE
0701    0663  0675          TEMP2
0702    0664  0471          APO I
0703    0665  6674          JMP     CONVER          /TOO LARGE
0704    0666  1020          LDA I                   /OK STEP EXIT
0705    0667  0001          1
0706    0670  1140          ADM
0707    0671  0674          .+3
0710    0672  1000          LDA                     /EXIT WITH VALUE IN AC
0711    0673  0114          TEMP1
0712    0674  0000  CONVER,  0
0713    0675  0000  TEMP2,   0
0714                        /
0715    0676  1000  MULPLY,  LDA             /VALUE SO FAR
0716    0677  0114          TEMP1
0717    0700  1260          MUL I
0720    0701  0010  MPLIER,  10
0721    0702  4114          STC     TEMP1
0722    0703  1301          LDH     1               /+ THIS VALUE
0723    0704  1560          BCL I
0724    0705  7760          7760
0725    0706  1140          ADM
0726    0707  0114          TEMP1
0727    0710  6634          JMP     NXTCHR
0730                        /
0731                        /CHANGE PARAMETERS SO CONV & MULPLY WILL HANDLE OCTAL NUMBERS
0732    0711  1020  OCTL,   LDA I
0733    0712  0010          10
0734    0713  4701          STC     MPLIER
0735    0714  1020          LDA I
0736    0715  7710          -67
0737    0716  4645          STC     UPLEGL
0740    0717  6000          JMP     0
0741                        /
0742                        /DISPLAY QUESTIONS
0743    0720  4734  ASK,    STC     QUESNO          /ADDR OF TEXT
0744    0721  2000          ADD     0
0745    0722  4740          STC     ASKX
0746    0723  0500          IOB
0747                        PMODE
0750    4724  6234          RIB
0751                        LMODE
0752    0725  0343          SCR 3
0753    0726  1560          BCL I
0754    0727  7740          7740
0755    0730  1120          ADA I
0756    0731  0600          LIF 0
0757    0732  4737          STC ASKX-1
0760    0733  7000          JMP     QAINIT          /DISPLAY
0761    0734  0000  QUESNO,  0
0762    0735  3043          ANSWER+2000
0763    0736  7053          JMP     QARFSH          /WAIT FOR ANSWERS
0764    0737  0000          0
0765    0740  0000  ASKX,    0
0766
0767                        EJECT
```

```
0770                            SEGMNT 3
0771                            LMODE
0772                            *1
0773    0001    0002  DISPLY,   PDP
0774                            PMODE
0775    6002    7200            CLA
0776    6003    1003            TAD     N
0777    6004    7110            CLL RAR
1000    6005    3037            DCA     GR      /NO OF PTS/2
1001    6006    1037            TAD     GR
1002    6007    7041            CIA
1003    6010    3033            DCA     ADD2    /-NO OF PTS/2
1004    6011    1140            TAD     C2000
1005    6012    3364            DCA     LOADDR  /LOWER ADDR OF DISPLAY
1006    6013    1110            TAD     C1777
1007    6014    1003            TAD     N
1010    6015    3366            DCA     UPADDR  /UPPER ADDR OF DISPLAY
1011    6016    6212            CIF     10      /READ IN DATA
1012    6017    4534            JMS I   PREAD
1013    6020    6405            RWPARM
1014    6021    6141            LINC
1015                            LMODE
1016    0022    1020  DISPL1,   LDA I           /WHICH DISPLAY
1017    0023    2665            QUES13+2000
1020    0024    0602            LIF 2
1021    0025    6720            JMP ASK
1022    0026    1300            LDH
1023    0027    7043            ANSWER+6000
1024    0030    0450            AZE
1025    0031    6034            JMP .+3
1026    0032    0602            LIF 2
1027    0033    6026            JMP IFDIAL      /LINE FEED
1030    0034    0002            PDP
1031                            PMODE
1032    6035    1077  WCHDIS,   TAD M11
1033    6036    7450            SNA
1034    6037    5255            JMP DPIMAG      /IMAG
1035    6040    1075            TAD M4
1036    6041    7450            SNA
1037    6042    5476            JMP I PDPMAG    /MAGNITUDE
1040    6043    1100            TAD M5
1041    6044    7450            SNA
1042    6045    5317            JMP DPREAL      /REAL
1043    6046    1070            TAD M1
1044    6047    7650            SNA CLA
1045    6050    5654            JMP I PDPSCL    /SCALE FACTOR
1046    6051    6141  DISPER,   LINC            /ERROR
1047                            LMODE
1050    0052    6022            JMP DISPL1
1051    0053    0000  REALFG,   0
1052                            PMODE
1053    6054    6226  PDPSCL,   DPSCAL
1054    6055    1253  DPIMAG,   TAD REALFG
1055    6056    7640            SZA CLA
1056    6057    5251            JMP DISPER      /NO IMAG PARTS TO DISPLAY
1057    6060    1533            TAD I   PIFTFG  /IF TRANSFORM WAS DONE, SWAP HALVES
1060    6061    7640            SZA CLA
1061    6062    5310            JMP     NOSWPI  /INVERSE WAS DONE
1062    6063    7040            CMA
1063    6064    1003            TAD     N
1064    6065    3010            DCA     10      /OLD LOW ADDR OF 1ST 1/2 = NO OF PTS
1065    6066    1037            TAD     GR      /NEW LOW ADDR OF 1ST 1/2 = 2000 + NO OF PTS/2
```

