INFORMATION TO USERS

The most advanced technology has been used to photograph and reproduce this manuscript from the microfilm master. UMI films the original text directly from the copy submitted. Thus, some dissertation copies are in typewriter face, while others may be from a computer printer.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyrighted material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps. Each oversize page is available as one exposure on a standard 35 mm slide or as a 17″ × 23″ black and white photographic print for an additional charge.

Photographs included in the original manuscript have been reproduced xerographically in this copy. 35 mm slides or 6″ × 9″ black and white photographic prints are available for any photographs or illustrations appearing in this copy for an additional charge. Contact UMI directly to order.

Accessing the World's Information since 1938

300 North Zeeb Road, Ann Arbor, MI 48106-1346 USA

Order Number 8820878

Fast Fourier transform algorithms for special N's and the implementations on VAX

Lu, Chao, Ph.D.

City University of New York, 1988

Copyright ©1988 by Lu, Chao. All rights reserved.

U·M·I
300 N. Zeeb Rd.
Ann Arbor, MI 48106

PLEASE NOTE:

In all cases this material has been filmed in the best possible way from the available copy. Problems encountered with this document have been identified here with a check mark __✓__.

1. Glossy photographs or pages _____
2. Colored illustrations, paper or print _____
3. Photographs with dark background _____
4. Illustrations are poor copy _____
5. Pages with black marks, not original copy _____
6. Print shows through as there is text on both sides of page _____
7. Indistinct, broken or small print on several pages _____
8. Print exceeds margin requirements _____
9. Tightly bound copy with print lost in spine _____
10. Computer printout pages with indistinct print _____
11. Page(s) _____ lacking when material received, and not available from school or author.
12. Page(s) _____ seem to be missing in numbering only as text follows.
13. Two pages numbered _____. Text follows.
14. Curling and wrinkled pages ✓
15. Dissertation contains pages with print at a slant, filmed as received _____
16. Other _____

_____

_____

U·M·I

FAST FOURIER TRANSFORM ALGORITHMS FOR SPECIAL N's AND THE IMPLEMENTATIONS ON VAX

by

CHAO LU

A dissertation submitted to the Graduate Faculty in Engineering in partial fulfillment of the requirements for the degree of Doctor of Philosophy, The City University of New York.

1988

© 1988

CHAO LU

All Rights Reserved

This manuscript has been read and accepted for the Graduate Faculty in Engineering in satisfaction of the dissertation requirement for the degree of Doctor of Philosophy.

January 22, 1988
date

Richard Palmer
Chairman of Examining Committee

Jan 22, 1988
date

Jacques E. Beneviste
Executive Officer

Louis Auslander

Michael Conner

George Eichmann

Allen Gorin

Robert Johnson
Supervisory Committee

Abstract

FAST FOURIER TRANSFORM ALGORITHMS FOR SPECIAL N's AND THE IMPLEMENTATIONS ON VAX

by

CHAO LU

Advisor: Professor Richard Tolimieri

A new class of FFT algorithms based upon multiplicative structure of indexing set have been designed for transform sizes 4p, 4pq, p^2, p^2q, where p and q are odd primes. These algorithms combine the arithmetic efficiency of the Winograd's multiplicative algorithms with the simplicity of programming of the Cooley-Tukey algorithms, and offer significant advantage over the Winograd's algorithms in practical applications. Programs have been written for transform sizes N less than 128 on the Micro VAX II whose run-time is about 40% to 60% better and in some cases are even better than the radix-2 or radix-4 optimized Cooley-Tukey programs (DEC. LABSTAR DSP Package) of the next applicable size.

The fundamental algorithm is derived in the form of a factorization CA, where C is a block-diagonal matrix having skew-circulant blocks and tensor products of skew-circulant blocks, which can be computed by corresponding smaller FFT subroutines, and A is a pre-addition matrix.

For each case, a family of variants of the fundamental algorithm are also designed which present options as to whether additions or multiplications dominate arithmetic cost and which are further distinguished by data flow. These algorithmic parameters play an important role in deciding which algorithm is best suited for given computer architecture.

A very efficient Small DFT Library is given in section 11 for the purpose of small FFT subroutine calls. Many algorithms have been tried in VAX assembly code for each size N, the fastest one has been collected into the Library.

ACKNOWLEDGEMENT

With the deepest gratitude, I wish to thank Prof. R. Tolimieri, my mentor, for his invaluable support, guidance, innumberable insights, stimulation, and the opportunity to learn and work with him. I also wish to thank Prof. L. Auslander for his important guidance and expert insights of the research field, and Prof. R. W. Johnson for his expert guidance on computer implementation, and Mr. H. Y. Liu, my colleague, who spends a lot of time with me doing programming.

I also want to express my deepest gratitude to Prof. David H. Cheng, who offered me the scholarship four years ago, made it possible for me to study in The City University of New York towards the Ph.D degree.

TABLE OF CONTENTS

1. INTRODUCTION

The efficient computation of large size discrete Fourier transforms has become an important tool in advancing science and modern technology. In various scientific and engineering disciplines, such as weather forecasting, oil exploration, aircraft design, X-ray image processing in hospital and artificial intelligence etc. require real time signal processing, and require that Fourier transforms be computed as fast as possible to reflect information for analysis and control. There are two ways to speed up the computations, first one can build large high speed computers, second one can design faster algorithms. Looking back into history, these two sides have been stimulating each other to advance modern scientific computations.

The discrete Fourier transform (DFT) of a N-point sequence a is defined as

$$b(k) = \sum_{0 \le j < N} a(j) w^{jk}\,, \quad 0 \le k < N\,; \quad w = e^{2\pi i/N}\,, \tag{1}$$

where b(k) represents the kth-term of the DFT sequence b. Write (1) in the matrix form, we have

$$\begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ \vdots \\ b_{N-1} \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & w & w^2 & \cdots & w^{N-1} \\ 1 & w^2 & w^4 & \cdots & w^{2(N-1)} \\ \cdots & & & & \\ 1 & w^{N-1} & w^{3(N-1)} & \cdots & w^{(N-1)(N-1)} \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ \vdots \\ a_{N-1} \end{bmatrix}, \tag{2}$$

we denote the above N by N matrix by F(N), and call it the DFT matrix of order N.

The design of fast DFT algorithms can be dated historically back to the times of Gauss (1805) [1]. The collected work of some unpublished manuscripts by Gauss contained the essentials of FFT algorithms. Since computers were not yet developed at that time, Gauss' ideas did not attract much attention. In 1965, when Cooley and Tukey published their paper [2], known as the Fast Fourier Transform algorithm(FFT), the computing science started a revolutionary era[3]. FFT algorithm reduced the DFT computations from the order of N^2 to the order of Nlog(N) operations . Many good algorithms have been designed since then, but the FFT algorithm is still the most widely used, because of programming simplicity and well structuring.

In 1976 and 1978, Winograd, in his papers [4] [5], presented a potentially more efficient algorithm, known as Winograd Fourier transform algorithm (WFTA), which provided a much deeper understanding of computing DFT. In the WFTA, the computation of the DFT's is accomplished by convolutions. In this way Winograd achieved theoretically the optimum arithmetic count, and he exhibited a lower computational bound for DFT algorithms by using the underlying finite ring structure. One form of the WFTA is given by the factorization:

$$F(p) = C_p B_p A_p \tag{3}$$

where A_p and C_p are pre- and post-addition matrices with only 1, 0, -1's, and B_p is a diagonal matrix with only pure real or pure imaginary entries. The order of B_p represents the number of multiplications required. Winograd showed that it has an order of p.

Even before Cooley and Tukey annouced their FFT algorithm, Good and Thomas demonstrated in their papers [6] and [7] that if p and q are relatively prime, the Fourier transform can be written in the form:

$$F(pq) = P_2\big(F(q) \otimes F(p)\big)P_1 \tag{4}$$

where P_1 and P_2 are permutation matrices due to the Good-Thomas input and output data indexing. $\otimes$ denotes the tensor product . (4) is known as Good-Thomas algorithm(GTA).

Combining the GTA with small size WFTA, we can get efficient algorithms for computing composite size N=pq, where p and q are relatively prime:

$$F(pq) = P_2\big[(C_q \otimes C_p)(B_q \otimes B_p)(A_q \otimes A_p)\big]P_1 \tag{5}$$

(5) is known as the Large Winograd FT algorithm(LWFT) or Good-Thomas -Winograd algorithm(GTW).

Although WFTA is theoretically superior to FFT algorithm as far as arithmetic complexity is concerned, implementation shows that it has certain defects: firstly for each N, it needs a whole new programming effort; secondly for large N, numberical stability becomes a problem [8].

Comparing the Cooley-Tukey FFT algorithms with the Winograd algorithms, FFT algorithms are much easier for implementation, but thier timing efficiency is not as good as expected, and also they are only very good for transform sizes of two's power $N=2^m$, as m goes bigger, the interval between two nearest applicable points becomes larger.

Tolimieri, [9], [10], [11], developed a new series of algorithms from the ring structures of data indexing set which preserve the arithmetic simplicity of the Winograd algorithms and can also be programmed with loops and subroutine calls for number of sizes which have the same ring structure. For example in the pq case, p and q are distinct primes, we have the basic factorization:

$$F(pq) = P^{-1}CAP \tag{6}$$

where P is permutation matrix, A is pre-addition matrix and C is block-diagonal matrix with each submatrix on the diagonal skew- circulant.

In this thesis, we extend Tolimieri's Algorithms to the cases of 4p, 4pq, p^2 and p^2q, where p and q are distinct odd primes. The 4p case can be considered as a special case of p^2q for p =2,

since the non-zero divisor of 4 is only 2, and 2 corresponding to the unit circle of Z/4 is -1, this makes the 4p and 4pq cases have much better structure, so that we seperate the 4p case from p^2q case.

Using the number theoretical methods, we have produced in the cases N=4p , 4pq and p^2q orbital decompositions of indexing set, producing highly structured algorithms, with an accompanying simplicity of code which can be efficiently matched to variety of machine environments.

For each case, after the fundamental factorization is derived, a family of variants of the fundamental algorithm are designed which present options as to whether additions or multiplications dominate arithmetic cost and which are further distinguished by data flow. These algorithmic parameters play an important role in deciding which algorithm of the class is best suited for the given computer architecture. For example as in Variant 2 and 4, we use integer multiplying real number to trade off with real additions, for the reasons that some of the super computers such as Cray and Cyber 205, as in [12] says, multiplication is almost for free.

All the algorithms in this thesis are given by matrix factorizations of the finite Fourier transform matrix containing tensor product factors which make them fairly easy to be implemented.

Immediately after the algorithms for each case presented, an example is given for good understanding and complete picture of the algorithms.

From the algorithms of pq, 4p, 4pq, p^2, p^2q and also the paper [13], we can see that they all need corresponding smaller sizes of Fourier transform subroutines. So that the efficient small DFT library is a very important basis for the efficiency of large size Fourier transforms. In section 11, a mixed algorithms DFT library is given, many algorithms have been tried for each size of N's, the fastest one has been selected into the DFT library.

In section 12, implementations of the algorithms on Micro VAX II has been considered. In section 13, efficiency comparisons between algorithms and with available DSP package are given. They show that the achievement is obvious. These programs are not only fullfilling the sizes which Cooley-Tukey algorithms do not have, and also for some sizes are much more efficient than the nearest Cooley-Tukey. We will see this from the timing tables given in section 13. The features of these new class of FFT algorithms are summarized in section 14.

2. THE CASE N=4

In order to derive a new class of FFT algorithms for special sizes of N=4p and N=4pq, p and q distinct odd primes, let us first consider the simplest case of N=4.

Ordering the indexing set $Z/4$ by the permutation

$$\Pi = \{ 0, 2, 1, 3 \}, \tag{1}$$

the corresponding Fourier transform matrix F_Π is

$$F_\Pi = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & i & -i \\ 1 & -1 & -i & i \end{bmatrix}. \tag{2}$$

There are several methods which can be used to factorize F_Π, each leads to a slightly different factorization . We will discuss each in turn in some detail for they provide the basic ideas underlying those algorithms of N=4p and N=4pq.

Method 1. First we can write

$$F_\Pi = CA, \tag{3}$$

where

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & i \\ 0 & 0 & 1 & -i \end{bmatrix}, \tag{4}$$

$$A = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & -1 \end{bmatrix}. \tag{5}$$

Although C is block-diagonal, it does not have skew-circulant blocks. Factorization (3) corresponds to the 4-point Winograd algorithm as described in [14]. To see this, note that Π corresponds to the permutation matrix

$$P = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \tag{6}$$

and

$$F(4) = P^{-1} F_\Pi P = (P^{-1}C)(AP), \tag{7}$$

where by direct computation, we can see

$$AP = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & -1 \end{bmatrix}, \tag{8}$$

$$P^{-1}C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & i \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & -i \end{bmatrix} . \tag{9}$$

From (4) and (5), for complex input data, 16 real additions and no multiplications are needed to compute the action of F_Π, if we ignore the multiplications by -1 and i.

Method 2. A partial diagonalization method will be applied to (2). Direct computation shows that

$$(F(2)^{-1} \oplus F(2)^{-1})F_\Pi = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & -1 & 0 & 0 \\ 0 & 0 & i & -i \end{bmatrix} , \tag{10}$$

implying

$$F_\Pi = (F(2) \oplus F(2)) \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & -1 & 0 & 0 \\ 0 & 0 & i & -i \end{bmatrix} . \tag{11}$$

The arithmetic is the same as that in method 1., but the data flow is different.

3. THE CASE N=4p

A class of algorithms computing the N=4p point Fourier transforms will be derived from the ring-structure of the indexing set Z/4p. The ring structure determines a permutation of input and output data.

From the Chinese Remainder Theorem, we have a set

$$\{e_1, e_2\}, \qquad e_1, e_2 \in Z/4p, \tag{1}$$

which is called a system of idempotents for the ring Z/4p, since it satisfies the following conditions:

$$e_1 \equiv 1 \bmod 4, \quad e_1 \equiv 0 \bmod p, \tag{2}$$

$$e_2 \equiv 0 \bmod 4, \quad e_2 \equiv 1 \bmod p, \tag{3}$$

then

$$1 \equiv e_1 + e_2 \bmod 4p, \tag{4}$$

$$0 \equiv e_1 e_2 \bmod 4p, \tag{5}$$

$$e_j^2 \equiv e_j \bmod 4p, \quad j = 1, 2. \tag{6}$$

We begin our study of the ring Z/4p by describing the unit group U of Z/4p. Take a generator z of U(p), the U is the set of 2(p-1) elements

$$e_1 + z^j e_2, \quad 0 \le j < p-1, \tag{7}$$

and

$$3e_1 + z^k e_2, \quad 0 \le k < p-1. \tag{8}$$

Denote the set of p-1 elements given in (7) by U^+, and the set of p-1 elements given in (8) by U^-. Set

$$f_1 = 2e_1, \tag{9}$$

$$f_2 = 2e_1 + e_2, \tag{10}$$

and observe that

$$f_1^2 = 0, \quad f_2^2 = e_2, \tag{11}$$

$$e_1 f_1 = f_1, \quad e_2 f_2 = e_2, \tag{12}$$

$$e_2 f_1 = 0, \quad e_1 f_2 = f_1. \tag{13}$$

From the formulas, we see that

$$0U = \{0\}, \tag{14}$$

$$f_1 U = \{2e_1\}, \tag{15}$$

$$e_1U = \{e_1, 3e_1\}\,, \tag{16}$$

$$e_2U = \{z^j e_2 \ : 0 \le j < p-1\}\,, \tag{17}$$

$$f_2U = \{2e_1 + z^j e_2 \ : 0 \le j < p-1\}\,. \tag{18}$$

The set of U-orbits corresponding to

$$0,\ f_1,\ e_1,\ e_2,\ f_2,\ 1 \tag{19}$$

defines a partition of Z/4p. Order the partition by (19).

The unit group U is ordered by first running over $0 \le j < p-1$ and then over $0 \le k < p-1$ in (7) and (8). The remaining U-orbits are ordered as listed. Denote the corresponding permutation by Π. We will describe

$$F_\Pi = PF(4p)P^{-1}\,, \tag{20}$$

where P is the permutation matrix corresponding to Π.

If α and β are in Z/4p, denote by

$$C(\alpha, \beta)\,, \tag{21}$$

the submatrix of F_Π corresponding to $\alpha U \times \beta U$. Then, since $e_1^2 = e_1$,

$$C(e_1, e_1) = \begin{bmatrix} v^{e_1} & v^{3e_1} \\ v^{3e_1} & v^{e_1} \end{bmatrix}\,, \quad v = e^{2\pi i/4p}\,. \tag{22}$$

By (2), $e_1 = p$ or 3p, implying $v^{e_1} = i$ or -i, respectively. In any case

$$v^{3e_1} = -v^{e_1}\,. \tag{23}$$

Set $C_4 \equiv C(e_1, e_1)$, then

$$C_4 = \begin{bmatrix} v^{e_1} & -v^{e_1} \\ -v^{e_1} & v^{e_1} \end{bmatrix}\,. \tag{24}$$

In the same way, since $e_2^2 = e_2$,

$$C(e_2, e_2) = \left[(v^{e_2})^{z^{j+k}} \right]_{0 \le j,k < p-1}\,, \tag{25}$$

by (3),

$$u = v^{e_2}\,, \tag{26}$$

is a primitive p-th root of unity. Set

$$C_p \equiv C(e_2, e_2)\,, \tag{27}$$

which we recognized as a rotated Winograd core.

By (7) and (8),

$$C(1,1) = \begin{bmatrix} v^{e_1}C_p & -v^{e_1}C_p \\ -v^{e_1}C_p & v^{e_1}C_p \end{bmatrix} = C_4 \otimes C_p\,. \tag{28}$$

The off-diagonal blocks are computed in the same fashion. Consider, for example, $C(1, e_2)$. By (7), (8) and (17), we have, using

$$(e_1 + z^j e_2) z^k e_2 = z^{j+k} e_2 \,, \tag{29}$$

$$(3e_1 + z^j e_2) z^k e_2 = z^{j+k} e_2 \,, \tag{30}$$

that

$$C(1, e_2) = 1_2 \otimes C_p \,. \tag{34}$$

Continuing in this way, the matrix F_Π is given by the following table.

Table 1.

	0	f_1	e_1	e_2	f_2	1
0	1	1	1_2^t	$1_{p'}^t$	$1_{p'}^t$	$1_{2p'}^t$
f_1	1	1	-1_2^t	$1_{p'}^t$	$1_{p'}^t$	$-1_{2p'}^t$
e_1	1_2	-1_2	C_4	$E(2,p')$	$-E(2,p')$	$C_4 \otimes 1_{p'}^t$
e_2	$1_{p'}$	$1_{p'}$	$E(2,p')$	C_p	C_p	$1_2^t \otimes C_p$
f_2	$1_{p'}$	$1_{p'}$	$-E(2,p')$	C_p	C_p	$-1_2^t \otimes C_p$
1	$1_{2p'}$	$-1_{2p'}$	$C_4 \otimes 1_{p'}$	$1_2 \otimes C_p$	$-1_2 \otimes C_p$	$C_4 \otimes C_p$

where 1_n is a n-dimensional vector with all 1's, and E(n,m) is n by m matrix with all elements equal 1.

Set

$$F_\Pi(4) = \begin{bmatrix} 1 & 1 & 1_2^t \\ 1 & 1 & -1_2^t \\ 1_2 & -1_2 & C_4 \end{bmatrix} \,, \tag{35}$$

then we can rewrite F_Π as

$$F_\Pi = \begin{bmatrix} F_\Pi(4) & F_\Pi(4) \otimes 1_{p'}^t \\ F_\Pi(4) \otimes 1_{p'} & F_\Pi(4) \otimes C_p \end{bmatrix} \,. \tag{36}$$

Since $C_p 1_{p'} = -1_{p'}$, we have

$$F_\Pi = \left(F_\Pi(4) \oplus (F_\Pi(4) \otimes C_p)\right) \begin{bmatrix} I_4 & I_4 \otimes 1_{p'}^t \\ -I_4 \otimes 1_{p'} & I_{4p'} \end{bmatrix} \,, \tag{37}$$

where $F_\Pi(4)$ can be factorized as

$$F_\Pi(4) = C' A' \,, \tag{38}$$

where

$$C' = I_2 \oplus \begin{bmatrix} 1 & v^{e_1} \\ 1 & -v^{e_1} \end{bmatrix}, \tag{39}$$

$$A' = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & -1 \end{bmatrix}, \tag{40}$$

which are similar to the case of N=4 as in section 2. Substitute (38) to (36), direct computation will show that

$$F_\Pi = CA\,, \tag{41}$$

where

$$C = C' \oplus (C' \otimes C_p)\,, \tag{42}$$

$$A = \begin{bmatrix} A' & A' \otimes 1^t_{p'} \\ -A' \otimes 1_{p'} & A' \otimes I_{p'} \end{bmatrix}. \tag{43}$$

Arithmetically, A is equivalent to

$$A' \otimes A(p)\,, \tag{44}$$

where $A(p)$ is defined as

$$A(p) = \begin{pmatrix} 1 & 1^t_{p'} \\ -1_{p'} & I_{p'} \end{pmatrix}. \tag{45}$$

Direct computation will give the arithmetic count as:

Table 3-2. $F_\Pi = CA$

Factor	R.A	R.M
C	$4(p-1)(4p-5)+4$	$16(p-1)^2$
A	4(7p-4)	0
F_Π	$4(4p^2-2p+2)$	$16(p-1)^2$

Table 3-3. $F_\Pi = CA$

Factor	R.A	R.M
12	128	64
20	368	256
28	736	576
44	1856	1600

Variant 1.

