1973

Mini-computer oriented high-speed transform algorithms.

Charles A. Tam
University of Windsor

MINI-COMPUTER ORIENTED HIGH-SPEED TRANSFORM ALGORITHMS

by

CHARLES A. TAM

A Thesis

Submitted to the Faculty of Graduate Studies through the Department of Electrical Engineering in Partial Fulfillment of the Requirement for the Degree of Master of Applied Science at the University of Windsor

Windsor, Ontario

1973

ABSTRACT

This thesis investigates the use of Discrete Transform Algorithms for the representation of signals, for computing convolution and finally for efficient implementation on a mini-computer.

The Transforms to be considered are the Fast Walsh Transform and the Fast Fourier Transform.

From this investigation, a modified method for computing the Fast Fourier Transform of Real Data based on Bergland(1) is developed.

ACKNOWLEDGEMENT

The author wishes to express his appreciation to Dr. G. A. Jullien for his supervision and suggestions. Acknowledgement also goes to the other members of the faculty who have taken an interest to accomplish this project.

Special thanks go to the author's wife, Debbie A. Tam, and mother, Mrs. Sylvia Tam, who have given their support and encouragement at all times.

TABLE OF CONTENTS

NOTATION

Type	Meaning
$((x))_N$	x Modulo N = x+rN $0 \leq x+rN < N$, $r \in \{0,-1,-2,\ldots\}$
$\underline{P}$	An Operator
DWT	Discrete Walsh Transform
DFT	Discrete Fourier Transform
FFT	Fast Fourier Transform
FFTRVI	Fast Fourier Transform for Real Valued Input
FFTBRRVI	Fast Fourier Transform for Bit-Reversed Real Valued Input
M. S. E.	Mean Square Error
S/P	Single Precision
D/P	Double Precision
A/D	Analog-to-Digital
D/A	Digital-to-Analog

CHAPTER I

INTRODUCTION

In linear systems, the relationship between the driving function and the response can be expressed by the convolution integral.

In communication theory, important theorems such as the modulation theorem and the sampling theorem can be viewed as special cases of convolution.

The correlation functions (auto-correlation and cross-correlation) which occur in signal detection theory and in the study of random noise are also a form of convolution.

Convolution is therefore, one of the most important tools in the analysis of both systems and signals, and forms the basis of our investigation.

Orthogonal transforms will be used for the represent--ation of signals. Part of the work is devoted to investig--ating methods of computing these transforms. The properties of these transforms will then be considered with a view to computing convolution.

This chapter is devoted to reviewing some basic theory with regard to continuous and discrete analysis. The topics will include firstly, some definitions, secondly, a brief description of a signal processing system, thirdly, the use of complete orthogonal sets of functions for the represent--ation of signals, and finally, methods of computing convolution using these orthogonal transforms.

1.1 Definitions

"Analog" generally means a waveform that is continuous in time (or any other appropriate independent variable) and that belongs to a class that can take on a continuous range of amplitude values. Eg. sin wt.

"Continuous-time" implies that only the independent variable necessarily takes on a continuous range of values. Therefore analog waveforms are continuous waveforms with continuous amplitude. In practice, "continuous-time" waveforms and "analog" waveforms are equivalent.

"Discrete-time" implies that time (the independent variable) is quantized. i.e. Discrete-time signals are defined only for discrete values of the independent variable. Such signals are represented mathematically as sequences of numbers.

"Digital" implies that both time and amplitude are quantized. Thus a "digital" system is one in which signals are represented as sequences of numbers which take on a finite set of values. Thus discrete-time signal processing systems are called "digital" systems.

A "digital" signal or "digital" waveform is a sequence produced by digital circuitry or by an analog-to-digital converter which is sampling a continuous-time waveform. In digital signal processing, these terms are commonly shortened to "signals" and "waveform".

"Truncation" is accomplished by discarding all bits (or digits) less significant than the least significant bit (or digit) which is retained.

"Rounding-off" a number to "b" bits; when the number is initially specified to more than "b" bits, is accomplished by choosing the rounded result as the "b"-bit number closest to the original unrounded quantity.

"The Fourier Transform", or spectrum, of a signal f(t) is defined as

$$F(\omega) = \int_{-\infty}^{\infty} f(t)\, e^{-j\omega t}\, dt$$

"The Inverse Fourier Transform" is given by

$$f(t) = 1/2\pi \int_{-\infty}^{\infty} F(\omega)\, e^{j\omega t}\, d\omega$$

1.2 Description of a digital Signal Processing System.

A digital computer is a desirable unit in any system where data processing is done. For signal processing the computer must have certain features since there exists a great need for close communication between operator and the computer. The devices which have these features are known as peripheral devices and include such equipment as computer-controlled oscilloscope, analog -to- digital and digital-to-analog converters, pulse interrupts, toggle switches, multiplexer and demultiplexer, etc. All these devices make on-line computer operation more meaningful.

A model of a system where processing is done digitally but the input and output are continuous signals is shown in fig. 1.

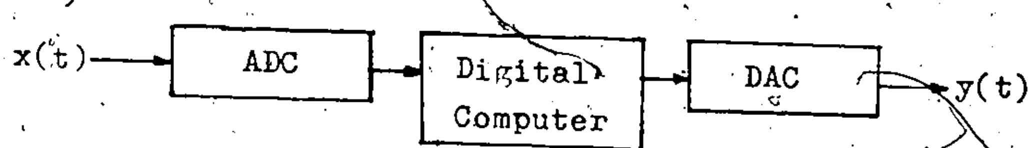


Fig. 1. General Purpose Signal Processing system using a Digital Computer.

The Analog-to-Digital (A/D) converter is a device which operates on a continuous-time waveform to produce a digital output consisting of a sequence of numbers each of which approximates a corresponding sample of the input waveform.

The components that comprise the Analog-to-Digital converter are as follows:

(1) Sampler: The analog signal is sampled at uniform time intervals T to produce the sequence x(nT).

$$S\left[x(t)\right] = 1/2\epsilon \int_{kT-\epsilon}^{kT+\epsilon} x(t)\,dt$$

Sampling Theorem.

A bandlimited signal which has no spectral components above a frequency f_m cycles per second is uniquely determ- -ined by its values at uniform intervals less than $1/2f_m$ seconds apart.

This theorem can be proved by multiplying the band-limited signal with a periodic impulse function, then passing the sampled signal through a low pass filter which permits the transmission of all frequency components below f_m and attenuates all frequency components above f_m. The Nyquist rate equals $2f_m$ and is the lowest rate at which f(t) can be sampled and still recovered.

(2) Quantizer: This is the process of representing the signal by certain discrete amplitude levels.

The level of representation depends on

i. Limit on acceptable voltage.

ii. The number of quantization levels.

iii. Truncation or Rounding-off.

(3) Encoder : This is the process of representing the quantized levels by numbers.

The Digital-to-Analog (D/A) converter is a device which operates on a digital input signal s(nT) to produce a continous-time output signal s(t) ideally defined by

$$s(t) = \sum_n s(nT)\, h(t-nT)$$

where h(t) characterizes the particular converter. For example h(t) is a square pulse of duration T for a zero order hold D/A converter.

The D/A converter is usually followed by a linear time-invariant low-pass continuous-time filter called a postfilter. The combination of D/A converter and postfilter is called a reconstruction device or a reconstruction filter.

The Components that comprisethe Digital-to-Analog Converter are

1) Decoder: This is the process of representing the numbers as voltages. This is the inverse of the encoding process.

2) Extrapolator: This is the process in which a continuous signal is reconstructed from the sampled data.

This process is not perfectly possible because of

i. Prediction

ii. Sampling Theorem is not reversible in practice.

iii. Band Limited System required.

1.3 Continuous-Discrete

A linear time-invariant continuous system can be characterized by linear differential equations with constant coefficients. On the other hand, a linear time--invariant discrete system is characterized by linear difference equations with constant coefficients which can be realized by manipulating numbers on a general purpose computer.

A continuous signal can be converted to a discrete signal by sampling the continuous signal and converting the samples to numerical values. i.e., in a discrete system the data signal consists of a sequence of pulses which are modulated in accordance with the continuous signal from which samples are taken.

The input to the system and output of the system are continuous signals. However, since the input to the digital computer and the output from the digital computer are discrete sequences, the analysis will be directed towards the discrete approach. viz., the discrete algorithms will be used for performing the signal process--ing operations.

1.4 Description of signals via orthogonal functions.

In studying and analyzing systems, we have to deal with signals or time functions. Although a signal is defined directly as a function of time, this representation is not always adequate for our purposes. Hence, we need to become familiar with other ways of describing time functions so that we may more easily analyze the behaviour of systems.

1.4.1 Approximation of a time function by another set of functions.

We first consider the problem of describing a time function f(t) on an interval (0,T). We would like to describe this function by specifying a discrete set of coefficients. Therefore, we consider a series expansion of the form

$$f(t) = C_1g_1(t) + C_2g_2(t) + \dots\dots\dots\dots + C_Ng_N(t)$$

$$= \sum_{i=1}^{N} C_ig_i(t) \qquad \dots\dots\dots\dots 1.1$$

where

(i) The N coefficients C_i depend only on the function f(t) to be represented and not on time.

(ii) The N functions of time, $g_i(t)$, are specified independently of f(t).

1.4.2 Minimization using M. S. E. criterion.

The error obtained by representing f(t) by the series expansion is given by

$$f_e(t) = f(t) - \sum_{i=1}^{N} C_i g_i(t) \qquad \ldots\ldots\ldots\ldots 1.2$$

The m.s.e. is given by

$$E = 1/T \int_0^T f_e^2(t)dt = 1/T \int_0^T \left[f(t) - \sum_{i=1}^{N} C_i g_i(t) \right]^2 dt \qquad \ldots\ldots\ldots\ldots 1.3$$

Case 1. If $g_i(t)$ are an arbitrary set of functions.

To minimize E, we make use of the following:

$$\frac{\partial E}{\partial C_j} = 0 \qquad \text{for } j = 1,2,\ldots\ldots,N \qquad \ldots\ldots 1.4$$

Applying these conditions to equation 1.3 we shall obtain a set of N simultaneous equations which we have to solve in order to find the N values of the C_j.

To overcome this problem of solving N simultaneous equations we choose functions which have certain properties.

Case 2. If $g_i(t)$ are a set of orthogonal functions.

We say that the functions $g_i(t)$, i = 1,2,.........,N are orthogonal in the interval (0,T) if

$$\int_0^T g_j(t) g_i(t) dt = 0 \quad \text{if } j \neq i \qquad ....1.5$$
$$= K_j \quad \text{if } j = i$$

Using this property, from equations 1.3 and 1.5 we obtain the simple relationship for the C_j viz.,

$$C_j = 1/K_j \int_0^T f(t)\, g_j(t) dt \qquad ....1.6$$

for j = 1, 2, 3.........N

The above equation is independent of N, the number of terms used in the representation. This is due solely to the orthogonality of $g_i(t)$.

1.4.3 Representation of a function by a closed or complete set of orthogonal functions.

If we increase N, i.e., if we approximate f(t) by a larger number of terms, the error should become smaller. Hence defining $_n$ as

$$\Xi_N = 1/T \left(\int_0^T f^2(t)dt - \sum_{i=1}^{N} C_i{}^2 K_i \right) \qquad \ldots 1.8$$

If $\quad \underset{N\to\infty}{Lt}\, \Xi_N = 0$

We say that $\{g_i(t)\}$ is complete on the interval (0,T).

Then

$$\int_0^T f^2(t)dt = \sum_{i=1}^{\infty} C_i{}^2 K_i \qquad \ldots 1.9$$

Under these conditions, f(t) can be represented by the infinite series as follows:

$$f(t) = C_1 g_1(t) + C_2 g_2(t) + \ldots\ldots + C_i g_i(t) + \ldots\ldots$$

$$= \sum_{i=1}^{\infty} C_i g_i(t) \qquad \ldots\ldots\ldots 1.10$$

The reason for using a complete set of orthogonal functions is that we can limit our error by choosing N to be sufficiently large.

There exists a large number of sets of orthogonal functions and hence, a given function may be expressed in terms of different sets of orthogonal functions. Examples of sets of orthogonal functions are trigonometric functions, exponential functions, Legendre polynomials and Jacobi polynomials.(6)

1.5 Convolution.

The input x(t) and the output y(t) of a continuous-time system are related by the convolution integral given by

$$y(t) = \int_{-\infty}^{\infty} x(\tau)h(t-\tau)d\tau$$

where h(t) is the impulse response of the system.

In a discrete system, if x(nT) and y(nT) are the discrete input and output sequences, then the discrete convolution summation is given by

$$y(nT) = \sum_{k=-\infty}^{\infty} w(nT-kT)x(kT)$$

$$= \sum_{k=-\infty}^{\infty} w(kT)x(nT-kT)$$

or in shortened notation as

$$y_n = \sum_{k=-\infty}^{\infty} w(n-k)x(k) = \sum_{k=-\infty}^{\infty} w(k)x(n-k)$$

where the weighting function $w_n = w(nT) = w(t)\big|_{t=nT}$

1.5.1 Periodic Convolution

Periodic or circular convolution is one in which values of the kernel that are shifted from one end of a period are circulated into the other end, thereby introdu--cing what is called interperiod interference.

Consider fig. 2 which shows a 5-point sequence x(nT) circularly convolved with another 5-point sequence y(nT).

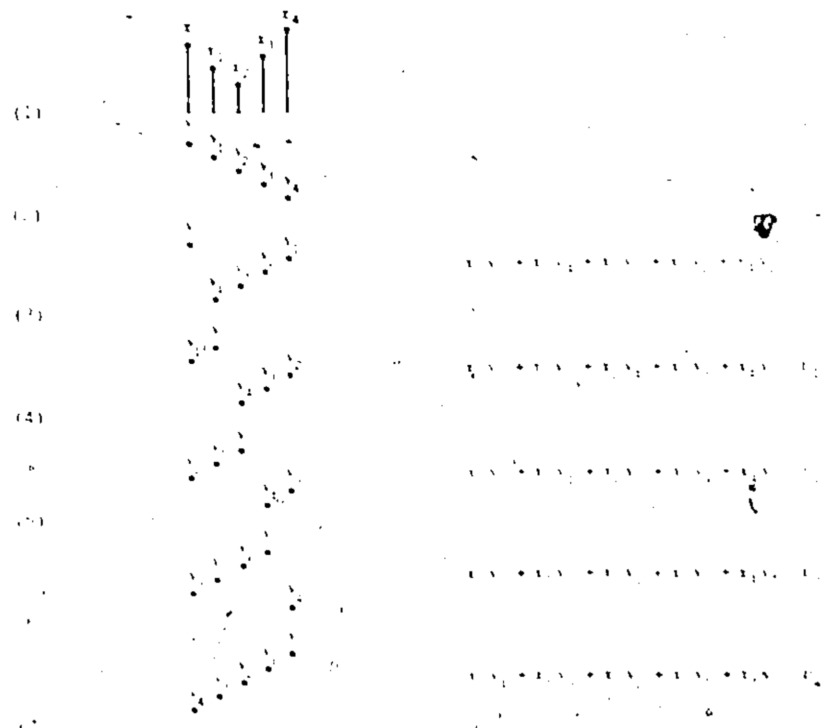


Fig. 2 [illegible]

Line 1 shows the x sequence which is kept fixed.

Line 2 shows the y sequence.

Line 3 shows the position after one circular shift. Note that the y sequence has been reversed prior to the circular shift. The convolution sum v_0 is shown.

Lines 4,5,6,7 show how the reversed y sequence is shifted for the computation of v_1, v_2, v_3 and v_4.

1.5.2 Aperiodic Convolution

In most applications, aperiodic convolution is desired. A periodic convolution can be made to give results numerically identical to those of an aperiodic one by inserting an appropriate number of zero-valued samples to each of the component functions depending on how much of the periodic convolution is to be rendered into an aperiodic equivalent.

1.6 Summary.

Some definitions have been presented. A description of a signal processing system is given. The components of an A/D and D/A converters are mentioned very briefly. Representation of a signal by orthogonal functions was shown to reduce the data to a set of discrete coefficients or spectrum numbers. Finally, the method of obtaining the periodic convolution of two sequences was given along with a method for obtaining partial aperiodic convolution results from the periodic algorithm.

CHAPTER II

WALSH TRANSFORM AND CONVOLUTION.

Introduction

The orthogonal time series representation was shown to reduce the input data to a discrete set of coefficients or spectrum numbers.

For signal processing applications, the only theoretical requirement for selecting an orthogonal set of functions is that of completeness. The Fourier set has been used quite extensively because of its easy interpretation in physical terms (amplitude and phase) and its close relationship with R-L-C circuitry.

Within recent times, however, another complete orthonormal set of functions known as the Walsh functions has become quite popular. This is so because the Walsh functions are a set of two-valued functions, (either 1 or -1), inherently suited to high speed computation in two-state digital computers, and thus, the Walsh transform can be computed very quickly as operations consist of additions and subtractions only.

To this end, many researchers are finding this transform very useful in digital signal processing applications(2). One such application by Pitassi(3) will be investigated.

2.1. Walsh Functions.

The Discrete Walsh Functions can be described in terms of the multiplicative iterative equations given by

$$wal(0,N) = 1 \quad \text{for } N = 0,1,\ldots\ldots,M-1 \qquad \ldots 2.1$$

$$wal(1,N) = \begin{cases} 1 & \text{for } N = 0,1,\ldots\ldots,[M/2]-1 \\ -1 & \text{for } N = [M/2],\ldots\ldots,M-1 \end{cases} \qquad \ldots 2.2$$

$$wal(M,N) = wal([M/2],2N).\ wal(M-2[M/2],N) \qquad \ldots 2.3$$

where $[M/2]$ denotes the integer part of M/2.

Figure 3. shows the first eight discrete Walsh functions of length M=8.

Fig. 3 The eight discrete Walsh functions of length 8.

Note - The Walsh Functions generated from the equations 2.1, 2.2, and 2.3 are symmetric with respect to the argument (M,N) i.e.,

$$wal(M,N) = wal(N,M)$$

2.2. The Discrete Walsh Transform.

The Discrete Walsh Transform of an M-length array f(N) is defined as

$$F(M) = \sum_{N=0}^{M-1} f(N)\, wal(M,N)$$

$$\text{for } M = 0,1,2,\ldots\ldots,M-1 \qquad ..2.4$$

The Inverse Discrete Walsh Transform is given by

$$f(N) = 1/M \sum_{M=0}^{M-1} F(M)\, wal(N,M)$$

$$\text{for } N = 0,1,2,\ldots\ldots\ldots,M-1 \qquad ..2.5$$

Figure 4 shows the signal flow graph of an 8-length discrete Walsh transform. The A_0 values represent the input sequence and the A_3 values represent the output sequence or the Walsh transform coefficients. The A_1 and A_2 values are the results at the intermediate stages.

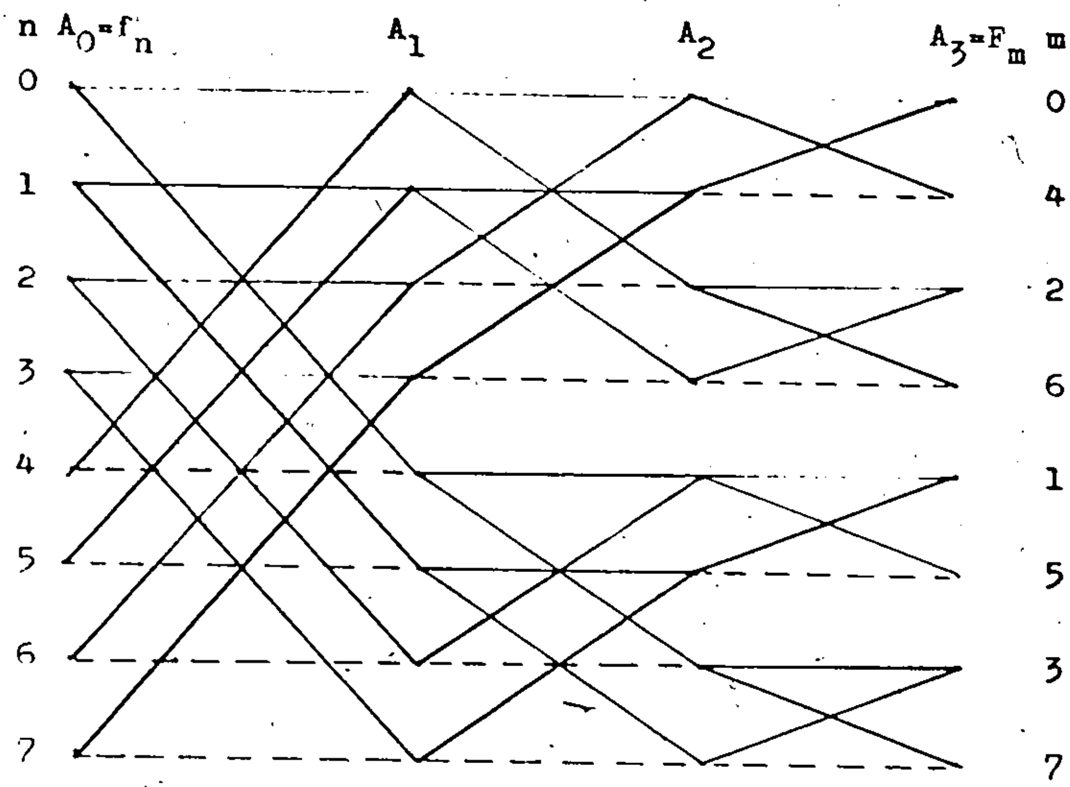


Fig. 4. Signal flow graph of 8-length discrete Walsh transforms. Multiplying factors are +1 and -1, indicated by solid and dotted lines.

The following observations should be noted.

(i)The number of iterations equals g where $g=\log_2 N$.

(ii)Half the members in each group are associated with an addition and the other half a subtraction.

(iii)The total number of operations to compute the transform is $N\log_2 N$.

(iv)The same algorithm can be used to compute the inverse transform except that the results will have a scaling factor of N.

2.3. Fast Convolution using the Walsh Transform.

The following is taken from a research paper by Pitassi(3) entitled "Fast Convolution using the Walsh Transform". The algorithm will be developed, then ways of implementing it will be discussed.

2.3.1. Definitions.

A finite time series $X_0, X_1, \ldots\ldots, X_{N-1}$ will be represented by a column vector as

$$X = (X_0, X_1, \ldots\ldots\ldots, X_{N-1})^T \qquad \ldots 2.6$$

A one element circular shift of the time series will be denoted by

$$X' = (X_1, X_2, \ldots\ldots\ldots, X_{N-1}, X_0)^T \qquad \ldots 2.7$$

The cyclic or periodic convolution of two time series x and y, each of length N, is also a time series of length N denoted by the column vector r where

$$r_n = \sum_{m=0}^{N-1} x_{((m))} y_{((m+n))} = x * y \qquad \ldots 2.8$$

for $n = 0, 1, \ldots\ldots\ldots N-1$

where $*$ represents cyclic or periodic convolution.

