Chapter 11.05
Informal Development of Fast Fourier Transform (FFT)

Introduction

Recalled the DFT pairs of Equations (22) and (23) (of Chapter 11.04) and swapping the indices n,k one obtains:

$$\widetilde{C}_n = \sum_{k=0}^{N-1} f(k) e^{-in\left(w_0=\frac{2\pi}{N}\right)k} \tag{1}$$

$$f(k) = \left(\frac{1}{N}\right)\sum_{n=0}^{N-1} \tilde{C}_n\, e^{in\left(w_0=\frac{2\pi}{N}\right)k} \tag{2}$$

$$\text{where } n,k = 0,1,2,3,...N-1 \tag{3}$$

While the above DFT pairs of equations are convenient for computer implementation, they still require substantial computation effort. The objective of this chapter, therefore, is to develop the improved version of DFT (namely Fast Fourier Transform, or FFT) so that much larger sampling data can be handled more efficiently.

Let

$$E = e^{-i\frac{2\pi}{N}} \text{ (hence } E^N = e^{-i2\pi} = \cos(2\pi) - i\sin(2\pi) = 1) \tag{4}$$

Then Equation (1) and Equation (2) become

$$\widetilde{C}_n = \widetilde{C}(n) = \sum_{k=0}^{N-1} f(k) E^{nk} \tag{5}$$

$$f(k) = \left(\frac{1}{N}\right)\sum_{n=0}^{N-1} \widetilde{C}_n E^{-nk}$$

It should be emphasized here that in performing interpolation, one usually has to solve a system of equations to determine the unknown coefficients of the linear combination of basis functions that fit the given data. For example, if $N = 4$, then one need to solve the following system (see the second part of Equation (5)), for obtaining $\left\{\widetilde{C}\right\}$, with a given vector $\{f\}$.

$$\left(\frac{1}{N}\right)\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & E^{-1} & E^{-2} & E^{-3} \\ 1 & E^{-2} & E^{-4} & E^{-6} \\ 1 & E^{-3} & E^{-6} & E^{-9} \end{bmatrix} \begin{Bmatrix} \widetilde{C}(0) \\ \widetilde{C}(1) \\ \widetilde{C}(2) \\ \widetilde{C}(3) \end{Bmatrix} = \begin{Bmatrix} f(0) \\ f(1) \\ f(2) \\ f(3) \end{Bmatrix} \tag{5a}$$

However, the inverse of the above coefficient matrix can be easily obtained as

$$\left[\left(\frac{1}{N}\right)\begin{bmatrix}1 & 1 & 1 & 1\\ 1 & E^{-1} & E^{-2} & E^{-3}\\ 1 & E^{-2} & E^{-4} & E^{-6}\\ 1 & E^{-3} & E^{-6} & E^{-9}\end{bmatrix}\right]^{-1} = \begin{bmatrix}1 & 1 & 1 & 1\\ 1 & E^{1} & E^{2} & E^{3}\\ 1 & E^{2} & E^{4} & E^{6}\\ 1 & E^{3} & E^{6} & E^{9}\end{bmatrix}$$

Thus, the unknown vector $\{\widetilde{C}\}$ can be computed as matrix times vector operations, as following:

Assuming $N = 4 = 2^{(r=2)}$, then (see the first part of Equation (5))

$$\begin{Bmatrix}\widetilde{C}(0)\\ \widetilde{C}(1)\\ \widetilde{C}(2)\\ \widetilde{C}(3)\end{Bmatrix} = \begin{bmatrix}E^{(0)(0)} & E^{(0)(1)} & E^{(0)(2)} & E^{(0)(3)}\\ E^{(1)(0)} & E^{(1)(1)} & E^{(1)(2)} & E^{(1)(3)}\\ E^{(2)(0)} & E^{(2)(1)} & E^{(2)(2)} & E^{(2)(3)}\\ E^{(3)(0)} & E^{(3)(1)} & E^{(3)(2)} & E^{(3)(3)}\end{bmatrix}\begin{Bmatrix}f(0)\\ f(1)\\ f(2)\\ f(3)\end{Bmatrix} \tag{6}$$

$$\begin{Bmatrix}\widetilde{C}(0)\\ \widetilde{C}(1)\\ \widetilde{C}(2)\\ \widetilde{C}(3)\end{Bmatrix} = \begin{bmatrix}E^{0} & E^{0} & E^{0} & E^{0}\\ E^{0} & E^{1} & E^{2} & E^{3}\\ E^{0} & E^{2} & E^{4} & E^{6}\\ E^{0} & E^{3} & E^{6} & E^{9}\end{bmatrix}\begin{Bmatrix}f(0)\\ f(1)\\ f(2)\\ f(3)\end{Bmatrix} \tag{7}$$

For $N = 4$, $n = 2$ and $k = 3$, then

$$\begin{aligned}E^{nk} &= E^{6}\\ &= \left[E^{(N=4)}\right]E^{2}\\ &= \left(e^{\frac{-i2\pi}{N}}\right)^{N} E^{2}\\ &= \left[e^{-i2\pi}\right]E^{2}\\ &= E^{2}\end{aligned}$$

The term inside the square bracket is equal to 1, since

$$\begin{aligned}[e^{-i2\pi}] &= \cos(-2\pi) + i\sin(-2\pi)\\ &= \cos(2\pi) - i\sin(2\pi)\\ &= 1 - i(0) = 1\end{aligned}$$