From the factorization (41), we will derive several variants offering a variety of arithmetic and data flow. First set

$$F_0 = I_4 \oplus \left(I_4 \otimes F(p')\right) , \tag{46}$$

then

$$C = F_0 D F_0 , \tag{47}$$

where

$$D = C' \oplus (C' \otimes D_p) , \tag{48}$$

and D_p is the diagonal matrix

$$D_p = F^{-1}(p') C_p F^{-1}(p') . \tag{49}$$

Note that D is not diagonal. Placing (47) in (41), we have

$$F_\Pi = F_0 D F_0 A . \tag{50}$$

Table 3-4. $F_\Pi = F_0 D F_0 A$

Factor	R.A	R.M
F_0	$4 \times A_p$	$4 \times M_p$
D	4(3p-6)	16(p-2)
A	4(7p-4)	0
F_Π	$8A_p + 40(p-1)$	$8M_p + 16(p-2)$

where A_p and M_p are the number of real additions and the number of real multiplications required to compute F(p-1).

Variant 2.

We can rewrite (50) as

$$F_\Pi = F_0 D B F_0 , \tag{51}$$

where the matrix

$$B = F_0 A F_0^{-1} , \tag{52}$$

is given by

$$B = \begin{bmatrix} A' & A' \otimes \underline{e}(p)^t \\ -p' A' \otimes \underline{e}(p) & A' \otimes I_{p'} \end{bmatrix} , \tag{53}$$

the vector $\underline{e}(p)$ is the p'-tuple beginning with 1 and followed by 0's. The matrix B requires 12p+16 real additions and 8 multiplications by p'.

Table 3-5. $F_\Pi = F_0 D B F_0$

Factor	R.A	R.M
F_0	$4 \times A_p$	$4 \times M_p$
D	4(3p-6)	16(p-2)
B	4(3p+4)	{8}
F_Π	$8A_p + 8(3p-1)$	$8M_p + 16(p-2) + \{8\}$

Multiplications by an integer are placed in {}.

Variant 3.

The next method replaces the complex multiplications required to compute the action of

$$C = C' \oplus (C' \otimes C_p) \tag{54}$$

by real multiplications. The matrix C_p has the form

$$C_p = \begin{pmatrix} X_p & X_p^* \\ X_p^* & X_p \end{pmatrix} , \tag{55}$$

with the result that

$$C_p = \left(F(2) \otimes I_{\frac{p'}{2}}\right) Y_p \left(F(2) \otimes I_{\frac{p'}{2}}\right) , \tag{56}$$

where

$$Y_p = \frac{1}{2}\left((X_p + X_p^*) \oplus (X_p - X_p^*)\right) . \tag{57}$$

The multiplications required to compute the action of Y_p are all pure real or pure imaginary.

Placing (56) in (54), we have

$$C = HYH , \tag{58}$$

where

$$H = I_4 \oplus \left(I_4 \otimes F(2) \otimes I_{\frac{p'}{2}}\right) , \tag{59}$$

$$Y = C' \oplus (C' \otimes Y_p) . \tag{60}$$

Substitute (58) into (41), we will have the form

$$F_\Pi = HYHA . \tag{61}$$

Table 3-6. $F_\Pi = HYHA$

Factor	R.A	R.M
Y	$4(p^2-3p+3)$	$4(p-1)^2$
H	8(p-1)	0
A	$4(7p-4)$	0
F_Π	$4(p^2+8p-5)$	$4(p-1)^2$

Variant 4

Combining the ideas in Variant 2 and 3, we can have the factorization

$$F_\Pi = HYB_1H\,, \tag{62}$$

where by direct computation, the matrix

$$B_1 = HAH^{-1}\,, \tag{63}$$

is given by

$$B_1 = \begin{pmatrix} A' & A' \otimes \underline{e}^t \otimes 1^t_{\frac{p'}{2}} \\ -2A' \otimes \underline{e} \otimes 1_{\frac{p'}{2}} & A' \otimes I_{p'} \end{pmatrix}\,, \tag{64}$$

where

$$\underline{e} = \begin{pmatrix} 1 \\ 0 \end{pmatrix}\,. \tag{65}$$

Arithmetically, B_1 is equivalent to

$$A' \otimes B_1(p)\,, \tag{66}$$

where the matrix $B_1(p)$ is given as

$$B_1(p) = \begin{pmatrix} 1 & \underline{e}^t \otimes 1^t_{\frac{p'}{2}} \\ -2\underline{e} \otimes 1_{\frac{p'}{2}} & I_{p'} \end{pmatrix}\,. \tag{67}$$

Table 3-7. $F_\Pi = HYB_1H$

Factor	R.A	R.M
Y	$4(p^2-3p+3)$	$4(p-1)^2$
H	8(p-1)	0
B_1	20p-8	$\{4(p-1)\}$
F_Π	$4(p^2+6p-3)$	$4(p-1)^2+\{4(p-1)\}$

Table 3-8. $F_\Pi = HYB_1H$

Factor	R.A	R.M
12	96	$16+\{8\}*$
20	208	$64+\{16\}$
28	352	$144+\{24\}$
44	736	$400+\{40\}$

Programs have been written for n=12, 20, 28, 52 68 and etc. Efficiency comparisons with nearest point Cooley-Tukey algorithms will be given in section 13.

$\{8\}*$ in the case of N=12, the 8 multiplications by 2 can be eliminated by a trick which will be explained in section 12.

4. AN EXAMPLE OF N=4p: N=20

We choose N=20 as an example of N=4p to give a complete picture of the 4p algorithms, where p=5. The implementation results of N=20 are excellent for the 4p algorithms derived in this section comparing with the nearest Cooley-Tukey algorithms.

The algorithm for computing 20-point finite Fourier transform is designed from the ring-structure of Z/20. The set

$$\{5, 16\} \quad , \tag{1}$$

is the complete system of idempotents for Z/20, and the following conditions are satisfied

$$5^2 \equiv 5 \ mod\ 20 \ , \quad 16^2 \equiv 16 \ mod\ 20 \ , \tag{2}$$

$$0 \equiv 5 \cdot 16 \ mod\ 20 \ , \tag{3}$$

$$1 \equiv 5 + 16 \ mod\ 20 \ , \tag{4}$$

$$5 \equiv 1 \ mod\ 4 \ , \quad 5 \equiv 0 \ mod\ 5 \ , \tag{5}$$

$$16 \equiv 0 \ mod\ 4 \ , \quad 16 \equiv 1 \ mod\ 5 \ . \tag{6}$$

The complete system of idempotents (1) determines a ring-isomorphism σ

$$Z/4 \times Z/5 \cong Z/20 \ , \tag{7}$$

by the formula

$$\sigma(a, b) \equiv 5a + 16b \ mod\ 20 \ , \tag{8}$$

where $0 \leq a < 4$ and $0 \leq b < 5$. Table 4-1 explicitly describes σ.

Denote the unit group U(20) of Z/20 by U. The ring-isomorphism σ restricts to a group-isomorphism

$$U(4) \times U(5) \cong U \ . \tag{9}$$

Since

$$U(4) = \{1, 3\} \ , \tag{10}$$

$$U(5) = \{1, 2, 3, 4\} \ , \tag{11}$$

we can see from Table 4-1, that

$$U = \{1, 17, 9, 13, 11, 7, 19, 3\} \ . \tag{12}$$

Table 4-1.

$Z/4 \times Z/5$	Z/20
0,0	0
0,1	16
0,2	12
0,3	8
0,4	4
1,0	5
1,1	1
1,2	17
1,3	13
1,4	9
2,0	10
2,1	6
2,2	2
2,3	18
2,4	14
3,0	15
3,1	11
3,2	7
3,3	3
3,4	19

Of course, in the special case of the example, the unit group U can be determined directly, but for the purpose of generalization, we have emphasized the role of the ring-isomorphism σ.

The set of zero divisors D of Z/20 is

$$D = \{0, 2, 4, 5, 6, 8, 10, 12, 14, 15, 16, 18\} . \tag{13}$$

We will partition the set using the action of the unit group U. First the action of U on the idempotents gives rise to the sets

$$5U = \{5, 15\} , \tag{14}$$

$$16U = \{16, 12, 4, 8\} . \tag{15}$$

From (9) and (10) in section 3, we have

$$f_1 = 2e_1 = 10 , \tag{16}$$

$$f_2 = 2e_1 + e_2 = 6 . \tag{17}$$

Then the other elements of D will be given by

$$10U = \{10\} , \tag{18}$$

$$6U = \{6, 2, 14, 18\} , \tag{19}$$

and

$$0U = \{0\}\,. \tag{20}$$

The U-orbits

$$0U,\ 10U,\ 5U,\ 16U,\ 6U,\ U \tag{21}$$

partition the indexing set Z/20. The elements

$$0,\ 10,\ 5,\ 16,\ 6,\ 1 \tag{22}$$

are called orbit leaders.

Reorder the indexing set by the partition

$$\Pi = \{0;\ 10;\ 5, 15;\ 16, 12, 4, 8;\ 6, 2, 14, 18;\ 1, 17, 9, 13, 11, 7, 19, 3\} \tag{23}$$

and denote the corresponding permutation by P. The matrix

$$F_\Pi = PF(20)P^{-1} \tag{24}$$

computes 20-point FFT for input and output data indexed by Π. It is given by

$$F_\Pi = \left[v^{\Pi(j)\Pi(k)}\right]_{0\le j,k<20}, \quad v = e^{2\pi i/20}\,. \tag{25}$$

We describe F_Π by the following sequence of defination. Set

$$F_\Pi(4) = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & & \\ 1 & -1 & & \end{bmatrix}, \qquad \begin{matrix} & & & \\ & & & \\ & & C_4 & \\ & & & \end{matrix} \tag{26}$$

where

$$C_4 = \begin{bmatrix} i & -i \\ -i & i \end{bmatrix}, \tag{27}$$

and set

$$C_5 = \begin{bmatrix} u^4 & u^3 & u & u^2 \\ u^3 & u & u^2 & u^4 \\ u & u^2 & u^4 & u^3 \\ u^2 & u^4 & u^3 & u \end{bmatrix}, \quad u = v^4 = e^{2\pi i/5}\,. \tag{28}$$

The matrix $F_\Pi(4)$ is the same matrix given in (2) of section 2, and C_5 is rotated 5-point Winograd core. The matrix F_Π is

$$F_\Pi = \begin{bmatrix} F_\Pi(4) & F_\Pi(4) \otimes 1_4^t \\ F_\Pi(4) \otimes 1_4 & F_\Pi(4) \otimes C_5 \end{bmatrix}. \tag{29}$$

Since $C_5 1_4 = -1_4$, we can factor F_Π as

$$F_\Pi = \left(F_\Pi(4) \oplus (F_\Pi(4) \otimes C_5)\right) \begin{bmatrix} I_4 & I_4 \otimes 1_4^t \\ -I_4 \otimes 1_4 & I_{16} \end{bmatrix}. \tag{30}$$

By the same argument as in section 2,

$$F_{\Pi}(4) = C'A' , \tag{31}$$

where

$$C' = I_2 \oplus \begin{bmatrix} 1 & i \\ 1 & -i \end{bmatrix} , \tag{32}$$

$$A' = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & -1 \end{bmatrix} . \tag{33}$$

Placing (31) into (30), we have

$$F_{\Pi} = CA , \tag{34}$$

where

$$C = C' \oplus (C' \otimes C_5) , \tag{35}$$

$$A = \left(A' \oplus (A' \otimes I_4)\right) \begin{bmatrix} I_4 & I_4 \otimes 1_4^t \\ -I_4 \otimes 1_4 & I_{16} \end{bmatrix} , \tag{36}$$

which can be written as

$$A = \begin{bmatrix} A' & A' \otimes 1_4^t \\ -A' \otimes 1_4 & A' \otimes I_4 \end{bmatrix} . \tag{37}$$

Arithmetically, A is equivalant to

$$A' \otimes A(5) , \tag{38}$$

where the matrix A(5) is given as

$$A(5) = \begin{bmatrix} 1 & 1_4^t \\ -1_4 & I_4 \end{bmatrix} . \tag{39}$$

Direct computation gives the arithmetic count as in Table 4-2.

Table 4-2. $F_{\Pi} = CA$

Factor	R.A	R.M
C	244	256
A	124	0
F_{Π}	368	256

<u>Variant</u> 1.

The matrix C_5 can be diagonalised by the formula

$$D_5 = F(4)^{-1} C_5 F(4)^{-1} , \tag{40}$$

where

$$D_5 = \frac{1}{4}\begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & u^4+iu^3-u-iu^2 & 0 & 0 \\ 0 & 0 & u^4-u^3+u-u^2 & 0 \\ 0 & 0 & 0 & u^4-iu^3-u+iu^2 \end{bmatrix}, \quad u = e^{2\pi i/5}. \tag{41}$$

Set

$$F_0 = I_4 \oplus \left(I_4 \otimes F(4)\right), \tag{42}$$

then

$$C = F_0 D F_0, \tag{43}$$

where

$$D = C' \oplus (C' \otimes D_5). \tag{44}$$

Placing (43) into (34), we have

$$F_\Pi = F_0 D F_0 A. \tag{45}$$

Table 4-3. $F_\Pi = F_0 D F_0 A$

Factor	R.A	R.M
F_0	64	0
D	36	48
A	124	0
F_Π	288	48

Variant 2.

We rewrite (45) as

$$F_\Pi = F_0 D B F_0, \tag{46}$$

where the matrix

$$B = F_0 A F_0^{-1} \tag{47}$$

is given as

$$B = \begin{pmatrix} A' & A' \otimes [1000] \\ -4A' \otimes \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} & A' \otimes I_{p'} \end{pmatrix}. \tag{48}$$

Table 4-4. $F_\Pi = F_0 D B F_0$

Factor	R.A	R.M
F_0	64	0
D	36	48
B	76	{8}
F_Π	240	48+{8}

Variant 3.

Replacing the complex multiplications by real multiplications, we set

$$C = C' \oplus (C' \otimes C_5) , \tag{49}$$

where C_5 can be written as

$$C_5 = \big(F(2) \otimes I_2\big) Y_5 \big(F(2) \otimes I_2\big) , \tag{50}$$

where

$$Y_5 = \frac{1}{2}\big((X_5 + X_5^*) \oplus (X_5 - X_5^*)\big) , \tag{51}$$

and X_5 is 2 by 2 matrix given as

$$X_5 = \begin{bmatrix} u^4 & u^3 \\ u^3 & u \end{bmatrix} . \tag{52}$$

Factorize the matrix C as

$$C = HYH , \tag{53}$$

where

$$H = I_4 \oplus \big(I_4 \otimes F(2) \otimes I_2\big) , \tag{54}$$

$$Y = C' \oplus (C' \otimes Y_5) . \tag{55}$$

Then

$$F_\Pi = HYHA . \tag{56}$$

Table 4-5. $F_\Pi = HYHA$

Factor	R.A	R.M
H	32	0
Y	52	64
A	124	0
F_Π	240	64

Variant 4.

We can rewrite (56) as

$$F_{\Pi} = HYB_1H\,, \tag{57}$$

where the matrix

$$B_1 = HAH^{-1} \tag{58}$$

is, by direct computation, given by

$$B_1 = \begin{bmatrix} A' & A' \otimes \underline{e}^t \otimes 1_2^t \\ -2A' \otimes \underline{e} \otimes 1_2 & A' \otimes I_4 \end{bmatrix}\,, \tag{59}$$

where

$$\underline{e} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}\,.$$

Arithmetically, B is equivalent to

$$A' \otimes B_1(5)\,, \tag{60}$$

where the matrix

$$B_1(5) = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ -2 & 1 & 0 & 0 & 0 \\ -2 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}\,. \tag{61}$$

Table 4-6. $F_{\Pi} = HYB_1H$

Factor	R.A	R.M
H	32	0
Y	52	64
B	92	{16}
F_{Π}	208	64+{16}

Multiplication by 2 is placed in brackets.

5. THE CASE N=4pq

In this section, we extend the 4p algorithms to the case of 4pq. The algorithms of 4pq case will be designed based on the ring-structure of Z/4pq.

By Chinese Remainder Theorem, we know that there exists a ring- isomorphism Φ

$$Z/4 \times Z/p \times Z/q \cong Z/4pq\,, \tag{1}$$

which restricts to a group-isomorphism

$$U(4) \times U(p) \times U(q) \cong U(4pq)\,. \tag{2}$$

Let $z_1 = 3$, z_2 and z_3 be the generators of U(4), U(p) and U(q) respectively, and let

$$\{e_1,\ e_2,\ e_3\} \tag{3}$$

be the system of idempotents for the ring of Z/4pq, it satisfies the following conditions

$$e_1 \equiv 1\ mod\ 4,\ \ e_1 \equiv 0\ mod\ p\ \ and\ \ e_1 \equiv 0\ mod\ q\,, \tag{4}$$

$$e_2 \equiv 0\ mod\ 4,\ \ e_2 \equiv 1\ mod\ p\ \ and\ \ e_2 \equiv 0\ mod\ q\,, \tag{5}$$

$$e_3 \equiv 0\ mod\ 4,\ \ e_3 \equiv 0\ mod\ p\ \ and\ \ e_3 \equiv 1\ mod\ q\,, \tag{6}$$

and

$$e_j^2 \equiv e_j\ mod\ n\,,\quad 1 \le j \le 3\,, \tag{7}$$

$$e_j e_k \equiv 0\ mod\ n\,,\quad 1 \le j,k \le 3,\quad j \ne k\,, \tag{8}$$

$$e_1 + e_2 + e_3 \equiv 1\ mod\ n\,. \tag{9}$$

Define the mapping by the formula

$$\Phi(a_1, a_2, a_3) \equiv a_1 e_1 + a_2 e_2 + a_3 e_3\,, \tag{10}$$

where $0 \le a_1 < 4$, $0 \le a_2 < p$, $0 \le a_3 < q$. Then the unit group

$$U \equiv U(4pq) \tag{11}$$

is given by

$$z_1^j e_1 + z_2^k e_2 + z_3^l e_3,\quad 0 \le j < 2,\ \ 0 \le k < p-1,\ \ 0 \le l < q-1. \tag{12}$$

U will be ordered by the assumption given in (12), l is the fastest variable, k is the next fastest and j is the slowest variable,

$$e_1 + e_2 + z_3^l e_3,\quad 0 \le l < q-1\,, \tag{13}$$

$$e_1 + z_2 e_2 + z_3^l e_3, \quad 0 \le l < q-1\,, \tag{14}$$

$$e_1 + z_2^2 e_2 + z_3^l e_3, \quad 0 \le l < q-1\,, \tag{15}$$

...

$$e_1 + z_2^{p-2} e_2 + z_3^l e_3, \quad 0 \le q-1\,, \tag{16}$$

and then

$$z_1 e_1 + e_2 + z_3^l e_3, \quad 0 \le l < q-1\,, \tag{17}$$

...

$$z_1 e_1 + z_2^{p-2} e_2 + z_3^l e_3, \quad 0 \le l < q-1\,. \tag{18}$$

The order of U is 2(p-1)(q-1).

The indexing set Z/4pq will be partitioned and ordered by U-orbits. First using (7)-(9), we have

$$e_1 U = \{e_1,\ z_1 e_1\}\,, \tag{19}$$

$$e_2 U = \{z_2^k e_2 \quad : 0 \le k < p-1\}\,, \tag{20}$$

$$e_3 U = \{z_3^l e_3 \quad : 0 \le l < q-1\}\,. \tag{21}$$

Under the isomorphism Φ, $e_1 U$ corresponds to U(4), $e_2 U$ corresponds U(p) and $e_3 U$ corresponds to U(q). Define

$$f_1 = e_1 + e_2\,, \tag{22}$$

$$f_2 = e_1 + e_3\,, \tag{23}$$

$$f_3 = e_2 + e_3\,, \tag{24}$$

then

$$f_1 U = \{z_1^j e_1 + z_2^k e_2 \quad : 0 \le j < 2,\ 0 \le k < p-1\}\,, \tag{25}$$

$$f_2 U = \{z_1^j e_1 + z_3^k e_3 \quad : 0 \le j < 2,\ 0 \le k < q-1\}\,, \tag{26}$$

$$f_3 U = \{z_2^k e_2 + z_3^l e_3 \quad : 0 \le k < p-1,\ 0 \le l < q-1\}\,. \tag{27}$$

The U-orbits are ordered by the same convention on the variables j,k, and l introduced above. Under the isomorphism Φ, $f_1 U$ corresponds to $U(4) \times U(p)$, $f_2 U$ corresponds to $U(4) \times U(q)$ and $f_3 U$ corresponds to $U(p) \times U(q)$. Define

$$g_1 = 2e_1\,, \tag{28}$$

$$g_2 = 2e_1 + e_2\,, \tag{29}$$

$$g_3 = 2e_1 + e_3\,, \tag{30}$$

$$h = 2e_1 + e_2 + e_3\,, \tag{31}$$

then

$$g_1U = \{2e_1\}\,, \tag{32}$$

$$g_2U = \{2e_1 + z_2^k e_2 \quad : 0 \le k < p-1\}\,, \tag{33}$$

$$g_3U = \{2e_1 + z_3^l e_3 \quad : 0 \le l < q-1\}\,, \tag{34}$$

$$hU = \{2e_1 + z_2^k e_2 + z_3^l e_3 \quad : 0 \le k < p-1,\ 0 \le l < q-1\}\,. \tag{35}$$

Again, these sets are ordered by the above convention of running first through l , $0 \le l < q-1$, then through k, $0 \le k < p-1$. Under the ring-isomorphism Φ, g_1U corresponds to $\{2e_1\} \times \{0\}$, g_2U corresponds to $\{2e_1\} \times U(p)$, g_3U corresponds to $\{2e_1\} \times U(q)$ and hU corresponds to $\{2e_1\} \times U(p) \times U(q)$.