A set of operators $\underline{E}$, $\underline{O}$, $\underline{P}$ and $\underline{S}$ which transform a vector of length N to another of length N/2 are defined as,

$$\underline{E}x = (X_0, X_2, X_4, \ldots\ldots\ldots, X_{N-2})^T \qquad \ldots 2.9$$

$$\underline{O}x = (X_1, X_3, X_5, \ldots\ldots\ldots, X_{N-1})^T \qquad \ldots 2.10$$

$$\underline{P}x = (\underline{E}+\underline{O})\, x \qquad \ldots 2.11$$

$$\underline{S}x = (\underline{E}-\underline{O})\, x \qquad \ldots 2.12$$

2.3.2. Development of algorithm.

Starting with the definition of cyclic convolution

$$r_n = \sum_{m=0}^{N-1} x_{((m))}\, y_{((m+n))} \qquad n=0,1,\ldots\ldots,N-1$$

$$= \sum_{m=0}^{N/2\,-1} x_{((2m))}\, y_{((2m+n))} + \sum_{m=0}^{N/2\,-1} x_{((2m+1))} y_{((2m+1+n))}$$

Splitting into even and odd components of r and expressing in terms of the $\underline{E}$ and $\underline{O}$ operators we get

$$\underline{E}r = \underline{E}x * \underline{E}\, y + \underline{O}x * \underline{O}y$$

$$\underline{O}r = \underline{E}x * \underline{O}y + \underline{O}x * \underline{E}y'$$

Consider the auxiliary convolution functions c,d, and f defined as follows

$$c = \underline{P}x * \underline{P}y = (\underline{E}+\underline{O})x * (\underline{E}+\underline{O})y$$

$$d = \underline{S}x * \underline{S}y = (\underline{E}-\underline{O})x * (\underline{E}-\underline{O})y$$

$$f = (\underline{P}-\underline{S})x * \underline{E}\,y = 2\,\underline{O}x * \underline{E}y$$

From these we get

$$c+d = 2(\underline{E}x * \underline{E}y + \underline{O}x * \underline{O}y) = 2\,\underline{E}r$$

$$c-d+f'-f = 2(\underline{E}x * \underline{O}y + \underline{O}x * \underline{E}y) = 2\,\underline{O}r$$

Hence the algorithm is defined as follows:

(i) Perform the operations $\underline{P}$, $\underline{S}$ and $\underline{O}$ on x-vector and $\underline{P}$, $\underline{S}$ and $\underline{E}$ on y-vector.

(ii) Form the auxiliary convolution functions c,d and f defined as

$$c = \underline{P}x * \underline{P}y \qquad d = \underline{S}x * \underline{S}y \qquad f = 2\,\underline{O}x * \underline{E}y$$

(iii) The even and odd components of r are then given by

$$2\,\underline{E}r = c+d$$

$$2\,\underline{O}r = c-d+f'-f$$

Many different forms of the algorithm can be derived by defining different auxiliary convolutions. eg.,

If

$$c = \underline{P}x * \underline{P}y \qquad d = \underline{O}x * \underline{E}y \qquad f = \underline{E}x * \underline{O}y$$

Then

$$\underline{E}r = c-d-f \qquad \text{and} \qquad \underline{O}r = f+d$$

Let us now consider the successive halving operations on a sequence of length N=8 as shown in figure 5. Note that at the final stage of the expansion there are eight terms not involving an $\underline{O}$ operator. These terms can be shown to be none other than the Walsh Transform coefficients. Thus there are two methods to arrive at the results of the last stage of the halving operations.

(i) Direct expansion which produces all the terms.

(ii) Using the Walsh Transform, then generating the rest of the terms since

$$\underline{O} = 1/2\ (\underline{P}-\underline{S}) \qquad \text{and} \qquad \underline{E} = 1/2\ (\underline{P}+\underline{S})$$

For example,

$$\underline{SOS}x = 1/2\ (\underline{SPS}x - \underline{SSS}x)$$

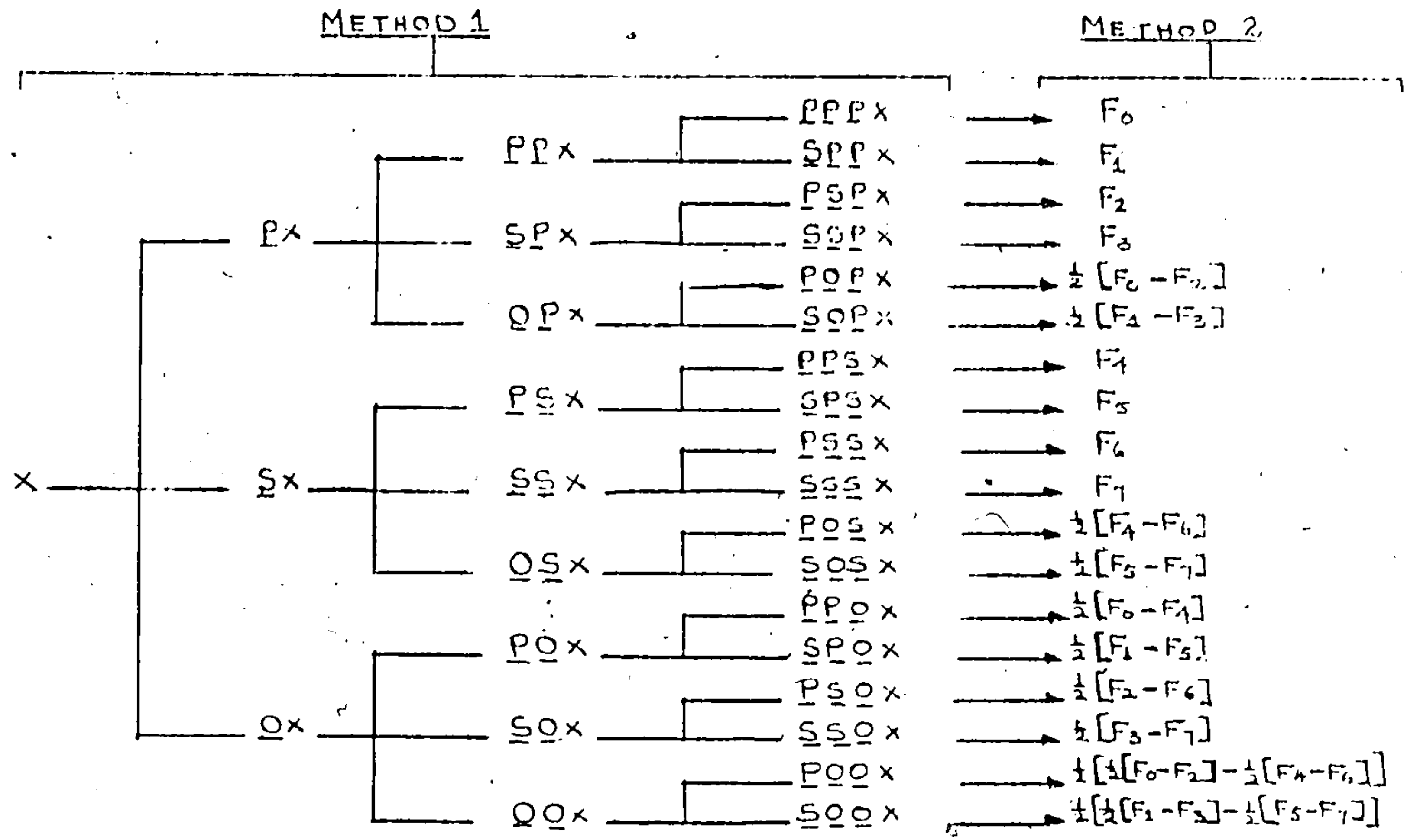


Fig. **5.** Successive halving operations on an 8-length sequence.

Many different forms of the algorithm can be derived by defining different auxiliary convolutions. eg.,

If

$$c = \underline{E}x * \underline{E}y \qquad d = \underline{O}x * \underline{E}y \qquad f = \underline{E}x * \underline{O}y$$

Then

$$\underline{E}r = c-d-f \qquad \text{and} \qquad \underline{O}r = f+d'$$

Let us now consider the successive halving operations on a sequence of length N=8 as shown in figure 5. Note that at the final stage of the expansion there are eight terms not involving an $\underline{O}$ operator. These terms can be shown to be none other than the Walsh Transform coefficients. Thus there are two methods to arrive at the results of the last stage of the halving operations.

(i) Direct expansion which produces all the terms.

(ii) Using the Walsh Transform, then generating the rest of the terms since

$$\underline{O} = 1/2\ (\underline{P}-\underline{S}) \qquad \text{and} \qquad \underline{E} = 1/2\ (\underline{P}+\underline{S})$$

For example,

[illegible] = 1/2 ([illegible]x − [illegible]x)

Method 1. Direct expansion of sequences of length N where $N=2^M$.

(i) Perform P,S and O on x-vector.

(ii) Reorder the resulting vectors and combine to form one vector. (or reordering may be done after final expansion stage.

(iii) Repeat (i) and (ii) M-1 times.

(iv) Perform P and S operations on vector formed in the $M-1^{th}$ stage to obtain the M^{th} stage.

(v) Repeat (i) to (iv) for y-vector except that O is replaced by E.

To complete the convolution process the following procedure is carried out:

(i) Form the auxiliary convolution functions c,d, and f by multiplying corresponding terms of both sequences.

(ii) Perform P and S operations, then divide terms of vector by 2.

(iii) Perform $2\,\underline{E}r = c+d$ and $2\,\underline{O}r = c-d+f'-f$ until the result is obtained.

Example using method 1.

Consider two sequences, each of length N=8 denoted as follows

$$x = y = (5,4,3,2,1,0,0,0)^T$$

To compute the cyclic convolution we proceed as follows

(i) Expand until the M^{th} stage.

				PPx	14	PPPx	15
					1	SPPx	13
			9	SPx	4	PSPx	5
			5		1	SSPx	3
		Px	1	OPx	5	POPx	5
	5		0		0	SOPx	5
	4			PSx	2	PPSx	3
	3		1		1	SPSx	1
x =	2	Sx	1	SSx	0	PSSx	1
	1		1		1	SSSx	-1
	0		0	OSx	1	POSx	1
	0				0	SOSx	1
	0		4	POx	6	PPOx	6
		Ox	2		0	SPOx	6
			0	SOx	2	PSOx	2
			0		0	SSOx	2
				OOx	2	POOx	2
					0	SOOx	2

				PPy	14	PPPy	15
					1	SPPy	13
			9	SPy	4	PSPy	5
			5		1	SSPy	3
		Py	1	EPy	9	PEPy	10
	5		0		1	SEPy	8
	4			PSy	2	PPSy	3
	3		1		1	SPSy	1
y =	2	Sy	1	SSy	0	PSSy	1
	1		1		1	SSSy	-1
	0		0	ESy	1	PESy	2
	0				1	SESy	0
	0	Ey	5	PEy	8	PPEy	9
			3		1	SPEy	7
			1	SEy	2	PSEy	3
			0		1	SSEy	1
				EEy	5	PEEy	6
					1	SEEy	4

(ii) Form the auxiliary convolution functions.

ccc = PPPx . PPPy = 225

dcc = SPPx . SPPy = 169

....

....

cfc = 2 POPx . PEPy = 100

....

(iii) Performing P and S operations followed by c+d=2Er and c-d+f'-f=2Or until the result is obtained.

ccc	225	cc	197		107			
dcc	169		28		50			
cdc	25	dc	17	c	18			
ddc	9		8		50	55		
cfc	100	fc	90			26	=	Er
dfc	80		10			10		
ccd	9	cd	5		3	26		
dcd	1		4		2			
cdd	1	dd	1	d	2			
ddd	1		0		2	40		
cfd	4	fd	2			14	=	Or
dfd	0		2			14		
ccf	108	cf	96		52	40		
dcf	84		12		28			
cdf	12	df	8	f	8			
ddf	4		4		20			
cff	48	ff	40					
dff	32		8					

Method 2. Using the Walsh Transform Coefficients.

(i) Obtain the Walsh transform coefficients of the x and y sequences using the Fast Walsh transform.

(ii) Generate all the 1st level terms from the Walsh coefficients using O for the x sequence and E for the y sequence.

(iii) Multiply the corresponding terms of the Walsh coefficients and the 1st level 1st iteration terms of both sequences.

(iv) Perform P and S operations, divide each term of vector by two, then 2Er=c+d and 2Or=c-d+f'-f and store.

(v) Generate all the 2nd level terms from the 1st level terms.

(vi) Multiply the corresponding terms of the 1st level (except the 1st iteration 1st level) and the 1st iteration 2nd level.

(vii) Repeat operation (iv) on multiplied terms.

(viii) Continue until M-1 levels of terms are generated.

(ix) Arrange all four-term sequences formed, into one vector and perform c+d=2Er and c-d+f'-f=2Or until the result is obtained.

To generate the rest of the terms using Method 2.

Example. For N=8, the procedure is as follows:

Walsh Coeff.	1st Level		2nd Level
	1st It.	2nd It.	1st It.
F_0	$(F_0-F_2)/2$	$(F_0-F_4)/2$	$(F_0-F_2-F_4+F_6)/4$
F_1	$(F_1-F_3)/2$	$(F_1-F_5)/2$	$(F_1-F_3-F_5+F_7)/4$
F_2		$(F_2-F_6)/2$	
F_3		$(F_3-F_7)/2$	
F_4	$(F_4-F_6)/2$		
F_5	$(F_5-F_7)/2$		
F_6			
F_7			

Hence required terms consist of 8 Walsh Transform coefficients, 8 1st level terms and 2 2nd level terms.

For the y sequence two changes are made to take care of the multiplication factor in 2$\underline{O}$x * $\underline{E}$y. These changes are

(i) Perform addition instead of subtraction.

(ii) No division is done.

For example, the 1st Iteration 1st Level terms for the y sequence are $F_0 + F_2$

$F_1 + F_3$ etc.

Same example as in Method 1.

$$x = y = (5,4,3,2,1,0,0,0)^T$$

W.T."x"	1st Level Mult. It. 1	It. 2			Store	2nd Level	Mult	
15	5	6	225	100	107	2	108	48
13	5	6	169	80	50	2	84	32
5		2	25		18		12	
3		2	9		50		4	
3	1		9	4	3			
1	1		1	0	2			
1			1		2			
-1			1		2			
W.T."y"								
15	20	18				24		
13	16	14				16		
5		6						
3		2						
3	4							
1	0							
1								
-1								

Store	Collect 4-terms	Result	
52	107		
28	50	55	
8	18	26	= $\underline{E}$ r
20	50	10	
	3	26	
	2		
	2		
	2	40	
	52	14	= $\underline{O}$ r
	28	14	
	8	40	
	20		

Note: Six terms from Mult. are reduced to four terms which are then stored. eg.,

225	100		107
169	80	are reduced to	50
25			18
9			50

by performing $\underline{P}$ and $\underline{S}$ operations, divide by 2, then

$c + d = 2\,\underline{E}\,r$ and $c - d + f' - f = 2\,\underline{O}\,r.$

Note: When using the Walsh Transform method, the 1st Itera--tion terms are always matched up with the preceding level terms (excluding the 1st Iteration terms). The four-term sequences which are stored, and then later recombined to form one vector, have to be placed in the proper order to be able to continue the reduction process.

For example: If each number represents a four-term sequence, the order in which they are placed are as follows

For N=8

1	3
2	

For N=16

1	3	7	9
2		8	
4	6		
5			

and so on.

N	Method 1 Total No. of Generated Items	Method 2								
		Walsh Transform Coefficient	1st Level	2nd Level	3rd Level	4th Level	5th Level	6th Level	7th Level	8th Level
8	18	8 ↔	8	2						
16	54	16	24	12	2					
32	162	32	64	48	16	2				
64	486	64	160 ↔	160	80	20	2			
128	1458	128	384	480	320	120	24	2		
256	4374	256	896	1344	1120	560	168	28	2	
512	13,122	512	2048	3584 ↔	3584	2240	896	224	32	2
1024	39,366	1024	4608	9216	10,752	8064	4032	1344	288	36
2048	118,098	2048	10,240	23,040	39,720	26,880	15,552	6,720	1,520	360

Fig. 6. Relationship between N and the number of terms to be generated.

2.4 Summary of both methods.

Method 1 requires

(i) The expansion of both sequences until 2.3^{M-1} terms of each are obtained.

(ii) The reordering of terms to be done before the reduction process can begin.

Hence the main drawbacks of method 1 are

(i) Storage requirements for 4.3^{M-1} terms.

(ii) A difficult reordering procedure.

Method 2 requires

(i) The generation of M-1 levels of terms starting with the Walsh transform coefficients.

(ii) The storage requirements to be based on at least the sum of the two largest levels of terms generated, since the next level of terms can be generated from the previous level.

Hence, although method 2 still requires a fair amount of storage, it is much less than that required for method 1. In addition, a large number of terms has to be generated, thus keeping track of the data causes an addressing problem.

2.5 Summary.

The Discrete Walsh functions have been presented. The Discrete Walsh Transform and its Inverse were shown to be computed very quickly as operations consisted of additi--ons and subtractions only.

The algorithm by Pitassi to compute cyclic convolution has been developed. Two methods of arriving at the final expansion stage are shown. In the first instance, the storage requirements are of the order 4.3^{M-1} and in the other, less storage is needed by using the Walsh Transform and generating the appropriate terms when they are needed. Both methods require the reordering of data at various points in the algorithm. This reordering procedure along with the large storage requirements are the factors which make this algorithm unattractive for computing cyclic convolution.

CHAPTER III

THE FAST FOURIER TRANSFORM

3.1 Introduction

Many methods (4) have been developed for computing the Discrete Fourier Transform based on the Fast Fourier Transform Technique. As the transformation of real data is considered, particular emphasis will be placed on a method based on Bergland's "Fast Fourier Transform for Real Valued Input". (1).

3.2 Definition

The Discrete Fourier Transform (DFT) of a sequence of N samples is given by

$$F_k = \sum_{n=0}^{N-1} f_n W^{nk} \qquad \text{where} \quad W = e^{-2\pi i/N}$$

It is possible to recover the original sequence from its DFT by

$$f_n = 1/N \sum_{k=0}^{N-1} F(k) \; W^{-nk}$$

The Fast Fourier Transform (FFT) is a special technique used for calculating the DFT which results in reduced operation when the number of samples is a certain value. In fact, for N an integral power of two, the number of operations is reduced to $N\log_2 N$ compared to N^2 for conventional methods.

3.3 Types of FFT algorithms

There are two basic types of FFT Algorithms. These are known as "decimation in time" and "decimation in frequency".

(i) Decimation in time - in which the transforms of shorter sequences, each composed of every r^{th} sample are computed, and then combined into one big transform.

Consider a sequence of N input samples $f_0, f_1, \ldots f_{N-1}$. Defining two shorter sequences as follows

$$g_l = f_{2l} = \text{even numbered samples}$$

$$h_l = f_{2l+1} = \text{odd numbered samples}$$

where $l = 0, 1, \ldots, N/2-1$.

By definition

$$G_k = \sum_{l=0}^{N/2-1} g_l \, (W^2)^{lk}$$

$$H_k = \sum_{l=0}^{N/2-1} h_l \, (W^2)^{lk}$$

As the DFT of the entire sequence is required, we get

$$F_k = \sum_{l=0}^{N/2-1} (g_l \, W^{2lk} + h_l \, W^{(2l+1)k})$$

$$= G_k + W^k H_k$$

Since index k runs from 0 to N-1 we get

$$F_k = G_k + W^k H_k \qquad 0 \leqslant k \leqslant N/2 - 1$$

$$= G_{k-N/2} + W^k H_{k-N/2} \qquad N/2 \leqslant k \leqslant N-1$$

(ii) Decimation in frequency- in which short pieces of the sequence are combined in r ways to form short sequences, whose separate transforms taken together constitute the complete transform.

Consider a sequence of N input samples $f_0, f_1, \ldots$ $\ldots, f_{N-1}$. We define two new sequences as follows

$g_l = f_l$ = the first N/2 samples

$h_l = f_{l+N/2}$ = the last N/2 samples

where $l = 0, 1, \ldots\ldots, N/2 - 1$

By definition

$$F_k = \sum_{l=0}^{N/2-1} \left(g_l W^{lk} + h_l W^{(l+N/2)k} \right)$$

Replacing k by 2k we get the following

$$F_{2k} = \sum_{l=0}^{N/2-1} (g_l + h_l) W^{2lk}$$

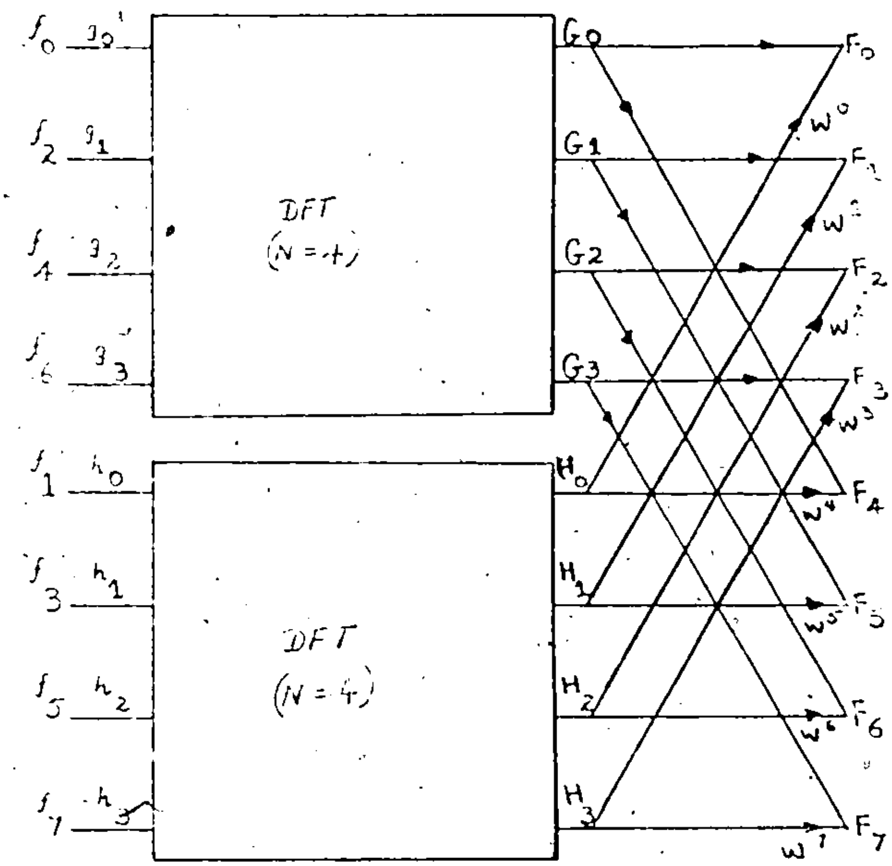


Fig. 7. Eight-point DFTs reduced to 2 four-point DFTs by decimation in time.