For $N = 4$, $n = 3$ and $k = 3$, then

$$\begin{aligned}E^{nk} &= E^{9}\\ &= [E^{8}]E^{1}\\ &= [E^{2N}]E^{1}\\ &= \left[e^{-\frac{i2\pi}{N}\times 2N}\right]E^{1}\\ &= \left[e^{-i4\pi}\right]E^{1}\\ &= E^{1}\end{aligned}$$

In the above equation, one should recall the following Euler identity

$$\begin{aligned} e^{-i4\pi} &= Cos(4\pi) - iSin(4\pi) \\ &= 1 \end{aligned}$$

Thus, in general (for $nk \geq N$)

$$E^{nk} = E^{U}$$

where

$$\begin{aligned} U &= \text{mod}(nk, N) \\ &= \text{remainder of } \left(\frac{nk}{N}\right) \end{aligned} \tag{8}$$

Remarks:

Matrix times vector, shown in Equation (7), will require 16 (or N^2) complex multiplications and 12 (or $N*(N-1)$) complex additions.

Usage of Equation (8) will help to reduce the number of operation counts, as explained in the next section.

Factorized Matrix and Further Operation Count

Equation (7) can be factorized as:

$$\begin{Bmatrix} \widetilde{C}(0) \\ \widetilde{C}(2) \\ \widetilde{C}(1) \\ \widetilde{C}(3) \end{Bmatrix} = \begin{bmatrix} 1 & E^0 & 0 & 0 \\ 1 & E^2 & 0 & 0 \\ 0 & 0 & 1 & E^1 \\ 0 & 0 & 1 & E^3 \end{bmatrix} \begin{bmatrix} 1 & 0 & E^0 & 0 \\ 0 & 1 & 0 & E^0 \\ 1 & 0 & E^2 & 0 \\ 0 & 1 & 0 & E^2 \end{bmatrix} \begin{Bmatrix} f(0) \\ f(1) \\ f(2) \\ f(3) \end{Bmatrix} \tag{9}$$

Remarks:

The theory behind the 2 matrices on the right hand side (RHS) of Equation (9) will be clearly explained soon (see Equations 11 and 15, in chapter 11.06).

The order of the left-hand-side (LHS) vector has been changed, such as rows 2 and 3 have been swapped.

Let the row-interchanged LHS vector be defined as

$$\widetilde{C}^{*}(n) = \begin{Bmatrix} \widetilde{C}(0) \\ \widetilde{C}(2) \\ \widetilde{C}(1) \\ \widetilde{C}(3) \end{Bmatrix} \tag{10}$$

Now performing the inner-product (matrix times vector) on the RHS of Equation (9), one obtains

$$\begin{Bmatrix} f_1(0) \\ f_1(1) \\ f_1(2) \\ f_1(3) \end{Bmatrix} = \begin{bmatrix} 1 & 0 & E^0 & 0 \\ 0 & 1 & 0 & E^0 \\ 1 & 0 & E^2 & 0 \\ 0 & 1 & 0 & E^2 \end{bmatrix} \begin{Bmatrix} f(0) \\ f(1) \\ f(2) \\ f(3) \end{Bmatrix} \tag{11}$$

or

$$f_1(0) = f(0) + E^0 f(2) \tag{11a}$$

$$f_1(1) = f(1) + E^0 f(3) \tag{11b}$$

$$f_1(2) = f(0) - E^0 f(2) \tag{11c}$$

since

$$\begin{aligned} E^2 &= e^{-i\frac{2\pi}{4}*2} \\ &= e^{-i\pi} \\ &= -1 \\ &= -E^0 \end{aligned}$$

$$\begin{aligned} f_1(3) &= f(1) + E^2 f(3) \\ &= f(1) - E^0 f(3) \end{aligned} \tag{11d}$$

Equations (11a through 11d) for the "inner" matrix times vector requires 2 complex multiplications and 4 complex additions.

In Equations (11a–11d), E^0 is intentionally not reduced to the numerical value of 1.0 to facilitate the discussions of more general cases.

Finally, performing the "outer" product (matrix times vector) on the RHS of Equation (9), one obtains:

$$\begin{Bmatrix} \widetilde{C}(0) \\ \widetilde{C}(2) \\ \widetilde{C}(1) \\ \widetilde{C}(3) \end{Bmatrix} = \begin{Bmatrix} f_2(0) \\ f_2(1) \\ f_2(2) \\ f_2(3) \end{Bmatrix} = \begin{bmatrix} 1 & E^0 & 0 & 0 \\ 1 & E^2 & 0 & 0 \\ 0 & 0 & 1 & E^1 \\ 0 & 0 & 1 & E^3 \end{bmatrix} \begin{Bmatrix} f_1(0) \\ f_1(1) \\ f_1(2) \\ f_1(3) \end{Bmatrix} \tag{13}$$

or

$$f_2(0) = f_1(0) + E^0 f_1(1) \tag{14a}$$

$$\begin{aligned} f_2(1) &= f_1(0) + E^2 f_1(1) \\ &= f_1(0) - E^0 f_1(1) \end{aligned} \tag{14b}$$

$$f_2(2) = f_1(2) + E^1 f_1(3) \tag{14c}$$

$$\begin{aligned} f_2(3) &= f_1(2) + E^3 f_1(3) \\ &= f_1(2) + E^2 E^1 f_1(3) \\ &= f_1(2) - E^1 f_1(3) \end{aligned} \tag{14d}$$

Again, Equations (14a-14d) requires 2 complex multiplications and 4 complex additions. Thus, the complete RHS of Equation (9) can be computed by only 4 complex multiplications (or $N\frac{r}{2} = 4\frac{2}{2}$) and 8 complex additions (or $Nr = 4 \times 2$). Since computational time is mainly controlled by the number of multiplications, hence implementing Equation (9) will significantly reduce the number of multiplication, as compared to direct matrix times vector operations (as shown in Equation (7)).

