Spring 1979

An Introduction to the Winograd Discrete Fourier Transform

Janice S. Agnello
University of Central Florida

ABSTRACT

This paper illustrates Winograd's approach to computing the Discrete Fourier Transform (DFT). This new approach changes the DFT into a cyclic convolution of 2 sequences, and illustrates shortcuts for computing this cyclic convolution. This method is known to reduce the number of multiplies required to about 20% less than the number of multiplies used by the techniques of the Fast Fourier Transform.

Three approaches are discussed, one for prime numbers, one for products of primes, and lastly one for powers of odd primes. For powers of 2 Winograd's algorithm is, in general, inefficient and best if it is not used.

A computer simulation is illustrated for the 35 point transform and its execution time is compared with that of the Fast Fourier Transform algorithm for 32 points.

AN INTRODUCTION TO THE WINOGRAD DISCRETE FOURIER TRANSFORM

BY

JANICE S. AGNELLO
B.A., Hunter College of the City University of New York, 1974

RESEARCH REPORT

Submitted in partial fulfillment of the requirements for the degree of Master of Science: Operations Research in the Graduate Studies Program of the College of Engineering at the University of Central Florida; Orlando, Florida

Summer Quarter
1979

TABLE OF CONTENTS

INTRODUCTION

This paper will introduce a new approach to computing the Discrete Fourier Transform (DFT). This new approach was developed by Dr. Schmuel Winograd and hence coined the 'Winograd Discrete Fourier Transform' (WDFT). This new approach was developed ideally for computing the DFT for a prime number of points. The underlying principle is to reorder the input elements in such a fashion that the DFT has the appearence of a cyclic convolution of 2 sequences. From this point onward the analysis is that of illustrating shortcuts for computing this cyclic convolution.

Prime Number Theory plays a large role in this technique because for every prime number there exists a 'primitive root' which is utilized in regenerating the input sequence, such that the DFT becomes a cyclic convolution of 2 sequences. Winograd, in ref [1] ,states that computing the cyclic convolution of 2 sequences of N points $(W_0,W_1,W_2,...,W_{N-1})$ and $(x_0,x_1,x_2,...,x_{N-1})$ is equivalent to finding the coefficients of the following polynomial of z:

$$(0.1)\quad (W_0 + W_1 z^1 + W_2 z^2 + \ldots + W_{N-1} z^{N-1}) \cdot (x_0 + x_1 z^1 + x_2 z^2 + \ldots + x_{N-1} z^{N-1}) \bmod (z^N - 1)$$

Computing the coefficients of the above polynomial is the crux of this paper. For large N this computation is cumbersome. Kolba, in ref [2], describes an alternate method for computing (0.1) which uses the Chinese Remainder Theorem. This method is illustrated in sect 1.3.

Winograd, in ref. [3], proves that the minimum number of multiplies required to compute (0.1) is:

$$2(N) - k$$

where k is the number of irreducible factors of (z^N-1). Although the minimum number of multiplies required to compute (0.1) is known, finding these multiplies is a completely different task. At present these multiply algorithms have only been determined for 2,3,5,and 7 point transforms. Other algorithms can be obtained by using combinations of these. Appendix B [4] gives the algorithms for computing the 2,3,4,5,7,8,9,and 16 point transforms.

There are 4 basic approaches fo the WDFT depending on the characteristics of the number of the input samples; namely if it is a prime, product of primes, power of an odd prime, or power of 2. This paper will address 3 of these cases by going through its theoretical development, summary of steps needed for computation and lastly an illustration. The First Chapter will deal with the theoretical development of the WDFT for a prime number of points. Chapter Two will cover the WDFT for a number of points equal to the product of primes. In Chapter Three the powers of odd primes will be dealt with. Powers of the unique even prime 2 uses a different approach. For this last case the structure of the WDFT becomes inefficient computational wise and is not a good approach. This case will be omitted.

Appendix C illustrates a computer simulation of a 35 point WDFT which utilizes the algorithms in Appendix B for the 5 and 7 point transforms. This simulation does the 5 point transform 7 times then puts this output into the 7 point transform and does this 5 times. The input data is of the form,

$2000.[\cos(2\pi 3t) + j\sin(2\pi 3t)] \quad t=0,1/35,2/35,3/35,\ldots,34/35$

and $j = \sqrt{-1}$.

The execution time of the 35 point WDFT simulation is contrasted with the execution time of a Fast Fourier Transform (FFT) for 32 points. The program setup and conclusions are discussed in the last Chapter.

CHAPTER I

WDFT FOR A PRIME NUMBER OF POINTS

1.1 Definition of the DFT

The Discrete Fourier Transform of N points is of the form

$$(1.1.1) \quad y_k = \sum_{n=0}^{N-1} x_n \, W^{nk} \qquad k = 0,1,2,\ldots,N-1$$

where W^1 is the N^{th} root of unity i.e.,

$$W^1 = e^{-j\left(\frac{2\pi}{N}\right)}$$

The matrix representation is :

(1.1.2)

$$\begin{bmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \\ \cdot \\ \cdot \\ y_{N-1} \end{bmatrix} = \begin{bmatrix} W^0 & W^0 & W^0 & W^0 & \ldots & W^0 \\ W^0 & W^1 & W^2 & W^3 & \ldots & W^{(N-1)} \\ W^0 & W^2 & W^4 & W^6 & \ldots & W^{2(N-1)} \\ W^0 & W^3 & W^6 & W^9 & \ldots & W^{3(N-1)} \\ \cdot & \cdot & \cdot & \cdot & & \cdot \\ \cdot & \cdot & \cdot & \cdot & & \cdot \\ W^0 & W^{(N-1)} & W^{2(N-1)} & W^{3(N-1)} & \ldots & W^{(N-1)(N-1)} \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \\ \cdot \\ \cdot \\ x_{N-1} \end{bmatrix}$$

Since the first row and first column are all 1's (1.1.1) can be rewritten as:

$$(1.1.3) \quad y_0 = \sum_{n=0}^{N-1} x$$

$$y_k = x_0 + \bar{y}_k \qquad k = 1,2,3,\ldots,N-1$$

where $\bar{y}_k$ is the (N-1) by (N-1) lower right portion of the matrix in (1.1.2) i.e.,

$$(1.1.4) \quad \bar{y}_k = \sum_{n=1}^{N-1} x_n \, W^{kn} \qquad k = 1,2,3,\ldots,N-1$$

Winograd's technique will be applied to eq (1.1.4).

1.2 Generating the Reordered Set of Elements for a Prime Number of Points N and Restructuring the Matrix

Since N is prime, all the non zero integers less than N form a cyclic group under multiplication modulo N. The generator, g, of the group is called a primitive root of N. A table of primitive roots for all primes less than 5000 is given in reference [5].

The first step is to permute the input data as follows

$$
\begin{array}{l}
g^{1} \pmod N \\
g^{2} \pmod N \\
g^{3} \pmod N \\
\cdot \\
\cdot \\
\cdot \\
g^{N-1} \pmod N
\end{array}
\tag{1.2.1}
$$

Rewrite (1.1.2) using the above permutation and eliminating the 1^{st} row and 1^{st} column.

(1.2.2)

$$\begin{bmatrix} y_{g^1 \bmod N} \\ y_{g^2 \bmod N} \\ y_{g^3 \bmod N} \\ \cdot \\ \cdot \\ \cdot \\ y_{g^{N-1} \bmod N} \end{bmatrix} = \begin{bmatrix} W^{g^{(1+1)} \bmod N} & \cdots & W^{g^{(1+(N-1))} \bmod N} \\ W^{g^{(2+1)} \bmod N} & \cdots & W^{g^{(2+(N-1))} \bmod N} \\ W^{g^{(3+1)} \bmod N} & \cdots & W^{g^{(3+(N-1))} \bmod N} \\ \cdot & & \cdot \\ \cdot & & \cdot \\ \cdot & \cdot & \cdot \\ W^{g^{((N-1)+1)} \bmod N} & \cdots & W^{g^{(2(N-1))} \bmod N} \end{bmatrix} \begin{bmatrix} x_{g^1 \bmod N} \\ x_{g^2 \bmod N} \\ x_{g^3 \bmod N} \\ \cdot \\ \cdot \\ \cdot \\ x_{g^{N-1} \bmod N} \end{bmatrix}$$

By ordering the data according to the exponents of g, (1.1.4) can be changed into a circular convolution for any prime N [2].

For example, if N=7 the DFT structure is:

$$\begin{bmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \end{bmatrix} = \begin{bmatrix} W^0 & W^0 & W^0 & W^0 & W^0 & W^0 & W^0 \\ W^0 & W^1 & W^2 & W^3 & W^4 & W^5 & W^6 \\ W^0 & W^2 & W^4 & W^6 & W^8 & W^{10} & W^{12} \\ W^0 & W^3 & W^6 & W^9 & W^{12} & W^{15} & W^{18} \\ W^0 & W^4 & W^8 & W^{12} & W^{16} & W^{20} & W^{24} \\ W^0 & W^5 & W^{10} & W^{15} & W^{20} & W^{25} & W^{30} \\ W^0 & W^6 & W^{12} & W^{18} & W^{24} & W^{30} & W^{36} \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{bmatrix}$$

where $W^1 = e^{-j(\frac{2\pi}{7})}$

using the properties of complex exponentials gives:

$$W^1 = W^8 = W^{15} = \ldots = W^{7n+1} \qquad n=0,1,2,3,\ldots$$
$$W^2 = W^9 = W^{16} = \ldots = W^{7n+2}$$
$$W^3 = W^{10} = W^{17} = \ldots = W^{7n+3}$$
$$W^4 = W^{11} = W^{18} = \ldots = W^{7n+4}$$
$$W^5 = W^{12} = W^{19} = \ldots = W^{7n+5}$$
$$W^6 = W^{13} = W^{20} = \ldots = W^{7n+6}$$

rewrite the W matrix of the DFT using the above reduction of exponents.

(1.2.3)

$$\begin{bmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \end{bmatrix} = \begin{bmatrix} W^0 & W^0 & W^0 & W^0 & W^0 & W^0 & W^0 \\ W^0 & W^1 & W^2 & W^3 & W^4 & W^5 & W^6 \\ W^0 & W^2 & W^4 & W^6 & W^1 & W^3 & W^5 \\ W^0 & W^3 & W^6 & W^2 & W^5 & W^1 & W^4 \\ W^0 & W^4 & W^1 & W^5 & W^2 & W^6 & W^3 \\ W^0 & W^5 & W^3 & W^1 & W^6 & W^4 & W^2 \\ W^0 & W^6 & W^5 & W^4 & W^3 & W^2 & W^1 \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{bmatrix}$$

Convert (1.2.3) into a system of equations of the form of (1.1.3), i.e. eliminate the 1st row and 1st column.

$$(1.2.4)\quad \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \end{bmatrix} = \begin{bmatrix} W^1 & W^2 & W^3 & W^4 & W^5 & W^6 \\ W^2 & W^4 & W^6 & W^1 & W^3 & W^5 \\ W^3 & W^6 & W^2 & W^5 & W^1 & W^4 \\ W^4 & W^1 & W^5 & W^2 & W^6 & W^3 \\ W^5 & W^3 & W^1 & W^6 & W^4 & W^2 \\ W^6 & W^5 & W^4 & W^3 & W^2 & W^1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{bmatrix}$$

From reference [5] 3 is found to be a primitive root for the set of integers 1,2,3,4,5,6.

$$3^1 \pmod 7 \equiv 3$$
$$3^2 \pmod 7 \equiv 2$$
$$3^3 \pmod 7 \equiv 6$$
$$3^4 \pmod 7 \equiv 4$$
$$3^5 \pmod 7 \equiv 5$$
$$3^6 \pmod 7 \equiv 1$$

Rewrite the y and x vectors using this reordering and change the W matrix so as to preserve eq (1.2.4)

$$\begin{bmatrix} y_3 \\ y_2 \\ y_6 \\ y_4 \\ y_5 \\ y_1 \end{bmatrix} = \begin{bmatrix} W^2 & W^6 & W^4 & W^5 & W^1 & W^3 \\ W^6 & W^4 & W^5 & W^1 & W^3 & W^2 \\ W^4 & W^5 & W^1 & W^3 & W^2 & W^6 \\ W^5 & W^1 & W^3 & W^2 & W^6 & W^4 \\ W^1 & W^3 & W^2 & W^6 & W^4 & W^5 \\ W^3 & W^2 & W^6 & W^4 & W^5 & W^1 \end{bmatrix} \cdot \begin{bmatrix} x_3 \\ x_2 \\ x_6 \\ x_4 \\ x_5 \\ x_1 \end{bmatrix}$$

Notice, that in rearranging the W matrix it becomes of the form,

$$W_{i,j} = W^{g^{(i+j)}\mathrm{mod}\ N}$$

and that the above is a cyclic convolution of $(W^2, W^6, W^4, W^5, W^1, W^3)$ and $(x_3, x_2, x_6, x_4, x_5, x_1)$.

1.3 Computations on a Cyclic Convolution Matrix

The basis of computation for eqs (1.2.2) is on the following property:

To cyclicly convolve the sequences,

$h_0, h_1, h_2, h_3, \ldots, h_{N-1}$

and

$x_0, x_1, x_2, x_3, \ldots, x_{N-1}$

one only needs to find the N coefficients of the polymonial[2]

(1.3.1) $Y(z) = H(z)X(z) \bmod(z^N - 1)$

where,

$$X(z) = x_0 + \sum_{k=1}^{N-1} x_{N-k} \cdot z^k$$

$$H(z) = \sum_{k=0}^{N-1} h_k \cdot z^k$$

Applying the above to matrix notation one can express the cyclic convolution of two sequences of N points $(h_0, h_1, h_2, \ldots, h_{N-1})$ and $(x_0, x_1, x_2, \ldots x_{N-1})$ as follows [6]:

(1.3.2)
$$\begin{bmatrix} h_0 & h_1 & h_2 & h_3 \cdots & h_{N-1} \\ h_1 & h_2 & h_3 & h_4 \cdots & h_0 \\ h_2 & h_3 & h_4 & h_5 \cdots & h_1 \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ h_{N-1} & h_0 & h_1 & h_2 \cdots & h_{N-2} \end{bmatrix} \cdot \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ \cdot \\ \cdot \\ x_{N-1} \end{bmatrix}$$

From (1.3.1) one knows that (1.3.2) is the system of coefficients of the polynomial

$$(h_0+h_1z+h_2z^2+\ldots+h_{N-1}z^{N-1})\cdot(x_0+x_{N-1}z+\ldots+x_1z^{N-1})\bmod(z^N-1)$$

As an illustration consider the cyclic convolution with N = 3.

$$\begin{bmatrix} y_0 \\ y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} h_0 & h_1 & h_2 \\ h_1 & h_2 & h_0 \\ h_2 & h_0 & h_1 \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ x_2 \end{bmatrix}$$

working this out long hand gives:

$$(1.3.3)\quad \begin{bmatrix} y_0 \\ y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} h_0x_0 + h_1x_1 + h_2x_2 \\ h_1x_0 + h_2x_1 + h_0x_2 \\ h_2x_0 + h_0x_1 + h_1x_2 \end{bmatrix}$$

Now compare this with the system of coefficients of the polynomial in (1.3.1)

$$(h_0+h_1z+h_2z^2)\cdot(x_0+x_2z^1+x_1z^2)\bmod(z^3-1)$$

$$\begin{aligned} = h_0x_0 &+ h_0x_2z^1 + h_0x_1z^2 \\ &+ h_1x_0z^1 + h_1x_2z^2 + h_1x_1z^3 \\ &+ h_2x_0z^2 + h_2x_2z^3 + h_2x_1z^4 \quad \bmod(z^3-1) \end{aligned}$$

$$= h_0x_0 + (h_0x_2 + h_1x_0)z^1 + (h_0x_1 + h_1x_2 + h_2x_0)z^2 + (h_1x_1 + h_2x_2)z^3 + h_2x_1z^4 \pmod{(z^3-1)}$$

After dividing by z^3-1 the coefficients of the remainder will give eqs (1.3.3)

$$\text{Quotient: } h_2x_1z^1 + (h_1x_1 + h_2x_2)$$

$$z^3-1 \;\Big|\; h_2x_1z^4 + (h_1x_1 + h_2x_2)z^3 + (h_0x_1 + h_1x_2 + h_2x_0)z^2 + (h_0x_2 + h_1x_0)z^1 + h_0x_0$$

$$\underline{+h_2x_1z^4 \qquad - h_2x_1z^1}$$

$$(h_1x_1 + h_2x_2)z^3 + (h_0x_1 + h_1x_2 + h_2x_0)z^2 + (h_0x_2 + h_1x_0 + h_2x_1)z^1 + h_0x_0$$

$$\underline{+(h_1x_1 + h_2x_2)z^3 \qquad - (h_1x_1 + h_2x_2)}$$

$$(h_2x_0 + h_0x_1 + h_1x_2)z^2 + (h_1x_0 + h_2x_1 + h_0x_2)z^1 + (h_0x_0 + h_1x_1 + h_2x_2)z^0$$

Notice that the:

coefficient of z^0 is y_0 of eq (1.3.3)

coefficient of z^1 is y_1 of eq (1.3.3)

coefficient of z^2 is y_2 of eq (1.3.3)

For large N the above method for computing (1.3.1) is very cumbersome and time consuming. To reduce the number of steps required for computation ,$Y(z)$ is decomposed into k simplier parts using the polynomial version of the Chinese Remainder Theorem [2].

If $Q_i(z)$ are irreducible relatively prime polynomials with rational coefficients such that

$$z^N - 1 = \prod_{i=1}^{k} Q_i(z) \quad .$$

then the set of congruences

$$(1.3.4) \qquad Y_i(z) \equiv H_i(z)\, X_i(z) \bmod Q_i(z) \qquad i=1,2,3,\ldots,k$$

where

$$H_i(z) \equiv H(z) \bmod Q_i(z) \qquad i=1,2,3,\ldots,k$$

$$X_i(z) \equiv X(z) \bmod Q_i(z) \qquad i=1,2,3,\ldots,k$$

has a unique solution:

$$Y(z) = \sum_{i=1}^{k} Y_i(z)\, S_i(z) \quad \bmod(z^N-1)$$

$S_i(z)$ is defined as follows:

$$S_i(z) \equiv 1 \bmod Q_i(z) \qquad i=1,2,3,\ldots,k$$

$$S_i(z) \equiv 0 \bmod Q_j(z) \qquad \text{for all } i \neq j$$

$S_i(z)$ can be constructed using:

$$S_i(z) = T_i(z) \cdot R_i(z)$$

where,

$$T_i(z) = z^N - 1 / Q_i(z)$$

$$R_i(z) \equiv [T_i(z)]^{-1} \bmod Q_i(z)$$

1.4 Winograds Theorem on the Minimum Number of Multiplications Required to Compute the Circular Convolution of Two Length N Sequences

Let ,

$$P_n = u^n + a_1u^1 + a_2u^2 + \ldots + a_{n-1}u^{n-1} = \prod_{i=1}^{k} Q_i$$

be a polynomial with coefficients in a field G (where all the Q_i's are pairwise relatively prime).

Let,

$$R_n = x_1u^1 + x_2u^2 + x_3u^3 + \ldots + x_{n-1}u^{n-1}$$

$$S_n = y_1u^1 + y_2u^2 + y_3u^3 + \ldots + y_{n-1}u^{n-1}$$

be two polynomials with indeterminant coefficients.

The minumum number of multiplications needed to compute the coefficients of,

$$T_p = R_n \cdot S_n \bmod P_n$$

is $2 \cdot n - k$

See references [3] and [4] for the proof.

1.5 Summary of Steps Needed to Compute the WDFT for a Prime Number of Points

This section will use the prime number of points equal to N+1.

Let, $(x_0, x_1, x_2, x_3, \ldots, x_N)$ denote the input sample

and $(y_0, y_1, y_2, y_3, \ldots, y_N)$ denote the output.

