ARGONNE NATIONAL LABORATORY
9700 South Cass Avenue
Argonne, Illinois 60439

Fast Fourier Transform Using the PDP-11

by

L. R. Windmiller, J. B. Ketterson, and J. C. Shaw

Solid State Science Division

June 1972

NOTICE

This report was prepared as an account of work sponsored by the United States Government. Neither the United States nor the United States Atomic Energy Commission, nor any of their employees, nor any of their contractors, subcontractors, or their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness or usefulness of any information, apparatus, product or process disclosed, or represents that its use would not infringe privately owned rights.

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency Thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

DISCLAIMER

Portions of this document may be illegible in electronic image products. Images are produced from the best available original document.

TABLE OF CONTENTS

	<u>Page</u>
Abstract	4
Introduction	5
Relationship between the Finite Fourier Series and the Fourier Integral Transform.	6
Description of Algorithm	11
Operating Instructions	18
Programs	19
Program Listings	21
Acknowledgments	72
References	73

ABSTRACT

This report describes programs to perform FFT on a real data sequence using a PDP-11 without EAE. The sequence to be transformed can be stored either internally in the memory of the PDP-11, externally in the memory of a Nicolet (Fabri-Tek) Model 1070 signal averager interfaced with the PDP-11, or in some other storage device interfaced with the PDP-11. The calculation is performed in place, using the storage location of the original input data for intermediate results and eventually replacing this data with the Fourier transform. All points in the sequence are considered real and the even odd point method is used in order to avoid the necessity of doubling the data storage area to include the zero imaginary terms. These programs use fixed point arithmetic and can perform 1K FFT in a time typically of order 3 seconds for single precision and 8 seconds for double precision.

Material available includes:

1. Detailed explanation of the algorithm used.
2. Operating instructions.
3. Source or binary tapes.
4. Listings.
5. Executive program for teletype control.
6. Various data handling programs including apodization, magnitude calculation, etc.

Minimum Hardware: 4K PDP-11

Storage Requirements: Program - 900 words
Sine lookup table - dependent
on maximum size transform
required.
Of order 256 words for 1K or
less transform.
Data - Number of elements
transformed.

Source Language: PAL-11A

Fast Fourier Transform Using the PDP-11

by

L. R. Windmiller, J. B. Ketterson, and J. C. Shaw

INTRODUCTION

The computational efficiency introduced by the Fast Fourier Transform technique¹ has had a major effect on many areas of research, broadening both the depth and scope of many investigations while opening up other new areas of endeavor. One of the revolutionary new applications using this technique was the use of a minicomputer for both data acquisition and analysis on line within an experimenter's own laboratory. Under such circumstances, the speed at which the minicomputer could perform the Fast Fourier Transform often became the limiting parameter in the design and performance of experiments. With the advent of the Digital Equipment Corporations' PDP-11 minicomputer,² it became evident that appreciable reductions in computational time could be realized if programs were written which would take advantage of the unique features of this machine. Such programs have now been written and are the subject of this report. These programs use fixed point arithmetic and can perform 1 K Fast Fourier Transforms in a time typically of order 3 seconds for single precision and 8 seconds for double precision using a computer without hardware multiply/divide. Core storage requirements for the program are modest, the entire program including the routines for calculating the absolute magnitude of the Fourier coefficients reside in about 900 words. In order to achieve high computing speed, the program uses a cosine lookup table. This table adds to the length of the main program by an amount dependent upon the maximum size transform the program is capable of handling. Thus, for a program capable of handling up to $M = 2^m$ point transforms, the cosine lookup table will contain $M/4$ entries for single precision calculations and $M/2$ entries for double precision.

The calculation performs the Fast Fourier Transform in place, using the storage locations of the original input data for intermediate storage and eventually replacing this data with the Fourier transform. The data can be stored either internally in the memory of the PDP-11, externally in the memory of a Nicolet Series 1070 signal averager³ interfaced with the PDP-11, or in some other storage device interfaced with the PDP-11.

Relationship between the Finite Fourier Transform and the Fourier Integral Transform^{4,5}

The typical experimental situation in which Fast Fourier Transform techniques are applied approximates the wave form to be transformed by a series of samples of the wave form. (That is, the wave form is sampled at some rate Δt for a finite time T and the $m = T/\Delta t$ samples form a finite discrete series.) It is important to understand the effects that the sampling and the finite observation time have on the transform derived from the fast Fourier transform and how this transform relates to the actual Fourier transform of the complete wave form.

In order to do this, consider the infinite continuous wave form $X(t)$. The Fourier transform is given by

$$a(f) = \int_{-\infty}^{+\infty} X(t) e^{-2\pi i f t} dt \quad (1)$$

The inverse transform is

$$X(t) = \int_{-\infty}^{+\infty} a(f) e^{2\pi i f t} df \quad (2)$$

When $X(t)$ is sampled at intervals Δt , Eq. (2) is written at the sampling points

$$\begin{aligned} X(j \Delta t) &= \int_{-\infty}^{+\infty} a(f) e^{2\pi i f j \Delta t} df \\ &= \sum_{k=-\infty}^{+\infty} \int_{kF}^{(k+1)F} a(f) e^{2\pi i j f \Delta t} df \\ &= \int_0^F a_p(f) e^{2\pi i j f \Delta t} df \end{aligned} \quad (3)$$

where $F = 1/\Delta t$ and

$$a_p(f) = \sum_{k=-\infty}^{+\infty} a(f + kF) \quad (4)$$

is a continuous function that is periodic with period F . This function is formed by the superposition of the non-periodic function $a(f)$ shifted by all multiples of the fundamental period F . The function $a_p(f)$ is said to be

the "aliased" version of the function $a(f)$ with the aliasing occurring relative to the Nyquist frequency $F/2$.

Since $a_p(f)$ is a periodic function of f , it has a Fourier series expansion given by

$$a_p(f) = \frac{1}{F} \sum_{j=-\infty}^{+\infty} X(j \cdot \Delta t) e^{-2\pi i f j / F} \quad (5)$$

Thus, the effect of sampling a continuous function is to produce a periodic continuous Fourier series transform which is the aliased version of the continuous Fourier integral transform.

Eq. (5) still represents an infinite Fourier series. The effect of sampling for only a finite time so that there are only N samples in the series $X(j \cdot \Delta t)$ can be seen by writing Eq. (5) as

$$\begin{aligned} a_p(f) &= \frac{1}{F} \sum_{j=0}^{N-1} \sum_{l=-\infty}^{+\infty} X(j \cdot \Delta t + lN\Delta t) e^{2\pi i f (j+lN)/F} \\ &= \frac{1}{F} \sum_{j=0}^{N-1} X_p(j \cdot \Delta t) e^{2\pi i f j / F} \end{aligned} \quad (6)$$

where

$$X_p(j \cdot \Delta t) = \sum_{l=-\infty}^{+\infty} X(j \cdot \Delta t + lN\Delta t) e^{2\pi i f lN/F} \quad (7)$$

In order that $X_p(j \cdot \Delta t) = X(j \cdot \Delta t)$ on the interval $0 \leq j < N-1$, we must have $X(j \cdot \Delta t) = 0$ outside the interval. Thus, the transform (6) is that of a function whose value is zero everywhere but in the interval $0 \leq T < (N-1)\Delta t$ or else it is the transform of a function whose superposition as given by Eq. (7) yields the sampled series.

Before examining the effect of this truncation of the sampled series on the transform, it is worthwhile to point out that Eqs. (6) and (7) simplify if the Fourier series transform $a_p(f)$ is also sampled at intervals $\Delta f = F/N = 1/N\Delta t$.

Then

$$\begin{aligned}
 a_p(n\Delta f) &= \frac{1}{F} \sum_{j=0}^{N-1} \sum_{\ell=-\infty}^{+\infty} X(j\Delta t + \ell N\Delta t) e^{2\pi i n(j+\ell N)/N} \\
 &= \frac{1}{F} \sum_{j=0}^{N-1} X_p(j\Delta t) e^{2\pi i n j/N} \quad (6')
 \end{aligned}$$

where

$$X_p(j\Delta t) = \sum_{\ell=-\infty}^{+\infty} X(j\Delta t + \ell N\Delta t) \quad (7')$$

In this case $X_p(j\Delta t)$ is the aliased version of the function $X(j\Delta t)$ with aliasing occurring relative to the number of samples N . Here we can relax the condition that the function be zero everywhere outside the interval so long as it is periodic. Thus we have derived the important result that sampling for a finite number of points produces a transform of a function "aliased" as in Eq. (7) or Eq. (7').

Thus, if $X(t)$, $-\infty < t < \infty$ and $a(f)$, $-\infty < f < \infty$ are a Fourier integral transform pair, then

$$T X_p(j\Delta t), \quad j = 0, 1, \dots, N-1$$

and

$$a_p(n\Delta f) \quad n = 0, 1, \dots, N-1$$

are a finite Fourier series pair.

Because of this, if we take the finite Fourier series transform of a function $X(j\Delta t)$, the equivalent Fourier integral function is

$$X'(t) = X(t) \cdot A(t)$$

where $A(t) \neq 0$ $0 \leq t < (N-1)\Delta t$
 $= 0$ elsewhere.

Thus

$$\begin{aligned}
 a(f) &= \int_{-\infty}^{+\infty} X(t)A(t)e^{-2\pi ift} dt \\
 &= \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} [\bar{X}(f')e^{2\pi if't} df'] [\bar{A}(f'')e^{2\pi if''t} df''] \\
 &\quad e^{-2\pi ift} dt \\
 &= \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} \bar{X}(f')\bar{A}(f'')e^{2\pi i(f'+f''-f)t} dt df' df'' \\
 &= \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} \bar{X}(f')\bar{A}(f'')\delta(f' + f'' - f) df' df'' \\
 &= \int_{-\infty}^{+\infty} \bar{X}(f')\bar{A}(f - f') df' \tag{8}
 \end{aligned}$$

$$\text{where } \bar{X}(f) = \int_{-\infty}^{+\infty} X(t)e^{-2\pi ift} dt$$

$$\bar{A}(f) = \int_{-\infty}^{+\infty} A(t)e^{-2\pi ift} dt \tag{9}$$

The effect of aliasing on a time sequence is to convolve its transform with the transform of the cutoff function.

This convolution is the source of much of the problems with the fast Fourier transform technique. The transform of a rectangular data window ($A(t) = 1, 0 \leq t < (N - 1)\Delta t$) is the $\sin f/f$ function. This function produces a spread in the frequency spectra of the transform and worse produces extra peaks due to the side lobes of the $\sin f/f$ function. In the special case that the function to be transformed is exactly periodic over the sampling interval, the central peak of the $\sin f/f$ occurs at the frequency sample and all other frequency samples occur at the zeros of this function. Thus the convolution does not affect the sampled frequency spectrum. However, as soon as the function is not exactly periodic, the frequency samples no longer occur at the zeros of the $\sin f/f$ function and extra peaks in the frequency spectrum appear.

In order to minimize the distortion of the frequency spectrum due to this cutoff function, a function other than a rectangular data window can be used.⁶ This procedure, called apodization, consists of multiplying the sampled series with a function whose transform is sharply peaked and which does not have the undesirable side lobes of the $\sin f/f$ function.

Examples of such apodization functions are triangular weighting $A(t) = 1 - |t - T/2|$ and Gaussian weighting

$$A(t) = e^{-k(t-T/2)^2}.$$

Description of the Algorithm⁷

The Fast Fourier Transform (FFT) is a method for efficiently computing the discrete Fourier transform of a discrete function. Thus for the discrete series $X(t)$ ($t = \text{integer}, 0 \leq t \leq M - 1$), the discrete Fourier transform is

$$A(f) = \sum_{t=0}^{M-1} X(t) \exp(2\pi i ft/M) \quad (10)$$

($f = \text{integer}, 0 \leq f \leq M - 1$).

For computing efficiency, we must have the number of terms in the series equal to a power of two; that is, $M = 2^m$ where m is an integer. In the following, the binary representation for the integer t and f will be employed. Thus

$$t = t_{m-1} 2^{m-1} + t_{m-2} 2^{m-2} + \dots + t_0 \quad t_0 = (t_{m-1}, t_{m-2}, \dots, t_0) \quad (11)$$

$$f = f_{m-1} 2^{m-1} + f_{m-2} 2^{m-2} + \dots + f_0 \quad f_0 = (f_{m-1}, f_{m-2}, \dots, f_0) \quad (12)$$

where the t_i (or f_i), which have the values either 0 or 1, represent the value of the i^{th} bit in the binary representation of the integer t (or f).

The particular form of the FFT used in this calculation is called decimation in frequency. This form was chosen because of its convenient indexing features. These features will be pointed out in the following.