For large value of data points ($= N$), the ratio of complex multiplications by using Equation (7) and Equation (9) can be computed as

$$Ratio = \frac{N^2}{\left(\frac{Nr}{2}\right)} = \left(\frac{2N}{r}\right) \tag{15}$$

For $N = 2048 = 2^{(r=11)}$, Equation (15) gives

$$Ratio = \frac{2(2048)}{11} = 372.36\,,$$

which basically implies that the number of complex multiplications involved in Equation (9) is about 372 times less than the one involved in Equation (7).

Graphical flow of Equation (9), for case $N = 2^r = 2^2 = 4$

Equation (9) can also be presented in the graphical form, as shown in Figure 1.

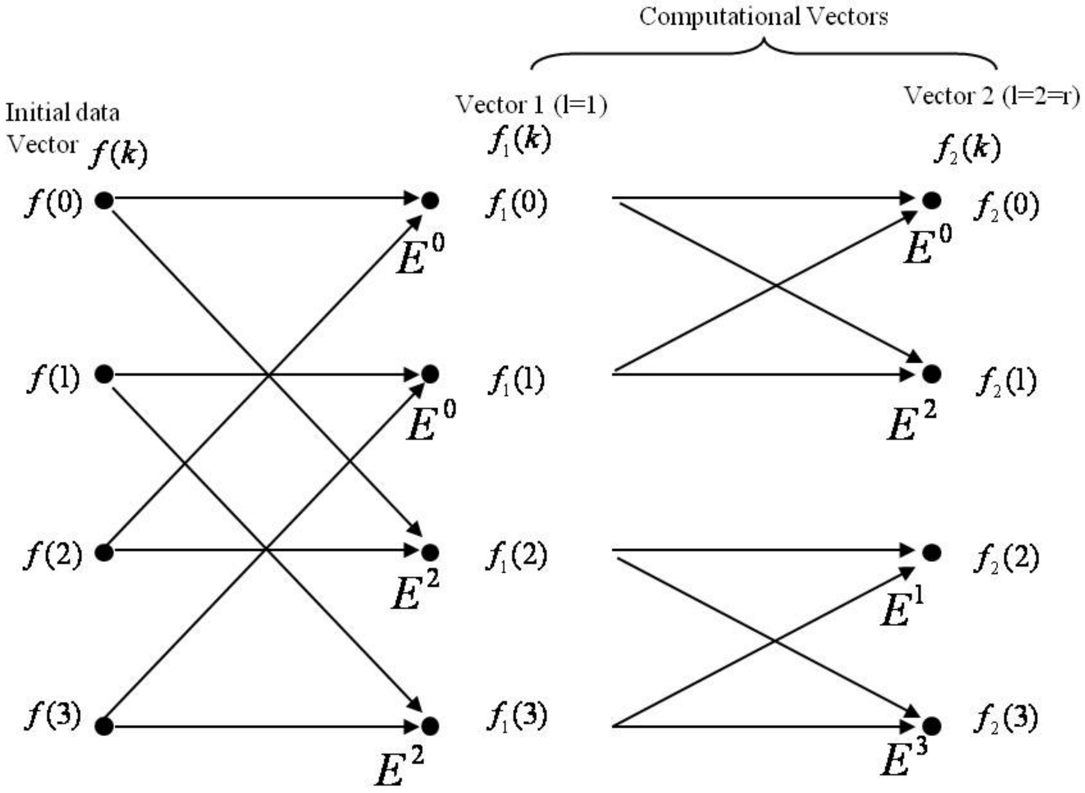

Figure 1 Graphical form of FFT (Equation 9) for the case $N = 2^r = 2^2 = 4$.

Remarks:

a) Computed vector 1 does correspond to Equations (11a–11d).
b) Computed vector 2 does correspond to Equations (14a-14d).
c) Since $r = 2$ in this example, one needs to compute 2 vectors $\{= f_1(k) \text{ and } f_2(k)\}$
d) Each node in the graph is computed from 2($= r$) nodes in the “previous” vector.
e) Factor E^U (such as E^0, E^1, E^2, E^3) appears near the arrow head of the transmission path. Absence of E^U implies that $E^U = E^0 = 1$.

For example: $f_2(2) = f_1(2) + f_1(3)E^1$, which is the same as Equation (14c).

Graphical Flow of Equation (9), for case $N = 2^r = 2^4 = 16$

To see a more detailed computational patterns of FFT, a slightly larger data size ($N = 2^r = 2^4 = 16$) is shown in the graphical form, as indicated in Figure 2.