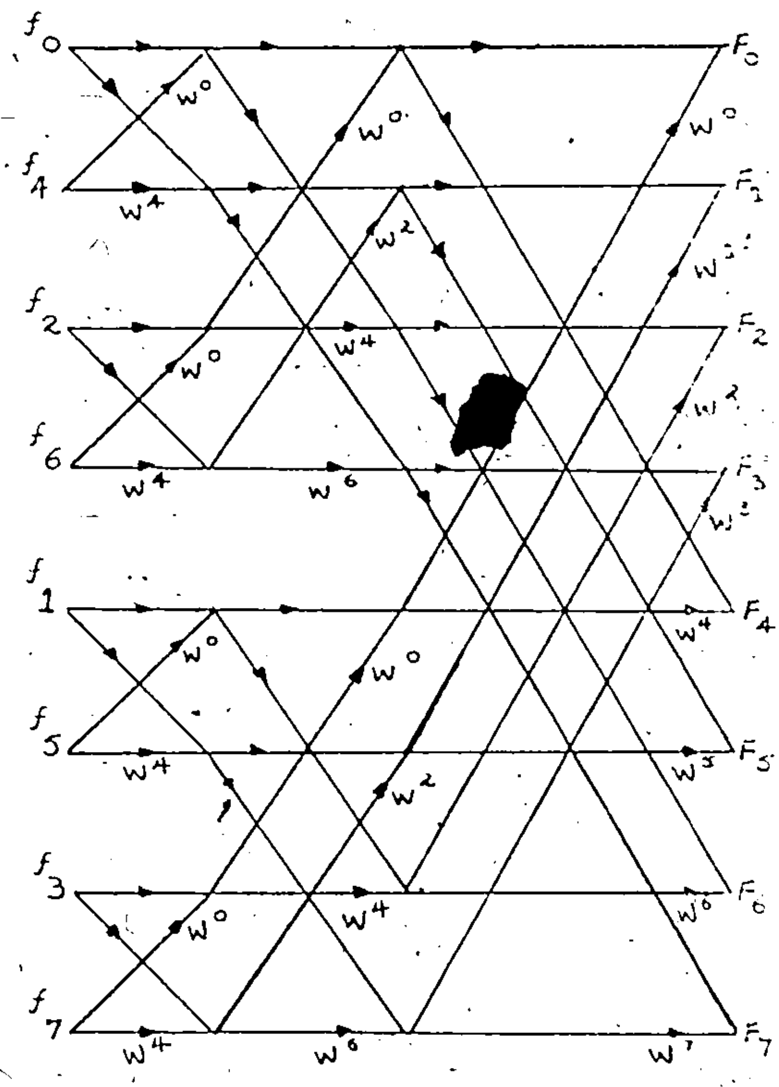


Fig. 8. Eight-point DFTs reduced to complex multiplications and additions by repeated decimation in time.

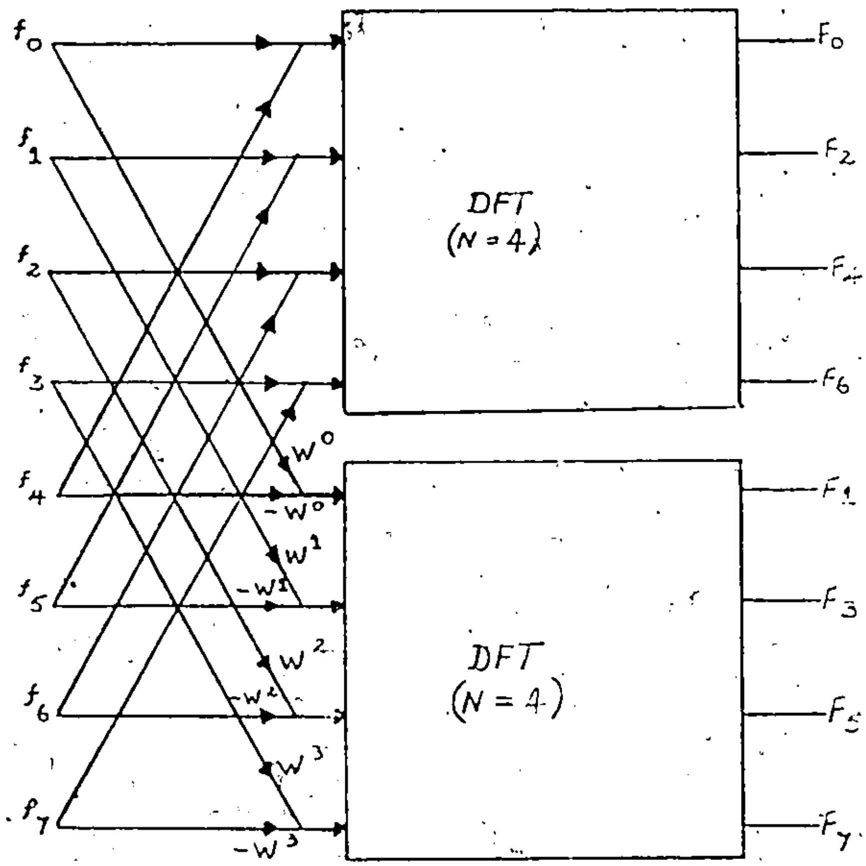


Fig. 9. Eight-point DFTs reduced to 2 four-point DFTs by decimation in frequency.

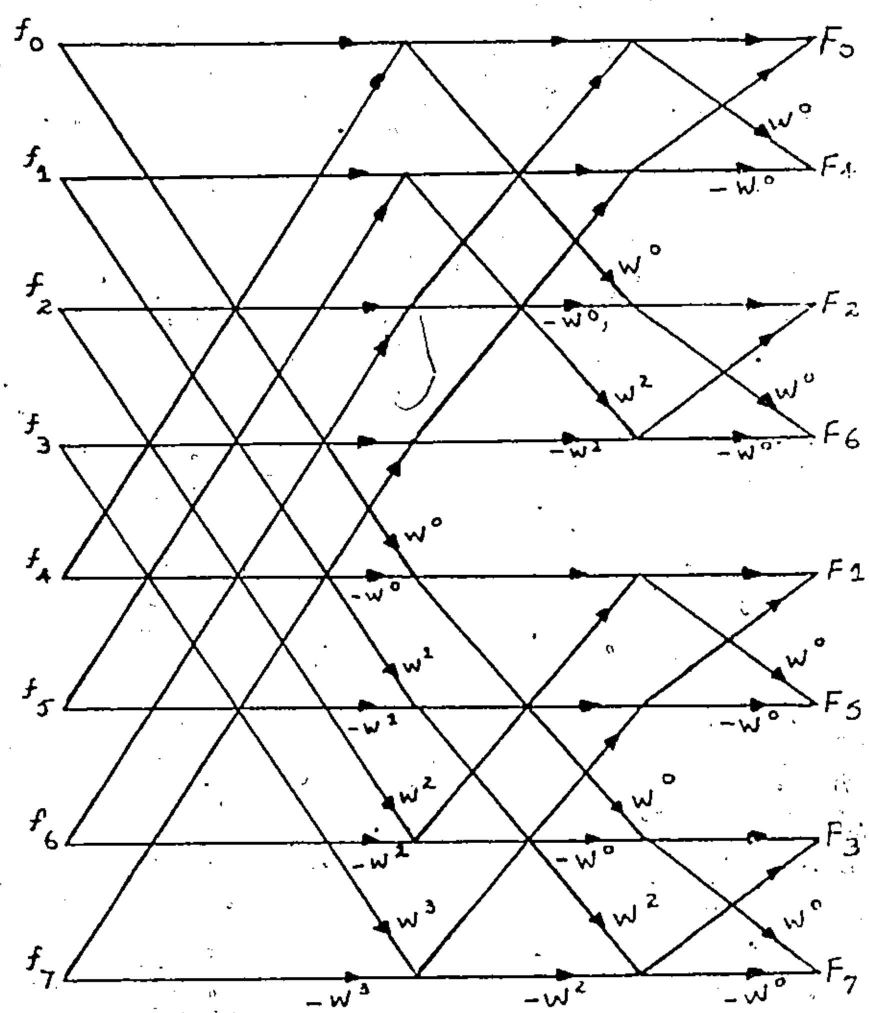


Fig. 10 . Eight-point DFTs reduced to complex multiplications and additions by repeated decimation in frequency.

Replacing k by 2k+1 we get the following

$$F_{2k+1} = \sum_{l=0}^{N/2-1} \left\{(g_l - h_l)\, W^l \right\} W^{2lk}$$

Figure 7. shows the DFT of an 8-point sequence formed from two 4-point DFTs by the decimation in time algorithm. Figure 8. shows the DFT of an 8-point sequence when it is reduced to only multiplications by repeated decimation in time.

Figure 9. shows the DFT of an 8-point sequence formed from two 4-point DFTs by the decimation in frequency algorithm. Figure 10. shows the DFT of an 8-point sequence when it is reduced to complex multiplications and additions by repeated decimation in frequency.

3.4 Factors involved in choosing the best FFT algorithm

The various factors to consider when choosing the best FFT algorithm are

(i) Input Ordering.

(ii) Output Ordering.

(iii) Computation of W exponents.

(iv) In-place Computation at each level.

Table 1 summarizes some of the possible combinations.

N = Normal and B = Bit-Reversed

Input	Computation	"W" Exponents	Output
N	In-place	B	B
N	In-place	N	B
N	Not-In-place	N	N
B	In-place	N	N
B	In-place	B	N

Table 1.

Note - There is the problem of either bit-reversed input or bit-reversed output ordering to consider. In the algorithm which uses both normal input and output ordering, the computation cannot be done in-place and thus an additional register is required in this instance.

3.5 Computation of "W" exponents

These algorithms assume either a table of stored exponents or some means of generating the required number of exponents when needed.

3.6 Algorithms for handling different types of data

(A). Complex Data.

Either the decimation in time or the decimation in frequency algorithm can be used to compute the Fast Fourier Transform of complex data.
In either case, the transform is computed using N complex location and requires $N\log_2 N$ operations.

(B). Real Data.

For real data any of the following methods may be used.

i. A complex FFT algorithm and set all the input imaginary parts to zero.
ii. A complex FFT algorithm which utilize an artificial N/2 term complex series called "Radix 2" by Bergland.

iii. Bergland's method called "Fast Fourier Transform for real valued input" which makes use of not computing redundant terms of the complex FFT algorithm.
iv. A modified method based on Bergland's FFTRVI called FFTBRRVI.

Of the methods for handling real data, Bergland's FFTRVI and the modified FFTBRRVI will be discussed.

3.7 Bergland's Method FFTRVI

Bergland's FFTRVI makes use of:

(A). Symmetry Considerations in calculating exponential weightings.

$$W^{k \text{ modulo } N} = W^k$$

$$W^{k + N/2} = -W^k \qquad 0 < k < N/2$$

$$W^{k + N/4} = -(W^k)^* \qquad 0 < k < N/4$$

(B). Hermitian Symmetry of Fourier Coefficients for real input data.

$$F_k = F_{N-k}^*$$

Therefore, it is not necessary to compute and store terms above one half the effective sampling frequency.

(C). A Different Complex Calculation which results in the FFT being computed in M-1 complete iterations.

3.8 Development of FFTRVI Algorithm

Consider the Cooley-Tukey[5] Complex FFT Algorithm shown in Figure 11.

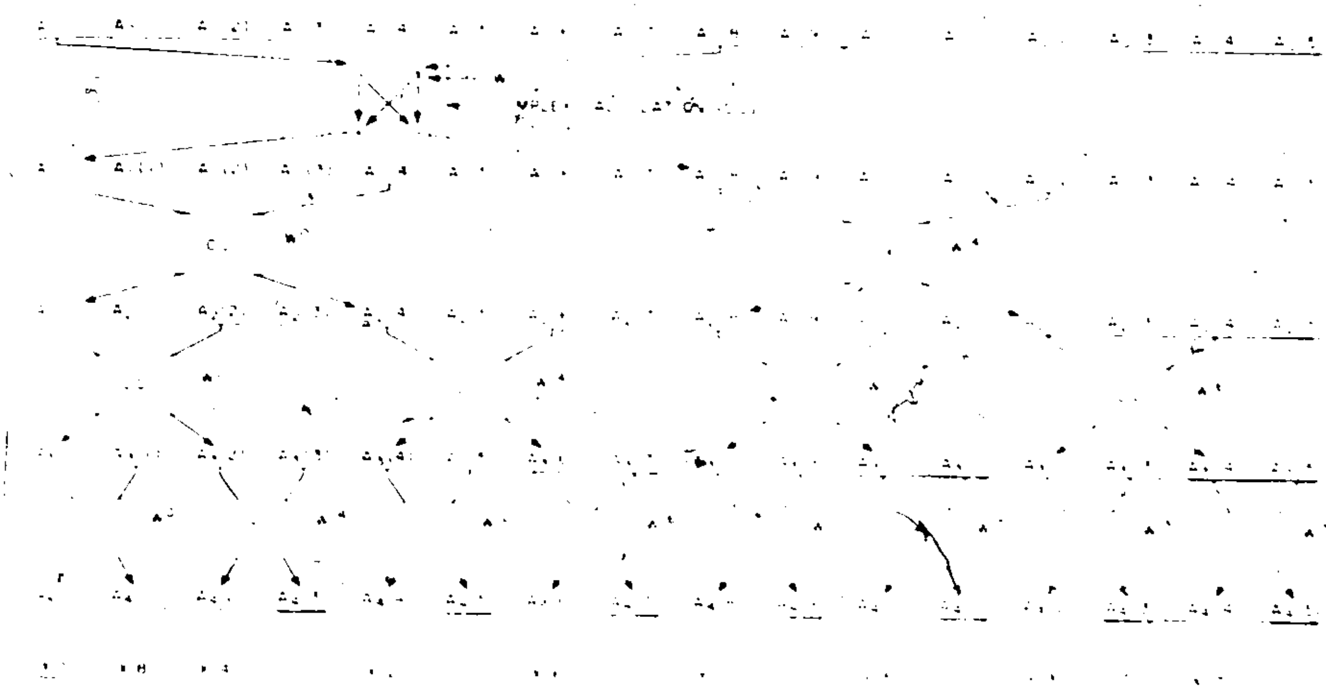


FIG. 11 The Cooley-Tukey complex FFT ... [illegible]

Note the following:

i. All storage location are complex.

ii. N/2 operations are performed in each iteration.

iii. Complex Calculation (C.C.) involves two Complex Inputs and two Complex Outputs.

Applying the following constraints to Figure 11.

i. All terms above one half the effective sampling frequency are neither computed nor stored, i.e. F9 to F15. These redundant terms are shown underlined at each iteration.

ii. Using the Symmetry of W exponents i.e.

$$W^{k \bmod N} = W^k$$

$$W^{k + N/2} = -W^k \qquad 0 < k < N/2$$

$$W^{k + N/4} = -(W^k)^* \qquad 0 < k < N/4$$

Hence the largest W exponent value needed is $W^{N/4}$.

iii. Since $W^{-N/4} = -i$, the complex calculation used by Cooley-Tukey is modified from :

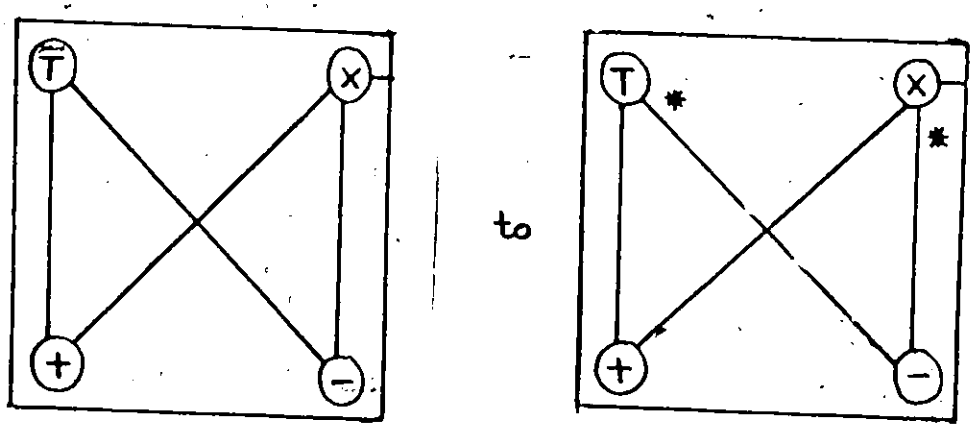


Fig. 12

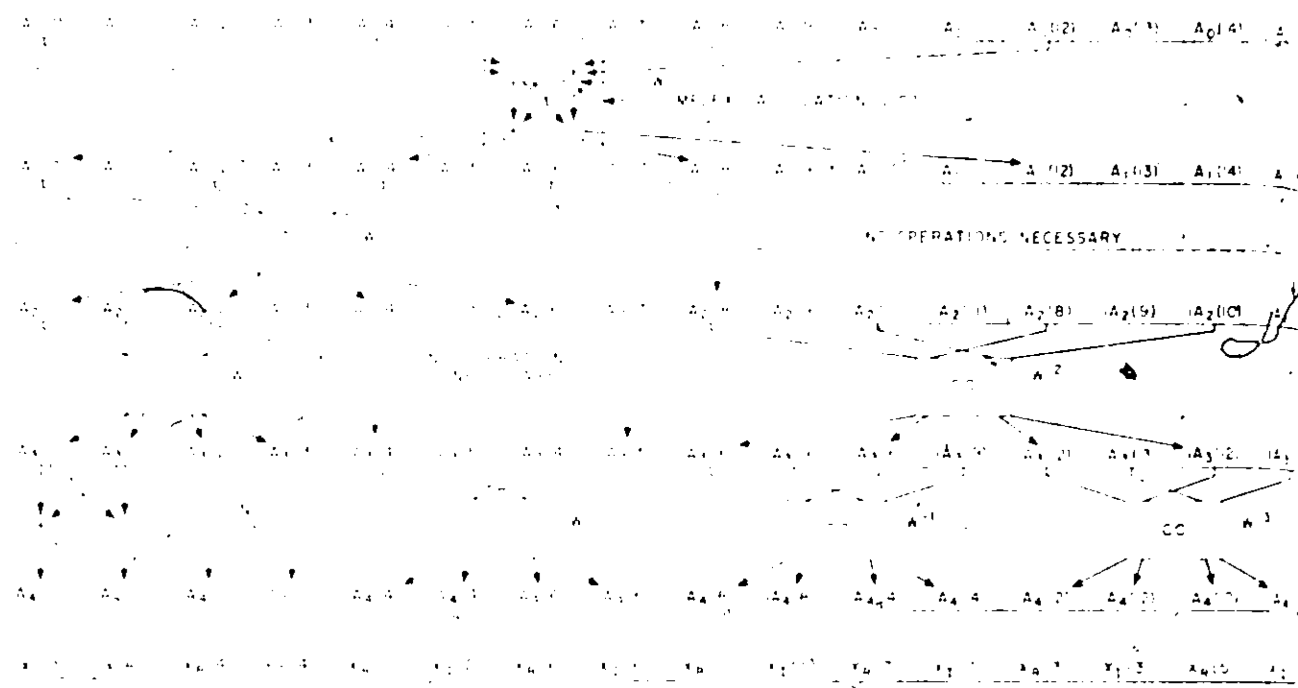


Fig. 13. [illegible]

Consider Figure 13 which contains the modifications. By performing all the operations below the "NO OPERATIONS NECESSARY" one iteration earlier, the M^{th} iteration is eliminated except for the 1^{st} two terms which must be replaced by their sum and difference respectively.

Hence in Figure 13 the following should be noted:

i. All storage locations are real.

ii. The Redundant Cooly-Tukey Intermediate Results are neither computed nor stored.

iii. A similar Complex Calculation (C.C.) involving two Complex Inputs and two Complex Outputs is used except that Real and Imaginary parts are stored in different locations.

iv. Imaginary parts of the Saved Terms are stored in locations vacated by Discarded Terms.

v. difference in the Complex Calculation of the Cooley -Tukey is that one of the results must be conjugated before being stored back in memory.

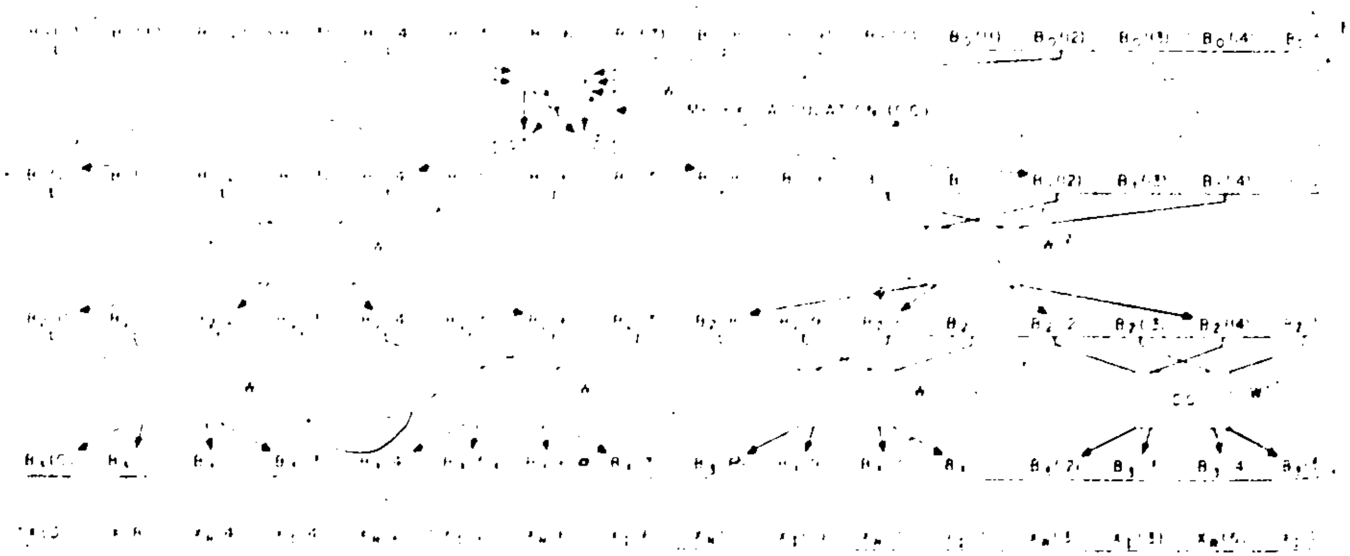


Fig. 14 The reduced-input fast Fourier transform algorithm for N = 16

3.9 Properties of the FFTRVI Algorithm

i. The Redundant Fourier Coefficients in each iteration above one half the effective sampling frequency are neither computed nor stored.

ii. The Intermediate results are accessed and stored in a regular and easily implemented pattern.

iii. The Real and Imaginary parts of the final Fourier Coefficients are formed in Adjacent Storage Locations.

iv. Only N Real Storage Locations are required throughout the computation these store the OriginalData Points, Intermediate, and final results.

v. The same set of Complex Arithmetic Operations is performed during the entire algorithm only the accessing order has to be changed when operands are real.

vi. The powers of W are called in the same order during each iteration.

vii. Only M-1 complete iterations are required when $N=2^M$.