Decimation in frequency corresponds to writing the argument of the complex exponential of Eq. (10) in the following way:

$$\begin{aligned} ft/m &= (f_{m-1}, f_{m-2}, \dots, f_0)(t_{m-1}, t_{m-2}, \dots, t_0)/2^m \\ &= f_{m-1} \cdot \frac{t}{2} + f_{m-2} \cdot \frac{t}{4} + \dots + f_0 \cdot \frac{t}{2^m} \end{aligned} \quad (13)$$

Because of the property

$$\exp(2\pi i[\text{integer} + \text{fraction}]) = \exp(2\pi i[\text{fraction}])$$

we can drop all the integer portions of the fraction $t/2^n$ in Eq. (13) so that Eq. (10) can be rewritten as

$$\begin{aligned}
 A(f) &= \sum_{t=0}^{m-1} X(t) \left[\exp \left(2\pi i \cdot \frac{f_{m-1} t_0}{2} \right) \times \exp \left(2\pi i \cdot \frac{f_{m-2} (2t_1 + t_0)}{4} \right) \right. \\
 &\quad \dots \times \exp \left(2\pi i \cdot \frac{f_{m-n} (2^{n-1} t_{n-1} + 2^{n-2} t_{n-2} + \dots + t_0)}{2^n} \right) \\
 &\quad \left. \dots \times \exp \left(2\pi i \cdot \frac{f_0 t}{2^m} \right) \right] \quad (14) \\
 &= \sum_{t_{m-1}=0}^1 \sum_{t_{m-2}=0}^1 \dots \sum_{t_0=0}^1 \left[\exp \left(2\pi i \cdot \frac{f_{m-1} t_0}{2} \right) \right. \\
 &\quad \left. \times \exp \left(2\pi i \cdot \frac{f_{m-2} (2t_1 + t_0)}{4} \right) \times \dots \times \exp \left(2\pi i \cdot \frac{f_0 t}{2^m} \right) \right]
 \end{aligned}$$

The first pass takes all the elements in pairs and performs the calculation

$$X'(f_0, t_{n-2}, \dots, t_0) = \sum_{t_{m-1}=0}^1 \exp \left(2\pi i \cdot \frac{f_0 t}{2^m} \right) X(t)$$

Each pair is taken from its place and the calculation performed creating a new pair which is returned to the place just vacated. The next pass sums on the t_{n-2} bit and each subsequent pass sums on the next bit to the right until the entire sum over t is performed. A typical pass performs the calculation

$$\begin{aligned}
 X''(f_0, \dots, f_{m-n}, t_{n-2}, \dots, t_0) \\
 &= \sum_{t_{n-1}} X'(f_0, \dots, f_{m-n-1}, t_{n-1}, \dots, t_0) \\
 &\quad \times \exp \left(2\pi i \cdot \frac{f_{m-n} (2^{n-1} t_{n-1} + \dots + t_0)}{2^n} \right) \quad (15)
 \end{aligned}$$

This calculation operates on the pairs of elements (where only the underlined argument in Eq. (15) is indicated, the other arguments, of course, determining the pairs and taking on all possible values) in the following way:

$$\begin{aligned}
 \text{Re } X''(\text{---}, 0, \text{---}) &= \text{Re } X'(\text{---}, 0, \text{---}) + \text{Re } X'(\text{---}, 1, \text{---}) \\
 \text{Imag } X''(\text{---}, 0, \text{---}) &= \text{Imag } X'(\text{---}, 0, \text{---}) + \text{Imag } X'(\text{---}, 1, \text{---}) \\
 \text{Re } X''(\text{---}, 1, \text{---}) &= \cos \theta \text{ Re } C - \sin \theta \text{ mag } C \\
 \text{Imag } X''(\text{---}, 1, \text{---}) &= \cos \theta \text{ Imag } C + \sin \theta \text{ Re } C
 \end{aligned} \tag{16}$$

where

$$\begin{aligned}
 C &= X'(\text{---}, 0, \text{---}) - X'(\text{---}, 1, \text{---}) \\
 \theta &= 2\pi \cdot \left(\frac{2^{n-2} t_{n-2} + \dots + t_0}{2^n} \right)
 \end{aligned} \tag{17}$$

The advantages of this procedure are now clear since it can be seen that the same bit arrangement used for the address calculation of the elements being operated upon also directly determine the phase θ of the weighting factor.

This calculation does have the disadvantage that the final result has the Fourier components stored in memory in bit reversed order, that is the elements are stored in memory as

$$X''(f_0, f_1, \dots, f_{m-1})$$

Thus one run through the result is necessary interchanging so that

$$A(f_{m-1}, \dots, f_0) = X''(f_0, \dots, f_{m-1})$$

The subroutine used to perform this rearrangement is called BITINV.

An additional calculation is necessary because of the form of the input data which is, of course, all real. If all the imaginary components of the input series were assumed zero, considerable efficiency in core storage and calculational speed would be lost. In order to avoid this difficulty, the even odd point method is used and only the Fourier components below the Nyquist frequency

are calculated. The even odd point method considers the j^{th} and $j + 1$ data elements as the real and imaginary parts respectively of the $j/2$ element of the series to be transformed. That is, the real series is regrouped so that we actually take the transform of the modified series

$$A'(f) = \sum_{t=0}^{M/2-1} [X(2t) + iX(2t + 1)] e^{i\left(\frac{tf}{M/2}\right)} \quad (18)$$

However, this can be rewritten as

$$A'(f) = \sum_{t=0}^{N-1} [X(t) + iY(t)] e^{i\left(\frac{tf}{N}\right)} \quad (18')$$

Using

$$A'(N - f) = \sum_{t=0}^{N-1} [X(t) + iY(t)] e^{i\left(-\frac{tf}{N}\right)} \quad (18'')$$

it can be shown that

$$\begin{aligned} 2\hat{X}(f) &= \text{Re } C + i \text{Imag } D \\ 2\hat{Y}(f) &= \text{Imag } C - i \text{Re } D \end{aligned} \quad (19)$$

where $\hat{X}(f)$ and $\hat{Y}(f)$ are the transform of $X(t)$ and $Y(t)$ respectively, and

$$\begin{aligned} C &= A'(f) + A'(N - f) \\ D &= A'(f) - A'(N - f) \end{aligned} \quad (19')$$

However, the Fourier transform actually desired is

$$\begin{aligned}
A(f) &= \sum_{t=0}^{M-1} X(t) e^{j \left(\frac{t f}{M} \right)} \\
&= \sum_{t=0}^{N-1} X(2t) e^{j \left(\frac{t f}{N} \right)} + \sum_{t=0}^{N-1} X(2t+1) e^{j \left(\frac{t f}{N} \right)} e^{j \left(\frac{f}{M} \right)} \\
&= \hat{X}(f) + e^{j \left(\frac{f}{M} \right)} \hat{Y}(f)
\end{aligned} \tag{20}$$

So after the Fourier transform is computed, it is necessary to make an additional pass through the result modifying the pairs at f and $N - f$. From above it can be shown that the calculation necessary is

$$\begin{aligned}
\text{Re } A(f) &= \frac{1}{2} \left[\text{Re } C + \cos \left(\frac{f}{M} \right) \text{Imag } C + \sin \left(\frac{f}{M} \right) \text{Re } D \right] \\
\text{Imag } A(f) &= \frac{1}{2} \left[\text{Imag } D + \sin \left(\frac{f}{M} \right) \text{Imag } C - \cos \left(\frac{f}{M} \right) \text{Re } D \right]
\end{aligned} \tag{21}$$

and

$$\begin{aligned}
\text{Re } A(N - f) &= \frac{1}{2} \left[\text{Re } C - \cos \left(\frac{f}{M} \right) \text{Imag } C - \sin \left(\frac{f}{M} \right) \text{Re } D \right] \\
\text{Imag } A(N - f) &= \frac{1}{2} \left[-\text{Imag } D + \sin \left(\frac{f}{M} \right) \text{Imag } C - \cos \left(\frac{f}{M} \right) \text{Re } D \right]
\end{aligned} \tag{21'}$$

The subroutine used to perform this calculation is called COMPLEX.

After performing the calculations, the transformed elements are stored in the data area in sequential order with the real and imaginary parts alternating. For many purposes (such as convenient data display), it is desirable to sort the data so that the real elements occupy the first half of the data area in sequential order followed by the corresponding ordering of the imaginary elements in the second half of the data area. Such sorting can be done with a minimum of interchanging by recognizing that the series actually is composed of two interpenetrating series; the series of the real elements and the series of the imaginary elements. Thus, writing the series as

$$R_0, I_0, R_1, I_1, R_2, I_2, R_3, I_3, R_4, I_4, \dots$$

it is evident that each element of the real series is twice as far along as it should be; that is, R_1 in position 2 (position numbers start at 0 and go to $M-1$) should be at position 1, R_2 at position 4 should be at position 2, etc. Less obvious is the fact that the imaginary series also has a similar relationship only instead of with respect to the beginning of the series each element is twice as far as it should be from the end of the series. Thus, sequences of interchanges can be set up based solely on the position of the elements in the original series. The first sequence is then

$$1 \rightarrow 2 \rightarrow 4 \rightarrow 8 \rightarrow \dots \rightarrow M/2 \rightarrow 1$$

The recipe of doubling each element to find the next element terminates when the next doubling points to a number larger than the total number of elements. When this occurs, what has actually happened is that the sequence has switched from the real series to the imaginary series. Then, further development of the series is found by doubling the offset from the end of the original series rather than from the beginning. Again, when this doubling leads to an element further from the end than the number of elements in the series, a switch has occurred back to the real series again and the original recipe can be followed. In this way, a sequence of changes can be generated which ends when the first element of the sequence reappears. Since the original series is symmetric with respect to the two interpenetrating series, this sequence can also be used interpreting it as applying to positions from the end rather than the beginning of the series (the case where these two sequences happen to be degenerate in that they operate on the same series elements must be distinguished so that the elements are not shifted twice). This method is applied by generating sequences starting first at element 1, then 3, 5, etc. Sequences can never start on even number elements because of the rule that subsequent members of the sequences are generated by doubling the previous element. Each sequence after the first can be checked to see that it has not been done before by examining its lowest and highest element. The sequence is new if its lowest element is higher than any of the previous sequences lowest elements and its highest element is lower than any of the previous sequences highest elements. Since the first and last elements are in their proper place initially, only $M-2$ changes are necessary. However, for the case when $M/3 = \text{integer}$, one

of the changes is the interchange of elements $M/3$ and $2M/3$. Thus, in this case the above prescription need be followed for only $M-4$ changes. The subroutine used to perform this operation is called SORT.

After performing the entire calculation including subroutine SORT, the transformed elements are stored in the data area in sequential order with the real parts in the first half followed by the corresponding imaginary parts in the second half. It is sometimes convenient to have the absolute magnitude of the Fourier transform which entails making the calculation

$$|A(f)| = [(\text{Re } A(f))^2 + (\text{Imag } A(f))^2]^{1/2}$$

for all f . Since no division routine is necessary for the FFT calculation, a square root algorithm was chosen which required no divisions. This algorithm uses the iterative sequence

$$U_0 = x$$

$$V_0 = x - 1$$

$$U_{k+1} = (U_k (1 - \frac{1}{2} V_k)) \quad k = 0, 1, 2, 3 \dots$$

$$U_{k+1} = \frac{1}{4} V_k^2 (V_k - 3)$$

where U_k converges to $x^{1/2}$. For $1/2 \leq x \leq 1$, this requires 3 iterations to converge for single precision and 4 iterations for double precision. The subroutine used to calculate square roots is called SQR00T.

In order to maintain maximum accuracy in the fixed point arithmetic, each pass of the FFT calculation is preceded by a normalization procedure. This procedure looks for the largest data element (in absolute magnitude) and then shifts all the data, either left or right so that $1/8 < |\text{largest element}| \leq 1/4$. The number of shifts performed on the data is maintained at location SCALER. The subroutine used to perform this operation is called either SIZER or NORM depending on the calculational configuration as outlined in the next section.

It should be noted that although the arithmetic is done in fixed point, the operands are always viewed as fractions with the most significant bit used as a sign bit and the following bits as powers of $1/2$. Thus 0 111 000 000 000 is $+1/2 + 1/4 + 1/8 = 7/8$.

Operating Instructions

Two basic forms of the program are presented here. These programs illustrate both single and double precision fixed point arithmetic with data storage either in the PDP-11 memory or in the memory of a Nicolet Instrument Model 1070 signal averager interfaced with the PDP-11. The Model 1070 is an 18 bit machine in which the most significant 16 bits are accessed by the PDP-11 in the same way as any internal memory word. The least significant 2 bits are accessed rather differently, however, resulting in differences accessing the data between the double precision routine using internal storage and that using storage in the 1070.

The difference between the scaling routines NORM and SIZER is that SIZER performs double precision rotates using the addressing necessary to access the least 2 significant bits in the 1070 while NORM accesses all memory locations in the standard manner. Thus, for double precision arithmetic using the 1070 storage SIZER must be used, while for internal storage NORM must be used. However, for single precision arithmetic SIZER need be used only for the first scaling operating when 1070 storage is used and after that NORM can be used independent of the data storage location.

Since double precision arithmetic using 1070 storage really requires only 18 bits of significance, full double precision multiplication routines and cosine lookup tables are not needed. The integer multiplication routine is therefore modified to operate on 18 bits rather than 32. Seventeen bits of significance are stored in a single word of the double precision cosine lookup table by multiplying each cosine by 2 and using the sign bit (which is always zero).

The main program is completely relocatable and can be loaded anywhere in core.⁸ The input needed by the program is the address of the first word of the data storage area which must be put in location LOCATR and the numbers of data words to be transformed (a power of 2) which must be put at location SIZE. The cosine lookup table begins at location SCTABL and extends for $M/4 + 1$ words for single precision arithmetic with data storage in the 1070 and for $M/2 + 2$ words for double precision arithmetic with data storage in the PDP-11.

Operation of the program is simple. The FFT program starts at location START and can be invoked by a calling program with the JSR PC, START instruction.

None of the general registers are saved and return to the calling program finds the real part of the transform in the first half of the data area and the imaginary part in the second half. SCALER then contains the factor by which the data has been scaled by the transformation. The absolute magnitude calculation starts at location MAGNTD and also can be entered from a calling program by the JSR PC, MAGNTD instruction. Return to the calling program finds the magnitude replacing the real part of the transform and the imaginary part unchanged. No general registers are saved.

Programs

Two examples of the programs are given here. The first program is a general, single precision computation with data pointed to be location LOCATR. This program can be used to operate on data in the Model 1070 by first using SIZER (SIZER can be found in the second program example) in order not to lose significant bits during the first normalization. After this, the main program using NORM can be used to perform the FFT.

The second program is a double precision routine specialized to operate on 18 bit data in the Model 1070. All the actual computations, however, are performed in a double precision format so that conversion to full 32 bit data is straightforward. All that is necessary is to change the accessing of the data from that peculiar to the Model 1070 to that necessary for full double precision format. In addition the subroutine MULT must be changed to a full 32 bit routine (making sure that it reflects the fact that the numbers are interpreted as fractions so that $40000g \times 40000g = 20000g$). An extra iteration in the SQRT routine is necessary. It will also be necessary to access the cosine lookup table in full double precision.