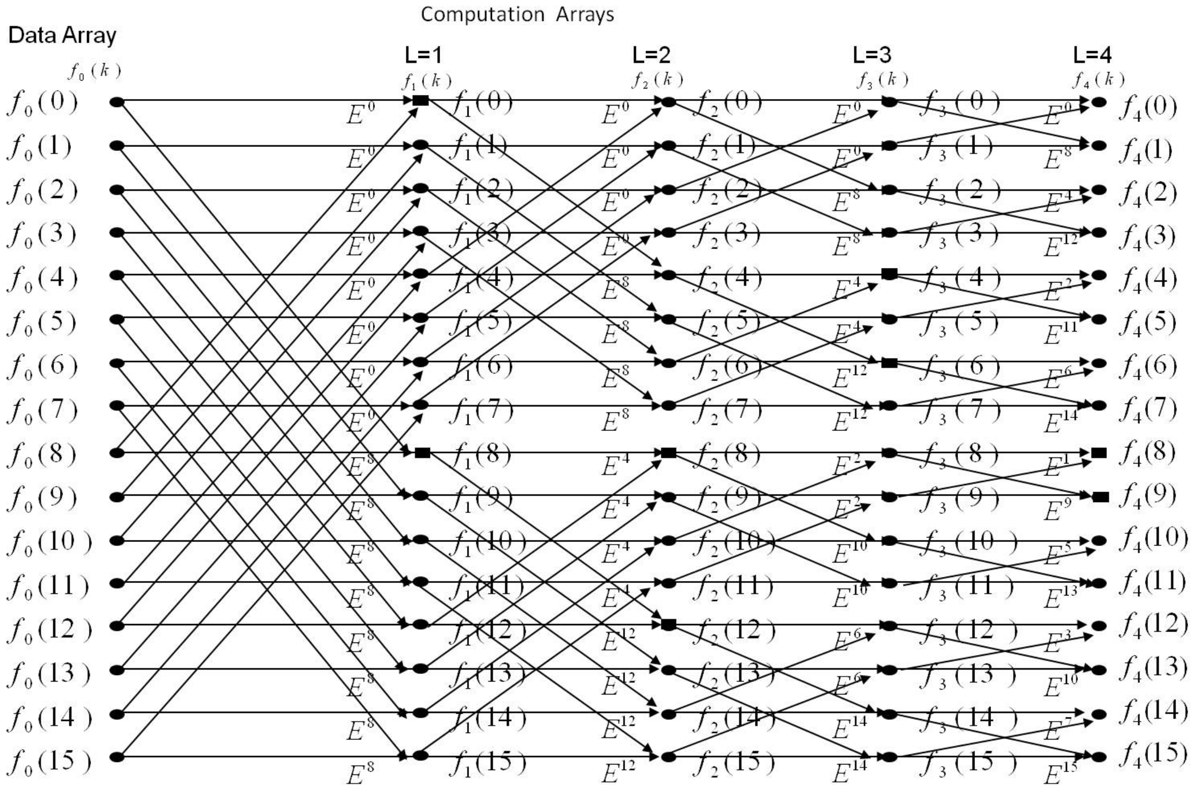

Figure 2 Graphical form of FFT (Equation 9) for the case $N = 2^r = 2^4 = 16$.

Companion Node Observation

Careful observation of Figure 2 reveals that for each computed l^{th}-vector (where $l = 1,2......r$; and $N = 2^r = 2^4 = 16$), we can always find two (companion) nodes which came from the same pair of nodes in the previous vector. For example, $f_1(0)$ and $f_1(8)$ are computed in terms of $f(0)$ and $f(8)$. Similarly, the companion nodes $f_2(8)$ and $f_2(12)$ are computed from the same pair of nodes $f_1(8)$ and $f_1(12)$.

Furthermore, the computation of companion nodes is independent of other nodes (within the l^{th}-vector). Therefore, the computed $f_1(0)$ and $f_1(8)$ will override the original space of $f(0)$ and $f(8)$. Similarly, the computed $f_2(8)$ and $f_2(12)$ will over ride the space occupied by $f_1(8)$ and $f_1(12)$, which in turn, will occupy the original space of $f(8)$ and $f(12)$. Hence, only one complex vector (or 2 real vectors) of length N are needed for the entire FFT process.

Companion Node Spacing

Observing Figure 2, the following statements can be made:

a) in the first vector $(l = 1)$, the companion nodes $f_1(0)$ and $f_1(8)$ is separated by $k = 8$ (or $\frac{N}{2^l} = \frac{16}{2^1} = 8$) spaces.

b) In the second vector $(l = 2)$, the companion nodes $f_2(8)$ and $f_2(12)$ is separated by $k = 4$ (or $\frac{N}{2^l} = \frac{16}{2^2} = \frac{16}{4}$).

Companion Node Computation

The operation counts in any companion nodes (of the $l^{th} = 2^{nd}$ vector), such as $f_2(8)$ and $f_2(12)$ can be explained as (see Figure 2):

$$f_2(8) = f_1(8) + f_1(12) \times E^4 \tag{16}$$

$$\begin{aligned} f_2(12) &= f_1(8) + f_1(12) \times E^{12} \\ &= f_1(8) + f_1(12) \times E^8 E^4 \\ &= f_1(8) + f_1(12)\left[e^{-i\frac{2\pi}{(N=16)}} \right]^8 E^4 \\ &= f_1(8) + f_1(12)\left[e^{-i\pi} \right] E^4 \\ &= f_1(8) - f_1(12) \times E^4 \end{aligned} \tag{17}$$

Thus, the companion nodes $f_2(8)$ and $f_2(12)$ computation will require 1 complex multiplication and 2 complex additions (see Equations (16-17)). The weighting factors for the companion nodes [$f_2(8)$ and $f_2(12)$] are $E^4 (or\ E^U)$ and $E^{12} (or\ E^{U+\frac{N}{2}})$, respectively.