The DFT is of the form,

$$y_k = \sum_{i=0}^{N} W^{ik} x_i \qquad k = 0,1,2,3,\ldots,N$$

where,

$$W^1 = e^{-j\left(\frac{2\pi}{N+1}\right)}$$

$$\begin{bmatrix} y_0 \\ y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & \ldots & 1 \\ 1 & W^1 & W^2 & \ldots & W^N \\ 1 & W^2 & W^4 & \ldots & W^{2(N)} \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & W^{1(N)} & W^{2(N)} & \ldots & W^{(N)(N)} \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ \vdots \\ x_N \end{bmatrix}$$

i) Since the first row and first column of the W matrix have all 1's, the N by N sub matrix will be dealt with.

ii) Find a primitive root, g, for prime number N+1 using reference [5].

iii) Generate the reordered set of subscripts for the input and output elements using:

$$g^i \bmod (N+1) \qquad i = 1,2,3,\ldots,N$$

use the following reordering mapping:

$$x_0 \dashrightarrow x_{g^1 \bmod (N+1)}$$
$$x_1 \dashrightarrow x_{g^2 \bmod (N+1)}$$
$$x_2 \dashrightarrow x_{g^3 \bmod (N+1)}$$
$$\vdots$$
$$x_{N-1} \dashrightarrow x_{g^N \bmod (N+1)}.$$

$$w_0 \dashrightarrow w_{g^{(2+N)} \bmod (N+1)}$$
$$w_1 \dashrightarrow w_{g^{(2+(N-1))} \bmod (N+1)}$$
$$w_2 \dashrightarrow w_{g^{(2+(N-2))} \bmod (N+1)}$$
$$w_3 \dashrightarrow w_{g^{(2+(N-3))} \bmod (N+1)}$$
$$\vdots$$
$$w_{N-1} \dashrightarrow w_{g^{(2+(N-(N-1)))} \bmod (N+1)}$$

$$y_0 \dashrightarrow y_{g^1 \bmod (N+1)}$$
$$y_1 \dashrightarrow y_{g^2 \bmod (N+1)}$$
$$y_2 \dashrightarrow y_{g^3 \bmod (N+1)}$$
$$\vdots$$
$$y_{N-1} \dashrightarrow y_{g^N \bmod (N+1)}$$

iv) Form the following polynomials in z

$$X(z) = \sum_{k=0}^{N-1} x_k z^k$$

$$W(z) = \sum_{k=0}^{N-1} W_k z^k$$

Note: The W's were specifically reordered in descending powers of g so the above polynomials would have identical structures. If $W_{i-1} \dashrightarrow W_{g^i \bmod (N+1)}$ then, $W(z)$ would be $W_0 + \sum_{k=1}^{N-1} W_k z^{N-k}$.

The coefficients of the polynomial

$$(W_0 + W_1 z^1 + W_2 z^2 + \ldots + W_{N-1} z^{N-1}) \cdot$$
$$(x_0 + x_1 z^1 + x_2 z^2 + \ldots + W_{N-1} z^{N-1}) \bmod (z^N - 1)$$

are the values of $(y_0, y_1, y_2, y_3, \ldots, y_{N-1})$

i.e., the coefficient of:

z^{N-1}	is	y_1
z^{N-2}	is	y_2
z^{N-3}	is	y_3
	⋮	
z^1	is	y_{N-1}
z^0	is	y_0

v) Evaluate

$$Y(z) \equiv W(z)\ X(z) \bmod (z^N-1)$$

using the Chinese Remainder Theorem [2]

$$(1.5.1)\quad Y(z) \equiv \left[\ \sum_{i=1}^{k} Y_i(z)\ S_i(z)\ \right] \bmod (z^N-1)$$

where,

$$Y_i(z) \equiv W_i(z)\ X_i(z) \bmod Q_i(z) \qquad i = 1,2,3,\ldots,k$$

with k being the number of irreducible relatively prime polynomials over rational coefficients, such that

$$z^N-1 = \prod_{i=1}^{k} Q_i(z)$$

and,

$$(1.5.2)\quad W_i(z) \equiv W(z) \bmod Q_i(z)$$

$$X_i(z) \equiv X(z) \bmod Q_i(z) \qquad k = 1,2,3,\ldots,k$$

After computing $W_i(z)$ and $X_i(z)$ define intermediate multiply steps,

$$m_1, m_2, m_3, \ldots, m_L$$

where L, the number of multiplies, is defined by Winograd's Theorem to be,

$$L = 2N - k$$

Next, compute $S_i(z)$ $i = 1,2,\ldots,k$

$$S_i(z) \equiv 1 \bmod Q_i(z) \qquad i = 1,2,3,\ldots,k$$

$$S_i(z) \equiv 0 \bmod Q_i(z) \qquad \text{for all } i \neq j$$

$S_i(z)$ can be computed using Euclid's Algorithm for Polynomials [7].

(1.5.3) $S_i(z) = T_i(z) \cdot R_i(z) \qquad i = 1,2,3,\ldots,k$

where,

(1.5.4) $T_i(z) = z^N - 1 \;/\; Q_i(z) \qquad i = 1,2,\ldots,k$

$R_i(z) \equiv [\ T_i(z)\ ]^{-1} \bmod Q_i(z)$

1.6 Example of a 7 Point WDFT

This section will go through the entire process for computing the Discrete Fourier Transform for a prime number of points, 7, using Winograds technique.

Note: N+1 = 7 will be used throughout this example.

The 7 point DFT is as follows

$$\bar{y}_i = \sum_{i=0}^{6} W^{in} \bar{x}_i \qquad n = 0,1,2,3,4,5,6$$

where,

$$W^1 = e^{-j(\frac{2\pi}{7})}$$

Note: the bars are used over x, and y to distinguish from the reordered set of x's and y's.

The matrix representation is:

i)

$$\begin{bmatrix} \bar{y}_0 \\ \bar{y}_1 \\ \bar{y}_2 \\ \bar{y}_3 \\ \bar{y}_4 \\ \bar{y}_5 \\ \bar{y}_6 \end{bmatrix} = \begin{bmatrix} W^0 & W^0 & W^0 & W^0 & W^0 & W^0 & W^0 \\ W^0 & W^1 & W^2 & W^3 & W^4 & W^5 & W^6 \\ W^0 & W^2 & W^4 & W^6 & W^1 & W^3 & W^5 \\ W^0 & W^3 & W^6 & W^2 & W^5 & W^1 & W^4 \\ W^0 & W^4 & W^1 & W^5 & W^2 & W^6 & W^3 \\ W^0 & W^5 & W^3 & W^1 & W^6 & W^4 & W^2 \\ W^0 & W^6 & W^5 & W^4 & W^3 & W^2 & W^1 \end{bmatrix} \cdot \begin{bmatrix} \bar{x}_0 \\ \bar{x}_1 \\ \bar{x}_2 \\ \bar{x}_3 \\ \bar{x}_4 \\ \bar{x}_5 \\ \bar{x}_6 \end{bmatrix} \qquad (1.6.1)$$

Convert the 6×6 augmented matrix to a cyclic convolution matrix.

ii) Using reference [5] g = 3 is found to be a primitive root for prime number 7.

iii) Generate $g^i \bmod (N+1)$ $\qquad i = 1,2,3,4,5,6$

$$3 \equiv 3^1 \bmod 7$$
$$2 \equiv 3^2 \bmod 7$$
$$6 \equiv 3^3 \bmod 7$$
$$4 \equiv 3^4 \bmod 7$$
$$5 \equiv 3^5 \bmod 7$$
$$1 \equiv 3^6 \bmod 7$$

Use the following reordering mapping:

(1.6.2)

$x_0 \dashrightarrow \bar{x}_3$	$y_0 \dashrightarrow \bar{y}_3$	$W_0 \dashrightarrow W^2$
$x_1 \dashrightarrow \bar{x}_2$	$y_1 \dashrightarrow \bar{y}_2$	$W_1 \dashrightarrow W^3$
$x_2 \dashrightarrow \bar{x}_6$	$y_2 \dashrightarrow \bar{y}_6$	$W_2 \dashrightarrow W^1$
$x_3 \dashrightarrow \bar{x}_4$	$y_3 \dashrightarrow \bar{y}_4$	$W_3 \dashrightarrow W^5$
$x_4 \dashrightarrow \bar{x}_5$	$y_4 \dashrightarrow \bar{y}_5$	$W_4 \dashrightarrow W^4$
$x_5 \dashrightarrow \bar{x}_1$	$y_5 \dashrightarrow \bar{y}_1$	$W_5 \dashrightarrow W^6$

Note : the reordering for the W's was obtained from

$$W_0 \text{ --- } W^{g^8 \bmod 7} = W^2$$
$$W_1 \dashrightarrow W^{g^7 \bmod 7} = W^3$$
$$W_2 \dashrightarrow W^{g^6 \bmod 7} = W^1$$
$$W_3 \dashrightarrow W^{g^5 \bmod 7} = W^5 \qquad \text{with } g = 3$$
$$W_4 \dashrightarrow W^{g^4 \bmod 7} = W^4$$
$$W_5 \dashrightarrow W^{g^3 \bmod 7} = W^6$$

Rearrange the $\bar{x}$'s and $\bar{y}$'s and restructure the W matrix so as to be consistent with eq (1.6.1).

$$(1.6.3)\quad \begin{bmatrix} \bar{y}_3 \\ \bar{y}_2 \\ \bar{y}_6 \\ \bar{y}_4 \\ \bar{y}_5 \\ \bar{y}_1 \end{bmatrix} = \begin{bmatrix} W^2 & W^6 & W^4 & W^5 & W^1 & W^3 \\ W^6 & W^4 & W^5 & W^1 & W^3 & W^2 \\ W^4 & W^5 & W^1 & W^3 & W^2 & W^6 \\ W^5 & W^1 & W^3 & W^2 & W^6 & W^4 \\ W^1 & W^3 & W^2 & W^6 & W^4 & W^5 \\ W^3 & W^2 & W^6 & W^4 & W^5 & W^1 \end{bmatrix} \begin{bmatrix} \bar{x}_3 \\ \bar{x}_2 \\ \bar{x}_6 \\ \bar{x}_4 \\ \bar{x}_5 \\ \bar{x}_1 \end{bmatrix}$$

Notice that in reordering the W matrix a circular convolution matrix is obtained.

Substituting (1.6.2) into (1.6.3) gives :

$$\begin{bmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \end{bmatrix} = \begin{bmatrix} W_0 & W_5 & W_4 & W_3 & W_2 & W_1 \\ W_5 & W_4 & W_3 & W_2 & W_1 & W_0 \\ W_4 & W_3 & W_2 & W_1 & W_0 & W_5 \\ W_3 & W_2 & W_1 & W_0 & W_5 & W_4 \\ W_2 & W_1 & W_0 & W_5 & W_4 & W_2 \\ W_1 & W_0 & W_5 & W_4 & W_3 & W_2 \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix}$$

iv) The right hand side of the above equation is a cyclic convolution of $(W_0, W_5, W_4, W_3, W_2, W_1)$ with $(x_0, x_1, x_2, x_3, x_4, x_5)$

Let

$$X(z) = \sum_{i=0}^{5} x_i z^i$$

$$W(z) = \sum_{i=0}^{5} W_i z^i$$

then the system of coefficients of the polynomial

$$Y(z) = W(z)\ X(z)\ \mathrm{mod}(z^N-1) \equiv (W_0 + W_1 z^1 + W_2 z^2 + W_3 z^3 + W_4 z^4 + W_5 z^5)\cdot (x_0 + x_1 z^1 + x_2 z^2 + x_3 z^3 + x_4 z^4 + x_5 z^5)\ \mathrm{mod}(z^6-1)$$

gives the values of y_i $i = 0,1,2,3,4,5$

i.e., the coefficient of :

z^5 gives y_1

z^4 gives y_2

z^3 gives y_3

z^2 gives y_4

z^1 gives y_5

z^0 gives y_0

v) Evaluate $Y(z) = W(z)X(z)\ \mathrm{mod}\ (z^N-1)$

First find the factors of z^6-1

$$z^6-1 = (z+1)\ (z-1)\ (z^2+z+1)\ (z^2-z+1)$$

$Q_1 = z + 1$

$Q_2 = z - 1$

$Q_3 = z^2 + z + 1$

$Q_4 = z^2 - z + 1$

Note: since there are 4 factors , $k = 4$ in Winograds Theorem.

Next, using eqs (1.5.1) and (1.5.2) find the intermediate polynomials X_i, W_i, Y_i $i = 1,2,3,4$

$$X_i(z) = X(z)\ \mathrm{mod}\ Q_i(z) \qquad i = 1,2,3,4$$

The output will be expressed with superscripts and subscripts.

The superscript denotes the polynomial number and the subscript orders the output from a given polynomial.

$$X_1(z) = (x_0 + x_1 z^1 + x_2 z^2 + x_3 z^3 + x_4 z^4 + x_5 z^5) \bmod (z+1)$$
$$X_1(z) = (x_0 - x_1 + x_2 - x_3 + x_4 - x_5) = x_0^1$$

see appendix A1 for the above calculation.

$$X_2(z) = (x_0 + x_1 z^1 + x_2 z^2 + x_3 z^3 + x_4 z^4 + x_5 z^5) \bmod (z-1)$$
$$X_2(z) = (x_0 + x_1 + x_2 + x_3 + x_4 + x_5) = x_0^2$$

$$X_3(z) = (x_0 + x_1 z^1 + x_2 z^2 + x_3 z^3 + x_4 z^4 + x_5 z^5) \bmod (z^2+z+1)$$
$$X_3(z) = (x_1 - x_2 + x_4 - x_5)\, z + (x_0 - x_2 + x_3 - x_5) = x_1^3 z + x_0^3$$

see A2 for the calculation.

$$X_4(z) = (x_0 + x_1 z^1 + x_2 z^2 + x_3 z^3 + x_4 z^4 + x_5 z^5) \bmod (z^2-z+1)$$
$$X_4(z) = (x_1 + x_2 - x_4 - x_5)\, z + (x_0 - x_2 - x_3 + x_5) = x_1^4 z + x_0^4$$

see A3 for the calculation.

The coefficients of $X_i(z)$ and $W_i(z)$ are just linear combinations of x_i, W_i , because the modulo $Q_i(z)$ process folds back higher powers of z into lower powers of z [5].

The W polynomial is of the same form giving :

$$W_1(z) = (W_0 - W_1 + W_2 - W_3 + W_4 - W_5) = W_0^1$$
$$W_2(z) = (W_0 + W_1 + W_2 + W_3 + W_4 + W_5) = W_0^2$$
$$W_3(z) = (W_1 - W_2 + W_4 - W_5)\, z + (W_0 - W_2 + W_3 - W_5) = W_1^3\, z + W_0^3$$
$$W_4(z) = (W_1 + W_2 - W_4 - W_5)\, z + (W_0 - W_2 - W_3 + W_5) = W_1^4\, z + W_0^4$$

Now formulate $Y_i(z) \quad i = 1,2,3,4$

$$Y_1(z) = W_0^1\, x_0^1 \bmod(z+1) = (x_0 - x_1 + x_2 - x_3 + x_4 - x_5)\cdot (W_0 - W_1 + W_2 - W_3 + W_4 - W_5) = W_0^1\, x_0^1 = y_0^1$$

$$Y_2(z) = W_0^2\, x_0^2 \bmod(z-1) = (W_0 + W_1 + W_2 + W_3 + W_4 + W_5)\cdot (x_0 + x_1 + x_2 + x_3 + x_4 + x_5) = W_0^2\, x_0^2 = y_0^2$$

$$Y_3(z) = (W_1^3\, z^1 + W_0^3)\cdot(x_1^3\, z^1 + x_0^3) \bmod (z^2+z+1) =$$
$$(W_1^3\, x_1^3\, z^2 + (W_1^3\, x_0^3 + W_0^3\, x_1^3)\, z^1 + W_0^3\, x_0^3) \bmod(z^2+z+1) =$$
$$(W_0^3\, x_0^3 - W_1^3\, x_1^3) + (W_1^3\, x_0^3 + W_0^3\, x_1^3 - W_1^3\, x_1^3)\, z^1 = y_0^3 + y_1^3\, z$$

see A4 for the calculation.

$$Y_4(z) = (W_1^4\, z^1 + W_0^4)\,(x_1^4\, z^1 + x_1^4) \bmod (z^2+z+1) =$$
$$(W_0^4\, x_0^4 - W_1^4\, x_1^4) + (W_1^4\, x_0^4 + W_0^4\, x_1^4 - W_1^4\, x_1^4)\, z = y_0^4 + y_1^4\, z$$

According to Winograd's Theorem [3], the minimum number of multiplications required to compute an N point cyclic convolution is , $2N - k$, where k is the number of irreducible factors of z^6-1. Therefore,

$$2N - k = 2(6) - 4 = 8$$

Hence, expressions can be found for Y_1, Y_2, Y_3, Y_4 which use only 8 intermediate multiplications.

The next step is to define these 8 multiplications $m_1, m_2, m_3, m_4, m_5, m_6, m_7, m_8$.

Since $Y_1(z)$ and $Y_2(z)$ consist of only one multiplication each, let,

$$m_1 = W_0^1 x_0^1$$

$$m_2 = W_0^2 x_0^2$$

To get the multiplies associated with $Y_3(z)$, let,

$$m_3 = (W_0^3 - W_1^3)(x_1^3 - x_0^3) = W_0^3 x_1^3 - W_0^3 x_0^3 - W_1^3 x_1^3 + W_1^3 x_0^3$$

$$m_4 = W_0^3 x_0^3$$

$$m_5 = W_1^3 x_1^3$$

$$\Longrightarrow \quad y_1^3 = m_3 + m_4$$

$$y_0^3 = m_4 - m_5$$

For $Y_4(z)$ define:

$$m_6 = (W_0^4 + W_1^4)(x_0^4 + x_1^4) = W_0^4 x_0^4 + W_0^4 x_1^4 + W_1^4 x_0^4 + W_1^4 x_1^4$$

$$m_7 = W_0^4 x_0^4$$

$$m_8 = W_1^4 x_1^4$$

$$\Longrightarrow \quad y_1^4 = m_6 - m_7$$

$$y_0^4 = m_7 - m_8$$

Now that the 8 intermediate multiplies have been defined the system looks as follows:

$$Y_1(z) = m_1$$

$$Y_2(z) = m_2$$

$$Y_3(z) = (m_3 + m_4)\, z + (m_4 - m_5)$$

$$Y_4(z) = (m_6 - m_7)\, z + (m_7 - m_8)$$

To determine $Y(z)$ form,

$$Y(z) = \left[\sum_{i=1}^{4} Y_i(z)\, S_i(z) \right] \bmod (z^6-1)$$

First find $\quad S_i(z) \quad i = 1,2,3,4$

Using eq (1.5.3) and (1.5.4) gives:

$$S_i(z) = T_i(z)\, R_i(z) \qquad i = 1,2,3,4$$

where

$$T_i(z) = z^6 - 1 \,/\, Q_i(z)$$

$$R_i(z) = [\, T_i(z) \,]^{-1} \bmod (Q_i(z))$$

Therefore,

$$T_1(z) = (z^6-1)/(z+1) = (z-1)\,(z^2+z+1)\,(z^2-z+1) = z^5-z^4+z^3-z^2+z-1$$

$$T_2(z) = (z^6-1)/(z-1) = (z+1)\,(z^2+z+1)\,(z^2-z+1) = z^5+z^4+z^3+z^2+z+1$$

$$T_3(z) = (z^6-1)/(z^2+z+1) = (z+1)\,(z-1)\,(z^2-z+1) = z^4-z^3+z-1$$

$$T_4(z) = (z^6-1)/(z^2-z+1) = (z+1)\,(z-1)\,(z^2+z+1) = z^4+z^3-z-1$$

Since

$$S_i(z) \equiv 1 \bmod Q_i(z) \qquad i=1,2,3,4$$

$$S_i(z) \equiv 0 \bmod Q_j(z) \qquad i \neq j$$

the following holds,

$$S_i(z) = 1 \equiv T_i(z) \cdot R_i(z) \bmod Q_i(z)$$

which is equivalent to,

$$1 \equiv [(T_i(z) \bmod Q_i(z)) \cdot (R_i(z) \bmod Q_i(z)] \bmod Q_i(z)$$

$$T_1(z) \bmod (z+1) = z^5 - z^4 + z^3 - z^2 + z - 1 \bmod (z+1)$$

$T_1(z) \bmod (z+1) \equiv -6$ see A5 for the calculation.