```
1067    6070  3011          DCA     11
1070    6071  1033          TAD     ADD2
1071    6072  3034          DCA     TEMPR    /MOVE 1/2 OF PTS
1072    6073  7201          CLA IAC
1073    6074  3130          DCA     CMPFLG   /COMPLEMENT VALUES
1074    6075  4527          JMS I   PMVPTS   /MOVE THEM
1075    6076  7040          CMA              /OLD ADDR OF 2ND 1/2 = 3/2 NO OF PTS
1076    6077  1037          TAD     GR
1077    6100  1003          TAD     N
1100    6101  3010          DCA     10
1101    6102  1110          TAD     C1777    /NEW ADDR OF 2ND 1/2 = 2000
1102    6103  3011          DCA     11
1103    6104  1033          TAD     ADD2
1104    6105  3034          DCA     TEMPR    /1/2 OF PTS
1105    6106  4527          JMS I   PMVPTS   /MOVE THEM - 1ST 1/2 IS NOW 2ND 1/2; 2ND 1/2 IS NOW 1ST 1/2
1106    6107  5330          JMP     PREPAR
1107    6110  1003  NOSWPI, TAD     N        /LOW ADDR OF IMAG = NO OF PTS
1110    6111  3364          DCA     LOADDR
1111    6112  7040          CMA              /HIGH ADDR = 2*NO OF PTS-1
1112    6113  1003          TAD     N
1113    6114  1003          TAD     N
1114    6115  3366          DCA     UPADDR
1115    6116  5330          JMP     PREPAR
1116
1117    6117  1533  DPREAL, TAD I   PIFTFG   /IF TRANSFORM WAS DONE, SWAP HALVES
1120    6120  7640          SZA CLA
1121    6121  5324          JMP     NOSWPR
1122    6122  4526          JMS I   PMRLMG   /SWAP
1123    6123  5330          JMP     PREPAR
1124    6124  3364  NOSWPR, DCA     LOADDR   /LOW ADDR OF REAL
1125    6125  7040          CMA
1126    6126  1003          TAD     N
1127    6127  3366          DCA     UPADDR   /HIGH ADDR = NO OF PTS-1
1130
1131    6130  1033  PREPAR, TAD ADD2         /NO OF PTS <1000?
1132    6131  7104          CLL RAL
1133    6132  1137          TAD C1000
1134    6133  7550          SPA SNA
1135    6134  5347          JMP GQ1000
1136    6135  7110          CLL RAR          /YES
1137    6136  7001          IAC              /CENTER DISPLAY
1140    6137  1141          TAD M1K
1141    6140  3525          DCA I PLEFTX     /1000-(1000-NO OF PTS/2) 1,S COMP
1142    6141  1146          TAD CCLR
1143    6142  3524          DCA I PMVDIS
1144    6143  1033          TAD ADD2         /WIDTH OF DISPLAY
1145    6144  7104          CLL RAL          /NO OF PTS
1146    6145  3131          DCA MINPTS
1147    6146  5362          JMP SHOWIT
1150    6147  7200  GQ1000, CLA
1151    6150  1141          TAD M1K          /LEFT JUSTIFY DISPLAY
1152    6151  3525          DCA I PLEFTX     /-1000 1,S COMP
1153    6152  1525          TAD I PLEFTX     /WIDTH OF DISPLAY
1154    6153  7001          IAC
1155    6154  3131          DCA MINPTS
1156    6155  1145          TAD SCR4         /MOVE DISPLAY
1157    6156  3524          DCA I PMVDIS
1160    6157  5362          JMP SHOWIT
1161                  /DISPLAY DATA
1162    6160  6141  REDPLY, LINC
1163                  LMODE
1164    0161  6001          JMP DISPLY
```