The U-orbits corresponding to the elements

$$0,\ g_1,\ e_1,\ e_2,\ g_2,\ f_1,\ e_3,\ g_3,\ f_2,\ f_3,\ h,\ 1 \tag{36}$$

define a partition of Z/4pq. The set of U-orbits will be ordered by (36) inducing an ordering or permutation Π of Z/4pq. Denote the corresponding permutation matrix by P and set

$$F_\Pi = PF(4pq)P^{-1}\,. \tag{37}$$

If a and b are in the list (36), then the submatrix of F_Π

$$W(a,b) = (\,w^{rs}\,)_{r \in aU,\ s \in bU}\,, \qquad w = e^{2\pi i/4pq}\,, \tag{38}$$

corresponds to the Cartesian product $aU \times bU$. Set

$$C_4 = W(e_1, e_1)\,, \tag{39}$$

$$C_p = W(e_2, e_2)\,, \tag{40}$$

$$C_q = W(e_3, e_3)\,, \tag{41}$$

direct computation from (19)-(21) shows that

$$C_4 = \begin{pmatrix} w^{e_1} & -w^{e_1} \\ -w^{e_1} & w^{e_1} \end{pmatrix}, \quad w = e^{2\pi i/4pq}\,, \tag{42}$$

$$C_p = \left[(w^{e_2})^{z_2^{j+k}}\right]_{0 \le j,k < p-1}\,, \qquad w = e^{2\pi i/4pq}\,, \tag{43}$$

$$C_q = \left[(w^{e_3})^{z_3^{l+j}}\right]_{0 \le j,l < q-1}\,, \qquad w = e^{2\pi i/4pq}\,. \tag{44}$$

Consider the submatrix W(1,1), a typical product $r \cdot s$, r,s $\in$ U is given by

$$r \cdot s = (z_1^j e_1 + z_2^k e_2 + z_3^l e_3)(z_1^{j'} e_1 + z_2^{k'} e_2 + z_3^{l'} e_3) = z_1^{j+j'} e_1 + z_2^{k+k'} e_2 + z_3^{l+l'} e_3\,, \tag{45}$$

then

$$w^{r \cdot s} = (w^{e_1})^{z_1^{j+j'}} (w^{e_2})^{z_2^{k+k'}} (w^{e_3})^{z_3^{l+l'}}\,, \tag{46}$$

running first over $0 \leq l, l' < q-1$, and then over $0 \leq k, k' < p-1$ and at last running over $0 \leq j, j' < 2$. We form from (46) the matrix

$$C_4 \otimes C_p \otimes C_q \,, \tag{47}$$

which is W(1,1).

The same arguments can be used to describe the remaining submatrices of F_Π. For example, consider the submatrix W(1,h), a typical product $r \cdot s$, $r \in U$, $s \in hU$ is given by

$$r \cdot s = (z_1^j e_1 + z_2^k e_2 + z_3^l e_3)(2e_1 + z_2^{k'} e_2 + z_3^{l'} e_3) = 2e_1 + z_2^{k+k'} e_2 + z_3^{l+l'} e_3 \,, \tag{48}$$

then

$$w^{r \cdot s} = (-i)^2 (w^{e_2})^{z_2^{k+k'}} (w^{e_3})^{z_3^{l+l'}} \,, \tag{49}$$

running over $0 \leq l, l' < q-1$, and then over $0 \leq k, k' < p-1$, we form from (49) the matrix

$$-C_p \otimes C_q \,, \tag{50}$$

which is independent of the variable j ,$0 \leq j < 2$. Running over $0 \leq j < 2$, we form from (50) the matrix

$$-1_2^t \otimes C_p \otimes C_q \,, \tag{51}$$

which is W(1,h).

Continuing in this way, we can construct F_Π by the following sequence of constructions. Set

$$F_\Pi(4) = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & w^{e_1} & -w^{e_1} \\ 1 & -1 & -w^{e_1} & w^{e_1} \end{pmatrix} \,, \tag{52}$$

which should be compared to (35) of section 3. Set

$$F_\Pi(4p) = \begin{pmatrix} F_\Pi(4) & F_\Pi(4) \otimes 1_{p-1}^t \\ F_\Pi(4) \otimes 1_{p-1} & F_\Pi(4) \otimes C_p \end{pmatrix} \,, \tag{53}$$

which should be compared with (36) of section 3. Direct computation shows that

$$F_\Pi = \begin{pmatrix} F_\Pi(4p) & F_\Pi(4p) \otimes 1_{q-1}^t \\ F_\Pi(4p) \otimes 1_{q-1} & F_\Pi(4p) \otimes C_q \end{pmatrix} \,. \tag{54}$$

The matrices (53) and (54) should be compared to the matrix (36) constructed in section 3. Arguing as in section 3,

$$F_\Pi(4) = C'' A'' \,, \tag{55}$$

where

$$C'' = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & w^{e_1} \\ 0 & 0 & 1 & -w^{e_1} \end{pmatrix} \,, \tag{56}$$

$$A'' = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & -1 \end{pmatrix} . \tag{57}$$

These are the same as the matrices given in (39) and (40) in section 3.

Then by the same arguments,

$$F_{\Pi}(4p) = C'A' , \tag{58}$$

where

$$C' = C'' \oplus (C'' \otimes C_p) , \tag{59}$$

$$A' = \begin{pmatrix} A'' & A'' \otimes 1^t_{p-1} \\ -A'' \otimes 1_{p-1} & A'' \otimes I_{p-1} \end{pmatrix} , \tag{60}$$

and

$$F_{\Pi} = CA , \tag{61}$$

where

$$C = C' \oplus (C' \otimes C_q) = C'' \oplus (C'' \otimes C_p) \oplus (C'' \otimes C_q) \oplus (C'' \otimes C_p \otimes C_q) , \tag{62}$$

$$A = \begin{pmatrix} A' & A' \otimes 1^t_{q-1} \\ -A' \otimes 1_{q-1} & A' \otimes I_{q-1} \end{pmatrix} . \tag{63}$$

Direct computation will give the arithmetic count as

for matrix C:

$$R.A = 4(4p^2 - 9p + 6)q + 8p(q-1)(2q-3) , \tag{64}$$

$$R.M = 16[(p-1)^2 q + p(q-1)^2] , \tag{65}$$

and for matrix A:

$$R.A = 4(11pq - 4p - 4q) , \tag{66}$$

$$R.M = 0 . \tag{67}$$

Variant 1.

All the methods of designing variants developed in the preceeding sections apply to (61). First by convolution theorem

$$C_p = F(p')D_pF(p') , \tag{68}$$

$$C_q = F(q')D_qF(q') , \tag{69}$$

where D_p and D_q are diagonal matrices.

Define

$$F_0 = I_4 \oplus (I_4 \otimes F(p')) \oplus (I_4 \otimes F(q')) \oplus (I_4 \otimes F(p') \otimes F(q')) , \tag{70}$$

we will have that

$$C = F_0 D F_0 \,, \tag{71}$$

where

$$D = C'' \oplus (C'' \otimes D_p) \oplus (C'' \otimes D_q) \oplus (C'' \otimes D_p \otimes D_q) \,. \tag{72}$$

Then (61) becomes

$$F_\Pi = F_0 D F_0 A \,. \tag{73}$$

The arithmetic count will become

for matrix F_0:

$$R.A = 4(A_p q + A_q p) \,, \tag{74}$$

$$R.M = 4(M_p q + M_q p) \,, \tag{75}$$

and for matrix D:

$$R.A = 4\big(5pq - 6(p+q)\big) \,, \tag{76}$$

$$R.M = 32(pq - p - q) \,. \tag{77}$$

Variant 2.

Rewrite (73) as

$$F_\Pi = F_0 D B F_0 \,, \tag{78}$$

where

$$B = \begin{bmatrix} B' & B' \otimes \underline{e}^t(q) \\ -q' B' \otimes \underline{e}(q) & B' \otimes I_{q'} \end{bmatrix} \,, \tag{79}$$

and

$$B' = \begin{bmatrix} A'' & A'' \otimes \underline{e}^t(p) \\ -p' A'' \otimes \underline{e}(p) & A'' \otimes I_{p'} \end{bmatrix} \,. \tag{80}$$

The arithmetic count for matrix B is

$$R.A = 4(3pq + 4p + 4q) \,, \tag{81}$$

and the number of multiplications, for real number multiplied by integer , is given as

$$R.M = \{8(p+q)\} \,. \tag{82}$$

Variant 3.

Complex multiplications can be eliminated by the following method:

let

$$C_p = \left(F(2) \otimes I_{\frac{q'}{2}}\right) Y_p \left(F(2) \otimes I_{\frac{q'}{2}}\right) , \tag{83}$$

and

$$C_q = \left(F(2) \otimes I_{\frac{q'}{2}}\right) Y_q \left(F(2) \otimes I_{\frac{q'}{2}}\right) , \tag{84}$$

where

$$Y_p = \frac{1}{2} \begin{pmatrix} X_P + X_p^* & 0 \\ 0 & X_p - X_p^* \end{pmatrix} , \tag{85}$$

$$Y_q = \frac{1}{2} \begin{pmatrix} X_q + X_q^* & 0 \\ 0 & X_q - X_q^* \end{pmatrix} . \tag{86}$$

The entries of Y_p and Y_q are either real or purely imaginary implying that only real multiplications are required to compute their actions.

Set

$$H' = I_4 \oplus \left(I_4 \otimes F(2) \otimes I_{\frac{q'}{2}}\right) , \tag{87}$$

$$H = H' \oplus \left(H' \otimes F(2) \otimes I_{\frac{q'}{2}}\right) , \tag{88}$$

$$Y' = C'' \oplus \left(C'' \otimes Y_p\right) , \tag{89}$$

and

$$Y = Y' \oplus \left(Y' \otimes Y_q\right) . \tag{90}$$

Then

$$C = HYH , \tag{91}$$

and

$$F_\Pi = HYHA . \tag{92}$$

The arithmetic count now becomes

for matrix H:

$$R.A = 8(2pq - p - q) , \tag{93}$$

$$R.M = 0 , \tag{94}$$

and for the matrix Y:

$$R.A = 4(p^2 - 3p + 3)q + 4p(q-1)(q-3) , \tag{95}$$

$$R.M = 4(p-1)^2 q + 4p(q-1)^2 . \tag{96}$$

<u>Variant</u> 4.

We rewrite (92) as

$$F_\Pi = HYB_1H , \tag{97}$$

where

$$B_1 = HAH^{-1} . \tag{98}$$

B_1 can be explicitly written as

$$B_1 = \begin{pmatrix} B' & B' \otimes \underline{f}^t(q) \\ -2B'\underline{f}(q) & B' \otimes I_{q'} \end{pmatrix} , \tag{99}$$

where

$$B' = H'A'H'^{-1} , \tag{100}$$

which is given as

$$B' = \begin{pmatrix} A'' & A'' \otimes \underline{f}^t(p) \\ -2A'' \otimes \underline{f}(p) & A'' \otimes I_{p'} \end{pmatrix} , \tag{101}$$

where $\underline{f}(n)$ is defined as

$$\underline{f}(n) = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \otimes 1_{\frac{n-1}{2}} . \tag{102}$$

The arithmetic count of B_1 is

$$R.A = 4(7pq - 2p - 2q) , \tag{103}$$

and the number of multiplications, real number multiplied by 2, is

$$R.M = \{4(2pq - p - q)\} . \tag{104}$$

6. AN EXAMPLE OF N=4pq: N=60

In this section, we will pick up N=60 as an example of N=4pq case to give a complete picture of the algorithms, and also that N=60 is a very important case in applications. From the ring-structure of Z/60, the following elements

$$e_1 = 45, \ \ e_2 = 40, \ \ e_3 = 36, \tag{1}$$

define a system of idempotents for the ring Z/60, since

$$e_j^2 \equiv e_j \ , \quad j = 1, 2, 3, \tag{2}$$

$$e_j e_k \equiv 0 \ mod \, 60 \ , \quad 1 \leq j, k \leq 3 \ , j \neq k \ , \tag{3}$$

$$1 \equiv e_1 + e_2 + e_3 \ mod \, 60 \ . \tag{4}$$

A ring-isomorphism

$$\Phi : Z/4 \times Z/3 \times Z/5 \cong Z/60 \tag{5}$$

is defined by the formula

$$\Phi(a_1, a_2, a_3) \equiv a_1 e_1 + a_2 e_2 + a_3 e_3 \ , \tag{6}$$

where $0 \leq a_1 < 4$, $0 \leq a_2 < 3$, $0 \leq a_3 < 5$. Choose generators

$$z_1 = 3, \ z_2 = 2, \ and \ z_3 = 2 \ , \tag{7}$$

of the unit group U(4), U(3) and U(5), respectively. Then the unit group

$$U \equiv U(60) \tag{8}$$

is given by

$$z_1^j e_1 + z_2^k e_2 + z_3^l e_3, \quad 0 \leq j < 2, \ 0 \leq k < 2, \ 0 \leq l < 4. \tag{9}$$

U will be ordered by the assumption that in (9), l is the fastest variable, k is the next fastest and j is the slowest variable. The order of U is 16. Explicitly, U is given in order by

$$e_1 + e_2 + z_3^l e_3, \quad 0 \leq l < 4 \ , \tag{10}$$

$$e_1 + z_2 e_2 + z_3^l e_3, \quad 0 \leq l < 4 \ , \tag{11}$$

$$z_1 e_1 + e_2 + z_3^l e_3, \quad 0 \leq l < 4 \ , \tag{12}$$

$$z_1 e_1 + z_2 e_2 + z_3^l e_3, \quad 0 \leq l < 4 \ . \tag{13}$$

The indexing set Z/60 will be partitioned and ordered by U-orbits. First using (2)-(4), we have

$$e_1 U = \{e_1, \ z_1 e_1\} = \{45, \ 15\} \ , \tag{14}$$

$$e_2U = \{e_2,\ z_2e_2\} = \{40,\ 20\}\,, \tag{15}$$

$$e_3U = \{e_3,\ z_3e_3,\ z_3^2e_3,\ z_3^3e_3\} = \{36,\ 12,\ 24,\ 48\}\,. \tag{16}$$

Under the isomorphism Φ, e_1U corresponds to U(4), e_2U corresponds U(3) and e_3U corresponds to U(5). Define

$$f_1 = e_1 + e_2 = 25\,, \tag{17}$$

$$f_2 = e_1 + e_3 = 21\,, \tag{18}$$

$$f_3 = e_2 + e_3 = 16\,, \tag{19}$$

then

$$f_1U = \{z_1^je_1 + z_2^ke_2 \quad : 0 \le j < 2,\ 0 \le k < 2\}\,, \tag{20}$$

$$f_2U = \{z_1^je_1 + z_3^ke_3 \quad : 0 \le j < 2,\ 0 \le k < 4\}\,, \tag{21}$$

$$f_3U = \{z_2^ke_2 + z_3^le_3 \quad : 0 \le k < 2,\ 0 \le l < 4\}\,. \tag{22}$$

The U-orbits are ordered by the same convention on the variables j,k, and l introduced above. Under the isomorphism Φ, f_1U corresponds to $U(4) \times U(3)$, f_2U corresponds to $U(4) \times U(5)$ and f_3U corresponds to $U(3) \times U(5)$. Define

$$g_1 = 2e_1 = 30\,, \tag{23}$$

$$g_2 = 2e_1 + e_2 = 10\,, \tag{24}$$

$$g_3 = 2e_1 + e_3 = 6\,, \tag{25}$$

$$h = 2e_1 + e_2 + e_3 = 46\,, \tag{26}$$

then

$$g_1U = \{2e_1\}\,, \tag{27}$$

$$g_2U = \{2e_1 + z_2^ke_2 \quad : 0 \le k < 2\}\,, \tag{28}$$

$$g_3U = \{2e_1 + z_3^le_3 \quad : 0 \le l < 4\}\,, \tag{29}$$

$$hU = \{2e_1 + z_2^ke_2 + z_3^le_3 \quad : 0 \le k < 2,\ 0 \le l < 4\}\,. \tag{30}$$

Again, these sets are ordered by the above convention of running first through l , $0 \le l < 4$, then through k, $0 \le k < 2$. Under the ring-isomorphism Φ, g_1U corresponds to $\{2e_1\} \times \{0\}$, g_2U corresponds to $\{2e_1\} \times U(3)$, g_3U corresponds to $\{2e_1\} \times U(5)$ and hU corresponds to $\{2e_1\} \times U(3) \times U(5)$.

The U-orbits corresponding to the elements

$$0,\ g_1,\ e_1,\ e_2,\ g_2,\ f_1,\ e_3,\ g_3,\ f_2,\ f_3,\ h,\ 1 \tag{31}$$

define a partition of Z/60. The set of U-orbits will be ordered by (31) inducing an ordering or permutation Π of Z/60. Denote the corresponding permutation matrix by P and set

$$F_\Pi = PF(60)P^{-1}\,. \tag{32}$$

If a and b are in the list (31), then the submatrix of F_Π

$$W(a,b) = \left(w^{rs}\right)_{r\in aU,\, s\in bU}\,, \qquad w = e^{2\pi i/60}\,, \tag{33}$$

corresponds to the Cartesian product $aU \times bU$. Set

$$C_4 = W(e_1,e_1)\,, \tag{34}$$

$$C_3 = W(e_2,e_2)\,, \tag{35}$$

$$C_5 = W(e_3,e_3)\,, \tag{36}$$

direct computation from (14)-(16) shows that

$$C_4 = \begin{pmatrix} -i & i \\ i & -i \end{pmatrix}\,, \tag{37}$$

$$C_3 = \begin{pmatrix} u^2 & u \\ u & u^2 \end{pmatrix}\,, \qquad u = e^{2\pi i/3}\,, \tag{38}$$

$$C_5 = \begin{pmatrix} v^3 & v & v^2 & v^4 \\ v & v^2 & v^4 & v^3 \\ v^2 & v^4 & v^3 & v \\ v^4 & v^3 & v & v^2 \end{pmatrix}\,, \qquad v = e^{2\pi i/5}\,. \tag{39}$$

Consider the submatrix W(1,1), a typical product $r\cdot s$, r,s ∈ U is given by

$$r\cdot s = (z_1^j e_1 + z_2^k e_2 + z_3^l e_3)(z_1^{j'} e_1 + z_2^{k'} e_2 + z_3^{l'} e_3) = z_1^{j+j'} e_1 + z_2^{k+k'} e_2 + z_3^{l+l'} e_3\,, \tag{40}$$

then

$$w^{r\cdot s} = \left(w^{e_1}\right)^{z_1^{j+j'}} \left(w^{e_2}\right)^{z_2^{k+k'}} \left(w^{e_3}\right)^{z_3^{l+l'}}\,, \tag{41}$$

which, since $w^{e_1} = -i$, $w^{e_2} = u^2$, *and* $w^{e_3} = v^3$, can be rewritten as

$$w^{r\cdot s} = (-i)^{z_1^{j+j'}} (u^2)^{z_2^{k+k'}} (v^3)^{z_3^{l+l'}}\,, \tag{42}$$

running first over $0 \le l, l' < 4$, we form from (42) the matrix

$$(-i)^{z_1^{j+j'}} (u^2)^{z_2^{k+k'}} C_5\,, \tag{43}$$

running over $0 \le k, k' < 2$, we form from (43) the matrix

$$(-i)^{z_1^{j+j'}} C_3 \otimes C_5\,, \tag{44}$$

and running over $0 \leq j, j' < 2$, we form from (44) the matrix

$$C_4 \otimes C_3 \otimes C_5 \,, \tag{45}$$

which is W(1,1).

