Large DFT Modules: 11, 13, 16, 17, 19, and 25. Revised ECE Technical Report 8105

By:

Howard Johnson

C. Sidney Burrus

Large DFT Modules: 11, 13, 16, 17, 19, and 25. Revised ECE Technical Report 8105

By:
Howard Johnson
C. Sidney Burrus

Online:
< http://cnx.org/content/col10569/1.7/ >

C O N N E X I O N S

Rice University, Houston, Texas

Collection structure revised: September 14, 2009
PDF generated: August 18, 2011
For copyright and attribution information for the modules contained in this collection, see p. 61.

Table of Contents

Chapter 1

Large DFT Modules: 11, 13, 16, 17, 19, and 25[1]

1.1 Introduction

This report describes three large DFT modules (17,19,25) which were developed by the first author, Howard Johnson, in June of 1981, and two previously undocumented modules (11,13) which were originally generated at Stanford in 1978 [8].

The length 17 and 19 modules were created in the style of Winograd's convolutional DFT programs with strict adherence to three additional module development principles. First, as much code as possible was automatically generated. This included use of FORTRAN programs to generate the input and output mapping statements and the multiplication statements, and heavy use of EDIT commands to copy redundant sections of code. The code for imaginary data manipulation was copied directly from a working listing of code for the real part. All discussion below therefore centers on producing code only for the real part of the input data array. Even the EDIT commands for copying sections of code and substituting variable names were themselves listed in a command file. In this way, the programmer was prevented from introducing occasional typographical errors which are the bane of the DFT module debugger. Errors which did occur tended to be very large and obvious. Test routines were written to test particularly difficult sections of code before they were inserted into the DFT module (such as the modulo z^8+1 convolution subsection).

Once the reduction, or PRE-WEAVE, section was written, the reconstruction, or POST-WEAVE, section was arranged to be the transpose of the reduction equations, according to the method of 'transposing the tensor' [6]. Although the problem of minimizing the number of additions in a module is not necessarily solved by transposing the tensor, due to the inordinate difficulty of finding suitable substitutions which would abate the addition count, and the high probability of error involved in making such substitutions, it was decided to use this method. This method also provides a convenient way to check the correctness of the reconstruction procedure by computing the matrices of the reduction and reconstruction subroutines and testing to see that they are indeed a transpose pair.

Intrinsic to the method of transposing the tensor is the fact that the matrix B used to compute the algorithm's multiplication coefficients from the Nth roots of unity is generally more complicated than either the reduction matrix or its transpose, the reconstruction matrix. This result is a consequence of B having been generated from Toom-Cook polynomial reconstruction procedures and also CRT polynomial reconstructions, which are both known to be more complicated than their associated reduction procedures. The problem of finding B in order to compute a set of multipliers may be neatly circumvented by directly solving a set of linear equations to find a coefficient vector which makes the algorithm work. The details of this trick are not reported here, but may be found in [3]. Suffice to say that given working FORTRAN subroutines for

[1] This content is available online at <http://cnx.org/content/m17413/1.7/>.

the reduction and reconstruction procedures, a FORTRAN program exists which will solve for the correct coefficients.

The length 25 module does not follow the traditional Winograd approach. This module is an in-line code version of a common-factor 5x5 DFT. Each length 5 DFT is a prime-length convolutional module. The output unscrambling is included in the assignment statements at the end of the program. Some of the length 5 modules used in this program are implemented as scaled versions of conventional length 5 modules in order to save some multiplies by $1/4$. The scaling factors are then compensated for by adjusting the twiddle factors. This module has three multiply sections, one for the row DFT's with a data expansion factor of $6/5$, one for the twiddle factors (expansion=$33/25$) and on for the column DFT's (expansion=$6/5$).

Modules for lengths 11 and 13 are very similar in spirit to the length 19 and 17 modules. Derivations are presented for both the 11 and 13 length modules which are consistent with the listings, although these interpretations may not agree with the original intentions of the designer [8] they are correct in the sense that the algorithms could have been derived in the stated manner. Both the modules are of prime length and they are implemented in Winograd's convolutional style.

FORTRAN listings for all five modules are included with this report in a subroutine form suitable for use in Burrus' PFA program [1]. Addition and multiplication counts given are for complex input data.

1.2 17 Module: 314 Adds / 70 Mpys

This module closely follows the traditional Winograd prime-length approach.

1. Use the index map $\overline{x}(n) = x(<3^n>_{mod17})$ to convert the DFT into a length 16 convolution, plus a correction term for the DC component.
2. Reduce the length 16 convolution modulo all the irreducible factors of $z^{16}-1$. (Irreducible over the rationals).

$$\begin{aligned} mod z^8+1 &: r108 - r115 \\ mod z^8-1 &: r100 - r107 \end{aligned} \quad (1.1)$$

From z^8-1 data

$$\begin{aligned} mod z^4+1 &: r31 - r34 \\ mod z^4-1 &: r200 - r203 \end{aligned} \quad (1.2)$$

From z^4-1 data

$$\begin{aligned} mod z^2+1 &: r35 - r36 \\ mod z^2-1 &: r204 - r205 \end{aligned} \quad (1.3)$$

From z^2-1 data

$$\begin{aligned} mod z+1 &: r38 \\ mod z-1 &: r37 \end{aligned} \quad (1.4)$$

3. Reduce the convolution modulo z^2+1 using Toom-Cook factors of z, $1/z$ and $z+1$. This creates variables r35, r36, and r314.
4. Reduce the modulo z^4+1 convolution with an iterated Toom-Cook reduction using the factors z, $1/z$ and $z-1$ for the first step, and the factors z, $1/z$ and $z+1$ for the second step. The first step produces r310 and r39, and the second step computes r313, r312 and r311. This is exactly the reduction procedure used in Nussbaumer's z^4+1 convolution algorithm.
5. Patch up the DC term by adding the $z-1$ reduction result to $x(i(1))$.
6. Use Nussbaumer's z^8+1 convolution algorithm [5] on r108-r115. This is the only exception to the strict use of transposing the tensor, as his algorithm saves two additions by computing the transposed reconstruction procedure in an obscure fashion. The result, however, is an exact calculation

of the transpose. This reduction computes twenty-one values, r315-r335, which must be weighted by coefficients to produce the reconstructed z^8+1 output, t115-t135.

7. Weight the variables r31-r39, r310-r314 by coefficients to produce t11-t19, t110-t114.
8. The reconstruction procedure for the z^8-1 terms is a straightforward transpose of the reduction procedure.
9. The $z^{16}-1$ convolution result is reconstructed from the z^8-1 (real) and z^8+1 (imaginary) vectors and mapped back to the outputs using the reverse of the input map.
10. All coefficients were computed using the author's QR decomposition linear equation solver and are accurate to at least 14 places.

1.3 Length 19 Module: 372 Adds / 76 Mpys

This module closely follows the traditional Winograd prime-length approach.

1. Use the index map $\overline{x}(n) = x(<2^n>_{mod19})$ to convert the DFT into a length 18 convolution plus a correction term for the DC component.
2. Reduce the length 16 convolution modulo z^9+1 and z^9-1.

$$\begin{aligned} mod z^9 - 1 &\quad : \quad r100 - r108 \\ mod z^9 + 9 &\quad : \quad r109 - r117 \end{aligned} \tag{1.5}$$

3. Use Nussbaumer's z^9-1 convolution algorithm on r100-r108. This is a transposed tensor method, however it again uses an obscure reconstruction procedure. This algorithm computes nineteen intermediate quantities, r31-r319, which are then weighted against nineteen coefficients to produce t11-t119. This data is then partially reconstructed to yield the final result of the $modz^9-1$ convolution, t32-t310.

4. In the course of the z^9-1 convolution algorithm the z^9-1 data is reduced modulo $z-1$ and stored in r31. This quantity is added to $x(i(1))$ to patch up the DC term.
5. An algebraic trick is used to compute the z^9+1 convolution using the z^9-1 algorithm. Suppose there exists a ring homomorphism H which maps elements of the ring of real polynomials modulo z^9+1 into the ring of polynomials modulo z^9-1. Then H could be used on the z^9+1 data, the resulting polynomial could be convolved in the modulo z^9-1 domain using the existing procedure, and the output of that procedure could be mapped back through H^{-1} into the modulo z^9+1 domain. Such a homomorphism does exist, and moreover it happens to be its own inverse. $H(p)$ where p is a polynomial (in either $R[x]/z^9-1$ or $R[x]/z^9+1$) may be formed from p by negating the sign on all odd-numbered coefficients, that is, $H(p)(z) = p(-z)$. The alternate negation of data values going into and coming out of the $modz^9-1$ convolution algorithm is accomplished without an increase in computing time by appropriate placement of negative signs. The nineteen intermediate values formed are r320-r338 which are then weighted by the (purely imaginary) coefficients to produce t120-t138. A partial reconstruction yields the z^9+1 convolution result, t311-t319.
6. The z^{18-1} convolution result is reconstructed from the z^9-1 (real) and z^9+1 (imaginary) vectors and mapped back to the outputs using the reverse of the input map.
7. All coefficients were computed using the author's QR decomposition linear equation solver and are accurate to at least 14 places.

1.4 Length 25 Module: 420 Adds / 132 Mpys

This module is a common factor type module which uses length 5 convolutional DFT submodules. The length 5 submodules are implemented in a transposed tensor configuration using an index map $x(\overline{n}) = x(<2^n>_{mod5})$ followed by a reduction modulo all the irreducible factors of z^4-1. The z^2+1 convolution is implemented using Toom-Cook factors of z, $1/z$ and $z-1$. The reconstruction matrix is exactly the transpose of the reduction procedure. The coefficients for the length 5 submodules were found using the author's QR procedure, and the twiddle factors were generated in a special FORTRAN program. The details of saving multiplies by scaling some of the prime length submodules in a common factor algorithm are discussed below in Section 1.5 (Scaling in a Common Factor DFT). This length 25 module has a total of 132 multiplies and 420 adds. Using Winograd's decomposition of the length 25 OFT into two length 5 DFT's and a length 20 convolution the best operation count generated by this author was 108 multiplies and 604 adds.

1.5 Scaling in a Common Factor DFT

Scaling short length DFT algorithms can sometimes save multiplies. The prime length modules ($p > 2$) generally include one constant equal to $1/(p-1)$, corresponding to convolution modulo $x-1$ This convenient constant can in some cases be exploited. One particularly nice example is the length 25 DFT.

Use length 5 DFT modules to put together a length 25 DFT with Singleton's algorithm. This results in an algorithm which uses the length 5 module ten times, and has sixteen non-trivial twiddle factors. Counting a twiddle factor as $3/2$ multiplies, and using DFT modules with 5 multiplies, the full length 25 algorithm will have 74 multiplies.

In order to exploit the constant $1/4$ which appears in each length 5 module the basic length 5 module must be modified to create alternate modules A and B (Figure I). The regular length 5 DFT is represented as R. Algorithm A computes the same DFT, but with outputs 1 through 4 scaled up by a factor of 4. Algorithm B expects inputs 1 through 4 to be scaled down by a factor of $1/4$. Algorithms A and B have each traded 1 multiply for 2 additions. The additions are used to implement the -factor of 4 which appears in both algorithms.

To implement a scaled algorithm:

i: Assume the input data has been appropriately mapped into a 5 by 5 array.

ii: Use R on the first column of data and A on all other columns. This will scale the data in the twiddle area[2] up by a factor of 4.

iii: Scale down all twiddle factors by a factor of $1/16$. This leaves data in the twiddle area scaled down by a composite factor of $1/4$ when compared to a normal length 25 DFT.

iv: Use R on the first row of data and use B on all other rows. B is modified to expect the scaled down data in the twiddle area.

Since 4 A's and 4 B's were used, a total trade has been made of 8 multiplies for 16 adds. Such a trade may in many instances be a reasonable exchange. The great thing about this scaling is that the D.C. terms did not have to be scaled, which would have generated more adds in modification A and multiplies in modification B. No additional counter-scaling multiplies were needed in this special example because the twiddle factor were available to absorb the scaling mismatches. Similar approaches should be possible for lengths 9, 49, and 121.

The PFA case is similar in spirit, but is lacking the twiddle factors to perform counter-scaling. One of the modules will have to be modified to perform the counter-scaling function.