```
1166
1167    6162  4517  SHOWIT, JMS I KIDORA
        6163  0001          1                 /LOW ADDRESS FIELD
1171    6164  0000  LOADDR, 0                 / "    "       "
1172    6165  0001          1                 /HIGH   "     "
1173    6166  0000  UPADDR, 0                 / "     "
1174    6167  0000          0                 /Y OFFSET
1175                        LMODE
1176    0170  0343  SIZE,   SCR 3              /SCALE
1177                        PMODE
1200    6171  4520  RFRSH,  JMS I KRDORA      /REFRESH UNTIL LF IS HIT
1201    6172  6031          KSF
1202    6173  5371          JMP .-2
1203    6174  6036          KRB
1204    6175  1102          TAD M215
1205    6176  7650          SNA CLA
1206    6177  5360          JMP REDPLY
1207    6200  6036          KRB
1210    6201  1106          TAD M261
1211    6202  7650          SNA CLA
1212    6203  5216          JMP LARGER
1213    6204  6036          KRB
1214    6205  1103          TAD M321
1215    6206  7650          SNA CLA
1216    6207  5211          JMP SMALLR
1217    6210  5522          JMP I PRFRSH
1220    6211  1536  SMALLR, TAD I KYSCAL
1221    6212  1104          TAD M353
1222    6213  7710          SPA CLA
1223    6214  2536          ISZ I KYSCAL
1224    6215  5522          JMP I PRFRSH
1225    6216  1536  LARGER, TAD I KYSCAL
1226    6217  1105          TAD M340
1227    6220  7750          SPA SNA CLA
1230    6221  5522          JMP I PRFRSH
1231    6222  7040          CMA
1232    6223  1536          TAD I KYSCAL
1233    6224  3536          DCA I KYSCAL
1234    6225  5522          JMP I PRFRSH
1235
1236                  /DISPLAY SCALE FACTOR
1237
1240    6226  1532  DPSCAL, TAD I   PRELFG
1241    6227  7640          SZA CLA
1242    6230  5772          JMP I   PDSPER   /JUST REAL MEANS I DIDNT MAKE FILE = NO SCALE FACTOR
1243    6231  1003          TAD     N
1244    6232  7104          CLL RAL
1245    6233  3034          DCA     TEMPR    /ADDR = NO OF PTS*2
1246    6234  6211          CDF1
1247    6235  1434          TAD I   TEMPR
1250    6236  1077          TAD     M11
1251    6237  7740          SMA SZA CLA
1252    6240  5244          JMP     GR9      />9
1253    6241  1074          TAD     LESS10
1254    6242  1434          TAD I   TEMPR    /SPACE + ASCII SCALE FACTOR
1255    6243  5247          JMP     SHOSCL
1256    6244  1071  GR9,    TAD     M12
1257    6245  1434          TAD I   TEMPR
1260    6246  1073          TAD     GRET10   /10+SCALE FACTOR-10
1261    6247  6201  SHOSCL, CDF0
1262    6250  3266          DCA     DPMAG-2  /STORE IN DISPLAY PARAMETERS
1263    6251  6141          FINC
```