The same arguments can be used to describe the remaining submatrices of F_Π. For example, consider the submatrix W(1,h), a typical product $r \cdot s$, $r \in U$, $s \in Uh$ is given by

$$r \cdot s = (z_1^j e_1 + z_2^k e_2 + z_3^l e_3)(2e_1 + z_2^{k'} e_2 + z_3^{l'} e_3) = 2e_1 + z_2^{k+k'} e_2 + z_3^{l+l'} e_3 \,, \tag{46}$$

then

$$W^{r \cdot s} = (-i)^2 (u^2)^{z_2^{k+k'}} (v^3)^{z_3^{l+l'}} \,, \tag{47}$$

running over $0 \leq l, l' < 4$, we form from (47) the matrix

$$-1(u^2)^{z_2^{k+k'}} C_5 \,, \tag{48}$$

running over $0 \leq k, k' < 2$, we form from (48) the matrix

$$-C_3 \otimes C_5 \,, \tag{49}$$

which is independent of the final variable j ,$0 \leq j < 2$. Running over $0 \leq j < 2$, we form from (49) the matrix

$$-1_2^t \otimes C_3 \otimes C_5 \,, \tag{50}$$

which is W(1,h).

Continuing in this way, we can construct F_Π by the following sequence of constructions. Set

$$F_\Pi(4) = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -i & i \\ 1 & -1 & i & -i \end{pmatrix} \,, \tag{51}$$

and

$$F_\Pi(12) = \begin{pmatrix} F_\Pi(4) & F_\Pi(4) \otimes 1_2^t \\ F_\Pi(4) \otimes 1_2 & F_\Pi(4) \otimes C_3 \end{pmatrix} \,, \tag{52}$$

direct computation shows that

$$F_\Pi = \begin{pmatrix} F_\Pi(12) & F_\Pi(12) \otimes 1_4^t \\ F_\Pi(12) \otimes 1_4 & F_\Pi(12) \otimes C_5 \end{pmatrix} \,. \tag{53}$$

Arguing as in section 3,

$$F_\Pi(4) = C'' A'' \,, \tag{54}$$

where

$$C'' = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & -i \\ 0 & 0 & 1 & i \end{pmatrix} \,, \tag{55}$$

$$A'' = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & -1 \end{pmatrix} . \tag{56}$$

Then

$$F_{\Pi}(12) = C'A' , \tag{57}$$

where

$$C' = C'' \oplus (C'' \otimes C_3) , \tag{58}$$

$$A' = \begin{pmatrix} A'' & A'' \otimes 1_2^t \\ -A'' \otimes 1_2 & A'' \otimes I_2 \end{pmatrix} , \tag{59}$$

and

$$F_{\Pi} = CA , \tag{60}$$

where

$$C = C' \oplus (C' \otimes C_5) = C'' \oplus (C'' \otimes C_3) \oplus (C'' \otimes C_5) \oplus (C'' \otimes C_3 \otimes C_5) , \tag{61}$$

$$A = \begin{pmatrix} A' & A' \otimes 1_4^t \\ -A' \otimes 1_4 & A' \otimes I_4 \end{pmatrix} . \tag{62}$$

Direct computation of (60) will give the arithmetic count as in Table 6-1.

Table 6-1. $F_{\Pi} = CA$

Factor	R.A	R.M
C	972	1088
A	532	0
F_{Π}	1504	1088

Variant 1:

All the methods of designing variants developed in the preceeding sections apply to (60). Let us first apply convolution theorem to

$$C_3 = F(2)D_3F(2) , \tag{63}$$

and

$$C_5 = F(4)D_5F(4) , \tag{64}$$

where

$$D_3 = \begin{bmatrix} -0.5 & 0 \\ 0 & \frac{1}{2}(u^2 - u) \end{bmatrix} , \tag{65}$$

and

$$D_5 = diag \begin{bmatrix} -0.25 \\ \frac{1}{4}(v^3 - iv - v^2 + iv^4) \\ \frac{1}{4}(v^3 - v + v^2 - v^4) \\ \frac{1}{4}(v^3 + iv - v^2 - iv^4) \end{bmatrix} . \tag{66}$$

Define

$$F_0 = I_4 \oplus (I_4 \otimes F(2)) \oplus (I_4 \otimes F(4)) \oplus (I_4 \otimes F(2) \otimes F(4)) , \tag{67}$$

then

$$C = F_0 D F_0 , \tag{68}$$

where

$$D = C'' \oplus (C'' \otimes D_3) \oplus (C'' \otimes D_5) \oplus (C'' \otimes D_3 \otimes D_5) . \tag{69}$$

Substitute (68) to (60), we will have

$$F_\Pi = F_0 D F_0 A . \tag{70}$$

Table 6-2. $F_\Pi = F_0 D F_0 A$

Factor	R.A	R.M
F_0	272	0
D	108	224
A	532	0
F_Π	1184	224

<u>Variant</u> **2:**

We can rewrite (70) as

$$F_\Pi = F_0 D B F_0 , \tag{71}$$

wher B is given by

$$B = F_0 A F_0^{-1} , \tag{72}$$

which can be explicitly written as

$$B = \begin{bmatrix} B' & B' \otimes [1\,0\,0\,0] \\ -4B' \otimes \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} & B' \otimes I_4 \end{bmatrix} , \tag{73}$$

and

$$B' = \begin{bmatrix} A'' & A'' \otimes [1\,0] \\ -2A'' \otimes \begin{pmatrix} 1 \\ 0 \end{pmatrix} & A'' \otimes I_2 \end{bmatrix} . \tag{74}$$

Table 6-3. $F_\Pi = F_0 D B F_0$

Factor	R.A	R.M
F_0	272	0
D	108	224
B	308	{64}
F_Π	960	224+{64}

Variant 3:

Complex multiplications can be reduced by the following method, let

$$C_3 = F(2) Y_3 F(2)\,, \tag{75}$$

$$C_5 = \big(F(2) \otimes I_2\big) Y_5 \big(F(2) \otimes I_2\big)\,, \tag{76}$$

where

$$Y_3 = \frac{1}{2}\begin{pmatrix} -1 & 0 \\ 0 & u^2 - u \end{pmatrix}\,, \tag{77}$$

$$Y_5 = \frac{1}{2}\begin{pmatrix} v^2+v^3 & v+v^4 & 0 & 0 \\ v+v^4 & v^2+v^3 & 0 & 0 \\ 0 & 0 & v^3-v^2 & v-v^4 \\ 0 & 0 & v-v^4 & v^2-v^3 \end{pmatrix}\,. \tag{78}$$

The entries of Y_3 and Y_5 are either real or purely imaginary implying that only real multiplications are required to compute their actions.

Set

$$H' = I_4 \oplus \big(I_4 \otimes F(2)\big)\,, \tag{79}$$

$$H = H' \oplus \big(H' \otimes F(2) \otimes I_2\big)\,, \tag{80}$$

$$Y' = C'' \oplus \big(C'' \otimes Y_3\big)\,, \tag{81}$$

then

$$Y = Y' \oplus \big(Y' \otimes Y_5\big) = C'' \oplus \big(C'' \otimes Y_3\big) \oplus \big(C'' \otimes Y_5\big) \oplus \big(C'' \otimes Y_3 \otimes Y_5\big)\,, \tag{82}$$

and matrix C can be written as

$$C = HYH\,, \tag{83}$$

and

$$F_\Pi = HYHA\,. \tag{84}$$

Table 6-4. $F_\Pi = HYHA$

Factor	R.A	R.M
H	176	0
Y	156	272
A	532	0
F_Π	1040	272

Variant 4:

We can rewrite (84) as

$$F_\Pi = HYBH\,, \tag{85}$$

where

$$B_1 = HAH^{-1}\,, \tag{86}$$

we have that

$$B_1 = \begin{pmatrix} B' & B' \otimes [1100] \\ -2B' \begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \end{pmatrix} & B' \otimes I_4 \end{pmatrix}\,, \tag{87}$$

where

$$B' = H'A'H'^{-1}\,, \tag{88}$$

is given as

$$B' = \begin{pmatrix} A'' & A'' \otimes [1\ 0] \\ -2A'' \otimes \begin{pmatrix} 1 \\ 0 \end{pmatrix} & A'' \otimes I_2 \end{pmatrix}\,. \tag{89}$$

Table 6-5. $F_\Pi = H\ Y\ B_1\ H$

Factor	R.A	R.M
H	176	0
Y	156	272
B_1	356	{88}
F_Π	864	272+{88}

7. THE CASE N=p^2

For the case of N=p^2, p an odd prime, Z/N is no longer a field, but we can still have the unit group $U(p^2)$ to be a cyclic group, and it is given as

$$U = U(p^2) = \{0 \le j < p^2 : \quad (j, p^2) = 1\} . \tag{1}$$

Define the subsets D_k of Z/p^2, $0 \le k < 3$ as

$$D_k = \{0 \le j < p^2 : \quad (j, p^2) = p^k\} , \tag{2}$$

then we have

$$D_0 = U , \tag{3}$$

$$D_1 = \{p,\ 2p,\ \ldots\ (p-1)p\} , \tag{4}$$

$$D_2 = \{0\} . \tag{5}$$

It is easy to see that D_1 has an order of (p-1) and D_2 has an order of 1, so that D_0 has an order of p(p-1). Since U is a cyclic group, we can find $z \in U$, a generator, such that

$$U = \{z^j : \quad 0 \le j < p(p-1)\} . \tag{6}$$

By Fermat's theorem, we see that

$$z^{p-1} \equiv 1 \ mod \ p . \tag{7}$$

Multiplying both sides of (7) by p, we have

$$pz^{p-1} \equiv p \ mod \ p^2 . \tag{8}$$

The elements of D_1 are all of the form pz^j. Since the order of D_1 is (p-1), each element $x \in D_1$ has a unique representation

$$x = pz^j : \quad 0 \le j < p-1 , \tag{9}$$

then we have

$$D_1 = \{pz^j : \quad 0 \le j < p-1\} . \tag{10}$$

Equations (6) and (10) define a partition of Z/p^2. Ordering the partition by the set of U-orbits

$$D_2,\ D_1,\ D_0 , \tag{11}$$

then the Fourier transform of $F(p^2)$ will have this form

$$F(p^2) = P^{-1} F_{\Pi}(p^2) P , \tag{12}$$

where P is the permutation matrix corresponding to permutation Π defined in (11), and

$$F_{\Pi}(p^2) = [\, w^{\Pi(j)\Pi(k)} \,]_{0 \le j,k < p^2}\,, \quad w = e^{2\pi i/p^2}. \tag{13}$$

Let p'=p-1, and s=p(p-1), $F_{\Pi}(p^2)$ can be written as

$$F_{\Pi}(p^2) = \begin{bmatrix} 1 & 1^t_{p'} & 1^t_s \\ 1_{p'} & E(p',p') & 1^t_p \otimes C_p \\ 1_s & 1_p \otimes C_p & C_{p^2} \end{bmatrix}, \tag{14}$$

where C_{p^2} is the s by s skew-circulant matrix with the first row as

$$w^{z^0},\ w^{z^1},\ \ldots\ w^{z^{s-1}}, \tag{15}$$

C_p is the (p-1) by (p-1) skew-circulant matrix having the first row as

$$w^{pz^0},\ w^{pz^1},\ \ldots\ w^{pz^{p-2}}, \tag{16}$$

and $E(p',p')$ is a p' by p' matrix with all elements equal 1. Observe the matrix $F_{\Pi}(p^2)$ in (14), since D_0 and D_1 are in disjoint set of Z/p^2, C_{p^2} and C_p are disjoint . We can not factorize the $F(p^2)$ to be of the form as CA as in section 3 the 4p case, where the C is block-diagonal and A is a matrix with 1, 0, and -1's. We need to modify the submatrix C_{p^2} to C'_{p^2}, such that C'_{p^2} has the information of C_p, and that C'_{p^2} must be still skew-circulant.

Let

$$F_{\Pi}(p^2) + 0(p,p) \oplus \left(\frac{1}{p}E(p,p) \otimes C_p\right) = F_c(p^2)\,, \tag{17}$$

where 0(p,p) is a p by p matrix with all elements equal 0. Then $F_c(p^2)$ is given as

$$F_c(p^2) = \begin{bmatrix} 1 & 1^t_{p'} & 1^t_s \\ 1_{p'} & E(p',p') & 1^t_p \otimes C_p \\ 1_s & 1_p \otimes C_p & C_{p^2} + \frac{1}{p}E(p,p) \otimes C_p \end{bmatrix}. \tag{18}$$

Let

$$C'_{p^2} = C_{p^2} + \frac{1}{p}E(p,p) \otimes C_p\,, \tag{19}$$

direct computation can show that

$$C_{p^2} 1_s = \underline{0}_s, \tag{20}$$

$$E(p,p) \otimes C_p 1_s = -p 1_s, \tag{21}$$

$$C_{p^2}\left(1_p \otimes I_{p-1}\right) = 0(s,s), \tag{22}$$

$$(E(p,p) \otimes C_p)(1_p \otimes I_{p-1}) = p(1_p \otimes C_p). \tag{23}$$

Now that $F_c(p^2)$ can be factorized as

$$F_c(p^2) = CA\,, \tag{24}$$

where C is a block-diagonal matrix

$$C = 1 \oplus C_p \oplus C'_{p^2}, \tag{25}$$

and A is the pre-addition matrix given as

$$A = \begin{bmatrix} 1 & 1^t_{p'} & 1^t_s \\ -1_{p'} & -E(p',p') & 1^t_p \otimes I_{p'} \\ -1_s & 1_p \otimes I_{p'} & I_s \end{bmatrix}. \tag{26}$$

Go back to the computation of $F_\Pi(p^2)$, since $F_\Pi(p^2)$ has two parts

$$F_\Pi(p^2) = F_c(p^2) - 0(p,p) \oplus (\frac{1}{p}E(p,p) \otimes C_p)\,, \tag{27}$$

the second part is very easy to compute, since E(p,p) is p by p matrix with all elements equal 1.

Let

$$\underline{b'} = P\underline{b}\,, \tag{28}$$

$$\underline{a'} = P\underline{a}\,, \tag{29}$$

then

$$\underline{b'} = F_\Pi(p^2)\underline{a'} = F_c(p^2)\underline{a'} - 0(p,p) \oplus (\frac{1}{p}E(p,p) \otimes C_p)\underline{a'}\,. \tag{30}$$

Let $\underline{a'}_{p(p-1)}$ be a vector of the p(p-1) elements of $\underline{a'}$ corresponding to the indexing set of U,

$$\underline{a'}_{p(p-1)} = [\underline{a}^t(0),\ \underline{a}^t(1),\ \ldots\ \underline{a}^t(p-1)]\,, \tag{31}$$

where each $\underline{a}^t(i)$ has (p-1) elements. Let

$$\underline{a'_p} = \sum_{i=0}^{p-1} \underline{a}^t(i), \tag{32}$$

then

$$\underline{b'_p} = \frac{1}{p}C_p\underline{a'_p}. \tag{33}$$

The whole computation will be

$$\underline{b'} = CA\underline{a'} - \underline{b_2}\,, \tag{34}$$

where

$$\underline{b_2} = \begin{bmatrix} \underline{0_p} \\ \underline{b'_p} \\ \vdots \\ \underline{b'_p} \end{bmatrix}. \tag{35}$$

Direct computation of (34) will give the arithmetic count as:

Table 7 – 1: $F_{\Pi}(p^2)$

Factor	R.A	R.M
C	$2(2p^4-4p^3+3p^2-4p+3)$	$4(p-1)^2(p^2+1)$
A	$2(p^2+5p-6)$	0
$F_c(p^2)$	$2(2p^4-4p^3+4p^2+p-3)$	$4(p-1)^2(p^2+1)$
$F_{\Pi}(p^2)$	$2(2p^4-4p^3+7p^2-5p)$	$4(p-1)^2(p^2+1)$

Variant 1.

Matrix C can be further diagonalized as

$$C = FDF, \tag{36}$$

where

$$F = 1 \oplus F(p') \oplus F(s)\,, \tag{37}$$

and D is a diagonal matrix

$$D = 1 \oplus D_p \oplus D_s\,, \tag{38}$$

where

$$D_p = F(p')^{-1} C_p F(p')^{-1}\,, \tag{39}$$

and

$$D_s = F(s)^{-1} C'_{p^2} F(s)^{-1}\,. \tag{40}$$

Diagonalize $\frac{1}{p}C_p$ by F(p-1), we have

$$\underline{b'_p} = F(p-1) D'_p F(p-1) \underline{a'_p}\,, \tag{41}$$

where D'_p is a (p-1) by (p-1) diagonal matrix, and

$$D'_p = \frac{1}{p} D_p\,.$$

Equation (24) now can be written as

$$F_c(p^2) = FDFA\,. \tag{42}$$

Table 7 – 2: Variant 1. $F_{\Pi}(p^2)$

Factor	R.A	R.M
F	$A_p + A_s$	$M_p + M_s$
D	$2(p^2-5)$	$4(p^2-3)$
A	$2(p^2+5p-6)$	0
$F_c(p^2)$	$2(2p^2+5p-11+A_p+A_s)$	$2(M_p+M_s)+4(p^2-3)$
$F_{\Pi}(p^2)$	$2(3p^2+5p-14+2A_p+A_s)$	$2(2M_p+M_s)+4(p^2+p-5)$

Variant 2.

The cost of additions in (42) can be reduced by interchanging the order of the computations. Let

$$F_c(p^2) = FDBF\,, \tag{43}$$

where

$$B = FAF^{-1}\,, \tag{44}$$

B can be explicitly written as

$$B = \begin{bmatrix} 1 & e^t_{p'} & e^t_s \\ -p'e_{p'} & -p'e(p',p') & I_{p'} \otimes e^t_p \\ -se_s & pI_{p'} \otimes e_p & I_s \end{bmatrix}\,, \tag{45}$$

where e_n is a vector having n elements with only the first being 1 , all others being 0, and

$$e(n,n) = e_n \otimes e^t_n\,. \tag{46}$$

It is clear that computation (43) has fewer additions than that of (42), but we introduced some multiplications by rational number.

Matrix B has only 2(p+3) real additions and 6 real number multiplied by integer.

Variant 3.

To reduce the cost of additions required to perform the complex multiplications coming from the action of C, we note that each skew-circulant matrix has the form

$$C_p = \begin{bmatrix} X_p & X^*_p \\ X^*_p & X_p \end{bmatrix}\,, \tag{47}$$

and

$$C'_{p^2} = \begin{bmatrix} X_s & X^*_s \\ X^*_s & X_s \end{bmatrix}\,. \tag{48}$$

By the action of

$$C_p = \big(F(2) \otimes I_{p'/2}\big) Y_p \big(F(2) \otimes I_{p'/2}\big)\,, \tag{49}$$

where

$$Y_p = \frac{1}{2}\begin{bmatrix} X_p + X^*_p & 0 \\ 0 & X_p - X^*_p \end{bmatrix}\,, \tag{50}$$

and

$$C'_{p^2} = \big(F(2) \otimes I_{s/2}\big) Y_s \big(F(2) \otimes I_{s/2}\big)\,, \tag{51}$$

where

$$Y_s = \frac{1}{2}\begin{bmatrix} X_s + X^*_s & 0 \\ 0 & X_s - X^*_s \end{bmatrix}\,. \tag{52}$$

The important thing is that both Y_p and Y_s have only real entries or purely imaginary entries.

Set

$$Y = 1 \oplus Y_p \oplus Y_s \,, \tag{53}$$

and

$$H = 1 \oplus \left(F(2) \otimes I_{p'/2}\right) \oplus \left(F(2) \otimes I_{s/2}\right) \,, \tag{54}$$

now we can write

$$C = HYH \,, \tag{55}$$

and

$$F_c(p^2) = HYHA \,. \tag{56}$$

Table 7 – 3 : Variant 3. C

Factor	R.A	R.M
H	$2(p^2-1)$	0
Y	p^4-2p^3-2p+3	$(p-1)^2(p^2+1)$
C	$p^4-2p^3+4p^2-2p-1$	$(p-1)^2(p^2+1)$

8. AN EXAMPLE OF N=p^2: N=9

In this section, let us derive in detail an example of $N = p^2$, N=9, to see a complete picture of the algorithms.