Two basic facts will be needed. First, any Winograd-type prime length DFT module contains one constant equal to $1/(p-1)$ and can be modified like algorithm A to scale up all of its outputs except the DC term. This modification trades one multiply for the number of adds needed to implement a multiply by $(p-1)$. Secondly, any Winograd-type prime length DFT module can be modified to scale all of its outputs by an arbitrary constant at the expense of only one multiply. This is accomplished by nesting the scaling constant with the multiplies in the middle of the Winograd module. Since only one of the module's original constants is trivial (that is the unity constant on the DC term) only one extra multiply is generated. This procedure

[2] The twiddle area is the collection of data locations which will be multiplied by non-trivial twiddles and in this instance is composed of all data which falls both in the last four columns and the last four rows of the data array.

assumes the module has first been re-arranged to eliminate the "cross" computation as illustrated in Figure II. Such a rearrangement can always be accomplished in prime length modules.

Now, suppose we combine length p and q modules with Good's prime factor algorithm (not using twiddles). The following scaling procedure will work:

i: Assume the input data has been appropriately loaded into a pxq data array

ii: Scale the non-DC outputs of the length p module and apply the modified module to all columns of the data array.

iii: Now all the rows are scaled by $(p-1)$ except the zeroeth row, corresponding to the DC outputs of the length p modules. Apply a normal length q module to the zeroeth row. Modify the length q module to scale by $1/(p-1)$ and apply the modified version to all the other rows. The DFT is now complete.

As an example, consider the 3x7 DFT. In the length 3 module scaling the non-DC outputs trades one multiply for one add. When the scaled DFT is constructed, the modified length 3 module is used 7 times. But two rows must be scaled by modified length 7 modules, which brings the total multiply savings to 5 at a cost of 7 adds. This looks like a nice tradeoff. The total number of multiplies in a normal 3x7 PFA is 38.

These ideas can be expanded to multidimensional cases, although it quickly becomes difficult to keep track of which rows and columns need to be counter-scaled.

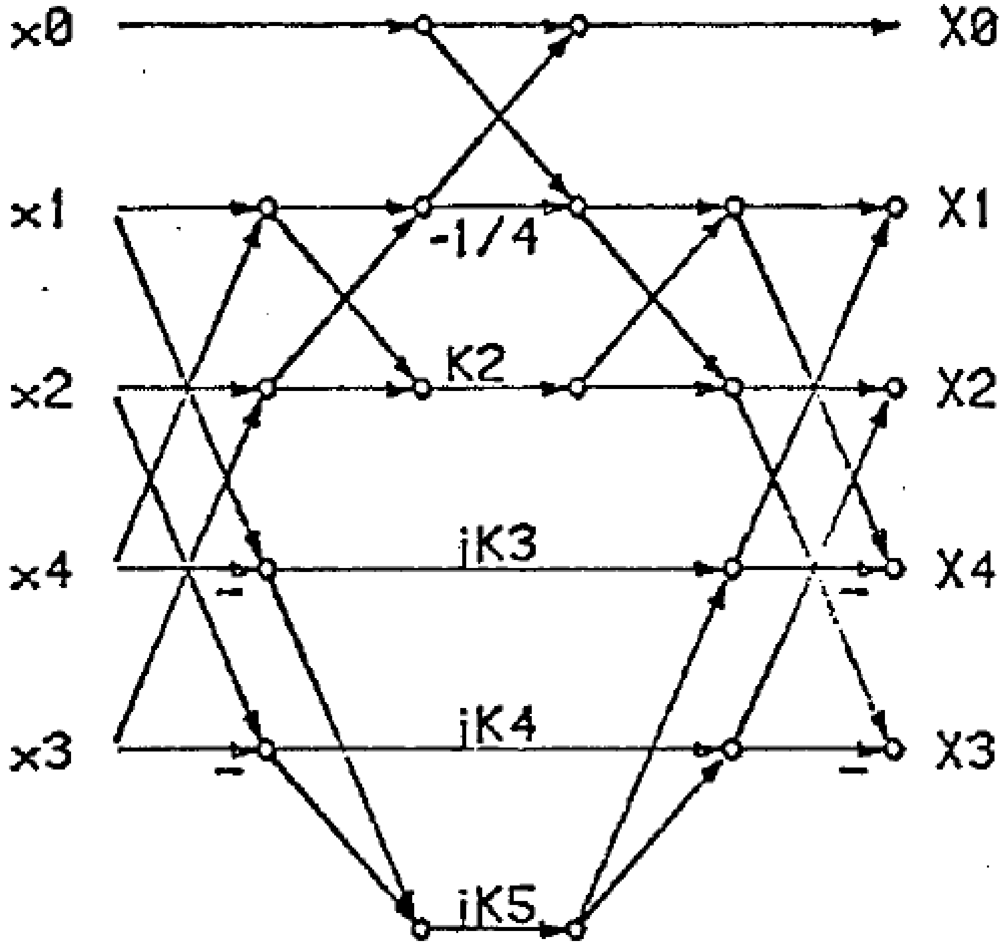

Figure 1.1: Length 5 DFT Algorithm R

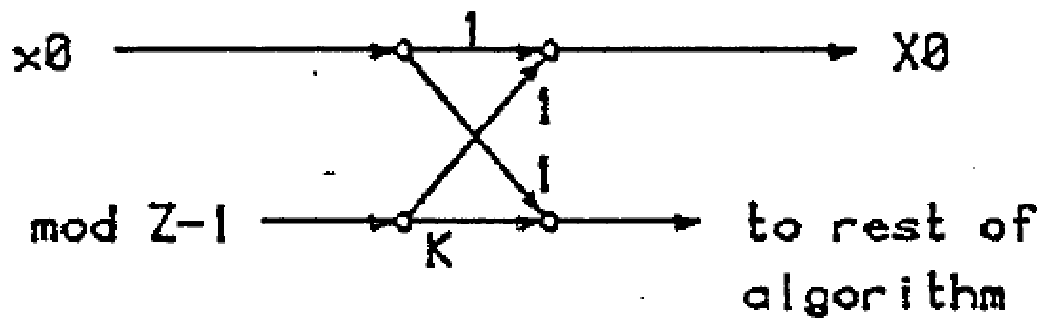

Figure 1.2: Crossed Flow Graph

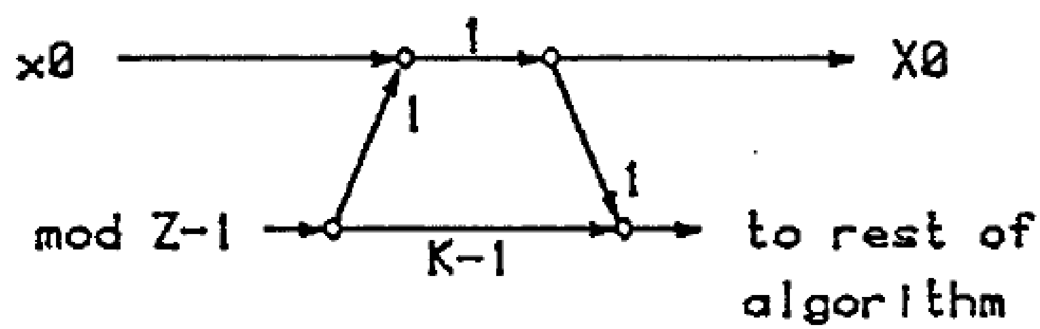

Figure 1.3: Equivalent Uncrossed Flow Graph

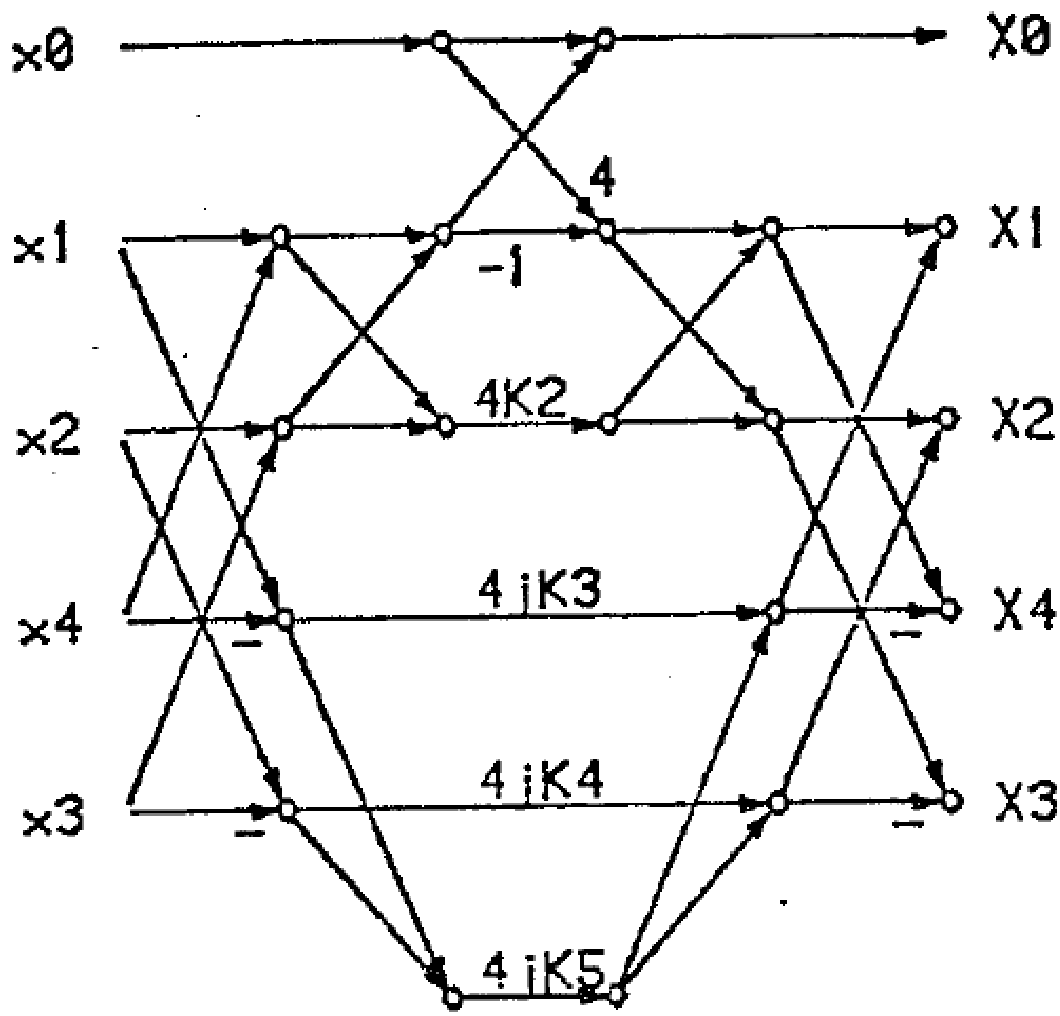

Figure 1.4: Length 5 DFT Algorithm A

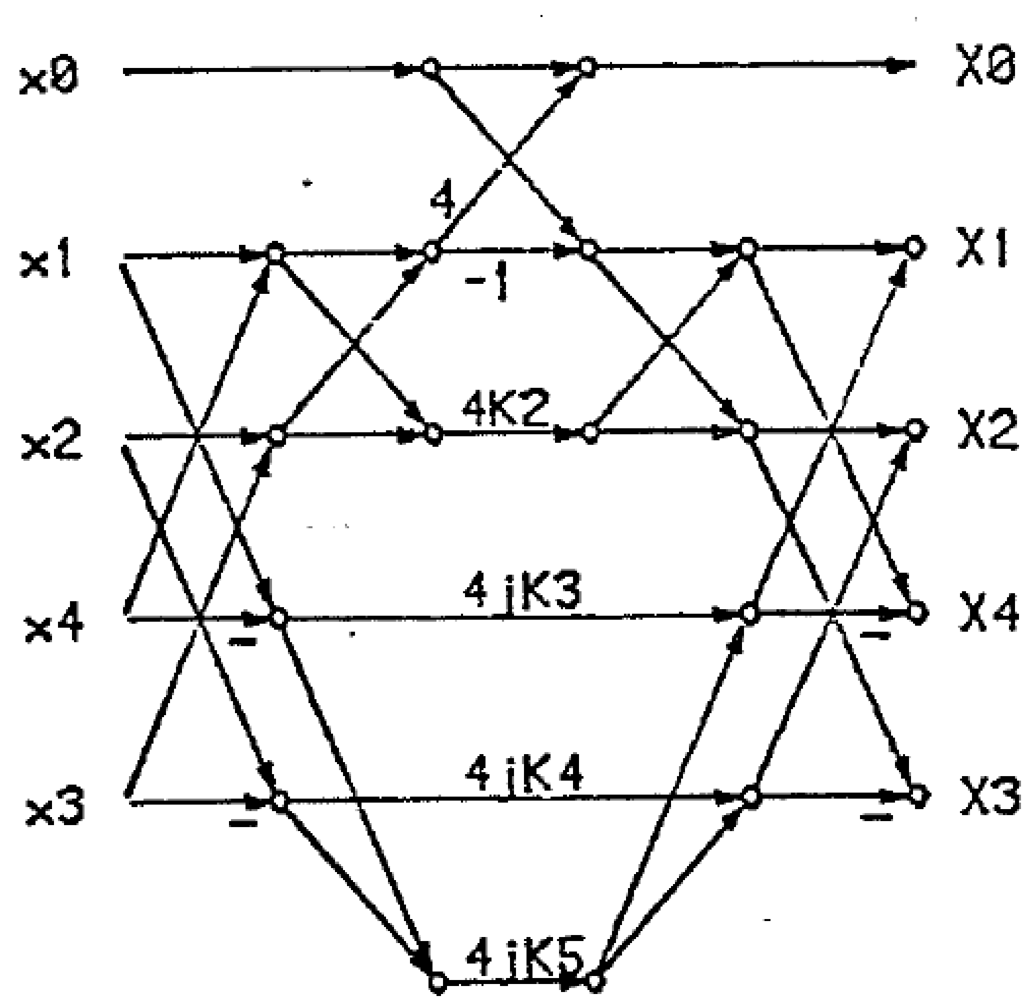

Figure 1.5: Length 5 DFT Algorithm B

1.6 Length 11 Module: 168 Adds / 40 Mpys

1. Use the index map $\overline{x}(n) = x(<8^n>_{mod11})$ to convert the DFT into a length 10 convolution, plus a correction term for the DC components.
2. Reduce the length 10 convolution modulo all the irreducible factors of $z^{10}-1$