Thus, in general

$$f_l(k) = f_{l-1}(k) + E^U f_{l-1}(k + \frac{N}{2^l}) \quad (18)$$

$$f_l(k + \frac{N}{2^l}) = f_{l-1}(k) - E^U f_{l-1}(k + \frac{N}{2^l}) \quad (19)$$

Skipping computation of certain nodes

Because the pair of companion nodes "k" and "$k + \frac{N}{2^L}$" are separated by the "distance". ($= \frac{N}{2^L}$), hence, at the L^{th} level, after every $\frac{N}{2^L}$ node computation, then the next $\frac{N}{2^L}$ nodes will be skipped! (see Figure 2).

Determination of E^U

The values of "U" can de determined by the following steps:

Step 1: Express the index k $(= 0,1,2,..., N-1)$ in binary form, using r bits. For $k = 8$, $L = 2$ and $r = 4$; or $N = 2^r = 2^4 = 16$, one obtains

$$\begin{aligned} k &= 8 \\ &= 1,0,0,0 \\ &= (1)2^{r-1=3} + (0)2^2 + (0)2^1 + (0)2^0 \end{aligned}$$

Step2: Sliding this binary number "$r - L = 4 - 2 = 2$" positions to the right, and fill in zeros, the results are

$$1,0,0,0 \rightarrow X, X, 1,0 \rightarrow 0,0,1,0$$

It is important to realize that the results of Step 2 (0,0,1,0) is equivalent to express an integer

$$\begin{aligned} M &= \frac{k}{2^{r-L}} \\ &= \frac{8}{2^{4-2}} \\ &= 2 \end{aligned}$$

in the binary formats. In other words, $M = 2 = (0,0,1,0)$.

Step3: Reverse the order of the bits, then (0,0,1,0) becomes 0,1,0,0 $= E$. Thus,

$$\begin{aligned} U &= (0)2^3 + (1)2^2 + (0)2^1 + (0)2^0 \\ &= 4 \end{aligned}$$

It is "NOT" really necessary to perform Step 3, since the results of Step 2 can be used to compute "E" as following

$$\begin{aligned} U &= (0)2^0 + (0)2^1 + (1)2^2 + (0)2^3 \\ &= 4 \end{aligned}$$

In conclusion, for $N = 2^r = 2^4 = 16; L = 2; k = 8$ and $U = 4$; the computation of companion nodes from general formulas (see Equations (18) and (19)) gives

$$f_2(8) = f_1(8) + E^4 f_1(12)$$

$$f_2(12) = f_1(8) - E^4 f_1(12)$$

The above 2 equations are identical to Equations (16) and (17).

Computer Implementation to Find Value of "U" (in E^U)

Based on the previous discussions (with the 3-step procedures), to find the value of "U", one only needs a procedure to express an integer $M = \frac{k}{2^{r-L}}$ in binary formats, with "r" bits.

Assuming M (a base 10 number) can be expressed as (assuming $r = 4$ bits)

$$\begin{aligned} M &= a_4 a_3 a_2 a_1 \\ &= J_1 \end{aligned} \tag{20}$$

Divide M by 2 (say, $J_2 = \frac{J_1}{2}$), multiply the truncated result by 2 (say, $JJ_2 = J_2 \times 2$), and compute the difference between the original number $= M = J_1$ and JJ_2.

$$IDIFF = J_1 - JJ_2 \left\{ = M - \left(\frac{M}{2}\right)_{Truncated} \times 2 \right\} \tag{21}$$

If $IDIFF = 0,$ then the bit $a_1 = 0$

If $IDIFF \neq 0,$ then the bit $a_1 = 1$

Once the bit a_1 has been determined, the value of J_1 is set to J_2 (or value of J_1 is reduced by a factor of 2), since the previous

$$\begin{aligned} J_1 &= M \\ &= a_4 a_3 a_2 a_1 \\ J_1 &= (a_1)2^0 + (a_2)2^1 + (a_3)2^2 + (a_4)2^3 \end{aligned}$$

and similar process can be used to determine the value of bit a_2, etc.

Example 1

For $k = 8$; $N = 16 = 2^r; r = 4$ bits and $L = 2$, Find the value of U.

$$\begin{aligned} M &= \frac{k}{2^{r-L}} \\ &= \frac{8}{2^{4-2}} \\ &= 2 \\ &= J_1 \end{aligned}$$

Determine the bit a_1: (Index $I = 1$)

Initialize $U = 0$

$$\begin{aligned} J_2 &= \frac{J_1}{2} \\ &= \frac{2}{2} \\ &= 1 \end{aligned}$$

$$\begin{aligned} IDIFF &= J_1 - (JJ_2 = J_2 \times 2) \\ &= 2 - (1)(2) \\ &= 0 \end{aligned}$$

Thus

$$\begin{aligned} a_1 &= 0 \\ U &= U \times 2 + IDIFF \\ &= 0 \times 2 + 0 \\ &= 0 \end{aligned}$$

or

$$\begin{aligned} U &= U + (a_1) 2^{r-1} \\ &= 0 + (0) 2^3 \\ &= 0 \end{aligned}$$