$\Longrightarrow$ $R_1(z) = [\, T_1(z)\,]^{-1} \bmod Q_1(z) = -\frac{1}{6}$

Therefore,

$$S_1(z) = -\frac{1}{6}\,(\, z^5 - z^4 + z^3 - z^2 + z - 1)$$

Similarly,

$T_2(z) \bmod(z-1) = +\,6$ see A6 for the calculation.

$\Longrightarrow$ $R_2(z) = [\, T_2(z)\,]^{-1} \bmod Q_2(z) = +\frac{1}{6}$

Therefore,

$$S_2(z) = \frac{1}{6}\,(z^5 + z^4 + z^3 + z^2 + z + 1)$$

$T_3(z) \bmod(z^2 + z + 1) = 2\,z - 2$ see A7 for the calculation.

$$1 \equiv R_3(z)\, T_3(z) \bmod (z^2+z+1)$$
$$1 \equiv (L\, z + r)\,(2\, z - 2) \bmod (z^2+z+1)$$

(it's known that $R_3(z)$ is of the form $L\,z + r$ since the degree must be less than 2).

$$1 \equiv (2\, L\, z^2 + (-2\, L + 2\, r)\, z - 2\, r) \bmod (z^2+z+1)$$

see A8 for the calculation.

$$1 \equiv (2\, r - 4\, L)\, z - 2(r + L)$$

$\Longrightarrow$ $2\,r - 4\,L = 0$

$\underline{-2\,r - 2\,L = 1}$

$-\,6\,L = 1$ $\Longrightarrow$ $L = -\frac{1}{6}$

$-2r = -\frac{4}{6}$ $\Longrightarrow$ $r = -\frac{2}{6}$

$$S_3(z) = (-\tfrac{1}{6}\,z - \tfrac{2}{6}\,)\,(z^4 - z^3 + z - 1) = -\tfrac{1}{6}(z+2)\,(z^4 - z^3 + z + 1)$$

$$S_3(z) = -\frac{1}{6}\,(z^5 + z^4 - 2\,z^3 + z^2 + z - 2)$$

$T_4(z) \bmod (z^2-z+1) = -2z-2$ see A9 for the calculation.

$1 \equiv -(L z + 2 r)(2 z + 2) \bmod (z^2-z+1)$

$1 \equiv -((4 L + 2 r) z + (2 r - 2 L))$ see A10.

$$\Longrightarrow \quad -4 L - 2 r = 0$$

$$\underline{+2 L - 2 r = 1}$$

$$-6 L = -1 \Longrightarrow L = \frac{1}{6}$$

$$r = -\frac{4}{2} L = -\frac{2}{6}$$

$$S_4(z) = \left(\frac{1}{6} z - \frac{2}{6}\right)(z^4 + z^3 - z - 1)$$

$$S_4(z) = \frac{1}{6}(z^5 - z^4 - 2 z^3 - z^2 + z + 2)$$

Now form $Y(z) = \left[\sum_{i=1}^{4} Y_i(z) S_i(z)\right] \bmod (z^6-1)$

$$\begin{aligned} Y(z) = [\, & m_1 \left(-\frac{1}{6}\right)(z^5 - z^4 + z^3 - z^2 + z - 1) + \\ & m_2 \left(+\frac{1}{6}\right)(z^5 + z^4 + z^3 + z^2 + z - 1) + \\ & ((m_3 + m_4) z + (m_4 - m_5)) \left(-\frac{1}{6}\right)(z^5 + z^4 - 2z^3 + z^2 + z - 2) + \\ & ((m_6 - m_7) z + (m_7 - m_8)) \left(+\frac{1}{6}\right)(z^5 - z^4 - 2z^3 + z^2 + z + 2)] \\ & \bmod (z^6-1) \end{aligned}$$

Combine like powers of z using the property that powers of z^6 fold back to z^0.

$$
\begin{aligned}
Y(z) = &+\tfrac{1}{6}[-m_1+m_2-(m_4-m_5)-(m_3+m_4)+(m_7-m_8)-(m_6-m_7)\ ]\ z^5\\
&+\tfrac{1}{6}[+m_1+m_2-(m_4-m_5)+2(m_3+m_4)-(m_7-m_8)-2(m_6-m_7)\ ]\ z^4\\
&+\tfrac{1}{6}[-m_1+m_2+2(m_4-m_5)-(m_3+m_4)-2(m_7-m_8)-(m_6-m_7)\ ]\ z^3\\
&+\tfrac{1}{6}[+m_1+m_2-(m_4-m_5)-(m_3+m_4)-(m_7-m_8)+(m_6-m_7)\ ]\ z^2\\
&+\tfrac{1}{6}[-m_1+m_2-(m_4-m_5)+2(m_3+m_4)+(m_7-m_8)+2(m_6-m_7)\ ]\ z^1\\
&+\tfrac{1}{6}[m_1+m_2-(m_3+m_4)+2(m_4-m_5)+(m_6-m_7)+2(m_7-m_8)\ ]\ z^0
\end{aligned}
$$

Reducing terms we get:

$$
\begin{aligned}
Y(z) = &+\tfrac{1}{6}[\ -m_1+m_2-m_3-2m_4+m_5-m_6+2m_7-m_8\ ]\ z^5\\
&+\tfrac{1}{6}[\ +m_1+m_2+2m_3+m_4+m_5-2m_6+m_7+m_8\ ]\ z^4\\
&+\tfrac{1}{6}[\ -m_1+m_2-m_3+m_4-2m_5-m_6-m_7+2m_8\ ]\ z^3\\
&+\tfrac{1}{6}[\ +m_1+m_2-m_3-2m_4+m_5+m_6-2m_7+m_8\ ]\ z^2\\
&+\tfrac{1}{6}[\ -m_1+m_2+2m_3+m_4+m_5+2m_6-m_7-m_8\ ]\ z^1\\
&+\tfrac{1}{6}[\ +m_1+m_2-m_3+m_4-2m_5+m_6+m_7-2m_8\ ]\ z^0
\end{aligned}
$$

$$
\begin{bmatrix} y_1\\ y_2\\ y_3\\ y_4\\ y_5\\ y_0 \end{bmatrix}
= \frac{1}{6}
\begin{bmatrix}
-1 & 1 & -1 & -2 & 1 & -1 & 2 & -1\\
1 & 1 & 2 & 1 & 1 & -2 & 1 & 1\\
-1 & 1 & -1 & 1 & -2 & -1 & -1 & 2\\
1 & 1 & -1 & -2 & 1 & 1 & -2 & 1\\
-1 & 1 & 2 & 1 & 1 & 2 & -1 & -1\\
1 & 1 & -1 & 1 & -2 & 1 & 1 & -2
\end{bmatrix}
\cdot
\begin{bmatrix} m_1\\ m_2\\ m_3\\ m_4\\ m_5\\ m_6\\ m_7\\ m_8 \end{bmatrix}
$$

[2]

Thus one has the Discrete Fourier Transform of 7 points after the 1st row and 1st column are accounted for.

$$\bar{y}_0 = \bar{x}_0 + \bar{x}_1 + \bar{x}_2 + \bar{x}_3 + \bar{x}_4 + \bar{x}_5 + \bar{x}_6$$
$$\bar{y}_1 = y_5 - \bar{x}_0$$
$$\bar{y}_2 = y_1 - \bar{x}_0$$
$$\bar{y}_3 = y_0 - \bar{x}_0$$
$$\bar{y}_4 = y_3 - \bar{x}_0$$
$$\bar{y}_5 = y_4 - \bar{x}_0$$
$$\bar{y}_6 = y_2 - \bar{x}_0$$

CHAPTER II

WDFT FOR A PRODUCT OF PRIME NUMBER OF POINTS

2.1 Theory Behind the Product of 2 Prime WDFT Algorithm

The concept used in Chapter I for finding the Fourier Transform of a prime number of points can be expanded to products of primes. The idea is to convert a 1-dimensional length $N = p_1 \cdot p_2$ transform into a 2-dimensional transform requiring computation of 2 shorter length p_1, p_2 transforms[2]. The equation for an N point transform is of the form:

(2.1.1)

$$\begin{bmatrix} A_0 \\ A_1 \\ A_2 \\ A_3 \\ \vdots \\ A_{p_1p_2-1} \end{bmatrix} = \begin{bmatrix} W^0 & W^0 & \cdots & W^0 \\ W^{1\cdot 0} & W^{1\cdot 1} & \cdots & W^{1(p_1p_2-1)} \\ W^{2\cdot 0} & W^{2\cdot 1} & \cdots & W^{2(p_1p_2-1)} \\ W^{3\cdot 0} & W^{3\cdot 1} & \cdots & W^{3(p_1p_2-1)} \\ \vdots & \vdots & & \vdots \\ W^{(p_1p_2-1)\cdot 0} & W^{(p_1p_2-1)1} & \cdots & W^{(p_1p_2-1)\cdot(p_1p_2-1)} \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \\ \vdots \\ a_{p_1p_2-1} \end{bmatrix}$$

From the Chinese Remainder Theorem it is known that each of the indicies from 1 to $p_1 p_2 - 1$ can be uniquely represented by an ordered pair of numbers (i_1, i_2) [8], where,

$$i_1 \equiv i \bmod (p_1)$$
$$i_2 \equiv i \bmod (p_2)$$

Using this we can reorder the input and output sequence in the following subscripted order:

$(0,0)$

$(0,1)$

$(0,2)$

$\vdots$

$(0,p_2-1)$

$(1,0)$

$(1,1)$

$(1,2)$

$\vdots$

$(1,p_2-1)$

$\vdots$

$(p_1-1,0)$

$(p_1-1,1)$

$(p_1-1,2)$

$\vdots$

(p_1-1,p_2-1)

Reordering (2.1.1) in the above fashion leads to the following system of equations:

(2.1.2)

$$
\begin{bmatrix} A_{(0,0)} \\ A_{(0,1)} \\ \vdots \\ A_{(0,p_2-1)} \\ \vdots \\ A_{(p_1-1,p_2-1)} \end{bmatrix}
=
\begin{bmatrix}
W^{(0,0)(0,0)} & \dots & W^{(0,0)(p_1-1,p_2-1)} \\
W^{(0,1)(0,0)} & \dots & W^{(0,1)(p_1-1,p_2-1)} \\
\vdots & & \vdots \\
W^{(0,p_2-1)(0,0)} & \dots & W^{(0,p_2-1)(p_1-1,p_2-1)} \\
\vdots & & \vdots \\
W^{(p_1-1,p_2-1)(0,0)} & \dots & W^{(p_1-1,p_2-1)^2}
\end{bmatrix}
\begin{bmatrix} a_{(0,0)} \\ a_{(0,1)} \\ \vdots \\ a_{(0,p_2-1)} \\ \vdots \\ a_{(p_1-1,p_2-1)} \end{bmatrix}
$$

(2.1.2) can be partitioned into $p_2 \times p_2$ submatricies starting in the upper left hand corner.

The $i+1^{st}$ column of marticies in the $j+1^{st}$ row looks as follows:

(2.1.3)

$$
M_{(j,i)} =
\begin{bmatrix}
W^{(j,0)(i,0)} & W^{(j,0)(i,1)} & W^{(j,0)(i,2)} & \dots & W^{(j,0)(i,p_2-1)} \\
W^{(j,1)(i,0)} & W^{(j,1)(i,1)} & W^{(j,1)(i,2)} & \dots & W^{(j,1)(i,p_2-1)} \\
W^{(j,2)(i,0)} & W^{(j,2)(i,1)} & W^{(j,2)(i,2)} & \dots & W^{(j,2)(i,p_2-1)} \\
W^{(j,3)(i,0)} & W^{(j,3)(i,1)} & W^{(j,3)(i,2)} & \dots & W^{(j,3)(i,p_2-1)} \\
\vdots & \vdots & \vdots & & \vdots \\
W^{(j,p_2-1)(i,0)} & W^{(j,p_2-1)(i,1)} & & \dots & W^{(j,p_2-1)(i,p_2-1)}
\end{bmatrix}
$$

This matrix resembles eq (1.1.2) (the definition of a DFT) if only the second coordinate of the exponents ordered pair is considered. Therefore the rows and columns of (2.1.3) can be rearranged so as to put it in the form of a cyclic convolution matrix. In fact the entire matrix in (2.1.2) can be arranged such that <u>all</u> the $p_2 \times p_2$ blocks are cyclic convolution matricies $M_{(j,i)}$ [8] :

$$\begin{bmatrix} M_{(0,0)} & M_{(0,1)} & M_{(0,2)} & \cdots & M_{(0,p_1-1)} \\ M_{(1,0)} & M_{(1,1)} & M_{(1,2)} & \cdots & M_{(1,p_1-1)} \\ M_{(2,0)} & M_{(2,1)} & M_{(2,2)} & \cdots & M_{(2,p_1-1)} \\ \vdots & \vdots & \vdots & & \vdots \\ M_{(p_1-1,0)} & M_{(p_1-1,1)} & M_{(p_1-1,2)} & \cdots & M_{(p_1-1,p_1-1)} \end{bmatrix}$$

2.2 Summary of Steps Needed to Compute the WDFT for N= the Product of any 2 Primes

$$N = p_1 p_2$$

i) Write the integers from 0 to N-1 as ordered pairs of numbers (i_1, i_2) where $i_j = i \bmod(p_j)$ $j = 1,2$

ii) Using reference [5] find generators g_1, g_2 for p_1 and p_2 respectively. Reorder the input sequence according to the generator sequences:

$$a_{(0,0)} \quad a_{(0, g_2^1 \bmod(p_2))} \quad \ldots\ldots\ldots a_{(0, g_2^{p_2-1} \bmod(p_2))}$$

$$a_{(g_1^1 \bmod(p_1), 0)}, \quad \ldots, \quad a_{(g_1^{p_1-1} \bmod(p_1), g_2^{p_2-1} \bmod(p_2))}$$

iii) Group the reordered input vector as p_2 length p_1 vectors in sequence. Consider each of these vectors as an input element to the p_1 order WDFT.

iv) Perform the operations required for the p_1 order WDFT except that each input element is a vector of p_2 elements.

v) For the p_2 order WDFT seperate the output of the p_1 point transform into vectors of length p_1 and perform the p_2 order WDFT.

vi) To put the output in its final order, write the numbers $0,1,2,\ldots,N-1$ as multiples of p_1 and p_2, and use these multiples as the reordering. This technique will be illustrated in sect 2.3 . Note: it does not matter if the reordering of i) or vii) is used first, but the opposite reordering from what was used on the input must be used for the output[9].

2.3 Example of a 15 Point WDFT $\qquad N = 3 \cdot 5$

Reorder the DFT using:

$i = (i \bmod 3, i \bmod 5) \qquad i = 0,1,2,\ldots 14$

0=(0,0)	5=(2,0)	10=(1,0)
1=(1,1)	6=(0,1)	11=(2,1)
2=(2,2)	7=(1,2)	12=(0,2)
3=(0,3)	8=(2,3)	13=(1,3)
4=(1,4)	9=(0,4)	14=(2,4)

The reordering becomes:

0 = (0,0)
6 = (0,1)
12= (0,2)
3 = (0,3)
9 = (0,4)
10= (1,0)
1 = (1,1)
7 = (1,2)
13= (1,3)
4 = (1,4)
5 = (2,0)
11= (2,1)
2 = (2,2)
8 = (2,3)
14= (2,4)

When indicating the W matrix throughout the rest of this chapter only the exponents will be denoted, the W will be omitted. The reordered DFT becomes:

$$\begin{bmatrix} A_{(0,0)} \\ A_{(0,1)} \\ A_{(0,2)} \\ A_{(0,3)} \\ A_{(0,4)} \\ A_{(1,0)} \\ A_{(1,1)} \\ A_{(1,2)} \\ A_{(1,3)} \\ A_{(1,4)} \\ A_{(2,0)} \\ A_{(2,1)} \\ A_{(2,2)} \\ A_{(2,3)} \\ A_{(2,4)} \end{bmatrix} = \begin{bmatrix} (0,0)(0,0) & (0,0)(0,1) & (0,0)(0,2) & \ldots & (0,0)(2,4) \\ (0,1)(0,0) & (0,1)(0,1) & (0,1)(0,2) & \ldots & (0,1)(2,4) \\ (0,2)(0,0) & (0,2)(0,1) & (0,2)(0,2) & \ldots & (0,2)(2,4) \\ (0,3)(0,0) & (0,3)(0,1) & (0,3)(0,2) & \ldots & (0,3)(2,4) \\ (0,4)(0,0) & (0,4)(0,1) & (0,4)(0,2) & \ldots & (0,4)(2,4) \\ (1,0)(0,0) & (1,0)(0,1) & (1,0)(0,2) & \ldots & (1,0)(2,4) \\ (1,1)(0,0) & (1,1)(0,1) & (1,1)(0,2) & \ldots & (1,1)(2,4) \\ (1,2)(0,0) & (1,2)(0,1) & (1,2)(0,2) & \ldots & (1,2)(2,4) \\ (1,3)(0,0) & (1,3)(0,1) & (1,3)(0,2) & \ldots & (1,3)(2,4) \\ (1,4)(0,0) & (1,4)(0,1) & (1,4)(0,2) & \ldots & (1,4)(2,4) \\ (2,0)(0,0) & (2,0)(0,1) & (2,0)(0,2) & \ldots & (2,0)(2,4) \\ (2,1)(0,0) & (2,1)(0,1) & (2,1)(0,2) & \ldots & (2,1)(2,4) \\ (2,2)(0,0) & (2,2)(0,1) & (2,2)(0,2) & \ldots & (2,2)(2,4) \\ (2,3)(0,0) & (2,3)(0,1) & (2,3)(0,2) & \ldots & (2,3)(2,4) \\ (2,4)(0,0) & (2,4)(0,1) & (2,4)(0,2) & \ldots & (2,4)(2,4) \end{bmatrix} \cdot \begin{bmatrix} a_{(0,0)} \\ a_{(0,1)} \\ a_{(0,2)} \\ a_{(0,3)} \\ a_{(0,4)} \\ a_{(1,0)} \\ a_{(1,1)} \\ a_{(1,2)} \\ a_{(1,3)} \\ a_{(1,4)} \\ a_{(2,0)} \\ a_{(2,1)} \\ a_{(2,2)} \\ a_{(2,3)} \\ a_{(2,4)} \end{bmatrix}$$

Multiply the exponents and reduce mod3 and mod5 using,

$$W^{(i,j)\ (k,L)} = W^{(ik \bmod 3,\, jL \bmod 5)}$$

This will give us the following DFT structure:

(2.3.2.a)

$$
\begin{bmatrix} A_{(0,0)} \\ A_{(0,1)} \\ A_{(0,2)} \\ A_{(0,3)} \\ A_{(0,4)} \\ A_{(1,0)} \\ A_{(1,1)} \\ A_{(1,2)} \\ A_{(1,3)} \\ A_{(1,4)} \\ A_{(2,0)} \\ A_{(2,1)} \\ A_{(2,2)} \\ A_{(2,3)} \\ A_{(2,4)} \end{bmatrix}
=
\begin{bmatrix}
(0,0) & (0,0) & (0,0) & (0,0) & (0,0) & (0,0) & (0,0) & (0,0) & (0,0) & (0,0) & (0,0) & (0,0) & \cdots & (0,0) \\
(0,0) & (0,1) & (0,2) & (0,3) & (0,4) & (0,0) & (0,1) & (0,2) & (0,3) & (0,4) & (0,0) & (0,1) & \cdots & (0,4) \\
(0,0) & (0,2) & (0,4) & (0,1) & (0,3) & (0,0) & (0,2) & (0,4) & (0,1) & (0,3) & (0,0) & (0,2) & \cdots & (0,3) \\
(0,0) & (0,3) & (0,1) & (0,4) & (0,2) & (0,0) & (0,3) & (0,1) & (0,4) & (0,2) & (0,0) & (0,3) & \cdots & (0,2) \\
(0,0) & (0,4) & (0,3) & (0,2) & (0,1) & (0,0) & (0,4) & (0,3) & (0,2) & (0,1) & (0,0) & (0,4) & \cdots & (0,1) \\
(0,0) & (0,0) & (0,0) & (0,0) & (0,0) & (1,0) & (1,0) & (1,0) & (1,0) & (1,0) & (2,0) & (2,0) & \cdots & (2,0) \\
(0,0) & (0,1) & (0,2) & (0,3) & (0,4) & (1,0) & (1,1) & (1,2) & (1,3) & (1,4) & (2,0) & (2,1) & \cdots & (2,4) \\
(0,0) & (0,2) & (0,4) & (0,1) & (0,3) & (1,0) & (1,2) & (1,4) & (1,1) & (1,3) & (2,0) & (2,2) & \cdots & (2,3) \\
(0,0) & (0,3) & (0,1) & (0,4) & (0,2) & (1,0) & (1,3) & (1,1) & (1,4) & (1,2) & (2,0) & (2,3) & \cdots & (2,2) \\
(0,0) & (0,4) & (0,3) & (0,2) & (0,1) & (1,0) & (1,4) & (1,3) & (1,2) & (1,1) & (2,0) & (2,4) & \cdots & (2,1) \\
(0,0) & (0,0) & (0,0) & (0,0) & (0,0) & (2,0) & (2,0) & (2,0) & (2,0) & (2,0) & (1,0) & (1,0) & \cdots & (1,0) \\
(0,0) & (0,1) & (0,2) & (0,3) & (0,4) & (2,0) & (2,1) & (2,2) & (2,3) & (2,4) & (1,0) & (1,1) & \cdots & (1,4) \\
(0,0) & (0,2) & (0,4) & (0,1) & (0,3) & (2,0) & (2,2) & (2,4) & (2,1) & (2,3) & (1,0) & (1,2) & \cdots & (1,3) \\
(0,0) & (0,3) & (0,1) & (0,4) & (0,2) & (2,0) & (2,3) & (2,1) & (2,4) & (2,2) & (1,0) & (1,3) & \cdots & (1,2) \\
(0,0) & (0,4) & (0,3) & (0,2) & (0,1) & (2,0) & (2,4) & (2,3) & (2,2) & (2,1) & (1,0) & (1,4) & \cdots & (1,1)
\end{bmatrix}
\cdot
\begin{bmatrix} a_{(0,0)} \\ a_{(0,1)} \\ a_{(0,2)} \\ a_{(0,3)} \\ a_{(0,4)} \\ a_{(1,0)} \\ a_{(1,1)} \\ a_{(1,2)} \\ a_{(1,3)} \\ a_{(1,4)} \\ a_{(2,0)} \\ a_{(2,1)} \\ a_{(2,2)} \\ a_{(2,3)} \\ a_{(2,4)} \end{bmatrix}
$$

(2.3.2b)

$$
\begin{bmatrix} A_0 \\ A_6 \\ A_{12} \\ A_3 \\ A_9 \\ A_{10} \\ A_1 \\ A_7 \\ A_{13} \\ A_4 \\ A_5 \\ A_{11} \\ A_2 \\ A_8 \\ A_{14} \end{bmatrix}
=
\left[\begin{array}{ccccc|ccccc|cccc}
(0,0) & (0,0) & (0,0) & (0,0) & (0,0) & (0,0) & (0,0) & (0,0) & (0,0) & (0,0) & (0,0) & (0,0) & \ldots & (0,0) \\
(0,0) & (0,1) & (0,2) & (0,3) & (0,4) & (0,0) & (0,1) & (0,2) & (0,3) & (0,4) & (0,0) & (0,1) & \ldots & (0,4) \\
(0,0) & (0,2) & (0,4) & (0,1) & (0,3) & (0,0) & (0,2) & (0,4) & (0,1) & (0,3) & (0,0) & (0,2) & \ldots & (0,3) \\
(0,0) & (0,3) & (0,1) & (0,4) & (0,2) & (0,0) & (0,3) & (0,1) & (0,4) & (0,2) & (0,0) & (0,3) & \ldots & (0,2) \\
(0,0) & (0,4) & (0,3) & (0,2) & (0,1) & (0,0) & (0,4) & (0,3) & (0,2) & (0,1) & (0,0) & (0,4) & \ldots & (0,1) \\
(0,0) & (0,0) & (0,0) & (0,0) & (0,0) & (1,0) & (1,0) & (1,0) & (1,0) & (1,0) & (2,0) & (2,0) & \ldots & (2,0) \\
(0,0) & (0,1) & (0,2) & (0,3) & (0,4) & (1,0) & (1,1) & (1,2) & (1,3) & (1,4) & (2,0) & (2,1) & \ldots & (2,4) \\
(0,0) & (0,2) & (0,4) & (0,1) & (0,3) & (1,0) & (1,2) & (1,4) & (1,1) & (1,3) & (2,0) & (2,2) & \ldots & (2,3) \\
(0,0) & (0,3) & (0,1) & (0,4) & (0,2) & (1,0) & (1,3) & (1,1) & (1,4) & (1,2) & (2,0) & (2,3) & \ldots & (2,2) \\
(0,0) & (0,4) & (0,3) & (0,2) & (0,1) & (1,0) & (1,4) & (1,3) & (1,2) & (1,1) & (2,0) & (2,4) & \ldots & (2,1) \\
(0,0) & (0,0) & (0,0) & (0,0) & (0,0) & (2,0) & (2,0) & (2,0) & (2,0) & (2,0) & (1,0) & (1,0) & \ldots & (1,0) \\
(0,0) & (0,1) & (0,2) & (0,3) & (0,4) & (2,0) & (2,1) & (2,2) & (2,3) & (2,4) & (1,0) & (1,1) & \ldots & (1,4) \\
(0,0) & (0,2) & (0,4) & (0,1) & (0,3) & (2,0) & (2,2) & (2,4) & (2,1) & (2,3) & (1,0) & (1,2) & \ldots & (1,3) \\
(0,0) & (0,3) & (0,1) & (0,4) & (0,2) & (2,0) & (2,3) & (2,1) & (2,4) & (2,2) & (1,0) & (1,3) & \ldots & (1,2) \\
(0,0) & (0,4) & (0,3) & (0,2) & (0,1) & (2,0) & (2,4) & (2,3) & (2,2) & (2,1) & (1,0) & (1,4) & \ldots & (1,1)
\end{array}\right]
\cdot
\begin{bmatrix} a_0 \\ a_6 \\ a_{12} \\ a_3 \\ a_9 \\ a_{10} \\ a_1 \\ a_7 \\ a_{13} \\ a_4 \\ a_5 \\ a_{11} \\ a_2 \\ a_8 \\ a_{14} \end{bmatrix}
$$

Notice that each of the $5 \cdot 5$ submatricies can be rearranged into cyclic sub-matricies. Thus the subsequences need to be regenerated.

From reference [5] one finds that 2 is a generator of 3 and 5.

$$2 \equiv 2^1 \bmod 5 \qquad 2 \equiv 2^1 \bmod 3$$
$$4 \equiv 2^2 \bmod 5 \qquad 1 \equiv 2^2 \bmod 3$$
$$3 \equiv 2^3 \bmod 5$$
$$1 \equiv 2^4 \bmod 5$$

Note: The algorithms in Appendix B regenerate the input sequence automatically, therefore this step is really unnecessary in this example, but the regeneration will be illustrated anyway.

Reorder the 5×5 sub-matricies using the above reordering.

$$\begin{bmatrix} A_{(0,0)} \\ A_{(0,2)} \\ A_{(0,4)} \\ A_{(0,3)} \\ A_{(0,1)} \\ A_{(2,0)} \\ A_{(2,2)} \\ A_{(2,4)} \\ A_{(2,3)} \\ A_{(2,1)} \\ A_{(1,0)} \\ A_{(1,2)} \\ A_{(1,4)} \\ A_{(1,3)} \\ A_{(1,1)} \end{bmatrix} = \begin{bmatrix} (0,0) & (0,0) & (0,0) & (0,0) & (0,0) & (0,0) & (0,0) & (0,0) & (0,0) & (0,0) & (0,0) & (0,0) & \ldots & (0,0) \\ (0,0) & (0,4) & (0,3) & (0,1) & (0,2) & (0,0) & (0,4) & (0,3) & (0,1) & (0,2) & (0,0) & (0,4) & \ldots & (0,2) \\ (0,0) & (0,3) & (0,1) & (0,2) & (0,4) & (0,0) & (0,3) & (0,1) & (0,2) & (0,4) & (0,0) & (0,3) & \ldots & (0,4) \\ (0,0) & (0,1) & (0,2) & (0,4) & (0,3) & (0,0) & (0,1) & (0,2) & (0,4) & (0,3) & (0,0) & (0,1) & \ldots & (0,3) \\ (0,0) & (0,2) & (0,4) & (0,3) & (0,1) & (0,0) & (0,2) & (0,4) & (0,3) & (0,1) & (0,0) & (0,2) & \ldots & (0,1) \\ (0,0) & (0,0) & (0,0) & (0,0) & (0,0) & (1,0) & (1,0) & (1,0) & (1,0) & (1,0) & (2,0) & (2,0) & \ldots & (2,0) \\ (0,0) & (0,4) & (0,3) & (0,1) & (0,2) & (1,0) & (1,4) & (1,3) & (1,1) & (1,2) & (2,0) & (2,4) & \ldots & (2,2) \\ (0,0) & (0,3) & (0,1) & (0,2) & (0,4) & (1,0) & (1,3) & (1,1) & (1,2) & (1,4) & (2,0) & (2,3) & \ldots & (2,4) \\ (0,0) & (0,1) & (0,2) & (0,4) & (0,3) & (1,0) & (1,1) & (1,2) & (1,4) & (1,3) & (2,0) & (2,1) & \ldots & (2,3) \\ (0,0) & (0,2) & (0,4) & (0,3) & (0,1) & (1,0) & (1,2) & (1,4) & (1,3) & (1,1) & (2,0) & (2,2) & \ldots & (2,1) \\ (0,0) & (0,0) & (0,0) & (0,0) & (0,0) & (2,0) & (2,0) & (2,0) & (2,0) & (2,0) & (1,0) & (1,0) & \ldots & (1,0) \\ (0,0) & (0,4) & (0,3) & (0,1) & (0,2) & (2,0) & (2,4) & (2,3) & (2,1) & (2,2) & (1,0) & (1,4) & \ldots & (1,2) \\ (0,0) & (0,3) & (0,1) & (0,2) & (0,4) & (2,0) & (2,3) & (2,1) & (2,2) & (2,4) & (1,0) & (1,3) & \ldots & (1,4) \\ (0,0) & (0,1) & (0,2) & (0,4) & (0,3) & (2,0) & (2,1) & (2,2) & (2,4) & (2,3) & (1,0) & (1,1) & \ldots & (1,3) \\ (0,0) & (0,2) & (0,4) & (0,3) & (0,1) & (2,0) & (2,2) & (2,4) & (2,3) & (2,1) & (1,0) & (1,2) & \ldots & (1,1) \end{bmatrix} \cdot \begin{bmatrix} a_{(0,0)} \\ a_{(0,2)} \\ a_{(0,4)} \\ a_{(0,3)} \\ a_{(0,1)} \\ a_{(2,0)} \\ a_{(2,2)} \\ a_{(2,4)} \\ a_{(2,3)} \\ a_{(2,1)} \\ a_{(1,0)} \\ a_{(1,2)} \\ a_{(1,4)} \\ a_{(1,3)} \\ a_{(1,1)} \end{bmatrix}$$

$$
\begin{bmatrix} A_0 \\ A_{12} \\ A_9 \\ A_3 \\ A_6 \\ A_5 \\ A_2 \\ A_{14} \\ A_8 \\ A_{11} \\ A_{10} \\ A_7 \\ A_4 \\ A_{13} \\ A_1 \end{bmatrix}
=
\begin{bmatrix}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 9 & 3 & 6 & 12 & 0 & 9 & 3 & 6 & 12 & 0 & 9 & 3 & 6 & 12 \\
0 & 3 & 6 & 12 & 9 & 0 & 3 & 6 & 12 & 9 & 0 & 3 & 6 & 12 & 9 \\
0 & 6 & 12 & 9 & 3 & 0 & 6 & 12 & 9 & 3 & 0 & 6 & 12 & 9 & 3 \\
0 & 12 & 9 & 3 & 6 & 0 & 12 & 9 & 3 & 6 & 0 & 12 & 9 & 3 & 6 \\
0 & 0 & 0 & 0 & 0 & 10 & 10 & 10 & 10 & 10 & 5 & 5 & 5 & 5 & 5 \\
0 & 9 & 3 & 6 & 12 & 10 & 4 & 13 & 1 & 7 & 5 & 14 & 8 & 11 & 2 \\
0 & 3 & 6 & 12 & 9 & 10 & 13 & 1 & 7 & 4 & 5 & 8 & 11 & 2 & 14 \\
0 & 6 & 12 & 9 & 3 & 10 & 1 & 7 & 4 & 13 & 5 & 11 & 2 & 14 & 8 \\
0 & 12 & 9 & 3 & 6 & 10 & 7 & 4 & 13 & 1 & 5 & 2 & 14 & 8 & 11 \\
0 & 0 & 0 & 0 & 0 & 5 & 5 & 5 & 5 & 5 & 10 & 10 & 10 & 10 & 10 \\
0 & 9 & 3 & 6 & 12 & 5 & 14 & 8 & 11 & 2 & 10 & 4 & 13 & 1 & 7 \\
0 & 3 & 6 & 12 & 9 & 5 & 8 & 11 & 2 & 14 & 10 & 13 & 7 & 1 & 4 \\
0 & 6 & 12 & 9 & 3 & 5 & 11 & 2 & 14 & 8 & 10 & 1 & 7 & 4 & 13 \\
0 & 12 & 9 & 3 & 6 & 5 & 2 & 14 & 8 & 11 & 10 & 7 & 4 & 13 & 1
\end{bmatrix}
\cdot
\begin{bmatrix} a_0 \\ a_{12} \\ a_9 \\ a_3 \\ a_6 \\ a_5 \\ a_2 \\ a_{14} \\ a_8 \\ a_{11} \\ a_{10} \\ a_7 \\ a_4 \\ a_{13} \\ a_1 \end{bmatrix}
$$

Notice that each of the 5×5 sub-matricies contain a cyclic convolution of a 4×4 matrix (i.e., remove the 1^{st} row and 1^{st} column).

To perform this 15 point transform the 3 and 5 point transform algorithms form Appendix B will be utilized. As stated earlier each of these algorithms have the regeneration of the input elements built into the algorithm, therefore all that is really needed is the reordering of the input according to $i = (i \bmod(p_1), i \bmod(p_2))$. Therefore, eqs (2.3.2.b) will be the input data. The entire 3 point transform will be done first 5 times, then this output will be put into the 5 point transform and this will be done 3 times.

Now, consider each 5×5 matrix as 1 element, and do the 3 point transform on the 3 input vectors which are:

$$\vec{\alpha}_0 = \begin{bmatrix} a_0 \\ a_6 \\ a_{12} \\ a_3 \\ a_9 \end{bmatrix} \qquad \vec{\alpha}_1 = \begin{bmatrix} a_{10} \\ a_1 \\ a_7 \\ a_{13} \\ a_4 \end{bmatrix} \qquad \vec{\alpha}_2 = \begin{bmatrix} a_5 \\ a_{11} \\ a_2 \\ a_8 \\ a_{14} \end{bmatrix}$$

Superscripts will be used to denote which point transform the given algorithm is from. Using Appendix B, find the first adds of the 3 point transform, and do them 5 times.

$$\vec{s}^{\,3}_1 = \vec{\alpha}_1 + \vec{\alpha}_2 = \begin{bmatrix} a_{10} \\ a_1 \\ a_7 \\ a_{13} \\ a_4 \end{bmatrix} + \begin{bmatrix} a_5 \\ a_{11} \\ a_2 \\ a_8 \\ a_{14} \end{bmatrix} = \begin{bmatrix} a_{10} + a_5 \\ a_1 + a_{11} \\ a_7 + a_2 \\ a_{13} + a_8 \\ a_4 + a_{14} \end{bmatrix} = \begin{bmatrix} s^3_{1,0} \\ s^3_{1,1} \\ s^3_{1,2} \\ s^3_{1,3} \\ s^3_{1,4} \end{bmatrix}$$

$$\vec{s}^{\,3}_2 = \vec{\alpha}_1 - \vec{\alpha}_2 = \begin{bmatrix} a_{10} - a_5 \\ a_1 - a_{11} \\ a_7 - a_2 \\ a_{13} - a_8 \\ a_4 - a_{14} \end{bmatrix} = \begin{bmatrix} s^3_{2,0} \\ s^3_{2,1} \\ s^3_{2,2} \\ s^3_{2,3} \\ s^3_{2,4} \end{bmatrix}$$

$$\vec{s}^{\,3}_3 = \vec{s}^{\,3}_1 + \vec{\alpha}_0 = \begin{bmatrix} a_0 + a_{10} + a_5 \\ a_6 + a_{11} + a_{11} \\ a_{12} + a_7 + a_2 \\ a_3 + a_{13} + a_8 \\ a_9 + a_4 + a_{14} \end{bmatrix} = \begin{bmatrix} s^3_{3,0} \\ s^3_{3,1} \\ s^3_{3,2} \\ s^3_{3,3} \\ s^3_{3,4} \end{bmatrix}$$

Notice that the first subscript element denotes the addition algorithm from Appendix B, the second subscript element denotes an internal ordering.

Next, do the multiplies for the 3 point transform 5 times using, $\vec{s}^{\,3}_1$, $\vec{s}^{\,3}_2$, and $\vec{s}^{\,3}_3$ as input. Appendix B indicates:

$$\vec{m}^{\,3}_0 = 1 \cdot \vec{s}^{\,3}_3 = \begin{bmatrix} m^3_{0,0} \\ m^3_{0,1} \\ m^3_{0,2} \\ m^3_{0,3} \\ m^3_{0,4} \\ m^3_{0,5} \end{bmatrix}$$

$$\vec{m}^{\,3}_1 = (\cos(v) - 1) \cdot \vec{s}^{\,3}_1 = \begin{bmatrix} m^3_{1,0} \\ m^3_{1,1} \\ m^3_{1,2} \\ m^3_{1,3} \\ m^3_{1,4} \\ m^3_{1,5} \end{bmatrix} \qquad v = \frac{2\pi}{3}$$

$$\vec{m}^{\,3}_2 \quad j \sin(v) \cdot \vec{s}^{\,3}_2 = \begin{bmatrix} m^3_{2,0} \\ m^3_{2,1} \\ m^3_{2,2} \\ m^3_{2,3} \\ m^3_{2,4} \\ m^3_{2,5} \end{bmatrix}$$

Next, apply the last 3 additions of the 3 point transform 5 times:

$$\overrightarrow{s_4^3} = \overrightarrow{m_0^3} + \overrightarrow{m_1^3}$$

$$\overrightarrow{s_5^3} = \overrightarrow{s_4^3} + \overrightarrow{m_2^3}$$

$$\overrightarrow{s_6^3} = \overrightarrow{s_4^3} + \overrightarrow{m_2^3} \quad .$$

Select the output of the 3 point transform:

$$\overrightarrow{A_0^3} = \overrightarrow{m_0^3} = \begin{bmatrix} a_{0,0}^3 \\ a_{0,1}^3 \\ a_{0,2}^3 \\ a_{0,3}^3 \\ a_{0,4}^3 \end{bmatrix}$$

$$\overrightarrow{A_1^3} = \overrightarrow{s_5^3} = \begin{bmatrix} a_{1,0}^3 \\ a_{1,1}^3 \\ a_{1,2}^3 \\ a_{1,3}^3 \\ a_{1,4}^3 \end{bmatrix}$$

$$\overrightarrow{A_2^3} = \overrightarrow{s_6^3} = \begin{bmatrix} a_{2,0}^3 \\ a_{2,1}^3 \\ a_{2,2}^3 \\ a_{2,3}^3 \\ a_{2,4}^3 \end{bmatrix}$$

.