$$\begin{array}{rcc} modz^5-1 & : & T1, T3, T2, T5, T4 \\ modz^5+1 & : & T6, -T8, -T7, -T10, T9 \end{array} \quad (1.6)$$

from $z^5 - 1$ data

$$\begin{array}{rcc} mod z - 1 & : & T13 \\ mod z^5 - 1/z - 1 & : & AM4, AM7, AM3, AM6\,(after weighting) \end{array} \quad (1.7)$$

from $z^5 + 1$ data

$$\begin{array}{rcc} mod z + 1 & : & AM2\,(after weighting) \\ mod z^5 + 1/z + 1 & : & S9, S11, S10, S12\,(appears in) \end{array} \quad (1.8)$$

3. Patch up the DC terms by adding the $z-1$ reduction result to $X(I(1))$ and store the result in AMO.
4. The $z^5 - 1$ convolution proceeds in four steps. First, do the irreducible factor reductions, then reduce further with an iterated Toom-Cook procedure, weight all remaining variables, and apply the transpose of the complete reduction stage to the weighted results. The first Toom-Cook reduction uses the factors z, $1/z$ and $z+1$ on the vectors AM4,AM3 and AM7,AM6 which generates the new vector AM4-AM7,AM3-AM6. Each of the original two vectors is then individually reduced using factors of z, $1/z$ and $z+1$, while the new vector is reduced by A, $1/z$ and $z-1$. This procedure generates nine variables: AM4,AM3,AM5; AM7,AM6,AM8; S7,S8,AM11. (The expressions for S6 and S8 contain the variables of interest).
5. The nine variables from 4) are weighted along with T13.
6. An exact transpose of the reduction algorithm is applied to the weighted variables (and AMO).
7. The result S16,S15,S18,S17,S19 is the real part of the answer and is mapped back to the output using the map $\overline{x}(n) = x\left(< 8^{n+1} > mod 11\right.$. This is an unusual map, but it is perfectly acceptable.
8. A in the length 19 transform the z^5+1 convolution is computed with a variation of the z^5-1 algorithm. First the inputs T6,-T8,-T7,-T1O,T9 are alternately negated, then the $z^5 - 1$ algorithm is applied[3] and the outputs alternately negated.
9. The result S21,S20,S23,S22,S24, representing the imaginary part of the answer, is mapped back to the output using the map $\overline{x}(n) = x\left(< 8^{n+1} > mod 11\right)$.
10. In both this algorithm and the length 13 DFT plus and minus signs have been freely altered to force all constants to be positive. Also, many shortcut computations were used to save adds, obscuring in some places the logical flow of the algorithm.
11. All coefficients were computed using the author's QR decomposition linear equation solver and are accurate to at least 14 places.

1.7 Length 13 Module: 188 Adds / 40 Mpys

1. Use the index map $\overline{x}(n) = x(< 2^n >_{mod 13})$ to convert the DFT into a length 12 convolution, plus a correction term for the DC components.
2. Reduce the length 12 convolution modulo all the irreducible factors of $z^{12} - 1$

$$\begin{array}{rcc} mod z^6 + 1 & : & A7, A8, A9, A10, A11, A12 \\ mod z^6 - 1 & : & A1, A2, A3, A4, A5, A6 \end{array} \quad (1.9)$$

from $z^6 - 1$ data

$$\begin{array}{rcl} mod z^2 - 1 & : & A14, A13 \\ mod z^2 - z + 1 & : & A23, A22 \\ mod z^2 + z + 1 & : & A25, A24 \end{array} \quad (1.10)$$

[3]The second stage of the Toom-Cook reductions uses the factors z, liz and z+l for all three length two vectors. Also, the DC patch is not used here.

from $z^2 - 1$ data

$$\begin{aligned} modz-1 &: \quad A15 \\ modz+1 &: \quad implicit\,(A13-A14) \end{aligned} \tag{1.11}$$

from $z^6 + 1$ data

$$\begin{aligned} modz^2+1 &: \quad A17, A16 \\ modz^4-z^2+1 &: \quad A27, A26, -A31, -A30 \end{aligned} \tag{1.12}$$

3. Patch up the DC terms by adding the $z-1$ reduction result to $X\,(I\,(1))$ and store the result in AMO.
4. The $z^2 - z + 1$ and $z^2 + z + 1$ convolutions are reduced using Toom-cook factors of z, $1/z$ and $z+1$ in one case and z, $1/z$ and $z-1$ in the other case, and then all the reduced quantities are weighted by constants generating new variables: from $z^2 - z + 1$

$$\begin{aligned} z &\quad AM7 \\ 1/z &\quad AM6 \\ z-1 &\quad AM8 \end{aligned} \tag{1.13}$$

from $z^2 + z + 1$

$$\begin{aligned} z &\quad AM10 \\ 1/z &\quad AM9 \\ z+1 &\quad AM11 \end{aligned} \tag{1.14}$$

5. The original $modz+1$ reduction quantity is weighted and passed, along with AMO and the above six variables, to a reconstruction procedure which first combines the $z-1$ and z^2+z+1 data to compute the convolution mod z^3-1 (CC4,CC5,CC6), and then combines the $z+1$ and z^2-z+1 data to compute the convolution mod z^3+1 (CC1,CC2,CC3). These two vectors are combined to compute the complete z^6-1 output, which appears in permuted form in CC15 through CC20.
6. The z^2+1 vector is decomposed with Toom-Cook factors of z, $1/z$ and $z+1$ yielding A17,A16 and the implicit term (A16+A17).
7. The $z^4 - z^2 + 1$ vector is decomposed with a double iterated Toom-Cook scheme. First the vector is broken into two length two pieces: A27,A26 and A31,A30. Then the vectors are reduced by the factors of z, $1/z$ and $z+1$ operating on whole vectors to produce a set of three length two vectors: A27,A26 A31,A30 A29,A28 = (A27+A31), (A26+A30) These vectors are not calculated in a straightforward manner. Each length two vector is further reduced, in the second iteration, by the factors z, $1/z$ and $z+1$ to create three new implicit variables $(A27+A26)$, $(A31+A30)$ and $(A29+A28)$.
8. The nine variables from Section 1.6 (Length 11 Module: 168 Adds / 40 Mpys) and the three variables from Section 1.5 (Scaling in a Common Factor DFT) are weighted by constants and the $modz^6+1$ reconstruction proceeds in an ad-hoc fashion which closely resembles a transposed tensor method, but has some differences. The add count for the reconstruction would have been the same if the transposed tensor method had been applied. The z^6+1 result appears in permuted form in variables CC21 through CC26.
9. The final result is reconstructed from the z^6-1 and z^6-1 vectors. The DC term, $x\,(i\,(1))$ is set equal' to AMO.
10. All coefficients were computed using the author's QR decomposition linear equation solver and are accurate to at least 14 places.

Chapter 2

N = 11 Winograd FFT module[1]

2.1 N=11 FFT module

A FORTRAN implementation of a length-11 FFT module to be used in a Prime Factor Algorithm program.

```
    C
    DATA  C111,C112 /  1.10000000, 0.33166250 /
    DATA  C113,C114 /  0.51541500, 0.94125350 /
    DATA  C115,C116 /  1.41435370, 0.85949300 /
    DATA  C117,C118 /  0.04231480, 0.38639280 /
    DATA  C119,C1110/  0.51254590, 1.07027569 /
    DATA  C1111,C1112/ 0.55486070, 1.24129440 /
    DATA  C1113,C1114/ 0.20897830, 0.37415717 /
    DATA  C1115,C1116/ 0.04992992, 0.65815896 /
    DATA  C1117,C1118/ 0.63306543, 1.08224607 /
    DATA  C1119,C1120/ 0.81720738, 0.42408709 /
C
C-----------------WFTA N=11----------------------------
C
  111   T1 = X(I(2)) + X(I(11))
    T6 = X(I(2)) - X(I(11))
    T2 = X(I(3)) + X(I(10))
    T7 = X(I(3)) - X(I(10))
    T3 = X(I(4)) + X(I(9))
    T8 = X(I(4)) - X(I(9))
    T4 = X(I(5)) + X(I(8))
    T9 = X(I(5)) - X(I(8))
    T5 = X(I(6)) + X(I(7))
    T10= X(I(6)) - X(I(7))
C
    U1 = Y(I(2)) + Y(I(11))
    U6 = Y(I(2)) - Y(I(11))
    U2 = Y(I(3)) + Y(I(10))
    U7 = Y(I(3)) - Y(I(10))
    U3 = Y(I(4)) + Y(I(9))
    U8 = Y(I(4)) - Y(I(9))
```