Determine the bit a_2 [Index $I = 2$]

$$\begin{aligned} J_1 &= J_2 \\ &= 1 \\ J_2 &= \frac{J_1}{2} \\ &= \frac{1}{2} \\ &= 0 \\ IDIFF &= J_1 - (JJ_2 = J_2 \times 2) \\ &= 1 - (0 \times 2) \\ &= 1 \end{aligned}$$

Thus $a_2 = 1$

$$\begin{aligned} U &= U \times 2 + IDIFF \\ &= 0 \times 2 + 1 \\ &= 1 \end{aligned}$$

or

$$\begin{aligned} U &= U + (a_2) 2^{r-1} \\ &= 0 + (1) 2^2 \\ &= 4 \end{aligned}$$

Determine the bit a_3 [Index $I = 3$]

$$\begin{aligned} J_1 &= J_2 \\ &= 0 \\ J_2 &= \frac{J_1}{2} \\ &= \frac{0}{2} \\ &= 0 \\ IDIFF &= J_1 - (JJ_2 = J_2 \times 2) \\ &= 0 - (0 \times 2) \\ &= 0 \end{aligned}$$

Thus $a_3 = 0$

$$\begin{aligned} U &= U \times 2 + IDIFF \\ &= 1 \times 2 + 0 \\ &= 2 \end{aligned}$$

or

$$\begin{aligned} U &= U + (a_3)2^{r-I} \\ &= 4 + (0)2^1 \\ &= 4 \end{aligned}$$

Determine the bit a_4 [Index $I = 4 = r$]

$$\begin{aligned} J_1 &= J_2 \\ &= 0 \\ J_2 &= \frac{J_1}{2} \\ &= \frac{0}{2} \\ &= 0 \\ IDIFF &= J_1 - (JJ_2 = J_2 \times 2) \\ &= 0 - (0) \times 2 \\ &= 0 \end{aligned}$$

Thus $a_4 = 0$

$$\begin{aligned} U &= U \times 2 + IDIFF \\ &= 2 \times 2 + 0 \\ &= 4 \end{aligned}$$

or

$$\begin{aligned} U &= U + (a_4)2^{r-I} \\ &= 4 + (0)2^0 \\ &= 4 \end{aligned}$$

Remarks:
Although the "intermediate" results might be different, at the end of the do-loop process (computing a_4), both formulas for "U", such as

$$U = U \times 2 + IDIFF; or \tag{22}$$

$$U = U + (a_I)2^{r-I}; \text{ where } U = 1,2,3,\ldots,r \tag{23}$$

will eventually give the same final answers for "U".

Example 2

For $k = 12$; $N = 16 = 2^{r=4}$; and $L = 3$. Compute the corresponding value of U?

One has

$$\begin{aligned} M &= \frac{k}{2^{r-L}} \\ &= \frac{12}{2^{4-3}} \\ J_1 &= J_2 \\ &= 3 \end{aligned}$$

Determine the bit a_1: (Index $I = 1$)

Initialize $U = 0$

$$\begin{aligned} J_2 &= \frac{J_1}{2} \\ &= \frac{6}{2} \\ &= 3 \end{aligned}$$

$$\begin{aligned} IDIFF &= J_1 - (JJ_2 = J_2 \times 2) \\ &= 6 - (3)(2) \\ &= 0 \end{aligned}$$

Thus

$$a_1 = 0$$

$$\begin{aligned} U &= U \times 2 + IDIFF \\ &= 0 \times 2 + 0 \\ &= 0 \end{aligned}$$

or

$$\begin{aligned} U &= U + (a_1) 2^{r-1} \\ &= 0 + (0) 2^3 \\ &= 0 \end{aligned}$$

Determine the bit a_2 [Index $I = 2$]

$$\begin{aligned} J_1 &= J_2 \\ &= 3 \end{aligned}$$

$$\begin{aligned} J_2 &= \frac{J_1}{2} \\ &= \frac{3}{2} \\ &= 1 \end{aligned}$$

$$\begin{aligned} IDIFF &= J_1 - (JJ_2 = J_2 \times 2) \\ &= 3 - (1) \times 2 \\ &= 1 \end{aligned}$$

Thus $a_2 = 1$

$$\begin{aligned} U &= U \times 2 + IDIFF \\ &= 0 \times 2 + 1 \\ &= 1 \end{aligned}$$

or

$$\begin{aligned} U &= U + (a_2) 2^{r-2} \\ &= 0 + (1) 2^2 \\ &= 4 \end{aligned}$$

Determine the bit a_3 [Index $I = 3$]

$$\begin{aligned} J_1 &= J_2 \\ &= 1 \end{aligned}$$

$$J_2 = \frac{J_1}{2}$$

$$\begin{aligned} &= \frac{1}{2} \\ &= 0 \end{aligned}$$

$$\begin{aligned} IDIFF &= J_1 - (JJ_2 = J_2 \times 2) \\ &= 1 - (0) \times 2 \\ &= 1 \end{aligned}$$

Thus $a_3 = 1$

$$\begin{aligned} U &= U \times 2 + IDIFF \\ &= 1 \times 2 + 1 \\ &= 3 \end{aligned}$$

or

$$\begin{aligned} U &= U + (a_3) 2^{r-3} \\ &= 4 + (1) 2^1 \\ &= 6 \end{aligned}$$