```
1265    0252  1020          LDA I
1266    0253  2257          SCLFAC+2000
1267    0254  0602          LIF 2
1270    0255  6720          JMP     ASK     /DISPLAY IT
1271    0256  6001          JMP     DISPLY
1272                  SCLFAC, TEXT Z
1273    0257  4347
1273
1274    0260  4043
1274
1275    0261  4043
1275    0262  0640
1275    0263  4040
1275    0264  4040
1275    0265  4040
1275    0266  4040
1275    0267  3400
1275                  F       \Z
1276
1277                  PMODE
1300
1301                  /COMPUTE MAGNITUDE
1302    6270  1532    DPMAG,  TAD I PRELFG
1303    6271  7640            SZA CLA
1304    6272  5772            JMP I PDSPER
1305    6273  7040            CMA
1306    6274  1003            TAD     N
1307    6275  3010            DCA     10      /LOW ADDR OF IMAG = NO OF PTS
1310    6276  1110            TAD     C1777
1311    6277  3011            DCA     11      /MOVE TO 2000
1312    6300  1003            TAD     N
1313    6301  7041            CIA
1314    6302  3034            DCA     TEMPR   /CTR
1315    6303  3130            DCA     CMPFLG  /DONT COMPLEMENT VALUES
1316    6304  4527            JMS I   PMVPTS  /MOVE IMAG TO 2000
1317    6305  3036            DCA     COSINE  /ADDR OF 1ST REAL
1320    6306  1110            TAD     C1777   /ADDR-1 OF IMAG PARTS
1321    6307  3011            DCA     11
1322    6310  1101            TAD     C6000   /FRAC MULT
1323    6311  3324            DCA     RELPTR
1324    6312  1101            TAD     C6000
1325    6313  3336            DCA     IMGPTR
1326    6314  1033            TAD     ADD2
1327    6315  7104            CLL RAL
1330    6316  3034            DCA     TEMPR   /-NO OF PTS
1331    6317  6211            CDF1
1332    6320  1436    NXTMAG, TAD I   COSINE  /REAL PART
1333    6321  6141            LINC
1334                          LMODE
1335    0322  0644            LDF     4       /FIELD OF REAL
1336    0323  1240            MUL             /MULT BY ITSELF
1337    0324  0000    RELPTR, 0
1340    0325  0363            SCR I   3       /1 BECAUSE PROD IS SHIFTED LEFT 1, 2 BECAUSE MAX VALUE WILL OVERFLOW - TAKE OUT
                      SQUARED
1341    3326  0002            PDP
1342                          PMODE
1343    6327  3143            DCA     DPSQ+1  /SAVE D.P. SQ
1344    6330  7501            MQA
1345    6331  3142            DCA     DPSQ
1346    6332  1411            TAD I   11      /IMAG PART
1347    6333  6141            LINC
1350                          LMODE
```

```
1352    0335  1240            MUL
1353    0336  0000  IMGPTR,   0
1354    0337  0363            SCR I   3       /SAME REASON    REAL
1355    0340  0002            PDP
1356                          PMODE
1357    6341  3035            DCA     SINE    /SAVE H.O.
1360    6342  7100            CLL
1361    6343  7501            MQA             /L.O. OF IMAG
1362    6344  1142            TAD     DPSQ    /L.O. OF REAL
1363    6345  3142            DCA     DPSQ
1364    6346  7004            RAL             /OVERFLOW IF ANY
1365    6347  1035            TAD     SINE    /H.O. OF IMAG
1366    6350  1143            TAD     DPSQ+1  /H.O. OF REAL
1367    6351  3143            DCA     DPSQ+1
1370    6352  4773            JMS I   PSQRT   /TAKE SQ RT
1371    6353  3436            DCA I   COSINE  /STORE IN PLACE OF REAL
1372    6354  2036            ISZ     COSINE  /STEP REAL PTR
1373    6355  2336            ISZ     IMGPTR  / "   IMAG ADDR - ON LAST PT OF 1024 PTS WILL SKIP
1374    6356  2324            ISZ     RELPTR  /STEP REAL ADDR
1375    6357  2034            ISZ     TEMPR   /STEP CTR
1376    6360  5320            JMP     NXTMAG
1377    6361  6141            LINC
1400                          LMODE
1401    0362  0643            LDF     3
1402    0363  0002            PDP
1403                          PMODE
1404    6364  1533            TAD I   PIFTFG  /IF TRANSFORM WAS DONE, SWAP HALVES
1405    6365  7640            SZA CLA
1406    6366  5774            JMP I   PNSWPR
1407    6367  4526            JMS I   PMRLMG
1410    6370  5771            JMP I   PPREPR  /DISPLAY MAG
1411                  /
1412    6371  6130  PPREPR,   PREPAR
1413    6372  6051  PDSPER,   DISPER
1414    6373  7052  PSQRT,    SQRT
1415    6374  6124  PNSWPR,   NOSWPR
1416                          EJECT
```