[1] This content is available online at <http://cnx.org/content/m17377/1.10/>.

```
      U4 = Y(I(5)) + Y(I(8))
      U9 = Y(I(5)) - Y(I(8))
      U5 = Y(I(6)) + Y(I(7))
      U10= Y(I(6)) - Y(I(7))
C
      T11 = T1 + T2
      T12 = T3 + T5
      T13 = T4 + T11 + T12
      T14 = T7 - T8
      T15 = T6 + T10
C
      U11 = U1 + U2
      U12 = U3 + U5
      U13 = U4 + U11 + U12
      U14 = U7 - U8
      U15 = U6 + U10
C
      AM0 = X(I(1)) + T13
      AM2 = (T14 - T15 - T9) * C112
      AM3 = (T2 - T4) * C113
      AM4 = (T1 - T4) * C114
      AM5 = (T2 - T1) * C115
      AM6 = (T5 - T4) * C116
      AM7 = (T3 - T4) * C117
      AM8 = (T5 - T3) * C118
      AM11 = (T12 - T11) * C1111
      AM14 = (T6 + T7) * C1114
      AM17 = (T8 - T10) * C1117
      AM20 = (T14 + T15) * C1120
C
      AN0 = Y(I(1)) + U13
      AN2 = (U14 - U15 - U9) * C112
      AN3 = (U2 - U4) * C113
      AN4 = (U1 - U4) * C114
      AN5 = (U2 - U1) * C115
      AN6 = (U5 - U4) * C116
      AN7 = (U3 - U4) * C117
      AN8 = (U5 - U3) * C118
      AN11 = (U12 - U11) * C1111
      AN14 = (U6 + U7) * C1114
      AN17 = (U8 - U10) * C1117
      AN20 = (U14 + U15) * C1120
C
      S0 = AM0 - C111 * T13
      S7 = AM11 + C1110 * (T1 - T3)
      S8 = AM11 + (T2 - T5) * C119
      S9 = AM14 + (T6 - T9) * C1113
      S10 =-AM14 + (T7 + T9) * C1112
      S11 = AM17 + (T8 - T9) * C1116
      S12 =-AM17 + (T9 - T10) * C1115
      S13 = AM20 + (T6 - T8) * C1119
```

```
      S14 =-AM20 + (T7 + T10) * C1118
C
      V0 = AN0 - C111 * U13
      V7 = AN11 + C1110 * (U1 - U3)
      V8 = AN11 + (U2 - U5) * C119
      V9 = AN14 + (U6 - U9) * C1113
      V10 =-AN14 + (U7 + U9) * C1112
      V11 = AN17 + (U8 - U9) * C1116
      V12 =-AN17 + (U9 - U10) * C1115
      V13 = AN20 + (U6 - U8) * C1119
      V14 =-AN20 + (U7 + U10) * C1118
C
      S15 = S0 + S7 + AM7 + AM8
      S16 = S0 - S7 - AM4 - AM5
      S17 = S0 + S8 + AM6 - AM8
      S18 = S0 - S8 - AM3 + AM5
      S19 = S0 + AM3 + AM4 - AM6 - AM7
      S20 = S13 + AM2 + S11
      S21 = S13 - AM2 - S9
      S22 = S14 + AM2 + S12
      S23 = S14 - AM2 - S10
      S24 = S9 + S10 + S11 + S12 - AM2
C
      V15 = V0 + V7 + AN7 + AN8
      V16 = V0 - V7 - AN4 - AN5
      V17 = V0 + V8 + AN6 - AN8
      V18 = V0 - V8 - AN3 + AN5
      V19 = V0 + AN3 + AN4 - AN6 - AN7
      V20 = V13 + AN2 + V11
      V21 = V13 - AN2 - V9
      V22 = V14 + AN2 + V12
      V23 = V14 - AN2 - V10
      V24 = V9 + V10 + V11 + V12 - AN2
C
      X(I(1)) = AM0
      X(I(2)) = S19 + V24
      X(I(3)) = S15 + V20
      X(I(4)) = S16 + V21
      X(I(5)) = S17 - V22
      X(I(6)) = S18 + V23
      X(I(7)) = S18 - V23
      X(I(8)) = S17 + V22
      X(I(9)) = S16 - V21
      X(I(10))= S15 - V20
      X(I(11))= S19 - V24
C
      Y(I(1)) = AN0
      Y(I(2)) = V19 - S24
      Y(I(3)) = V15 - S20
      Y(I(4)) = V16 - S21
      Y(I(5)) = V17 + S22
```

```
      Y(I(6)) = V18 - S23
      Y(I(7)) = V18 + S23
      Y(I(8)) = V17 - S22
      Y(I(9)) = V16 + S21
      Y(I(10))= V15 + S20
      Y(I(11))= V19 + S24
C
      GOTO 20
C
```

Figure. Length-11 FFT Module

Chapter 3

N = 13 Winograd FFT module[1]

3.1 N=13 FFT module

A FORTRAN implementation of a length-13 FFT module to be used in a Prime Factor Algorithm program.

```
    C
    DATA  C131, C132 / 1.08333333, 0.30046261 /
    DATA  C133, C134 / 0.74927933, 0.40113213 /
    DATA  C135, C136 / 0.57514073, 0.52422664 /
    DATA  C137, C138 / 0.51652078, 0.00770586 /
    DATA  C139, C1310/ 0.42763400, 0.15180600 /
    DATA  C1311,C1312/ 0.57944000, 1.15439534 /
    DATA  C1313,C1314/ 0.90655220, 0.81857027 /
    DATA  C1315,C1316/ 1.19713677, 0.86131171 /
    DATA  C1317,C1318/ 1.10915484, 0.04274143 /
    DATA  C1319,C1320/ 0.04524049, 0.29058457 /
C
C------------------WFTA N=13--------------------------------
C
  113   A1  = X(I(2)) + X(I(13))
    A2  = X(I(3)) + X(I(12))
    A3  = X(I(4)) + X(I(11))
    A4  = X(I(5)) + X(I(10))
    A5  = X(I(6)) + X(I(9))
    A6  = X(I(7)) + X(I(8))
    A7  = X(I(2)) - X(I(13))
    A8  = X(I(3)) - X(I(12))
    A9  = X(I(4)) - X(I(11))
    A10 = X(I(5)) - X(I(10))
    A11 = X(I(6)) - X(I(9))
    A12 = X(I(7)) - X(I(8))
    B1  = Y(I(2)) + Y(I(13))
    B2  = Y(I(3)) + Y(I(12))
    B3  = Y(I(4)) + Y(I(11))
    B4  = Y(I(5)) + Y(I(10))
    B5  = Y(I(6)) + Y(I(9))
```

[1]This content is available online at <http://cnx.org/content/m17378/1.7/>.

```
B6  = Y(I(7)) + Y(I(8))
B7  = Y(I(2)) - Y(I(13))
B8  = Y(I(3)) - Y(I(12))
B9  = Y(I(4)) - Y(I(11))
B10 = Y(I(5)) - Y(I(10))
B11 = Y(I(6)) - Y(I(9))
B12 = Y(I(7)) - Y(I(8))
A13 = A2 + A5 + A6
A14 = A1 + A3 + A4
A15 = A13 + A14
A16 = A8 + A11 + A12
A17 = A7 + A9 - A10
A18 = A2 - A6
A19 = A3 - A4
A20 = A1 - A4
A21 = A5 - A6
A22 = A18 - A19
A23 = A20 - A21
A24 = A18 + A19
A25 = A20 + A21
A26 = A8 - A12
A27 = A7 - A9
A28 = A8 - A11
A29 = A7 + A10
A30 = A11 - A12
A31 =-A9 - A10
B13 = B2 + B5 + B6
B14 = B1 + B3 + B4
B15 = B13 + B14
B16 = B8 + B11 + B12
B17 = B7 + B9 - B10
B18 = B2 - B6
B19 = B3 - B4
B20 = B1 - B4
B21 = B5 - B6
B22 = B18 - B19
B23 = B20 - B21
B24 = B18 + B19
B25 = B20 + B21
B26 = B8 - B12
B27 = B7 - B9
B28 = B8 - B11
B29 = B7 + B10
B30 = B11 - B12
B31 =-B9 - B10
AM0  = X(I(1)) + A15
AM2  = (A13 - A14) * C132
AM5  = (A16 + A17) * C135
AM6  =  A22 * C136
AM7  =  A23 * C137
AM8  = (A22 + A23) * C138
```

```
AM9  =  A24 * C139
AM10 =  A25 * C1310
AM11 = (A24 - A25) * C1311
AM14 = (A26 + A27) * C1314
AM17 = (A28 + A29) * C1317
AM20 = (A30 + A31) * C1320
BM0  = Y(I(1)) + B15
BM2  = (B13 - B14) * C132
BM5  = (B16 + B17) * C135
BM6  =  B22 * C136
BM7  =  B23 * C137
BM8  = (B22 + B23) * C138
BM9  =  B24 * C139
BM10 =  B25 * C1310
BM11 = (B24 - B25) * C1311
BM14 = (B26 + B27) * C1314
BM17 = (B28 + B29) * C1317
BM20 = (B30 + B31) * C1320
CC0  = AM0 - A15 * C131
CC1  = AM7 + AM6 - AM2
CC2  = AM7 + AM8 + AM2
CC3  = AM8 - AM6 - AM2
CC4  = CC0 + AM9 + AM10
CC5  = CC0 - AM10 - AM11
CC6  = CC0 - AM9 + AM11
CC7  = AM14 - A26 * C1312
CC8  = AM14 - A27 * C1313
CC9  = -AM17 + A28 * C1315
CC10 = -AM17 + A29 * C1316
CC11 =  AM20 - A30 * C1318
CC12 =  AM20 + A31 * C1319
CC13 = -AM5  + A16 * C133
CC14 = -AM5  + A17 * C134
CC15 = CC1 + CC4
CC16 = CC2 + CC5
CC17 = CC5 - CC2
CC18 = CC3 + CC6
CC19 = CC4 - CC1
CC20 = CC6 - CC3
CC21 = CC14 + CC7 + CC9
CC22 = CC10 - CC12 + CC13
CC23 =-CC7 - CC11 + CC14
CC24 = CC9 - CC11 - CC14
CC25 = CC8 + CC12 + CC13
CC26 = CC13 - CC8 - CC10
DD0  = BM0 - B15 * C131
DD1  = BM7 + BM6 - BM2
DD2  = BM7 + BM8 + BM2
DD3  = BM8 - BM6 - BM2
DD4  = DD0 + BM9 + BM10
DD5  = DD0 - BM10 - BM11
```

```
      DD6  = DD0 - BM9 + BM11
      DD7  = BM14 - B26 * C1312
      DD8  = BM14 - B27 * C1313
      DD9  = -BM17 + B28 * C1315
      DD10 = -BM17 + B29 * C1316
      DD11 =  BM20 - B30 * C1318
      DD12 =  BM20 + B31 * C1319
      DD13 = -BM5  + B16 * C133
      DD14 = -BM5  + B17 * C134
      DD15 = DD1 + DD4
      DD16 = DD2 + DD5
      DD17 = DD5 - DD2
      DD18 = DD3 + DD6
      DD19 = DD4 - DD1
      DD20 = DD6 - DD3
      DD21 = DD14 + DD7 + DD9
      DD22 = DD10 - DD12 + DD13
      DD23 =-DD7 - DD11 + DD14
      DD24 = DD9 - DD11 - DD14
      DD25 = DD8 + DD12 + DD13
      DD26 = DD13 - DD8 - DD10
      X(I(1))  = AM0
      X(I(2))  = CC15 - DD21
      X(I(3))  = CC16 - DD22
      X(I(4))  = CC17 - DD23
      X(I(5))  = CC18 - DD24
      X(I(6))  = CC19 - DD25
      X(I(7))  = CC20 - DD26
      X(I(8))  = CC20 + DD26
      X(I(9))  = CC19 + DD25
      X(I(10)) = CC18 + DD24
      X(I(11)) = CC17 + DD23
      X(I(12)) = CC16 + DD22
      X(I(13)) = CC15 + DD21
      Y(I(1))  = BM0
      Y(I(2))  = CC21 + DD15
      Y(I(3))  = CC22 + DD16
      Y(I(4))  = CC23 + DD17
      Y(I(5))  = CC24 + DD18
      Y(I(6))  = CC25 + DD19
      Y(I(7))  = CC26 + DD20
      Y(I(8))  =-CC26 + DD20
      Y(I(9))  =-CC25 + DD19
      Y(I(10)) =-CC24 + DD18
      Y(I(11)) =-CC23 + DD17
      Y(I(12)) =-CC22 + DD16
      Y(I(13)) =-CC21 + DD15
C
      GOTO 20
C
```

Figure: Length-13 FFT Module

Chapter 4

N = 16 FFT module[1]

4.1 N=16 FFT module

A FORTRAN implementation of a length-16 FFT module to be used in a Prime Factor Algorithm program.

```
    C
C--------------------WFTA N=16----------------------------
C
  116   R1 = X(I(1)) + X(I(9))
    R2 = X(I(1)) - X(I(9))
    S1 = Y(I(1)) + Y(I(9))
    S2 = Y(I(1)) - Y(I(9))
    R3 = X(I(2)) + X(I(10))
    R4 = X(I(2)) - X(I(10))
    S3 = Y(I(2)) + Y(I(10))
    S4 = Y(I(2)) - Y(I(10))
    R5 = X(I(3)) + X(I(11))
    R6 = X(I(3)) - X(I(11))
    S5 = Y(I(3)) + Y(I(11))
    S6 = Y(I(3)) - Y(I(11))
    R7 = X(I(4)) + X(I(12))
    R8 = X(I(4)) - X(I(12))
    S7 = Y(I(4)) + Y(I(12))
    S8 = Y(I(4)) - Y(I(12))
    R9 = X(I(5)) + X(I(13))
    R10= X(I(5)) - X(I(13))
    S9 = Y(I(5)) + Y(I(13))
    S10= Y(I(5)) - Y(I(13))
    R11 = X(I(6)) + X(I(14))
    R12 = X(I(6)) - X(I(14))
    S11 = Y(I(6)) + Y(I(14))
    S12 = Y(I(6)) - Y(I(14))
    R13 = X(I(7)) + X(I(15))
    R14 = X(I(7)) - X(I(15))
    S13 = Y(I(7)) + Y(I(15))
    S14 = Y(I(7)) - Y(I(15))
```