Using Appendix B apply the preliminary adds of the 5 point transform 3 times:

$$\overrightarrow{s_1^5} = \begin{bmatrix} a_{0,1}^3 + a_{0,4}^3 \\ a_{1,1}^3 + a_{1,4}^3 \\ a_{2,1}^3 + a_{2,4}^3 \end{bmatrix} = \begin{bmatrix} s_{1,1}^5 \\ s_{1,2}^5 \\ s_{1,3}^5 \end{bmatrix}$$

$$\overrightarrow{s_2^5} = \begin{bmatrix} a_{0,1}^3 - a_{0,4}^3 \\ a_{1,1}^3 - a_{1,4}^3 \\ a_{2,1}^3 - a_{2,4}^3 \end{bmatrix} = \begin{bmatrix} s_{2,1}^5 \\ s_{2,2}^5 \\ s_{2,3}^5 \end{bmatrix}$$

$$\vec{s}^{\,5}_{3} = \begin{bmatrix} a^3_{0,3} + a^3_{0,2} \\ a^3_{1,3} + a^3_{1,2} \\ a^3_{2,3} + a^3_{2,2} \end{bmatrix} = \begin{bmatrix} s^5_{3,1} \\ s^5_{3,2} \\ s^5_{3,3} \end{bmatrix}$$

$$\vec{s}^{\,5}_{4} = \begin{bmatrix} a^3_{0,3} - a^3_{0,2} \\ a^3_{1,3} - a^3_{1,2} \\ a^3_{2,3} - a^3_{2,2} \end{bmatrix} = \begin{bmatrix} s^5_{4,1} \\ s^5_{4,2} \\ s^5_{4,3} \end{bmatrix}$$

$$\vec{s}^{\,5}_{5} = \vec{s}^{\,5}_{1} + \vec{s}^{\,5}_{3} = \begin{bmatrix} s^5_{5,1} \\ s^5_{5,2} \\ s^5_{5,3} \end{bmatrix}$$

$$\vec{s}^{\,5}_{6} = \vec{s}^{\,5}_{1} - \vec{s}^{\,5}_{3} = \begin{bmatrix} s^5_{6,1} \\ s^5_{6,2} \\ s^5_{6,3} \end{bmatrix}$$

$$\vec{s}^{\,5}_{7} = \vec{s}^{\,5}_{2} - \vec{s}^{\,5}_{4} = \begin{bmatrix} s^5_{7,1} \\ s^5_{7,2} \\ s^5_{7,3} \end{bmatrix}$$

$$\vec{s}^{\,5}_{8} = \vec{s}^{\,5}_{5} + \begin{bmatrix} s^3_{1,0} \\ s^3_{2,0} \\ s^3_{3,0} \end{bmatrix} = \begin{bmatrix} s^5_{8,1} \\ s^5_{8,2} \\ s^5_{8,3} \end{bmatrix}$$

Next compute the 6 multiplies of the 5 point transform, 3 times, i.e., find $m_{0,i}, m_{1,i}, m_{2,i}, m_{3,i}, m_{4,i}, m_{5,i}$ $i=1,2,3$ using $s^5_{1,i}, s^5_{2,i}, s^5_{3,i}, s^5_{4,i}, s^5_{5,i}, s^5_{6,i}, s^5_{7,i}, s^5_{8,i}$ $i=1,2,3$ as input. From Appendix B the 6 multiplies of the 5 point transform are found to be:

$$m^5_{0,i} = 1 \cdot s^5_{8,i} \qquad i = 1,2,3$$

$$m^5_{1,i} = \frac{(\cos u + \cos 2u - 1)}{2} \cdot s^5_{5,i} \qquad u = \frac{2\pi}{5}$$

$$m^5_{2,i} = \frac{(\cos u - \cos 2u)}{2} \cdot s^5_{6,i}$$

$$m^5_{3,i} = \frac{j(\sin u + \sin 2u)}{2} \cdot s^5_{2,i} \qquad j = \sqrt{-1}$$

$$m^5_{4,i} = j \sin(2u)\, s^5_{7,i}$$

$$m^5_{5,i} = j\,(\sin u - \sin 2u)\, s^5_{4,i}$$

Next compute the remaining adds of the 5 point transform, 3 times, using $m^5_{0,k}$, $m^5_{1,k}$, $m^5_{2,k}$, $m^5_{3,k}$, $m^5_{4,k}$, $m^5_{5,k}$ $k=1,2,3$

Using Appendix B apply the last 9 adds of the 5 point transform, and use these to get the final output of the 5 point transform.

$$\vec{s}^{\,5}_{9} = \begin{bmatrix} m^5_{0,0} \\ m^5_{0,1} \\ m^5_{0,2} \end{bmatrix} + \begin{bmatrix} m^5_{1,0} \\ m^5_{1,1} \\ m^5_{1,2} \end{bmatrix} = \begin{bmatrix} s^5_{9,0} \\ s^5_{9,1} \\ s^5_{9,2} \end{bmatrix}$$

$$\vec{s}^{\,5}_{10} = \vec{s}^{\,5}_{9} + \begin{bmatrix} m^5_{2,0} \\ m^5_{2,1} \\ m^5_{2,2} \end{bmatrix} = \begin{bmatrix} s^5_{10,0} \\ s^5_{10,1} \\ s^5_{10,2} \end{bmatrix}$$

$$\vec{s}^{\,5}_{11} = \vec{s}^{\,5}_{9} - \begin{bmatrix} m^5_{2,0} \\ m^5_{2,1} \\ m^5_{2,2} \end{bmatrix} = \begin{bmatrix} s^5_{11,0} \\ s^5_{11,1} \\ s^5_{11,2} \end{bmatrix}$$

$$\vec{s}^{\,5}_{12} = \begin{bmatrix} m^5_{3,0} \\ m^5_{3,1} \\ m^5_{3,2} \end{bmatrix} - \begin{bmatrix} m^5_{4,0} \\ m^5_{4,1} \\ m^5_{4,2} \end{bmatrix} = \begin{bmatrix} s^5_{12,0} \\ s^5_{12,1} \\ s^5_{12,2} \end{bmatrix}$$

$$\vec{s}^{\,5}_{13} = \begin{bmatrix} m^5_{4,0} \\ m^5_{4,1} \\ m^5_{4,2} \end{bmatrix} + \begin{bmatrix} m^5_{5,0} \\ m^5_{5,1} \\ m^5_{5,2} \end{bmatrix} = \begin{bmatrix} s^5_{13,0} \\ s^5_{13,1} \\ s^5_{13,2} \end{bmatrix}$$

$$\vec{s}^{\,5}_{14} = \vec{s}^{\,5}_{10} + \vec{s}^{\,5}_{12} = \begin{bmatrix} s^5_{14,0} \\ s^5_{14,1} \\ s^5_{14,2} \end{bmatrix}$$

$$\vec{s}^{\,5}_{15} = \vec{s}^{\,5}_{10} - \vec{s}^{\,5}_{12} = \begin{bmatrix} s^5_{15,0} \\ s^5_{15,1} \\ s^5_{15,2} \end{bmatrix}$$

$$\vec{s}^{\,5}_{16} = \vec{s}^{\,5}_{11} + s^5_{13} = \begin{bmatrix} s^5_{16,0} \\ s^5_{16,1} \\ s^5_{16,2} \end{bmatrix}$$

$$s^5_{17} = \vec{s}^{\,5}_{11} - \vec{s}^{\,5}_{13} = \begin{bmatrix} s^5_{17,0} \\ s^5_{17,1} \\ s^5_{17,2} \end{bmatrix}$$

Now, using $\vec{s}_9^{\,5}$, $\vec{s}_{10}^{\,5}, \vec{s}_{11}^{\,5}, \ldots, \vec{s}_{16}^{\,5}, \vec{s}_{17}^{\,5}$ denote the final output of the 5 point transform, 3 times, using Appendix B.

$$\vec{A}_0^{\,5} = \begin{bmatrix} m_{0,0}^5 \\ m_{0,1}^5 \\ m_{0,2}^5 \end{bmatrix} = \begin{bmatrix} A_{0,0}^{5'} \\ A_{0,1}^5 \\ A_{0,2}^5 \end{bmatrix}$$

$$\vec{A}_1^{\,5} = \vec{s}_{14}^{\,5} = \begin{bmatrix} s_{14,0}^5 \\ s_{14,1}^5 \\ s_{14,2}^5 \end{bmatrix} = \begin{bmatrix} A_{1,0}^5 \\ A_{1,1}^5 \\ A_{1,2}^5 \end{bmatrix}$$

$$\vec{A}_2^{\,5} = \vec{s}_{16}^{\,5} = \begin{bmatrix} s_{16,0}^5 \\ s_{16,1}^5 \\ s_{16,2}^5 \end{bmatrix} = \begin{bmatrix} A_{2,0}^5 \\ A_{2,1}^5 \\ A_{2,2}^5 \end{bmatrix}$$

$$\vec{A}_3^{\,5} = \vec{s}_{17}^{\,5} = \begin{bmatrix} s_{17,0}^5 \\ s_{17,1}^5 \\ s_{17,2}^5 \end{bmatrix} = \begin{bmatrix} A_{3,0}^5 \\ A_{3,1}^5 \\ A_{3,2}^5 \end{bmatrix}$$

$$\vec{A}_4^{\,5} = \vec{s}_{15}^{\,5} = \begin{bmatrix} s_{15,0}^5 \\ s_{15,1}^5 \\ s_{15,2}^5 \end{bmatrix} = \begin{bmatrix} A_{4,0}^5 \\ A_{4,1}^5 \\ A_{4,2}^5 \end{bmatrix}$$

To determine the ordering of the output, write the numbers from 0 to 14 as multiples of 3 and 5:

$$
\begin{aligned}
0 &= 0\cdot 3 + 0\cdot 5 = (0,0)\\
3 &= 1\cdot 3 + 0\cdot 5 = (1,0)\\
6 &= 2\cdot 3 + 0\cdot 5 = (2,0)\\
9 &= 3\cdot 3 + 0\cdot 5 = (3,0)\\
12 &= 4\cdot 3 + 0\cdot 5 = (4,0)\\[1em]
1 &= 2\cdot 3 + 2\cdot 5 = (2,2)\\
4 &= 3\cdot 3 + 2\cdot 5 = (3,2)\\
7 &= 4\cdot 3 + 2\cdot 5 = (4,2)\\
10 &= 0\cdot 3 + 2\cdot 5 = (0,2)\\
13 &= 1\cdot 3 + 2\cdot 5 = (1,2)\\[1em]
2 &= 4\cdot 3 + 1\cdot 5 = (4,1)\\
5 &= 0\cdot 3 + 1\cdot 5 = (0,1)\\
8 &= 1\cdot 3 + 1\cdot 5 = (1,1)\\
11 &= 2\cdot 3 + 1\cdot 5 = (2,1)\\
14 &= 3\cdot 3 + 1\cdot 5 = (3,1)
\end{aligned}
$$

Therefore, the output is in the following order:

$$A_0 = A^5_{0,0}$$
$$A_1 = A^5_{2,2}$$
$$A_2 = A^5_{4,1}$$
$$A_3 = A^5_{1,0}$$
$$A_4 = A^5_{3,2}$$
$$A_5 = A^5_{0,1}$$
$$A_6 = A^5_{2,0}$$
$$A_7 = A^5_{4,2}$$
$$A_8 = A^5_{1,1}$$
$$A_9 = A^5_{3,0}$$
$$A_{10} = A^5_{0,2}$$
$$A_{11} = A^5_{2,1}$$
$$A_{12} = A^5_{4,0}$$
$$A_{13} = A^5_{1,2}$$
$$A_{14} = A^5_{3,1}$$

CHAPTER III

WDFT FOR POWERS OF ODD PRIMES

3.1 Theory behind the WDFT for Powers of Odd Primes

The procedure for computing the WDFT for $N=p^r$,where p is an odd prime, becomes more involved due to the fact that there does not exist a generator for the entire group of numbers 1,2,... ,p^r-1, but if all the factors of p are removed a generator can be found.

The following properties are used:

Let $\phi(N)$, with $N=p^r$ be the number of positive integers not exceeding N that are coprime with N. Then we have [8],

$$\phi(N) = \phi(p^r) = p^r - p^{r-1} \text{ for all primes p.}$$

N has a primative root g, such that $\phi(N)$ is the smallest integer such that,

$$g^{\phi(N)} \equiv 1 \bmod (N) \quad \text{iff } N=2,4,p^r,2p^r \quad \text{where p is an}$$
odd prime [8].

A cyclic group of order $\phi(N)$ can be formed using the primative root of N. Therefore the cyclic convolution algorithms of the WDFT used for a prime number of points can be used on the $\phi(N) = p^r - p^{r-1}$ points. Hence the DFT matrix will be written with these elements in the upper left hand corner giving a $(p^r-p^{r-1}) \times (p^r-p^{r-1})$ augmented matrix.

At this point there are p^{r-1} elements which have not been accounted for. All these remaining elements are divisible by p [8]. Notice that there are p^{r-1} elements from 1 to p^r-1 that are divisible by p, therefore all the elements excluded from the cyclic subgroup are all those elements from 1 to p^{r-1} that have p as a factor and zero. The elements are:

(3.1.1) $0,p,2p,3p,4p,\ldots,(p^{r-1}-1)p$

The sequence in (3.1.1) can be divided by p giving the set of integers:

$$0,1,2,3,\ldots,p^{r-1}-1$$

From the preceding theory a cyclic group of order $p^{r-2}(p-1)$ can be found, then make the next row and column elements in the matrix this ordered set of elements, using powers of the primative root mod (p^{r-1}). The entries form a cyclic convolution matrix where the matrix starts over after $p^{r-1}-p^{r-2}$ rows. Thus the computations associated with these columns can be done with the computation of one $p^{r-2}(p-1)$ order cyclic convolution matrix.

Now there are p^{r-2} elements left, each of which is divisible by p^2. Dividing by p^2 gives,

$$0,1,2,3,4,\ldots,p^{r-2}-1$$

A $p^{r-3}(p-1)$ order cyclic subgroup of these elements can be found which will be ordered as the next columns and rows of the matrix, using powers of the primative root $\mathrm{mod}(p^{r-2})$. This is again a cyclic convolution matrix and thus the computations associated with these columns are obtained by computing a $p^{r-3}(p-1) \times p^{r-3}(p-1)$ cyclic convolution matrix.

This process continues until all elements are used up. In each case the additional columns added to the first $p^{r-1}(p-1) \times p^{r-1}(p-1)$ matrix have computations that are made using a cyclic convolution matrix.

Using these same arguments on the rows added to the $p^{r-1}(p-1) \times p^{r-1}(p-1)$ matrix these additional computations are also performed by a cyclic convolution matrix where several of the input elements are added together because the cyclic convolution matrix repeats itself across the columns of the matrix.

This procedure continues until all computations are complete [8].

3.2 Summary of Steps Needed to Compute the WDFT for Powers of Odd Primes [8]

$N = p^r$ p is an odd prime number

i) Using reference [5], find a primitive root, g, for p.

ii) Generate the subgroup of p^r by taking:

$$g^L \bmod(p^r) \qquad L = 1,2,3,4,\ldots,p^r - p^{r-1}$$

iii) Divide all the remaining integers from 1 to N-1 by p and order these elements using:

$$g^L \bmod(p^{r-1}) \qquad L = 1,2,3,4,\ldots,p^{r-1} - p^{r-2}$$

iv) Divide all remaining elements from 1 to N-1 by p^2 and order the elements using:

$$g^L \bmod(p^{r-2}) \qquad L = 1,2,3,4,\ldots,p^{r-2} - p^{r-3}$$

v) Continue the process of (iii) and (iv) until all numbers from 1 to N-1 are ordered.

vi) Write the reordered matrix equations for the WDFT.

vii) Using the steps for a prime number of points, do the computations required for the $p^{r-1}(p-1) \times p^{r-1}(p-1)$ matrix of computations generated in (ii) (i.e., in the upper left hand corner of (vi)).

viii) Using the steps from a prime number of points, do the computation required for the $p^{r-2}(p-1) \times p^{r-2}(p-1)$ matrix of computations generated in (iii).

ix) Continue the process of (vii),(viii) for the matricies of computations generated in (v).

x) using the steps from a prime number of points, do the computations required for the $p^{r-3}(p-1) \times p^{r-3}(p-1)$ matrix of computations generated in (iii).

xi) Continue the process in (x) for the matricies of computation generated in (iv) and (v).

xii) Combine additively the results of (vii) through (xi).

3.3 Example of the WDFT for a 9 Point Transform

$$N = 3^2$$

i) Using reference [5] shows that 5 is a primitive root of 3.

ii) generate:

$$g^L \bmod(3^2)$$

since $\phi(N) = \phi(p^r) = p^{r-1}(p-1) = 3\cdot(2) = 6$

$L = 1,2,3,4,5,6$

i.e., 6 of the elements from 1 to 9 are coprime with 9 and will be generated above and 3 of the elements will be left out, namely multiples of 3 and 0.

$$5 \equiv 5^1 \bmod (9)$$
$$7 \equiv 5^2 \bmod (9)$$
$$8 \equiv 5^3 \bmod (9)$$
$$4 \equiv 5^4 \bmod (9)$$
$$2 \equiv 5^5 \bmod (9)$$
$$1 \equiv 5^6 \bmod (9)$$

iii) The remaining integers are (0,3,6) excluding 0 gives $3\cdot(1,2)$ order these elements using,

$$g^L \bmod (p^{r-1}) \qquad L = 1,2$$

$$2 \equiv 5^1 \bmod(3)$$
$$1 \equiv 5^2 \bmod (3)$$

$\Longrightarrow$ (6,3) is the reordering.

steps (iv) and (v) are unnecessary here.

vi) Write the reordered matrix equation for the WDFT (W^k will be represented by k).

(3.3.1)

$$\begin{bmatrix} A_0 \\ A_5 \\ A_7 \\ A_8 \\ A_4 \\ A_2 \\ A_1 \\ A_6 \\ A_3 \end{bmatrix} = \begin{bmatrix} 1&1&1&1&1&1&1&1&1 \\ 1&7&8&4&2&1&5&3&6 \\ 1&8&4&2&1&5&7&6&3 \\ 1&4&2&1&5&7&8&3&6 \\ 1&2&1&5&7&8&4&6&3 \\ 1&1&5&7&8&4&2&3&6 \\ 1&5&7&8&4&2&1&6&3 \\ 1&3&6&3&6&3&6&0&0 \\ 1&6&3&6&3&6&3&0&0 \end{bmatrix} \cdot \begin{bmatrix} a_0 \\ a_5 \\ a_7 \\ a_8 \\ a_4 \\ a_2 \\ a_1 \\ a_6 \\ a_3 \end{bmatrix}$$

vii) Notice that the upper left hand corner of (3.3.1) can be treated as a 7 point WDFT,

i.e., use $a_0, a_5, a_7, a_8, a_4, a_2, a_1$ as input to the 7 point WDFT algorithm.