```
                        LMODE
                /LOOKUP, ENTER PARAMETER LIST
0375  0000    FDV,    0               /UNIT 0
0376  0000            0               /FILE NAME - 8 CHAR
0377  0000            0
0400  0000            0
0401  0000            0
0402  0002            2               /BINARY
0403  0000            0               /BLK NO
0404  0000            0               /NO OF BLKS
                /
0405  0000    RWPARM, 0000            /UNIT
0406  0020            0020            /BUFFER ADDR
0407  0000            0               /BLK NO
0410  0000            0               /NO OF BLKS
                /
                LMODE
                /QUESTIONS
0411  4040
0412  4040
0413  4023
0414  1116
0415  0714
0416  0540
0417  2022
0420  0503
0421  1123
0422  1117
0423  1640
0424  0606
                QUES1,  TEXT Z     SINGLE PRECISION FFT
0425  2443
0426  4740
0427  4347
0430  4043
0431  0640
0432  4011
0433  1620
0434  2524
0435  4017
                F  INPUT ON
0436  1643
0437  4740
0440  4306
0441  4040
0442  0411
0443  0114
0444  4025
0445  1611
0446  2477
0447  4031
0450  5716
0451  7461
0452  3400
                F  DIAL UNIT? Y/N<1\Z
                QUES2,  TEXT Z
-453  4043
  54  0625
-455  1611
0456  2440
```

```
1450    0457  1625
1450    0460  1502
1450    0461  0522
1450                  FUNIT NUMBER<2
1451    0462  7462
1451    0463  4347
1451
1452    0464  4043
1452    0465  0606
1452    0466  1114
1452    0467  0540
1452    0470  1601
1452    0471  1505
1452    0472  4040
1452    0473  7470
1452    0474  3400
1452                  FFILE NAME  <8\Z
1453                  QUES3,  TEXT Z
1454    0475  4043
1454    0476  0640
1454    0477  4025
1454    0500  1611
1454    0501  2440
1454    0502  1625
1454    0503  1502
1454    0504  0522
1454                  F  UNIT NUMBER<2
1455    0505  7462
1455    0506  4347
1455
1456    0507  4043
1456    0510  0640
1456    0511  4002
1456    0512  1413
1456    0513  4016
1456    0514  2515
1456    0515  0205
1456    0516  2240
1456    0517  7463
1456    0520  3400
1456                  F  BLK NUMBER <3\Z
1457                  QUES4,  TEXT Z
1460    0521  4043
1460    0522  0610
1460    0523  1727
1460    0524  4015
1460    0525  0116
1460    0526  3140
1460    0527  2024
1460    0530  2377
1460                  FHOW MANY PTS? <4
1461    0531  7464
1461    0532  4347
1461
1462    0533  4043
1462    0534  4740
1462    0535  5064
1462    0536  5561
1462    0537  6062
1462    0540  6440
1462    0541  0231
1462    0542  4020
```

```
1462      0545  2340
1462      0546  1706
1462      0547  4062
1462                        (4=1024 BY POWERS OF 2)
1463      0550  5143
1463
1464      0551  4740
1464      0552  4306
1464      0553  2205
1464      0554  0114
1464      0555  4017
1464                  FREAL OR
1465      0556  2243
1465
1466      0557  4740
1466      0560  4306
1466      0561  0317
1466      0562  1520
1466      0563  1405
1466      0564  3077
1466      0565  4022
1466      0566  5703
1466      0567  7461
1466      0570  3400
1466                  FCOMPLEX? R/C<1\Z
1467                  QUES5,  TEXT Z
1470      0571  4306
1470      0572  4040
1470      0573  1725
1470      0574  2420
1470      0575  2524
1470      0576  4017
1470                  F  OUTPUT ON
1471      0577  1643
1471      0600  0640
1471      0601  4004
1471      0602  1101
1471      0603  1440
1471      0604  2516
1471      0605  1124
1471      0606  7740
1471      0607  3157
1471      0610  1674
1471      0611  6134
1471                  F  DIAL UNIT? Y/N<1\Z
1472                  QUES6,  TEXT Z
1473      0612  4043
1473
1474      0613  4740
1474      0614  4306
1474      0615  4022
1474      0616  0520
1474      0617  1401
1474      0620  0305
1474      0621  7740
1474      0622  3157
1474      0623  1674
1474      0624  6134
1474                  F REPLACE? Y/N<1\Z
1475                  QUES11, TEXT Z
1476      _ _5  4306
1476      0626  0606
```