The cyclic group of Z/9 is

$$U = \{1,\ 2,\ 4,\ 5,\ 7,\ 8\}\ . \tag{1}$$

Reordering U by the generator z=2, we have

$$D_0 = U = \{1,\ 2,\ 4,\ 8,\ 7,\ 5\}\ . \tag{2}$$

The other two U-orbits D_1 and D_2 will be

$$D_1 = \{3,\ 6\}, \tag{3}$$

$$D_2 = \{0\}. \tag{4}$$

The permutation Π is defined as

$$\Pi = \begin{bmatrix} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ 0 & 3 & 6 & 1 & 2 & 4 & 8 & 7 & 5 \end{bmatrix}\ . \tag{5}$$

Direct computation shows that the matrix $F_\Pi(9)$ as

$$F_\Pi(9) = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & v & v^2 & v & v^2 & v & v^2 \\ 1 & 1 & 1 & v^2 & v & v^2 & v & v^2 & v \\ 1 & v & v^2 & w & w^2 & w^4 & w^8 & w^7 & w^5 \\ 1 & v^2 & v & w^2 & w^4 & w^8 & w^7 & w^5 & w \\ 1 & v & v^2 & w^4 & w^8 & w^7 & w^5 & w & w^2 \\ 1 & v^2 & v & w^8 & w^7 & w^5 & w & w^2 & w^4 \\ 1 & v & v^2 & w^7 & w^5 & w & w^2 & w^4 & w^8 \\ 1 & v^2 & v & w^5 & w & w^2 & w^4 & w^8 & w^7 \end{bmatrix},\ w = e^{2\pi i/9},\ and\ v = e^{2\pi i/3}\ , \tag{6}$$

which can be written as

$$F_\Pi(9) = \begin{bmatrix} 1 & 1_2^t & 1_6^t \\ 1_2 & E(2,2) & 1_3^t \otimes C_3 \\ 1_6 & 1_3 \otimes C_3 & C_9 \end{bmatrix}\ . \tag{7}$$

Modifying $F_\Pi(9)$ as

$$F_\Pi(9) + 0(3,3) \oplus (\frac{1}{3}E(3,3) \otimes C_3) = F_c(9)\ , \tag{8}$$

then $F_c(9)$ is given as

$$F_c(9) = \begin{bmatrix} 1 & 1_2^t & 1_6^t \\ 1_2 & E(2,2) & 1_3^t \otimes C_3 \\ 1_6 & 1_3 \otimes C_3 & C_9' \end{bmatrix}\ , \tag{9}$$

where

$$C_9' = C_9 + \frac{1}{3}E(3,3) \otimes C_3\ . \tag{10}$$

The properties we use in factoring $F_c(9)$ are

$$C_9 1_6 = \underline{0}_6, \tag{11}$$

$$E(3,3) \otimes C_3 1_6 = -3\ 1_6, \tag{12}$$

$$C_9(1_3 \otimes I_2) = 0(6,6), \tag{13}$$

$$(E(3,3) \otimes C_3)(1_3 \otimes I_2) = 3(1_3 \otimes C_3). \tag{14}$$

$F_c(9)$ can be factorized as

$$F_c(9) = CA\,, \tag{15}$$

where C is a block-diagonal matrix

$$C = 1 \oplus C_3 \oplus C_9', \tag{16}$$

and A is the pre-addition matrix given as

$$A = \begin{bmatrix} 1 & 1_2^t & 1_6^t \\ -1_2 & -E(2,2) & 1_3^t \otimes I_2 \\ -1_6 & 1_3 \otimes I_2 & I_6 \end{bmatrix}. \tag{17}$$

Go back to the computation of $F_\Pi(9)$, since $F_\Pi(9)$ has two parts

$$F_\Pi(9) = F_c(9) - 0(3,3) \oplus (\frac{1}{3}E(3,3) \otimes C_3)\,, \tag{18}$$

the second part is computed as follows:

let

$$\underline{a_3'} = \sum_{i=0}^{2} \underline{a^t}(i) = \begin{bmatrix} a_1 + a_4 + a_7 \\ a_2 + a_5 + a_8 \end{bmatrix}, \tag{19}$$

then

$$\underline{b_3'} = \frac{1}{3} C_3 \underline{a_3'}. \tag{20}$$

Diagonalizing $\frac{1}{3}C_3$ by F(2), we have

$$\underline{b_3'} = F(2) D_3 F(2) \underline{a_3'} = F(2) \begin{bmatrix} \frac{v+v^2}{6} & 0 \\ 0 & \frac{v-v^2}{6} \end{bmatrix} F(2) \begin{bmatrix} a_1 + a_4 + a_7 \\ a_2 + a_5 + a_8 \end{bmatrix}. \tag{21}$$

The whole computation will be

$$\underline{b'} = CA\underline{a'} - \begin{bmatrix} \underline{0_3} \\ \underline{b_3'} \\ \underline{b_3'} \\ \underline{b_3'} \end{bmatrix}. \tag{22}$$

Table 8 – 1 : $F_{\Pi}(9)$

Factor	R.A	R.M
C	144	160
A	36	0
$F_c(9)$	180	160
$F_{\Pi}(9)$	204	176

Variant 1.

Matrix C can be further diagonalized as

$$C = FDF, \tag{23}$$

where

$$F = 1 \oplus F(2) \oplus F(6)\ , \tag{24}$$

and D is a diagonal matrix

$$D = 1 \oplus D_3 \oplus D_6\ , \tag{25}$$

where

$$D_3 = F(2)^{-1} C_3 F(2)^{-1}\ , \tag{26}$$

and

$$D_6 = F(6)^{-1} C_9' F(6)^{-1}\ . \tag{27}$$

Equation (15) now can be written as

$$F_c(9) = FDFA\ . \tag{28}$$

Table 8 – 2 : Variant 1. $F_{\Pi}(9)$

Factor	R.A	R.M
F	40	8
D	8	24
A	36	0
$F_c(9)$	124	40
$F_{\Pi}(9)$	144	44

Variant 2.

The cost of additions in (28) can be reduced by interchanging the order of the computations. Let

$$F_c(9) = FDBF\,, \tag{29}$$

where

$$B = FAF^{-1}\,. \tag{30}$$

B can be explicitly written as

$$B = \begin{bmatrix} 1 & e_2^t & e_6^t \\ -2e_2 & -2e(2,2) & I_2 \otimes e_3^t \\ -6e_6 & 3I_2 \otimes e_3 & I_6 \end{bmatrix}\,. \tag{31}$$

It is clear that computation (29) has fewer additions than that of (28), but we introduced some multiplications by rational numbers.

Table 8 – 3 : Variant 2. $F_{II}(9)$

Factor	R.A	R.M
F	40	8
D	8	24
B	12	{6}
$F_c(9)$	100	40+{6}
$F_{II}(9)$	120	44+{6}

Variant 3.

To reduce the cost of additions required to perform the complex multiplications coming from the action of C, we note that each skew-circulant matrix has the form

$$C_3 = \begin{bmatrix} v & v^* \\ v^* & v \end{bmatrix}\,, \tag{32}$$

and

$$C_9' = \begin{bmatrix} X_6 & X_6^* \\ X_6^* & X_6 \end{bmatrix}\,, \tag{33}$$

where

$$X_6 = \begin{bmatrix} w & w^2 & w^4 \\ w^2 & w^4 & w^8 \\ w^4 & w^8 & w^7 \end{bmatrix}\,. \tag{34}$$

By the action of

$$C_3 = F(2)Y_2F(2)\,, \tag{35}$$

where

$$Y_3 = \frac{1}{2}\begin{bmatrix} v+v^2 & 0 \\ 0 & v-v^2 \end{bmatrix}, \tag{36}$$

and

$$C_9' = (F(2)\otimes I_3)Y_6(F(2)\otimes I_3)\,, \tag{37}$$

where

$$Y_6 = \frac{1}{2}\begin{bmatrix} X_6+X_6^* & 0 \\ 0 & X_6-X_6^* \end{bmatrix}. \tag{38}$$

It is easy to see that both Y_2 and Y_6 have only real entries or purely imaginary entries.

Set

$$Y = 1\oplus Y_3\oplus Y_6\,, \tag{39}$$

and

$$H = 1\oplus F(2)\oplus (F(2)\otimes I_3)\,, \tag{40}$$

Now we can write

$$C = HYH\,, \tag{41}$$

and

$$F_c(9) = HYHA\,. \tag{42}$$

Table 8 – 4: Variant 3. $F_\Pi(9)$

Factor	R.A	R.M
H	16	0
Y	24	36
A	36	0
$F_c(9)$	92	36
$F_\Pi(9)$	112	40

Variant 4.

By the same method as in Variant 2, we can have from (42)

$$F_c(9) = HYB_1H\,, \tag{43}$$

where

$$B_1 = HAH^{-1}\,. \tag{44}$$

Direct computation can show that

$$B_1 = \begin{bmatrix} 1 & f_2^t & f_6^t \\ -2f_2 & -2f(2,2) & E_6^t \\ -2f_6 & E_6 & I_6 \end{bmatrix}, \tag{45}$$

where

$$E_6^t = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & -1 & 1 \end{bmatrix}. \tag{46}$$

It is clear from (45) that computing the action of B_1 requires fewer additions than that of computing the action of A.

Table 8 – 5 : Variant 4. $F_{\Pi}(9)$

Factor	R.A	R.M
H	16	0
Y	24	36
B_1	24	{4}
$F_c(9)$	80	36+{4}
$F_{\Pi}(9)$	100	40+{4}

9. THE CASE N=p^2q

For the case of N=p^2q, where p and q are distinct odd primes, we have that p^2 and q are relatively primes. The Chinese Remainder Theorem asserts the existance of a ring-isomorphism σ, such that

$$Z/p^2 \times Z/q \cong Z/N, \tag{1}$$

which restricts to a group-isomorphism

$$U(p^2) \times U(q) \cong U(N), \tag{2}$$

U(N) is the direct product of the cyclic groups of $U(p^2)$ and U(q). They were defined in the previous sections.

A class of algorithms computing the $N = p^2q$ point Fourier transform will be derived from the ring-structure of the indexing set Z/p^2q. The ring-structure determines a permutation of input and output data.

From the Chinese Remainder Theorem, we have a set

$$\{e_1, e_2\}, \qquad e_1, e_2 \in Z/p^2q, \tag{3}$$

which is called a system of idempotents for the ring Z/p^2q, since it satisfies the following conditions:

$$e_1 \equiv 1 \bmod p^2, \quad e_1 \equiv 0 \bmod q, \tag{4}$$

$$e_2 \equiv 0 \bmod p^2, \quad e_2 \equiv 1 \bmod q, \tag{5}$$

and

$$1 \equiv e_1 + e_2 \bmod p^2q, \tag{6}$$

$$0 \equiv e_1e_2 \bmod p^2q, \tag{7}$$

$$e_j^2 \equiv e_j \bmod p^2q, \quad j = 1,2. \tag{8}$$

Define a mapping σ:

$$\sigma(a,b) = e_1a + e_2b, \quad a \in Z/p^2, \ b \in Z/q \tag{9}$$

Fix the generators z_1 and z_2 of the cyclic groups $U(p^2)$ and U(q) respectively, let

$$y_1 = \sigma(z_1, 1) = e_1z_1 + e_2 \mod p^2q, \tag{10}$$

$$y_2 = \sigma(1, z_2) = e_1 + e_2z_2 \mod p^2q, \tag{11}$$

and $G(y_1)$ and $G(y_2)$ denote the cyclic groups generated by y_1 and y_2, then we have

$$U(N) = G(y_1) \times G(y_2). \tag{12}$$

This induces the order on U(N) defined by the listing

$$y_2^j y_1^k : \quad 0 \le j < (q-1), \ \ and \ \ 0 \le k < p(p-1) , \tag{13}$$

running first over j and then over k, and we call it the exponential order of U(N) associated with y_1 and y_2. Let

$$U = U(N), \quad s = p(p-1), \quad p' = p-1, \quad and \, q' = q-1 , \tag{14}$$

and set

$$f_1 = pe_1 \ mod \, p^2 q , \tag{15}$$

$$f_2 = pe_1 + e_2 \ mod \, p^2 q , \tag{16}$$

observe that

$$f_1^2 = 0 \ mod \, p^2 q, \quad f_2^2 = e_2 \ mod \, p^2 q , \tag{17}$$

$$e_1 f_1 = f_1 \ mod \, p^2 q, \quad e_2 f_2 = e_2 \ mod \, p^2 q, \tag{18}$$

$$e_2 f_1 = 0 \ mod \, p^2 q, \quad e_1 f_2 = f_1 \ mod \, p^2 q . \tag{19}$$

From the formulas, we see that

$$0U = \{0\} , \tag{20}$$

$$e_1 U = \{e_1 z_1^j : \quad 0 \le j < s\}, \tag{21}$$

$$e_2 U = \{e_2 z_2^j : \quad 0 \le j < q'\}, \tag{22}$$

$$f_1 U = \{f_1 z_1^j : \quad 0 \le j < p'\}, \tag{23}$$

$$f_2 U = \{f_1 z_1^k + e_2 z_2^j : 0 \le j < q' \ \ and \, 0 \le k < p'\} . \tag{24}$$

The set of U-orbits corresponding to

$$0, \ e_2, \ f_1, \ f_2, \ e_1, \ 1 \tag{25}$$

defines a partition of Z/p^2q. Order the partition by (25), and denote the corresponding permutation by Π. We will describe

$$F_\Pi = PF(p^2 q)P^{-1} , \tag{26}$$

where P is the permutation matrix corresponding to Π.

If α and β are in Z/p^2q, denote by

$$C(\alpha, \beta) , \tag{27}$$

the submatrix of F_Π corresponding to $\alpha U \times \beta U$. Then, since $e_1^2 = e_1$, by (21), we have

$$C(e_1, e_1) = \left[\, (w^{e_1})^{z_1^{j+k}} \, \right]_{0 \le j,k < s} , \quad w = e^{2\pi i / p^2 q} . \tag{28}$$

By (4),

$$v = w^{e_1}$$

is a primitive p^2-th root of unity. Set

$$C_{p^2} = C(e_1, e_1), \tag{29}$$

which is the rotated version of C_{p^2} given in section 7. In the same way, since $e_2^2 \equiv e_2$, by (22), we have

$$C(e_2, e_2) = \left[(w^{e_2})^{x_2^{j+k}} \right]_{0 \le j,k < q'} . \tag{30}$$

By (5),

$$u = w^{e_2} \tag{31}$$

is a primitive q-th root of unity. Set

$$C_q \equiv C(e_2, e_2) , \tag{32}$$

which we recognized as a rotated Winograd core.

By (10), (11) and (13), we have

$$C(1,1) = \left[w^{y_1^{k+l} y_2^{j+s}} \right] = \left[(w^{e_1})^{x_1^{k+l}} (w^{e_2})^{x_2^{j+s}} \right] = C_{p^2} \otimes C_q . \tag{33}$$

Continuing in this way, the matrix F_Π is given by the following table.

Table 9 − 1: $F_\Pi(p^2q)$

	0	e_2	f_1	f_2	e_1	1
0	1	$1^t_{q'}$	$1^t_{p'}$	$1^t_{p'q'}$	1^t_s	$1^t_{sq'}$
e_2	$1_{q'}$	C_q	$E(q', p')$	$1^t_{p'} \otimes C_q$	$E(q', s)$	$1^t_s \otimes C_q$
f_1	$1_{p'}$	$E(p', q')$	$E(p', p')$	$E(p', p'q')$	$1^t_p \otimes C_p$	$1^t_p \otimes C_p \otimes 1^t_{q'}$
f_2	$1_{p'q'}$	$1_{p'} \otimes C_q$	$E(p'q', p')$	$E(p', p') \otimes C_q$	$1^t_p \otimes C_p \otimes 1_{q'}$	$1^t_p \otimes C_p \otimes C_q$
e_1	1_s	$E(s, q')$	$1_p \otimes C_p$	$1_p \otimes C_p \otimes 1^t_{q'}$	C_{p^2}	$C_{p^2} \otimes 1^t_{q'}$
1	$1_{sq'}$	$1_s \otimes C_q$	$1_p \otimes C_p \otimes 1_{q'}$	$1_p \otimes C_p \otimes C_q$	$C_{p^2} \otimes 1_{q'}$	$C_{p^2} \otimes C_q$

where C_p is (p-1) by (p-1) skew-circulant matrix with elements of p-th root of unity, a rotated p-point Winograd core.

The same argument as in section 7, the $N = p^2$ case, modification got to be taken in order to factorize the matrix $F_\Pi(p^2q)$ into two matrices production CA, where C is a block-diagonal matrix with each submatrix skew-circulant, and A is a pre-addition matrix with only 1, 0, -1's.

Let

$$C'_{p^2} = C_{p^2} + \frac{1}{p} E(p,p) \otimes C_p \,, \tag{34}$$

then

$$F_\Pi(p^2 q) = F_c(p^2 q) - 0(pq, pq) \oplus \frac{1}{p} \begin{bmatrix} E(p,p) \otimes C_p & E(p,p) \otimes C_p \otimes 1^t_{q'} \\ E(p,p) \otimes C_p \otimes 1_{q'} & E(p,p) \otimes C_p \otimes C_q \end{bmatrix} , \tag{35}$$

where $F_c(p^2 q)$ has the same structure as $F_\Pi(p^2 q)$ in Table 9-1 except that C_{p^2} is substituted by C'_{p^2}.

Now that $F_c(p^2 q)$ can be factorised as

$$F_c(p^2 q) = CA \,, \tag{36}$$

Where C is given as

$$1 \oplus C_q \oplus C_p \oplus (C_p \otimes C_q) \oplus C'_{p^2} \oplus (C'_{p^2} \otimes C_q) \,, \tag{37}$$

and A is given as

Table 9 − 2 : A

	0	e_2	f_1	f_2	e_1	1
0	1	$1^t_{q'}$	$1^t_{p'}$	$1^t_{p'q'}$	1^t_s	$1^t_{sq'}$
e_2	$-1_{q'}$	$I_{q'}$	$-E(q',p')$	$1^t_{p'} \otimes I_{q'}$	$-E(q',s)$	$1^t_s \otimes I_{q'}$
f_1	$-1_{p'}$	$-E(p',q')$	$-E(p',p')$	$-E(p',p'q')$	$1^t_p \otimes I_{p'}$	$1^t_p \otimes I_{p'} \otimes 1^t_{q'}$
f_2	$1_{p'q'}$	$-1_{p'} \otimes I_{q'}$	$E(p'q',p')$	$-E(p',p') \otimes I_{q'}$	$-1^t_p \otimes I_{p'} \otimes 1_{q'}$	$1^t_p \otimes I_{p'} \otimes I_{q'}$
e_1	-1_s	$-E(s,q')$	$1_p \otimes I_{p'}$	$1_p \otimes I_{p'} \otimes 1^t_{q'}$	I_s	$I_s \otimes 1^t_{q'}$
1	$1_{sq'}$	$-1_s \otimes I_{q'}$	$-1_p \otimes I_{p'} \otimes 1_{q'}$	$1_p \otimes I_{p'} \otimes I_{q'}$	$-I_s \otimes 1_{q'}$	$I_{sq'}$

By a permutation Q, A can be written as

$$A' = Q^{-1} A Q = \begin{bmatrix} A(p^2) & A(p^2) \otimes 1^t_{q'} \\ -A(p^2) \otimes 1_{q'} & A(p^2) \otimes I_{q'} \end{bmatrix} , \tag{38}$$

where $A(p^2)$ is given as

$$A(p^2) = \begin{bmatrix} 1 & 1^t_{p'} & 1^t_s \\ -1_{p'} & -E(p',p') & 1^t_p \otimes I_{p'} \\ -1_s & 1_p \otimes I_{p'} & I_s \end{bmatrix} , \tag{39}$$

which was given in section 7. The permutation matrix Q is easy to define . Actually, if we order the input and output indexing set by listing this way

$$0\ f_1,\ e_1,\ e_2,\ f_2,\ 1 \,, \tag{40}$$

we will get the form of A'. The reason we take the ordering as (25) is that we want to put C_{p^2} and $C_{p^2} \otimes C_q$ close. The advantage can be seen from equation (35).

If we take the order listed in (40), the corresponding Fourier transform matrix will be

$$F_c'(p^2q) = Q^{-1}F_c(p^2q)Q = C'A', \tag{41}$$

where

$$C' = 1 \oplus C_p \oplus C'_{p^2} \oplus C_q \oplus (C_p \otimes C_q) \oplus (C'_{p^2} \otimes C_q) . \tag{42}$$

Set

$$C(p^2) = 1 \oplus C_p \oplus C'_{p^2} , \tag{43}$$

then equation (42) can be written as

$$C' = C(p^2) \oplus (C(p^2) \otimes C_q) . \tag{44}$$

Arithmetically, A is equivalent to

$$A(p^2) \otimes A(q) , \tag{45}$$

where A(q) is given as

$$A(q) = \begin{bmatrix} 1 & 1_{q'}^t \\ -1_{q'} & I_{q'} \end{bmatrix} . \tag{46}$$

Let us go back to the computation of $F_\Pi(p^2q)$. Observe the second part of equation (35), and set

$$F_s = \frac{1}{p} \begin{bmatrix} E(p,p) \otimes C_p & E(p,p) \otimes C_p \otimes 1_{q'}^t \\ E(p,p) \otimes C_p \otimes 1_{q'} & E(p,p) \otimes C_p \otimes C_q \end{bmatrix} . \tag{47}$$

A permutation matrix P_1 can be easily found such that

$$F_s = P_1^{-1}(E(p,p) \otimes F_r)P_1 , \tag{48}$$

where F_r is given as

$$F_r = \frac{1}{p} \begin{bmatrix} C_p & C_p \otimes 1_{q'}^t \\ C_p \otimes 1_{q'} & C_p \otimes C_q \end{bmatrix} . \tag{49}$$

It is easy to see that F_r can be written as

$$F_r = C_r A_r , \tag{50}$$

where C_r is a block-diagonal matrix given as

$$C_r = \frac{1}{p} \begin{bmatrix} C_p & 0 \\ 0 & C_p \otimes C_q \end{bmatrix} , \tag{51}$$

and A_r is pre-addition matrix given as

$$A_r = \begin{bmatrix} I_{p'} & I_{p'} \otimes 1_{q'}^t \\ I_{p'} \otimes 1_{q'} & I_{p'} \otimes I_{q'} \end{bmatrix} . \tag{52}$$

The permutation matrix P_1 is defined as

$$P_1 = I_s \oplus P(sq', p'q') \,, \tag{53}$$

where P(n,m) is defined in reference [10]. Consider the action of F_s on an input data array $\underline{x}$,

$$\underline{y} = F_s \underline{x} \,, \tag{54}$$

let

$$\underline{y}' = P_1 \underline{y} \,, \tag{55}$$

and

$$\underline{x}' = P_1 \underline{x} \,. \tag{56}$$

Since E(p,p) is a p by p matrix with all elements equal 1, so that the computation of F_r repeats p times. Let us first cut $\underline{x}'$ into p pieces, each one has q(p-1) elements, that is

$$(\underline{x}')^t = (\underline{x}(0)^t, \underline{x}(1)^t, \ \ldots \ \underline{x}(p-1)^t) \,, \tag{57}$$

let

$$\underline{x}_r = \sum_{i=0}^{p-1} \underline{x}(i) \,, \tag{58}$$

we have

$$\underline{y}_r = F_r \underline{x}_r = C_r A_r \underline{x}_r \,, \tag{59}$$

then

$$\underline{y}' = \begin{bmatrix} \underline{y}_r \\ \underline{y}_r \\ \vdots \\ \underline{y}_r \end{bmatrix} \,. \tag{60}$$

The whole computation of $F_\Pi(p^2q)$ will be

$$\underline{b}' = F_c(p^2q)\underline{a}' - \underline{b}_2 = CA\underline{a}' - \underline{b}_2 \,, \tag{61}$$

where

$$\underline{b}_2 = \begin{bmatrix} \underline{0}_{pq} \\ \underline{y} \end{bmatrix} \,, \tag{62}$$

and 0_{pq} is a vector with pq elements all equal 0.