This part of the operation takes 8 multiplies and 36 adds.

viii) Notice that the 2×2 equations in the upper right hand corner of the matrix in (3.3.1) can be identified with the non W^0 terms of the 3 point WDFT, which can be computed with 2 multiplies and 2 adds.

i.e.,

(3.3.2)
$$\begin{bmatrix} W^3 & W^6 \\ W^6 & W^3 \end{bmatrix} \begin{bmatrix} a_6 \\ a_3 \end{bmatrix} = \begin{bmatrix} W^3 a_6 + W^6 a_3 \\ W^6 a_6 + W^3 a_3 \end{bmatrix}$$

Since W^3, and W^6 are complex conjugates the following holds:

$W^3 = R + j\, I$ and $W^6 = R - j\, I$

and (3.3.2) becomes,

(3.3.3)
$$\begin{bmatrix} R+jI & R-jI \\ R-jI & R+jI \end{bmatrix} \begin{bmatrix} a_6 \\ a_3 \end{bmatrix} = \begin{bmatrix} R(a_6+a_3) + jI(a_6-a_3) \\ R(a_6+a_3) - jI(a_6-a_3) \end{bmatrix}$$

Similarly for the 2 × 2 matricies along the bottom of the matrix (less the last 2 columns) in (3.3.1) one has:

$$\begin{bmatrix} A_6-a_0 \\ A_3-a_0 \end{bmatrix} = \begin{bmatrix} 3 & 6 & 3 & 6 & 3 & 6 \\ 6 & 3 & 6 & 3 & 6 & 3 \end{bmatrix} \begin{bmatrix} a_5 \\ a_7 \\ a_8 \\ a_4 \\ a_2 \\ a_1 \end{bmatrix} = \begin{bmatrix} W^3 & W^6 \\ W^6 & W^3 \end{bmatrix} \begin{bmatrix} a_5+a_8+a_2 \\ a_7+a_4+a_1 \end{bmatrix}$$

Using (3.3.3) we get,

$$\begin{bmatrix} A_6 \\ A_3 \end{bmatrix} = \begin{bmatrix} R((a_5+a_8+a_2)+(a_7+a_4+a_1)) + jI((a_5+a_8+a_2)-(a_7+a_4+a_1)) \\ R((a_5+a_8+a_2)+(a_7+a_4+a_1)) - jI((a_5+a_8+a_2)-(a_7+a_4+a_1)) \end{bmatrix}$$

These operations combine for 2 multiplies and 0 adds, since the adds inside the parenthesis are computed as part of the 7×7 transform.

ix) This step is not needed for this example.

x) The 2×2 matrix in the lower right hand corner of (3.3.1) requires no multiplies and no adds since $a_3 + a_6$ is already computed in (viii).

xi) This step is not needed for this example.

xii) Finally to combine the parts we need 9 additional adds.

Therefore the total number of multiplies is,

$$8 + 2 + 2 = 12$$

The total number of adds is,

$$36 + 2 + 9 = 42$$

where the last 2 adds are those required to include a_0 in the expressions for A_3 and A_6.

CHAPTER IV

DISCUSSION OF COMPUTER SIMULATION AND CONCLUSION

Although Winograd's Theorem states the number of multiplies needed to compute a WDFT, only for 3,5,and 7 points have the multiply algorithms actually been found. The WDFT is in general not as efficient as the FFT for powers of 2, and it is advisable to add or leave out 1 input sample point and use one of the other WDFT algorithms.

The WDFT has its greatest application for hardware uses i.e., for cases in which the number of input samples is known before hand. Using the WDFT for a computer simulation requires a before hand knowledge of the number of input points, due to the various structures of the algorithms.

In Table 4.1 is a listing of the number of multiplies required to do the WDFT, and $n \cdot \log(n)$ for the FFT.

no. of input points	no. of multiplies	no. of multiplies by W^0	$n\log_2(n)$
2	0	2	2
3	2	1	
4	0	4	8
5	5	1	
7	8	1	
8	2	6	24
9	10	1	
16	10	8	64

Table 4.1 Number of Multiplies Required to Do the WDFT Contrasted with $n\log_2(n)$ [6]

Program Analysis

Appendix C illustrates a FORTRAN IV computer simulation of a 35 point Winograd Discrete Fourier Transform. The 35 point transform is implemented by using the 5 and the 7 point WDFT algorithms. The program reorders the input data into a 5 by 7 matrix and does the 5 point transform 7 times. The 7 point transform is then done 5 times using the output of the 5 point transform as input. The output of the 7 point transform is unordered using a different unordering than the input scheme.

The algorithms in Appendix B, if used individually, take the input data in its original order and regenerate it so as to give a cyclic convolution matrix, i.e. there is no need to regenerate the input data if the algorithms in Appendix B are used one at a time. When 2 or more of these algorithms are used for 1 transform, a reordering(not a re-generating) of the input and output data is needed. For the input reordering each number from 0 to 34 is expressed as a multiple of 5 and 7 as follows:

$$
\begin{aligned}
0 &= 0\cdot 7 + 0\cdot 5 = (0,0)\\
5 &= 0\cdot 7 + 1\cdot 5 = (0,1)\\
10 &= 0\cdot 7 + 2\cdot 5 = (0,2)\\
15 &= 0\cdot 7 + 3\cdot 5 = (0,3)\\
20 &= 0\cdot 7 + 4\cdot 5 = (0,4)\\
25 &= 0\cdot 7 + 5\cdot 5 = (0,5)\\
30 &= 0\cdot 7 + 6\cdot 5 = (0,6)
\end{aligned}
$$

$$
\begin{aligned}
1 &= 3\cdot 7 + 3\cdot 5 = (3,3)\\
6 &= 3\cdot 7 + 4\cdot 5 = (3,4)\\
11 &= 3\cdot 7 + 5\cdot 5 = (3,5)\\
16 &= 3\cdot 7 + 6\cdot 5 = (3,0)\\
21 &= 3\cdot 7 + 0\cdot 5 = (3,1)\\
26 &= 3\cdot 7 + 1\cdot 5 = (3,2)
\end{aligned}
$$

$$
\begin{aligned}
2 &= 1\cdot 7 + 6\cdot 5 = (1,6)\\
7 &= 1\cdot 7 + 0\cdot 5 = (1,0)\\
12 &= 1\cdot 7 + 1\cdot 5 = (1,1)\\
17 &= 1\cdot 7 + 2\cdot 5 = (1,2)\\
22 &= 1\cdot 7 + 3\cdot 5 = (1,3)\\
27 &= 1\cdot 7 + 4\cdot 5 = (1,4)\\
32 &= 1\cdot 7 + 5\cdot 5 = (1,5)
\end{aligned}
$$

$$
\begin{aligned}
3 &= 4\cdot 7 + 2\cdot 5 = (4,2)\\
8 &= 4\cdot 7 + 3\cdot 5 = (4,3)\\
13 &= 4\cdot 7 + 4\cdot 5 = (4,4)\\
18 &= 4\cdot 7 + 5\cdot 5 = (4,5)\\
23 &= 4\cdot 7 + 6\cdot 5 = (4,6)\\
28 &= 4\cdot 7 + 0\cdot 5 = (4,0)\\
33 &= 4\cdot 7 + 1\cdot 5 = (4,1)
\end{aligned}
$$

$$
\begin{aligned}
4 &= 2\cdot 7 + 5\cdot 5 = (2,5)\\
9 &= 2\cdot 7 + 6\cdot 5 = (2,6)\\
14 &= 2\cdot 7 + 0\cdot 5 = (2,0)\\
19 &= 2\cdot 7 + 1\cdot 5 = (2,1)\\
24 &= 2\cdot 7 + 2\cdot 5 = (2,2)\\
29 &= 2\cdot 7 + 3\cdot 5 = (2,3)\\
34 &= 2\cdot 7 + 4\cdot 5 = (2,4)
\end{aligned}
$$

Which gives the following input order:

0 = (0,0)
1 = (3,3)
2 = (1,6)
3 = (4,2)
4 = (2,5)

5 = (0,1)
6 = (3,4)
7 = (1,0)
8 = (4,3)
9 = (2,6)

10 = (0,2)
11 = (3,5)
12 = (1,1)
13 = (4,4)
14 = (2,0)

15 = (0,3)
16 = (3,6)
17 = (1,2)
18 = (4,5)
19 = (2,1)

20 = (0,4)
21 = (3,0)
22 = (1,3)
23 = (4,6)
24 = (2,2)

25 = (0,5)
26 = (3,1)
27 = (1,4)
28 = (4,0)
29 = (2,3)

30 = (0,6)
31 = (3,2)
32 = (1,5)
33 = (4,1)
34 = (2,4)

The 1-dimensional transform of 35 points is converted to a 2-dimensional 5 by 7 WDFT using the above reordering. Since there is no zeroth array element in FORTRAN, 1 was added to each element.

For the output the following reordering was used:

$$i = (i \bmod(5), i \bmod(7))$$

Again since there does not exist a zeroth array element in FORTRAN, the output reordering really is:

$$i = (i \bmod(5) + 1, i \bmod(7) + 1)$$

It does not matter which reordering scheme is choosen for the output or the input, but the output must use the opposite ordering scheme from the input[9].

The execution time of the program is .19 seconds for 35 points, this gives .0054 seconds for 1 point. An FFT program was run with an input of 32 points, the execution time is .05 seconds, which is .0016 seconds for 1 point. This indicates that the FFT is faster than the WDFT, which is consistent with a paper done by Morris [10], in which he compares the FFT to the WDFT and finds the FFT to be more efficient. This is due to the fact that the WDFT requires much maneuvering of data from one array to the next, this loading and storing of data is what accounts for most of the execution time of the WDFT. Therefore, it does not necessarily follow that the Discrete Fourier Transform which uses the least number of multiplies will have the shortest execution time. The time required for loading and storing data, which is characteristic of the particular machine being used, versus the time saved by doing fewer multiplies is the crucial factor. For this program an IBM 3330 was used, which is about twice as fast as the IBM 370. If there does exist a machine which can do the loading and storage of data faster than the IBM 3330, then the WDFT may prove to be faster than the FFT, but at this point the FFT is still the better method.

APPENDIX A

Various Calculations Used in Chapter 1

A1

$$
\begin{array}{r|l}
 & x_5z^4 + (x_4-x_5)z^3 + (x_3-x_4+x_5)z^2 + (x_2-x_3+x_4-x_5)z + (x_1-x_2+x_3-x_4+x_5) \\
\hline
z+1 & x_5z^5 + x_4z^4 + x_3z^3 + x_2z^2 + x_1z^1 + x_0 \\
 & +x_5z^5 + x_5z^4 \\
\hline
 & (x_4-x_5)z^4 + x_3z^3 + x_2z^2 + x_1z^1 + x_0 \\
 & +(x_4-x_5)z^4 + (x_4-x_5)z^3 \\
\hline
 & (x_3-x_4+x_5)z^3 + x_2z^2 + x_1z^1 + x_0 \\
 & +(x_3-x_4+x_5)z^3 + (x_3-x_4+x_5)z^2 \\
\hline
 & (x_2-x_3+x_4-x_5)z^2 + x_1z^1 + x_0 \\
 & +(x_2-x_3+x_4-x_5)z^2 + (x_2-x_3+x_4-x_5)z^1 \\
\hline
 & (x_1-x_2+x_3-x_4+x_5)z^1 + x_0 \\
 & +(x_1-x_2+x_3-x_4+x_5)z^1 + (x_1-x_2+x_3-x_4+x_5) \\
\hline
 & (x_0-x_1+x_2-x_3+x_4-x_5)
\end{array}
$$

A2

$$\qquad\qquad x_5z^3 + (x_4-x_5)z^2 + (x_3-x_4)z^1 + (x_2+x_5-x_3)$$

$$z^2+z+1 \;\big|\; x_5z^5 + x_4z^4 + x_3z^3 + x_2z^2 + x_1z^1 + x_0$$

$$+x_5z^5 + x_5z^4 + x_5z^3$$

$$(x_4-x_5)z^4 + (x_3-x_5)z^3 + x_2z^2 + x_1z^1 + x_0$$

$$+(x_4-x_5)z^4 + (x_4-x_5)z^3 + (x_4-x_5)z^2$$

$$(x_3-x_4)z^3 + (x_2-x_4+x_5)z^2 + x_1z^1 + x_0$$

$$+(x_3-x_4)z^3 + (x_3-x_4)z^2 + (x_3-x_4)z^1$$

$$(x_2+x_5-x_3)z^2 + (x_1-x_3+x_4)z^1 + x_0$$

$$+(x_2+x_5-x_3)z^2 + (x_2+x_5-x_3)z^1 + (x_2+x_5-x_3)$$

$$(x_1+x_4-x_2-x_5)z^1 + (x_0-x_2-x_5+x_3)$$

A3

$$
\begin{array}{r|l}
 & x_5z^3 + (x_4+x_5)z^2 + (x_3+x_4)z^1 + (x_2-x_5+x_3) \\
\hline
z^2-z+1 & x_5z^5 + x_4z^4 + x_3z^3 + x_2z^2 + x_1z^1 + x_0 \\
 & +x_5z^5 - x_5z^4 + x_5z^3 \\
\cline{2-2}
 & (x_4+x_5)z^4 + (x_3-x_5)z^3 + x_2z^2 + x_1z^1 + x_0 \\
 & +(x_4+x_5)z^4 - (x_4+x_5)z^3 + (x_4+x_5)z^2 \\
\cline{2-2}
 & (x_2-x_5+x_3)z^2 + (x_1-x_3-x_4)z^1 + x_0 \\
 & +(x_2-x_5+x_3)z^2 - (x_2-x_5+x_3)z^1 + (x_2-x_5+x_3) \\
\cline{2-2}
 & (x_1-x_4+x_2-x_5)z^1 + (x_0-x_2-x_3+x_5)
\end{array}
$$

A4

$$
\begin{array}{r|l}
 & W_1^3x_1^3 \\
\hline
z^2+z+1 & W_1^3x_1^3z^2 + (W_1^3x_0^3 + W_0^3x_1^3)z^1 + W_0^3x_0^3 \\
 & +W_1^3x_1^3z^2 + W_1^3x_1^3z^1 + W_1^3x_1^3 \\
\cline{2-2}
 & (W_1^3x_0^3 + W_0^3x_1^3 + W_1^3x_1^3)z^1 + (W_0^3x_0^3 - W_1^3x_1^3)
\end{array}
$$

A5

$$
\begin{array}{r|l}
 & z^4 - 2z^3 + 3z^2 - 4z^1 + 5 \\
\hline
z+1 & z^5 - z^4 + z^3 - z^2 + z - 1 \\
 & \underline{+z^5 + z^4} \\
 & -2z^4 + z^3 - z^2 + z - 1 \\
 & \underline{-2z^4 - 2z^3} \\
 & 3z^3 - z^2 + z - 1 \\
 & \underline{+3z^3 + 3z^2} \\
 & -4z^2 + z^1 - 1 \\
 & \underline{-4z^2 - 4z^1} \\
 & 5z - 1 \\
 & \underline{+5z + 5} \\
 & -6
\end{array}
$$

A6

$$z^4 + 2z^3 + 3z^2 + 4z^1 + 5$$

$$z-1 \;\overline{\big)\; z^5 + z^4 + z^3 + z^2 + z^1 + 1}$$

$$\underline{+z^5 - z^4}$$

$$2z^4 + z^3 + z^2 + z^1 + 1$$

$$\underline{+2z^4 - 2z^3}$$

$$3z^3 + z^2 + z^1 + 1$$

$$\underline{+3z^3 - 3z^2}$$

$$4z^2 + z^1 + 1$$

$$\underline{+4z^2 - 4z^1}$$

$$5z^1 + 1$$

$$\underline{+5z^1 - 5}$$

$$+6$$

A7

$$z^2 - 2z + 1$$

$$z^2+z+1 \;\overline{\big)\; z^4 - z^3 + z^1 - 1}$$

$$\underline{+z^4 + z^3 + z^2}$$

$$-2z^3 - z^2 + z^1 - 1$$

$$\underline{-2z^3 - 2z^2 - 2z}$$

$$z^2 + 3z^1 - 1$$

$$\underline{+z^2 + z^1 + 1}$$

$$2z^1 - 2$$

A8

$$\begin{array}{r|l} & 2L \\ \hline z^2+z^1+1 & 2Lz^2 + (-2L+2m)z^1 - 2m \\ & +2Lz^2 + 2Lz^1 \qquad + 2L \\ \cline{2-2} & (2m - 4L)z^1 - 2(m+L) \end{array}$$

A9

$$\begin{array}{r|l} & \\ \hline z^2-z^1+1 & z^4 + z^3 - z^1 - 1 \\ & +z^4 - z^3 + z^2 \\ \cline{2-2} & 2z^3 - z^2 - z^1 - 1 \\ & +2z^3 - 2z^2 + 2z^1 \\ \cline{2-2} & z^2 - 3z^1 - 1 \\ & +z^2 - z^1 + 1 \\ \cline{2-2} & -2z^1 - 2 \end{array}$$

A10

$$\begin{array}{r|l} & 2L \\ \hline z^2-z+1 & 2Lz^2 + (2L+2m)z^1 + 2m \\ & +2Lz^2 - 2Lz^1 \qquad + 2L \\ \cline{2-2} & (4L + 2m)z^1 + (2m - 2L) \end{array}$$

APPENDIX B

Winograd's Algorithms for the 2,3,4,5,7,8,9, and 16 Point Transforms

B1.

$$\begin{vmatrix} \lambda_0 \\ \lambda_1 \end{vmatrix} = \begin{vmatrix} w^0 & w^0 \\ w^0 & -w^0 \end{vmatrix} \begin{vmatrix} a_0 \\ a_1 \end{vmatrix} \qquad w = e^{\frac{2\pi i}{2}} = -1$$

Algorithm:

$s_1 = a_0 + a_1$ $\qquad$ $s_2 = a_0 - a_1$

$m_0 = 1 \cdot s_1$ $\qquad$ $m_1 = 1 \cdot s_2$

$\lambda_0 = m_0$ $\qquad$ $\lambda_1 = m_1$

B2.

$$\begin{pmatrix} \lambda_0 \\ \lambda_1 \\ \lambda_2 \end{pmatrix} = \begin{pmatrix} w^0 & w^0 & w^0 \\ w^0 & w^1 & w^2 \\ w^0 & w^2 & w^1 \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ a_2 \end{pmatrix} \qquad w = e^{\frac{2\pi i}{3}}$$

Algorithm:

$s_1 = a_1 + a_2$ $\qquad$ $s_2 = a_1 - a_2$ $\qquad$ $s_3 = s_1 + a_0$

$m_0 = 1 \cdot s_3$ $\qquad$ $m_1 = (\cos u - 1) \cdot s_1$ $\qquad$ $m_2 = i \sin u \cdot s_2$ $\qquad$ $u = \frac{2\pi}{3}$

$s_4 = m_0 + m_1$ $\qquad$ $s_5 = s_4 + m_2$ $\qquad$ $s_6 = s_4 - m_2$

$\lambda_0 = m_0$ $\qquad$ $\lambda_1 = s_5$ $\qquad$ $\lambda_2 = s_6$

B3.

$$\begin{pmatrix} \lambda_0 \\ \lambda_1 \\ \lambda_2 \\ \lambda_3 \end{pmatrix} = \begin{pmatrix} w^0 & w^0 & w^0 & w^0 \\ w^0 & w^1 & -w^0 & -w^1 \\ w^0 & -w^0 & w^0 & -w^0 \\ w^0 & -w^1 & -w^0 & w^1 \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{pmatrix} \qquad w = e^{\frac{2\pi i}{4}}$$

$s_1 = a_0 + a_2$ $\qquad$ $s_2 = a_0 - a_2$ $\qquad$ $s_3 = a_1 + a_3$ $\qquad$ $s_4 = a_1 - a_3$

$s_5 = s_1 + s_3$ $\qquad$ $s_6 = s_1 - s_3$

$m_1 = 1 \cdot s_5$ $\qquad$ $m_2 = 1 \cdot s_6$ $\qquad$ $m_3 = 1 \cdot s_2$ $\qquad$ $m_4 = i \sin u \cdot s_4$ $\qquad$ $u = \frac{2\pi}{4}$

$s_7 = m_3 + m_4$ $\qquad$ $s_8 = m_3 - m_4$

$\lambda_0 = m_1$ $\qquad$ $\lambda_1 = s_7$ $\qquad$ $\lambda_2 = m_2$ $\qquad$ $\lambda_3 = s_8$

B4.

$$\begin{pmatrix}\lambda_0\\ \lambda_1\\ \lambda_2\\ \lambda_3\\ \lambda_4\end{pmatrix} \begin{pmatrix} w^0 & w^0 & w^0 & w^0 & w^0\\ w^0 & w^1 & w^2 & w^3 & w^4\\ w^0 & w^2 & w^4 & w^1 & w^3\\ w^0 & w^3 & w^1 & w^4 & w^2\\ w^0 & w^4 & w^3 & w^2 & w^1\end{pmatrix} \begin{pmatrix}a_0\\ a_1\\ a_2\\ a_3\\ a_4\end{pmatrix} \qquad w = e^{\frac{2\pi i}{5}}$$

Algorithm:

$s_1 = a_1+a_4 \quad s_2 = a_1-a_4 \quad s_3 = a_3+a_2 \quad s_4 = a_3-a_2$

$s_5 = s_1+s_3 \quad s_6 = s_1-s_3 \quad s_7 = s_2+s_4 \quad s_8 = s_5+a_0$

$m_0 = 1\cdot s_8 \quad m_1 = \left(\frac{\cos u+\cos 2u}{2} - 1\right)\cdot s_5 \quad m_2 = \left(\frac{\cos u-\cos 2u}{2}\right)\cdot s_6 \quad u = \frac{2\pi}{5}$

$m_3 = i(\sin u+\sin 2u)\cdot s_2 \quad m_4 = i\ \sin 2u\cdot s_7 \quad m_5 = i(\sin u-\sin 2u)\cdot s_4$

$s_9 = m_0+m_1 \quad s_{10} = s_9+m_2 \quad s_{11} = s_9-m_2 \quad s_{12} = m_3-m_4$

$s_{13} = m_4+m_5 \quad s_{14} = s_{10}+s_{12} \quad s_{15} = s_{10}-s_{12} \quad s_{16} = s_{11}+s_{13}$

$s_{17} = s_{11}-s_{13}$

$\lambda_0 = m_0 \quad \lambda_1 = s_{14} \quad \lambda_2 = s_{16} \quad \lambda_3 = s_{17} \quad \lambda_4 = s_{15}$

B5.

$$\begin{pmatrix}\lambda_0\\ \lambda_1\\ \lambda_2\\ \lambda_3\\ \lambda_4\\ \lambda_5\\ \lambda_6\end{pmatrix} = \begin{pmatrix} w^0 & w^0 & w^0 & w^0 & w^0 & w^0 & w^0\\ w^0 & w^1 & w^2 & w^3 & w^4 & w^5 & w^6\\ w^0 & w^2 & w^4 & w^6 & w^1 & w^3 & w^5\\ w^0 & w^3 & w^6 & w^2 & w^5 & w^1 & w^4\\ w^0 & w^4 & w^1 & w^5 & w^2 & w^6 & w^3\\ w^0 & w^5 & w^3 & w^1 & w^6 & w^4 & w^2\\ w^0 & w^6 & w^5 & w^4 & w^3 & w^2 & w^1\end{pmatrix} \begin{pmatrix}a_0\\ a_1\\ a_2\\ a_3\\ a_4\\ a_5\\ a_6\end{pmatrix} \qquad w = e^{\frac{2\pi i}{7}}$$

Algorithm:

$s_1 = a_1+a_6 \quad s_2 = a_1-a_6 \quad s_3 = a_4+a_3 \quad s_4 = a_4-a_3$

$s_5 = a_2-a_5 \quad s_6 = a_2-a_5 \quad s_7 = s_1+s_3 \quad s_8 = s_7+s_5$

$s_9 = s_8+a_0 \quad s_{10} = s_1-s_3 \quad s_{11} = s_3-s_5 \quad s_{12} = s_5-s_1$

$s_{13} = s_2+s_4 \quad s_{14} = s_{13}+s_6 \quad s_{15} = s_2-s_4 \quad s_{16} = s_4-s_6$

$s_{17} = s_6-s_2$

$m_0 = 1\cdot s_9 \quad m_1 = \left(\frac{\cos u+\cos 2u+\cos 3u}{3} -1\right)\cdot s_8 \quad u = \frac{2\pi i}{7}$

$m_2 = \left(\frac{2\cos u-\cos 2u-\cos 3u}{3}\right)\cdot s_{10} \quad m_3 = \left(\frac{\cos u-2\cos 2u+\cos 3u}{3}\right)\cdot s_{11}$

$m_4 = \left(\frac{\cos u+\cos 2u-2\cos 3u}{3}\right)\cdot s_{12} \quad m_5 = i\left(\frac{\sin u+\sin 2u-\sin 3u}{3}\right)\cdot s_{14}$

$m_6 = i\left(\frac{2\sin u - \sin 2u + \sin 3u}{3}\right)\cdot s_{15}$ $m_7 = i\left(\frac{\sin u - 2\sin 2u - \sin 3u}{3}\right)\cdot s_{16}$

$m_8 = i\left(\frac{\sin u + \sin 2u + 2\sin 3u}{3}\right)\cdot s_{17}$

$s_{18} = m_0+m_1$ $s_{19} = s_{18}+m_2$ $s_{20} = s_{19}+m_3$ $s_{21} = s_{18}-m_2$

$s_{22} = s_{21}-m_4$ $s_{23} = s_{18}-m_3$ $s_{24} = s_{23}+m_4$ $s_{25} = m_5+m_6$

$s_{26} = s_{25}+m_7$ $s_{27} = m_5-m_6$ $s_{28} = s_{27}-m_8$ $s_{29} = m_5-m_7$

$s_{30} = s_{29}+m_8$ $s_{31} = s_{20}+s_{26}$ $s_{32} = s_{20}-s_{26}$ $s_{33} = s_{22}+s_{28}$

$s_{34} = s_{22}-s_{28}$ $s_{35} = s_{24}+s_{30}$ $s_{36} = s_{24}-s_{30}$

$\lambda_0 = m_0$ $\lambda_1 = s_{31}$ $\lambda_2 = s_{33}$ $\lambda_3 = s_{36}$

$\lambda_4 = s_{35}$ $\lambda_5 = s_{34}$ $\lambda_6 = s_{32}$

B6.

$$\begin{pmatrix}\lambda_0\\\lambda_1\\\lambda_2\\\lambda_3\\\lambda_4\\\lambda_5\\\lambda_6\\\lambda_7\end{pmatrix} = \begin{pmatrix} w^0 & w^0 & w^0 & w^0 & w^0 & w^0 & w^0 & w^0\\ w^0 & w^1 & w^2 & w^3 & w^4 & w^5 & w^6 & w^7\\ w^0 & w^2 & w^4 & w^6 & w^0 & w^2 & w^4 & w^6\\ w^0 & w^3 & w^6 & w^1 & w^4 & w^7 & w^2 & w^5\\ w^0 & w^4 & w^0 & w^4 & w^0 & w^4 & w^0 & w^4\\ w^0 & w^5 & w^2 & w^7 & w^4 & w^1 & w^6 & w^3\\ w^0 & w^6 & w^4 & w^2 & w^0 & w^6 & w^4 & w^2\\ w^0 & w^7 & w^6 & w^5 & w^4 & w^3 & w^2 & w^1 \end{pmatrix} \begin{pmatrix}a_0\\a_1\\a_2\\a_3\\a_4\\a_5\\a_6\\a_7\end{pmatrix} \qquad w = e^{\frac{2\pi i}{8}}$$

Algorithm.

$s_1 = a_0+a_4$ $s_2 = a_0-a_4$ $s_3 = a_2+a_6$ $s_4 = a_2-a_6$

$s_5 = a_1+a_5$ $s_6 = a_1-a_5$ $s_7 = a_3+a_7$ $s_8 = a_3-a_7$

$s_9 = s_1+s_3$ $s_{10} = s_1-s_3$ $s_{11} = s_5+s_7$ $s_{12} = s_5-s_7$

$s_{13} = s_9+s_{11}$ $s_{14} = s_9-s_{11}$ $s_{15} = s_6+s_8$ $s_{16} = s_6-s_8$

$m_1 = 1\cdot s_{13}$ $m_2 = 1\cdot s_{14}$ $m_3 = 1\cdot s_{10}$ $m_4 = i\sin 2u\cdot s_{12}$ $u = \frac{2\pi}{8}$

$m_5 = 1\cdot s_2$ $m_6 = i\cdot\sin 2u\cdot s_4$ $m_7 = i\sin u\cdot s_{15}$ $m_8 = \cos u\cdot s_{16}$

$s_{17} = m_3+m_4$ $s_{18} = m_3-m_4$ $s_{19} = m_5+m_8$ $s_{20} = m_5-m_8$

$s_{21} = m_6+m_7$ $s_{22} = m_6-m_7$ $s_{23} = s_{19}+s_{21}$ $s_{24} = s_{19}-s_{21}$

$s_{25} = s_{20}+s_{22}$ $s_{26} = s_{20}-s_{22}$

$A_0 = m_1$ $\quad A_1 = s_{23}$ $\quad A_2 = s_{17}$ $\quad A_3 = s_{26}$

$A_4 = m_2$ $\quad A_5 = s_{25}$ $\quad A_6 = s_{18}$ $\quad A_7 = s_{24}$

B7.

$$\begin{pmatrix} A_0 \\ A_1 \\ A_2 \\ A_3 \\ A_4 \\ A_5 \\ A_6 \\ A_7 \\ A_8 \end{pmatrix} = \begin{pmatrix} w^0 & w^0 & w^0 & w^0 & w^0 & w^0 & w^0 & w^0 & w^0 \\ w^0 & w^1 & w^2 & w^3 & w^4 & w^5 & w^6 & w^7 & w^8 \\ w^0 & w^2 & w^4 & w^6 & w^8 & w^1 & w^3 & w^5 & w^7 \\ w^0 & w^3 & w^6 & w^0 & w^3 & w^6 & w^0 & w^3 & w^6 \\ w^0 & w^4 & w^8 & w^3 & w^7 & w^2 & w^6 & w^1 & w^5 \\ w^0 & w^5 & w^1 & w^6 & w^2 & w^7 & w^3 & w^8 & w^4 \\ w^0 & w^6 & w^3 & w^0 & w^6 & w^3 & w^0 & w^6 & w^3 \\ w^0 & w^7 & w^5 & w^3 & w^1 & w^8 & w^6 & w^4 & w^2 \\ w^0 & w^8 & w^7 & w^6 & w^5 & w^4 & w^3 & w^2 & w^1 \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \\ a_6 \\ a_7 \\ a_8 \end{pmatrix}$$

Algorithm:

$s_1 = a_1+a_8$ $\quad s_2 = a_1-a_8$ $\quad s_3 = a_7+a_2$ $\quad s_4 = a_7-a_2$

$s_5 = a_3+a_6$ $\quad s_6 = a_3-a_6$ $\quad s_7 = a_4+a_5$ $\quad s_8 = a_4-a_5$

$s_9 = s_1+s_3$ $\quad s_{10} = s_9+s_7$ $\quad s_{11} = s_{10}+s_5$ $\quad s_{12} = s_{11}+a_0$

$s_{13} = s_2+s_4$ $\quad s_{14} = s_{13}+s_8$ $\quad s_{15} = s_1-s_3$ $\quad s_{16} = s_3-s_7$

$s_{17} = s_7-s_1$ $\quad s_{18} = s_2-s_4$ $\quad s_{19} = s_4-s_8$ $\quad s_{20} = s_8-s_2$

$m_0 = 1\cdot s_{12}$ $\quad m_1 = (-\frac{1}{2})\cdot s_{10}$ $\quad m_2 = i\ \sin 3u\cdot s_{14}$ $\quad u = \frac{2\pi}{9}$

$m_3 = (\cos 3u-1)\cdot s_5$ $\quad m_4 = i\sin 3u\cdot s_6$ $\quad m_5 = (\frac{2\cos u-\cos 2u-\cos 4u}{3})\cdot s_{15}$

$m_6 = (\frac{\cos u+\cos 2u-2\cos 4u}{3})\cdot s_{16}$ $\quad m_7 = (\frac{\cos u-2\cos 2u+\cos 4u}{3})\cdot s_{17}$

$m_8 = i(\frac{2\sin u+\sin 2u-\sin 4u}{3})\cdot s_{18}$ $\quad m_9 = i(\frac{\sin u-\sin 2u-2\sin 4u}{3})\cdot s_{19}$

$m_{10} = i(\frac{\sin u+2\sin 2u+\sin 4u}{3})\cdot s_{20}$

$s_{21} = m_1+m_1$ $\quad s_{22} = s_{20}+m_1$ $\quad s_{23} = m_0+s_{22}$ $\quad s_{24} = s_{23}\ m_2$

$s_{25} = s_{23}-m_2$ $\quad s_{26} = m_0+m_3$ $\quad s_{27} = s_{26}+s_{21}$ $\quad s_{28} = s_{27}+m_5$

$s_{29} = s_{28}+m_6$ $\quad s_{30} = s_{27}-m_6$ $\quad s_{31} = s_{30}+m_7$ $\quad s_{32} = s_{27}-m_5$

$s_{33} = s_{32}-m_7$ $\quad s_{34} = m_4+m_8$ $\quad s_{35} = s_{34}+m_9$ $\quad s_{36} = m_4-m_9$

$s_{37} = s_{36}+m_{10}$ $\quad s_{38} = m_4-m_8$ $\quad s_{39} = s_{38}-m_{10}$ $\quad s_{40} = s_{29}+s_{35}$

$s_{41} = s_{29}-s_{35}$ $\quad s_{42} = s_{31}+s_{37}$ $\quad s_{43} = s_{31}-s_{37}$ $\quad s_{44} = s_{33}+s_{39}$

$s_{45} = s_{33}-s_{39}$

$A_0 = m_0$ $A_1 = s_{40}$ $A_2 = s_{43}$ $A_3 = s_{24}$ $A_4 = s_{43}$

$A_5 = s_{45}$ $A_6 = s_{23}$ $A_7 = s_{42}$ $A_8 = s_{41}$

88. $A_k = \sum_{j=0}^{15} w^{kj} a_j \quad k=0,1,\ldots,15 \quad w = e^{\frac{2\pi i}{16}}$.

Algorithm.

$s_1 = a_0+a_8$ $s_2 = a_0-a_8$ $s_3 = a_4+a_{12}$ $s_4 = a_4-a_{12}$

$s_5 = a_2+a_{10}$ $s_6 = a_2-a_{10}$ $s_7 = a_6+a_{14}$ $s_8 = a_6-a_{14}$

$s_9 = a_1+a_9$ $s_{10} = a_1-a_9$ $s_{11} = a_5+a_{13}$ $s_{12} = a_5-a_{13}$

$s_{13} = a_3+a_{11}$ $s_{14} = a_3-a_{11}$ $s_{15} = a_7+a_{15}$ $s_{16} = a_7-a_{15}$

$s_{17} = s_1+s_3$ $s_{18} = s_1-s_3$ $s_{19} = s_5+s_7$ $s_{20} = s_5-s_7$

$s_{21} = s_9+s_{11}$ $s_{22} = s_9-s_{11}$ $s_{23} = s_{13}+s_{15}$ $s_{24} = s_{13}-s_{15}$

$s_{25} = s_{17}+s_{19}$ $s_{26} = s_{17}-s_{19}$ $s_{27} = s_{21}+s_{23}$ $s_{28} = s_{21}-s_{23}$

$s_{29} = s_{25}+s_{27}$ $s_{30} = s_{25}-s_{27}$ $s_{31} = s_{22}+s_{24}$ $s_{32} = s_{22}-s_{24}$

$s_{33} = s_6+s_8$ $s_{34} = s_6-s_8$ $s_{35} = s_{10}+s_{16}$ $s_{36} = s_{10}-s_{16}$

$s_{37} = s_{12}+s_{14}$ $s_{38} = s_{12}-s_{14}$ $s_{39} = s_{35}+s_{37}$ $s_{40} = s_{36}+s_{38}$

$m_1 = 1\cdot s_{29}$ $m_2 = 1\cdot s_{30}$ $m_3 = 1\cdot s_{26}$ $m_4 = i\ \sin 4u\cdot s_{28}$ $u = \frac{2\pi}{16}$

$m_5 = 1\cdot s_{18}$ $m_6 = i\ \sin 4u\cdot s_{20}$ $m_7 = i\ \sin 2u\cdot s_{31}$ $m_8 = \cos 2u\cdot s_{32}$

$m_9 = 1\cdot\ {}_2$ $m_{10} = i\ \sin 4u\cdot\ {}_4$ $m_{11} = i\ \sin 2u\cdot\ {}_{33}$ $m_{12} = \cos 2u\cdot s_{34}$

$m_{13} = i\ \sin 3u\cdot s_{39}$ $m_{14} = i(\sin u-\sin 3u)\cdot s_{35}$ $m_{15} = i(\sin u + \sin 3u)\cdot s_{37}$

$m_{16} = \cos 3u\cdot s_{40}$ $m_{17} = (\cos u + \cos 3u)\cdot s_{36}$ $m_{18} = (\cos 3u - \cos u)\cdot s_{38}$

$s_{41} = m_3+ m_4$ $s_{42} = m_3 - m_4$ $s_{43} = m_5 + m_7$ $s_{44} = m_5 - m_7$

$s_{45} = m_6 + m_8$ $s_{46} = m_6 - m_8$ $s_{47} = s_{43} + s_{45}$ $s_{48} = s_{43} - s_{45}$

$s_{49} = s_{44} + s_{46}$ $s_{50} = s_{44} - s_{46}$ $s_{51} = m_9 + m_{12}$ $s_{52} = m_9 - m_{12}$

$s_{53} = m_{10} + m_{11}$ $s_{54} = m_{10} - m_{11}$ $s_{55} = m_{13} + m_{14}$ $s_{56} = m_{13} - m_{15}$

$s_{57} = m_{17} - m_{16}$ $s_{58} = m_{18} - m_{16}$ $s_{59} = s_{51} + s_{55}$ $s_{60} = s_{51} - s_{55}$

$s_{61} = s_{52} + s_{56}$ $s_{62} = s_{52} - s_{56}$ $s_{63} = s_{53} + s_{57}$ $s_{64} = s_{53} - s_{57}$

$s_{65} = s_{54} + s_{58}$ $s_{66} = s_{54} - s_{58}$ $s_{67} = s_{59} + s_{63}$ $s_{68} = s_{59} - s_{63}$

$s_{69} = s_{60} + s_{64}$ $s_{70} = s_{60} - s_{64}$ $s_{71} = s_{61} + s_{65}$ $s_{72} = s_{61} - s_{65}$

$s_{73} = s_{62} + s_{66}$ $s_{74} = s_{62} - s_{66}$

$\lambda_0 = m_1$ $\lambda_1 = s_{67}$ $\lambda_2 = s_{47}$ $\lambda_3 = s_{72}$ $\lambda_4 = s_{41}$ $\lambda_5 = s_{71}$

$\lambda_6 = s_{48}$ $\lambda_7 = s_{68}$ $\lambda_8 = m_2$ $\lambda_9 = s_{69}$ $\lambda_{10} = s_{49}$ $\lambda_{11} = s_{74}$

$\lambda_{12} = s_{42}$ $\lambda_{13} = s_{73}$ $\lambda_{14} = s_{50}$ $\lambda_{15} = s_{70}$