The programs reproduced here can transform a maximum of 1024 points. If larger transforms are desired, the necessary modifications are easy. In addition to the construction of a cosine lookup table having the appropriate number of entries, it is necessary to change the access to the sine in subroutines COMPLX and LOOKUP and the setting up of WGTINC at locations COMP0 and START1.

The double precision program also contains an executive routine in order to control all functions from the teletype keyboard. Thus, if the program is started at location INIT, the program will respond with a CR/LF

on the teletype and wait for an operator response indicating what function to perform. By typing any of the following letters, the functions indicated will be performed:

- T - place Model 1070 under computer control, compute FFT, remove Model 1070 from computer control and wait for another operator command.
- M - place Model 1070 under computer control, compute absolute magnitude, remove Model 1070 from computer control and wait for another operator command.
- O - place Model 1070 under computer control, copy the block of data starting at the address at location SOURCE and extending a number of words equal to the contents of location SIZE into a block starting at the address at location DEST, remove Model 1070 from computer control and wait for another operator command.
- I - place Model 1070 under computer control, perform the opposite function of O; that is, copy data block from DEST to SOURCE, remove Model 1070 from computer control and wait for another operator command.
- X - place Model 1070 under computer control, interchange the data blocks located at SOURCE and DEST, remove Model 1070 from computer control and wait for another operator command.
- S - place Model 1070 under computer control, perform subroutine SIZER on data, remove Model 1070 from computer control and wait for another operator command.

In addition, two more operations can be added to the executive by loading the teletype code for the call character at locations 4034 and 4036 (relative to INIT) and the start address of the subroutines (relative to INIT) at locations 4054 and 4056 (relative to INIT). Because the entire program is relocatable, the executive program uses this information to compute the branch vector in the JSR PC, (subroutine) instructions at locations 4316 and 4320 (relative to INIT). This executive can also be used with the single precision program when used with the Model 1070 or, if the Model 1070 control statements are deleted, with the general program.

Program Listings

```

;
; FAST FOURIER TRANSFORM MODULE.
; PROGRAMMED BY LEE MINOMILLER,
; ARGONNE NATIONAL LABORATORY,
; ARGONNE, ILLINOIS 60439.
;
;
;-----
;-----
;-----SUBROUTINE START-----
;-----
;

```

```

;
; CALLING SEQUENCE:
; JSR PC,START FOR TRANSFORM
; JSR PC,MAGNTO FOR MAGNITUDE
;

```

BITINV	001750	BITIN0	001754	BITIN1	001766	BITIN2	002006
BITIN3	002036	BITIN4	002056	COMPLX	000346	COMPL1	000420
COMPL2	000450	COMP0	000356	COMP0A	000374	COMP2A	000542
COMP2B	000554	COSW	000014	COUNT	001150	FFT	000110
FFT1	000114	FFT2	000314	GETADR	001606	GETADD	001624
GETAD1	001650	HIADD	001146	IMAGA	000340	IMAGB	000344
INDEX	000806	LOAD0	001144	LOCATR	000012	LOOKUP	001670
LOOK1	001716	LOOK2	001746	MAGNTO	002156	MAG1	002162
MAG2	002250	MULT	002060	MULT1	002074	MULT2	002106
MULT3	002114	MULT4	002124	MULT5	002144	MULT6	002146
NORM	000742	NORM1	000764	NORM2	000772	NORM3	001002
NORM4	001020	NORM6	001040	NORM7	001050	NORM7A	001102
NORM7B	001104	NORM7C	001056	NORM8	001122	NORM8A	001124
NORM9	001142	PASS	000002	PC	=X000007	REALA	000336
REALB	000342	R0	=X000000	R1	=X000001	R2	=X000002
R3	=X000003	R4	=X000004	R5	=X000005	SCALER	000010
SCTABL	002474	SINW	000016	SIZE	000000	SORT	001152
SORT0	001172	SORT1	001224	SORT1A	001246	SORT2	001262
SORT2A	001306	SORT2B	001304	SORT3	001314	SORT3A	001346
SORT3B	001344	SORT4	001360	SORT4A	001432	SORT4B	001414
SORT5	001460	SORT5A	001514	SORT6	001524	SORT6A	001522
SORT7	001546	SORT8	001604	SP	=X000006	SQROOT	002252
SQROT1	002266	SQROT2	002332	SQROT3	002344	SQROT4	002346
SQROT5	002352	SQRT	002354	SQRT1	002372	SQRT2	002424
START	000024	START1	000072	SWCTR	000020	WEIGHT	000004
WGTINC	000022	.	= 003476				

```

000000      R0=X0
000001      R1=X1
000002      R2=X2
000003      R3=X3
000004      R4=X4
000005      R5=X5
000006      SP=X6
000007      PC=X7
000008      .#0
000000 000000 SIZE: .WORD 0
000002 000000 PASS: .WORD 0
000004 000000 WEIGHT: .WORD 0
000006 000000 INDEX: .WORD 0
000010 000000 SCALER: .WORD 0
000012 000000 LOCATR: .WORD 0
000014 000000 COSHI: .WORD 0
000016 000000 SINHI: .WORD 0
000020 000000 SWCNTR: .WORD 0
000022 000000 WGTINC: .WORD 0
000024 005067 START: CLR SCALER
177760
000030 005067      CLR INDEX
177752
000034 016700      MOV SIZE,R0
177740
000040 006200      ASR R0
000042 010067      MOV R0,PASS
177734
000046 006200      ASR R0
000050 010067      MOV R0,SWCNTR
177744
000054 005067      CLR WEIGHT
177724
000060 012767      MOV #1,WGTINC
000001
177734
000066 016700      MOV SIZE,R0
177706
000072 022700 START1: CMP #2000,R0
002000
000076 001404      BEQ FFT
000100 006300      ASL R0
000102 006367      ASL WGTINC
177714
000106 000771      BR START1
000110 004767 FFT:   JSR PC,NORM
000626
000114 004767 FFT1:  JSR PC,GETADR
001466
000120 011402      MOV @R4,R2

```

000122 061514	ADD @R5,@R4
000124 160215	SUB R2,@R5
000126 005724	TST (R4)+
000130 005725	TST (R5)+
000132 011402	MOV @R4,R2
000134 061514	ADD @R5,@R4
000136 160215	SUB R2,@R5
000140 004767	JSR PC,LOOKUP
001524	
000144 010400	MOV R4,R0
000146 011502	MOV @R5,R2
000150 004767	JSR PC,MULT
001704	
000154 010146	MOV R1,-(SP)
000156 010302	MOV R3,R2
000160 014500	MOV -(R5),R0
000162 004767	JSR PC,MULT
001672	
000166 162601	SUB (SP)+,R1
000170 005401	NEG R1
000172 011502	MOV @R5,R2
000174 010125	MOV R1,(R5)+
000176 010400	MOV R4,R0
000200 004767	JSR PC,MULT
001654	
000204 010104	MOV R1,R4
000206 010300	MOV R3,R0
000210 011502	MOV @R5,R2
000212 004767	JSR PC,MULT
001642	
000216 060401	ADD R4,R1
000220 005401	NEG R1
000222 010115	MOV R1,@R5
000224 062767	ADD #2,INDEX
000002	
177554	
000232 066767	ADD WGTINC,WEIGHT
177564	
177544	
000240 005367	DEC SWCNT
177554	
000244 001323	BNE FFT1
000246 006367	ASL WGTINC
177550	
000252 006267	ASR PASS
177524	
000256 022767	CMP #1,PASS
000001	
177516	
000264 001413	BEO FFT2


```

000266 005067      CLR WEIGHT
      177512
000272 005067      CLR INDEX
      177510
000276 016700      MOV SIZE,R0
      177476
000302 006200      ASR R0
000304 006200      ASR R0
000306 010067      MOV R0,SHCNTR
      177506
000312 000676      BR FFT
000314 004767 FFT2: JSR PC,NORM
      000422
000320 004767      JSR PC,BITINV
      001424
000324 004767      JSR PC,COMPLX
      000016
000330 004767      JSR PC, SORT
      000616
000334 000207      RTS PC
      ;-----
      ;-----
      ;-----SUBROUTINE COMPLX-----
      ;-----
      ;-----
000336 000000 REALA: .WORD 0
000340 000000 IMAGA: .WORD 0
000342 000000 REALB: .WORD 0
000344 000000 IMAGB: .WORD 0
000346 005067 COMPLEX: CLR WGTINC
      177450
000352 016700      MOV SIZE,R0
      177422
000356 022700 COMP0: CMP #2000,R0
      002000
000362 001404      BEO COMP0A
000364 006300      ASL R0
000366 005267      INC WGTINC
      177430
000372 000771      BR COMP0
000374 016705 COMP0A: MOV LOCATR,R5
      177412
000400 010504      MOV R5,R4
000402 005724      TST (R4)+
000404 061415      ADD @R4,@R5
000406 005014      CLR @R4
000410 012703      MOV #4,R3
      000004
000414 010367      MOV R3,INDEX
      177366
000420 010304 COMPL1: MOV R3,R4
000422 066704      ADD LOCATR,R4
      177364
000426 016705      MOV SIZE,R5
      177346

```

```

000432 006305      ASL R5
000434 160305      SUB R3,R5
000436 066705      ADD LOCATR,R5
177350
000442 020405      CMP R4,R5
000444 001001      BNE COMPL2
000446 000207      RTS PC
000450 012400      COMPL2:MOV (R4)+,R0
000452 012501      MOV (R5)+,R1
000454 060100      ADD R1,R0
000456 010067      MOV R0,REALA
177654
000462 006267      ASR REALA
177650
000466 160100      SUB R1,R0
000470 160100      SUB R1,R0
000472 006200      ASR R0
000474 010067      MOV R0,REALB
177642
000500 011400      MOV #R4,R0
000502 011501      MOV #R5,R1
000504 060100      ADD R1,R0
000506 010067      MOV R0,IMAGA
177626
000512 006267      ASR IMAGA
177622
000516 160100      SUB R1,R0
000520 160100      SUB R1,R0
000522 006200      ASR R0
000524 010067      MOV R0,IMAGB
177614
000530 016700      MOV INDEX,R0
177252
000534 006200      ASR R0
000536 016701      MOV WGTINC,R1
177260
000542 005701      COMP2A:TST R1
000544 003403      BLE COMP2B
000546 006300      ASL R0
000550 005301      DEC R1
000552 000773      BR COMP2A
000554 016067      COMP2B:MOV SCTABL(R0),COSW
002474
177232
000562 162700      SUB #1000,R0
001000
000566 005400      NEG R0
000570 016067      MOV SCTABL(R0),SINH
002474
177220

```

```

000576 016700      MOV SINW,R0
          177214
000602 016702      MOV REALB,R2
          177534
000606 004767      JSR PC,MULT
          001246
000612 010103      MOV R1,R3
000614 016700      MOV COSW,R0
          177174
000620 016702      MOV IMAGA,R2
          177514
000624 004767      JSR PC,MULT
          001230
000630 060103      ADD R1,R3
000632 016700      MOV REALA,R0
          177500
000636 060300      ADD R3,R0
000640 010044      MOV R0,-(R4)
000642 160300      SUB R3,R0
000644 160300      SUB R3,R0
000646 010045      MOV R0,-(R5)
000650 005724      TST (R4)+
000652 005725      TST (R5)+
000654 016700      MOV COSH,R0
          177134
000660 016702      MOV REALB,R2
          177456
000664 004767      JSR PC,MULT
          001170
000670 010103      MOV R1,R3
000672 016700      MOV SINW,R0
          177120
000676 016702      MOV IMAGA,R2
          177436
000702 004767      JSR PC,MULT
          001152
000706 160301      SUB R3,R1
000710 016700      MOV IMAGB,R0
          177430
000714 060001      ADD R0,R1
000716 010114      MOV R1,0R4
000720 160001      SUB R0,R1
000722 160001      SUB R0,R1
000724 010115      MOV R1,0R5
000726 062767      ADD #4,INDEX
          000004
          177052
000734 016703      MOV INDEX,R3
          177046
000740 000627      BR COMPL1

```

```

;-----
;-----
;-----SUBROUTINE NORM-----
;-----
;-----
000742 016705 NORM:  MOV LOCATR,R5
          177044
000746 016703      MOV SIZE,R3
          177026
000752 005303      DEC R3
000754 010504      MOV R5,R4
000756 012500      MOV (R5)+,R0
000760 100401      BMI NORM1
000762 005400      NEG R0
000764 011501 NORM1: MOV @R5,R1
000766 100401      BMI NORM2
000770 005401      NEG R1
000772 020001 NORM2: CMP R0,R1
000774 002402      BLT NORM3
000776 010504      MOV R5,R4
001000 010100      MOV R1,R0
001002 005725 NORM3: TST (R5)+
001004 005303      DEC R3
001006 001366      BNE NORM1
001010 011400      MOV @R4,R0
001012 001453      BEQ NORM9
001014 100001      BPL NORM4
001016 005400      NEG R0
001020 012704 NORM4: MOV #2,R4
          000002
001024 022700      CMP #100000,R0
          100000
001030 001407      BEQ NORM7
001032 006300      ASL R0
001034 006300      ASL R0
001036 103404      BCS NORM7
001040 006300 NORM6: ASL R0
001042 005304      DEC R4
001044 103401      BCS NORM7
001046 000774      BR NORM6
001050 005700 NORM7: TST R0
001052 001001      BNE NORM7C
001054 005304      DEC R4
001056 005704 NORM7C: TST R4
001060 001430      BEQ NORM9
001062 016705      MOV LOCATR,R5
          176724
001066 016701      MOV SIZE,R1
          176706
001072 060467      ADD R4,SCALER
          176712
001076 005704      TST R4
001100 002410      BLT NORM8
001102 010400 NORM7A: MOV R4,R0
001104 006215 NORM7B: ASR @R5