Direct computation of (61) will give the arithmetic count:

for the block-diagonal matrix C:

$$R.A = 2(2p^4 - 4p^3 + 3p^2 - 4p + 3)q + 2p^2(q-1)(2q-3) \,, \tag{63}$$

$$R.M = 4(p-1)^2(p^2+1)q + 4(q-1)^2p^2 \,, \tag{64}$$

and for the pre-addition matrix A:

$$R.A = 2(p^2 + 5p - 6)q + 4p^2(q-1)\,, \tag{65}$$

$$R.M = 0\,. \tag{66}$$

For the second part of (61), some of the additions have been done in computing the action of A, the additional arithmetic count of $\underline{b}_2$ is:

$$R.A = 2(p-1)[(2p-3)q + (q-1)(2p-1)]\,, \tag{67}$$

$$R.M = 4(p-1)^2 q + 4(p-1)(q-1)^2\,. \tag{68}$$

Variant 1.

Matrix C' and C_r are block-diagonal with each submatrix being skew-circulant, so that C' and C_r can be further diagonalized by corresponding size of Fourier transform. Define F as

$$F = 1 \oplus F(p') \oplus F(s) \oplus F(q') \oplus \big(F(p') \otimes F(q')\big) \oplus \big(F(s) \otimes F(q')\big)\,. \tag{69}$$

Set

$$F_{p^2} = 1 \oplus F(p') \oplus F(s)\,, \tag{70}$$

(69) can be written as

$$F = F_{p^2} \oplus \big(F_{p^2} \otimes F(q')\big). \tag{71}$$

Then

$$C' = FDF\,, \tag{72}$$

where D is a diagonal matrix given as

$$D = F^{-1}CF^{-1}\,, \tag{73}$$

D can be written explicitly as

$$D = 1 \oplus D_p \oplus D_s \oplus D_q \oplus \big(D_p \otimes D_q\big) \oplus \big(D_s \otimes D_q\big)\,, \tag{74}$$

where

$$D_p = F(p')^{-1} C_p F(p')^{-1}\,, \tag{75}$$

$$D_q = F(q')^{-1} C_q F(q')^{-1}\,, \tag{76}$$

$$D_s = F(s)^{-1} C'_{p^2} F(s)^{-1}\,. \tag{77}$$

D can also be written as

$$D = D(p^2) \oplus \big(D(p^2) \otimes D_q\big)\,, \tag{78}$$

where

$$D(p^2) = F_{p^2}^{-1} C(p^2) F_{p^2}^{-1} . \tag{79}$$

Computation of (41) becomes

$$F_c'(p^2 q) = FDFA' . \tag{80}$$

Diagonalizing the matrix C_r by

$$F_0 = F(p') \oplus \left(F(p') \otimes F(q')\right) , \tag{81}$$

then

$$C_r = F_0 D_r F_0 , \tag{82}$$

where D_r is a diagonal matrix given as

$$D_r = F_0^{-1} C_r F_0^{-1} . \tag{83}$$

Then the computation of F_r becomes

$$F_r = F_0 D_r F_0 A_r . \tag{84}$$

Table 9 – 3 : F_r

Factor	R.A	R.M
F_0	$A_p q + A_q(p-1)$	$M_p q + M_q(p-1)$
D_r	2(pq-q-4)	4(pq-q-2)
A_r	4(p-1)(q-1)	0
F_r	$2(A_p q + (p-1)(A_q + 3q) - 2p - 2)$	$2(M_p q + (p-1)(M_q + 2q) - 4)$

Table 9 – 4 : Variant 1. F and D

Factor	R.A	R.M
F	$A_p q + A_q(p+s) + A_s q$	$M_p q + M_q(p+s) + M_s q$
D	$2(p^2 q - 9)$	$4(p^2 q - 4)$

where A_p , A_q and A_s are the number of real additions required to compute $F(p')$, $F(q')$ and F(s) respectively, and M_p , M_q and M_s are the number of real multiplications required to compute $F(p')$, $F(q')$ and F(s) respectively.

Variant 2.

The cost of additions in (80) can be reduced by interchanging the oder of of computations. Set

$$F_c'(p^2q) = FDBF\,, \tag{85}$$

where

$$B = FA'F^{-1}\,, \tag{86}$$

B can be explicitly written as

$$B = \begin{bmatrix} B(p^2) & B(p^2)\otimes e_q^t \\ -q'B(p^2)\otimes e_q & B(p^2)\otimes I_{q'} \end{bmatrix}\,, \tag{87}$$

where $B(p^2)$ was defined in section 7 equation (45). Arithmetically, B is equivalent to

$$B = B(p^2)\otimes B(q)\,, \tag{88}$$

where

$$B(q) = \begin{bmatrix} 1 & e_q^t \\ -q'e_q & I_{q'} \end{bmatrix}\,. \tag{89}$$

The arithmetic count of B is

$$R.A = 4p^2 + 12q\,, \tag{90}$$

$$R.M = \{2p^2 + 8q\}\,. \tag{91}$$

(91) is the number of multiplications of real multiplied by integer.

Variant 3.

To reduce the cost of additions required to perform the complex multiplications coming from the action of C, we note that each skew-circulant submatrix has the form

$$C_n = \begin{bmatrix} X_n & X_n^* \\ X_n^* & X_n \end{bmatrix}\,. \tag{92}$$

By the action of

$$C_n = \big(F(2)\otimes I_{n/2}\big)Y_n\big(F(2)\otimes I_{n/2}\big)\,, \tag{93}$$

where

$$Y_n = \frac{1}{2}\begin{bmatrix} X_n + X_n^* & 0 \\ 0 & X_n - X_n^* \end{bmatrix}\,. \tag{94}$$

Set

$$Y = Y(p^2)\oplus\big(Y(p^2)\otimes Y_q\big)\,, \tag{95}$$

and

$$H = H(p^2)\oplus\big(H(p^2)\otimes F(2)\otimes I_{q'/2}\big)\,, \tag{96}$$

where

$$Y(p^2) = 1 \oplus Y_p \oplus Y_s \,, \tag{97}$$

and

$$H(p^2) = 1 \oplus \left(F(2) \otimes I_{p'/2}\right) \oplus \left(F(2) \otimes I_{s/2}\right) . \tag{98}$$

The important thing is that Y has only real or purely imaginary entries. Now we can write

$$C' = HYH \,, \tag{99}$$

and

$$F_c'(p^2 q) = HYHA' \,. \tag{100}$$

The arithmetic count for H and Y are

H:

$$R.A = 2(p^2 - 1)q + 2p^2(q - 1) \,, \tag{101}$$

$$R.M = 0 \,, \tag{102}$$

and Y:

$$R.A = (p^4 - 2p^3 - 2p + 3)q + p^2(q^2 - 4q + 3) \,, \tag{103}$$

$$R.M = p^2\left((p - 1)^2 q + (q - 1)^2\right) . \tag{104}$$

10. AN EXAMPLE OF N=p^2q: N=45

In this section, we derive in detail an example of $N = p^2q$, N=45, to see a complete picture of the algorithms of $N = p^2q$.

The system of idempotents for Z/45 is

$$\{e_1, e_2\} = \{10, 36\} \quad . \tag{1}$$

It is easy to see that the following conditions are satisfied

$$10^2 \equiv 10 \ mod\ 45\ , \quad 36^2 \equiv 36 \ mod\ 45\ , \tag{2}$$

$$0 \equiv 10 \cdot 36 \ mod\ 45\ , \tag{3}$$

$$1 \equiv 10 + 36 \ mod\ 45\ , \tag{4}$$

Table 10-1.

$Z/3^2 \times Z/5$	Z/45
0,0	0
0,1	36
0,2	27
0,3	18
0,4	9
1,0	10
1,1	1
1,2	37
1,3	28
1,4	19
2,0	20
2,1	11
2,2	2
2,3	38
2,4	29
⋮	⋮
8,0	35
8,1	26
8,2	17
8,3	8
8,4	44

and

$$10 \equiv 1 \ mod\ 3^2\ , \quad 10 \equiv 0 \ mod\ 5\ , \tag{5}$$

$$36 \equiv 0 \ mod\ 3^2\ , \quad 36 \equiv 1 \ mod\ 5\ . \tag{6}$$

The complete system of idempotents (1) determines a ring-isomorphism σ

$$Z/3^2 \times Z/5 \cong Z/45\,, \tag{7}$$

by the formula

$$\sigma(a,b) \equiv 10a + 36b \ mod\ 45\,, \tag{8}$$

where $0 \leq a < 9$ and $0 \leq b < 5$. The Table 10-1 explicitly describes the mapping.

Denote the unit group U(45) of Z/45 by U. The ring-isomorphism σ restricts to a group-isomorphism

$$U(3^2) \times U(5) \cong U\,. \tag{9}$$

Choose $z_1 = 2$ and $z_2 = 2$ as the generators of $U(3^2)$ and U(5) respectively, and let

$$y_1 = \sigma(z_1, 1) = 11 \ mod\ 45\,, \tag{10}$$

$$y_2 = \sigma(1, z_2) = 37 \ mod\ 45\,, \tag{11}$$

then the order of U is given as

$$U = \{1,\ 37,\ 19,\ 28;\ 11,\ 2,\ 29,\ 38;\ 31,\ 22,\ 4,\ 13;\ 26,\ 17,\ 44,\ 8;\ 16,\ 7,\ 34,\ 43;\ 41,\ 32,\ 14,\ 23\}. \tag{12}$$

Set

$$f_1 = 30\,, \tag{13}$$

$$f_2 = 21\,, \tag{14}$$

and observe that

$$30^2 \equiv 0 \ mod\ 45\,, \quad 21^2 \equiv 36\,, \tag{15}$$

$$10 \times 30 \equiv 30 \ mod\ 45\,, \quad 36 \times 21 \equiv 36 \ mod\ 45\,, \tag{16}$$

$$36 \times 30 \equiv 0 \ mod\ 45\,, \quad 10 \times 21 \equiv 30 \ mod\ 45\,. \tag{17}$$

From the formulas, we see that

$$0U = \{0\}\,, \tag{18}$$

$$e_1U = \{10,\ 20,\ 40,\ 35,\ 25,\ 5\}\,, \tag{19}$$

$$e_2U = \{36,\ 27,\ 9,\ 18\}\,, \tag{20}$$

$$f_1U = \{30,\ 15\}\,, \tag{21}$$

$$f_2U = \{21,\ 12,\ 39,\ 3;\ 6,\ 42,\ 24,\ 33\}\,. \tag{22}$$

The set of U-orbits corresponding to

$$0,\ e_2,\ f_1,\ f_2,\ e_1,\ 1 \tag{23}$$

defines a partition of Z/45. Order the partition by (23), then the Fourier transform can be written as

$$F_{\Pi}(45) = PF(45)P^{-1}\,, \tag{24}$$

where P is the permutation matrix corresponding to (23).

Table 10 − 2 : $F_{\Pi}(45)$

	0	e_2	f_1	f_2	e_1	1
0	1	1_4^t	1_2^t	1_8^t	1_6^t	1_{24}^t
e_2	1_4	C_5	$E(4,2)$	$1_2^t \otimes C_5$	$E(4,6)$	$1_6^t \otimes C_5$
f_1	1_2	$E(2,4)$	$E(2,2)$	$E(2,8)$	$1_3^t \otimes C_3$	$1_3^t \otimes C_3 \otimes 1_4^t$
f_2	1_8	$1_2 \otimes C_5$	$E(8,2)$	$E(2,2) \otimes C_5$	$1_3^t \otimes C_3 \otimes 1_4$	$1_3^t \otimes C_3 \otimes C_5$
e_1	1_6	$E(6,4)$	$1_3 \otimes C_3$	$1_3 \otimes C_3 \otimes 1_4^t$	C_9	$C_9 \otimes 1_4^t$
1	1_{24}	$1_6 \otimes C_5$	$1_3 \otimes C_3 \otimes 1_4$	$1_3 \otimes C_4 \otimes C_5$	$C_9 \otimes 1_4$	$C_9 \otimes C_5$

where

$$C_3 = \begin{bmatrix} \beta^2 & \beta \\ \beta & \beta^2 \end{bmatrix}\,, \quad \beta = e^{2\pi i/3}\,, \tag{25}$$

$$C_5 = \begin{bmatrix} u^4 & u^3 & u & u^2 \\ u^3 & u & u^2 & u^4 \\ u & u^2 & u^4 & u^3 \\ u^2 & u^4 & u^3 & u \end{bmatrix}\,, \quad u = e^{2\pi i/5}\,, \tag{26}$$

and

$$C_9 = \begin{bmatrix} v^2 & v^4 & v^8 & v^7 & v^5 & v \\ v^4 & v^8 & v^7 & v^5 & v & v^2 \\ v^8 & v^7 & v^5 & v & v^2 & v^4 \\ v^7 & v^5 & v & v^2 & v^4 & v^8 \\ v^5 & v & v^2 & v^4 & v^8 & v^7 \\ v & v^2 & v^4 & v^8 & v^7 & v^5 \end{bmatrix}\,. \tag{27}$$

Modify C_9 as

$$C_9' = C_9 + \frac{1}{3}E(3,3) \otimes C_3\,, \tag{28}$$

then

$$F_{\Pi}(45) = F_c(45) - 0(15,15) \oplus \frac{1}{3}\begin{bmatrix} E(3,3) \otimes C_3 & E(3,3) \otimes C_3 \otimes 1_4^t \\ E(3,3) \otimes C_3 \otimes 1_4 & E(3,3) \otimes C_3 \otimes C_5 \end{bmatrix}\,, \tag{29}$$

where $F_c(45)$ has the same structure as $F_{\Pi}(45)$ in Table 10-2 except that C_9 is substituted by C_9'.

Now that $F_c(45)$ can be factorized as

$$F_c(45) = CA\,, \tag{30}$$

where C is given as

$$1 \oplus C_5 \oplus C_3 \oplus (C_3 \otimes C_5) \oplus C_9' \oplus (C_9' \otimes C_5)\,, \tag{31}$$

and A is given as

Table 10 – 3 : A

	0	e_2	f_1	f_2	e_1	1
0	1	1_4^t	1_2^t	1_8^t	1_6^t	1_{24}^t
e_2	-1_4	I_4	$-E(4,2)$	$1_2^t \otimes I_4$	$-E(4,6)$	$1_6^t \otimes I_4$
f_1	-1_2	$-E(2,4)$	$-E(2,2)$	$-E(2,8)$	$1_3^t \otimes I_2$	$1_3^t \otimes I_2 \otimes 1_4^t$
f_2	1_8	$-1_2 \otimes I_4$	$E(8,2)$	$-E(2,2) \otimes I_4$	$-1_3^t \otimes I_2 \otimes 1_4$	$1_3^t \otimes I_2 \otimes I_4$
e_1	-1_6	$-E(6,4)$	$1_3 \otimes I_2$	$1_3 \otimes I_2 \otimes 1_4^t$	I_6	$I_6 \otimes 1_4^t$
1	1_{24}	$-1_6 \otimes I_4$	$-1_3 \otimes I_2 \otimes 1_4$	$1_3 \otimes I_2 \otimes I_4$	$-I_6 \otimes 1_4$	I_{24}

By a permutation Q, A can be written as

$$A' = Q^{-1}AQ = \begin{bmatrix} A(9) & A(9) \otimes 1_4^t \\ -A(9) \otimes 1_4 & A(9) \otimes I_4 \end{bmatrix} , \tag{32}$$

where $A(9)$ is given as

$$A(9) = \begin{bmatrix} 1 & 1_2^t & 1_6^t \\ -1_2 & -E(2,2) & 1_3^t \otimes I_2 \\ -1_6 & 1_3 \otimes I_2 & I_6 \end{bmatrix} , \tag{33}$$

which was given in section 8, equation (17). If we take the order listed in (40) of section 9, the corresponding Fourier transform matrix will be

$$F_c'(45) = Q^{-1}F_c(45)Q = C'A' . \tag{34}$$

Set

$$C(9) = 1 \oplus C_3 \oplus C_9' , \tag{35}$$

then C' can be written as

$$C' = C(9) \oplus \big(C(9) \otimes C_5\big) . \tag{36}$$

Arithmetically, A is equivalent to

$$A(9) \otimes A(5) , \tag{37}$$

where A(5) is given as

$$A(5) = \begin{bmatrix} 1 & 1_4^t \\ -1_4 & I_4 \end{bmatrix} . \tag{38}$$

Observe the second part of equation (29), and set

$$F_s = \frac{1}{3}\begin{bmatrix} E(3,3) \otimes C_3 & E(3,3) \otimes C_3 \otimes 1_4^t \\ E(3,3) \otimes C_3 \otimes 1_4 & E(3,3) \otimes C_3 \otimes C_5 \end{bmatrix} , \tag{39}$$

by a permutation P_1

$$F_s = P_1^{-1}(E(3,3) \otimes F_{10})P_1 \,, \tag{40}$$

where F_{10} is given as

$$F_{10} = \frac{1}{3}\begin{bmatrix} C_3 & C_3 \otimes 1_4^t \\ C_3 \otimes 1_4 & C_3 \otimes C_5 \end{bmatrix} . \tag{41}$$

Factorize F_{10} as

$$F_{10} = C_{10}A_{10} \,, \tag{42}$$

where C_{10} is a block-diagonal matrix given as

$$C_{10} = \frac{1}{3}\begin{bmatrix} C_3 & 0 \\ 0 & C_3 \otimes C_5 \end{bmatrix} , \tag{43}$$

and A_{10} is pre-addition matrix given as

$$A_{10} = \begin{bmatrix} I_2 & I_2 \otimes 1_4^t \\ -I_2 \otimes 1_4 & I_2 \otimes I_4 \end{bmatrix} . \tag{44}$$

The permutation matrix P_1 is defined as

$$P_1 = I_6 \oplus P(24,8) \,. \tag{45}$$

Let

$$\underline{x}_{10} = \begin{bmatrix} a(10)+a(40)+a(25) \\ a(20)+a(35)+a(5) \\ a(1)+a(31)+a(16) \\ a(11)+a(26)+a(41) \\ a(37)+a(22)+a(7) \\ a(2)+a(17)+a(32) \\ a(19)+a(4)+a(34) \\ a(29)+a(44)+a(14) \\ a(28)+a(13)+a(43) \\ a(38)+a(8)+a(23) \end{bmatrix} , \tag{46}$$

we have

$$\underline{y}_{10} = F_{10}\underline{x}_{10} = C_{10}A_{10}\underline{x}_{10} \,. \tag{47}$$

Set

$$\underline{y} = P_1^{-1}\begin{bmatrix} \underline{y}_{10} \\ \underline{y}_{10} \\ \underline{y}_{10} \end{bmatrix} , \tag{48}$$

Then the whole computation of $F_{\Pi}(45)$ will be

$$\underline{b}' = F_c(45)\underline{a}' - \underline{b}_2 = CA\underline{a}' - \underline{b}_2 \,, \tag{49}$$

where

$$\underline{b}_2 = \begin{bmatrix} \underline{0}_{15} \\ \underline{y} \end{bmatrix} . \tag{50}$$

Direct computation of (49) will give the arithmetic count as in Table 10-4.