[1] This content is available online at <http://cnx.org/content/m17382/1.5/>.

```
R15 = X(I(8)) + X(I(16))
R16 = X(I(8)) - X(I(16))
S15 = Y(I(8)) + Y(I(16))
S16 = Y(I(8)) - Y(I(16))
T1 = R1 + R9
T2 = R1 - R9
U1 = S1 + S9
U2 = S1 - S9
T3 = R3 + R11
T4 = R3 - R11
U3 = S3 + S11
U4 = S3 - S11
T5 = R5 + R13
T6 = R5 - R13
U5 = S5 + S13
U6 = S5 - S13
T7 = R7 + R15
T8 = R7 - R15
U7 = S7 + S15
U8 = S7 - S15
T9 = C81 * (T4 + T8)
T10= C81 * (T4 - T8)
U9 = C81 * (U4 + U8)
U10= C81 * (U4 - U8)
R1 = T1 + T5
R3 = T1 - T5
S1 = U1 + U5
S3 = U1 - U5
R5 = T3 + T7
R7 = T3 - T7
S5 = U3 + U7
S7 = U3 - U7
R9 = T2 + T10
R11= T2 - T10
S9 = U2 + U10
S11= U2 - U10
R13 = T6 + T9
R15 = T6 - T9
S13 = U6 + U9
S15 = U6 - U9
T1 = R4 + R16
T2 = R4 - R16
U1 = S4 + S16
U2 = S4 - S16
T3 = C81 * (R6 + R14)
T4 = C81 * (R6 - R14)
U3 = C81 * (S6 + S14)
U4 = C81 * (S6 - S14)
T5 = R8 + R12
T6 = R8 - R12
U5 = S8 + S12
```

```
      U6 = S8 - S12
      T7 = C162 * (T2 - T6)
      T8 = C163 * T2 - T7
      T9 = C164 * T6 - T7
      T10 = R2 + T4
      T11 = R2 - T4
      R2 = T10 + T8
      R4 = T10 - T8
      R6 = T11 + T9
      R8 = T11 - T9
      U7 = C162 * (U2 - U6)
      U8 = C163 * U2 - U7
      U9 = C164 * U6 - U7
      U10 = S2 + U4
      U11 = S2 - U4
      S2 = U10 + U8
      S4 = U10 - U8
      S6 = U11 + U9
      S8 = U11 - U9
      T7 = C165 * (T1 + T5)
      T8 = T7 - C164 * T1
      T9 = T7 - C163 * T5
      T10 = R10 + T3
      T11 = R10 - T3
      R10 = T10 + T8
      R12 = T10 - T8
      R14 = T11 + T9
      R16 = T11 - T9
      U7 = C165 * (U1 + U5)
      U8 = U7 - C164 * U1
      U9 = U7 - C163 * U5
      U10 = S10 + U3
      U11 = S10 - U3
      S10 = U10 + U8
      S12 = U10 - U8
      S14 = U11 + U9
      S16 = U11 - U9
C
      X(I( 1)) = R1 + R5
      X(I( 9)) = R1 - R5
      Y(I( 1)) = S1 + S5
      Y(I( 9)) = S1 - S5
      X(I( 2)) = R2 + S10
      X(I(16)) = R2 - S10
      Y(I( 2)) = S2 - R10
      Y(I(16)) = S2 + R10
      X(I( 3)) = R9 + S13
      X(I(15)) = R9 - S13
      Y(I( 3)) = S9 - R13
      Y(I(15)) = S9 + R13
      X(I( 4)) = R8 - S16
```

```
      X(I(14)) = R8 + S16
      Y(I( 4)) = S8 + R16
      Y(I(14)) = S8 - R16
      X(I( 5)) = R3 + S7
      X(I(13)) = R3 - S7
      Y(I( 5)) = S3 - R7
      Y(I(13)) = S3 + R7
      X(I( 6)) = R6 + S14
      X(I(12)) = R6 - S14
      Y(I( 6)) = S6 - R14
      Y(I(12)) = S6 + R14
      X(I( 7)) = R11 - S15
      X(I(11)) = R11 + S15
      Y(I( 7)) = S11 + R15
      Y(I(11)) = S11 - R15
      X(I( 8)) = R4 - S12
      X(I(10)) = R4 + S12
      Y(I( 8)) = S4 + R12
      Y(I(10)) = S4 - R12
C
      GOTO 20
C
```

Figure. Length-16 FFT Module

Chapter 5

N = 17 Winograd FFT module[1]

5.1 N=17 FFT module

A FORTRAN implementation of a length-17 FFT module to be used in a Prime Factor Algorithm program. Errors discovered by Yuri Reznik have been corrected (8/17/11).

```
    C
C-------------------WFTA N=17----------------------------
C
C 314 ADDS; 70 MPYS
C DATA FOR LENGTH 17 DFT
DATA C1701  /    -.0426028491177360 /
DATA C1702  /     .2049796502326218 /
DATA C1703  /    1.0451835201736758 /
DATA C1704  /    1.7645848660222969 /
DATA C1705  /    -.7234079772860566 /
DATA C1706  /    -.0890555916206064 /
DATA C1707  /   -1.0625000000000000 /
DATA C1708  /     .2576941016011038 /
DATA C1709  /     .7798026078948376 /
DATA C1710  /     .5438931846457058 /
DATA C1711  /     .4201019349705270 /
DATA C1712  /    1.2810929434228074 /
DATA C1713  /     .4408890734817534 /
DATA C1714  /     .3171761928327251 /
DATA C1715  /    -.9013831864801668 /
DATA C1716  /    -.4324875636007231 /
DATA C1717  /     .6669353750404450 /
DATA C1718  /    -.6038900431251697 /
DATA C1719  /    -.3692487319858255 /
DATA C1720  /     .4865693875554976 /
DATA C1721  /     .2381371213676061 /
DATA C1722  /   -1.5573820617422459 /
DATA C1723  /     .6596224701873199 /
DATA C1724  /    -.1431696156986624 /
DATA C1725  /     .2390346995986077 /
```

[1]This content is available online at <http://cnx.org/content/m17380/1.10/>.

```
DATA C1726  /    -.0479325419499726 /
DATA C1727  /   -2.3188014856550064 /
DATA C1728  /     .7891456841920625 /
DATA C1729  /    3.8484572871179504 /
DATA C1730  /   -1.3003804568801376 /
DATA C1731  /    4.0814769046889033 /
DATA C1732  /   -1.4807159909286282 /
DATA C1733  /    -.0133324703635514 /
DATA C1734  /    -.3713977869055763 /
DATA C1735  /     .1923651286345638 /
C
C-----------------WFTA N=17-------------------------
C
R100=X(I(2))+X(I(17))
R108=X(I(2))-X(I(17))
R101=X(I(4))+X(I(15))
R109=X(I(4))-X(I(15))
R102=X(I(10))+X(I(9))
R110=X(I(10))-X(I(9))
R103=X(I(11))+X(I(8))
R111=X(I(11))-X(I(8))
R104=X(I(14))+X(I(5))
R112=X(I(14))-X(I(5))
R105=X(I(6))+X(I(13))
R113=X(I(6))-X(I(13))
R106=X(I(16))+X(I(3))
R114=X(I(16))-X(I(3))
R107=X(I(12))+X(I(7))
R115=X(I(12))-X(I(7))
R200=R100+R104
R201=R101+R105
R202=R102+R106
R203=R103+R107
R204=R200+R202
R205=R201+R203
R31=R100-R104
R32=R101-R105
R33=R102-R106
R34=R103-R107
R35=R200-R202
R36=R201-R203
R37=R204+R205
R38=R204-R205
R39=R32+R34
R310=R31+R33
R311=R310-R39
R312=R33-R34
R313=R31-R32
R314=R35+R36
R210=R108+R110
R211=R109+R111
```

```
R212=R108-R110
R213=R115-R113
R214=R112+R114
R215=R113+R115
R216=R112-R114
R217=R109-R111
R315=R210+R211
R316=R214+R215
R317=R315+R316
R318=R210-R211
R319=R214-R215
R320=R318+R319
R321=R212+R213
R322=R216+R217
R323=R321+R322
R324=R212-R213
R325=R216-R217
R326=R324+R325
R327=R108+R112
R328=R108
R329=R112
R330=R111+R115
R331=R111
R332=R115
R333=R322-R316+R108-R330
R334=R315-R321+R111+R112-R115
R335=R333+R334
X(I(1))=X(I(1))+R37
T11=R31*C1701
T12=R32*C1702
T13=R33*C1703
T14=R34*C1704
T15=R35*C1705
T16=R36*C1706
T17=R37*C1707
T18=R38*C1708
T19=R39*C1709
T110=R310*C1710
T111=R311*C1711
T112=R312*C1712
T113=R313*C1713
T114=R314*C1714
T115=R315*C1715
T116=R316*C1716
T117=R317*C1717
T118=R318*C1718
T119=R319*C1719
T120=R320*C1720
T121=R321*C1721
T122=R322*C1722
T123=R323*C1723
```

```
T124=R324*C1724
T125=R325*C1725
T126=R326*C1726
T127=R327*C1727
T128=R328*C1728
T129=R329*C1729
T130=R330*C1730
T131=R331*C1731
T132=R332*C1732
T133=R333*C1733
T134=R334*C1734
T135=R335*C1735
T17=T17+X(I(1))
T200=T19+T111
T201=T110-T111
T202=T14+T112
T203=T112-T13
T204=T12+T113
T205=T11-T113
T206=T114-T16
T207=T114+T15
T208=T18+T17
T209=T17-T18
T210=T200-T202
T211=T206+T208
T212=T201+T203
T213=T207+T209
T214=T200+T204
T215=-T206+T208
T216=T201+T205
T217=-T207+T209
T32=T210+T211
T37=T212+T213
T33=T214+T215
T36=T216+T217
T35=-T210+T211
T38=-T212+T213
T39=-T214+T215
T34=-T216+T217
T220=T115+T117
T221=T116+T117
T222=T118+T120
T223=T119+T120
T224=T121+T123
T225=T122+T123
T226=T124+T126
T227=T125+T126
T228=T135+T134
T229=T127+T228
T230=T229+T128
T231=T220+T222
```

```
T232=T220-T222
T233=T221+T223
T234=T221-T223
T235=T224+T226
T236=T224-T226
T237=T225+T227
T238=T225-T227
T239=T133-T134
T240=T229+T129
T241=T239+T239
T242=T130-T241
T243=T242+T131
T244=-T242-T132
T245=T228+T228
T246=T245+T245
T247=T239+T245
T310=T233+T237+T240
T315=T232-T238+T243
T311=T231-T235+T245
T314=-T232-T238-T247
T313=T231+T235+T230+T239
T316=-T234-T236+T244+T246
T317=-T233+T237+T241+T245
T312=T234-T236-T239
S100=Y(I(2))+Y(I(17))
S108=Y(I(2))-Y(I(17))
S101=Y(I(4))+Y(I(15))
S109=Y(I(4))-Y(I(15))
S102=Y(I(10))+Y(I(9))
S110=Y(I(10))-Y(I(9))
S103=Y(I(11))+Y(I(8))
S111=Y(I(11))-Y(I(8))
S104=Y(I(14))+Y(I(5))
S112=Y(I(14))-Y(I(5))
S105=Y(I(6))+Y(I(13))
S113=Y(I(6))-Y(I(13))
S106=Y(I(16))+Y(I(3))
S114=Y(I(16))-Y(I(3))
S107=Y(I(12))+Y(I(7))
S115=Y(I(12))-Y(I(7))
S200=S100+S104
S201=S101+S105
S202=S102+S106
S203=S103+S107
S204=S200+S202
S205=S201+S203
S31=S100-S104
S32=S101-S105
S33=S102-S106
S34=S103-S107
S35=S200-S202
```

```
S36=S201-S203
S37=S204+S205
S38=S204-S205
S39=S32+S34
S310=S31+S33
S311=S310-S39
S312=S33-S34
S313=S31-S32
S314=S35+S36
S210=S108+S110
S211=S109+S111
S212=S108-S110
S213=S115-S113
S214=S112+S114
S215=S113+S115
S216=S112-S114
S217=S109-S111
S315=S210+S211
S316=S214+S215
S317=S315+S316
S318=S210-S211
S319=S214-S215
S320=S318+S319
S321=S212+S213
S322=S216+S217
S323=S321+S322
S324=S212-S213
S325=S216-S217
S326=S324+S325
S327=S108+S112
S328=S108
S329=S112
S330=S111+S115
S331=S111
S332=S115
S333=S322-S316+S108-S330
S334=S315-S321+S111+S112-S115
S335=S333+S334
Y(I(1))=Y(I(1))+S37
U11=S31*C1701
U12=S32*C1702
U13=S33*C1703
U14=S34*C1704
U15=S35*C1705
U16=S36*C1706
U17=S37*C1707
U18=S38*C1708
U19=S39*C1709
U110=S310*C1710
U111=S311*C1711
U112=S312*C1712
```

```
U113=S313*C1713
U114=S314*C1714
U115=S315*C1715
U116=S316*C1716
U117=S317*C1717
U118=S318*C1718
U119=S319*C1719
U120=S320*C1720
U121=S321*C1721
U122=S322*C1722
U123=S323*C1723
U124=S324*C1724
U125=S325*C1725
U126=S326*C1726
U127=S327*C1727
U128=S328*C1728
U129=S329*C1729
U130=S330*C1730
U131=S331*C1731
U132=S332*C1732
U133=S333*C1733
U134=S334*C1734
U135=S335*C1735
U17=U17+Y(I(1))
U200=U19+U111
U201=U110-U111
U202=U14+U112
U203=U112-U13
U204=U12+U113
U205=U11-U113
U206=U114-U16
U207=U114+U15
U208=U18+U17
U209=U17-U18
U210=U200-U202
U211=U206+U208
U212=U201+U203
U213=U207+U209
U214=U200+U204
U215=-U206+U208
U216=U201+U205
U217=-U207+U209
U32=U210+U211
U37=U212+U213
U33=U214+U215
U36=U216+U217
U35=-U210+U211
U38=-U212+U213
U39=-U214+U215
U34=-U216+U217
U220=U115+U117
```

```
U221=U116+U117
U222=U118+U120
U223=U119+U120
U224=U121+U123
U225=U122+U123
U226=U124+U126
U227=U125+U126
U228=U135+U134
U229=U127+U228
U230=U229+U128
U231=U220+U222
U232=U220-U222
U233=U221+U223
U234=U221-U223
U235=U224+U226
U236=U224-U226
U237=U225+U227
U238=U225-U227
U239=U133-U134
U240=U229+U129
U241=U239+U239
U242=U130-U241
U243=U242+U131
U244=-U242-U132
U245=U228+U228
U246=U245+U245
U247=U239+U245
U310=U233+U237+U240
U315=U232-U238+U243
U311=U231-U235+U245
U314=-U232-U238-U247
U313=U231+U235+U230+U239
U316=-U234-U236+U244+U246
U317=-U233+U237+U241+U245
U312=U234-U236-U239
X(I(2))=T32-U310
X(I(17))=T32+U310
Y(I(2))=T310+U32
Y(I(17))=-T310+U32
X(I(3))=T33-U311
X(I(16))=T33+U311
Y(I(3))=T311+U33
Y(I(16))=-T311+U33
X(I(4))=T34-U312
X(I(15))=T34+U312
Y(I(4))=T312+U34
Y(I(15))=-T312+U34
X(I(5))=T35-U313
X(I(14))=T35+U313
Y(I(5))=T313+U35
Y(I(14))=-T313+U35
```

```
X(I(6))=T36-U314
X(I(13))=T36+U314
Y(I(6))=T314+U36
Y(I(13))=-T314+U36
X(I(7))=T37-U315
X(I(12))=T37+U315
Y(I(7))=T315+U37
Y(I(12))=-T315+U37
X(I(8))=T38-U316
X(I(11))=T38+U316
Y(I(8))=T316+U38
Y(I(11))=-T316+U38
X(I(9))=T39-U317
X(I(10))=T39+U317
Y(I(9))=T317+U39
Y(I(10))=-T317+U39
C
    GOTO 20
C
```