```

```

001106 005300      DEC R0
001110 003375      BGT NORM7B
001112 005725      TST (R5)+
001114 005301      DEC R1
001116 001411      BEQ NORM9
001120 000770      BR NORM7A
001122 010400      NORM8: MOV R4,R0
001124 006315      NORM8A: ASL @R5
001126 005200      INC R0
001130 002775      BLT NORM8A
001132 005725      TST (R5)+
001134 005301      DEC R1
001136 001401      BEQ NORM9
001140 000770      BR NORM8
001142 000207      NORM9: RTS PC
;-----
;-----
;-----SUBROUTINE SORT-----
;-----
;-----
001144 000000      LOAD0: .WORD 0
001146 000000      HIADD: .WORD 0
001150 000000      COUNT: .WORD 0
001152 016767      SORT:  MOV SIZE,COUNT
176622
177770
001160 162767      SUB #2,COUNT
000002
177762
001166 010700      MOV PC,R0
001170 000415      BR SORT1
001172 000000      SORT0: .WORD 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
001174 000000
001176 000000
001200 000000
001202 000000
001204 000000
001206 000000
001210 000000
001212 000000
001214 000000
001216 000000
001220 000000
001222 000000
001224 005720      SORT1: TST (R0)+
001226 005720      TST (R0)+
001230 010067      MOV R0,SORT0
177736
001234 016701      MOV SIZE,R1
176540
001240 005301      DEC R1
001242 012702      MOV #1,R2
000001
001246 005067      SORT1A: CLR HIADD
177674

```

```

001252 010267      MOV R2,LOAD0
      177666
001256 005005      CLR R5
001260 010203      MOV R2,R3
001262 006303  SORT2: ASL R3
001264 020103      CMP R1,R3
001266 003007      BGT SORT2A
001270 006203      ASR R3
001272 020367      CMP R3,HIADD
      177650
001276 002402      BLT SORT2B
001300 010367      MOV R3,HIADD
      177642
001304 000403  SORT2B: BR SORT3
001306 010320  SORT2A: MOV R3,(R0)+
001310 005205      INC R5
001312 000763      BR SORT2
001314 160103  SORT3: SUB R1,R3
001316 006303      ASL R3
001320 060103      ADD R1,R3
001322 005703      TST R3
001324 003010      BGT SORT3A
001326 014003      MOV -(R0),R3
001330 005720      TST (R0)+
001332 020367      CMP R3,LOAD0
      177606
001336 003002      BGT SORT3B
001340 010367      MOV R3,LOAD0
      177600
001344 000746  SORT3B: BR SORT2
001346 010320  SORT3A: MOV R3,(R0)+
001350 005205      INC R5
001352 020302      CMP R3,R2
001354 001401      BEQ SORT4
001356 000756      BR SORT3
001360 020267  SORT4: CMP R2,LOAD0
      177560
001364 003057      BGT SORT6
001366 160102      SUB R1,R2
001370 005402      NEG R2
001372 020267      CMP R2,HIADD
      177550
001376 002446      BLT SORT5A
001400 010567      MOV R5,LOAD0
      177540
001404 160567      SUB R5,COUNT
      177540
001410 160102      SUB R1,R2
001412 005402      NEG R2
001414 010203  SORT4B: MOV R2,R3

```

```

001416 006303      ASL R3
001420 066703      ADD LOCATR,R3
          176366
001424 011346      MOV @R3,-(SP)
001426 016700      MOV SORT0,R0
          177540
001432 011004      SORT4A:MOV @R0,R4
001434 160110      SUB R1,@R0
001436 005420      NEG (R0)+
001440 020204      CMP R2,R4
001442 001406      BEQ SORT5
001444 006304      ASL R4
001446 066704      ADD LOCATR,R4
          176340
001452 011413      MOV @R4,@R3
001454 010403      MOV R4,R3
001456 000765      BR SORT4A
001460 012613      SORT5:MOV (SP)+,@R3
001462 160102      SUB R1,R2
001464 005402      NEG R2
001466 006302      ASL R2
001470 020201      CMP R2,R1
001472 002413      BLT SORT6A
001474 006202      ASR R2
001476 020267      CMP R2,HIADD
          177444
001502 001404      BEQ SORT5A
001504 166767      SUB LOADD,COUNT
          177434
          177436
001512 000740      BR SORT4B
001514 160102      SORT5A:SUB R1,R2
001516 005402      NEG R2
001520 000401      BR SORT6
001522 006202      SORT6A:ASR R2
001524 022767      SORT6: CMP #3,COUNT
          000003
          177416
001532 003005      BGT SORT7
001534 062702      ADD #2,R2
          000002
001540 016700      MOV SORT0,R0
          177426
001544 000640      BR SORT1A
001546 005767      SORT7: TST COUNT
          177376
001552 001414      BEQ SORT8
001554 042701      BIC #125252,R1
          125252
001560 006301      ASL R1

```

```

001562 010102      MOV R1,R2
001564 066701      ADD LOCATR,R1
          176222
001570 011104      MOV R1,R4
001572 006302      ASL R2
001574 066702      ADD LOCATR,R2
          176212
001600 011211      MOV R2,R1
001602 010412      MOV R4,R2
001604 000207      SORT0: RTS PC
          ;-----
          ;-----
          ;-----SUBROUTINE GETADR-----
          ;-----
          ;-----
001606 016700      GETADR:MOV INDEX,R0
          176174
001612 016701      MOV PASS,R1
          176164
001616 005700      TST R0
001620 001001      BNE GETA00
001622 000412      BR GETAD1
001624 010002      GETA00:MOV R0,R2
001626 010103      MOV R1,R3
001630 005403      NEG R3
001632 040302      BIC R3,R2
001634 001005      BNE GETAD1
001636 060100      ADD R1,R0
001640 010067      MOV R0,INDEX
          176142
001644 005067      CLR WEIGHT
          176134
001650 006500      GETAD1:ASL R0
001652 066700      ADD LOCATR,R0
          176134
001656 010004      MOV R0,R4
001660 006301      ASL R1
001662 060100      ADD R1,R0
001664 010005      MOV R0,R5
001666 000207      RTS PC
          ;-----
          ;-----
          ;-----SUBROUTINE LOOKUP-----
          ;-----
          ;-----
001670 016700      LOOKUP:MOV WEIGHT,R0
          176110
001674 005002      CLR R2
001676 010001      MOV R0,R1
001700 042701      BIC #177577,R1
          177577
001704 001404      BEQ LOOK1
001706 162700      SUB #400,R0
          000400
001712 005400      NEG R0
001714 005202      INC R2
001716 006500      LOOK1: ASL R0
001720 006300      ASL R0
001722 016003      MOV SCTABL(R0),R3
          002474

```



```

001726 162700      SUB #1000,R0
001732 005400      NEG R0
001734 016004      MOV SGTABL(R0),R4
001740 005702      TST R2
001742 001401      BEQ LOOK2
001744 005403      NEG R3
001746 000207      LOOK2: RTS PC
;-----
;-----SUBROUTINE BITINV-----
;-----
001750 012700      BITINV:MOV #4,R0
001754 010001      BITIN0:MOV R0,R1
001756 010104      MOV R1,R4
001760 010703      MOV SIZE,R3
001764 005002      CLR R2
001766 006201      BITIN1:ASR R1
001770 006102      ROL R2
001772 006203      ASR R3
001774 103374      BCC BITIN1
001776 006302      ASL R2
002000 006302      ASL R2
002002 020204      CMP R2,R4
002004 003414      BLE BITIN3
002006 066704      BITIN2:ADD LOCATR,R4
002012 066702      ADD LOCATR,R2
002016 011405      MOV @R4,R5
002020 011214      MOV @R2,@R4
002022 010512      MOV R5,@R2
002024 005724      TST (R4)+
002026 005722      TST (R2)+
002030 011405      MOV @R4,R5
002032 011214      MOV @R2,@R4
002034 010512      MOV R5,@R2
002036 062700      BITIN3:ADD #4,R0
002042 006200      ASR R0
002044 020067      CMP R0,SIZE
002050 001402      BEQ BITIN4
002052 006300      ASL R0
002054 000737      BR BITIN0
002056 000207      BITIN4:RTS PC
;-----
;-----SUBROUTINE MULT-----
;-----
002060 005046      MULT: CLR -(SP)
002062 005702      TST R2
002064 001430      BEQ MULT6
002066 002002      BGE MULT1
002070 005402      NEG R2

```

```

002072 005216      INC @SP
002074 005700  MULT1: TST R0
002076 001423      BEQ MULT6
002100 002002      BGE MULT2
002102 005400      NEG R0
002104 005316      DEC @SP
002106 012746  MULT2: MOV #20,-(SP)
                   000020
002112 005001      CLR R1
002114 006001  MULT3: ROR R1
002116 006002      ROR R2
002120 103001      BCC MULT4
002122 060001      ADD R0,R1
002124 005316  MULT4: DEC @SP
002126 001372      BNE MULT3
002130 005726      TST (SP)+
002132 005726      TST (SP)+
002134 001403      BEQ MULT5
002136 005401      NEG R1
002140 005402      NEG R2
002142 005601      SBC R1
002144 000207  MULT5: RTS PC
002146 005001  MULT6: CLR R1
002150 005002      CLR R2
002152 005726      TST (SP)+
002154 000207      RTS PC
;-----
;-----
;-----SUBROUTINE MAGNTO-----
;-----
;-----
002156 016705  MAGNTO:MOV LOCATR,R5
                   175630
002162 011500  MAG1:  MOV @R5,R0
002164 010002      MOV R0,R2
002166 004767      JSR PC,MULT
                   177666
002172 010504      MOV R5,R4
002174 066704      ADD SIZE,R4
                   175600
002200 011400      MOV @R4,R0
002202 010103      MOV R1,R3
002204 010204      MOV R2,R4
002206 010002      MOV R0,R2
002210 004767      JSR PC,MULT
                   177644
002214 060402      ADD R4,R2
002216 005501      AOC R1
002220 060301      ADD R3,R1
002222 004767      JSR PC,SOROOT
                   000024
002226 010115      MOV R1,@R5
002230 005725      TST (R5)+
002232 010504      MOV R5,R4
002234 166704      SUB LOCATR,R4
                   175552

```

```

002240 026704      CMP SIZE,R4
          175534
002244 001401      BEQ MAG2
002246 000745      BR MAG1
002250 000207      MAG2: RTS PC
          ;-----
          ;-----
          ;-----SUBROUTINE SQROOT-----
          ;-----
          ;-----
002252 005004      SQROOT: CLR R4
002254 005701          TST R1
002256 002435          BLT SQROT5
002260 003002          BGT SQROT1
002262 005702          TST R2
002264 001430          BEQ SQROT4
002266 005204      SQROT1: INC R4
002270 006302          ASL R2
002272 006101          ROL R1
002274 100374          BPL SQROT1
002276 000241          CLC
002300 006001          ROR R1
002302 006002          ROR R2
002304 005304          DEC R4
002306 010102          MOV R1,R2
002310 004767          JSR PC,SQRT
          000040
002314 006204          ASR R4
002316 103005          BCC SQROT2
002320 012700          MOV #55202,R0
          055202
002324 010102          MOV R1,R2
002326 004767          JSR PC,MULT
          177526
002332 005704      SQROT2: TST R4
002334 001403          BEQ SQROT3
002336 006201          ASR R1
002340 005304          DEC R4
002342 000773          BR SQROT2
002344 000207      SQROT3: RTS PC
002346 005001      SQROT4: CLR R1
002350 000207          RTS PC
002352 000000      SQROT5: HALT
          ;-----
          ;-----
          ;-----SUBROUTINE SQRT-----
          ;-----
          ;-----
002354 012746      SQRT:  MOV #4,-(SP)
          000004
002360 010203          MOV R2,R3
002362 010246          MOV R2,-(SP)
002364 062703          ADD #100000,R3
          100000
002370 010346          MOV R3,-(SP)
002372 006203      SQRT1: ASR R3
002374 005403          NEG R3
002376 010300          MOV R3,R0
002400 004767          JSR PC,MULT
          177454

```

```

002404 066601      ADD 2(SP),R1
      000002
002410 005366      DEC 4(SP)
      000004
002414 001003      BNE SQRT2
002416 062706      ADD #6,SP
      000006
002422 000207      RTS PC
002424 010166      SQRT2: MOV R1,2(SP)
      000002
002430 011600      MOV #SP,R0
002432 010002      MOV R0,R2
002434 004767      JSR PC,MULT
      177420
002440 011602      MOV #SP,R2
002442 010146      MOV R1,-(SP)
002444 006202      ASR R2
002446 006202      ASR R2
002450 062702      ADD #120000,R2
      120000
002454 012600      MOV (SP)+,R0
002456 004767      JSR PC,MULT
      177376
002462 010116      MOV R1,#SP
002464 010103      MOV R1,R3
002466 016602      MOV 2(SP),R2
      000002
002472 000737      BR SQRT1
      //////////////////////////////////////
      //////////////////////////////////////
      ////////////////////////////////////// SINE-COSINE TABLE //////////////////////////////////////
      //////////////////////////////////////
      //////////////////////////////////////

002474 077777      SCTABL: .WORD 77777,77777,77775,77772
002476 077777
002500 077775
002502 077772
002504 077766      .WORD 77766,77760,77751,77741
002506 077760
002510 077751
002512 077741
002514 077730      .WORD 77730,77716,77702,77665
002516 077716
002520 077702
002522 077665
002524 077647      .WORD 77647,77627,77607,77565
002526 077627
002530 077607
002532 077565
002534 077542      .WORD 77542,77515,77470,77441
002536 077515
002540 077470
002542 077441
002544 077411      .WORD 77411,77360,77325,77272
002546 077360