3.10 Complex exponential weight table

This method presupposes a table of complex exponential weights which can be accessed sequentially in performing each iteration of the algroithm or they presuppose a method of sequentially computing these weights.

Seq.																
2	0								1							
4	0				2				1				3			
8	0		4		2		6		1		7		3		5	
16	0	8	4	12	2	14	6	10	1	15	7	9	3	13	5	11

Fig. 15. A method of generating the sequence of "W" exponents.

An algorithm for doubling the length of each number sequence.

i. Multiply the second entry of the sequence by two and make this product the second entry of the new sequence.

ii. Subtract each non-zero entry of the sequence from twice the product formed in step 1. (these differences form the rest of the even entries in new sequences.)

iii. Take the odd entries of the new sequence as the numbers of the original sequence.

Once the required sequence of W exponents is formed the corresponding W terms can be found and stored in this scrambled order.

3.11 Reordering of data for FFTRVI.

The same sequence used for placing the "W" exponents in the right order, is also needed for reordering of the final output data.

3.12 The modified method FFTBRRVI

As Bergland (1) hinted in his paper, if the input data is reordered, an algorithm should result in which the W exponents are needed in ascending order and hence can be generated recursively.

However, reordering the data does not produce the desired result. On the other hand, if the input data is bit- reversed and the W exponents are used in ascending order the modified algorithm is obtained.

Since a similar procedure to Bergland is adopted, the following will inherently occur as a result of using the input in bit reversed order instead of normal order.

i. The data is accessed in a different order.

ii. The real and imaginary parts are not formed in adjacent locations but are formed N/2 locations apart.

The main difference between the Bergland's FFTRVI and the modified method FFTBRRVI are:

i. The FFTBRRVI requiresthe W exponents in ascending order and thus can be generated recursively using the 2nd order difference equation whereas, FFTRVI has to use a stored table of W exponents.

ii. The FFTBRRVI has to do reordering of data after each iteration starting with the third , whereas, FFTRVI uses a stored sequence for reordering the final data.

As the modified method FFTBRRVI is based on Bergland's FFTRVI, both methods possess the following similar properties. These are as follows:

(i) The redundant Fourier coefficients in each iteration above one half the effective sampling frequency are neither computed nor stored.

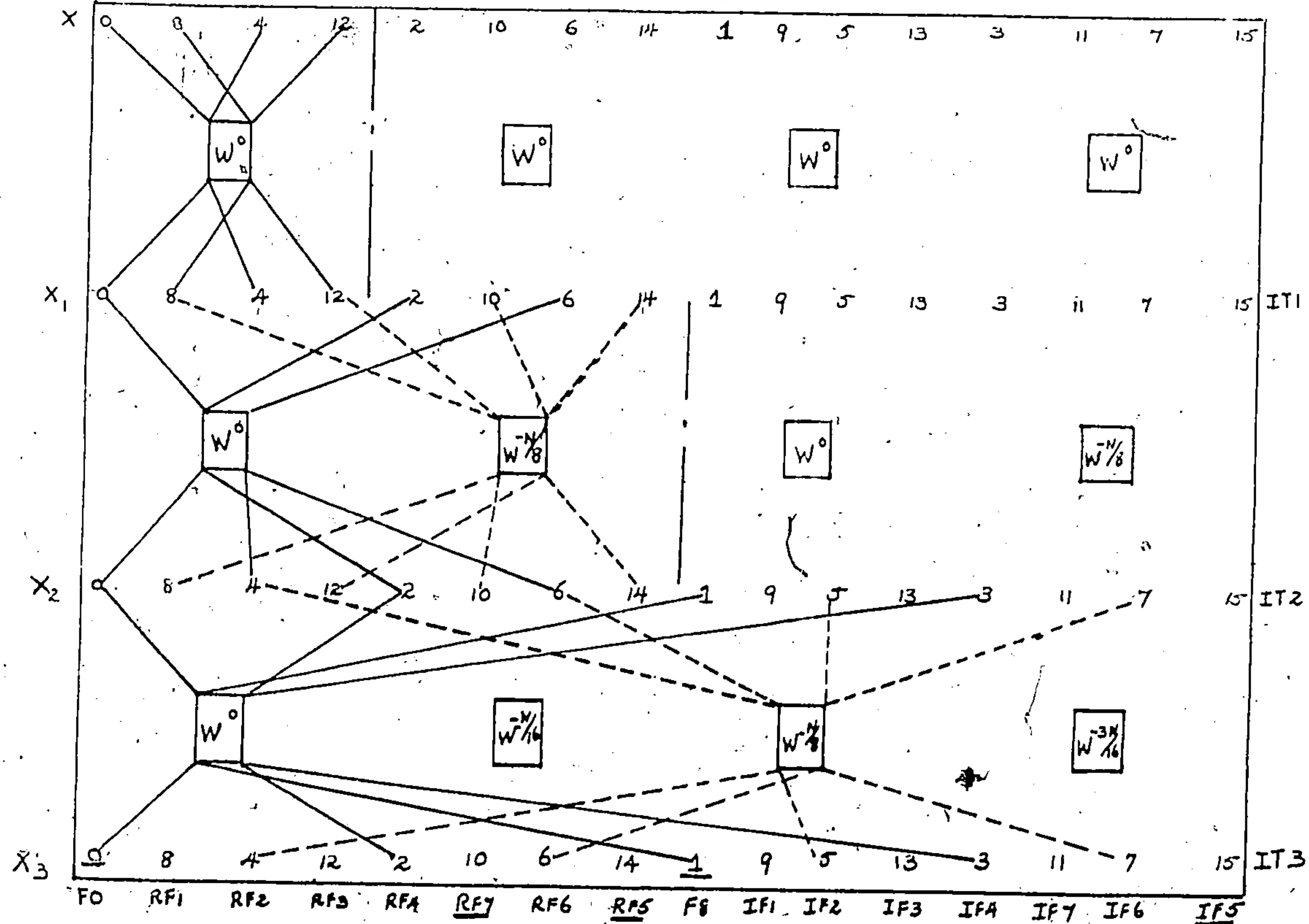
(ii) Intermediate results are stored and accessed in a regular and easily implemented pattern.

(iii) Only N real storage locations are required throughout the algorithm. These store the original data points, the intermediate results and the final results.

(iv) Only M-1 complete iterations are required.

(v) The same complex calculation is used.

Fig. 17. FFTBRRVI FOR N=16



3.13 No. of Operations required to generate Trigonometric values using the 2^{nd} order difference equation.

Iteration	Mult. Op.	Add. Op.
1	0	0
2	1	1
3	3N/16	3N/16
4	7N/32	7N/32
"	"	"
"	"	"
M-1	"	"

Number of Mult. Operation

$$= N \sum_{IT=3}^{M-1} \frac{2^{IT-1} - 1}{2^{IT+1}} = N \sum_{IT=3}^{M-1} \left(1/4 - 1/2^{IT+1}\right)$$

$$= N(M-4)/4 - N \sum_{3}^{M-1} 1/2^{IT+1}$$

$$= N(M-4)/4 - N/8 + 1$$

Total number of real multiplications = $N(M-4.5) + 4$

Total number of real additions = $N/2\,(M-4.5) + 2$

3.14 Reordering of data for FFTBRRVI.

Instead of reordering being done after the last iteration in most algorithms, it is done in-place after each iteration starting with the third. Hence, reordering of part of the data takes place for M-4 iterations.

Iteration	Number of Interchanges
3	N/8
4	3(N/16)
5	7(N/32)
"	"
"	"
M-1	"

Total number of interchanges $= \sum_{IT=3}^{M-1} (N/2^{IT})(2^{IT-2} - 1)$

$$= \sum_{IT=3}^{M-1} (N/2^{IT})(2^{IT-2}) - \sum_{IT=3}^{M-1} N/2^{IT}$$

$$= N/4\ (M-4) - N/4$$

$$= N(0.25M - 1.25)$$

3.15 Number of Operations to Compute Transform of Real Data using FFTBRRVI.

Real Operations

$$N/4 + N/8 + N/16 + \ldots\ldots\ldots\ldots\ldots + 1 = N/2 - 1$$

Complex Operations

$$(M - 1)\ N/4 - N/2 - 1 = MN/4 - 3N/4 + 1$$

1 Real Operation = 4 Real Additions.

1 Complex Operation = 4 Real Multiplications + 6 Real Additions.

$\therefore$ No. of Real Additions

$$= 4\left(N/2 - 1\right) + 2 + 6\left(MN/4 - 3N/4 + 1\right)$$

$$= 2N - 4 + 2 + 3MN/2 - 9N/2 + 6$$

$$= N\ (1.5M - 2.5) + 4$$

No. of Real Multiplications

$$= 4\left(MN/4 - 3N/4 + 1\right)$$

$$= N\ (M - 3) + 4$$

3.16 Convolution Using the DFT

If $x(K\Omega)$ and $y(K\Omega)$ are the DFT's of two sequences $x(nT)$ and $y(nT)$ respectively and $v(lT)$ is the result of convolving $x(nT)$ and $y(nT)$. Then using the short notation:

$$v_l = \sum_{n=0}^{N-1} x_n \, y_{((l-n))} = \sum_{n=0}^{N-1} y_n \, x_{((l-n))}$$

Expressed as DFT's we get:

$$v(k\Omega) = x(k\Omega) \quad y(k\Omega)$$

or

$$v(lT) = 1/N \sum_{k=0}^{N-1} x(k\Omega) \, y(k\Omega) \, e^{jlk\Omega T}$$

The discrete convolution can be computed from the inverse discrete Fourier Transform of the product of the DFT's of two sequences.

Thus one computes $x(k\Omega)$ and $y(k\Omega)$ using the FFT, multiplies them together to obtain $v(k\Omega)$, then evaluates $v(lT)$ using the inverse FFT.

This results in periodic convolution.

NOTE:

Aperiodic convolution is obtained by inserting zero valued samples to either of the components sequences as described earlier.

Since convolution and correlation are related by a change of variable, then an algroithm used for computing convolution can also be used for computing correlation by simply reversing one of the sequences.

3.17 Summary

The Fast Fourier Transform is a special technique used for calculating the DFT of a sequence. There are two basic types of FFT algorithms - the decimation in time and the decimation in frequency.

There are various factors to consider when choosing the best method of computing the DFT of a sequence. These are the input ordering, the output ordering, the computa--tion in place at each level and the computation of "W" exponents.

For complex data, either of the two types of FFT algorithms can be used. For real data, however, there are various methods. One of these, Bergland"s FFTRVI is prese--nted in some detail.and a method based on the FFTRVI is developed.

Finally, it was seen that convolution using the DFT is equivalent to taking three DFTs and a multiplication operation.

CHAPTER IV

RESULTS AND COMPARISONS.

Table 3. shows the comparison of methods for comput--ing the cyclic convolution of two real valued time series. The expression $4(M-2)(N-2)+2N$ is quoted from the research paper by Pitassi.[3] The following should be noted:

(i) The new and the Walsh transform methods require fewer multiplications than the FFT for $N < 1024$ and the FFTRVI for $N < 256$.

(ii) The difference in the number of multiplications between FFTBRRVI with stored exponents and FFTBRRVI without is the result of generating the "W" exponents from the second order difference equation.

Table 4. shows the comparison with regard to storage requirements. The following should be noted:

(i) To compute convolution using any of the FFT methods clearly requires much less storage than the new or Walsh transform methods.

(ii) The FFTBRRVI algorithm uses the least amount of storage.

Comparison of methods for computing the cyclic convolution of two real valued time series.

A. Multiplications

N	M	NEW (METHOD 1) AND WALSH TRANSFORM	FFT	FFTRYL AND FFTBRRYL (WITH STORED TABLE)	FFTBRRYL WITHOUT STORED TABLE
		$2 \cdot 3^{M-1}$	$4(M-2)(N-2)+2N$	$3[N(M-3)+4]+2N$	$6NM-20.5N+24$
16	4	54	144	92	108
32	5	162	424	268	328
64	6	486	1120	716	1016
128	7	1458	2276	1804	2776
256	8	4374	6608	4364 *	7064
512	9	13,122	15,304	10,252	17,176
1024	10	39,366	34,752 *	23,564	40,1148

TABLE 5

B. Storage Requirements

N	NEW (METHOD 1)	WALSH TRANSFORM (METHOD 2)	FFT	FFTRVI	FFTBRRVI WITH STORED W EXPONENTS	FFTBRRVI WITH GENERATED W EXPONENTS
	$4 \cdot 3^{M-1}$	2[2 LARGEST LEVELS]	$2N+N/2$	3N	$2N+N/4$	$2N+2(M-2)$
16	108	80	40	48	36	36
32	324	224	80	96	72	70
64	972	640	160	192	144	136
128	2916	1728	320	384	288	266
256	8748	4928	640	768	576	524
512	26,244	14,336	1280	1536	1152	1038
1024	78,732	39,936	2560	3072	2304	2061

TABLE 4

Table 5. shows the comparison of methods for computing the FFT of real data with regard to the number of multiplications and additions. The following should be noted:

(i) The FFTRVI and the FFTBRRVI (using a stored table of "W" exponents) compute the FFT in the least number of operations which is approximately half the amount required by Radix-2.

(ii) Although the FFTBRRVI without stored table requires more operations than FFTRVI, the amount is still less than the Radix-2 method for computing the FFT.

Table 6. shows the comparison of times by the different methods in computing the FFT of real data. The following should be noted:

(i) The FFTBRRVI, when using a stored table of "W" exponents, is faster than the Radix-2 method by about 20%.

(ii) When the "W" exponents are generated, FFTBRRVI computes the FFT in a time equivalent to the Radix-2.

(iii) Since a D/P multiplication is equivalent to four S/P multiplications, the time to compute the FFT using a D/P programme should be about four times that of a S/P programme. The time obtained for the D/P programme is therefore, quite good.

Comparison of methods to compute the FFT of real data.

N	MULTIPLICATIONS Radix2	FFTRV1	FFTBRRV1 with stored table	FFTBRRV1 without	ADDITIONS Radix2	FFTRV1	FFTBRRV1 with stored table	FFTBRRV1 without
	(2M-7)N+12	(M-3)N+6	(M-3)N+6	(2M-7.5)N+8	(3M-3)N+4	(1.5M-2.5)N+4	(1.5M-2.5)N+4	(2M-4.75)N+6
256	2316	1284	1284	2184	5380	2436	2436	2886
512	5644	3076	3076	5384	12,292	5636	5636	6780
1024	13,313	7172	7172	12,808	27,648	12,804	12,804	15,622
2048	30,720	16,364	16,364	28,782	61,4[illegible]	28,676	28,676	35,334

TABLE 5

Comparison of times for computing the FFT of real data.

N	Radix-2 S/P	FFTBRRVI (Table) S/P	FFTBRRVI (Generate) S/P	FFTBRRVI (Generate) D/P
64	.071	.065	.070	.185
128	.177	.155	.175	.465
256	.417	.340	.415	1.120
512	.963	.790	.960	2.65
1024	2.20	1.80	2.19	6.05

Table 6.

The above times are in seconds. The times using the FFTBRRVI algorithm were obtained by timing the program for 100 FFTs and IFFTs. The total time was then divided by 200 to obtain the average time for computing one FFT.

N	M	FFTRVI N	FFTBRRVI 2[M-2]
128	7	128	10
256	8	256	12
512	9	512	14
1024	10	1024	16
2048	11	2048	18
4096	12	4096	20

TABLE 1

Table 7. shows a comparison of the storage required by FFTRVI and FFTBRRVI for the "W" exponents. Note in the case of limited storage, FFTBRRVI uses negligible storage compared to FFTRVI. This is the Result of generating the "W" exponents which requires storage for just a few initial constants. The FFTRVI, on the other hand, has to make use of a table on account of the order in which the "W" exponents are needed.

Summary

The various methods for computing the cyclic convolution of two real valued time series are compared with reference to the number of operations (multiplications and additions) and the storage requirements. The method using the Walsh transform required the least number of multiplications for $N < 256$ whereas the FFTBRRVI required fewer for $N > 256$.

Of the various methods for computing the FFT of real data, FFTRVI and FFTBRRVI are the fastest with the latter being superior when storage requirements are limited.

Tests were carried out on the modified algorithm FFTBRRVI and the results were quite good.

CHAPTER V

DISCUSSION AND CONCLUSION.

5.1 Discussion

(i) Walsh Transform

As far as the computation of the Walsh transform is concerned, it is a very fast and easy operation to imple--ment on the digital computer. Consequently, the discrete Walsh transform has found many applications in digital signal processing.

With regard to using the Walsh transform for comput--ing periodic convolution, the procedure is not an easy one to implement on account of the amount of data to generate and access, and this leads to an addressing problem.

Though Pitassi's[3] method requires fewer multiplicat--ions than the FFT for computing the periodic convolution of two real time series for $N < 1024$, there are other fact--ors to consider such as the type of algorithm and the amount of storage needed.

Since this method lacks certain features, the FFT is now considered to evaluate its performance in computing periodic convolution.

(II) Fast Fourier Transform

Standard Method -"Radix-2"
Bergland's - "FFTRVI"
Modified - "FFTBRRVI"

The criteria for determining the best FFT algorithm for computing real data will be as follows

I. To minimize computer time which involves minimizing the number of operations (multiplications and additions.)

II. To minimize the storage requirements which involves

(a). In-place computation at each level.

(b). In-place reordering of data.

(c). Computation or Storage of "W" exponents.

Comparison of Radix-2 and FFTRVI.

i. FFTRVI requires approximately half the number of operations as Radix-2 for computing the FFT of real data. Hence FFTRVI is expected to be superior to the Radix-2 by computing the FFT in a faster time.

ii. FFTRVI, in addition to using a stored table of "W" exponents as Radix-2, has to generate and store a sequence which is required both for placing the "W" exponents in the right order for accessing and for the final output reordering.

Hence, if storage requirements are not limited, FFTRVI is superior to Radix-2 even taking into consideration the time taken for generating and storing the sequence. If storage requirements are limited, then FFTRVI may be ruled out on account of the additional storage required for the sequence. Consequently, a compromise must be made between the savings to be gained in computer time and the additional storage requirements. The existing conditions will therefore, determine which algorithm to use.

Comparison of FFTBRRVI and Radix-2.

i. FFTBRRVI computes the FFT of real data in fewer operations than the Radix-2 whether storage requirements are limited or not.

In the case where storage requirements are not limited, FFTBRRVI can use a stored table of "W" exponents as Radix-2 and thus compute the FFT with savings of up to 20% over the Radix-2.

When storage requirements are limited, FFTBRRVI can be used with reduced storage by generating the "W" exponents. Only 2(M-2) storage values are then needed for the Trigono--metric generating routine compared to a stored table of N/2 sine values required by Radix-2.

Thus FFTBRRVI is definitely superior to the Radix-2.

Comparison of FFTBRRVI and FFTRVI.

I. When storage requirements are not limited, FFTRVI and FFTBRRVI(by using a stored table of "W" exponents) can compute the FFT of real data in the same number of operations.

II. FFTRVI has to generate and store a sequence which is needed for the storing of "W" exponents and for reordering the final data whereas FFTBRRVI has not.

III. Additional storage is required for FFTRVI since it is difficult to perform in-place reordering of the final output data whereas in the case of FFTBRRVI, reorder--ing is done in-place, after each iteration starting with the third.

IV. When storage requirements are limited, FFTBRRVI can be adapted to generate the "W" exponents which results in reduced storage. FFTRVI cannot be changed since it has to make use of the sequence. Hence, compulsory storage is needed for (a)the sequence (b)"W" exponents.

The modified method FFTBRRVI has been shown to be superior to both the standard Radix-2 and Bergland's FFTRVI for computing the FFT of real data. FFTBRRVI is very flexible in that it can be adapted to suit the existing conditions. On the one hand, it computes the FFT in the fastest time when storage requirements are not limited. On the other, it can be used with a minimum of storage and still compute the FFT in a faster time than Radix-2, even when other methods may be ruled out on account of the small storage to be utilized.

Since convolution (or correlation) can be exchanged for 3FFT's and a multiplication operation, it is therefore, dependent on the time taken to compute the FFT. As the modified method FFTBRRVI is the superior algorithm, it is expected to take the least time to compute convolution.

The modified method FFTBRRVI has been tested under both conditions and the computation times have confirmed its performances in both instances.

The programme using FFTBRRVI is written in double precision and implemented on the PDP-8/I minicomputer. It includes computation of the FFT and IFFT, and also aperiodic convolution and correlation.

5.2 Conclusion

The computation of the Discrete Walsh Transform and the Fast Fourier Transform algorithms are easy to implement on the minicomputer with the former being easier and faster.

For computing periodic convolution however, it is seen that FFTBRRVI requires fewer multiplications than the Walsh transform method for $N > 256$. Although Pitassi (5) makes the comparison with another FFT method and finds his method requiring less multiplications for $N < 1024$, it is not really advantageous to use his method since it utilizes much more storage than any of the FFT methods, and has an addressing problem associated with it. Compared with the modified method FFTBRRVI, the Walsh transform method is proved inferior.

The modified method FFTBRRVI is seen to be the best method for computing the FFT of real data since it does so in the fastest time and can be used when storage conditions are limited.