Figure. Length-17 FFT Module

Chapter 6

N = 19 Winograd FFT module[1]

6.1 N=19 FFT module

A FORTRAN implementation of a length-19 FFT module to be used in a Prime Factor Algorithm program.

```
    C
C-------------------WFTA N=19---------------------------
C
C 372 ADDS; 76 MPYS
C DATA FOR LENGTH 19 DFT
DATA C1901  /     -1.0555555555555556 /
DATA C1902  /      .1775222851392708 /
DATA C1903  /     -.1282007750219153 /
DATA C1904  /      .0493215101173555 /
DATA C1905  /      .5761101149100590 /
DATA C1906  /     -.7499644965553628 /
DATA C1907  /     -.1738543816453038 /
DATA C1908  /    -2.1729997561977314 /
DATA C1909  /    -1.7021211726914737 /
DATA C1910  /      .4708785835062578 /
DATA C1911  /    -2.0239400846888438 /
DATA C1912  /      .1055164120166409 /
DATA C1913  /     2.1294564967054848 /
DATA C1914  /     -.7508754389737117 /
DATA C1915  /      .1481281769515716 /
DATA C1916  /      .8990036159252833 /
DATA C1917  /     -.6214824677260278 /
DATA C1918  /     -.7986935209871269 /
DATA C1919  /     -.4733919962377183 /
DATA C1920  /     -.2421610524189263 /
DATA C1921  /     -.0593686079675051 /
DATA C1922  /      .0125786882551762 /
DATA C1923  /     -.0467899197123289 /
DATA C1924  /     -.9375012191378236 /
DATA C1925  /     -.0501115370433529 /
DATA C1926  /     -.9876127561811766 /
```

[1] This content is available online at <http://cnx.org/content/m17381/1.7/>.

```
DATA C1927  /    -1.1745786501205960 /
DATA C1928  /     1.1114482296234993 /
DATA C1929  /     2.2860268797440954 /
DATA C1930  /      .2642052325793094 /
DATA C1931  /     2.1981792779352138 /
DATA C1932  /     1.9339740453559042 /
DATA C1933  /     -.7482584709125489 /
DATA C1934  /     -.4782083564276887 /
DATA C1935  /      .2700501144848602 /
DATA C1936  /     -.3464235615954227 /
DATA C1937  /     -.8348542936068828 /
DATA C1938  /     -.3937592850674352 /
C
C-----------------WFTA N=19-------------------------
C
R100=X(I(2))+X(I(19))
R109=X(I(2))-X(I(19))
R101=X(I(3))+X(I(18))
R110=-X(I(3))+X(I(18))
R102=X(I(5))+X(I(16))
R111=X(I(5))-X(I(16))
R103=X(I(9))+X(I(12))
R112=-X(I(9))+X(I(12))
R104=X(I(17))+X(I(4))
R113=X(I(17))-X(I(4))
R105=X(I(14))+X(I(7))
R114=-X(I(14))+X(I(7))
R106=X(I(8))+X(I(13))
R115=X(I(8))-X(I(13))
R107=X(I(15))+X(I(6))
R116=-X(I(15))+X(I(6))
R108=X(I(10))+X(I(11))
R117=X(I(10))-X(I(11))
R200=R100-R106
R201=R101-R107
R202=R102-R108
R203=R103-R106
R204=R104-R107
R205=R105-R108
R206=R100+R103+R106
R207=R101+R104+R107
R208=R102+R105+R108
R209=R200+R202
R210=R203+R205
R31=R206+R207+R208
R32=R210+R204
R33=R209+R201
R34=R33-R32
R35=R210-R204
R36=R209-R201
R37=R36-R35
```

```
R38=R203
R39=R200-R203
R310=R200
R311=R205
R312=R202-R205
R313=R202
R314=-R312+R200-R204
R315=R39+R205-R201
R316=-R315+R314
R317=R206-R208
R318=R207-R208
R319=R317+R318
R220=R109-R115
R221=R110-R116
R222=R111-R117
R223=R112-R115
R224=R113-R116
R225=R114-R117
R226=R109+R112+R115
R227=R110+R113+R116
R228=R111+R114+R117
R229=R220+R222
R230=R223+R225
R320=R226+R227+R228
R321=R230+R224
R322=R229+R221
R323=R322-R321
R324=R230-R224
R325=R229-R221
R326=R325-R324
R327=R223
R328=R220-R223
R329=R220
R330=R225
R331=R222-R225
R332=R222
R333=-R331+R220-R224
R334=R328+R225-R221
R335=-R334+R333
R336=R226-R228
R337=R227-R228
R338=R336+R337
X(I(1))=X(I(1))+R31
T11=R31*C1901
T12=R32*C1902
T13=R33*C1903
T14=R34*C1904
T15=R35*C1905
T16=R36*C1906
T17=R37*C1907
T18=R38*C1908
```

```
T19=R39*C1909
T110=R310*C1910
T111=R311*C1911
T112=R312*C1912
T113=R313*C1913
T114=R314*C1914
T115=R315*C1915
T116=R316*C1916
T117=R317*C1917
T118=R318*C1918
T119=R319*C1919
T120=R320*C1920
T121=R321*C1921
T122=R322*C1922
T123=R323*C1923
T124=R324*C1924
T125=R325*C1925
T126=R326*C1926
T127=R327*C1927
T128=R328*C1928
T129=R329*C1929
T130=R330*C1930
T131=R331*C1931
T132=R332*C1932
T133=R333*C1933
T134=R334*C1934
T135=R335*C1935
T136=R336*C1936
T137=R337*C1937
T138=R338*C1938
T11=T11+X(I(1))
T200=T12+T13
T201=T15+T16
T202=T115+T116
T203=T200+T201
T204=T12+T14
T205=T15+T17
T206=T114+T116
T207=-T203+T18
T208=T204+T205
T209=T111-T206
T210=T19+T202+T207
T211=T208+T112+T209
T212=T204-T205+T202
T213=T207+T208+T110+T206
T214=T203+T113+T209+T202
T215=T200-T201+T206
T216=T117-T119
T217=T118-T119
T218=T11+T216
T219=T11-T216-T217
```

```
T220=T11+T217
T2100=T121+T122
T2101=T124+T125
T2102=T134+T135
T2103=T2100+T2101
T2104=T121+T123
T2105=T124+T126
T2106=T133+T135
T2107=-T2103+T127
T2108=T2104+T2105
T2109=T130-T2106
T2110=T128+T2102+T2107
T2111=T2108+T131+T2109
T2112=T2104-T2105+T2102
T2113=T2107+T2108+T129+T2106
T2114=T2103+T132+T2109+T2102
T2115=T2100-T2101+T2106
T2116=T136-T138
T2117=T137-T138
T2118=T120+T2116
T2119=T120-T2116-T2117
T2120=T120+T2117
T32=T213-T210+T218
T310=T214-T211+T219
T36=T215-T212+T220
T38=-T213+T218
T37=-T214+T219
T34=-T215+T220
T39=T210+T218
T35=T211+T219
T33=T212+T220
T311=T2113-T2110+T2118
T319=T2114-T2111+T2119
T315=T2115-T2112+T2120
T317=-T2113+T2118
T316=-T2114+T2119
T313=T2115-T2120
T318=-T2110-T2118
T314=T2111+T2119
T312=-T2112-T2120
S100=Y(I(2))+Y(I(19))
S109=Y(I(2))-Y(I(19))
S101=Y(I(3))+Y(I(18))
S110=-Y(I(3))+Y(I(18))
S102=Y(I(5))+Y(I(16))
S111=Y(I(5))-Y(I(16))
S103=Y(I(9))+Y(I(12))
S112=-Y(I(9))+Y(I(12))
S104=Y(I(17))+Y(I(4))
S113=Y(I(17))-Y(I(4))
S105=Y(I(14))+Y(I(7))
```

```
S114=-Y(I(14))+Y(I(7))
S106=Y(I(8))+Y(I(13))
S115=Y(I(8))-Y(I(13))
S107=Y(I(15))+Y(I(6))
S116=-Y(I(15))+Y(I(6))
S108=Y(I(10))+Y(I(11))
S117=Y(I(10))-Y(I(11))
S200=S100-S106
S201=S101-S107
S202=S102-S108
S203=S103-S106
S204=S104-S107
S205=S105-S108
S206=S100+S103+S106
S207=S101+S104+S107
S208=S102+S105+S108
S209=S200+S202
S210=S203+S205
S31=S206+S207+S208
S32=S210+S204
S33=S209+S201
S34=S33-S32
S35=S210-S204
S36=S209-S201
S37=S36-S35
S38=S203
S39=S200-S203
S310=S200
S311=S205
S312=S202-S205
S313=S202
S314=-S312+S200-S204
S315=S39+S205-S201
S316=-S315+S314
S317=S206-S208
S318=S207-S208
S319=S317+S318
S220=S109-S115
S221=S110-S116
S222=S111-S117
S223=S112-S115
S224=S113-S116
S225=S114-S117
S226=S109+S112+S115
S227=S110+S113+S116
S228=S111+S114+S117
S229=S220+S222
S230=S223+S225
S320=S226+S227+S228
S321=S230+S224
S322=S229+S221
```

```
S323=S322-S321
S324=S230-S224
S325=S229-S221
S326=S325-S324
S327=S223
S328=S220-S223
S329=S220
S330=S225
S331=S222-S225
S332=S222
S333=-S331+S220-S224
S334=S328+S225-S221
S335=-S334+S333
S336=S226-S228
S337=S227-S228
S338=S336+S337
Y(I(1))=Y(I(1))+S31
U11=S31*C1901
U12=S32*C1902
U13=S33*C1903
U14=S34*C1904
U15=S35*C1905
U16=S36*C1906
U17=S37*C1907
U18=S38*C1908
U19=S39*C1909
U110=S310*C1910
U111=S311*C1911
U112=S312*C1912
U113=S313*C1913
U114=S314*C1914
U115=S315*C1915
U116=S316*C1916
U117=S317*C1917
U118=S318*C1918
U119=S319*C1919
U120=S320*C1920
U121=S321*C1921
U122=S322*C1922
U123=S323*C1923
U124=S324*C1924
U125=S325*C1925
U126=S326*C1926
U127=S327*C1927
U128=S328*C1928
U129=S329*C1929
U130=S330*C1930
U131=S331*C1931
U132=S332*C1932
U133=S333*C1933
U134=S334*C1934
```

```
U135=S335*C1935
U136=S336*C1936
U137=S337*C1937
U138=S338*C1938
U11=U11+X(I(1))
U200=U12+U13
U201=U15+U16
U202=U115+U116
U203=U200+U201
U204=U12+U14
U205=U15+U17
U206=U114+U116
U207=-U203+U18
U208=U204+U205
U209=U111-U206
U210=U19+U202+U207
U211=U208+U112+U209
U212=U204-U205+U202
U213=U207+U208+U110+U206
U214=U203+U113+U209+U202
U215=U200-U201+U206
U216=U117-U119
U217=U118-U119
U218=U11+U216
U219=U11-U216-U217
U220=U11+U217
U2100=U121+U122
U2101=U124+U125
U2102=U134+U135
U2103=U2100+U2101
U2104=U121+U123
U2105=U124+U126
U2106=U133+U135
U2107=-U2103+U127
U2108=U2104+U2105
U2109=U130-U2106
U2110=U128+U2102+U2107
U2111=U2108+U131+U2109
U2112=U2104-U2105+U2102
U2113=U2107+U2108+U129+U2106
U2114=U2103+U132+U2109+U2102
U2115=U2100-U2101+U2106
U2116=U136-U138
U2117=U137-U138
U2118=U120+U2116
U2119=U120-U2116-U2117
U2120=U120+U2117
U32=U213-U210+U218
U310=U214-U211+U219
U36=U215-U212+U220
U38=-U213+U218
```

```
U37=-U214+U219
U34=-U215+U220
U39=U210+U218
U35=U211+U219
U33=U212+U220
U311=U2113-U2110+U2118
U319=U2114-U2111+U2119
U315=U2115-U2112+U2120
U317=-U2113+U2118
U316=-U2114+U2119
U313=U2115-U2120
U318=-U2110-U2118
U314=U2111+U2119
U312=-U2112-U2120
X(I(2))=T32-U311
X(I(19))=T32+U311
Y(I(2))=T311+U32
Y(I(19))=-T311+U32
X(I(3))=T33-U312
X(I(18))=T33+U312
Y(I(3))=T312+U33
Y(I(18))=-T312+U33
X(I(4))=T34-U313
X(I(17))=T34+U313
Y(I(4))=T313+U34
Y(I(17))=-T313+U34
X(I(5))=T35-U314
X(I(16))=T35+U314
Y(I(5))=T314+U35
Y(I(16))=-T314+U35
X(I(6))=T36-U315
X(I(15))=T36+U315
Y(I(6))=T315+U36
Y(I(15))=-T315+U36
X(I(7))=T37-U316
X(I(14))=T37+U316
Y(I(7))=T316+U37
Y(I(14))=-T316+U37
X(I(8))=T38-U317
X(I(13)=T38+U317
Y(I(8))=T317+U38
Y(I(13)=-T317+U38
X(I(9))=T39-U318
X(I(12))=T39+U318
Y(I(9))=T318+U39
Y(I(12))=-T318+U39
X(I(10))=T310-U319
X(I(11))=T310+U319
Y(I(10))=T319+U310
Y(I(11))=-T319+U310
C
```

```
    GOTO 20
C
```