Table 10 – 4 : $\mathbf{F_{\Pi}(45)}$

Factor	R.A	R.M
C	1224	1376
A	324	0
$F_c(45)$	1548	1376
$F_{\Pi}(45)$	1812	1568

Variant 1.

Matrix C' and C_{10} are block-diagonal with each submatrix being skew-circulant, so that C' and C_{10} can be further diagonalized by corresponding size of Fourier transforms. Define F as

$$F = 1 \oplus F(2) \oplus F(6) \oplus F(4) \oplus \big(F(2) \otimes F(4)\big) \oplus \big(F(6) \otimes F(4)\big)\,, \tag{51}$$

set

$$F_9 = 1 \oplus F(2) \oplus F(6)\,, \tag{52}$$

then (51) can be written as

$$F = F_9 \oplus \big(F_9 \otimes F(4)\big). \tag{53}$$

Diadogalizing matrix C' by F, we have

$$C' = FDF\,, \tag{54}$$

where D can be written as

$$D = 1 \oplus D_3 \oplus D_6 \oplus D_5 \oplus \big(D_3 \otimes D_5\big) \oplus \big(D_6 \otimes D_5\big)\,, \tag{55}$$

where

$$D_3 = \frac{1}{2}\begin{bmatrix} -1 & 0 \\ 0 & \beta^2 - \beta \end{bmatrix}\,, \tag{56}$$

$$D_5 = F(4)^{-1}\begin{bmatrix} u^4 \\ u^3 \\ u \\ u^2 \end{bmatrix}\,, \tag{57}$$

and

$$D_6 = F(6)^{-1}\begin{bmatrix} v^2 \\ v^4 \\ v^8 \\ v^7 \\ v^5 \\ v \end{bmatrix}\,. \tag{58}$$

Rewrite D as

$$D = D(9) \oplus \left(D(9) \otimes D_5\right) , \tag{59}$$

where

$$D(9) = 1 \oplus D_3 \oplus D_6 . \tag{60}$$

Substitute (54) into (34), we have

$$F_c'(45) = FDFA' . \tag{61}$$

Diagonalizing the matrix C_{10} by

$$F_0 = F(2) \oplus \left(F(2) \otimes F(4)\right) , \tag{62}$$

then

$$C_{10} = F_0 D_{10} F_0 , \tag{63}$$

where D_{10} is a diagonal matrix given as

$$D_{10} = F_0^{-1} C_{10} F_0^{-1} . \tag{64}$$

Then the computation of F_{10} becomes

$$F_{10} = F_0 D_{10} F_0 A_{10} . \tag{65}$$

Table 10 – 5 : F_{10}

Factor	R.A	R.M
F_0	52	0
D_{10}	8	28
A_{10}	32	0
F_{10}	144	28

Table 10 – 6 : Variant 1. $F_{\Pi}(45)$

Factor	R.A	R.M
F	344	40
D	112	300
$F_c(45)$	1124	380
$F_{\Pi}(45)$	1328	408

Variant 2.

The cost of additions in (61) can be reduced by interchanging the order of computations. Set

$$F_c'(45) = FDBF\,, \tag{66}$$

where

$$B = FA'F^{-1}\,, \tag{67}$$

B can be explicitly written as

$$B = \begin{bmatrix} B(9) & B(9)\otimes e_5^t \\ -4B(9)\otimes e_5 & B(9)\otimes I_4 \end{bmatrix}\,, \tag{68}$$

where $B(9)$ was defined in section 8 equation (31). Arithmetically, B is equivalent to

$$B = B(9)\otimes B(5)\,, \tag{69}$$

where

$$B(5) = \begin{bmatrix} 1 & e_5^t \\ -4e_5 & I_4 \end{bmatrix}\,. \tag{70}$$

Table 10 – 7 : Variant 2. $F_c(45)$

Factor	R.A	R.M
B	96	{58}
$F_c(45)$	896	380+{58}

Variant 3.

To reduce the cost of additions required to perform the complex multiplications coming from the action of C, we note that each skew-circulant submatrix has the form

$$C_n = \begin{bmatrix} X_n & X_n^* \\ X_n^* & X_n \end{bmatrix}\,. \tag{71}$$

The same reasoning as in previous sections, we set

$$Y = Y(9)\oplus\left(Y(9)\otimes Y_5\right)\,, \tag{72}$$

and

$$H = H(9)\oplus\left(H(9)\otimes F(2)\otimes I_2\right)\,, \tag{73}$$

where

$$Y(9) = 1\oplus Y_3\oplus Y_6\,, \tag{74}$$

and

$$H(9) = 1 \oplus \left(F(2) \otimes I_2\right) \oplus \left(F(2) \otimes I_3\right) . \tag{75}$$

The important thing is that Y has only real or purely imaginary entries. Now we can write

$$C' = HYH , \tag{76}$$

and

$$F_c'(45) = HYHA' . \tag{77}$$

Table 10 – 8 : Variant 3. $F_c'(45)$

Factor	R.A	R.M
H	152	0
Y	336	468
$F_c'(45)$	964	468

11. BUILD UP A DFT LIBRARY BY MIXED ALGORITHMS

We can see from previous sections that the algorithms for the cases of 4p, 4pq, p^2, and p^2q (also for the case of pq in reference [11]) all have a basic factorization of a block-diagonal matrix C, which has skew-circulant matrices on the block-diagonal. We know that skew- circulant matrix can be computed by the corresponding Fourier transforms as given in Variant 1 of each section, so that the efficient small DFT library is a very important basis for the efficiency of large size Fourier transforms. Programming efforts have been made on Micro VAX II to build up a very efficient DFT library. Different algorithms have been tried for each size N, the fastest one has been selected into the DFT library as we can. That is why we call it Mixed Algorithms DFT Library.

For the small size N's, since the Micro VAX II has 16 registers, of which 12 can be used for general purposes, the most efficient algorithm is the one with the least number of real multiplications and real additions, that is Winograd small FFT algorithm. For the sizes N equal 2, 3, 4, 5, 7, 8, 16, the Winograd algorithms have been used to write the VAX assembly line-code.

For the primes larger than or equal to 11, the Winograd algorithm will increase the size of matrices very quickly, so that the stability becomes a problem. We choose Tolimieri's Prime Case Algorithm [10] to reduce the number of multiplications and additions.

For the sizes N having relatively prime factors, each factor has a very good Winograd small FFT, the Good-Thomas algorithm combined with the Winograd algorithm has been adopted for the line-coding. We call this Good-Thomas- Winograd algorithm (G-T-W) or Winograd Large FFT (WLFFT), which will be explained in detail in section 11-1.

We all know that Winograd algorithm is superior to the other algorithms as far as the multiplicative arithmetic complexity concerns, but the programming complexity makes it very difficult for applications. This is because that firstly it needs a whole new programming effort for each N, secondly to minimize the number of additions for the pre-addition and post-addition matrices is so arbitrary and difficult. The author suggests an approach to minimize the number of additions in section 11-2 called Huffman Coding Rule, because Huffman code ideas have been adopted.

11.1. Good-Thomas-Winograd algorithm on N=pq

When the size N has relative prime factors for which each of them has a very good Winograd small FFT algorithm, the G-T-W algorithm is very efficient for computing DFT. This algorithm

combines the Good-Thomas relative prime factor indexing scheme with the Winograd small FFT nesting algorithms. We will discuss in detail in this section the G-T-W algorithm for N with two relative prime factors

$$N = p\,q\,. \tag{1}$$

The Good-Thomas indexing scheme can be described as follows:

By Chinese Remainder Theorem, there exist N_1 and N_2, such that

$$N_1 p + N_2 q \equiv 1 \bmod N\,, \quad N_1, N_2 \in Z/N\,, \tag{2}$$

the input data is indexed as

$$i = i_1 N_2 q + i_2 N_1 p \bmod N\,, \tag{3}$$

and the output data is indexed as

$$k = i_1 q + i_2 p \bmod N\,, \tag{4}$$

where

$$i_1 = 0,\ 1,\ 2,\ \ldots\ (p-1)\,, \tag{5}$$

and

$$i_2 = 0,\ 1,\ 2,\ \ldots\,,(q-1)\,, \tag{6}$$

running first over i_1 and then over i_2.

Let P_1 and P_2 be the permutation matrices corresponding to the indexing set of (3) and (4), then the N-point Fourier transform will be

$$F(N) = P_2^{-1}\big(F(q) \otimes F(p)\big)P_1\,. \tag{7}$$

If both p and q have Winograd small FFT algorithms

$$F(p) = C_p B_p A_p\,, \tag{8}$$

and

$$F(q) = C_q B_q A_q\,, \tag{9}$$

where A_i is pre-addition matrix, B_i is diagonal matrix with only real entries, and C_i is post-addition matrix, i=p or q.

Then we have

$$F(N) = P_2^{-1}\big(C_q \otimes C_p\big)\big(B_q \otimes B_p\big)\big(A_q \otimes A_p\big)P_1\,, \tag{10}$$

rewrite it as

$$F(N) = P_2^{-1} C_{pq} B_{pq} A_{pq} P_1\,, \tag{11}$$

where

$$A_{pq} = A_q \otimes A_p\,, \tag{12}$$

$$B_{pq} = B_q \otimes B_p \,, \tag{13}$$

$$C_{pq} = C_q \otimes C_p \,. \tag{14}$$

Let us look at an example of N=pq: N=15. We have

$$F(3) = C_3 B_3 A_3 \,, \tag{15}$$

where

$$A_3 = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & 1 \\ 0 & 1 & -1 \end{bmatrix} , \tag{16}$$

$$B_3 = diag \begin{bmatrix} 1 \\ -1.5 \\ 0.8660254 \end{bmatrix} , \tag{17}$$

$$C_3 = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & i \\ 1 & 1 & -i \end{bmatrix} , \tag{18}$$

and

$$F(5) = C_5 B_5 A_5 \,, \tag{19}$$

where

$$A_5 = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & -1 & -1 & 1 \\ 0 & 1 & -1 & 1 & -1 \\ 0 & 1 & 0 & 0 & -1 \\ 0 & 0 & -1 & 1 & 0 \end{bmatrix} , \tag{20}$$

$$B_5 = diag \begin{bmatrix} 1 \\ -1.25 \\ 0.55901700 \\ 0.95105654 \\ -0.36327127 \\ 1.53884172 \end{bmatrix} , \tag{21}$$

$$C_5 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & -i & 0 & i \\ 1 & 1 & -1 & -i & -i & 0 \\ 1 & 1 & -1 & i & i & 0 \\ 1 & 1 & 1 & i & 0 & -i \end{bmatrix} . \tag{22}$$

The Good-Thomas indexing sets for input and output are

$$input: \ 0,\ 6,\ 12,\ 3,\ 9,\ 10,\ 1,\ 7,\ 13,\ 4,\ 5,\ 11,\ 2,\ 8,\ 14, \tag{23}$$

$$output: \ 0,\ 3,\ 6,\ 9,\ 12,\ 5,\ 8,\ 11,\ 14,\ 2,\ 10,\ 13,\ 1,\ 4,\ 7\,. \tag{24}$$

P_1 corresponds to (23), P_2 corresponds to (24), then

$$F(15) = P_2^{-1} C_{15} B_{15} A_{15} P_1 \,, \tag{25}$$

where

$$A_{15} = \begin{bmatrix}
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
0 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 \\
0 & 1 & -1 & -1 & 1 & 0 & 1 & -1 & -1 & 1 & 0 & 1 & -1 & -1 & 1 \\
0 & 1 & -1 & 1 & -1 & 0 & 1 & -1 & 1 & -1 & 0 & 1 & -1 & 1 & -1 \\
0 & 1 & 0 & 0 & -1 & 0 & 1 & 0 & 0 & -1 & 0 & 1 & 0 & 0 & -1 \\
0 & 0 & -1 & 1 & 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 & -1 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & -1 & 1 & 0 & 1 & -1 & -1 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 1 & -1 & 0 & 1 & -1 & 1 & -1 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & -1 & 0 & 1 & 0 & 0 & -1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 & -1 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 & -1 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & -1 & -1 & -1 & -1 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & -1 & 1 & 0 & -1 & 1 & 1 & -1 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 1 & -1 & 0 & -1 & 1 & -1 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & -1 & 0 & -1 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 & 1 & -1 & 0
\end{bmatrix}, \quad (26)$$

$$B_{15} = B_3 \otimes B_5 , \quad (27)$$

and

$$C_{15} = C_3 \otimes C_5 . \quad (28)$$

Explicit forms of C_{15} and B_{15} are similarly taking the tensor products as in A_{15}.

11.2 Huffman Coding Rule to minimize the number of additions

The case N=p

It is clear that to write a program for Winograd algorithm, the first thing is to figure out how many additions are required and how many temporary storages we have to adopt for the pre-addition matrix. Certainly we want the additions and the temporary storages as few as possible in general.

For small N's, the procedures to minimize the number of additions with the condition of minimum temporary storages, we suggest here, is to match the Huffman coding rule [15], the procedures are stated as follows:

In each step of reduction, find the pair columns with maximum number of matching elements, since each pair columns correspond to each pair input data, then set one temporary storage T_i

to equal the sum (or difference) of the pair input data, put T_i at the position of one of the pair input data, and set all the none zero matching elements in this column to be another symble, if there are other not matching elements in the column, and the matching elements in the other column to be 0. For each step follows the same rule, we will stop the reduction when there are no more than one matching element in any pair columns.

For example of N=5 of equation (20), the first thing is to compute A_5X, X is the input data vector. The procedures are listed as follows:

Since each input data corresponds to each column, let us put the input data vector X on top of A_5 matrix,

$$\begin{bmatrix} x_0 & x_1 & x_2 & x_3 & x_4 \\ 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & -1 & -1 & 1 \\ 0 & 1 & -1 & 1 & -1 \\ 0 & 1 & 0 & 0 & -1 \\ 0 & 0 & -1 & 1 & 0 \end{bmatrix} . \tag{29}$$

Step 1. Column 1 and column 4 have three matching elements , and column 2 and column 3 have three matching elements, let us set $T_0 = x_1 + x_4$ and $T_1 = x_2 + x_3$, put T_0 and T_1 on top of x_1 and x_2 respectively. Set the matching elements in column 1 and 2 to be $\times$, and the corresponding matching elements in column 3 and 4 to be 0, then the matrix will be

$$\begin{bmatrix} & T_0 & T_1 & & \\ x_0 & x_1 & x_2 & x_3 & x_4 \\ 1 & \times & \times & 0 & 0 \\ 0 & \times & \times & 0 & 0 \\ 0 & \times & -\times & 0 & 0 \\ 0 & 1 & -1 & 1 & -1 \\ 0 & 1 & 0 & 0 & -1 \\ 0 & 0 & -1 & 1 & 0 \end{bmatrix} . \tag{30}$$

Step 2. Set $x_3 = x_3 - x_2$ and $x_4 = x_1 - x_4$, then x_1 and x_2 are out of line, we can reuse them later. Now we change the symble $\times$ back to 1,

$$\begin{bmatrix} x_0 & T_0 & T_1 & x_3 & x_4 \\ 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix} . \tag{31}$$

Step 3. Set $x_1 = T_0 + T_1$, then (31) will be

$$\begin{bmatrix} & x_1 & & & \\ x_0 & T_0 & T_1 & x_3 & x_4 \\ 1 & \times & 0 & 0 & 0 \\ 0 & \times & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix} . \tag{32}$$

Step 4. Set $x_0 = x_0 + x_1$, $T_1 = T_0 - T_1$ and $x_2 = x_3 + x_4$, then we will have a diagonal matrix

$$\begin{bmatrix} x_0 & x_1 & T_1 & x_2 & x_4 & x_3 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} . \tag{33}$$

The matrix A_5 multiplying a vector X becomes a vector

$$\begin{bmatrix} x_0 \\ x_1 \\ T_1 \\ x_2 \\ x_4 \\ x_3 \end{bmatrix} , \tag{34}$$

after 8 additions. But by direct computation we need 15 additions.

The case N=pq

For large N with two relatively prime factors, the pre-addition matrix $A_{pq} = A_q \otimes A_p$ is written as two matrices' tensor product, we will use the Nested Huffman Coding Rule. Let us state it by an example of N=15, the equation (26).

Let us rewrite (26) as

$$A_{15} = \begin{bmatrix} A_5 & A_5 & A_5 \\ 0 & A_5 & A_5 \\ 0 & A_5 & -A_5 \end{bmatrix} , \tag{35}$$

(35) has the same form as (16), the matrix A_3, except that A_5 replaced 1. Apply the Huffman Coding Rule to A_3, by the Good- Thomas data indexing given in (23), set

$$T_0 = x_{10} + x_5$$

$$T_1 = x_1 + x_{11}$$

$$T_2 = x_7 + x_2$$

$$T_3 = x_{13} + x_8$$

$$T_4 = x_4 + x_{14} ,$$

(35) will be

$$\begin{bmatrix} A_5 & \underline{\times} & 0 \\ 0 & \underline{\times} & 0 \\ 0 & A_5 & -A_5 \end{bmatrix} . \tag{36}$$

Set

$$x_5 = x_{10} - x_5$$

$$x_{11} = x_1 - x_{11}$$

$$x_2 = x_7 - x_2$$

$$x_8 = x_{13} - x_8$$

$$x_{14} = x_4 - x_{14}$$

and

$$x_0 = x_0 + T_0$$

$$x_6 = x_6 + T_1$$

$$x_{12} = x_{12} + T_2$$

$$x_3 = x_3 + T_3$$

$$x_9 = x_9 + T_4 ,$$

we will have

$$\begin{bmatrix}
x_0 & x_6 & x_{12} & x_3 & x_9 & T_0 & T_1 & T_2 & T_3 & T_4 & x_5 & x_{11} & x_2 & x_8 & x_{14} \\
1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & -1 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & -1 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 & & & & & & & & & & & & & & \\
0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & -1 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 1 & -1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
 & & & & & & & & & & & & & & \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & -1 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 1 & -1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & -1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 & 0
\end{bmatrix} . \quad (37)$$

We can see from (37) that the matrix becomes block-diagonal with A_5 on the diagonal. We can use exactly the same rule as we used before to minimize the number of additions for each submatrix A_5.

11.3 A list of DFT algorithms corresponding to specific N's

In this section, we give a list of algorithms corresponding to specific size N's which have been collected into the DFT Library. For each N, we give its arithmetic count: the number of real additions and the number of real multiplications due to complex input.

Table 11 – 1. Winograd Small FFT Algorithm

Size	R.A	R.M
2	4	0
3	12	4
4	16	0
5	34	10
7	72	16
8	52	4
16	148	20

Tolimieri's Prime Case Algorithm

Tolimieri's Prime Case algorithm in [10] is given as followwing:

$$F_{\Pi} = H^{-1} Y B_1 H \,, \tag{1}$$

where

$$H = 1 \oplus \left(F(2) \otimes I_{\frac{p'}{2}} \right) , \tag{2}$$

$$B_1 = \begin{pmatrix} 1 & f^t \\ -2\underline{f} & \overline{I}_{p'} \end{pmatrix} , \tag{3}$$

and

$$Y = 1 \oplus (X + X^*) \oplus (X - X^*) \,. \tag{4}$$

The permutation corresponding to Π is given by Rader algorithm.

Table 11 – 2 Tolimieri's Prime Case Algorithm

Size	R.A	R.M
11	140	100
13	192	144
17	320	256
19	396	361

Table 11 – 3. Good – Thomas – Winograd Algorithm for N = pq

Size	R.A	R.M	Algorithms
6	36	8	3-Winograd
10	92	20	5-Winograd
12	96	16	3,4-Winograd
14	180	32	7-Winograd
15	168	34	3,5-Winograd
18	204	40	9-Winograd
20	228	40	4,5-Winograd
21	312	52	3,7-Winograd
24	246	36	3,8-Winograd
28	420	64	4,7-Wimnograd
30	396	68	2,3,5-Winograd

Tolimieri's N = pq Case Algorithm

The arithmetic count given below is due to the Tolimieri's pq case algorithms the Variant 1 in [11].

$$F_{\Pi} = FDFA\,, \tag{5}$$

where

$$F = 1 \oplus F(p') \oplus F(q') \oplus \left(F(p') \otimes F(q')\right), \tag{6}$$

$$D = 1 \oplus D_p \oplus D_q \oplus (D_p \otimes D_q)\,, \tag{7}$$

and

$$A = \begin{pmatrix} A(p) & A(p) \otimes 1_{q'}^{t} \\ -A(p) \otimes 1_{q'} & A(p) \otimes I_{q'} \end{pmatrix}. \tag{8}$$

Table 11 – 4. Tolimieri's N = pq Case Algorithm

Size	R.A	R.M
15	236	40
21	424	112
33	896	232
35	844	200
39	988	232
51	1394	304
93	3316	760
105	3164	604

Table 11 – 5. N = 4p Case Algorithm

Size	R.A	R.M
12	96	16
20	224	64
28	528	144
44	1136	304
52	1248	304
68	1824	400
124	4368	1008

Table 11 – 6. $N = p^2(p^2q)$ Case Algorithm

Size	R.A	R.M
9	100	40+{4}
25	592	180
45	1328	408

Table 11 – 7. N = 4pq Case Algorithm

Size	R.A	R.M
60	1072	248
84	2072	260

For all N's listed in Table 11-5, 11-6, 11-7, the algorithms used are the Variant 1., the algorithms by convolution theorem in the previous sections.