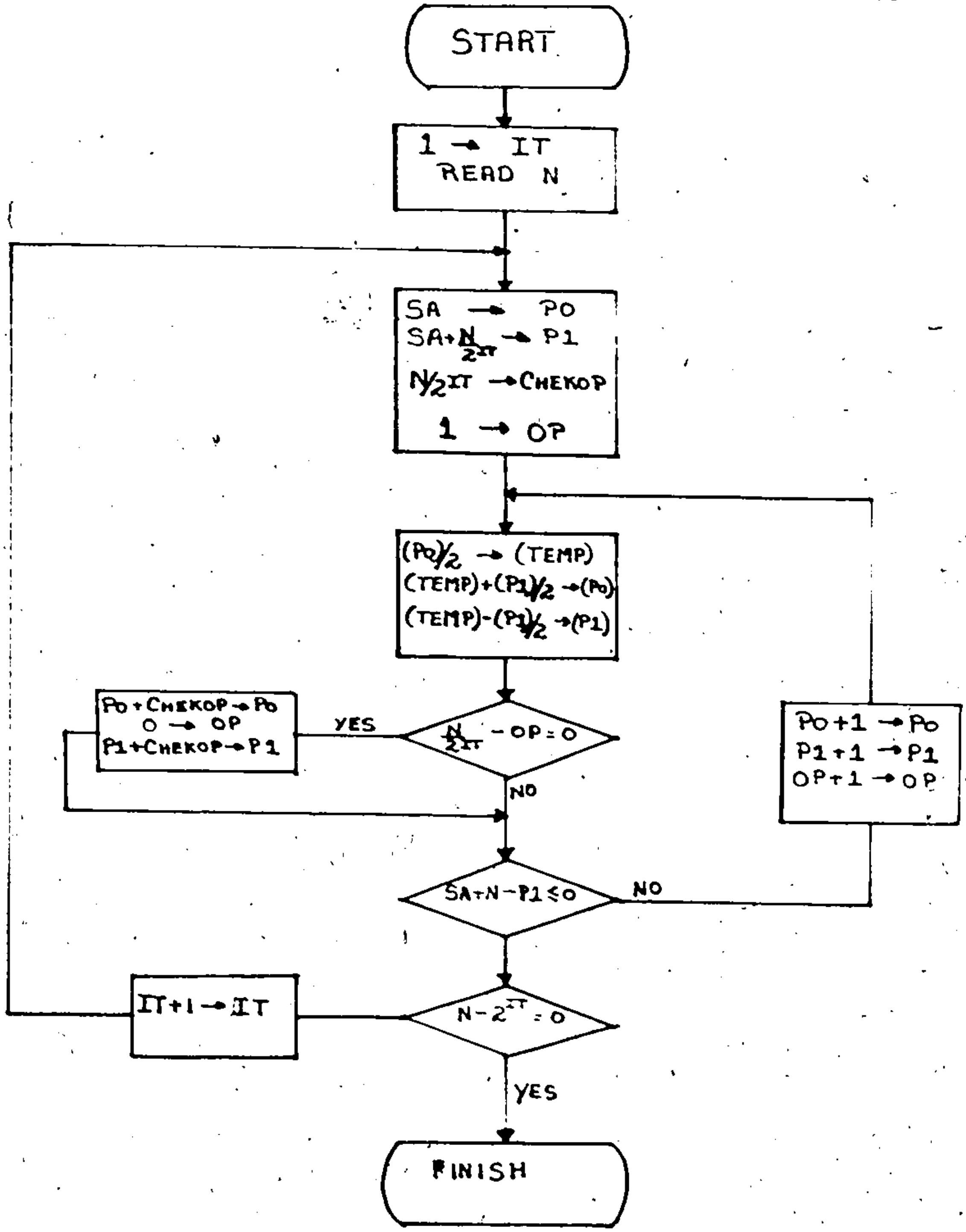
With improvements in computer hardware, the use of discrete transform algorithms in the signal processing field will become even more attractive.

APPENDIX A.

A. The Discrete Walsh Transform Flowchart.

A.1 Walsh Transform programme written in PAL III assembly language.

This programme accepts four-digit octal numbers as input. The end of the input is signified by typing a dollar($) sign. The amount of numbers must be an integral power of two. The output is printed out on the teletype machine.



WALSH TRANSFORM FLOWCHART

```
                MQA=7501
                CAM=7621
                ASR=7415
                SHL=7413
                *200
0200    7300    START, CLA CLL
0201    6046    TLS
0202    1107    TAD SA
0203    3110    DCA BUFFPT
0204    3112    DCA AMOUNT
0205    4073    ACCEPT, JMS CRLF
0206    1111    TAD M4
0207    3113    DCA DIGCTR
0210    1114    TAD TEMP1
0211    3115    DCA TEMP
0212    4101    NEXDIG, JMS LISN
0213    3515    DCA I TEMP
0214    1515    TAD I TEMP
0215    1122    TAD MDOLAR
0216    7650    SNA CLA
0217    5246    JMP RUN
0220    2115    ISZ TEMP
0221    2113    ISZ DIGCTR
0222    5212    JMP NEXDIG
0223    1114    PACK, TAD TEMP1
0224    3115    DCA TEMP
0225    3123    DCA HOLD
0226    1111    TAD M4
0227    3113    DCA DIGCTR
0230    1123    DIGPCK, TAD HOLD
0231    7104    CLL RAL
0232    7006    RTL
0233    1515    TAD I TEMP
0234    1124    TAD M260
0235    3123    DCA HOLD
0236    2115    ISZ TEMP
0237    2113    ISZ DIGCTR
0240    5230    JMP DIGPCK
0241    1123    TAD HOLD
0242    3510    DCA I BUFFPT
0243    2112    ISZ AMOUNT
0244    2110    ISZ BUFFPT
0245    5205    JMP ACCEPT
0246    1112    RUN, TAD AMOUNT
0247    3126    DCA N
0250    7001    IAC
0251    3125    DCA IT
0252    7240    AGAIN, STA
0253    1125    TAD IT
0254    3262    DCA TEMPA
```

```
0255  1262  TAD TEMPA
0256  3267  DCA TEMPB
0257  7621  CAM
0260  7001  IAC
0261  7413  SHL
0262  0000  TEMPA,0
0263  3140  DCA BIT
0264  7621  CAM
0265  1126  TAD N
0266  7415  ASR
0267  0000  TEMPB,0
0270  3135  DCA CHEKOP
0271  1135  TAD CHEKOP
0272  1107  TAD SA
0273  3136  DCA POINTI
0274  1107  TAD SA
0275  3127  DCA POINTO
0276  7001  IAC
0277  3137  DCA OP
0300  1527  REPEAT,TAD I POINTO
0301  3050  DCA ADD1
0302  1536  TAD  I POINTI
0303  4452  JMS I  ADDER
0304  3130  DCA TEMPO
0305  1536  TAD I  POINTI
0306  7041  CIA
0307  3050  DCA ADD1
0310  1527  TAD I  POINTO
0311  4452  JMS I  ADDER
0312  3536  DCA I POINTI
0313  1130  TAD  TEMPO
0314  3527  DCA  I POINTO
0315  1137  TAD OP
0316  7041  CIA
0317  1135  TAD CHEKOP
0320  7640  SZA CLA
0321  7410  SKP
0322  4053  JMS TESTA
0323  1136  TAD POINTI
0324  7041  CIA
0325  1126  TAD N
0326  1107  TAD SA
0327  7540  SMA SZA
0330  7410  SKP
0331  5344  JMP TESTB
0332  7201  CLA IAC
0333  1137  TAD OP
0334  3137  DCA OP
0335  1127  TAD POINTO
0336  7001  IAC
0337  3127  DCA POINTO
```

```
0340  1136  TAD POINTI
0341  7001  IAC
0342  3136  DCA POINTI
0343  5300  JMP REPEAT
0344  7200  TESTB, CLA
0345  1140  TAD BIT
0346  7041  CIA
0347  1126  TAD N
0350  7640  SZA CLA
0351  7410  SKP
0352  5357  JMP PRINT
0353  1125  TAD IT
0354  7001  IAC
0355  3125  DCA IT
0356  5252  JMP AGAIN
0357  4073  PRINT, JMS CRLF
0360  1107  TAD SA
0361  3110  DCA BUFFPT
0362  1112  TAD AMOUNT
0363  7041  CIA
0364  3141  DCA PRNTCT
0365  4073  ANOTHR, JMS CRLF
0366  1111  TAD M4
0367  3113  DCA DIGCTR
0370  3123  DCA HOLD
0371  1510  TAD I BUFFPT
0372  7104  CLL RAL
0373  1123  MORE, TAD HOLD
0374  7004  RAL
0375  7006  RTL
0376  3123  DCA HOLD
0377  1123  TAD HOLD
0400  0142  AND MASK7
0401  1143  TAD K260
0402  5531  JMP I NEXT
0403  4065  NEXT1, JMS TYPE
0404  2113  ISZ DIGCTR
0405  5534  JMP I MOD
0406  2110  ISZ  BUFFPT

0407  2141  ISZ PRNTCT
0410  5533  JMP I ANT
0411  4073  JMS CRLF
0412  5532  JMP I STAY
0413  0000  ADDR, 0
0414  3051  DCA ADD2
0415  7621  CAM
0416  1050  TAD ADD1
0417  7415  ASR
0420  0000  0
```

```
0421  3050  DCA ADD1
0422  1051  TAD ADD2
0423  7415  ASR
0424  0000  0
0425  3051  DCA ADD2
0426  7501  MQA
0427  7004  RAL
0430  7060  CMA CML
0431  7720  CLA  SMA SNL
0432  7001  IAC
0433  1050  TAD ADD1
0434  1051  TAD ADD2
0435  5613  JMP I ADDR
            *50
0050  0000  ADD1,0
0051  0000  ADD2,0
0052  0413  ADDER,ADDR
0053  0000  TESTA,0
0054  7200  CLA
0055  3137  DCA OP
0056  1135  TAD CHEKOP
0057  1127  TAD POINT0
0060  3127  DCA POINT0
0061  1135  TAD CHEKOP
0062  1136  TAD POINT1
0063  3136  DCA POINT1
0064  5453  JMP I TESTA
0065  0000  TYPE,0
0066  6041  TSF
0067  5066  JMP.-1
0070  6046  TLS
0071  7200  CLA
0072  5465  JMP I TYPE
0073  0000  CRLF,0
0074  1144  TAD K215
0075  4065  JMS TYPE
0076  1145  TAD K212
0077  4065  JMS TYPE
0100  5473  JMP I CRLF
0101  0000  LISN,0
0102  6031  KSF
0103  5102  JMP.-1
0104  6036  KRB
0105  6046  TLS
0106  5501  JMP I LISN
0107  2000  SA,2000
0110  0000  BUFFPT,0
0111  7774  M4,7774
0112  0000  AMOUNT,0
0113  0000  DIGCTR,0
0114  0116  TEMP12,.+2
```

```
0115  0000  TEMP,0
0116  0000  0
0117  0000  0
0120  0000  0
0121  0000  0
0122  7534  MDOLAR,7534
0123  0000  HOLD,0
0124  7520  M260,-260
0125  0000  IT,0
0126  0000  N,0
0127  0000  POINTO,0
0130  0000  TEMPO,0
0131  0403  NEXT,NEXTI
0132  0200  STAY,START
0133  0365  ANT,ANOTHR
0134  0373  MOD,MORE
0135  0000  CHEKOP,0
0136  0000  POINTI,0
0137  0000  OP,0
0140  0000  BIT,0
0141  0000  PRNTCT,0
0142  0007  MASK7,7
0143  0260  K260,260
0144  0215  K215,215
0145  0212  K212,212
```

APPENDIX B

B1. Flowcharts dealing with the FFTBRRVI programme.

i. FFT

ii. IFFT

iii. GETRIG

iv. REORD

v. INVERT

vi. RBITS

B2. Double Precision programme written in PAL III assembly language. The programme computes the following :

i. The FFT of real data (D/P).

ii. The IFFT of the data in the order left by FFT.

iii. Aperiodic convolution and correlation.

The following are the subroutines and their functions.

FFT - takes the FFT of real data stored in the buffer.

IFFT - takes the IFFT of the data as left by FFT.

GETRIG- computes the sine and cosine values recursively using the 2nd order difference equation.

RBITS- bit-reverses the data in the buffer.

REORD- performs special reordering needed with FFT and IFFT.

SQRT - takes the square root of a D/P number.

MULT - multiplies two D/P numbers.

INITCS- initializes the required starting values for GETRIG.

MODLUS - computes the modulus of the real and imaginary parts of a complex number.

DUBADD - adds two D/P numbers.

SCALE - scales data for display on oscilloscope.

GRAPH - displays 512 points of the buffer on oscilloscope.

SAMPLE - performs sampling of input device and stores values in buffer.

ERROR - performs two's complement of a D/P number.

REALOP - performs the real operations in FFT and IFFT.

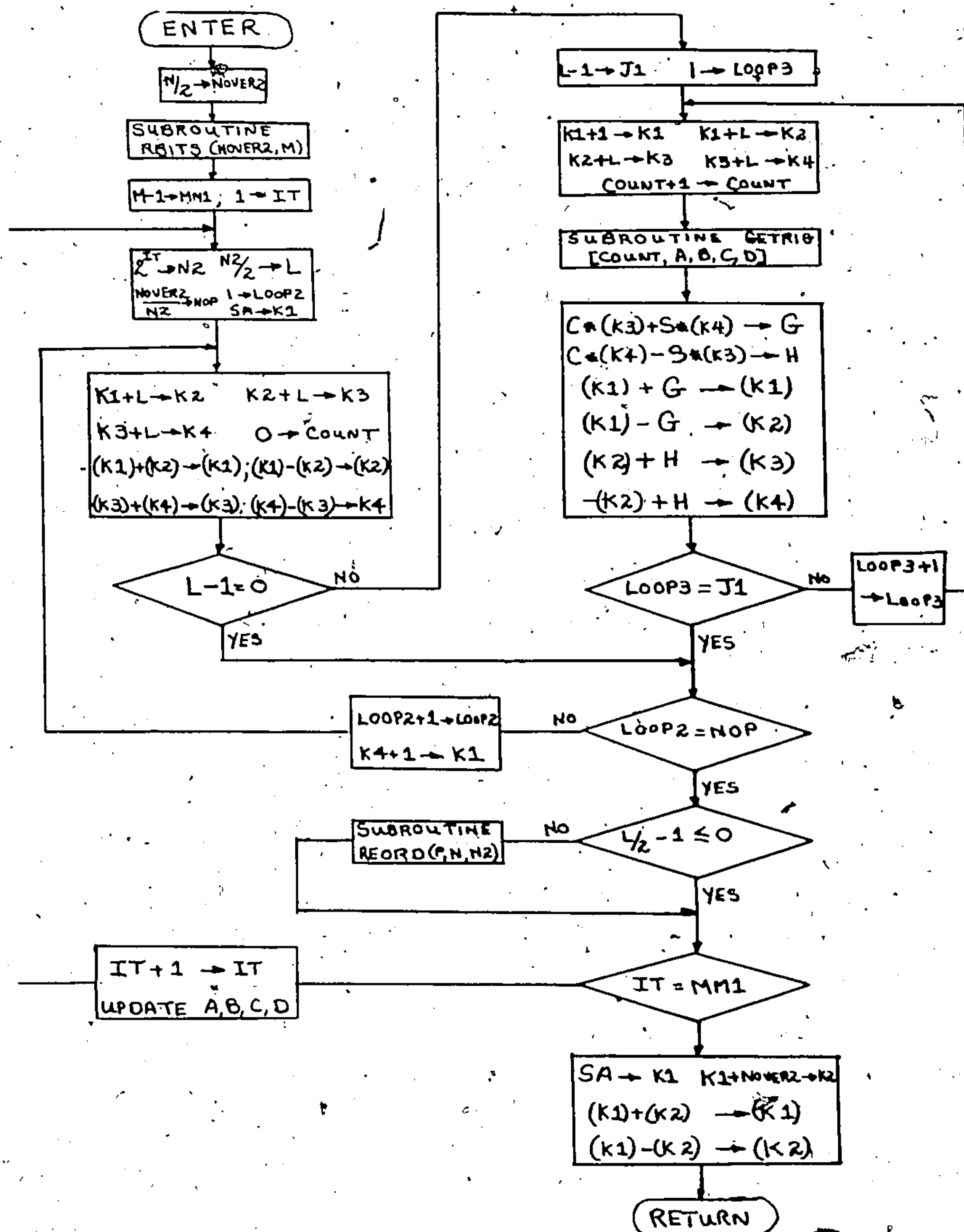
COMPLX - performs the complex operations in FFT and IFFT.

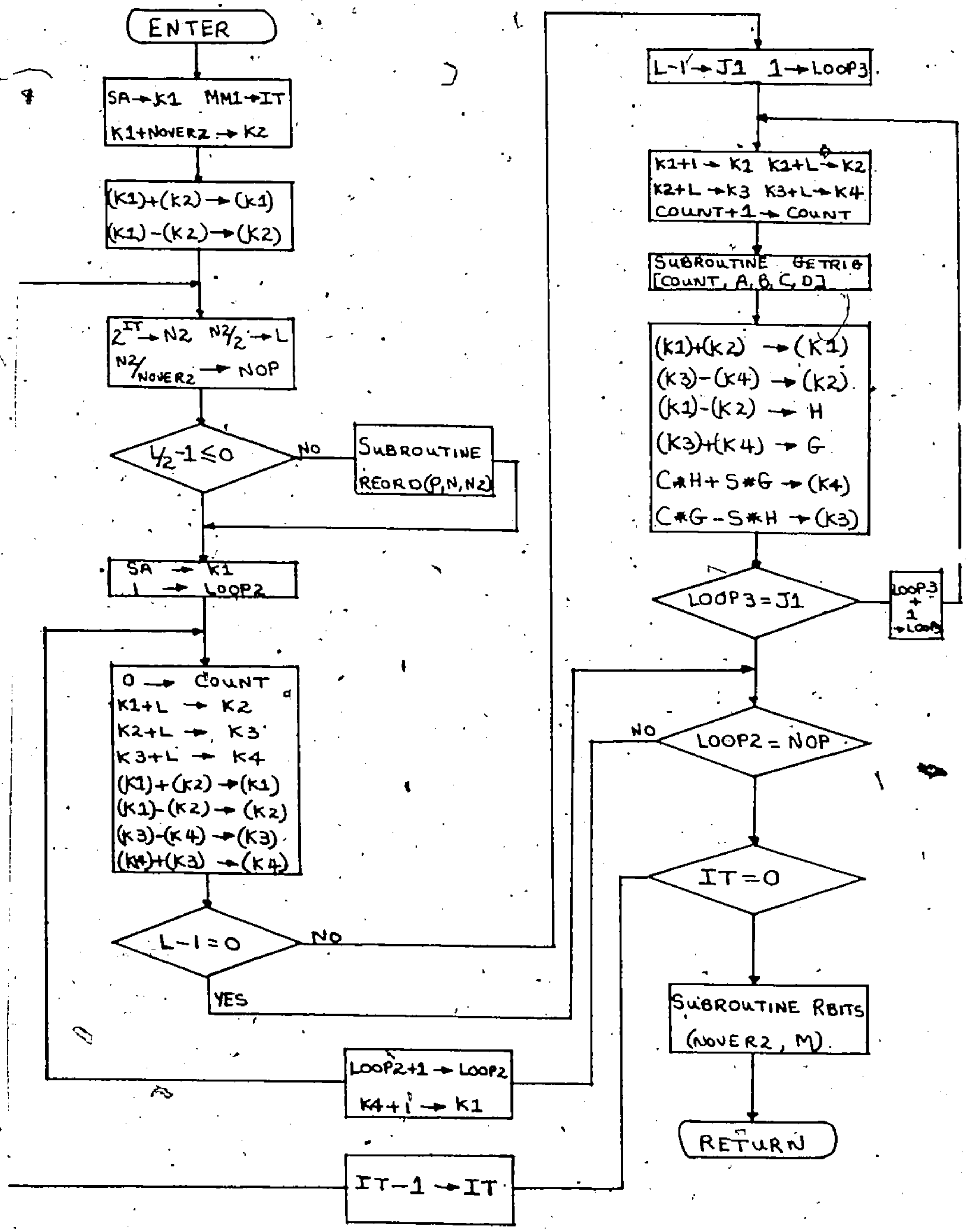
INIT - initializes counters for setting up loops.

INITI - initializes pointers for REALOP and COMPLX.

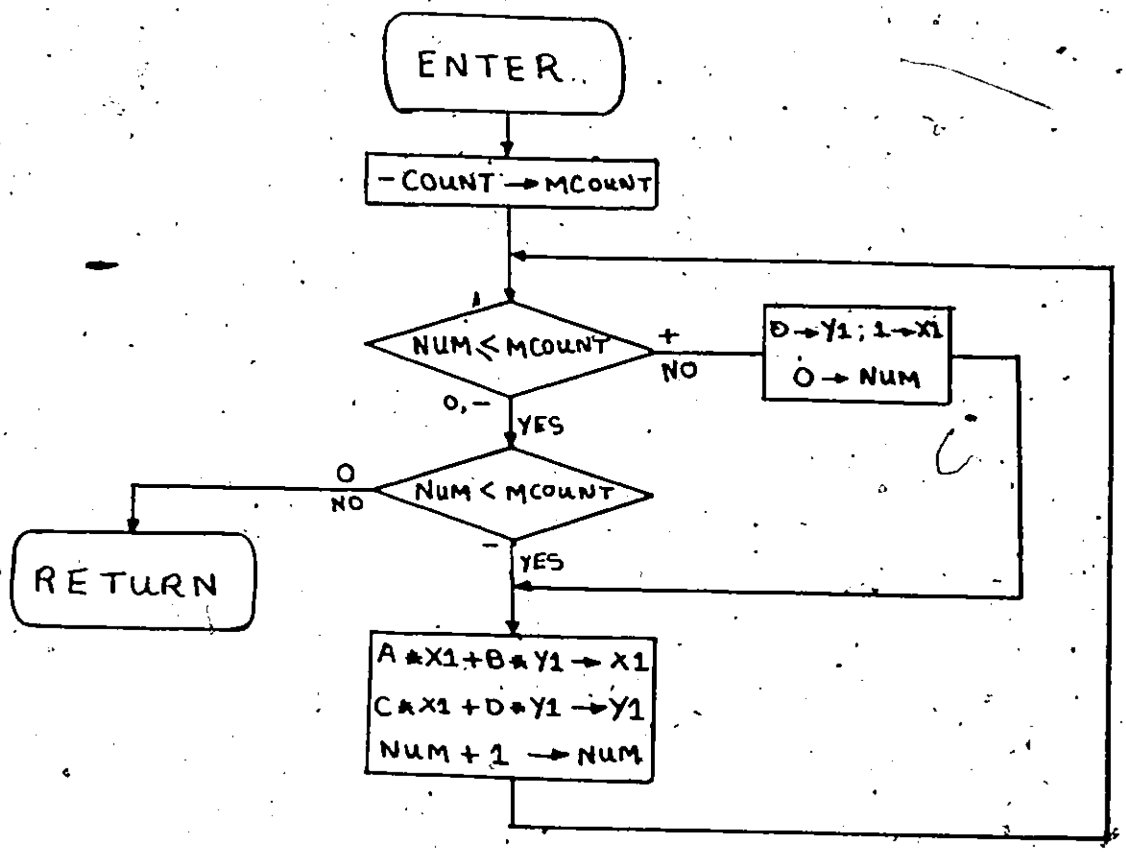
INVERT - inverts number in accumulator.