APPENDIX C

Listing and Output of a Computer Simulation for the 35 Point WDFT

```
C       JANICE S. DANNA
C       SUBMITTED IN PARTIAL FULFILMENT FOR THE MASTER OF S
C       OPERATIONS RESEARCH.
C       APPENDIX C
C
C       THIS PROGRAM IS USED TO DETERMINE EXECUTION TIME OF
C       35 POINT WINOGRAD DISCRETE FOURIER TRANSFORM.
C       THIS PROGRAM WILL DO THE 5 PT TRANSFORM FIRST,
C       THEN THE 7 POINT TRANSFORM WILL BE DONE.
C       THESE RESULTS WILL BE CONTRASTED WITH A 32 POINT FA
C       FOURIER TRANSFORM.
C
C
C
C.....THE FOLLOWING SUBROUTINES ARE USED:
C       PREMA5 ; DOES THE PRE-MULTIPLY ADDS FOR THE 5 PT TR
C       PREMA7; DOES THE PRE-MULTIPLY ADDS FOR THE 7 POINT
C       AFTMA5; DOES THE AFTER-MULTIPLY ADDS FOR THE 5 POIN
C       AFTMA7; DOES THE AFTER-MULTIPLY ADDS FOR THE 7 POIN
C       MULT5; DOES THE MULTIPLIES OF THE 5 PT TRANSFORM
C       MULT7; DOES THE MULTIPLIES OF THE 7 POINT TRANSFORM
C
C
        COMMON /SCALAR/NAMA5,NAMA7,NM5,NM7,NPMA5,
       1  NPMA7,P1,P1,P2
        COMPLEX A(5,7),BR(35),DATA(35)
        COMPLEX M5(5,7),M5C(7)
        COMPLEX M7(6,5),M7C(5)
        COMPLEX CLT5(4,7),CLT5C(7),CLT7(6,5),CUT70(5)
        COMPLEX S5(6,7),S5A(17,7),S7(17,5),S7A(36,5)
        INTEGER P1,P2
C
C
        NPMA5 = 8
        NPMA7 = 17
        NAMA5 = 9
        NAMA7 = 19
        NAT5 = NPMA5 + NAMA5
        NAT7 = NPMA7 + NAMA7
        NM5 = 6
        NM7 = 9
        P1 = 5
        P2 = 7
        N = P1*P2
```

```
      PI = 3.141592654
      ARG1 = 2.*PI*3./35.
C     NUMBER INPUT POINTS FROM 1 TO P1*P2
      DO 1 I = 1,N
      DATA(I) = 2000.*COS(ARG1*FLOAT(I-1)) +
     1 CMPLX(0.,1.)*2000.*SIN(ARG1*FLOAT(I-1))
    1 CONTINUE
C
      CALL REORD(DATA,A)
C
C     COMPUTE THE PRE-MULTIPLY ADDS OF THE 5 POINT TRANSF
C            P2 (7) TIMES
C
      CALL PREMA5(A,S5)
C
C
C     DO THE MULTIPLIES OF THE 5 POINT TRANSFORM
C
      CALL MULT5(S5,M5C,M5)
C
C     DO THE AFTER MULTIPLY ADDS OF THE 5 POINT TRANSFORM
C
      CALL AFTMA5(M5C,M5,S5A)
C
C     CHOOSE THE OUTPUT OF THE 5 POINT TRANSFORM
      CALL OUTPU5(M5C,S5A,OUT5C,OUT5)
C
C    NEXT COMPUTE THE 17 PRE-MULTIPLY ADDS OF THE
C     7 POINT TRANSFORM
C
      CALL PREMA7(OUT5C,OUT5,S7)
C
C     DO 9 MULTIPLIES OF THE 7 POINT TRANSFORM.
C
      CALL MULT7(S7,M7C,M7)
C
C     DO THE AFTER MULTIPLY ADDS OF THE 7 POINT TRANSFORM
C
      CALL AFTMA7(M7C,M7,S7A)
C
C     CHOOSE THE FINAL OUTPUT OF THE P2 (7) POINT
C     TRANSFORM.
C
      CALL OUTPU7(S7A,M7C,OUT7C,OUT7)
```

```
      CALL CNORD(CLT7C,CL17,BR)
C
      CALL PDF(BR)
C
      END
```

```
      SUBROUTINE REORD(DATA,A)
      COMMON /SCALAR/NAMA5,NAMA7,NM5,NM7,NPMA5,
     1  NPMA7,PI,P1,P2
C-----------------------------------------------------------------
      COMPLEX DATA(35),A(5,7)
      DIMENSION I1(35),I2(35)
      INTEGER P1,P2
      N=P1*P2
      DATA I1  /1,4,2,5,3,
     1              1,4,2,5,3,
     1              1,4,2,5,3,
     1              1,4,2,5,3,
     1              1,4,2,5,3,
     1              1,4,2,5,3,
     1              1,4,2,5,3/
      DATA I2   /1,4,7,3,6,2,5,
     1               1,4,7,3,6,2,5,
     1               1,4,7,3,6,2,5,
     1               1,4,7,3,6,2,5,
     1               1,4,7,3,6,2,5/
      DO 1 I=1,N
      A(I1(I),I2(I)) = DATA(I)
    1 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE PREMA5(A,S5)
C________________________________________________
C      THIS SUBROUTINE COMPUTES THE 8 PREMULTIPLY ADDS OF
C      P1 (5) POINT TRANSFORM P2 (7) TIMES.
C      AC(J) WILL CONTAIN THE C TH VALUE FOR J=1,...P2
C      REORDER THE INPUT SUCH THAT THE FIRST VALUE
C       BEGINS WITH ZERO.
C I.E.  A(I,J) I=1,...P1=5;  J=1,...P2=7  TO
C AC(J)    A(I,J) I=1,...P1-1=4;J=1,...,P2=7
C
C
      COMMON /SCALAR/NAMA5,NAMA7,NM5,NM7,NPMA5,
     1 NPMA7,PI,P1,P2
      INTEGER P1,P2,P11
      COMPLEX A(P1,P2),AC(7),S5(NPMA5,P2)
C
C
C
C
C
      DO 1 J=1,P2
      AC(J) = A(1,J)
      DO 2 I = 2,P1
      A(I-1,J) = A(I,J)
    2 CONTINUE
      P11=P1 -1
    1 CONTINUE
C
C.....THE PRE MULTIPLY ADDS OF THE 5 PT TRANSFORM
C
      DO 3 J = 1,P2
      S5(1,J) = A(1,J) + A(4,J)
      S5(2,J) = A(1,J) - A(4,J)
      S5(3,J) = A(3,J) + A(2,J)
      S5(4,J) = A(3,J) - A(2,J)
      S5(5,J) = S5(1,J) + S5(3,J)
      S5(6,J) = S5(1,J) - S5(3,J)
      S5(7,J) = S5(2,J) + S5(4,J)
      S5(8,J) = S5(5,J) + AC(J)
    3 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE MULT5(S5,M5C,M5)
C-----------------------------------------------------------------
C        THIS SUBROUTINE COMPUTES THE 6 MULTIPLIES OF THE
C       5 POINT TRANSFORM
      COMMON /SCALAR/NAMA5,NAMA7,NM5,NM7,NPMA5,
     1  NPMA7,PI,P1,P2
      COMPLEX M5(5,7),M5C(7)
      COMPLEX S5(8,7)
      INTEGER P2
C
C
C
C
C..... MULTIPLIES OF THE 5 POINT TRANSFORM
      U= 2.*PI*1./5.
C
      DO 3 I=1,P2
      M5C(I) = S5(8,I)
      M5(1,I) = ((COS(U)+COS(2.*U))/2. -1.) * S5(5,I)
      M5(2,I) = (COS(U)-COS(2.*U))/2. * S5(6,I)
      M5(3,I) = CMPLX(0.,-1.)*(SIN(U)+SIN(2.*U)) * S5(2,I
      M5(4,I) = CMPLX(0.,-1.)*SIN(2.*U) *S5(7,I)
      M5(5,I) = CMPLX(0.,-1.)*(SIN(U)-SIN(2.*U)) * S5(4,I
    3 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE AFTMA5(M5C,M5,S5A)
C-----------------------------------------------------------------------
C     THIS SUBROUTINE COMPUTES THE AFTER MULTIPLY ADDS OF
C     POINT TRANSFORM.
C
      COMMON /SCALAR/NAMA5,NAMA7,NM5,NM7,NPMA5,
     1  NPMA7,PI,P1,P2
      INTEGER P2
      COMPLEX M5C(7),M5(5,7)
      COMPLEX S5A(17,P2)
C
C
C
C
C     DO THE AFTER MULTIPLIES ADDS OF TH 5 POINT TRANSFOR
C
      DO 3 J =1,P2
      S5A(9,J) = M5C(J) + M5(1,J)
      S5A(10,J) = S5A(9,J) + M5(2,J)
      S5A(11,J) = S5A(9,J) - M5(2,J)
      S5A(12,J) = M5(3,J) - M5(4,J)
      S5A(13,J) = M5(4,J) + M5(5,J)
      S5A(14,J) = S5A(10,J) + S5A(12,J)
      S5A(15,J) = S5A(10,J) - S5A(12,J)
      S5A(16,J) = S5A(11,J) + S5A(13,J)
      S5A(17,J) = S5A(11,J) - S5A(13,J)
    3 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE OUTPU5(M50,S5A,OUT50,OUT5)
C-----------------------------------------------------------------
C      THIS SUBROUTINE CHOOSES THE FINAL OUTPUT OF THE
C      P1 (5) POINT TRANSFORM P2 (7) TIMES.
C
C
      COMMON /SCALAR/NAMA5,NAMA7,NM5,NM7,NPMA5,
     1  NPMA7,PI,P1,P2
C
      INTEGER P1,P2
      COMPLEX OUT50(P2),OUT5(4,P2),S5A(17,7),M50(7)
C
      DO 3 J = 1,P2
      OUT50(J) = M50(J)
      OUT5(1,J) = S5A(14,J)
      OUT5(2,J) = S5A(16,J)
      OUT5(3,J) = S5A(17,J)
      OUT5(4,J) = S5A(15,J)
    3 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE PREMA7(CLT5C,CLT5,S7)
C___________________________________________________________
C     THIS SUBROUTINE COMPUTES THE NPMA7 (17) PRE-MULTIPL
C     ADDS OF THE P2 (7) POINT TRANSFORM
C
      COMMON /SCALAR/NAMA5,NAMA7,NM5,NM7,NPMA5,
     1  NPMA7,PI,P1,P2
C
      COMPLEX AC(5),A(6,5),CLT5(4,7),CLT50(7)
      COMPLEX S7(17,5)
      INTEGER P1,P2
C
C
C
      AC(1) = OLT5C(1)
      DO 1 J=2,5
      AC(J) = OLT5(J-1,1)
      DO 2 I = 1,6
      A(I,1) = OLT5C(I+1)
      A(I,J) = CLT5(J-1,I+1)
    2 CONTINUE
    1 CONTINUE
C     COMPUTE THE 17 PRE MULTIPLY ADDS OF THE 7 PT TRANSF
C    NPMA5 (8) TIMES
      DO 3 I = 1,P1
      S7(1,I) = A(1,I) + A(6,I)
      S7(2,I) = A(1,I) - A(6,I)
      S7(3,I) = A(4,I) + A(3,I)
      S7(4,I) = A(4,I) - A(3,I)
      S7(5,I) = A(2,I) + A(5,I)
      S7(6,I) = A(2,I) - A(5,I)
      S7(7,I) = S7(1,I) + S7(3,I)
      S7(8,I) = S7(7,I) + S7(5,I)
      S7(9,I) = S7(8,I) + AC(I)
      S7(10,I) = S7(1,I) - S7(3,I)
      S7(11,I) = S7(3,I) - S7(5,I)
      S7(12,I) = S7(5,I) - S7(1,I)
      S7(13,I) = S7(2,I) + S7(4,I)
      S7(14,I) = S7(13,I) + S7(6,I)
      S7(15,I) = S7(2,I) - S7(4,I)
      S7(16,I) = S7(4,I) -S7(6,I)
      S7(17,I) = S7(6,I) - S7(2,I)
    3 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE MULT7(S7,M7C,M7)
C-----------------------------------------------------------------
C     THIS SUBROUTINE DOES THE NM7 (9) MULTIPLIES OF THE P
C     TRANSFORM .
C
      COMMON /SCALAR/NAMA5,NAMA7,NM5,NM7,NPMA5,
     1  NPMA7,PI,P1,P2
      COMPLEX M7C(5),M7(8,5)
      COMPLEX S7(17,5)
      INTEGER P1
C
C
C
C
C
C
      U= 2.*PI*1./7.
      DO 3 I = 1,P1
      M7C(I) = S7(9,I)
      M7(1,I) = ((COS(U) + COS(2.*U) + COS(3.*U))/3. - 1.
      M7(2,I) =  (2.*COS(U) - COS(2.*U) - COS(3.*U))/3. *
      M7(3,I) = (COS(U) - 2.*COS(2.*U) + COS(3.*U))/3. *S
      M7(4,I) = (COS(U) + COS(2*U) - 2.*COS(3.*U))/3. * S
      M7(5,I) = CMPLX(0.,-1.)*(SIN(U) + SIN(2.*U) - SIN(3
     1  S7(14,I)
      M7(6,I) = CMPLX(0.,-1.)*(2.*SIN(U) - SIN(2.*U) + SI
     1  S7(15,I)
      M7(7,I) = CMPLX(0.,-1.)*(SIN(U)-2.*SIN(2.*U)-SIN(3.
     1  S7(16,I)
      M7(8,I) = CMPLX(0.,-1.)*(SIN(U)+SIN(2.*U)+2.*SIN(3.
     1  S7(17,I)
    3 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE AFTMA7(M7C,M7,S7A)
C-----------------------------------------------------------------
C     THIS SUBROUTINE COMPUTES THE NAMA7 (19) AFTER MULTI
C     ADDITIONS OF THE P1 (5) POINT TRANSFORM .
      COMMON /SCALAR/NAMA5,NAMA7,NM5,NM7,NPMA5,
     1  NPMA7,PI,P1,P2
C
      INTEGER P1
      COMPLEX M7C(5),M7(8,5)
      COMPLEX S7A(36,P1)
C
C
      DO 3 I=1,P1
      S7A(18,I) = M7C(I) + M7(1,I)
      S7A(19,I) = S7A(18,I) + M7(2,I)
      S7A(20,I) = S7A(19,I) + M7(3,I)
      S7A(21,I) = S7A(18,I) - M7(2,I)
      S7A(22,I) = S7A(21,I) - M7(4,I)
      S7A(23,I) = S7A(18,I) - M7(3,I)
      S7A(24,I) = S7A(23,I) + M7(4,I)
      S7A(25,I) = M7(5,I) +     M7(6,I)
      S7A(26,I) = S7A(25,I) + M7(7,I)
      S7A(27,I) = M7(5,I) -     M7(6,I)
      S7A(28,I) = S7A(27,I) - M7(3,I)
      S7A(29,I) = M7(5,I) -   M7(7,I)
      S7A(30,I) = S7A(29,I) + M7(8,I)
      S7A(31,I) = S7A(20,I) + S7A(26,I)
      S7A(32,I) = S7A(20,I) - S7A(26,I)
      S7A(33,I) = S7A(22,I) + S7A(28,I)
      S7A(34,I) = S7A(22,I) - S7A(28,I)
      S7A(35,I) = S7A(24,I) + S7A(30,I)
      S7A(36,I) = S7A(24,I) - S7A(30,I)
    3 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE OUTPU7(S7A,M7C,OUT70,OUT7)
C----------------------------------------------------------------
C     THIS SUBROUTINE SETS THE FINAL OUTPUT OF THE
C     P2 (7)  POINT TRANSFORM
C
C
      COMMON /SCALAR/NAMA5,NAMA7,NM5,NM7,NPMA5,
     1  NPMA7,PI,P1,P2
C
      INTEGER P1
      COMPLEX M7C(P1),OUT7(6,P1),OUT70(P1),S7A(36,P1)
C
      DO 3 J=1,P1
      OUT70(J) = M7C(J)
      OUT7(1,J) = S7A(31,J)
      OUT7(2,J) = S7A(33,J)
      OUT7(3,J) = S7A(36,J)
      OUT7(4,J) = S7A(35,J)
      OUT7(5,J) = S7A(34,J)
      OUT7(6,J) = S7A(32,J)
    3 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE UNORD(OUT7C,OUT7,BR)
C-------------------------------------------------
C     THIS SUBROUTINE UNORDERS THE FINAL OUTPUT DATA.
C
      COMMON /SCALAR/NAMA5,NAMA7,NM5,NM7,NPMA5,
     1  NPMA7,PI,P1,P2
      INTEGER P1,P2,P11
      COMPLEX A(5,7),BR(35),OUT7C(P1),OUT7(6,P1)
      DIMENSION IG1(35),IG2(35)
C
C
      N=P1*P2
      DO 2 I = 1,P1
      A(I,1)  = OUT7C(I)
      DO 1 J = 2,P2
      A(I,J) = OUT7(J-1,I)
    1 CONTINUE
    2 CONTINUE
      DO 10 I=1,N
      II=I-1
      IG1(I) = MOD(II,P1) +1
   10 CONTINUE
      DO 11 J=1,N
      JJ=J-1
      IG2(J) = MOD(JJ,P2) +1
   11 CONTINUE
      DO 30 I=1,N
      BR(I) = A(IG1(I),IG2(I))
   30 CONTINUE
   31 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE PDF(BR)
C-------------------------------------------------------------------
C     THIS SUBROUTINE COMPUTES THE POWER SPECTRAL DENSITY
      COMMON /SCALAR/NAMA5,NAMA7,NM5,NM7,NPMA5,
     1 NPMA7,PI,P1,P2
      COMPLEX BR(35)
      DIMENSION RMAG(35),REAN(35),RIEAN(35),FREQ(35)
      INTEGER P1,P2
C
C
      WRITE(6,100)
      N=P1*P2
      T=1./35.
      DO 1 K= 1,N
      REAN(K) = REAL(BR(K))
      RIEAN(K) = AIMAG(BR(K))
      RMAG(K) = SQRT(REAN(K)**2 + RIEAN(K)**2)
    1 CONTINUE
    2 CONTINUE
      TOT = 0.
      DO 10 I=1,N
      TOT = TOT + RMAG(I)
   10 CONTINUE
      IF(TOT.NE.0.) GO TO 70
      WRITE(6,120) TOT
      TOT = 1.
   70 DO 50 I=1,N
      FREQ(I) = FLOAT(I-1)/(FLOAT(N)*T)
      RMAG(I) = RMAG(I)/TOT
      WRITE(6,80) FREQ(I),RMAG(I)
   50 CONTINUE
   80 FORMAT(1H ,1X,F10.4,19X,F15.5)
  100 FORMAT(////////1H0,' FREQUENCY(HERTZ)',20X,
     1 ' POWER SPECTRAL DENSITY')
  120 FORMAT('    TOT=',E10.4)
      RETURN
      END
```

FREQUENCY(HERTZ)	POWER SPECTRAL DENSITY
0.0	0.00000
1.0000	0.00000
2.0000	0.00000
3.0000	0.99999
4.0000	0.00000
5.0000	0.00000
6.0000	0.00000
7.0000	0.00000
8.0000	0.00000
9.0000	0.00000
10.0000	0.00000
11.0000	0.00000
12.0000	0.00000
13.0000	0.00000
14.0000	0.00000
15.0000	0.00000
16.0000	0.00000
17.0000	0.00000
18.0000	0.00000
19.0000	0.00000
20.0000	0.00000
21.0000	0.00000
22.0000	0.00000
23.0000	0.00000
24.0000	0.00000
25.0000	0.00000
26.0000	0.00000
27.0000	0.00000
28.0000	0.00000
29.0000	0.00000
30.0000	0.00000
31.0000	0.00000
32.0000	0.00000
33.0000	0.00000
34.0000	0.00000

REFERENCES

[1]Schmuel Winograd, "A New Method for Computing the DFT," Proceedings of IEEE International Conference on ASSP (Hartford, Conn.: May 1977), p 366-368.

[2]Dean P. Kolba, and Thomas W. Parks, "A Prime Factor FFT Algorithm Using High-Speed Convolution," IEEE Transactions on ASSP 24 (August 1977): 281-294.

[3]Schmuel Winograd, "Some Bilinear Forms Whose Multiplicative Complexity Depends on the Field of Constants," research report no. RC5669, IBM Research Center, Yorktown Heights, New York, October 1975.

[4]Schmuel Winograd, "On Computing the Discrete Fourier Transform," National Academy of Sciences Proceedings 73 (April 1976): 1005-1006.

[5]Herbert Haupman, Emanuel Vegh, and Janet Fisher, Table of All Primative Roots for Primes Less Than 5000 (Washington D. C.: U.S. Government Printing Office, 1970).

[6]Schmuel Winograd, "On Computing the Discrete Fourier Transform," Mathematics of Computations 32 (January 1978): 175-199.

[7]Ramesh C. Agarwal, and James W. Cooley, "New Algorithms for Digital Convolution," IEEE Transactions on ASSP, 35 (October 1977): 392-409.

[8]Wynn Smith Notes from Radar Signal Processing Lectures, Martin Marietta Aerospace, Orlando, Florida, Spring 1979.

[9]Communications with Dr. Schmuel Winograd, IBM Research Center, Yorktown Heights, 19 September 1979.

[10]L. Robert Morris, "A Comparative Study of Time Efficient FFT and WFTA Programs for General Purpose Computers," <u>IEEE Transactions on ASSP</u>, 26 (April 1978): 141-150.