```
1476       0627  2440
1476       0630  1722
1476       0631  4004
1476       0632  1123
1476       0633  2014
1476       0634  0131
1476       0635  7740
1476       0636  0657
1476       0637  0474
1476                     FFFT OR DISPLAY? F/D<1
1477       0640  6143
1477
1500       0641  4740
1500       0642  4347
1500
1501       0643  4043
1501       0644  0624
1501       0645  2201
1501       0646  1623
1501       0647  0617
1501       0650  2215
1501       0651  4017
1501                     FTRANSFORM OR
1502       0652  2243
1502
1503       0653  4740
1503       0654  4306
1503       0655  1116
1503       0656  2605
1503       0657  2223
1503       0660  0577
1503       0661  4024
1503       0662  5711
1503       0663  7461
1503       0664  3400
1503                     FINVERSE? T/I<1\Z
1504                     QUES13, TEXT Z
1505       0665  4306
1505       0666  2710
1505       0667  1103
1505       0670  1040
1505       0671  0411
1505       0672  2320
1505       0673  1401
1505       0674  3177
1505                     FWHICH DISPLAY?<1
1506       0675  7461
1506       0676  4347
1506
1507       0677  4043
1507       0700  4740
1507       0701  4040
1507       0702  4040
1507       0703  2250
1507       0704  0501
1507                               R(EAL)
1510       0705  1451
1510       0706  4347
1510       0707  4040
1510       0710  4040
1510       0711  4011
1510       0712  5015
1510       0713  0107
```

```
1510      0715  0122
1510                              I(MAGINARY)
1511      0716  3151
1511      0717  4347
1511      0720  4040
1511      0721  4040
1511      0722  4015
1511      0723  5001
1511      0724  0716
1511      0725  1124
1511      0726  2504
1511                              M(AGNITUDE)
1512      0727  0551
1512      0730  4347
1512      0731  4040
1512      0732  4040
1512      0733  4023
1512      0734  5003
1512      0735  0114
1512      0736  0540
1512      0737  0601
1512      0740  0324
1512      0741  1722
1512                              S(CALE FACTOR)
1513      0742  5143
1513      0743  4740
1513      0744  4040
1513      0745  4040
1513      0746  1411
1513      0747  1605
1513      0750  4006
1513      0751  0505
1513      0752  0450
1513      0753  2205
1513      0754  2324
1513      0755  0122
1513      0756  2451
1513      0757  3400
1513                              LINE FEED(RESTART)\Z
1514                    /MESSAGES
1515                    MSG1,   TEXT Z
1516      0760  4347
1516
1517      0761  4043
1517      0762  0640
1517      0763  4040
1517      0764  4003
1517      0765  0116
1517      0766  1617
1517      0767  2440
1517      0770  0611
1517                    F     CANNOT FIND
1520      0771  1604
1520      0772  4347
1520
1521      0773  4043
1521      0774  4740
1521      0775  4040
1521      0776  4040
1521        77  4040
1521      1000  4040
1521      1001  4010
```

```
1521      1002  1124
1521      1003  4022
1521      1004  0524
1521      1005  2522
1521      1006  1640
1521      1007  2417
1521      1010  4003
1521      1011  1716
1521      1012  2434
1521                                HIT RETURN TO CONT\Z
1522                    MSG2,   TEXT Z
1523      1013  4347
1523
1524      1014  4043
1524      1015  0640
1524      1016  4040
1524      1017  4016
1524      1020  1740
1524      1021  2320
1524      1022  0103
1524                    F     NO SPACE
1525      1023  0543
1525
1526      1024  4740
1526      1025  4347
1526      1026  4040
1526      1027  4040
1526      1030  4040
1526      1031  4010
1526      1032  1124
1526      1033  4022
1526      1034  0524
1526      1035  2522
1526      1036  1640
1526      1037  2417
1526      1040  4003
1526      1041  1716
1526      1042  2434
1526                                HIT RETURN TO CONT\Z
1527                    /
1530      1043  0000  ANSWER, 0
1531                          *,+6
1532
1533                          EJECT
-
```