Determine the bit a_4 [Index $I = 4$]

$$\begin{aligned} J_1 &= J_2 \\ &= 0 \\ J_2 &= \frac{J_1}{2} \\ &= \frac{0}{2} \\ &= 0 \end{aligned}$$

$$\begin{aligned} IDIFF &= J_1 - (JJ_2 = J_2 \times 2) \\ &= 0 - (0) \times 2 \\ &= 0 \end{aligned}$$

Thus $a_4 = 0$

$$\begin{aligned} U &= U \times 2 + IDIFF \\ &= 3 \times 2 + 0 \\ &= 6 \end{aligned}$$

or

$$\begin{aligned} U &= U + (a_4) 2^{r-4} \\ &= 6 + (0) 2^0 \\ &= 6 \end{aligned}$$

Remarks:

Although both formulas for “U”, shown in Equations (22) and (23), will yield the same “final” value of “U”. Implementation of Equation (22) will be more computationally efficient.

Unscrambling the FFT

For the case $N = 16 = 2^{r=4}$ (see Figure 2), the final ‘bit-reversing’ operation for FFT is shown in Figure 3.

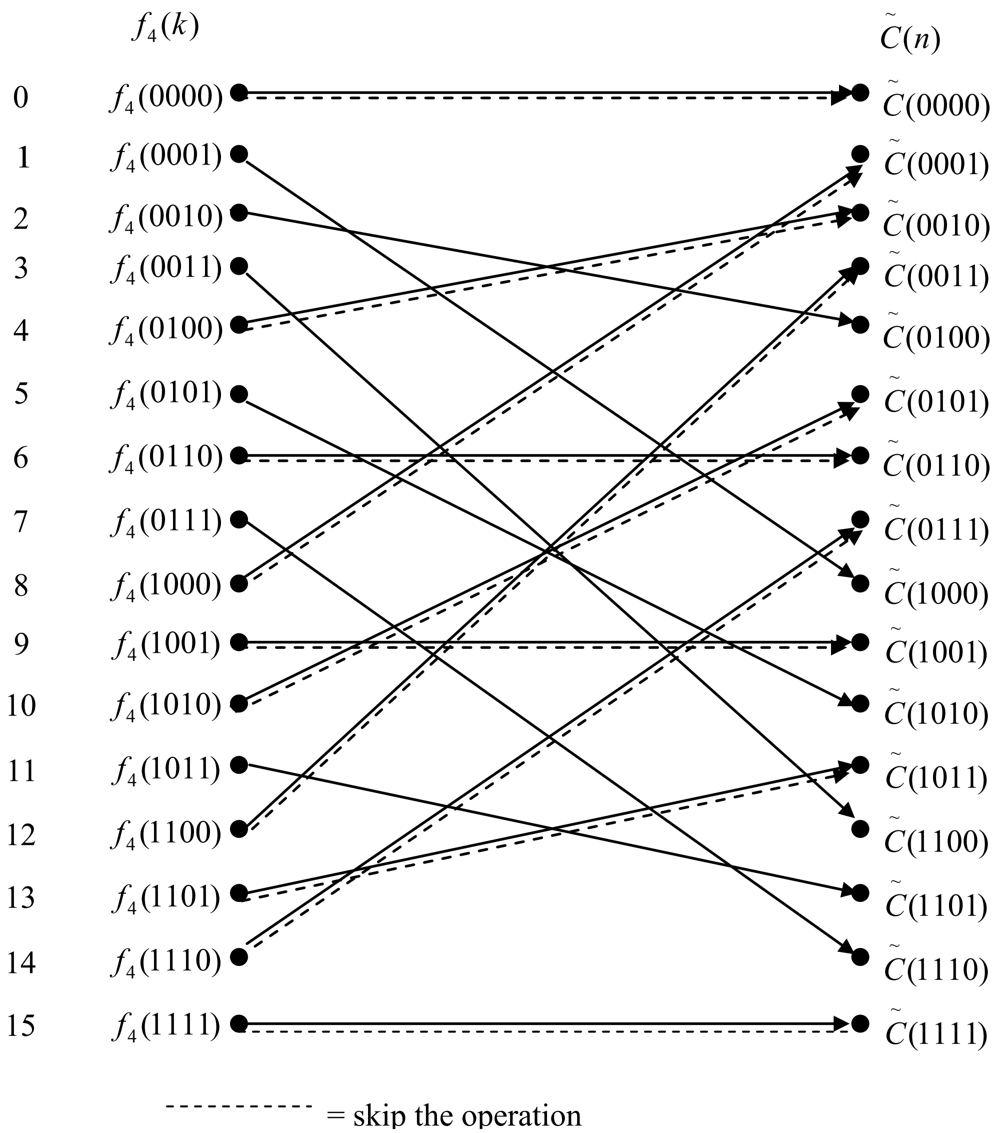

Figure 3 Final "bit-reversing" for FFT (with $N = 2^r = 2^4 = 16$).

For do-loop index $k = 0 = (0,0,0,0) \Rightarrow i = (0,0,0,0) = 0$

```
        $If\,(i.GT.k)Then$
        $T = f_4(k)$
        $f_4(k) = f_4(i)$
        $f_4(i) = T$
        $Endif$
```

Hence, $f_4(0) = f_4(0)$; no swapping.

For $k = 1 = (0,0,0,1) \Rightarrow i = (1,0,0,0) = 0$ =bit-reversion=8

```
$If\,(i.GT.k)Then$
$T = f_4(k)$
$f_4(k) = f_4(i)$
$f_4(i) = T$
$Endif$
```

Hence, $f_4(1) = f_4(8)$; are swapped.