SWAP - swaps two numbers.

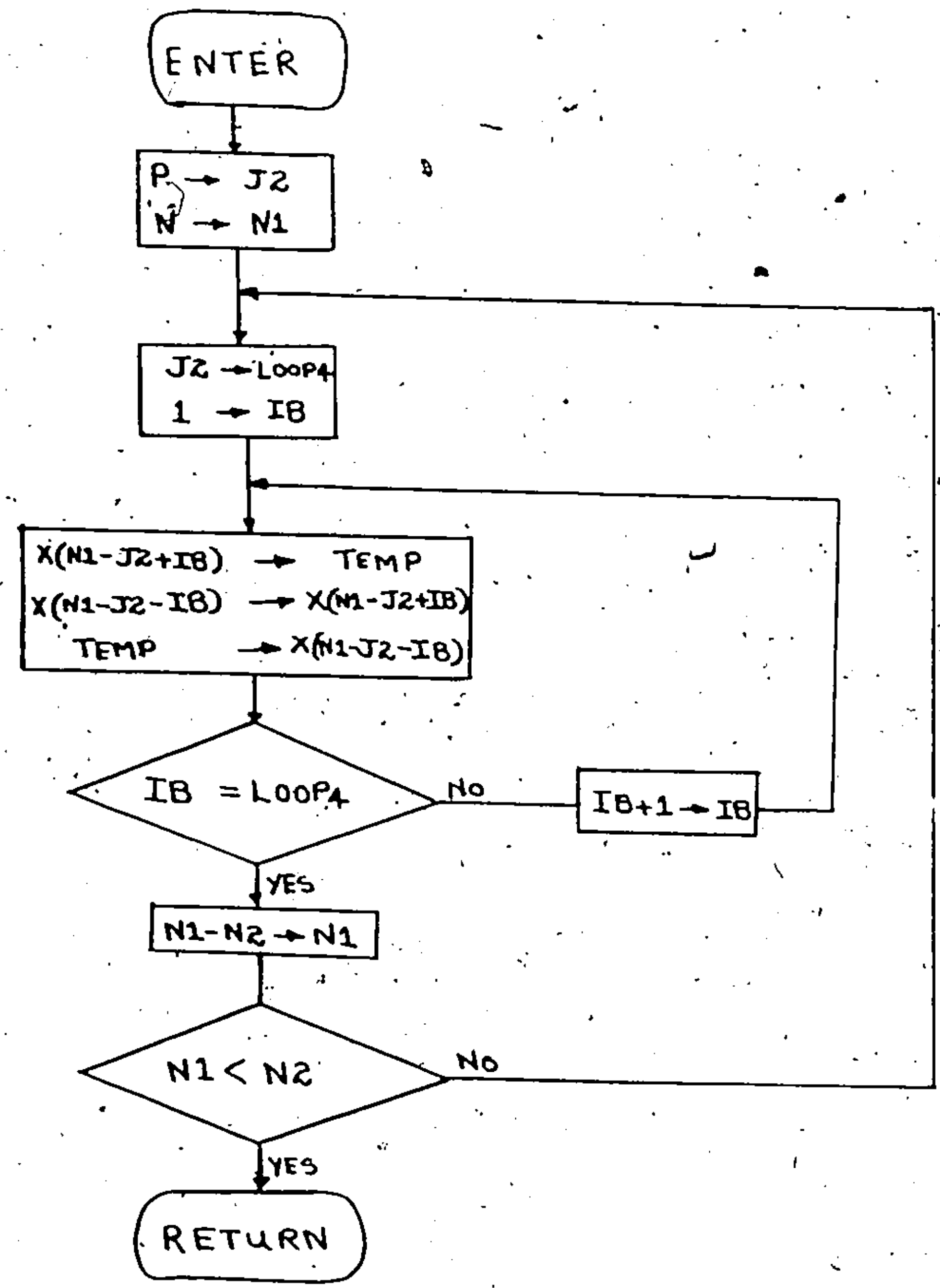

ENTER
N/2 → NOVER2
SUBROUTINE RBITS (NOVER2, M)
M-1 → MM1 ; 1 → IT
2^IT → N2   N2/2 → L
NOVER2/N2 → NOP   1 → LOOP2   SA → K1
K1+L → K2   K2+L → K3
K3+L → K4   0 → COUNT
(K1)+(K2) → (K1); (K1)-(K2) → (K2)
(K3)+(K4) → (K3); (K4)-(K3) → K4
L-1=0
NO
YES
L-1 → J1   1 → LOOP3
K1+1 → K1   K1+L → K2
K2+L → K3   K3+L → K4
COUNT+1 → COUNT
SUBROUTINE GETRIG [COUNT, A, B, C, D]
C*(K3)+S*(K4) → G
C*(K4)-S*(K3) → H
(K1) + G → (K1)
(K1) - G → (K2)
(K2) + H → (K3)
-(K2) + H → (K4)
LOOP3 = J1
NO
LOOP3+1 → LOOP3
YES
LOOP2 = NOP
NO
LOOP2+1 → LOOP2
K4+1 → K1
YES
L/2 - 1 ≤ 0
NO
SUBROUTINE REORD(F,N,N2)
YES
IT = MM1
IT+1 → IT
UPDATE A,B,C,D
SA → K1   K1+NOVER2 → K2
(K1)+(K2) → (K1)
(K1)-(K2) → (K2)
RETURN



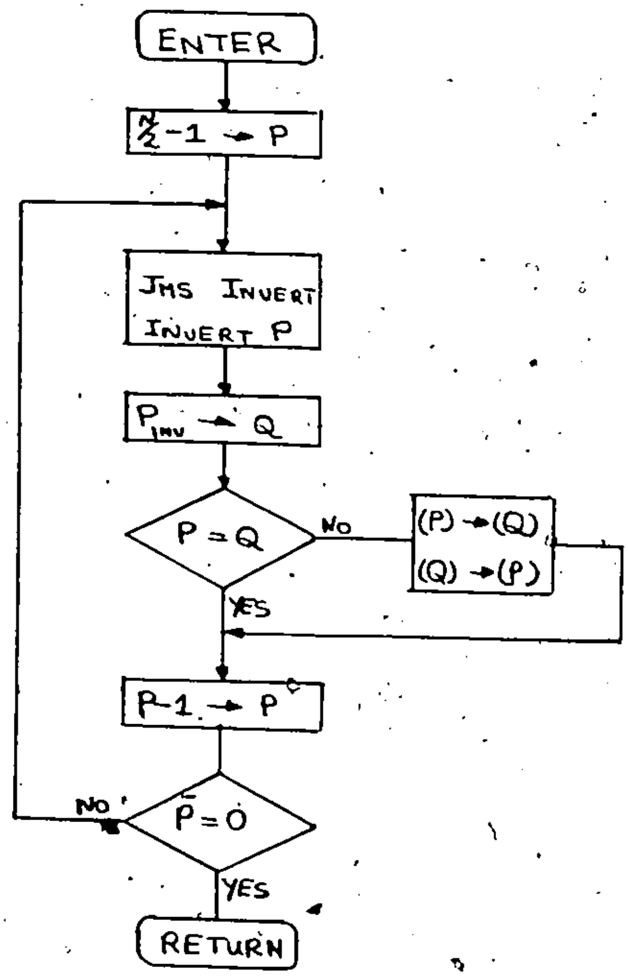
SUBROUTINE IFFT [N, M, SA, A, B, C, D]



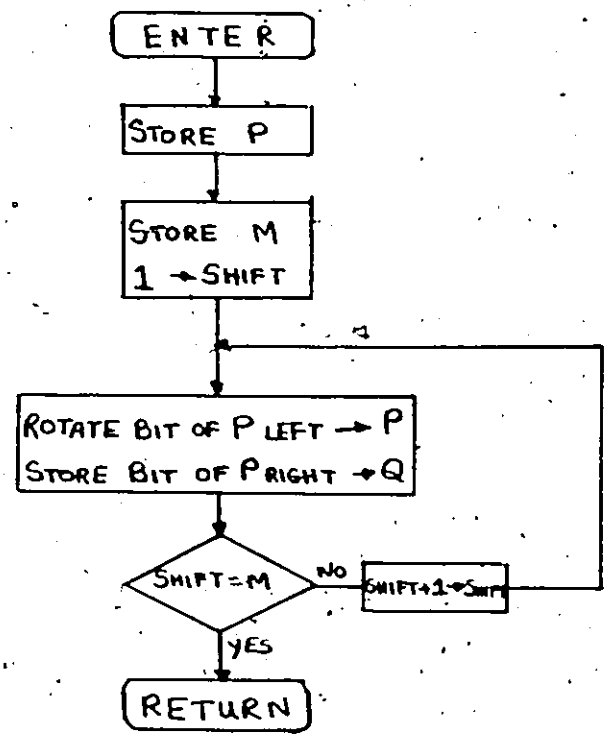
SUBROUTINE GETRIG [COUNT, A, B, C, D]



SUBROUTINE : REORD [P, N, N2]



SUBROUTINE RBITS



SUBROUTINE INVERT

```
                CDF=6201
                LSR=7417
                ASR=7415
                CAM=7621
                MQA=7501
                MQL=7421
                MUY=7405
                SHL=7413
                SCL=7403
                SCA=7441
                DVI=7407
                NMI=7411
                ACMX=6371
                RADC=6362
                ADCV=6364
                CLXK=6352
                SKXK=6321
                DXL=6302
                DYL=6312
                DXC=6301
                DIS=6304
                DYC=6311
                CLER=6351
                *0030
0030    1443    SWP,SWAP
0031    3200    MUI,MU
0032    3400    REVSEQ,SEQREV
0033    2263    SCLE,SCALE
0034    3432    ADDDUB,DUBADD
0035    1000    DOFFT,FFT
0036    0544    DOIFFT,IFFT
0037    1600    CMPLX,COMPLX
0040    1550    INIT,INIT0
0041    2000    INITI,INITI
0042    1677    OPREAL,REALOP
0043    1400    ORDER,REORD
```

```
0044  2600  SCLE1,SCALE1
0045  0473  SQU,SQRT
0046  1200  MOD,MODLUS
0047  2200  SAM,SAMPLE
0050  3521  CSINIT,INITCS
0051  2247  ERROR1,ERROR
0052  1463  BITREV,RBITS
0053  2027  TRIGET,GETRIG
0054  2400  MULT1,MULT
0055  0400  REED,READ
0056  2725  GRPH,GRAPH
0057  0000  A,0
0060  0000  AH,0
0061  0000  AL,0
0062  0000  A1,0
0063  0000  A2,0
0064  0000  BH,0
0065  0000  BL,0
0066  0000  B,0
0067  0000  B1,0
0070  0000  B2,0
0071  0000  C,0
0072  0000  C1,0
0073  0000  C2,0
0074  1330  C1330,1330
0075  0000  COS,0
0076  0000  COUNT,0
0077  0000  COUNT1,0
0100  0000  COUNT2,0
0101  0000  D,0
0102  0000  D1,0
0103  0000  D2,0
0104  0000  E,0
0105  0000  F,0
0106  0000  GH,0
0107  0000  GL,0
0110  0000  HH,0
0111  0000  HL,0
0112  0002  INC,2
0113  0000  IK1,0
0114  0000  IK2,0
0115  0000  IK3,0
0116  0000  IK4,0
0117  0000  IT,0
0120  0000  J1,0
0121  0000  K1,0
0122  0000  K2,0
0123  0212  K212,212
0124  0215  K215,215
0125  0000  K3,0
0126  0000  K4,0
```

```
0127  0000  KEEP,0
0130  0000  L,0
0131  0000  LOCAT,0
0132  0000  LOOP1,0
0133  0000  LOOP2,0
0134  0000  LOOP3,0
0135  0000  M,0
0136  7752  M26,-26
0137  7777  MONE,-1
0140  0000  MRATE,0
0141  4000  MASK,4000
0142  0000  MM1,0
0143  0000  N,0
0144  0000  N2,0
0145  0000  N0,0
0146  0000  N1,0
0147  0000  N3,0
0150  0000  N4,0
0151  0000  NOPS,0
0152  0000  NOVER2,0
0153  0000  NUM,0
0154  0000  QUAN,0
0155  0000  SA,0
0156  0000  SA1,0
0157  0000  SCALE2,0
0160  0001  SHFLAG,1
0161  0000  SHFCHK,0
0162  0000  SIN,0
0163  1354  S1354,1354
0164  0000  TEMP1,0
0165  0000  TEMP2,0
0166  0000  TEMP3,0
0167  0000  TEMPR,0
0170  0000  X,0
0171  0000  XHI,0
0172  0000  XLO,0
0173  0000  Y,0
0174  0000  YHI,0
0175  0000  YLO,0
            *0400
0400  0000  READ,0
0401  7300  CLA CLL
0402  6046  TLS
0403  4237  JMS CRLF
0404  1261  TAD M4
0405  3262  DCA DIGCTR
0406  1263  TAD TEMP12
0407  3264  DCA TEMP
0410  4245  NEWDIG,JMS LISN
0411  3664  DCA I TEMP
0412  2264  ISZ TEMP
```

```
0413  2262  ISZ DIGCTR
0414  5210  JMP NEWDIG
0415  1263  PACK,TAD TEMP12
0416  3264  DCA TEMP
0417  3271  DCA HOLD
0420  1261  TAD M4
0421  3262  DCA DIGCTR
0422  1271  DIGPCK,TAD HOLD
0423  7104  CLL RAL
0424  7006  RTL
0425  1664  TAD I TEMP
0426  1272  TAD M260
0427  3271  DCA HOLD
0430  2264  ISZ TEMP
0431  2262  ISZ DIGCTR
0432  5222  JMP DIGPCK
0433  1271  TAD HOLD
0434  5600  JMP I READ
0435  0007  MASK7,7
0436  0260  K260,260
0437  0000  CRLF,0      /CRLF
0440  1124  TAD K215
0441  4253  JMS TYPE
0442  1123  TAD K212
0443  4253  JMS TYPE
0444  5637  JMP I CRLF
0445  0000  LISN,0      /LISN
0446  6031  KSF
0447  5246  JMP.-1
0450  6036  KRB
0451  6046  TLS
0452  5645  JMP I LISN
0453  0000  TYPE,0      /TYPE
0454  6041  TSF
0455  5254  JMP.-1
0456  6046  TLS
0457  7200  CLA
0460  5653  JMP I TYPE
0461  7774  M4,7774
0462  0000  DIGCTR,0
0463  0465  TEMP12,.+2
0464  0000  TEMP,0
0465  0000  0
0466  0000  0
0467  0000  0
0470  0000  0
0471  0000  HOLD,0
0472  7520  M260,-260
0473  0000  SORT,0     /SORT
0474  7300  CLA CLL
0475  1170  TAD X
```

```
0476  7440  SZA
0477  5303  JMP.+4
0500  1173  TAD Y
0501  7450  SNA
0502  5673  JMP I SQRT
0503  7300  CLA CLL
0504  1173  TAD Y
0505  7421  MQL
0506  1170  TAD X
0507  7403  SCL
0510  0037  37
0511  7411  NMI
0512  7006  RTL
0513  7421  MQL
0514  7441  SCA
0515  1136  TAD M26
0516  7041  CIA
0517  7415  ASR
0520  0000  0
0521  1137  TAD MONE
0522  7510  SPA
0523  5331  JMP SHFT+2
0524  3327  DCA SHFT
0525  7001  IAC
0526  7413  SHL
0527  0000  SHFT,0
0530  5332  JMP.+2
0531  7001  IAC
0532  3337  DCA DIV
0533  1173  TAD Y
0534  7421  MQL
0535  1170  TAD X
0536  7407  DVI
0537  0000  DIV,0
0540  7701  MOA CLA
0541  1337  TAD DIV
0542  7110  CLL RAR
0543  5673  JMP I SQRT
0544  0000  IFFT,0      /IFFT
0545  7300  CLA CLL
0546  1155  TAD SA
0547  3121  DCA K1
0550  7001  IAC
0551  1121  TAD K1
0552  3113  DCA IK1
0553  1121  TAD K1
0554  1143  TAD N
0555  3122  DCA K2
0556  1122  TAD K2
0557  7001  IAC
0560  3114  DCA IK2
```

```
0561  1513  TAD I IK1
0562  1514  TAD I IK2
0563  7421  MQL
0564  7004  RAL
0565  1521  TAD I K1
0566  1522  TAD I K2
0567  7417  LSR
0570  0000  0
0571  3521  DCA  I K1
0572  7501  MQA
0573  3513  DCA I IK1
0574  1514  TAD I  IK2
0575  7421  MQL
0576  1522  TAD I K2
0577  4451  JMS I ERROR1
0600  3164  DCA TEMP1
0601  7501  MQA
0602  7100  CLL
0603  1513  TAD I IK1
0604  3514  DCA I IK2
0605  7004  RAL
0606  1164  TAD TEMP1
0607  1521  TAD I K1
0610  3522  DCA I K2
0611  7001  IAC
0612  3160  DCA  SHFLAG
0613  3161  DCA SHFCHK
0614  3157  DCA SCALE2
0615  4444  JMS I SCLE1
0616  1142  TAD MM1
0617  3117  ILOP1,DCA IT
0620  4440  JMS I INIT
0621  1130  TAD L
0622  7110  CLL RAR
0623  3167  DCA TEMPR
0624  7040  CMA
0625  1167  TAD TEMPR
0626  7550  SPA SNA
0627  7410  SKP
0630  4443  JMS I ORDER
0631  7201  CLA IAC
0632  7104  CLL RAL
0633  7041  CIA
0634  1117  TAD IT
0635  7510  SPA
0636  7410  SKP
0637  4450  JMS I CSINIT
0640  7300  CLA CLL
0641  1112  TAD  INC
0642  3153  DCA NUM
0643  1151  TAD NOPS
```

```
0644  7041  CIA
0645  3133  DCA LOOP2
0646  1155  TAD SA
0647  3121  DCA K1
0650  7200  ILOP2,CLA
0651  3076  DCA COUNT
0652  4441  JMS I INITI
0653  4442  JMS I OPREAL
0654  1516  TAD I IK4
0655  7421  MQL
0656  1526  TAD I K4
0657  4451  JMS I ERROR1
            PAUSE
0660  3164  DCA TEMP1
0661  7501  MQA
0662  3165  DCA TEMP2
0663  1525  TAD I K3
0664  3526  DCA I K4
0665  1515  TAD I IK3
0666  3516  DCA I IK4
0667  1164  TAD TEMP1
0670  3525  DCA I K3
0671  1165  TAD TEMP2
0672  3515  DCA I IK3
0673  7300  CLA CLL
0674  7040  CMA
0675  1130  TAD L
0676  7440  SZA
0677  7410  SKP
0700  5352  JMP IPOL2
0701  3120  DCA J1
0702  1120  TAD J1
0703  7041  CIA
0704  3134  DCA LOOP3
0705  7001  ILOP3,IAC
0706  7104  CLL RAL
0707  1121  TAD K1
0710  3121  DCA K1
0711  4441  JMS I INITI
0712  4442  JMS I OPREAL
0713  1516  TAD I IK4
0714  7421  MQL
0715  1526  TAD I K4
0716  4451  JMS I ERROR1
0717  3164  DCA TEMP1
0720  7501  MQA
0721  3165  DCA TEMP2
0722  1522  TAD I K2
0723  3526  DCA I K4
0724  1514  TAD I IK2
0725  3516  DCA I IK4
```

```
0726  1164  TAD TEMP1
0727  3522  DCA I K2
0730  1165  TAD TEMP2
0731  3514  DCA I IK2
0732  7001  IAC
0733  1076  TAD COUNT
0734  3076  DCA COUNT
0735  1076  TAD COUNT
0736  4453  JMS I TRIGET
0737  4437  JMS I CMPLX
0740  1106  TAD GH
0741  3526  DCA I K4
0742  1107  TAD GL
0743  3516  DCA I IK4
0744  1110  TAD HH
0745  3525  DCA I K3
0746  1111  TAD HL
0747  3515  DCA I IK3
0750  2134  ISZ LOOP3
0751  5305  JMP ILOP3
0752  7001  IPOL2,IAC
0753  1116  TAD IK4
0754  3121  DCA K1
0755  2133  ISZ LOOP2
0756  5250  JMP ILOP2
0757  1161  TAD SHFCHK
0760  7450  SNA
0761  2157  ISZ SCALE2
0762  3160  DCA SHFLAG
0763  3161  DCA SHFCHK
0764  7300  CLA CLL
0765  7040  CMA
0766  1117  TAD IT
0767  7440  SZA
0770  5217  JMP ILOP1
0771  4452  JMS I BITREV
0772  4444  JMS I SCLE1
0773  6201  CDF+00
0774  1436  TAD I  DOIFFT
0775  3071  DCA C
0776  6211  CDF+10
0777  5471  JMP I C
1000  0000  FFT,0        /FFT
1001  7300  CLA CLL
1002  7040  CMA
1003  1135  TAD M
1004  3142  DCA MM1
1005  7001  IAC
1006  3160  DCA SHFLAG
1007  3161  DCA SHFCHK
1010  3157  DCA SCALE2
```

```
1011   1143   TAD N
1012   7110   CLL RAR
1013   3152   DCA NOVER2
1014   4452   JMS I BITREV
1015   1142   TAD MMI
1016   7041   CIA
1017   3132   DCA LOOP1
1020   4444   JMS I SCLE1
1021   7001   IAC
1022   3117   DCA IT
1023   7200   LOP1, CLA
1024   4440   JMS I INIT
1025   1151   TAD NOPS
1026   7041   CIA
1027   3133   DCA LOOP2
1030   1112   TAD  INC
1031   3153   DCA NUM
1032   1155   TAD SA
1033   3121   DCA K1
1034   7200   LOP2, CLA
1035   3076   DCA COUNT
1036   4441   JMS I INITI
1037   4442   JMS I OPREAL
1040   7040   CMA
1041   1130   TAD L
1042   7440   SZA
1043   7410   SKP
1044   5302   JMP ALOP2
1045   3120   DCA J1
1046   1120   TAD J1
1047   7041   CIA
1050   3134   DCA LOOP3
1051   1112   LOP3,TAD INC
1052   1121   TAD K1
1053   3121   DCA K1
1054   4441   JMS I INITI
1055   7001   IAC
1056   1076   TAD COUNT
1057   3076   DCA COUNT
1060   1076   TAD COUNT
1061   4453   JMS I TRIGET
1062   4437   JMS I CMPLX
1063   1522   TAD I K2
1064   3525   DCA I K3
1065   1514   TAD I IK2
1066   3515   DCA I IK3
1067   1106   TAD GH
1070   3522   DCA I K2
1071   1107   TAD GL
1072   3514   DCA I IK2
1073   1110   TAD HH
```

```
1074  3526  DCA I K4
1075  1111  TAD HL
1076  3516  DCA I IK4
1077  4442  JMS I OPREAL
1100  2134  ISZ LOOP3
1101  5251  JMP LOP3
1102  7001  ALOP2,IAC
1103  1116  TAD IK4
1104  3121  DCA K1
1105  2133  ISZ LOOP2
1106  5234  JMP LOP2
1107  1130  TAD L
1110  7110  CLL RAR
1111  3167  DCA TEMPR
1112  7040  CMA
1113  1167  TAD TEMPR
1114  7550  SPA SNA
1115  7410  SKP
1116  4443  JMS I ORDER
1117  2117  ISZ IT
1120  7201  CLA IAC
1121  7104  CLL RAL
1122  7041  CIA
1123  1117  TAD IT
1124  4450  JMS I CSINIT
1125  1161  TAD SHFCHK
1126  7450  SNA
1127  2157  ISZ  SCALE2
1130  3160  DCA SHFLAG
1131  3161  DCA SHFCHK
1132  2132  ISZ LOOP1
1133  5223  JMP LOP1
1134  1155  TAD SA
1135  3121  DCA  K1
1136  7001  IAC
1137  1121  TAD K1
1140  3113  DCA IK1
1141  1121  TAD K1
1142  1143  TAD N
1143  3122  DCA K2
1144  7001  IAC
1145  1122  TAD K2
1146  3114  DCA IK2
1147  1513  TAD I IK1
1150  1514  TAD I IK2
1151  3164  DCA TEMP1
1152  7004  RAL
1153  1521  TAD I K1
1154  1522  TAD  I K2
1155  3165  DCA TEMP2
1156  1514  TAD I IK2
```

```
1157  7421  MQL
1160  1522  TAD I K2
1161  4451  JMS I ERROR1
1162  3166  DCA TEMP3
1163  7501  MQA
1164  7100  CLL
1165  1513  TAD I IK1
1166  3514  DCA I IK2
1167  7004  RAL
1170  1521  TAD  I K1
1171  1166  TAD TEMP3
1172  3522  DCA I K2
1173  1164  TAD TEMP1
1174  3521  DCA I K1
1175  1165  TAD TEMP2
1176  3513  DCA  I IK1
1177  5600  JMP  I FFT
1200  0000  MODLUS,0      /MODLUS
1201  7300  CLA CLL
1202  1155  TAD SA
1203  1143  TAD N
1204  3122  DCA  K2
1205  7001  IAC
1206  1122  TAD K2
1207  3114  DCA IK2
1210  3514  DCA I IK2
1211  3522  DCA I K2
1212  4444  JMS I SCLE1
1213  1152  TAD NOVER2
1214  7041  CIA
1215  3076  DCA COUNT
1216  1155  TAD SA
1217  3121  DCA K1
1220  7001  REP,IAC
1221  1121  TAD K1
1222  3113  DCA IK1
1223  1121  TAD  K1
1224  1143  TAD N
1225  3122  DCA  K2
1226  7001  IAC
1227  1122  TAD K2
1230  3114  DCA IK2
1231  1513  TAD I IK1
1232  3061  DCA AL
1233  1061  TAD AL
1234  3065  DCA BL
1235  1521  TAD I K1
1236  3060  DCA AH
1237  1060  TAD AH
1240  3064  DCA BH
1241  4454  JMS I MULTI
```