```
1535                    /SQUARE ROOT
1536    7052  0000  SQRT, 0
1537    7053  3273        DCA     ROOT    /CLR ROOT
1540    7054  1142        TAD     DPSQ    /IF SQ IS 0, EXIT
1541    7055  7640        SZA CLA
1542    7056  5262        JMP     NOT0
1543    7057  1143        TAD     DPSQ+1
1544    7060  7650        SNA CLA
1545    7061  5652        JMP I   SQRT
1546    7062  1143  NOT0, TAD     DPSQ+1  /1ST APPROX OF ROOT
1547    7063  7110        CLL RAR         /DIVIDE BY 2
1550    7064  1140        TAD     C2000   /+1/2
1551    7065  3273        DCA     ROOT
1552
1553    7066  1142  SQRT2, TAD     DPSQ    /SUM OF SQUARES
1554    7067  7421        MQL
1555    7070  1143        TAD     DPSQ+1
1556    7071  7100        CLL
1557    7072  7407        DVI             /DIVIDE BY ROOT APPROX
1560    7073  0000  ROOT, 0
1561    7074  7430        SZL
1562    7075  5314        JMP     SQRT1   /OVERFLOW
1563    7076  7200        CLA
1564    7077  7501        MQA             /QUOTIENT TO AC
1565    7100  1273        TAD     ROOT    /ADD APPROX TO QUOTIENT
1566    7101  7001        IAC
1567    7102  7110        CLL RAR         /DIVIDE BY 2
1570    7103  3035        DCA     SINE    /SAVE
1571    7104  1273        TAD     ROOT    /SUBTRACT OLD ROOT FROM NEW ONE
1572    7105  7041        CIA
1573    7106  1035        TAD     SINE
1574    7107  7650        SNA CLA
1575    7110  5314        JMP     SQRT1   /=0
1576    7111  1035        TAD     SINE
1577    7112  3273        DCA     ROOT    /NEW ROOT
1600    7113  5266        JMP     SQRT2
1601
1602    7114  1273  SQRT1, TAD     ROOT
1603    7115  5652        JMP I   SQRT
1604
1605
1606                    EJECT
```

```
1607                        PMODE
1610                    /MOVE PTS FROM ONE AREA TO ANOTHER
1611                    /10 = OLD BUFFER
1612                    /11 = NEW   "
1613                    /IF CMPFLG=1, COMPLEMENT VALUE
1614    7116  0000    MOVPTS, 0
1615    7117  6211            CDF1
1616    7120  1130    NXTPT,  TAD     CMPFLG
1617    7121  7113            CLL RAR         /TO LK
1620    7122  1410            TAD I   10
1621    7123  7430            SZL
1622    7124  7041            CIA
1623    7125  3411            DCA I   11
1624    7126  2034            ISZ     TEMPR
1625    7127  5320            JMP     NXTPT
1626    7130  6201            CDF0
1627    7131  5716            JMP I   MOVPTS
1630
1631                    /MOVE REAL OR MAGNITUDE VALUES
1632                    /FROM 0 TO 2000
1633                    /AND SWAP HALVES
1634                    /DO NOT COMPLEMENT
1635    7132  0000    MVRLMG, 0
1636    7133  7040            CMA             /OLD ADDR OF 1ST 1/2 = 0
1637    7134  3010            DCA     10
1640    7135  1037            TAD     GR      /NEW ADDR OF 1ST 1/2 = 2000 + 1/2 NO OF PTS
1641    7136  1110            TAD     C1777
1642    7137  3011            DCA     11
1643    7140  1033            TAD     ADD2
1644    7141  3034            DCA     TEMPR   /MOVE 1/2 NO OF PTS
1645    7142  3130            DCA     CMPFLG  /DONT COMPLEMENT
1646    7143  4527            JMS I   PMVPTS  /MOVE THEM
1647    7144  7040            CMA             /OLD ADDR OF 2ND 1/2 = 1/2 NO OF PTS
1650    7145  1037            TAD     GR
1651    7146  3010            DCA     10
1652    7147  1110            TAD     C1777   /NEW ADDR OF 2ND 1/2 = 2000
1653    7150  3011            DCA     11
1654    7151  1033            TAD     ADD2
1655    7152  3034            DCA     TEMPR   /1/2 NO OF PTS
1656    7153  4527            JMS I   PMVPTS  /MOVE THEM
1657    7154  5732            JMP I   MVRLMG
1660
1661                        EJECT
```

NO ERRORS