12. IMPLEMENTATION CONSIDERATIONS

For all algorithms and all size N's, the timings are based on the complex input and complex output with real and imaginary parts floating point real numbers.

We implemented the algorithms given in previous sections on the DEC's VAX architecture machine, the Micro VAX II, the timings will be listed in section 13.

As we know that register accessing is about ten times faster than memory accessing, keeping the best use of registers is the most important. The VAX has 16 registers, 12 of them can be used for general purposes. One register is sometimes reserved for addressing. Since most actions of Fourier transform are butterfly, that needs one register for temporary storage, so that we have, at most time, 10 registers on hand. If we can match as well as possible the register structure, the code will run fast.

The Micro VAX II has 14 addressing modes, most of which can be indexed by one of the registers. By experience, we realized that some more powerful instructions seem to run slower than that equivalent sequences of simpler instructions, so that when we do memory accessing, we only use auto increment , auto decrement and register defered addressing modes.

To reduce memory accessing, we should do as much computing as possible within registers between loads and stores. For small size FFT's, N is less than or equal to 6, we put all real and imaginary input data in registers by auto increment addressing mode. All the additions and multiplications can be done in registers, then by auto decrement addressing mode, we put back the transformed data to the output data array at the last additions stage.

For the sizes from 6 to 12, we put the real part of input data in registers by auto increment addressing mode, and do as much as possible the computations. As we know that for the Winograd algorithm or G-T-W algorithm and for some small sizes of pq and 4p algorithms, computations of real part and imaginary part will not interact till the last additions stage. When we arrive the last step of real part computation, we free registers for imaginary data computation by loading the data into stack pointer(SP). Since imaginary part and real part computation are exactly the same except the input data address, we simply copy down the code. After imaginary part computation is done, we leave the data still in registers, and pop out real part from (SP) one by one, and do the last stage additions. It is clear that registers are being used most efficiently.

For the sizes larger than 12, we use register defered addressing mode. First we put 10 real input data in registers corresponding to the maximum match of columns in the pre-addition

matrix by Huffman Coding Rule, and do as much as possible the computations, and leave part or all of the data in temporary memory, and do the same for the rest of real part to the last stage, then load the computed real part in (SP), and do the same to imaginary part.

Choosing the way of tensor products in an algorithm to match the architecture of computers is also very important. For example of F(30), by Good-Thomas algorithm, we have

$$F(30) = P_2[F(3) \otimes F(2) \otimes F(5)]P_1 \ . \tag{1}$$

The order of F(2), F(3), and F(5) we take in (1) is for the purpose of matching the number of registers, since $F(2) \otimes F(5)$ needs 11 registers to do the computation. And also for example of n=33, the pq case algorithm [11],

$$F_\Pi(33) = C_{33}A_{33} \ , \tag{2}$$

where

$$C_{33} = 1 \oplus C_3 \oplus C_{11} \oplus (C_3 \otimes C_{11}) \ , \tag{3}$$

and $C_3 \otimes C_{11}$ is computed by

$$F(2) \otimes F(10) \ . \tag{4}$$

Since the size of F(10) just matches the number of registers, most computation can be done within registers without memory accessing.

Even when writing Fortran program, we still can consider the factor of registers. For example, we try to put as many as possible operations, from the same input data, close in order to reduce memory accessing.

All the algorithms given in previous sections need corresponding small size FFT subroutines, it is clear that if we have a very efficient DFT library, we can build the very efficient large size Fourier transforms. A very efficient assembly code small DFT Library has been built and we will enlarge it. The timing results will be listed in section 13.

The interesting point is that we can recursively use the 4p algorithm to enlarge the DFT library. For example of n=12, $12 = 4 \times 3$, we use the 4p algorithm to write a very efficient code as indicated in Table 13-2., and n=52 can be built based upon F(12), from (50) of section 3, we have

$$F_\Pi(52) = F_0(52)D(52)F_0(52)A(52) \ , \tag{5}$$

where $F_0(52)$ from (45) of section 3 is given as

$$F_0(52) = I_4 \oplus \left(I_4 \otimes F(12)\right) \ . \tag{6}$$

The same reasoning as above, we can build F(212) from the efficient F(52) algorithm,

$$F_{\Pi}(212) = F_0(212)D(212)F_0(212)A(212)\,, \tag{7}$$

where $F_0(212)$ is based upon the F(52) as

$$F_0(212) = I_4 \oplus \left(I_4 \otimes F(52)\right)\,. \tag{8}$$

For the case of n=12, we use Variant 4. to write VAX assembly code to save some arithmetic count,

$$F_{\Pi}(12) = HYBH\,. \tag{9}$$

In the matrix B, there are some multiplications by 2

$$B = \begin{bmatrix} A' & A' \otimes [0\ \ 1] \\ -2A' \otimes \begin{pmatrix} 1 \\ 0 \end{pmatrix} & A' \otimes I_2 \end{bmatrix}\,. \tag{10}$$

Fortunately that corresponding to the multiplication by 2 in the next step computation Y, we have the case as

$$R * (-0.5) = (R_i\ -\ 2 * R_j) * (-0.5)\,, \tag{11}$$

(11) can be computed in the following way to save a multiplication

$$R_j\ + (-0.5) * R_i\,. \tag{12}$$

By (12), we saved 8 multiplications by 2, or we can say, 8 real additions. F(12) ends up with 96 real additions and 16 real multiplications.

The n=20 case was also wrtten in assembly based on Variant 4 algorithm, the size 20 just fits the number of registers which can be used twice, so that the code comes out very efficient as we can see from the timing in Table 13-3.

Observe the arthmetic count of N=93 and N=105 in Table 11-4, and the timing results in Table 13-2. We can predict that the more factors the size N has, the more efficient the Fourier transform will be, comparing to the close size which has fewer factors. This can be explained as follows: the smaller factor has the smaller size skew-circulant matrix, which means the smaller size FFT, a few small size FFT tensor product is much more efficient than a large size FFT, which is close to their product. For example of N=93 and N=105, F(93) has the form

$$F_{\Pi}(93) = C_{93}A_{93}\,, \tag{13}$$

where

$$C_{93} = 1 \oplus C_3 \oplus C_{31} \oplus (C_3 \otimes C_{31})\,, \tag{14}$$

which is computed by

$$F_{93} = 1 \oplus F(2) \oplus F(30) \oplus (F(2) \otimes F(30)) , \tag{15}$$

where F(30) consists of

$$F(3) \otimes F(2) \otimes F(5) . \tag{16}$$

But F(105) has the form

$$F_\pi(105) = C_{105} A_{105} , \tag{17}$$

where

$$C_{105} = 1 \oplus C_3 \oplus C_5 \oplus (C_3 \otimes C_5) \oplus C_7 \oplus (C_3 \otimes C_7) \oplus (C_5 \otimes C_7) \oplus (C_3 \otimes C_5 \otimes C_7) , \tag{18}$$

which is computed by

$$F_{105} = 1 \oplus F(2) \oplus F(4) \oplus (F(2) \otimes F(4)) \oplus F(6) \oplus (F(2) \otimes F(6)) \oplus (F(4) \otimes F(6)) \oplus (F(2) \otimes F(4) \otimes F(6)) , \tag{19}$$

where F(6) consists of

$$F(2) \otimes F(3) . \tag{20}$$

Since F(2) and F(4) are very simple, which need no multiplications, only F(3) has 4 real multiplications for complex input, so that F_{105} is simpler than F_{93}.

We have the same supporting results in Table 11-5. and Table 13-3., where F(60) is more efficient than F(52). For the case of F(60), it needs only F(2) and F(4) in its computation, but F(52) needs calling F(12), which consists F(3) and F(4), where F(3) needs multiplications.

13. EFFICIENCY COMPARISONS

For most sizes of N less than or equal to 30, VAX assembly codes have been created, whose run time is very efficient as we can see from Table 13-1. In Table 13-1., we also give the algorithm chosen which is the fastest one as we have tested on Micro VAX II.

For all N's listed in Table 13-2. and Table 13-3., the algorithms used are Variant 1 algorithms by convolution theorem. The main programs are written in Fortran and they call the small DFT subroutines given in Table 13-1.

Table 13-1. The Efficiency of Small DFT Library(Mixed Algorithms)

Size	Factors	CPUTIME	Algorithms
3		0.085ms.	WFTA
4	2^2	0.083ms.	WFTA
5		0.178ms.	WFTA
6	2×3	0.177ms.	G-T-W
7		0.317ms.	WFTA
8	2^3	0.222ms.	WFTA
9	3^2	0.535ms.	p^2 case
10	2×5	0.396ms.	G-T-W
11		0.859ms.	Tol'p
12	3×4	0.415ms.	4p case
14	2×7	0.725ms.	G-T-W
15	3×5	0.699ms.	G-T-W
16	2^4	0.680ms.	WFTA
20	4×5	0.901ms.	G-T-W
30	$2 \times 3 \times 5$	1.687ms.	G-T-W

All the programs for the sizes listed in Table 13-1. , Table 13-2. and Table 13-3., are tested in two ways, one is to create an arbitrary input data array, and run the program, then use the output data as the input data, to run the same program just by changing the sign, then the output data are compared with the original created arbitrary input data, they should be the same or very close.

Another way to test the programs is: using the same input data to run the program and the direct computation program which is based on defination

$$b(k) = \sum_{0 \le j < N} a(j) w^{jk}, \quad 0 \le k < N; \quad w = e^{2\pi i/N}. \tag{1}$$

The output data arraies of the two programs should be the same or very close.

Table 13-2. Timing Comparisons (pq and pqr cases)

Size	Factors	pq(pqr)	Dec.Labstar
4	2^2		0.82 ms.
8	2^3		1.49 ms.
15	3×5	1.13 ms.	
16	2^4		2.87 ms.
21	3×7	2.23 ms.	
32	2^5		5.78 ms.
33	3×11	4.35 ms.	
35	5×7	4.22 ms.	
39	3×13	4.78 ms.	
51	3×17	6.97 ms.	
64	2^6		12.49 ms.
93	3×31	16.03 ms.	
105	$3 \times 5 \times 7$	16.01 ms.	
128	2^7		27.80 ms.

Table 13-3. Timing Comparisons (4p ,4pq and p^2q cases)

Size	Factors	4p(4pq)	Dec.Labstar
4	2^2		0.82 ms.
8	2^3		1.49 ms.
12	4×3	0.415 ms.	
16	2^4		2.87 ms.
20	4×5	0.985 ms.	
28	4×7	2.92 ms.	
32	2^5		5.78 ms.
44	4×11	5.86 ms.	
45	$3^2 \times 5$	6.52 ms.	
52	4×13	6.53 ms.	
60	$4 \times 3 \times 5$	6.39 ms.	
64	2^6		12.49 ms.
68	4×17	9.27 ms.	
124	4×31	20.42 ms.	
128	2^7		27.80 ms.

where ms.= 10^{-3} second.

From Table 13-1., 13-2. and 13-3., we can see that these algorithms do not only fullfill those sizes which Cooley-Tukey FFT does not have, and their timing for most sizes are much better than the nearest point Cooley-Tukey algorithm. They cover most points of Fourier transform for n less than 128.

14. SUMMARY AND CONCLUSIONS

Algorithms are designed and implemented on Micro VAX II for N-point FFT, N equals 4p, 4pq, p^2 and p^2q, where p and q are distinct odd primes. By applying tensor product rules to matrix F_Π, which describes the FFT relative to input and output data ordered by the ring-structure of the indexing set Z/N, we arrive at the fundamental factorization

$$F_\Pi = CA\,, \tag{1}$$

where C is a block-diagonal matrix, and A is a matrix with all of whose entries are 0, 1, or -1.

For the simplest case of N=4, the algorithm is given by the factorization

$$F_\Pi(4) = C(4)A(4)\,, \tag{2}$$

where

$$C(4) = I_2 \oplus \begin{pmatrix} 1 & i \\ 1 & -i \end{pmatrix}\,, \tag{3}$$

and

$$A(4) = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & -1 \end{bmatrix}\,. \tag{4}$$

And in the case of N=p, p a prime,

$$C = 1 \oplus C(p)\,, \tag{5}$$

where C(p) is the skew-circulant matrix (Winograd core)

$$C(p) = \begin{bmatrix} v & v^z & \cdots & v^{z^{p-2}} \\ v^z & & & \\ \vdots & & & \\ v^{z^{p-2}} & & & \end{bmatrix}\,, \tag{6}$$

where $v = e^{2\pi i/p}$ and z is a generator of the unit group U(p) of Z/p, and

$$A = A(p) = \begin{bmatrix} 1 & 1_{p'}^t \\ -1_{p'} & I_{p'} \end{bmatrix}\,, \quad p' = p-1\,. \tag{7}$$

To derive (1), when N=4p, p odd prime, we use the Chinese Remainder Theorem. In this case

$$C = C' \oplus (C' \otimes C_p)\,, \tag{8}$$

where C' equals C(4) or $C^*(4)$, and C_p is rotated Winograd core, and

$$A = \begin{bmatrix} A(4) & A(4) \otimes 1_{p'}^t \\ -A(4) \otimes 1_{p'} & A(4) \otimes I_{p'} \end{bmatrix}\,. \tag{9}$$

Extensions of these methods to transform size N=4m, m a product of distinct primes, are given by the case of N=4pq in section 5. The algorithms can be constructed recursively. Set

$$C'' = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & w^{e_1} \\ 0 & 0 & 1 & -w^{e_1} \end{pmatrix}, \tag{10}$$

$$C' = C'' \oplus (C'' \otimes C_p), \tag{11}$$

$$C = C' \oplus (C' \otimes C_q), \tag{12}$$

and

$$A'' = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & -1 \end{pmatrix}, \tag{13}$$

$$A' = \begin{pmatrix} A'' & A'' \otimes 1^t_{p'} \\ -A'' \otimes 1_{p'} & A'' \otimes I_t \end{pmatrix}, \tag{14}$$

$$A = \begin{pmatrix} A' & A' \otimes 1^t_{q'} \\ -A' \otimes 1_{q'} & A' \otimes I_{q'} \end{pmatrix}, \tag{15}$$

then we have

$$F_\Pi(4pq) = CA. \tag{16}$$

C_p and C_q are rotated Winograd cores, determined from C(p) and C(q), by the idempotents of the ring Z/N.

For the case of $N=p^2$, after a modification has been taken, $F_\Pi(p^2)$ has two parts,

$$F_\Pi(p^2) = F_c(p^2) - 0(p,p) \oplus \left(\frac{1}{p} E(p,p) \otimes C_p\right), \tag{17}$$

the first part can be factorized as

$$F_c(p^2) = CA, \tag{18}$$

where

$$C = 1 \oplus C_p \oplus C'_{p^2}, \tag{19}$$

$$A = \begin{bmatrix} 1 & 1^t_{p'} & 1^t_s \\ -1_{p'} & -E(p',p') & 1^t_p \otimes I_{p'} \\ -1_s & 1_p \otimes I_{p'} & I_s \end{bmatrix}, \tag{20}$$

and the second part of (17) is very easy to compute, since E(p,p) represents the computation of C_p multiplying a vector repeating p times. The main part of the computation of $F_\Pi(p^2)$, $F_c(p^2)$ has the same form as in (1).

The algorithms for the case of $N=p^2q$ can be constructed in the same fashion as in the case of N=4p, except that $F_\Pi(4)$ is substituted by $F_\Pi(p^2)$.

The factorization (1) decomposes the computation of the FFT into two distinct stages, an addition stage given by the matrix A and a multiplication stage given by the matrix C. For implementation on the Micro VAX II, we prefer the variant of (1) given by the factorization

$$F_n = FDFA\,, \tag{21}$$

where F is a block-diagonal matrix having small FFT blocks and tensor product of these blocks, D is a diagonal matrix coming from the convolution theorem applied to C, and A is an addition matrix with only 1, 0, and -1's. Arithmetically more efficient algorithms are derived (Variant 2 and 4), but factorization (21) has a uniform structure which greatly simplifies programming.

The features of these new class of FFT algorithms can be stated as follows:

a. Arithmetic efficiency

these algorithms preserve the Winograd's multiplicative structure, they are constructed by smaller size FFT's, each of these small size FFT can be computed again by the most efficient available algorithms.

b. Programming simplicity

comparing with Winograd algorithms, these algorithms can be written in a nested way and with subroutine calls for a class of sizes which have the same ring-structure. We can build up a very efficient DFT library for the small DFT subroutine calls.

c. Good structuring

tensor product formulation of these algorithms makes them possible to be modified to produce variants which offer options as to operational count and arithmetic balance, and to match the hardware environments.

d. Easy to be parallelized

these algorithms are constructed by small pieces using tensor products, parameters can be chosen such that they are matching the parallel structure of the computers.

e. Linear coverage of transform sizes

the Cooley-Tukey algorithms are only very good for transform sizes of two's power $N=2^m$, they distribute exponentially, as m goes bigger, the interval between two nearest applicable points becomes larger. The algorithms derived in this thesis distribute the transform sizes linearly, they cover more points than Cooley-Tukey algorithms.

REFERENCES

[1] Heideman, M. T., Johnson, D. H. and Burrus, C. S. "Gauss and the History of the Fast Fourier Transform",*IEEE ASSP Magazine*, October 1984.

[2] Cooley, J. W., and Tukey, J. W. "An algorithm for the Machine Calculation of Complex Fourier Series", *Math. Comput.*, vol. 19, no. 2, pp 297-301.

[3] Temperton, C. "A Note on Prime Factor FFT Algorithms". *J. Comp. Phys.*, 52, (1983): 198-204.

[4] Winograd, S. "On Computing the Discrete Fourier Transform", *Proc. Nat. Acad. Sci. USA.*, vol. 73. no. 4,(April 1976):1005-1006.

[5] Winograd, S. "On Computing the Discrete Fourier Transform", *Math. of Computation*, Vol.32, No. 141, (Jan. 1978):pp 175-199.

[6] Good, I. J. "The Interaction Algorithm and Practical Fourier Analysis", *J.R. Statist. Soc. B*, vol 20, no. 2, (1958):pp 361-372.

[7] Thomas, L. H. "Using a Computer to Solve Problems in Physics, Application of Digital Computers", *Ginn and Co.*, Boston, Mass. 1963.

[8] Auslander, L., Cooley, J.W. and Silberger, A.J. "Number Stability of Fast Convolution Algorithms for Digital Filtering", *VLSI Signal Processing*, IEEE Press, (1984):pp. 172-213.

[9] Johnson, R. W. and Tolimieri, R. "Implementing Fast Fourier Transform Algorithms", to be published.

[10] Tolimieri, R. *Algorithms for DSP*, a book prepared.

[11] Johnson, R. W., LU, Chao and Tolimieri, R. "Fast Fourier Transform Algorithms for the Sizes of Product of Distinct Primes and Implementations on VAX", to be published.

[12] Temperton, C. "Implementation of Prime Factor FFT Algorithm on Cray-1", to be published.

[13] Lu, C. and Silberger, A. "One Variable and Several Variables FFT's Parallelized with Identical Data Flow", to be published.

[14] Blabut, R. E. *Fast Algorithms for Digital Signal Processing*, Addison-Wesley, Reading, Mass., 1985.

[15] Hamming, R. W. *Coding and Information Theory.* Prentice-Hall, Inc. 1980.

LITERATURE CONSULTED

Auslander, L., Feig, E. and Winograd, S. "New Algorithms for the Multi-dimensional Discrete Fourier Transform", *IEEE Trans. Acoust., Speech, Signal Processing*, v. ASSP-31, No. 2, (1983): pp. 388-403.

Auslander, L., Tolimieri, R. and Rodriguez, D. "On Tensor Products Formulation of Additive Fast Fourier Transform Algorithms" , to be published on *IEEE Trans. Acoust., Speech, Signal Processing.*

Burrus, C. S. and Eschenbacher, P. W. "An In-place, In-order Prime Factor FFT Algorithm", *IEEE Trans. Acoust., Speech, Signal Proc.*, ASSP-29(1979):806-817.

Johnson, H. and Burrus, C. S. "The Design of Optional DFT Algorithms Using Dynamic Programming", *IEEE Trans. Acoust., Speech, Signal Proc.* ASSP-31, (1983):378-387.

Kolba, D. P. and Parks, T. W. "Prime Factor FFT Algorithm Using High Speed Convolution", *IEEE Trans. Acoust., Speech and Signal Proc.*, ASSP-25(1977):281-294.

Temperton, C. "Self-sorting Mixed-radix Fast Fourier Transforms", *J. Comp. Phys.*, 52, (1983):1-23.

Temperton, C. "Implementation of Self-sorting In-place Prime Factor FFT Algorithm", *J. Comp. Phys.*, 58, (1985):283-299.

Winograd, S. *Arithmatic Complexity of Computations*, CBMS-NSF Conference Series in Applied Math. No. 33, SIAM, 1980.