Figure. Length-19 FFT Module

Chapter 7

N = 25 FFT module[1]

7.1 N=25 FFT module

A FORTRAN implementation of a length-25 FFT module to be used in a Prime Factor Algorithm program.

```
    C
C--------------------WFTA N=25----------------------------
C
C 420 ADDS; 132 MPYS
C DATA FOR LENGTH 25 DFT
DATA C5001  /     -.2500000000000000    /
DATA C5002  /      .5590169943749474    /
DATA C5003  /     -.3632712640026805    /
DATA C5004  /     1.5388417685876267    /
DATA C5005  /     -.5877852522924731    /
DATA C5102  /     .2236067977499788E+01 /
DATA C5103  /    -.1453085056010720E+01 /
DATA C5104  /     .6155367074350504E+01 /
DATA C5105  /    -.2351141009169892E+01 /
DATA C2510/       -.0760795655183429     /
DATA C2511/        .0449933296227360     /
DATA C2512/        .0605364475705394     /
DATA C2520/       -.0848787721340987     /
DATA C2521/        .0246595628713843     /
DATA C2522/        .0547691675027415     /
DATA C2530/       -.0883447333343813     /
DATA C2531/        .0027763450932952     /
DATA C2532/        .0455605392138382     /
DATA C2540/       -.0862596700300632     /
DATA C2541/       -.0192813206576887     /
DATA C2542/        .0334891746861873     /
DATA C2560/       -.0663010779973491     /
DATA C2561/       -.0584522630561849     /
DATA C2562/        .0039244074705821     /
DATA C2580/       -.0299404850563092     /
DATA C2581/       -.0831628965019433     /
```