```
ACDF0  1202
ADDER   41
ADDR    35
ADDWOS 1157
ADD1   1174
ADD2   0033
ADJSGN 0570
ANSWER 7043
ARG2   1020
ASK    4720
ASKX   4740
ASK2   4523
ASK2X  4571
ASK3   4572
ASK3X  4626
ASR    7415
BIGSNU 0012
BOUND  1516
BUFHI  1574
BUFLO  1575
BUFPTR 1504
BUILD  0544
C      0032
CAM    7621
CCDF0  1401
CCIA   0162
CCLR   0146
CDF0   6201
CDF1   6211
CHKEND 4650
CHKHI  1473
CHKPT  0515
CKEND  4211
CMPFLG 0130
CNOP   0163
CNOTS  0677
CONT   1400
CONV   4627
CONVER 4674
CORVAL 0114
COSINE 0036
COUNT  0116
CSAM   1351
CURCNT 1573
CURDIS 1551
CURLOP 1563
CURRTN 1450
CURSAM 0101
C1000  0137
C1777  0110
C2000  0140
C6000  0101
DADD   1504
DATTAP 4061
DBLHI  1541
DBLLO  1547
DISFLG 4242
DISPER 6051
DISPLY 6001
DISPL1 6022
-
```

```
DOFFT  0046
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
LOADDR 61
LOOP1  04
LSR    741
MAXADR 1475
```

```
MAXFLD 1474
MAXNU    21
MINADR   16
MINFLD 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
```

```
PDPSCL 6054
PDSPER 6372
PFDV7  0123
PI     0026
PIFTFG 0133
PLEFTX 0125
PMRLMG 0126
PMVDIS 0124
PMVPTS 0127
PNSWPR 6374
PPREPR 6371
PR     0025
PREAD  0134
PRELFG 0132
PREPAR 6130
PRFRSH 0122
PROC   4357
PSHOWT 0121
PSQRT  6373
PTS    4064
PWRITE 0135
P401   1243
Q      0027
QAINIT 1000
QARFSH 1053
QI     0024
QR     0023
QUAD1  1110
QUAD2  1072
QUESNO 4734
QUES1  6411
QUES11 6625
QUES13 6665
QUES2  6453
QUES3  6475
QUES4  6521
QUES5  6571
QUES6  6612
RBUILD 0064
RDDATA 4320
RDORA  1343
REALFG 6053
RECHK  0066
REDPLY 6160
RELPTR 6324
REPL   4446
RESETC 0065
REVERS 0705
RFRSH  6171
ROOT   7073
ROTAT  4121
ROT1   4136
RTNCDF 1257
RWPARM 6405
S      0006
SAMNAM 4430
SCA    7441
SCAL   0053
SCALE  0340
SCL    74
SCLFAC 62
SCR4   0145
```

```
SETC    0542
SETDF   0541
SETFL     32
SGNADJ  0161
SHFCHK  0255
SHFLAG  2054
SHFT1   1077
SHFT2   1114
SHFT3   1125
SHIFCT  2562
SHIFT1  0056
SHIFT2  0057
SHIFT3  0060
SHL     7413
SHOSCL  6247
SHOWIT  6162
SIGN    1037
SINE    0035
SINLOC  0050
SINRET  1122
SINTAB  7347
SIZE    6170
SMALLR  6211
SORT    0042
SORTX   0701
SQRT    7052
SQRT1   7114
SQRT2   7066
STAMU   4144
STSCAL  4367
SWAPED  0745
TAB     1323
TEMP    1276
TEMPR   0034
TEMP1   4114
TEMP2   4675
TRIGET  1060
UNTFIL  4044
UPADDR  6166
UPLEGL  4645
VCOORD  1274
WCHDIS  6035
WINSAM  0100
WORD    0061
WORDP   0062
WRAP    1424
WSAM    1360
XCORD   0001
XCURHI  0112
XCURLO  0113
XITAB   2000
XLOCDF  0052
XRLOC   0051
XRTAB   0000
XSUM    1175
YCUR    0115
YSCAL   1444
YSHFT   0111
```

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

FFTD
DEC-12-AMFTA-A-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 _____

— — — — — — — — — — — — — — — — Fold Here — — — — — — — — — — — — — — — — — —

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

FIRST CLASS
PERMIT NO. 33
MAYNARD, MASS

BUSINESS REPLY MAIL
NO POSTAGE STAMP NECESSARY IF MAILED IN THE UNITED STATES

Postage will be paid by:

digital

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