```

002550	077325		
002552	077272		
002554	077235	.WORD	77235,77177,77137,77077
002556	077177		
002560	077137		
002562	077077		
002564	077035	.WORD	77035,76772,76726,76660
002566	076772		
002570	076726		
002572	076660		
002574	076612	.WORD	76612,76542,76471,76417
002576	076542		
002600	076471		
002602	076417		
002604	076343	.WORD	76343,76267,76211,76132
002606	076267		
002610	076211		
002612	076132		
002614	076051	.WORD	76051,75770,75705,75622
002616	075770		
002620	075705		
002622	075622		
002624	075535	.WORD	75535,75446,75357,75266
002626	075446		
002630	075357		
002632	075266		
002634	075175	.WORD	75175,75102,75005,74710
002636	075102		
002640	075005		
002642	074710		
002644	074612	.WORD	74612,74512,74411,74307
002646	074512		
002650	074411		
002652	074307		
002654	074204	.WORD	74204,74100,73772,73664
002656	074100		
002660	073772		
002662	073664		
002664	073554	.WORD	73554,73443,73331,73216
002666	073443		
002670	073331		
002672	073216		
002674	073101	.WORD	73101,72764,72645,72525
002676	072764		
002700	072645		
002702	072525		
002704	072404	.WORD	72404,72262,72137,72013
002706	072262		
002710	072137		
002712	072013		

002714 071665	.WORD	71665,71537,71407,71257
002716 071537		
002720 071407		
002722 071257		
002724 071125	.WORD	71125,70772,70636,70501
002726 070772		
002730 070636		
002732 070501		
002734 070342	.WORD	70342,70203,70043,67701
002736 070203		
002740 070043		
002742 067701		
002744 067537	.WORD	67537,67373,67226,67060
002746 067373		
002750 067226		
002752 067060		
002754 066712	.WORD	66712,66542,66371,66217
002756 066542		
002760 066371		
002762 066217		
002764 066044	.WORD	66044,65670,65512,65334
002766 065670		
002770 065512		
002772 065334		
002774 065155	.WORD	65155,64775,64614,64431
002776 064775		
003000 064614		
003002 064431		
003004 064246	.WORD	64246,64062,63675,63506
003006 064062		
003010 063675		
003012 063506		
003014 063317	.WORD	63317,63127,62735,62543
003016 063127		
003020 062735		
003022 062543		
003024 062350	.WORD	62350,62154,61757,61561
003026 062154		
003030 061757		
003032 061561		
003034 061362	.WORD	61362,61161,60761,60557
003036 061161		
003040 060761		
003042 060557		
003044 060354	.WORD	60354,60150,57743,57536
003046 060150		
003050 057743		
003052 057536		
003054 057327	.WORD	57327,57120,56707,56476
003056 057120		

003060	056707		
003062	056476		
003064	056264	.WORD	56264,56051,55635,55420
003066	056051		
003070	055635		
003072	055420		
003074	055202	.WORD	55202,54763,54544,54324
003076	054763		
003100	054544		
003102	054324		
003104	054102	.WORD	54102,53660,53435,53212
003106	053660		
003110	053435		
003112	053212		
003114	052765	.WORD	52765,52540,52312,52063
003116	052540		
003120	052312		
003122	052063		
003124	051633	.WORD	51633,51402,51151,50716
003126	051402		
003130	051151		
003132	050716		
003134	050463	.WORD	50463,50227,47773,47536
003136	050227		
003140	047773		
003142	047536		
003144	047277	.WORD	47277,47041,46601,46341
003146	047041		
003150	046601		
003152	046341		
003154	046077	.WORD	46077,45636,45373,45130
003156	045636		
003160	045373		
003162	045130		
003164	044664	.WORD	44664,44417,44151,43703
003166	044417		
003170	044151		
003172	043703		
003174	043434	.WORD	43434,43165,42715,42444
003176	043165		
003200	042715		
003202	042444		
003204	042172	.WORD	42172,41720,41445,41172
003206	041720		
003210	041445		
003212	041172		
003214	040716	.WORD	40716,40441,40163,37705
003216	040441		
003220	040163		
003222	037705		

003224 037427	.WORD	37427,37150,36670,36407
003226 037150		
003230 036670		
003232 036407		
003234 036126	.WORD	36126,35645,35362,35100
003236 035645		
003240 035362		
003242 035100		
003244 034614	.WORD	34614,34330,34044,33557
003246 034330		
003250 034044		
003252 033557		
003254 033272	.WORD	33272,33004,32515,32226
003256 033004		
003260 032515		
003262 032226		
003264 031736	.WORD	31736,31446,31156,30665
003266 031446		
003270 031156		
003272 030665		
003274 030373	.WORD	30373,30101,27607,27314
003276 030101		
003300 027607		
003302 027314		
003304 027021	.WORD	27021,26525,26230,25734
003306 026525		
003310 026230		
003312 025734		
003314 025437	.WORD	25437,25141,24643,24345
003316 025141		
003320 024643		
003322 024345		
003324 024046	.WORD	24046,23547,23250,22750
003326 023547		
003330 023250		
003332 022750		
003334 022450	.WORD	22450,22147,21646,21345
003336 022147		
003340 021646		
003342 021345		
003344 021043	.WORD	21043,20541,20237,17734
003346 020541		
003350 020237		
003352 017734		
003354 017431	.WORD	17431,17126,16623,16317
003356 017126		
003360 016623		
003362 016317		
003364 016013	.WORD	16013,15507,15202,14675
003366 015507		


```

003370 015202
003372 014675
003374 014370      .WORD    14370,14063,13555,13250
003376 014063
003400 013555
003402 013250
003404 012742      .WORD    12742,12433,12125,11616
003406 012433
003410 012125
003412 011616
003414 011310      .WORD    11310,11001,10471,10162
003416 011001
003420 010471
003422 010162
003424 007653      .WORD    07653,07343,07033,06523
003426 007343
003430 007033
003432 006523
003434 006213      .WORD    06213,05703,05373,05063
003436 005703
003440 005373
003442 005063
003444 004552      .WORD    04552,04242,03731,03420
003446 004242
003450 003731
003452 003420
003454 003107      .WORD    03107,02577,02266,01755
003456 002577
003460 002266
003462 001755
003464 001444      .WORD    01444,01133,00622,00311
003466 001133
003470 000622
003472 000311
003474 000000      .WORD    00000
;
;
000001      .END

```

```

FAST FOURIER TRANSFORM MODULE.
DOUBLE WORD INTEGER FORMAT.
NICOLET 1070 STORAGE FORMAT.
PROGRAMMED BY LEE WINDMILLER.
ARGONNE NATIONAL LABORATORY.
ARGONNE, ILLINOIS 60439.

```

REGISTER DEFINITIONS

ADDR	003716	ADDR1I	000556	ADDR1R	000554	ADDR2I	000562
ADDR2R	000560	ADOTBL	000552	BITINV	002620	BITINB	002630
BITIN1	002642	BITIN2	002666	BITIN3	002724	BITIN4	002744
CHNGE1	000500	COMPLX	002746	COMPL1	003062	COMPL2	003134
COMP0	002756	COMP0A	002774	COMP2A	003324	COMP2B	003336
COMP2C	003412	COMP2D	003424	CONTRL	000006	COSF	002406
COSW	000526	COUNT	001600	CRLF	000366	DATIN	000440
DATIN1	000454	DATOT1	000430	DATPUT	000414	DATTBL	000600
DESI	000412	ERROR	000000	ERROR1	003742	EXEC	000002
EXEC0	000070	EXEC1	000140	EXEC10	000122	EXEC2	000226
EXEC3	000242	EXEC4	000334	FFT	000716	FFT1	000722
FFT2	001302	GETADR	002306	GETAD0	002324	GETAD1	002350
HIA00	001576	IMAG1	000600	INAG2	000616	INDEX	000522
LOAD0	001574	LOOKUP	002472	LOOK1	002520	LOOK2	002616
MAGNTO	004052	MAG1	004056	MAG2	004200	MULT	003746
MULT1	003764	MULT2	004000	MULT3	004010	MULT4	004024
MULT5	004050	PASS	000516	PC	*X000007	REAL1	000602
REAL2	000612	R0	*X000000	R1	*X000001	R2	*X000002
R3	*X000003	R4	*X000004	R5	*X000005	SCALER	000524
SCTABL	004542	SINF	002434	SINH	000532	SIZE	000514
SIZER	001324	SIZER1	001346	SIZER2	001354	SIZER3	001364
SIZER4	001402	SIZER6	001434	SIZER7	001444	SIZER8	001534
SIZER9	001572	SIZE7A	001476	SIZE7B	001506	SIZE7C	001452
SIZE8A	001544	SORT	001602	SORT0	001622	SORT1	001654
SORT1A	001676	SORT2	001712	SORT2A	001736	SORT2B	001734
SORT3	001744	SORT3A	001776	SORT3B	001774	SORT4	002010
SORT4A	002070	SORT4B	002044	SORT5	002126	SORT5A	002174
SORT6	002204	SORT6A	002202	SORT7	002226	SORT8	002304
SOURCE	000410	SP	*X000006	SQR00T	004204	SQR0T1	004212
SQR0T2	004274	SQR0T3	004310	SQR0T4	004314	SQR0T5	004324
SORT	004326	SQR1	004352	SQR2	004420	START	000546
START1	000564	START2	000622	START3	000700	STOR	000536
SUBF	003730	SWCNTR	000542	TYPE	000356	WEIGHT	000520
XGTINC	000544	XCHNGE	000464	.	* 005544		

```

000000      R0=X0
000001      R1=X1
000002      R2=X2
000003      R3=X3
000004      R4=X4
000005      R5=X5
000006      SP=X6
000007      PC=X7

000000      ;
000000      ;      .*=0
000000      ;      ERROR HALT
000000 000000  ERROR: HALT
;-----
;-----
;-----SUBROUTINE EXEC-----
;-----
;      CALLING SEQUENCE,
;      JSR PC,EXEC
;      SEE TEXT FOR EXPLANATION OF
;      OPTIONS AVAILABLE.
;
000002 010701  EXEC:  MOV PC,R1
000004 000431      BR EXEC0
000006 000000  CONTRL: WORD 0.324,315,317,311,323,330,0,0
000010 000324
000012 000315
000014 000317
000016 000311
000020 000323
000022 000330
000024 000000
000026 000000
000030 000546      .WORD START,MAGTID,DATOUT,DATIN,SIZER,XCHNGE,0,0
000032 000452
000034 000414
000036 000440
000040 001324
000042 000464
000044 000000
000046 000000
000050 000000      .WORD 0,0,0,0,0,0,0,0
000052 000000
000054 000000
000056 000000
000060 000000
000062 000000
000064 000000
000066 000000
000070 005721  EXEC0: TST (R1)+
000072 005721      TST (R1)+
000074 010167      MOV R1,CONTRL
000076 177706
000100 162700      SUB #40,R0
000040
000104 012702      MOV #10,R2
000010
000110 010104      MOV R1,R4
000112 062701      ADD #20,R1
000020
000116 062704      ADD #40,R4
000040

```

```

000122 011114 EXEC10:MOV @R1,@R4
000124 060014      ADD R0,@R4
000126 005302      DEC R2
000130 001403      BEQ EXEC1
000132 005721      TST (R1)+
000134 005724      TST (R4)+
000136 000771      BR EXEC10
000140 012700 EXEC1: MOV #177560,R0
000142 177560
000144 012701      MOV #177562,R1
000146 177562
000150 012702      MOV #177564,R2
000152 177564
000154 012703      MOV #177566,R3
000156 177566
000160 105712      TSTB @R2
000162 100376      BPL ,-2
000164 004767      JSR PC,CRLF
000166 000176
000170 005210      INC @R0
000172 105710      TSTB @R0
000174 100376      BPL ,-2
000176 111104      MOVB @R1,R4
000200 105712      TSTB @R2
000202 100376      BPL ,-2
000204 110405      MOVB R4,R5
000206 004767      JSR PC,TYPE
000208 000144
000212 005704      TST R4
000214 001447      BEQ EXEC4
000216 012700      MOV #10,R0
000218 000010
000222 016701      MOV CONTRL,R1
000224 177560
000226 120411 EXEC2: CMPB R4,@R1
000230 001404      BEQ EXEC3
000232 005721      TST (R1)+
000234 005300      DEC R0
000236 001436      BEQ EXEC4
000240 000772      BR EXEC2
000242 012700 EXEC3: MOV #167776,R0
000244 167776
000246 012710      MOV #114000,@R0
000248 114000
000252 013700      MOV #167774,R0
000254 167774
000256 042700      BIC #140000,R0
000258 140000
000262 005700      TST R0
000264 001372      BNE ,-12

```

```

000266 016100      MOV 40(R1),R0
000040
000272 010701      MOV PC,R1
000274 062701      ADD #16,R1
000016
000300 160100      SUB R1,R0
000302 010067      MOV R0,2(PC)
000002
000306 004767      .WORD 4767
000310 000000      .WORD 0
000312 005737      TST #140000
140000
000316 112737      MOV8 #0,#167774
000000
167774
000324 012737      MOV #124000,#167776
124000
167776
000332 000702      BR EXEC1
000334 105712      EXEC4: TSTB #R2
000336 100376      BPL .-2
000340 004767      JSR PC,CRLF
000022
000344 012705      MOV #277,R5
000277
000350 004767      JSR PC,TYPE
000002
000354 000671      BR EXEC1
000356 010513      TYPE:  MOV R5,#R3
000360 105712      TSTB #R2
000362 100376      BPL .-2
000364 000207      RTS PC
000366 012705      CRLF:  MOV #215,R5
000215
000372 004767      JSR PC,TYPE
177760
000376 012705      MOV #212,R5
000212
000402 004767      JSR PC,TYPE
177750
000406 000207      RTS PC
000410 000000      SOURCE: .WORD 0
000412 000000      DEST:  .WORD 0
000414 016700      DATOUT: MOV SIZE,R0
000074
000420 016701      MOV DEST,R1
177766
000424 016702      MOV SOURCE,R2
177760
000430 012221      DATOUT1: MOV (R2)+,(R1)+