Lai

```
1242   3164   DCA TEMP1
1243   7501   MQA
1244   3165   DCA TEMP2
1245   1522   TAD I K2
1246   3060   DCA AH
1247   1060   TAD AH
1250   3064   DCA BH
1251   1514   TAD I IK2
1252   3061   DCA AL
1253   1061   TAD AL
1254   3065   DCA BL
1255   4454   JMS I MULT1
1256   3060   DCA AH
1257   7501   MQA
1260   3061   DCA AL
1261   1164   TAD TEMP1
1262   3064   DCA BH
1263   1165   TAD TEMP2
1264   3065   DCA BL
1265   4434   JMS I ADDDUB
1266   3170   DCA X
1267   7501   MQA
1270   3173   DCA Y
1271   4445   JMS I SQU
1272   3521   DCA I K1
1273   3513   DCA I IK1
1274   3522   DCA  I K2
1275   3514   DCA I IK2
1276   1121   TAD K1
1277   1112   TAD INC
1300   3121   DCA K1
1301   2076   ISZ COUNT
1302   5220   JMP REP
1303   4444   JMS I  SCLE1
1304   1152   TAD NOVER2
1305   3154   DCA QUAN
1306   4433   JMS I SCLE
1307   1152   TAD NOVER2
1310   7041   CIA
1311   4456   JMS I GRPH
1312   5600   JMP I MODLUS
              *1330     /GETRIG COS  CONSTANTS
1330   2650   2650
1331   1171   1171
1332   3544   3544
1333   0654   0654
1334   3730   3730
1335   5134   5134
1336   3766   3766
1337   1074   1074
1340   3775   3775
```

```
1341  4204  4204
1342  3777  3777
1343  3036  3036
1344  3777  3777
1345  6605  6605
1346  3777  3777
1347  7537  7537
1350  3777  3777
1351  7725  7725
1352  3777  3777
1353  7763  7763
            *1354       /GETRIG SIN CONSTANTS
1354  2650  2650
1355  1171  1171
1356  1417  1417
1357  5704  5704
1360  0617  0617
1361  4267  4267
1362  0310  0310
1363  5722  5722
1364  0144  0144
1365  3730  3730
1366  0062  0062
1367  2052  2052
1370  0031  0031
1371  1034  1034
1372  0014  0014
1373  4416  4416
1374  0006  0006
1375  2207  2207
1376  0003  0003
1377  1103  1103
            *1400
1400  0000  REORD,0    /REORD
1401  3345  DCA J2
1402  7040  CMA
1403  1143  TAD N
1404  3146  ALOP4,DCA N1
1405  1345  TAD J2
1406  7041  CIA
1407  3347  DCA LOOP4
1410  1345  TAD J2
1411  7041  CIA
1412  1146  TAD N1
1413  7104  CLL RAL
1414  1155  TAD SA
1415  3344  DCA NS
1416  7001  IAC
1417  3346  DCA IB
1420  1346  LOP4,TAD IB
1421  7104  CLL RAL
```

```
1422  1344  TAD N5
1423  3150  DCA N4
1424  1346  TAD IB
1425  7104  CLL RAL
1426  7041  CIA
1427  1344  TAD N5
1430  3147  DCA N3
1431  4243  JMS SWAP
1432  2346  ISZ IB
1433  2347  ISZ LOOP4
1434  5220  JMP LOP4
1435  1144  TAD N2
1436  7041  CIA
1437  1146  TAD N1
1440  7500  SMA
1441  5204  JMP ALOP4
1442  5600  JMP I REORD
1443  0000  SWAP,0        /SWAP
1444  1547  TAD I N3
1445  3164  DCA TEMP1
1446  1550  TAD I N4
1447  3547  DCA I N3
1450  1164  TAD TEMP1
1451  3550  DCA I N4
1452  2147  ISZ N3
1453  2150  ISZ N4
1454  1547  TAD I N3
1455  3164  DCA TEMP1
1456  1550  TAD I N4
1457  3547  DCA I N3
1460  1164  TAD TEMP1
1461  3550  DCA I N4
1462  5643  JMP I SWAP
1463  0000  RBITS,0       /RBITS
1464  7040  CMA
1465  1143  TAD N
1466  3320  DCA Q
1467  1320  REVERS,TAD Q
1470  4321  JMS INVERT
1471  3317  DCA P
1472  1317  TAD P
1473  7041  CIA
1474  1320  TAD Q
1475  7750  CLA SNA SMA
1476  5310  JMP SWAPED
1477  1317  TAD P
1500  7104  CLL RAL
1501  1155  TAD SA
1502  3147  DCA N3
1503  1320  TAD Q
1504  7104  CLL RAL
```

```
1505  1155  TAD SA
1506  3150  DCA N4
1507  4243  JMS SWAP
1510  1320  SWAPED,TAD Q
1511  7650  SNA CLA
1512  5663  JMP I RBITS
1513  7040  CMA
1514  1320  TAD Q
1515  3320  DCA Q
1516  5267  JMP REVERS
1517  0000  P,0
1520  0000  Q,0
1521  0000  INVERT,0      /INVERT
1522  3341  DCA WORD
1523  3342  DCA WORDP
1524  1135  TAD M
1525  7041  CIA
1526  3343  DCA FLIPCT
1527  1341  FLIP,TAD  WORD
1530  7110  CLL RAR
1531  3341  DCA WORD
1532  1342  TAD WORDP
1533  7004  RAL
1534  3342  DCA WORDP
1535  2343  ISZ FLIPCT
1536  5327  JMP FLIP
1537  1342  TAD WORDP
1540  5721  JMP I INVERT
1541  0000  WORD,0
1542  0000  WORDP,0
1543  0000  FLIPCT,0
1544  0000  N5,0
1545  0000  J2,0
1546  0000  IB,0
1547  0000  LOOP4,0
1550  0000  INIT0,0      /INIT0
1551  7040  CMA
1552  1117  TAD IT
1553  3371  DCA TEM5
1554  1371  TAD TEM5
1555  3361  DCA TEM6
1556  7621  CAM
1557  7001  IAC
1560  7413  SHL
1561  0000  TEM6,0
1562  3144  DCA N2
1563  1144  TAD N2
1564  7110  CLL RAR
1565  3130  DCA L
1566  7621  CAM
1567  1152  TAD NOVER2
```

```
1570  7415  ASR
1571  0000  TEM5,0
1572  3151  DCA NOPS
1573  5750  JMP I INIT0
            *1600
1600  0000  COMPLX,0        /COMPLX
1601  1174  TAD YHI
1602  3060  DCA AH
1603  1175  TAD YLO
1604  3061  DCA AL
1605  1526  TAD I K4
1606  3064  DCA BH
1607  1516  TAD I IK4
1610  3065  DCA BL
1611  4454  JMS I MULT1
1612  3164  DCA TEMP1
1613  7501  MQA
1614  3165  DCA TEMP2
1615  1171  TAD XHI
1616  3060  DCA AH
1617  1172  TAD XLO
1620  3061  DCA AL
1621  1525  TAD I K3
1622  3064  DCA BH
1623  1515  TAD I IK3
1624  3065  DCA BL
1625  4454  JMS I MULT1
1626  3166  DCA TEMP3
1627  7501  MQA
1630  7100  CLL
1631  1165  TAD TEMP2
1632  3107  DCA GL
1633  7004  RAL
1634  1164  TAD TEMP1
1635  1166  TAD TEMP3
1636  3106  DCA GH
1637  1171  TAD XHI
1640  3060  DCA AH
1641  1172  TAD XLO
1642  3061  DCA AL
1643  1526  TAD I K4
1644  3064  DCA BH
1645  1516  TAD I IK4
1646  3065  DCA BL
1647  4454  JMS I MULT1
1650  3164  DCA TEMP1
1651  7501  MQA
1652  3165  DCA TEMP2
1653  1174  TAD YHI
1654  3060  DCA AH
1655  1175  TAD YLO
```

```
1656   3061   DCA AL
1657   1525   TAD I K3
1660   3064   DCA BH
1661   1515   TAD I IK3
1662   3065   DCA BL
1663   4454   JMS I MULT1
1664   4451   JMS I ERROR1
1665   3166   DCA TEMP3
1666   7501   MQA
1667   7100   CLL
1670   1165   TAD TEMP2
1671   3111   DCA HL
1672   7004   RAL
1673   1164   TAD TEMP1
1674   1166   TAD TEMP3
1675   3110   DCA HH
1676   5600   JMP I COMPLX
1677   0000   REALOP,0        /REALOP
1700   1521   TAD I K1
1701   3060   DCA AH
1702   1513   TAD I IK1
1703   3061   DCA AL
1704   1522   TAD I K2
1705   3064   DCA BH
1706   1514   TAD I IK2
1707   3065   DCA BL
1710   4434   JMS I ADDDUB
1711   3164   DCA TEMP1
1712   7501   MQA
1713   3165   DCA TEMP2
1714   1521   TAD I K1
1715   3060   DCA AH
1716   1513   TAD I IK1
1717   3061   DCA AL
1720   1514   TAD I IK2
1721   7421   MQL
1722   1522   TAD I K2
1723   4451   JMS I ERROR1
1724   3064   DCA BH
1725   7501   MQA
1726   3065   DCA BL
1727   4434   JMS I ADDDUB
1730   3522   DCA I K2
1731   7501   MQA
1732   3514   DCA I IK2
1733   1164   TAD TEMP1
1734   3521   DCA I K1
1735   1165   TAD TEMP2
1736   3513   DCA I IK1
1737   1526   TAD I K4
```

```
1740  3060  DCA AH
1741  1516  TAD I IK4
1742  3061  DCA AL
1743  1525  TAD I K3
1744  3064  DCA BH
1745  1515  TAD I IK3
1746  3065  DCA BL
1747  4434  JMS I ADDDUB
1750  3164  DCA TEMP1
1751  7501  MQA
1752  3165  DCA TEMP2
1753  1526  TAD I K4
1754  3060  DCA AH
1755  1516  TAD I IK4
1756  3061  DCA AL
1757  1515  TAD I IK3
1760  7421  MQL
1761  1525  TAD I K3
1762  4451  JMS I ERROR1
1763  3064  DCA BH
1764  7501  MQA
1765  3065  DCA BL
1766  4434  JMS I ADDDUB
1767  3526  DCA I K4
1770  7501  MQA
1771  3516  DCA I IK4
1772  1164  TAD TEMP1
1773  3525  DCA I K3
1774  1165  TAD TEMP2
1775  3515  DCA I IK3
1776  5677  JMP I REALOP
            *2000
2000  0000  INIT1,0         /INIT1
2001  7001  IAC
2002  1121  TAD K1
2003  3113  DCA IK1
2004  1144  TAD N2
2005  1121  TAD K1
2006  3122  DCA K2
2007  7001  IAC
2010  1122  TAD K2
2011  3114  DCA IK2
2012  1144  TAD N2
2013  1122  TAD K2
2014  3125  DCA K3
2015  7001  IAC
2016  1125  TAD K3
2017  3115  DCA IK3
2020  1144  TAD N2
2021  1125  TAD K3
2022  3126  DCA K4
```

```
2023  7001  IAC
2024  1126  TAD K4
2025  3116  DCA IK4
2026  5600  JMP I INITI
2027  0000  GETRIG,0     /GETRIG
2030  7041  CIA
2031  3364  DCA MCOUNT
2032  1153  CHEK1,TAD NUM
2033  1364  TAD MCOUNT
2034  7550  SPA SNA
2035  5247  JMP CHEK2
2036  7300  CLA CLL
2037  3174  DCA  YHI
2040  3175  DCA YLO
2041  1370  TAD K7777
2042  3172  DCA XLO
2043  1367  TAD K3777
2044  3171  DCA XHI
2045  3153  DCA NUM
2046  5252  JMP.+4
2047  7640  CHEK2,SZA CLA
2050  7410  SKP
2051  5627  JMP I GETRIG
2052  1174  TAD  YHI
2053  3060  DCA AH
2054  1175  TAD YLO
2055  3061  DCA AL
2056  1062  TAD A1
2057  3064  DCA BH
2060  1063  TAD A2
2061  3065  DCA BL
2062  4454  JMS I MULTI
2063  3164  DCA TEMP1
2064  7501  MQA
2065  3165  DCA TEMP2
2066  1171  TAD XHI
2067  3060  DCA AH
2070  1172  TAD XLO
2071  3061  DCA AL
2072  1067  TAD B1
2073  3064  DCA BH
2074  1070  TAD B2
2075  3065  DCA BL
2076  4454  JMS I MULTI
2077  3166  DCA TEMP3
2100  7501  MQA
2101  7100  CLL
2102  1165  TAD TEMP2
2103  7421  MQL
2104  7004  RAL
2105  1166  TAD TEMP3
```

```
2106   1164   TAD TEMP1
2107   7510   SPA
2110   4451   JMS I ERROR1
2111   3366   DCA YTEMPH
2112   7501   MQA
2113   3365   DCA YTEMPL
2114   1174   TAD YHI
2115   3060   DCA AH
2116   1175   TAD YLO
2117   3061   DCA AL
2120   1072   TAD C1
2121   3064   DCA BH
2122   1073   TAD C2
2123   3065   DCA BL
2124   4454   JMS I MULT1
2125   3164   DCA TEMP1
2126   7501   MQA
2127   3165   DCA TEMP2
2130   1171   TAD XHI
2131   3060   DCA AH
2132   1172   TAD XLO
2133   3061   DCA AL
2134   1102   TAD D1
2135   3064   DCA BH
2136   1103   TAD D2
2137   3065   DCA BL
2140   4454   JMS I MULT1
2141   3166   DCA TEMP3
2142   7501   MQA
2143   7100   CLL
2144   1165   TAD TEMP2
2145   7421   MQL
2146   7004   RAL
2147   1166   TAD TEMP3
2150   1164   TAD TEMP1
2151   7510   SPA
2152   4451   JMS I ERROR1
2153   3171   DCA XHI
2154   7501   MQA
2155   3172   DCA XLO
2156   1365   TAD YTEMPL
2157   3175   DCA YLO
2160   1366   TAD YTEMPH
2161   3174   DCA YHI
2162   2153   ISZ NUM
2163   5232   JMP CHEK1
2164   0000   MCOUNT,0
2165   0000   YTEMPL,0
2166   0000   YTEMPH,0
2167   3777   K3777,3777
2170   7777   K7777,7777
```

```
                  *2200
2200   0000       SAMPLE,0      /SAMPLE
2201   7200       CLA
2202   6377       ACMX RADC ADCV
2203   7200       CLA
2204   1131       TAD LOCAT
2205   3017       DCA 17
2206   1154       TAD QUAN
2207   7041       CIA
2210   3077       DCA COUNT1
2211   6352       CLXK
2212   4234       JMS SAMP
2213   7500       SMA
2214   5212       JMP.-2
2215   4234       JMS SAMP
2216   7510       SPA
2217   5215       JMP.-2
2220   1145       TAD N0
2221   7041       CIA
2222   3076       DCA  COUNT
2223   4234       JMS SAMP
2224   2076       ISZ COUNT
2225   5223       JMP.-2
2226   4234       JMS SAMP
2227   3417       DCA I 17
2230   3417       DCA I 17
2231   2077       ISZ COUNT1
2232   5226       JMP.-4
2233   5600       JMP I SAMPLE
2234   0000       SAMP,0   /SAMP
2235   7300       CLA CLL
2236   1140       TAD MRATE
2237   3100       DCA COUNT2
2240   6321       SKXK
2241   5240       JMP.-1
2242   6352       CLXK
2243   2100       ISZ  COUNT2
2244   5240       JMP.-4
2245   6377       ACMX RADC ADCV
2246   5634       JMP I SAMP
2247   0000       ERROR,0     /ERROR
2250   3167       DCA TEMPR
2251   7501       MQA
2252   7100       CLL
2253   7041       CIA
2254   7421       MQL
2255   7004       RAL
2256   3127       DCA KEEP
2257   1167       TAD TEMPR
2260   7040       CMA
2261   1127       TAD KEEP
```

```
2262  5647  JMP I ERROR
2263  0000  SCALE,0    /SCALE
2264  7300  CLA CLL
2265  1155  TAD SA
2266  3121  DCA K1
2267  1154  TAD QUAN
2270  7041  CIA
2271  3076  DCA COUNT
2272  3362  DCA  BIG
2273  1521  TAD I K1
2274  7450  SNA
2275  5313  JMP.+16
2276  7500  SMA
2277  5301  JMP.+2
2300  7041  CIA
2301  1363  TAD M377
2302  7550  SPA SNA
2303  5313  JMP.+10
2304  7041  CIA
2305  1362  TAD  BIG
2306  7500  SMA
2307  5313  JMP.+4
2310  7041  CIA
2311  1362  TAD BIG
2312  3362  DCA BIG
2313  7300  CLA CLL
2314  1121  TAD K1
2315  1112  TAD INC
2316  3121  DCA K1
2317  2076  ISZ COUNT
2320  5273  JMP.-25
2321  3077  DCA COUNT1
2322  1362  TAD BIG
2323  7450  SNA
2324  5663  JMP I SCALE
2325  1364  TAD K377
2326  7110  CLL RAR
2327  1363  TAD M377
2330  2077  ISZ  COUNT1
2331  7550  SPA SNA
2332  5335  JMP.+3
2333  1364  TAD K377
2334  5326  JMP.-6
2335  7300  CLA CLL
2336  1154  TAD QUAN
2337  7041  CIA
2340  3076  DCA COUNT
2341  1155  TAD SA
2342  3121  DCA  K1
2343  7040  CMA
2344  1077  TAD COUNT1
```

```
2345  3352  DCA CT2
2346  1521  TAD I K1
2347  7450  SNA
2350  5353  JMP.+3
2351  7415  ASR
2352  0000  CT2,0
2353  3521  DCA I  K1
2354  1121  TAD K1
2355  1112  TAD INC
2356  3121  DCA  K1
2357  2076  ISZ  COUNT
2360  5343  JMP.-15
2361  5663  JMP I SCALE
2362  0000  BIG,0.
2363  7401  M377,-377
2364  0377  K377,377
            *2400
2400  0000  MULT,0        /MULT
2401  7100  CLL
2402  7621  CAM
2403  1060  TAD AH
2404  7440  SZA
2405  5212  JMP.+5
2406  1061  TAD  AL
2407  7440  SZA
2410  5212  JMP.+2
2411  5600  JMP I MULT
2412  7300  CLA CLL
2413  1064  TAD BH
2414  7440  SZA
2415  5222  JMP.+5
2416  1065  TAD BL
2417  7440  SZA
2420  5222  JMP.+2
2421  5600  JMP I MULT
2422  7300  CLA  CLL
2423  1060  TAD  AH
2424  0141  AND MASK
2425  1064  TAD BH
2426  7430  SZL
2427  5243  JMP SAMSGN
2430  7004  RAL
2431  7630  SZL CLA
2432  5234  JMP DIFSGN
2433  5243  JMP SAMSGN
2434  4251  DIFSGN,JMS MULTIP
2435  7510  SPA
2436  4451  JMS I  ERROR1
2437  7413  SHL
2440  0000  0
2441  4451  JMS  I ERROR1
```

```
2442  5600  JMP I MULT
2443  4251  SAMSGN,JMS MULTIP
2444  7510  SPA
2445  4451  JMS I ERROR1
2446  7413  SHL
2447  0000  0
2450  5600  JMP I MULT
2451  0000  MULTIP,0      /MULTIP
2452  7100  CLL
2453  7621  CAM
2454  1060  TAD AH
2455  7700  SMA CLA
2456  5266  JMP.+10
2457  1061  TAD AL
2460  7421  MQL
2461  1060  TAD  AH
2462  4451  JMS I ERROR1
2463  3060  DCA AH
2464  7501  MQA
2465  3061  DCA AL
2466  7300  CLA CLL
2467  1064  TAD BH
2470  7700  SMA CLA
2471  5301  JMP.+10
2472  1065  TAD BL
2473  7421  MQL
2474  1064  TAD BH
2475  4451  JMS  I ERROR1
2476  3064  DCA BH
2477  7501  MQA
2500  3065  DCA  BL
2501  7100  CLL
2502  7621  CAM
2503  1061  TAD AL
2504  3310  DCA TEM1
2505  1065  TAD BL
2506  7421  MQL
2507  7405  MUY
2510  0000  TEM1,0
2511  3366  DCA TEMPL
2512  7621  CAM
2513  1061  TAD AL
2514  3320  DCA TEM2
2515  1064  TAD BH
2516  7421  MQL
2517  7405  MUY
2520  0000  TEM2,0
2521  3370  DCA TEMPHI
2522  7501  MQA
2523  3371  DCA TEMPLO
2524  7100  CLL
```

```
2525  7621  CAM