[1] This content is available online at <http://cnx.org/content/m17383/1.5/>.

```
DATA C2582/       -.0266112057228170     /
DATA C2590/       -.0083180783141937     /
DATA C2591/       -.0879960770327799     /
DATA C2592/       -.0398389993592931     /
DATA C25120 /        .0541738417343859    /
DATA C25121 /       -.0698404959299239    /
DATA C25122 /       -.0620071688321549    /
DATA C25160 /        .0879960770327799    /
DATA C25161 /        .0083180783141937    /
DATA C25162 /       -.0398389993592931    /
C
C--------------------CFA N=25----------------------------
C
R101=X(I(6))+X(I(21))
R102=X(I(11))+X(I(16))
R103=X(I(6))-X(I(21))
R104=X(I(11))-X(I(16))
R31=R101+R102
R32=R101-R102
R35=R103+R104
T11 =R31 *C5001+X(I(1))
T12 =R32 *C5002
T13 =R103 *C5003
T14 =R104 *C5004
T15 =R35 *C5005
T32=T11+T12
T33=T11-T12
T34=T13+T15
T35=T14+T15
S101=Y(I(6))+Y(I(21))
S102=Y(I(11))+Y(I(16))
S103=Y(I(6))-Y(I(21))
S104=Y(I(11))-Y(I(16))
S31=S101+S102
S32=S101-S102
S35=S103+S104
U11 =S31 *C5001+Y(I(1))
U12 =S32 *C5002
U13 =S103 *C5003
U14 =S104 *C5004
U15 =S35 *C5005
U32=U11+U12
U33=U11-U12
U34=U13+U15
U35=U14+U15
XT1=X(I(1))+R31
YT1=Y(I(1))+S31
XT6= T32-U34
XT21= T32+U34
YT6= T34+U32
YT21=-T34+U32
```

```
XT11= T33-U35
XT16= T33+U35
YT11= T35+U33
YT16=-T35+U33
R101=X(I(7))+X(I(22))
R102=X(I(12))+X(I(17))
R103=X(I(7))-X(I(22))
R104=X(I(12))-X(I(17))
T31=R101+R102
R32=R101-R102
R35=R103+T104
T16=X(I(2))+X(I(2))
T11=T16+T16-R31
T12 =R32 *5102
T13 =R103 *5103
T14 =R104 *5104
T15=R35*C5105
T32=T11+T12
T33=T11-T12
T34=T13+T15
T35=T14+T15
S101=Y(I(7))+Y(I(22))
S102=Y(I(12))+Y(I(17))
S103=Y(I(7))-Y(I(22))
S104=Y(I(12))-Y(I(17))
S31=S101+S102
S32=S101-S102
S35=S103+S104
U16=Y(I(2))+Y(I(2))
U11=U16+U16-S31
U12 =S32 *5102
U13 =S103 *5103
U14 =S104 *5104
U15=S35*5105
U32=U11+U12
U33=U11-U12
U34=U13+U15
U35=U14+U15
XT2=X(I(2))+R31
YT2=Y(I(2))+S31
XT7= T32-U34
XT22= T32+U34
YT7= T32+U34
YT22=-T32+U34
XT12= T33-U35
XT17= T33+U35
YT12= T35+U33
YT17=-T35+U33
R101=X(I(8))+X(I(23))
R102=X(I(13))+X(I(18))
R103=X(I(8))-X(I(23))
```

```
R104=X(I(13))-X(I(18))
R31=R101+R102
R32=R101-R102
R35=R103+R104
T16=X(I(3))+X(I(3))
T11=T16+T16-R31
T12 =R32 *C5102
T13 =R103 *C5103
T14 =R104 *C5104
T15=R35*C5105
T32=T11+T12
T33=T11-T12
T34=T13+T15
T35=T14+T15
S101=Y(I(8))+Y(I(23))
S102=Y(I(13))+Y(I(18))
S103=Y(I(8))-Y(I(23))
S104=Y(I(13))-Y(I(18))
S31=S101+S102
S32=S101-S102
S35=S103+S104
U16=Y(I(3))+Y(I(3))
U11=U16+U16-S31
U12 =S32 *C5102
U13 =S103 *C5103
U14 =S104 *C5104
U15=S35*C5105
U32=U11+U12
U33=U11-U12
U34=U13+U15
U35=U14+U15
XT3=X(I(3))+R31
YT3=Y(I(3))+S31
XT8= T32-U34
XT23= T32+U34
YT8= T34+U32
YT23=-T34+U32
XT13= T33-U35
XT18= T33+U35
YT13= T35+U33
YT18=-T35+U33
R101=X(I(9))+X(I(24))
R102=X(I(14))+X(I(19))
R103=X(I(9))-X(I(24))
R104=X(I(14))-X(I(19))
R31=R101+R102
R32=R101-R102
R35=R103+R104
T16=X(I(4))+X(I(4))
T11=T16+T16-R31
T12 =R32 *C5102
```

```
T13 =R103 *C5103
T14 =R104 *C5104
T15=R35*C5105
T32=T11+T12
T33=T11-T12
T34=T13+T15
T35=T14+T15
S101=Y(I(9))+Y(I(24))
S102=Y(I(14))+Y(I(19))
S103=Y(I(9))-Y(I(24))
S104=Y(I(14))-Y(I(19))
S31=S101+S102
S32=S101-S102
S35=S103+S104
U16=Y(I(4))+Y(I(4))
U11=U16+U16-S31
U12 =S32 *C5102
U13 =S103 *C5103
U14 =S104 *C5104
U15=S35*C5105
U32=U11+U12
U33=U11-U12
U34=U13+U15
U35=U14+U15
XT4=X(I(4))+R31
YT4=Y(I(4))+S31
XT9= T32-U34
XT24= T32+U34
YT9= T34+U32
YT24=-T34+U32
XT14= T33-U35
XT19= T33+U35
YT14= T35+U33
YT19=-T35+U33
R101=X(I(10))+X(I(25))
R102=X(I(15))+X(I(20))
R103=X(I(10))-X(I(25))
R104=X(I(15))-X(I(20))
R31=R101+R102
R32=R101-R102
R35=R103+R104
T16=X(I(5))+X(I(5))
T11=T16+T16-R31
T12 =R32 *C5102
T13 =R103 *C5103
T14 =R104 *C5104
T15=R35*C5105
T32=T11+T12
T33=T11-T12
T34=T13+T15
T35=T14+T15
```

```
S101=Y(I(10))+Y(I(25))
S102=Y(I(15))+Y(I(20))
S103=Y(I(10))-Y(I(25))
S104=Y(I(15))-Y(I(20))
S31=S101+S102
S32=S101-S102
S35=S103+S104
U16=Y(I(5))+Y(I(5))
U11=U16+U16-S31
U12 =S32 *C5102
U13 =S103 *C5103
U14 =S104 *C5104
U15=S35*C5105
U32=U11+U12
U33=U11-U12
U34=U13+U15
U35=U14+U15
XT5=X(I(5))+R31
YT5=Y(I(5))+S31
XT10= T32-U34
XT25= T32+U34
YT10= T34+U32
YT25=-T34+U32
XT15= T33-U35
XT20= T33+U35
YT15= T35+U33
YT20=-T35+U33
T1=(XT7+YT7)*C2512
T2=XT7*C2510
XT7=T1-YT7*C2511
YT7=T1+T2
T1=(XT12+YT12)*C2522
T2=XT12*C2520
XT12=T1-YT12*C2521
YT12=T1+T2
T1=(XT17+YT17)*C2532
T2=XT17*C2530
XT17=T1-YT17*C2531
YT17=T1+T2
T1=(XT22+YT22)*C2542
T2=XT22*C2540
XT22=T1-YT22*C2541
YT22=T1+T2
T1=(XT8+YT8)*C2522
T2=XT8*C2520
XT8=T1-YT8*C2521
YT8=T1+T2
T1=(XT13+YT13)*C2542
T2=XT13*C2540
XT13=T1-YT13*C2541
YT13=T1+T2
```

```
T1=(XT18+YT18)*C2562
T2=XT18*C2560
XT18=T1-YT18*C2561
YT18=T1+T2
T1=(XT23+YT23)*C2582
T2=XT23*C2580
XT23=T1-YT23*C2581
YT23=T1+T2
T1=(XT9+YT9)*C2532
T2=XT9*C2530
XT9=T1-YT9*C2531
YT9=T1+T2
T1=(XT14+YT14)*C2562
T2=XT14*C2560
XT14=T1-YT14*C2561
YT14=T1+T2
T1=(XT19+YT19)*C2592
T2=XT19*C2590
XT19=T1-YT19*C2591
YT19=T1+T2
T1=(XT24+YT24)*C25122
T2=XT24*C25120
XT24=T1-YT24*C25121
YT24=T1+T2
T1=(XT10+YT10)*C2542
T2=XT10*C2540
XT10=T1-YT10*C2541
YT10=T1+T2
T1=(XT15+YT15)*C2582
T2=XT15*C2580
XT15=T1-YT15*C2581
YT15=T1+T2
T1=(XT20+YT20)*C25122
T2=XT20*C25120
XT20=T1-YT20*C25121
YT20=T1+T2
T1=(XT25+YT25)*C25162
T2=XT25*C25160
XT25=T1-YT25*C25161
YT25=T1+T2
R101=XT2+XT5
R102=XT3+XT4
R103=XT2-XT5
R104=XT3-XT4
R31=R101+R102
R32=R101-R102
R35=R103+R104
T11 =R31 *C5001+XT1
T12 =R32 *C5002
T13 =R103 *C5003
T14 =R104 *C5004
```

```
T15 =R35 *C5005
T32=T11+T12
T33=T11-T12
T34=T13+T15
T35=T14+T15
S101=YT2+YT5
S102=YT3+YT4
S103=YT2-YT5
S104=YT3-YT4
S31=S101+S102
S32=S101-S102
S35=S103+S104
U11 =S31 *C5001+YT1
U12 =S32 *C5002
U13 =S103 *C5003
U14 =S104 *C5004
U15 =S35 *C5005
U32=U11+U12
U33=U11-U12
U34=U13+U15
U35=U14+U15
X(I(1))=XT1+R31
Y(I(1))=YT1+S31
X(I(6))= T32-U34
X(I(21))= T32+U34
Y(I(6))= T34+U32
Y(I(21))=-T34+U32
X(I(11))= T33-U35
X(I(16))= T33+U35
Y(I(11))= T35+U33
Y(I(16))=-T35+U33
R101=XT7+XT10
R102=XT8+XT9
R103=XT7-XT10
R104=XT8-XT9
R31=R101+R102
R32=R101-R102
R35=R103+R104
T11 =XT6-R31
T12 =R32 *C5102
T13 =R103 *C5103
T14 =R104 *C5104
T15=R35*C5105
T32=T11+T12
T33=T11-T12
T34=T13+T15
T35=T14+T15
S101=YT7+YT10
S102=YT8+YT9
S103=YT7-YT10
S104=YT8-YT9
```

```
S31=S101+S102
S32=S101-S102
S35=S103+S104
U11 =YT6-S31
U12 =S32 *C5102
U13 =S103 *C5103
U14 =S104 *C5104
U15=S35*C5105
U32=U11+U12
U33=U11-U12
U34=U13+U15
U35=U14+U15
R31=R31+R31
S31=S31+S31
X(I(2))=XT6+R31+R31
Y(I(2))=YT6+S31+S31
X(I(7))= T32-U34
X(I(22))= T32+U34
Y(I(7))= T34+U32
Y(I(22))=-T34+U32
X(I(12))= T33-U35
X(I(17))= T33+U35
Y(I(12))= T35+U33
Y(I(17))=-T35+U33
R101=XT12+XT15
R102=XT13+XT14
R103=XT12-XT15
R104=XT13-XT14
R31=R101+R102
R32=R101-R102
R35=R103+R104
T11 =XT11-R31
T12 =R32 *C5102
T13 =R103 *C5103
T14 =R104 *C5104
T15=R35*C5105
T32=T11+T12
T33=T11-T12
T34=T13+T15
T35=T14+T15
S101=YT12+YT15
S102=YT13+YT14
S103=YT12-YT15
S104=YT13-YT14
S31=S101+S102
S32=S101-S102
S35=S103+S104
U11 =YT11-S31
U12 =S32 *C5102
U13 =S103 *C5103
U14 =S104 *C5104
```

```
U15=S35*C5105
U32=U11+U12
U33=U11-U12
U34=U13+U15
U35=U14+U15
R31=R31+R31
S31=S31+S31
X(I(3))=XT11+R31+R31
Y(I(3))=YT11+S31+S31
X(I(8))= T32-U34
X(I(23))= T32+U34
Y(I(8))= T34+U32
Y(I(23))=-T34+U32
X(I(13))= T33-U35
X(I(18))= T33+U35
Y(I(13))= T35+U33
Y(I(18))=-T35+U33
R101=XT17+XT20
R102=XT18+XT19
R103=XT17-XT20
R104=XT18-XT19
R31=R101+R102
R32=R101-R102
R35=R103+R104
T11 =XT16-R31
T12 =R32 *C5102
T13 =R103 *C5103
T14 =R104 *C5104
T15=R35*C5105
T32=T11+T12
T33=T11-T12
T34=T13+T15
T35=T14+T15
S101=YT17+YT20
S102=YT18+YT19
S103=YT17-YT20
S104=YT18-YT19
S31=S101+S102
S32=S101-S102
S35=S103+S104
U11 =YT16-S31
U12 =S32 *C5102
U13 =S103 *C5103
U14 =S104 *C5104
U15=S35*C5105
U32=U11+U12
U33=U11-U12
U34=U13+U15
U35=U14+U15
R31=R31+R31
S31=S31+S31
```

```
X(I(4))=XT16+R31+R31
Y(I(4))=YT16+S31+S31
X(I(9))= T32-U34
X(I(24))= T32+U34
Y(I(9))= T34+U32
Y(I(24))=-T34+U32
X(I(14)= T33-U35
X(I(19))= T33+U35
Y(I(14))= T35+U33
Y(I(19))=-T35+U33
R101=XT22+XT25
R102=XT23+XT24
R103=XT22-XT25
R104=XT23-XT24
R31=R101+R102
R32=R101-R102
R35=R103+R104
T11 =XT21-R31
T12 =R32 *C5102
T13 =R103 *C5103
T14 =R104 *C5104
T15=R35*C5105
T32=T11+T12
T33=T11-T12
T34=T13+T15
T35=T14+T15
S101=YT22+YT25
S102=YT23+YT24
S103=YT22-YT25
S104=YT23-YT24
S31=S101+S102
S32=S101-S102
S35=S103+S104
U11 =YT21-S31
U12 =S32 *C5102
U13 =S103 *C5103
U14 =S104 *C5104
U15=S35*C5105
U32=U11+U12
U33=U11-U12
U34=U13+U15
U35=U14+U15
R31=R31+R31
S31=S31+S31
X(I(5))=XT21+R31+R31
Y(I(5))=YT21+S31+S31
X(I(10))= T32-U34
X(I(25))= T32+U34
Y(I(10))= T34+U32
Y(I(25))=-T34+U32
X(I(15))= T33-U35
```

```
X(I(20))= T33+U35
Y(I(15))= T35+U33
Y(I(20))=-T35+U33
C
    GOTO 20
C
```

Figure. Length-25 FFT Module

Bibliography

[1] C. S. Burrus and P. W. Eschenbacker. An in-place, in-order prime factor fft algorithm. *IEEE Trans. ASSP*, 29, August 1981.

[2] C.S. Burrus. Index mappings for multidimensional formulation of the dft and convolution. *IEEE Trans. ASSP*, June 1977.

[3] H. W. Johnson. *An Approach to the Design of Large DFT's.* Ph. d. thesis, Rice University, 1982.

[4] J. H. McClellan and C. M. Rader. *Number Theory in Digital Signal Processing.* Prentice-Hall, 1979.

[5] H. J. Nussbaumer. *FFT and Convolution Algorithms.* Springer-Verlag, 1981.

[6] B. Rice. Winograd convolution algorithms over finite fields. *Congressus Numeratium*, 29:827, August 1979.

[7] H. F. Silverman. An introduction to programming the winograd fourier transform algorithm. *IEEE Trans. ASSP*, April 1977.

[8] M.H. Narasimha S.S. Narayan and A.M. Peterson. Dft algorithms analysis and implementation. Report no. 3606-12, Standford University, May 1978. There are several errors in the algoritms in this report.

[9] S. Zohar. A prescription of winograd's discrete fourier transform algorithm. *IEEE Trans. ASSP*, August 1979.

Index of Keywords and Terms

Keywords are listed by the section with that keyword (page numbers are in parentheses). Keywords do not necessarily appear in the text of the page. They are merely associated with that section. *Ex.* apples, § 1.1 (1) **Terms** are referenced by the page they appear on. *Ex.* apples, 1

Attributions

Collection: *Large DFT Modules: 11, 13, 16, 17, 19, and 25. Revised ECE Technical Report 8105*
Edited by: Howard Johnson, C. Sidney Burrus
URL: http://cnx.org/content/col10569/1.7/

Module: "Large DFT Modules: 11, 13, 16, 17, 19, and 25"
By: Howard Johnson, C. Sidney Burrus
URL: http://cnx.org/content/m17413/1.7/
Pages: 1-11

Module: "N = 11 Winograd FFT module"
By: Howard Johnson, C. Sidney Burrus
URL: http://cnx.org/content/m17377/1.10/
Pages: 13-16

Module: "N = 13 Winograd FFT module"
By: Howard Johnson, C. Sidney Burrus
URL: http://cnx.org/content/m17378/1.7/
Pages: 17-21

Module: "N = 16 FFT module"
By: Howard Johnson, C. Sidney Burrus
URL: http://cnx.org/content/m17382/1.5/
Pages: 23-26

Module: "N = 17 Winograd FFT module"
By: Howard Johnson, C. Sidney Burrus
URL: http://cnx.org/content/m17380/1.10/
Pages: 27-35

Module: "N = 19 Winograd FFT module"
By: Howard Johnson, C. Sidney Burrus
URL: http://cnx.org/content/m17381/1.7/
Pages: 37-46

Module: "N = 25 FFT module"
By: Howard Johnson, C. Sidney Burrus
URL: http://cnx.org/content/m17383/1.5/
Pages: 47-58

Large DFT Modules: 11, 13, 16, 17, 19, and 25. Revised ECE Technical Report 8105
This is a slightly revised version of: "Large DFT Modules: 11, 13, 16, 17, 19, and 25" which was an ECE Technical Report 8105 at Rice University, Dec. 13, 1981. The original authors were H. W. Johnson and C. S. Burrus. The content is a report on part of the doctoral research by Howard Johnson. He derived and described the FFT modules contained in the report.