```

```

000432 005300      DEC R0
000434 001375      BNE DATOT1
000436 000207      RTS PC
000440 016/00      DATIN: MOV SIZE,R0
000440 000050
000444 016/01      MOV DEST,R1
000444 177742
000450 016/02      MOV SOURCE,R2
000450 177734
000454 012122      DATIN1:MOV (R1)+,(R2)+
000456 005300      DEC R0
000460 001375      BNE DATIN1
000462 000207      RTS PC
000464 016/00      XCHNGE:MOV SIZE,R0
000464 000024
000470 016/01      MOV DEST,R1
000470 177716
000474 016/02      MOV SOURCE,R2
000474 177710
000500 011103      CHNGE1:MOV @R1,R3
000502 011221      MOV @R2,(R1)+
000504 010322      MOV R3,(R2)+
000506 005300      DEC R0
000510 001373      BNE CHNGE1
000512 000207      RTS PC

;-----
;-----
;-----SUBROUTINE START-----
;-----
;-----

000514 000000      SIZE:      .WORD 0
000516 000000      PASS:      .WORD 0
000520 000000      HEIGHT:    .WORD 0
000522 000000      INDEX:      .WORD 0
000524 000000      SCALER:      .WORD 0
000526 000000      COSM:      .WORD 0,0
000530 000000
000532 000000      SINW:      .WORD 0,0
000534 000000
000536 000000      STOR:      .WORD 0,0
000540 000000
000542 000000      SHCNTR:    .WORD 0
000544 000000      WGTINC:    .WORD 0
000546 010/00      START:      MOV PC,R0
000550 000405      BR START1
000552 000000      ADDTBL:    .WORD 0
000554 000000      ADDR1R:    .WORD 0
000556 000000      ADDR1I:    .WORD 0
000560 000000      ADDR2R:    .WORD 0
000562 000000      ADDR2I:    .WORD 0
000564 005720      START1:    TST (R0)+
000566 005/20      TST (R0)+
000570 010067      MOV R0,ADDTBL
000570 177756
000574 010700      MOV PC,R0

```

```

000576 000411      BR START2
000600 000000  DATTL:  .WORD 0
000602 000000  REAL1:  .WORD 0,0
000604 000000
000606 000000  IMAG1:  .WORD 0,0
000610 000000
000612 000000  REAL2:  .WORD 0,0
000614 000000
000616 000000  IMAG2:  .WORD 0,0
000620 000000
000622 005720  START2:  TST (R0)+
000624 005720      TST (R0)+
000626 010067      MOV R0,DATTL
          177746
000632 005067      CLR SCALER
          177666
000636 005067      CLR INDEX
          177660
000642 016700      MOV SIZE,R0
          177646
000646 006200      ASR R0
000650 010067      MOV R0,PASS
          177642
000654 006200      ASR R0
000656 010067      MOV R0,SWCNTR
          177660
000662 005067      CLR HEIGHT
          177632
000666 012767      MOV #1,WGTINC
          000001
          177650
000674 016700      MOV SIZE,R0
          177614
000700 022700  START3:  CMP #2000,R0
          002000
000704 001404      BEO FFT
000706 006300      ASL R0
000710 006367      ASL WGTINC
          177630
000714 000771      BR START3
000716 004767  FFT1:  JSR PC,SIZER
          000402
000722 004767  FFT1:  JSR PC,GETADR
          001360
000726 016705      MOV ADDBL,R0
          177620
000732 016702      MOV DATTL,R2
          177642
000736 013700      MOV #(R5)+,R0
000740 013701      MOV #167774,R1
          167774

```

000744 010022	MOV R0,(R2)+
000746 010122	MOV R1,(R2)+
000750 013522	MOV @ (R5)+,(R2)+
000752 013722	MOV @#167774,(R2)+
167774	
000756 013503	MOV @ (R5)+,R3
000760 013704	MOV @#167774,R4
167774	
000764 010322	MOV R3,(R2)+
000766 010422	MOV R4,(R2)+
000770 013522	MOV @ (R5)+,(R2)+
000772 013712	MOV @#167774,R2
167774	
000776 004767	JSR PC,ADDF
002714	
001002 010137	MOV R1,@#167774
167774	
001006 010077	MOV R0,@ADDR1R
177542	
001012 004767	JSR PC,SUBF
002712	
001016 004767	JSR PC,SUBF
002706	
001022 010067	MOV R0,REAL1
177554	
001026 010167	MOV R1,REAL1+2
177552	
001032 016700	MOV IMAG1,R0
177550	
001036 016701	MOV IMAG1+2,R1
177546	
001042 016703	MOV IMAG2,R3
177550	
001046 016704	MOV IMAG2+2,R4
177546	
001052 004767	JSR PC,ADDF
002640	
001056 010137	MOV R1,@#167774
167774	
001062 010077	MOV R0,@ADDR1I
177470	
001066 004767	JSR PC,SUBF
002636	
001072 004767	JSR PC,SUBF
002632	
001076 010067	MOV R0,IMAG2
177514	
001102 010167	MOV R1,IMAG2+2
177512	
001106 004767	JSR PC,LOOKUP
001360	

001112 016702	MOV REAL1,R2
177464	
001116 016703	MOV REAL1+2,R3
177462	
001122 004767	JSR PC,COSF
001260	
001126 016702	MOV IMAG2,R2
177464	
001132 016703	MOV IMAG2+2,R3
177462	
001136 005402	NEG R2
001140 005403	NEG R3
001142 005602	SBC R2
001144 004767	JSR PC,SINF
001264	
001150 010077	MOV R0,@ADDR2R
177404	
001154 016702	MOV IMAG2,R2
177436	
001160 016703	MOV IMAG2+2,R3
177434	
001164 004767	JSR PC,COSF
001216	
001170 016702	MOV REAL1,R2
177406	
001174 016703	MOV REAL1+2,R3
177404	
001200 004767	JSR PC,SINF
001230	
001204 010077	MOV R0,@ADDR2I
177352	
001210 062767	ADD #2,INDEX
000002	
177304	
001216 066767	ADD WGTINC,WEIGHT
177322	
177274	
001224 005367	DEC SWCNTR
177312	
001230 001234	BNE FFT1
001232 006367	ASL WGTINC
177306	
001236 006267	ASH PASS
177254	
001242 022767	CMP #1,PASS
000001	
177246	
001250 001414	BEQ FFT2
001252 005067	CLR WEIGHT
177242	

```

001256 005067      CLH INDEX
          177240
001262 016700      MOV SIZE,R0
          177226
001266 006200      ASR R0
001270 006200      ASR R0
001272 010067      MOV R0,SWCNTN
          177244
001276 000167      JMP FFT
          177414
001302 004767 FFT2: JSR PC,SIZER
          000016
001306 004767      JSR PC,BITINV
          001306
001312 004767      JSR PC,COMPLX
          001430
001316 004767      JSR PC,SORT
          000260
001322 000207      RTS PC
          -----
          -----
          -----SUBROUTINE SIZER-----
          -----
          -----
001324 012705      SIZER: MOV #140000,R5
          140000
001330 016703      MOV SIZE,R3
          177100
001334 005303      DEC R3
001336 010504      MOV R5,R4
001340 012500      MOV (R5)+,R0
001342 100401      BMI SIZER1
001344 005400      NEG R0
001346 011501 SIZER1: MOV @R5,R1
001350 100401      BMI SIZER2
001352 005401      NEG R1
001354 020001 SIZER2: CMP R0,R1
001356 002402      BLT SIZER3
001360 010504      MOV R5,R4
001362 010100      MOV R1,R0
001364 005725 SIZER3: TST (R5)+
001366 005303      DEC R3
001370 001366      BNE SIZER1
001372 011400      MOV @R4,R0
001374 001476      BEQ SIZER9
001376 100001      BPL SIZER4
001400 005400      NEG R0
001402 000241 SIZER4: CLC
001404 013705      MOV #167774,R5
          167774
001410 012704      MOV #2,R4
          000002
001414 022700      CMP #100000,R0
          100000
001420 001411      BEQ SIZER7

```

```

001422 006305      ASL R5
001424 006100      ROL R0
001426 006305      ASL R5
001430 006100      ROL R0
001432 103404      BCS SIZE7
001434 006300      SIZE6: ASL R0
001436 005304      DEC R4
001440 103401      BCS SIZE7
001442 000774      BR SIZE6
001444 005700      SIZE7: TST R0
001446 001001      BNE SIZE7C
001450 005304      DEC R4
001452 005704      SIZE7C: TST R4
001454 001446      BEQ SIZE9
001456 012705      MOV #140000,R5
          140000
001462 016701      MOV SIZE,R1
          177026
001466 060467      ADD R4,SCALER
          177032
001472 005704      TST R4
001474 002417      BLT SIZE8
001476 010400      SIZE7A: MOV R4,R0
001500 011503      MOV #R5,R3
001502 013702      MOV #167774,R2
          167774
001506 006203      SIZE7B: ASR R3
001510 006002      ROR R2
001512 005300      DEC R0
001514 003374      BGT SIZE7B
001516 010237      MOV R2,#167774
          167774
001522 010315      MOV R3,#R5
001524 005725      TST (R5)+
001526 005301      DEC R1
001530 001420      BEQ SIZE9
001532 000761      BR SIZE7A
001534 010400      SIZE8: MOV R4,R0
001536 011503      MOV #R5,R3
001540 013702      MOV #167774,R2
          167774
001544 006302      SIZE9A: ASL R2
001546 006103      ROL R3
001550 005200      INC R0
001552 002774      BLT SIZE8A
001554 010237      MOV R2,#167774
          167774
001560 010315      MOV R3,#R5
001562 005725      TST (R5)+
001564 005301      DEC R1

```

```

001566 001401      BEQ SIZER9
001570 000761      BR SIZER8
001572 000207      SIZER9:      RTS PC
;-----
;-----
;-----SUBROUTINE SORT-----
;-----
;-----
001574 000000      LOADD:      .WORD 0
001576 000000      HIADD:      .WORD 0
001600 000000      COUNT:      .WORD 0
001602 016767      SORT:      MOV SIZE,COUNT
                                176706
                                177770
001610 162767      SUB #2,COUNT
                                000002
                                177762
001616 010700      MOV PC,R0
001620 000415      BR SORT1
001622 000000      SORT0:      .WORD 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
001624 000000
001626 000000
001630 000000
001632 000000
001634 000000
001636 000000
001640 000000
001642 000000
001644 000000
001646 000000
001650 000000
001652 000000
001654 005720      SORT1:      TST (R0)+
001656 005720      TST (R0)+
001660 010067      MOV R0,SORT0
                                177730
001664 016701      MOV SIZE,R1
                                176624
001670 005301      DEC R1
001672 012702      MOV #1,R2
                                000001
001676 005067      SORT1A:     CLR HIADD
                                177674
001702 010267      MOV R2,LOADD
                                177666
001706 005005      CLR R5
001710 010203      MOV R2,R3
001712 006303      SORT2:      ASL R3
001714 020103      CMP R1,R3
001716 003007      BGT SORT2A
001720 006203      ASR R3
001722 020367      CMP R3,HIADD
                                177650
001726 002402      BLT SORT2B
001730 010367      MOV R3,HIADD
                                177642

```

```

001734 000403 SORT2B: BR SORT3
001736 010320 SORT2A: MOV R3,(R0)+
001740 005205 INC R5
001742 000763 BR SORT2
001744 160103 SORT3: SUB R1,R3
001746 006303 ASL R3
001750 060103 ADD R1,R3
001752 005703 TST R3
001754 003010 BGT SORT3A
001756 014003 MOV -(R0),R3
001760 005720 TST (R0)+
001762 020367 CMP R3,LOAD0
177606
001766 003002 BGT SORT3B
001770 010367 MOV R3,LOAD0
177600
001774 000746 SORT3B: BR SORT2
001776 010320 SORT3A: MOV R3,(R0)+
002000 005205 INC R5
002002 020302 CMP R3,R2
002004 001401 BEQ SORT4
002006 000756 BR SORT3
002010 020267 SORT4: CMP R2,LOAD0
177560
002014 003073 BGT SORT6
002016 160102 SUB R1,R2
002020 005402 NEG R2
002022 020267 CMP R2,HIADD
177550
002026 002462 BLT SORT5A
002030 010567 MOV R5,LOAD0
177540
002034 160567 SUB R5,COUNT
177540
002040 160102 SUB R1,R2
002042 005402 NEG R2
002044 010203 SORT4B: MOV R2,R3
002046 006303 ASL R3
002050 016367 MOV 140000(R3),REAL1
140000
176524
002056 013767 MOV #167774,REAL1+2
167774
176520
002064 016700 MOV SORT0,R0
177532
002070 011004 SORT4A: MOV #R0,R4
002072 160110 SUB R1,#R0
002074 005420 NEG (R0)+
002076 020204 CMP R2,R4

```

```

002100 001412      BEQ SORT5
002102 006304      ASL R4
002104 016405      MOV 140000(R4),R5
          140000
002110 013737      MOV 0#167774,0#167774
          167774
          167774
002116 010563      MOV R5,140000(R3)
          140000
002122 010403      MOV R4,R3
002124 000761      BR SORT4A
002126 016737      MOV REAL1+2,0#167774
          176452
          167774
002134 016763      MOV REAL1,140000(R3)
          176442
          140000
002142 160102      SUB R1,R2
002144 005402      NEG R2
002146 006302      ASL R2
002150 020201      CMP R2,R1
002152 002413      BLT SORT6A
002154 006202      ASR R2
002156 020267      CMP R2,HIADD
          177414
002162 001404      BEQ SORT5A
002164 166767      SUB LOAD0,COUNT
          177404
          177406
002172 000724      BR SORT4B
002174 160102      SUB R1,R2
002176 005402      NEG R2
002200 000401      BR SORT6
002202 006202      ASR R2
002204 022767      CMP #3,COUNT
          000003
          177366
002212 003005      BGT SORT7
002214 062702      ADD #2,R2
          000002
002220 016700      MOV SORT0,R0
          177376
002224 000624      BR SORT1A
002226 005767      TST COUNT
          177346
002232 001424      BEQ SORT8
002234 042701      BIC #125252,R1
          125252
002240 006301      ASL R1
002242 016104      MOV 140000(R1),R4
          140000