2526  1060  TAD AH
2527  3333  DCA TEM3
2530  1065  TAD BL
2531  7421  MQL
2532  7405  MUY
2533  0000  TEM3,0
2534  3367  DCA TEMPCH
2535  7501  MQA
2536  1371  TAD TEMPLO
2537  1366  TAD TEMPL
2540  7200  CLA
2541  7004  RAL
2542  1370  TAD TEMPHI
2543  1367  TAD TEMPCH
2544  3371  DCA TEMPLO
2545  7004  RAL
2546  3127  DCA KEEP
2547  7621  CAM
2550  1060  TAD AH
2551  3355  DCA TEM4
2552  1064  TAD BH
2553  7421  MQL
2554  7405  MUY
2555  0000  TEM4,0
2556  3370  DCA TEMPHI
2557  7501  MQA
2560  1371  TAD TEMPLO
2561  7421  MQL
2562  7004  RAL
2563  1127  TAD KEEP
2564  1370  TAD TEMPHI
2565  5651  JMP I MULTIP
2566  0000  TEMPL,0
2567  0000  TEMPCH,0
2570  0000  TEMPHI,0
2571  0000  TEMPLO,0
            *2600
2600  0000  SCALE1,0      /SCALE1
2601  1143  TAD N
2602  7041  CIA
2603  3076  DCA COUNT
2604  3324  DCA T
2605  1323  TAD K26
2606  3322  DCA MAX
2607  1155  TAD SA
2610  3121  DCA K1
2611  7001  IAC
2612  1121  TAD K1
2613  3113  DCA IK1
2614  1521  TAD I K1
```

```
2615  7640  SZA CLA
2616  5223  JMP.+5
2617  1513  TAD I IK1
2620  7640  SZA CLA
2621  5223  JMP.+2
2622  5243  JMP.+21
2623  1513  TAD I IK1
2624  7421  MQL
2625  1521  TAD I K1
2626  7403  SCL
2627  0037  37
2630  7411  NMI
2631  7200  CLA
2632  7441  SCA
2633  3324  DCA  T
2634  1322  TAD MAX
2635  7041  CIA
2636  1324  TAD T
2637  7700  SMA  CLA
2640  5243  JMP.+3
2641  1324  TAD T
2642  3322  DCA  MAX
2643  1121  TAD K1
2644  1112  TAD INC
2645  3121  DCA K1
2646  2076  ISZ COUNT
2647  5211  JMP.-36
2650  1322  TAD MAX
2651  7650  SNA CLA
2652  5600  JMP I SCALE1
2653  1143  TAD N
2654  7041  CIA
2655  3076  DCA COUNT
2656  1155  TAD SA
2657  3121  DCA K1
2660  7001  RET,IAC
2661  1121  TAD K1
2662  3113  DCA IK1
2663  7040  CMA
2664  1322  TAD MAX
2665  3312  DCA  T1
2666  1312  TAD T1
2667  3300  DCA T2
2670  1521  TAD I K1
2671  7700  SMA  CLA
2672  5306  JMP.+14
2673  1513  TAD  I IK1
2674  7421  MQL
2675  1521  TAD I K1
2676  4451  JMS I ERROR1
2677  7413  SHL
```

```
2700   0000   T2,0
2701   4451   JMS I ERROR1
2702   3521   DCA I K1
2703   7501   MQA
2704   3513   DCA I IK1
2705   5314   JMP.+7
2706   1513   TAD I IK1
2707   7421   MQL
2710   1521   TAD I K1
2711   7413   SHL
2712   0000   T1,0
2713   5302   JMP.-11
2714   1112   TAD INC
2715   1121   TAD K1
2716   3121   DCA K1
2717   2076   ISZ  COUNT
2720   5260   JMP RET
2721   5600   JMP I SCALE1
2722   0000   MAX,0
2723   0026   K26,26
2724   0000   T,0
2725   0000   GRAPH,0        /GRAPH
2726   3164   DCA TEMP1
2727   1164   TAD TEMP1
2730   1375   TAD K1000
2731   7700   SMA CLA
2732   5336   JMP DSPLY
2733   1375   TAD K1000
2734   7041   CIA
2735   3164   DCA TEMP1
2736   1155   DSPLY,TAD SA
2737   3131   DCA LOCAT
2740   1164   TAD TEMP1
2741   3076   DCA COUNT
2742   3170   DCA  X
2743   6351   CLER
2744   1170   TAD X
2745   6303   DXC DXL
2746   7300   CLA CLL
2747   1531   TAD I LOCAT
2750   6317   DYC DYL DIS
2751   2170   ISZ  X
2752   7300   CLA CLL
2753   1131   TAD  LOCAT
2754   1112   TAD INC
2755   3131   DCA LOCAT
2756   2076   ISZ COUNT
2757   5343   JMP.-14
2760   7604   LAS
2761   1137   TAD MONE
2762   7640   SZA CLA
```

```
2763  5365  JMP.+2
2764  5725  JMP I GRAPH
2765  1164  TAD TEMP1
2766  3077  DCA COUNT1
2767  7000  NOP
2770  2077  ISZ COUNT1
2771  5367  JMP.-2
2772  7300  CLA CLL
2773  7604  LAS
2774  5336  JMP DSPLY
2775  1000  K1000,1000
            *3000
3000  0000  INPT,0      /INPT
3001  4455  JMS I REED
3002  3143  DCA  N
3003  4455  JMS I REED
3004  3135  DCA M
3005  4455  JMS I REED
3006  3140  DCA MRATE
3007  7040  CMA
3010  1155  TAD SA
3011  3131  DCA LOCAT
3012  1143  TAD N
3013  3154  DCA QUAN
3014  4447  JMS I SAM
3015  1143  TAD N
3016  7041  CIA
3017  4456  JMS I  GRPH
3020  5600  JMP I INPT
3021  0000  INPT1,0     /INPT1
3022  4455  JMS I REED
3023  3146  DCA N1
3024  4455  JMS I REED
3025  3144  DCA N2
3026  4455  JMS I REED
3027  3140  DCA MRATE
3030  5621  JMP I INPT1
3031  0000  INITAL,0    /INITAL
3032  7200  CLA
3033  1156  TAD SA1
3034  3155  DCA SA
3035  7040  CMA
3036  1156  TAD SA1
3037  3131  DCA LOCAT
3040  1146  TAD N1
3041  3147  DCA N3
3042  4273  JMS DATA
3043  4455  JMS I REED
3044  7450  SNA
3045  5250  JMP.+3
3046  3145  DCA N0
```

```
3047   4273   JMS DATA
3050   7040   CMA
3051   1143   TAD N
3052   1143   TAD N
3053   1156   TAD SA1
3054   3131   DCA LOCAT
3055   1143   TAD N
3056   7104   CLL RAL
3057   1156   TAD SA1
3060   3155   DCA SA
3061   1144   TAD N2
3062   3147   DCA N3
3063   3145   DCA N0
3064   4273   JMS  DATA
3065   4455   JMS I REED
3066   7450   SNA
3067   5272   JMP.+3
3070   3145   DCA N0
3071   4273   JMS DATA
3072   5631   JMP  I INITAL
3073   0000   DATA,0     /DATA
3074   7300   CLA CLL
3075   1147   TAD  N3
3076   3154   DCA QUAN
3077   4447   JMS I SAM
3100   1143   TAD N
3101   7041   CIA
3102   1147   TAD N3
3103   7450   SNA
3104   5312   JMP.+6
3105   3076   DCA  COUNT
3106   3417   DCA I 17
3107   3417   DCA I 17
3110   2076   ISZ COUNT
3111   5306   JMP.-3
3112   1143   TAD N
3113   7041   CIA
3114   4456   JMS I GRPH
3115   5673   JMP I DATA
3116   0000   PERD,0     /PERD
3117   7621   CAM
3120   1377   TAD  S
3121   7440   SZA
3122   5326   JMP CHANGE
3123   7040   CMA
3124   1144   TAD N2
3125   5333   JMP.+6
3126   7300   CHANGE,CLA CLL
3127   7040   CMA
3130   1144   TAD N2
3131   1377   TAD S
```

```
3132  3146  DCA N1
3133  1146  TAD N1
3134  7500  SMA
3135  5346  JMP POS
3136  1373  TAD K4000
3137  7440  SZA
3140  7402  HLT
3141  1373  TAD K4000
3142  3143  DCA N
3143  1375  TAD K13
3144  3135  DCA  M
3145  5716  JMP I PERD
3146  7403  POS,SCL
3147  0037  37
3150  7411  NMI
3151  1376  TAD M2000
3152  7450  SNA
3153  5355  JMP.+2
3154  7201  CLA IAC
3155  3127  DCA KEEP
3156  7441  SCA
3157  7041  CIA
3160  1374  TAD K11
3161  1127  TAD KEEP
3162  3370  DCA SHFT2
3163  7001  IAC
3164  1370  TAD SHFT2
3165  3135  DCA M
3166  7001  IAC
3167  7413  SHL
3170  0000  SHFT2,0
3171  3143  DCA N
3172  5716  JMP I PERD
3173  4000  K4000,4000
3174  0011  K11,11
3175  0013  K13,13
3176  6000  M2000,-2000
3177  0000  S,0
            *3200
3200  0000  MU,0        /MU
3201  1156  TAD SA1
3202  3121  DCA K1
3203  1143  TAD N
3204  3144  DCA N2
3205  4441  JMS I INITI
3206  7604  LAS
3207  7440  SZA
3210  4364  JMS ADJUST
3211  1521  TAD I K1
3212  3060  DCA AH
3213  1513  TAD I IK1
```

```
3214  3061  DCA AL
3215  1525  TAD I K3
3216  3064  DCA BH
3217  1515  TAD I IK3
3220  3065  DCA BL
3221  4454  JMS I MULTI
3222  3521  DCA I K1
3223  7501  MQA
3224  3513  DCA I IK1
3225  1522  TAD I K2
3226  3060  DCA AH
3227  1514  TAD I IK2
3230  3061  DCA AL
3231  1526  TAD I K4
3232  3064  DCA BH
3233  1516  TAD I IK4
3234  3065  DCA BL
3235  4454  JMS I MULTI
3236  3522  DCA I K2
3237  7501  MQA
3240  3514  DCA I IK2
3241  7040  CMA
3242  1152  TAD NOVER2
3243  7041  CIA
3244  3076  DCA COUNT
3245  1112  RET1,TAD INC
3246  1121  TAD K1
3247  3121  DCA K1
3250  4441  JMS I INITI
3251  7604  LAS
3252  7440  SZA
3253  4364  JMS ADJUST
3254  1521  TAD I K1
3255  3060  DCA AH
3256  1513  TAD I IK1
3257  3061  DCA AL
3260  1525  TAD I K3
3261  3064  DCA BH
3262  1515  TAD I IK3
3263  3065  DCA BL
3264  4454  JMS I MULTI
3265  3164  DCA TEMP1
3266  7501  MQA
3267  3165  DCA TEMP2
3270  1522  TAD I K2
3271  3060  DCA AH
3272  1514  TAD I IK2
3273  3061  DCA AL
3274  1526  TAD I K4
3275  3064  DCA BH
3276  1516  TAD I IK4
```

```
3277  3065  DCA BL
3300  4454  JMS I MULTI
3301  3060  DCA AH
3302  7501  MQA
3303  3061  DCA AL
3304  1164  TAD TEMP1
3305  3064  DCA BH
3306  1165  TAD TEMP2
3307  3065  DCA BL
3310  4434  JMS I ADDDUB
3311  3166  DCA TEMP3
3312  7501  MQA
3313  3167  DCA TEMPR
3314  1521  TAD I K1
3315  3060  DCA AH
3316  1513  TAD I IK1
3317  3061  DCA AL
3320  1526  TAD I K4
3321  3064  DCA BH
3322  1516  TAD I IK4
3323  3065  DCA BL
3324  4454  JMS I MULTI
3325  4451  JMS I ERROR1
3326  3164  DCA TEMP1
3327  7501  MQA
3330  3165  DCA TEMP2
3331  1522  TAD I K2
3332  3060  DCA AH
3333  1514  TAD I IK2
3334  3061  DCA AL
3335  1525  TAD I K3
3336  3064  DCA BH
3337  1515  TAD I IK3
3340  3065  DCA BL
3341  4454  JMS I MULTI
3342  3060  DCA AH
3343  7501  MQA
3344  3061  DCA AL
3345  1164  TAD TEMP1
3346  3064  DCA BH
3347  1165  TAD TEMP2
3350  3065  DCA BL
3351  4434  JMS I ADDDUB
3352  3522  DCA  I K2
3353  7501  MQA
3354  3514  DCA  I IK2
3355  1166  TAD TEMP3
3356  3521  DCA I K1
3357  1167  TAD TEMPR
3360  3513  DCA I IK1
3361  2076  ISZ COUNT
```

```
3362  5245  JMP RET1
3363  5600  JMP I MU
3364  0000  ADJUST,0      /ADJUST
3365  7300  CLA CLL
3366  1521  TAD I K1
3367  3525  DCA I K3
3370  1513  TAD I IK1
3371  3515  DCA I IK3
3372  1522  TAD I K2
3373  3526  DCA I K4
3374  1514  TAD I IK2
3375  3516  DCA I IK4
3376  5764  JMP I  ADJUST
            *3400
3400  0000  SEQREV,0      /SEQREV
3401  1155  TAD SA
3402  3062  DCA A1
3403  7040  CMA
3404  1143  TAD N
3405  7104  CLL RAL
3406  1155  TAD SA
3407  3067  DCA B1
3410  1152  TAD NOVER2
3411  7041  CIA
3412  3076  DCA COUNT
3413  3072  DCA C1
3414  1072  TAD C1
3415  7104  CLL RAL
3416  1062  TAD A1
3417  3147  DCA N3
3420  1072  TAD C1
3421  7104  CLL RAL
3422  7041  CIA
3423  1067  TAD B1
3424  3150  DCA N4
3425  4430  JMS I SWP
3426  2072  ISZ C1
3427  2076  ISZ COUNT
3430  5214  JMP.-14
3431  5600  JMP I SEQREV
3432  0000  DUBADD,0   /DUBADD
3433  7621  CAM
3434  7300  CLA CLL
3435  1160  TAD SHFLAG
3436  7650  SNA CLA
3437  5300  JMP ADDWOS
3440  1061  TAD AL
3441  7415  ASR
3442  0000  0
3443  7200  CLA
3444  1065  TAD BL
```

```
3445  7415  ASR
3446  0000  0
3447  7200  CLA
3450  7501  MQA
3451  7004  RAL
3452  7060  CMA CML
3453  7720  CLA  SMA SNL
3454  7001  IAC
3455  3127  DCA KEEP
3456  1061  TAD AL
3457  7421  MQL
3460  1060  TAD AH
3461  7415  ASR
3462  0000  0
3463  3060  DCA AH
3464  7501  MQA
3465  3061  DCA AL
3466  1065  TAD BL
3467  7421  MQL
3470  1064  TAD BH
3471  7415  ASR
3472  0000  0
3473  3064  DCA BH
3474  7501  MQA
3475  3065  DCA BL
3476  7100  CLL
3477  1127  TAD KEEP
3500  1065  ADDWOS,TAD BL
3501  1061  TAD AL
3502  7421  MQL
3503  7004  RAL
3504  1060  TAD AH
3505  1064  TAD BH
3506  3060  DCA AH
3507  1060  TAD AH
3510  7510  SPA
3511  7041  CIA
3512  7004  RAL
3513  7700  SMA CLA
3514  5317  JMP NOTNOR
3515  7001  IAC
3516  3161  DCA SHFCHK
3517  1060  NOTNOR,TAD AH
3520  5632  JMP I DUBADD
3521  0000  INITCS,0        /INITCS
3522  6201  CDF+00
3523  7104  CLL RAL
3524  3057  DCA A
3525  1074  TAD C1330
3526  1057  TAD A
3527  3075  DCA COS
```

```
3530   1163   TAD S1354
3531   1057   TAD A
3532   3162   DCA SIN
3533   1475   TAD I COS
3534   3062   DCA A1
3535   7001   IAC
3536   1075   TAD COS
3537   3075   DCA COS
3540   1475   TAD I COS
3541   3063   DCA A2
3542   1062   TAD A1
3543   3102   DCA D1
3544   1063   TAD A2
3545   3103   DCA D2
3546   1562   TAD I SIN
3547   3067   DCA B1
3550   7001   IAC
3551   1162   TAD SIN
3552   3162   DCA SIN
3553   1562   TAD I SIN
3554   3070   DCA B2
3555   1070   TAD B2
3556   7421   MQL
3557   1067   TAD B1
3560   4451   JMS I ERROR1
3561   3072   DCA C1
3562   7501   MQA
3563   3073   DCA C2
3564   6211   CDF+10
3565   5721   JMP I INITCS
              *0200
0200   7300   BEG,CLA CLL
0201   6211   CDF+10
0202   4455   JMS I REED
0203   7510   SPA
0204   5250   JMP CORREL
0205   7650   SNA CLA
0206   5243   JMP CONV
0207   4667   FFTS,JMS I  INPUT
0210   4455   JMS I REED
0211   7450   SNA
0212   5220   JMP.+6
0213   3145   DCA N0
0214   4447   JMS I SAM
0215   1143   TAD N
0216   7041   CIA
0217   4456   JMS I GRPH
0220   4435   JMS I DOFFT
0221   4446   JMS I MOD
0222   7300   CLA CLL
0223   7040   CMA
```

```
0224  1155  TAD SA
0225  3131  DCA LOCAT
0226  4447  JMS I SAM
0227  1143  TAD N
0230  7041  CIA
0231  4456  JMS I GRPH
0232  4435  JMS I DOFFT
0233  4436  JMS I DOIFFT
0234  1143  TAD  N
0235  3154  DCA QUAN
0236  4433  JMS I SCLE
0237  1143  TAD N
0240  7041  CIA
0241  4456  JMS I GRPH
0242  5200  JMP BEG
0243  4670  CONV,JMS I INPUT1
0244  4666  JMS I PERIOD
0245  4671  JMS I INTAL
0246  4272  JMS CONVL
0247  5200  JMP BEG
0250  4670  CORREL,JMS I INPUT1
0251  4666  JMS I PERIOD
0252  4671  JMS I INTAL
0253  1143  TAD N      /REPLACE BY JMP.+2
0254  7104  CLL RAL     IF SEQUENCE 1 IS TO BE REVERSED
0255  1156  TAD SA1
0256  3155  DCA SA
0257  4432  JMS I REVSEQ
0260  7000  NOP
0261  1143  TAD N
0262  7041  CIA
0263  4456  JMS I GRPH
0264  4272  JMS CONVL
0265  5200  JMP BEG
0266  3116  PERIOD,PERD
0267  3000  INPUT,INPT
0270  3021  INPUT1,INPT1
0271  3031  INTAL,INITAL
0272  0000  CONVL,0     /CONVL
0273  1156  TAD SA1
0274  3155  DCA SA
0275  4435  JMS I DOFFT
0276  7300  CLA CLL
0277  7604  LAS
0300  7440  SZA
0301  5307  JMP.+6
0302  1143  TAD N
0303  7104  CLL RAL
0304  1156  TAD SA1
0305  3155  DCA SA
0306  4435  JMS I DOFFT
```

```
0307   7300   CLA CLL
0310   4431   JMS I MUI
0311   1156   TAD SAI
0312   3155   DCA SA
0313   4436   JMS I DOIFPT
0314   1143   TAD N
0315   3154   DCA QUAN
0316   4433   JMS I SCLE
0317   1143   TAD N
0320   7041   CIA
0321   4456   JMS I GRPH
0322   5672   JMP I CONVL
```

REFERENCES

(1) Bergland G. -"A Fast Fourier Transform for Real Valued Input". Comm. ACM Vol. 11 / No. 10 / Oct. 1968 Pp. 703 - 710.

(2) 1970, 1971 Proceedings on the Application of Walsh Functions. Symposium and Workshop.

(3) Pitassi D. - "Fast Convolution using the Walsh Transform". Application of Walsh Functions. 1971 Proceedings. Pp. 130-133.

(4) Gold & Rader - "Digital Processing of Signals". McGraw-Hill Inc. 1969.

(5) Cooley J. - "An algorithm for the machine computation of Complex Fourier Series". Math. Comp.19 April 1965, Pp 297 - 301.

(6) Stegun I. A. - "Handbook of Mathematical Functions". National Bureau of Standards.

VITA AUCTORIS

1950 Born on January 3rd, in Barataria, Trinidad.

1967 Completed High School at St. Mary's College, Port-of-Spain, Trinidad.

1970 Graduated from the University of the West Indies, St. Augustine, Trinidad with the degree of B.Sc. in Electrical Engineering.

1973 Candidate for the degree of M.A.Sc. in Electrical Engineering at the University of Windsor.