For $k = 2 = (0,0,1,0) \Rightarrow i = (0,1,0,0) = 4$
Hence, $f_4(2) = f_4(4)$; are swapped.

For $k = 3 = (0,0,1,1) \Rightarrow i = (1,1,0,0) = 12$
Hence, $f_4(3) = f_4(12)$; are swapped.

For $k = 4 = (0,1,0,0) \Rightarrow i = (0,0,1,0) = 2$
In this case, since "i" is not greater than "k".
Hence, no swapping, since $f_4(k = 2)$ and $f_4(i = 4)$; had already been swapped earlier.
.
.
. etc…

Computer Implementation of FFT (for case $N = 2^r$).

The pair of companion nodes computation are given by Equations (18, 19). To avoid "complex number" operations, Equation (18) can be computed based on "real number" operations, as following:

$$\left\{f_L^R(k) + if_L^I(k)\right\} = \left\{f_{L-1}^R(k) + if_{L-1}^I(k)\right\} + \left\{E^{U,R} + iE^{U,I}\right\} \times \left\{f_{L-1}^R(k + \frac{N}{2^L}) + if_{L-1}^I(k + \frac{N}{2^L})\right\} \quad (24)$$

In Equation (24), the superscripts R and I denote real and imaginary components, respectively.

Multiplying the last 2 complex numbers, one obtains:

$$\left\{f_L^R(k) + if_L^I(k)\right\} = \left\{f_{L-1}^R(k) + if_{L-1}^I(k)\right\} + \left\{E^{U,R} \times f_{L-1}^R(k + \frac{N}{2^L}) - E^{U,I} \times f_{L-1}^I(k + \frac{N}{2^L})\right\} + i\left\{E^{U,R} \times f_{L-1}^I(k + \frac{N}{2^L}) + E^{U,I} \times f_{L-1}^R(k + \frac{N}{2^L})\right\} \quad (25)$$

Equating the real (and then, imaginary) components on the Left-Hand-Side (LHS), and the Right-Hand-Side (RHS) of Equation (25), one obtains

$$\left\{f_L^R(k)\right\} = \left\{f_{L-1}^R(k)\right\} + \left\{E^{U,R} \times f_{L-1}^R(k + \frac{N}{2^L}) - E^{U,I} \times f_{L-1}^I(k + \frac{N}{2^L})\right\} \quad (26a)$$

$$\left\{f_L^I(k)\right\}=\left\{f_{L-1}^I(k)\right\}+\left\{E^{U,R}\times f_{L-1}^I(k+\frac{N}{2^L})+E^{U,I}\times f_{L-1}^R(k+\frac{N}{2^L})\right\} \tag{26b}$$

Recall Equation (4)

$$E=e^{-i\frac{2\pi}{N}}$$

Hence

$$\begin{aligned} E^U &=\left(e^{-i\frac{2\pi}{N}}\right)^U \\ &=e^{-i\frac{2\pi U}{N}} \\ &=e^{-i\theta} \\ &=\cos(\theta)-i\sin(\theta) \end{aligned} \tag{27}$$

where

$$\begin{aligned} \theta &=\frac{2\pi U}{N} \\ &=\frac{6.28U}{N} \end{aligned} \tag{28}$$

Thus

$$E^{U,R}=\cos(\theta) \tag{29a}$$

$$E^{U,I}=-\sin(\theta) \tag{29b}$$

Substituting Equations (29a, 29b) into Equations (26a, 26b), one gets

$$\left\{f_L^R(k)\right\}=\left\{f_{L-1}^R(k)\right\}+\left\{\cos(\theta)\times f_{L-1}^R(k+\frac{N}{2^L})+\sin(\theta)\times f_{L-1}^I(k+\frac{N}{2^L})\right\} \tag{30a}$$

$$\left\{f_L^I(k)\right\}=\left\{f_{L-1}^I(k)\right\}+\left\{\cos(\theta)\times f_{L-1}^I(k+\frac{N}{2^L})-\sin(\theta)\times f_{L-1}^R(k+\frac{N}{2^L})\right\} \tag{30b}$$

Similarly, the single (complex number) Equation (19) can be expressed as 2 equivalent (real number) equations, such as equations (30a, 30b).

Listing of computer implementation of serial FFT algorithm is given at http://numericalmethods.eng.usf.edu/simulations/mtl/11fft/general_fft.m

References

[1] E.Oran Brigham, The Fast Fourier Transform, Prentice-Hall, Inc. (1974).

[2] S.C. Chapra, and R.P. Canale, Numerical Methods for Engineers, 4th Edition, Mc-Graw Hill (2002).

[3] W.H . Press, B.P. Flannery, S.A. Tenkolsky, and W.T. Vetterling, Numerical Recipies, Cambridge University Press (1989), Chapter 12.

[4] M.T. Heath, Scientific Computing, Mc-Graw Hill (1997).

[5] H. Joseph Weaver, Applications of Discrete and Continuous Fourier Analysis, John Wiley & Sons, Inc. (1983).

FAST FOURIER TRANSFORM	
Topic	Informal Development of Fast Fourier Series
Summary	Textbook notes on the informal development of fast Fourier series
Major	General Engineering
Authors	Duc Nguyen
Date	February 9, 2012
Web Site	http://numericalmethods.eng.usf.edu