```

```

002246 013705      MOV 0#167774,R5
          167774
002252 010102      MOV R1,R2
002254 006302      ASL R2
002256 016203      MOV 140000(R2),R3
          140000
002262 013737      MOV 0#167774,0#167774
          167774
          167774
002270 010361      MOV R3,140000(R1)
          140000
002274 010537      MOV R5,0#167774
          167774
002300 010462      MOV R4,140000(R2)
          140000
002304 000207      SORT0:      RTS PC
          ;-----
          ;-----
          ;-----SUBROUTINE GETADR-----
          ;-----
          ;-----
002306 016700      GETADR:      MOV INDEX,R0
          176210
002312 010701      MOV PASS,R1
          176200
002316 005700      TST R0
002320 001001      BNE GETAD0
002322 000412      BR GETAD1
002324 010002      GETAD0:      MOV R0,R2
002326 010103      MOV R1,R3
002330 005403      NEG R3
002332 040302      BIC R3,R2
002334 001005      BNE GETAD1
002336 060100      ADD R1,R0
002340 010067      MOV R0,INDEX
          176156
002344 005067      CLR WEIGHT
          176150
002350 006300      GETAD1:      ASL R0
002352 062700      ADD #140000,R0
          140000
002356 016702      MOV ADOTBL,R2
          176170
002362 010022      MOV R0,(R2)+
002364 005720      TST (R0)+
002366 010022      MOV R0,(R2)+
002370 006301      ASL R1
002372 060100      ADD R1,R0
002374 005740      TST =(R0)
002376 010022      MOV R0,(R2)+
002400 005720      TST (R0)+
002402 010012      MOV R0,R2
002404 000207      RTS PC
          ;-----
          ;-----
          ;-----SUBROUTINES COSF AND SINF-----
          ;-----
          ;-----
002406 016704      COSF:      MOV COSW,R4
          176114

```

```

002412 016705      MOV COSW+2,R5
          176112
002416 004767      JSR PC,MULT
          001324
002422 010067      MOV R0,STOR
          176110
002426 010167      MOV R1,STOR+2
          176106
002432 000207      RTS PC
002434 016704      SINF:  MOV SINW,R4
          176072
002440 016705      MOV SINW+2,R5
          176070
002444 004767      JSR PC,MULT
          001276
002450 016703      MOV STOR,R3
          176062
002454 016704      MOV STOR+2,R4
          176060
002460 004767      JSR PC,ADDF
          001232
002464 010157      MOV R1,#167774
          167774
002470 000207      RTS PC
          ;-----+-----
          ;-----+-----
          ;-----SUBROUTINE LOOKUP-----
          ;-----+-----
          ;-----+-----
002472 016700      LOOKUP:  MOV WEIGHT,R0
          176022
002476 005002      CLR R2
002500 010001      MOV R0,R1
002502 042701      BIC #177577,R1
          177577
002506 001404      BEQ LOOK1
002510 162700      SUB #400,R0
          000400
002514 005400      NEG R0
002516 005202      INC R2
002520 006300      LOOK1:  ASL R0
002522 006300      ASL R0
002524 016067      MOV 7000(R0),COSH
          007000
          175774
002532 005067      CLR COSW+2
          175772
002536 006067      ROR COSW
          175764
002542 006067      ROR COSW+2
          175762
002546 162700      SUB #1000,R0
          001000
002552 005400      NEG R0
002554 016067      MOV 7000(R0),SINW
          007000
          175750

```



```

002562 005067      CLR SINW+2
          175746
002566 006067      ROR SINW
          175740
002572 006067      ROR SINW+2
          175736
002576 005702      TST R2
002600 001406      BEQ LOOK2
002602 005467      NEG COSW
          175720
002606 005467      NEG COSW+2
          175716
002612 005667      SBC COSW
          175710
002616 000207      LOOK2:      RTS PC
          ;-----
          ;-----
          ;-----SUBROUTINE BITINV-----
          ;-----
          ;-----

002620 012700      BITINV:      MOV #4,R0
          000004
002624 012705      MOV #167774,R5
          167774
002630 010001      BITIN0:      MOV R0,R1
002632 010104      MOV R1,R4
002634 016703      MOV SIZE,R3
          175654
002640 005002      CLR R2
002642 006201      BITIN1:      ASR R1
002644 006102      ROL R2
002646 006203      ASR R3
002650 103374      BCC BITIN1
002652 006302      ASL R2
002654 006302      ASL R2
002656 020204      CMP R2,R4
002660 003421      BLE BITIN3
002662 012701      MOV #2,R1
          000002
002666 016446      BITIN2:      MOV 140000(R4),-(SP)
          140000
002672 011546      MOV @R5,-(SP)
002674 016203      MOV 140000(R2),R3
          140000
002700 011515      MOV @R5,@R5
002702 010364      MOV R3,140000(R4)
          140000
002706 012615      MOV (SP)+,@R5
002710 012662      MOV (SP)+,140000(R2)
          140000
002714 005722      TST (R2)+
002716 005724      TST (R4)+
002720 005301      DEC R1
002722 001361      BNE BITIN2
002724 062700      BITIN3:      ADD #4,R0
          000004

```

```

002730 006200      ASR R0
002732 020067      CMP R0,SIZE
          175556
002736 001402      BEQ BITIN4
002740 006300      ASL R0
002742 000732      BR BITIN0
002744 000207      RTS PC
          BITIN4:
          ;-----
          ;-----
          ;-----SUBROUTINE COMPLX-----
          ;-----
          ;-----
002746 005067      COMPLX:      CLR HGTINC
          175572
002752 016700      MOV SIZE,R0
          175536
002756 022700      COMP0:      CMP #2000,R0
          002000
002762 001404      BEQ COMP0A
002764 006300      ASL R0
002766 005267      INC HGTINC
          175552
002772 000771      BR COMP0
002774 005005      COMP0A:      CLR R5
002776 016503      MOV 140000(R5),R3
          140000
003002 013704      MOV #167774,R4
          167774
003006 005725      TST (R5)+
003010 016500      MOV 140000(R5),R0
          140000
003014 013701      MOV #167774,R1
          167774
003020 004767      JSR PC,ADDF
          000672
003024 005745      TST -(R5)
003026 010137      MOV R1,#167774
          167774
003032 010005      MOV R0,140000(R5)
          140000
003036 005000      CLR R0
003040 005725      TST (R5)+
003042 010057      MOV R0,#167774
          167774
003046 010065      MOV R0,140000(R5)
          140000
003052 012700      MOV #4,R0
          000004
003056 010067      MOV R0,INDEX
          175440
003062 010001      COMPL1:      MOV R0,R1
003064 062700      ADD #140000,R0
          140000
003070 010067      MOV R0,ADDR1R
          175460

```

```

003074 005720      TST (R0)+
003076 010067      MOV R0,ADDR11
      175454
003102 016700      MOV SIZE,R0
      175406
003106 006300      ASL R0
003110 160100      SUB R1,R0
003112 062700      ADD #140000,R0
      140000
003116 010067      MOV R0,ADDR2R
      175436
003122 026767      CMP ADDR1R,ADDR2R
      175426
      175430
003130 001001      BNE COMPL2
003132 000207      RTS PC
003134 005720      TST (R0)+
003136 010067      MOV R0,ADDR21
      175420
003142 017703      MOV @ADDR2R,R3
      175412
003146 013704      MOV @#167774,R4
      167774
003152 006203      ASR R3
003154 006004      ROR R4
003156 017700      MOV @ADDR1R,R0
      175372
003162 013701      MOV @#167774,R1
      167774
003166 006200      ASR R0
003170 006001      ROR R1
003172 004767      JSR PC,ADDF
      000520
003176 010067      MOV R0,REAL1
      175400
003202 010167      MOV R1,REAL1+2
      175376
003206 004767      JSR PC,SUBF
      000516
003212 004767      JSR PC,SUBF
      000512
003216 010067      MOV R0,REAL2
      175370
003222 010167      MOV R1,REAL2+2
      175366
003226 017703      MOV @ADDR21,R3
      175330
003232 013704      MOV @#167774,R4
      167774
003236 006203      ASR R3

```

COMPL2:

003240	006004		ROR R4
003242	017700		MOV #ADDR11,R0
	175310		
003246	013701		MOV #167774,R1
	167774		
003252	006200		ASR R0
003254	006001		ROR R1
003256	004767		JSR PC,ADDF
	000434		
003262	010067		MOV R0,IMAG1
	175320		
003266	010167		MOV R1,IMAG1+2
	175316		
003272	004767		JSR PC,SUBF
	000432		
003276	004767		JSR PC,SUBF
	000426		
003302	010067		MOV R0,IMAG2
	175310		
003306	010167		MOV R1,IMAG2+2
	175306		
003312	016700		MOV INDEX,R0
	175204		
003316	006200		ASR R0
003320	016701		MOV MGTINC,R1
	175220		
003324	005701	COMP2A:	TST R1
003326	003403		BLE COMP2B
003330	006300		ASL R0
003332	005301		DEC R1
003334	000773		BR COMP2A
003336	016004	COMP2B:	MOV \$CTABL(R0),R4
	004542		
003342	005005		CLR R5
003344	006004		ROR R4
003346	006005		ROR R5
003350	010467		MOV R4,COSH
	175152		
003354	010567		MOV R5,COSH+2
	175150		
003360	016702		MOV IMAG1,R2
	175222		
003364	016703		MOV IMAG1+2,R3
	175220		
003370	004767		JSR PC,MULT
	000352		
003374	010046		MOV R0,-(SP)
003376	010146		MOV R1,-(SP)
003400	016700		MOV INDEX,R0
	175116		

```

003404 006200      ASH R0
003406 016701      MOV WGTINC,R1
      175132
003412 005701 COMP2C: TST R1
003414 003403      BLE COMP2D
003416 006300      ASL R0
003420 005301      DEC R1
003422 000773      BR COMP2C
003424 162700 COMP2D: SUB #1000,R0
      001000
003430 005400      NEG R0
003432 016004      MOV SCTABL(R0),R4
      004542
003436 005005      CLR R5
003440 006004      ROR R4
003442 006005      ROR R5
003444 010467      MOV R4,SINH
      175062
003450 010567      MOV R5,SINH+2
      175060
003454 016702      MOV REAL2,R2
      175132
003460 016703      MOV REAL2+2,R3
      175130
003464 004767      JSR PC,MUL1
      000256
003470 012604      MOV (SP)+,R4
003472 012603      MOV (SP)+,R3
003474 004767      JSR PC,ADDF
      000216
003500 016703      MOV REAL1,R3
      175076
003504 016704      MOV REAL1+2,R4
      175074
003510 004767      JSR PC,ADDF
      000202
003514 010137      MOV R1,#167774
      167774
003520 010077      MOV R0,#ADDR1R
      175030
003524 004767      JSR PC,SUBF
      000200
003530 004767      JSR PC,SUBF
      000174
003534 005400      NEG R0
003536 005401      NEG R1
003540 005600      SRC R0
003542 010137      MOV R1,#167774
      167774
003546 010077      MOV R0,#ADDR2R
      175006

```

003552 016704	MOV COSW,R4
174750	
003556 016705	MOV COSW+2,R5
174746	
003562 016702	MOV REAL2,R2
175024	
003566 016703	MOV REAL2+2,R3
175022	
003572 004767	JSR PC,MULT
000150	
003576 010046	MOV R0,-(SP)
003600 010146	MOV R1,-(SP)
003602 016704	MOV SINW,R4
174724	
003606 016705	MOV SINW+2,R5
174722	
003612 016702	MOV IMAG1,R2
174770	
003616 016703	MOV IMAG1+2,R3
174766	
003622 004767	JSR PC,MULT
000120	
003626 012604	MOV (SP)+,R4
003630 012603	MOV (SP)+,R3
003632 004767	JSR PC,SUBF
000072	
003636 016703	MOV IMAG2,R3
174754	
003642 016704	MOV IMAG2+2,R4
174752	
003646 004767	JSR PC,ADDF
000044	
003652 010137	MOV R1,#167774
167774	
003656 010077	MOV R0,#ADDR1
174674	
003662 004767	JSR PC,SUBF
000042	
003666 004767	JSR PC,SUBF
000036	
003672 010137	MOV R1,#167774
167774	
003676 010077	MOV R0,#ADDR2
174660	
003702 005267	INC INDEX
174614	
003706 016700	MOV INDEX,R0
174610	
003712 000167	JMP COMPL
177144	

```

;-----
;-----
;-----SUBROUTINES ADDF AND SUBF-----
;-----
;-----
003716 000401 ADDF:      ADD R4,R1
003720 005500      ADC R0
003722 000300      ADD R3,R0
003724 102406      BVS ERROR1
003726 000207      RTS PC
003730 100401 SUBF:      SUB R4,R1
003732 005600      SBC R0
003734 100300      SUB R3,R0
003736 102401      BVS ERROR1
003740 000207      RTS PC
003742 000167 ERROR1:    JMP ERROR
174032

;-----
;-----
;-----SUBROUTINE MULT-----
;-----
;-----
003746 005046 MULT1:    CLR -(SP)
003750 005702      TST R2
003752 002004      BGE MULT1
003754 005402      NEG R2
003756 005403      NEG R3
003760 005602      SBC R2
003762 005216      INC @SP
003764 005704 MULT1:    TST R4
003766 002004      BGE MULT2
003770 005404      NEG R4
003772 005405      NEG R5
003774 005604      SBC R4
003776 005316      DEC @SP
004000 005000 MULT2:    CLR R0
004002 005001      CLR R1
004004 012746      MOV #22,-(SP)
000022
004010 006103 MULT3:    ROL R3
004012 006102      ROL R2
004014 103003      BCC MULT4
004016 000501      ADD R5,R1
004020 005500      ADC R0
004022 000400      ADD R4,R0
004024 006004 MULT4:    ROR R4
004026 006005      ROR R5
004030 005316      DEC @SP
004032 001306      BNE MULT3
004034 005726      TST (SP)+
004036 005726      TST (SP)+
004040 001403      BEQ MULT5
004042 005400      NEG R0
004044 005401      NEG R1
004046 005600      SBC R0
004050 000207 MULT5:    RTS PC

;-----
;-----
;-----SUBROUTINE MAGNTD-----
;-----
;-----
004052 005046 MAGNTD:   CLR -(SP)
004054 005005      CLR R5
004056 016502 MAG1:    MOV 140000(R5),R2
140000

```

```

004062 013703      MOV 0#167774,R3
          167774
004066 010204      MOV R2,R4
004070 010305      MOV R3,R5
004072 004767      JSR PC,MULT
          177650
004076 011605      MOV @SP,R5
004100 010046      MOV R0,~(SP)
004102 010146      MOV R1,~(SP)
004104 066705      ADD SIZE,R5
          174404
004110 016502      MOV 140000(R5),R2
          140000
004114 013703      MOV 0#167774,R3
          167774
004120 010204      MOV R2,R4
004122 010305      MOV R3,R5
004124 004767      JSR PC,MULT
          177616
004130 012604      MOV (SP)+,R4
004132 012603      MOV (SP)+,R3
004134 004767      JSR PC,ADDF
          177556
004140 004767      JSR PC,SQROOT
          000040
004144 011605      MOV @SP,R5
004146 066705      ADD SIZE,R5
          174342
004152 010137      MOV R1,0#167774
          167774
004156 010065      MOV R0,140000(R5)
          140000
004162 011605      MOV @SP,R5
004164 005725      TST (R5)+
004166 026705      CMP SIZE,R5
          174322
004172 001402      BEQ MAG2
004174 010516      MOV R5,@SP
004176 000727      BR MAG1
004200 005726      MAG2: TST (SP)+
004202 000207      RTS PC
          ;-----
          ;-----
          ;-----SUBROUTINE SQROOT-----
          ;-----
          ;-----
004204 005700      SQROOT: TST R0
004206 003442      BLE SQROT4
004210 005046      CLR ~(SP)
004212 005216      SQROT1: INC @SP
004214 006301      ASL R1
004216 006100      ROL R0
004220 100374      BPL SQROT1
004222 000241      CLC
004224 006000      ROR R0

```



```

004226 006001      ROR R1
004230 005316      DEC @SP
004232 010002      MOV R0,R2
004234 010103      MOV R1,R3
004236 004767      JSR PC,SQRT-
          000864
004242 011602      MOV @SP,R2
004244 006202      ASR R2
004246 103012      BCC SQROT2
004250 010002      MOV R0,R2
004252 010103      MOV R1,R3
004254 012704      MOV #055202,R4
          055202
004260 012705      MOV #074631,R5
          074631
004264 004767      JSR PC,MULT
          177456
004270 011602      MOV @SP,R2
004272 006202      ASR R2
004274 005702      SQROT2:  TST R2
004276 001404      BEQ SQROT3
004300 006200      ASR R0
004302 006001      ROR R1
004304 005302      DEC R2
004306 000772      BR SQROT2
004310 005726      SQROT3:  TST (SP)+
004312 000207      RTS PC
004314 002403      SQROT4:  BLT SQROT5
004316 005000      CLR R0
004320 005001      CLR R1
004322 000207      RTS PC
004324 000000      SQROT5:  HALT
;-----
;-----
;-----SUBROUTINE SQRT-----
;-----
;-----
004326 012746      SQRT1:  MOV #4,-(SP)
          000004
004332 010204      MOV R2,R4
004334 010305      MOV R3,R5
004336 010246      MOV R2,-(SP)
004340 010346      MOV R3,-(SP)
004342 002704      AOD #100000,R4
          100000
004346 010446      MOV R4,-(SP)
004350 010546      MOV R5,-(SP)
004352 006204      SQRT1:  ASR R4
004354 006005      ROR R5
004356 005404      NEG R4
004360 005405      NEG R5
004362 005604      SBC R4
004364 004767      JSR PC,MULT
          177356
004370 016603      MOV 6(SP),R3
          000006

```

004374	016604		MOV 4(SP),R4
	000004		
004400	004767		JSR PC,ADDF
	177312		
004404	005366		DEC 10(SP)
	000010		
004410	001003		BNE SORT2
004412	062706		ADD #12,SP
	000012		
004416	000207		RTS PC
004420	010066	SORT2:	MOV R0,6(SP)
	000006		
004424	010166		MOV R1,4(SP)
	000004		
004430	011603		MOV @SP,R3
004432	016602		MOV 2(SP),R2
	000002		
004436	010204		MOV R2,R4
004440	010305		MOV R3,R5
004442	004767		JSR PC,MULT
	177300		
004446	016603		MOV 2(SP),R3
	000002		
004452	011604		MOV @SP,R4
004454	010046		MOV R0,-(SP)
004456	010146		MOV R1,-(SP)
004460	006203		ASH R3
004462	006004		ROR R4
004464	006203		ASH R3
004466	006004		ROR R4
004470	012700		MOV #120000,R0
	120000		
004474	005001		CLR R1
004476	004767		JSR PC,ADDF
	177214		
004502	010002		MOV R0,R2
004504	010103		MOV R1,R3
004506	012605		MOV (SP)+,R5
004510	012604		MOV (SP)+,R4
004512	004767		JSR PC,MULT
	177230		
004516	010066		MOV R0,2(SP)
	000002		
004522	010116		MOV R1,@SP
004524	010004		MOV R0,R4
004526	010105		MOV R1,R5
004530	016602		MOV 6(SP),R2
	000006		
004534	016603		MOV 4(SP),R3
	000004		

004540 000704

OR SORT1

```

////////////////////////
////////////////////////
//////////////////////// SINE-COSINE TABLE ///////////////////
////////////////////////
////////////////////////
SCTABL1.WORD 177777,177776,177773,177764

```

004542 177777

004544 177776

004546 177773

004550 177764

004552 177754

.WORD 177754,177741,177723,177703

004554 177741

004556 177723

004560 177703

004562 177661

.WORD 177661,177634,177604,177552

004564 177634

004566 177604

004570 177552

004572 177516

.WORD 177516,177457,177416,177352

004574 177457

004576 177416

004600 177352

004602 177304

.WORD 177304,177233,177160,177103

004604 177233

004606 177160

004610 177103

004612 177023

.WORD 177023,176740,176653,176564

004614 176740

004616 176653

004620 176564

004622 176472

.WORD 176472,176376,176277,176176

004624 176376

004626 176277

004630 176176

004632 176073

.WORD 176073,175765,175654,175541

004634 175765

004636 175654

004640 175541

004642 175424

.WORD 175424,175305,175163,175036

004644 175305

004646 175163

004650 175036

004652 174707

.WORD 174707,174556,174422,174264

004654 174556

004656 174422

004660 174264

004662 174123

.WORD 174123,173761,173613,173444

004664 173761

004666 173613

004670 173444

004672 173272

.WORD 173272,173115,172736,172555

004674 173115

004676 172736

004700 172555

004702 172372

.WORD 172372,172204,172013,171621

004704	172204	
004706	172013	
004710	171621	
004712	171424	.WORD 171424,171224,171023,170617
004714	171224	
004716	171023	
004720	170617	
004722	170411	.WORD 170411,170200,167765,167550
004724	170200	
004726	167765	
004730	167550	
004732	167330	.WORD 167330,167106,166662,166434
004734	167106	
004736	166662	
004740	166434	
004742	166203	.WORD 166203,165750,165513,165253
004744	165750	
004746	165513	
004750	165253	
004752	165011	.WORD 165011,164545,164277,164026
004754	164545	
004756	164277	
004760	164026	
004762	163553	.WORD 163553,163276,163017,162536
004764	163276	
004766	163017	
004770	162536	
004772	162252	.WORD 162252,161764,161474,161202
004774	161764	
004776	161474	
005000	161202	
005002	160705	.WORD 160705,160406,160106,157603
005004	160406	
005006	160106	
005010	157603	
005012	157276	.WORD 157276,156766,156455,156141
005014	156766	
005016	156455	
005020	156141	
005022	155624	.WORD 155624,155304,154762,154436
005024	155304	
005026	154762	
005030	154436	
005032	154110	.WORD 154110,153560,153225,152671
005034	153560	
005036	153225	
005040	152671	
005042	152333	.WORD 152333,151772,151430,151063
005044	151772	
005046	151430	

005050	151063	
005052	150515	.WORD 150515,150144,147572,147215
005054	150144	
005056	147572	
005060	147215	
005062	146637	.WORD 146637,146256,145673,145307
005064	146256	
005066	145673	
005070	145307	
005072	144721	.WORD 144721,144330,143736,143342
005074	144330	
005076	143736	
005100	143342	
005102	142744	.WORD 142744,142343,141742,141336
005104	142343	
005106	141742	
005110	141336	
005112	140730	.WORD 140730,140320,137707,137274
005114	140320	
005116	137707	
005120	137274	
005122	136656	.WORD 136656,136240,135617,135174
005124	136240	
005126	135617	
005130	135174	
005132	134550	.WORD 134550,134122,133472,133040
005134	134122	
005136	133472	
005140	133040	
005142	132404	.WORD 132404,131747,131310,130650
005144	131747	
005146	131310	
005150	130650	
005152	130205	.WORD 130205,127541,127073,126424
005154	127541	
005156	127073	
005160	126424	
005162	125753	.WORD 125753,125300,124624,124146
005164	125300	
005166	124624	
005170	124146	
005172	123466	.WORD 123466,123005,122322,121635
005174	123005	
005176	122322	
005200	121635	
005202	121147	.WORD 121147,120457,117766,117274
005204	120457	
005206	117766	
005210	117274	
005212	116577	.WORD 116577,116102,115402,114702

005214	116102	
005216	115402	
005220	114702	
005222	114177	.WORD 114177,113474,112766,112260
005224	113474	
005226	112766	
005230	112260	
005232	111550	.WORD 111550,111036,110323,107607
005234	111036	
005236	110323	
005240	107607	
005242	107071	.WORD 107071,106352,105632,105110
005244	106352	
005246	105632	
005250	105110	
005252	104365	.WORD 104365,103641,103113,102364
005254	103641	
005256	103113	
005260	102364	
005262	101634	.WORD 101634,101102,100347,77613
005264	101102	
005266	100347	
005270	077613	
005272	077056	.WORD 77056,76320,75560,75017
005274	076320	
005276	075560	
005300	075017	
005302	074255	.WORD 74255,73512,72745,72200
005304	073512	
005306	072745	
005310	072200	
005312	071431	.WORD 71431,70661,70111,67337
005314	070661	
005316	070111	
005320	067337	
005322	066564	.WORD 66564,66010,65233,64455
005324	066010	
005326	065233	
005330	064455	
005332	063675	.WORD 63675,63115,62334,61552
005334	063115	
005336	062334	
005340	061552	
005342	060767	.WORD 60767,60203,57416,56630
005344	060203	
005346	057416	
005350	056630	
005352	056042	.WORD 56042,55252,54461,53670
005354	055252	
005356	054461	

005360	053670	
005362	053070	.WORD 53076,52303,51507,50712
005364	052303	
005366	051507	
005370	050712	
005372	050115	.WORD 50115,47317,46520,45720
005374	047317	
005376	046520	
005400	045720	
005402	045120	.WORD 45120,44316,43515,42712
005404	044316	
005406	043515	
005410	042712	
005412	042107	.WORD 42107,41303,40476,37671
005414	041303	
005416	040476	
005420	037671	
005422	037063	.WORD 37063,36255,35446,34637
005424	036255	
005426	035446	
005430	034637	
005432	034027	.WORD 34027,33216,32405,31573
005434	033216	
005436	032405	
005440	031573	
005442	030761	.WORD 30761,30146,27333,26520
005444	030146	
005446	027333	
005450	026520	
005452	025704	.WORD 25704,25067,24252,23435
005454	025067	
005456	024252	
005460	023435	
005462	022620	.WORD 22620,22002,21163,20345
005464	022002	
005466	021163	
005470	020345	
005472	017526	.WORD 17526,16707,16067,15247
005474	016707	
005476	016067	
005500	015247	
005502	014427	.WORD 14427,13607,12766,12146
005504	013607	
005506	012766	
005510	012146	
005512	011325	.WORD 11325,10504,7662,7041
005514	010504	
005516	007662	
005520	007041	
005522	006217	.WORD 6217,5376,4554,3732

005524 005376
005526 004554
005530 003732
005532 003110
005534 002266
005536 001444
005540 000622
005542 000000
000001

.WORD 3110,2266,1444,622

.WORD 0

.END

ACKNOWLEDGMENTS

We would like to acknowledge several stimulating conversations with Professor R. J. Higgins. These conversations helped considerably in clarifying our understanding of the relationship between the infinite Fourier integral transform and the finite Fourier series transform. The help of the staff of the Nicolet Instrument Corporation who related their experiences in programming FFT on minicomputers and helped with the software problems relating to the Model 1070 interface is also gratefully acknowledged. R. L. Aurbach, in using a preliminary version of this program, has helped in finding several coding errors and has made several valuable suggestions for modifying our codes for the relocatable assembler, PAL-11S.

REFERENCES

1. J. W. Cooley and J. W. Tukey, Math. of Computation 19, 297 (1965).
2. Digital Equipment Corporation, Maynard, Massachusetts.
3. Nicolet Instruments Corporation (Formerly Fabri-Tek Instruments Corp.), Madison, Wisconsin.
4. J. W. Cooley, P. A. W. Lewis, and P. D. Welch, IEEE Trans. on Audio and Electroacoustics, Vol. AU-15, No. 2, 79 (1967).
5. R. J. Higgins, private communication.
6. C. H. Costain, Decuscope Vol. 11, No. 2, 25 (1972).
7. W. T. Cochran, et al., IEEE Trans. on Audio and Electroacoustics, Vol. AU-15, No. 2, 45 (1967).
8. During the time this manuscript was being prepared, a relocating assembler PAL-11S was released by Digital Equipment Corporation. Versions of the programs described here are now being prepared using this assembler and these will be supplied upon request.