Spectral Analysis Of Discrete Signals

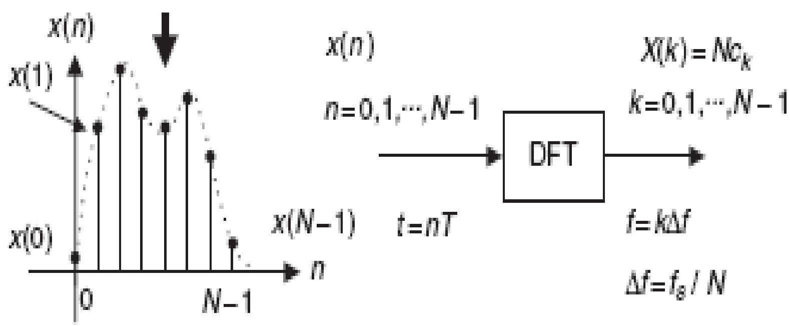
8.1 Discrete Fourier Transform

In time domain, representation of digital signals describes the signal amplitude versus the sampling time instant or the sample number. However, in some applications, signal frequency content is very useful than as digital signal samples.

The algorithm transforming the time domain signal samples to the frequency domain components is known as the discrete Fourier transform, or DFT. The DFT also establishes a relationship between the time domain representation and the frequency domain representation. Therefore, we can apply the DFT to perform frequency analysis of a time domain sequence. In addition, the DFT is widely used in many other areas, including spectral analysis, acoustics, imaging/ video, audio, instrumentation, and communications systems.

8.2 Discrete Fourier Transform Formulas

Given a sequence x(n), $0 \le n \le N - 1$, its **DFT** is defined as:



$$X(k) = \sum_{n=0}^{N-1} x(n)\, e^{-j\frac{2\pi k n}{N}} = \sum_{n=0}^{N-1} x(n)\, W_N^{kn}, \; for\ k = 0,1,..N-1$$

Where the factor W_N (called the twiddle factor in some textbooks) is defined as

$$W_N = e^{-j\frac{2\pi}{N}} = \cos\left(\frac{2\pi}{N}\right) - j\sin\left(\frac{2\pi}{N}\right)$$

The **inverse DFT** is given by:

$$x(n)=\frac{1}{N}\sum_{n=0}^{N-1}X(k)\,e^{j\frac{2\pi kn}{N}}=\frac{1}{N}\sum_{n=0}^{N-1}X(k)\,W_N^{-kn},for\ n=0,1,..N-1$$

Example (1): Given a sequence x(n) for $0\leq n\leq 3$, where x(0) = 1, x(1) = 2, x(2) = 3, and x(3) = 4. Evaluate its DFT X(k).

Solution:

Since N = 4, $W_4 = e^{-j\pi/2}$, then using:

$$X(k)=\sum_{n=0}^{3}x(n)W_4^{kn}=\sum_{n=0}^{3}x(n)e^{-j\frac{\pi kn}{2}}.$$

Thus, for $k=0$

$$X(0)=\sum_{n=0}^{3}x(n)e^{-j0}=x(0)e^{-j0}+x(1)e^{-j0}+x(2)e^{-j0}+x(3)e^{-j0}$$
$$=x(0)+x(1)+x(2)+x(3)$$
$$=1+2+3+4=10$$

for $k=1$

$$X(1)=\sum_{n=0}^{3}x(n)e^{-j\frac{\pi n}{2}}=x(0)e^{-j0}+x(1)e^{-j\frac{\pi}{2}}+x(2)e^{-j\pi}+x(3)e^{-j\frac{3\pi}{2}}$$
$$=x(0)-jx(1)-x(2)+jx(3)$$
$$=1-j2-3+j4=-2+j2$$

for $k = 2$

$$X(2) = \sum_{n=0}^{3} x(n)e^{-j\pi n} = x(0)e^{-j0} + x(1)e^{-j\pi} + x(2)e^{-j2\pi} + x(3)e^{-j3\pi}$$
$$= x(0) - x(1) + x(2) - x(3)$$
$$= 1 - 2 + 3 - 4 = -2$$

and for $k = 3$

$$X(3) = \sum_{n=0}^{3} x(n)e^{-j\frac{3\pi n}{2}} = x(0)e^{-j0} + x(1)e^{-j\frac{3\pi}{2}} + x(2)e^{-j3\pi} + x(3)e^{-j\frac{9\pi}{2}}$$
$$= x(0) + jx(1) - x(2) - jx(3)$$
$$= 1 + j2 - 3 - j4 = -2 - j2$$

Example 2:
Using the DFT coefficients X(k) for $0 \le k \le 3$ computed in Example 1,
a. Evaluate its inverse DFT to determine the time domain sequence x(n).

Solution:

a. Since $N = 4$ and $W_4^{-1} = e^{j\frac{\pi}{2}}$,

$$x(n) = \frac{1}{4}\sum_{k=0}^{3} X(k)W_4^{-nk} = \frac{1}{4}\sum_{k=0}^{3} X(k)e^{j\frac{\pi kn}{2}}.$$

Then for $n = 0$

$$x(0) = \frac{1}{4}\sum_{k=0}^{3} X(k)e^{j0} = \frac{1}{4}\left(X(0)e^{j0} + X(1)e^{j0} + X(2)e^{j0} + X(3)e^{j0}\right)$$
$$= \frac{1}{4}(10 + (-2 + j2) - 2 + (-2 - j2)) = 1$$

for $n = 1$

$$x(1) = \frac{1}{4}\sum_{k=0}^{3} X(k)e^{j\frac{k\pi}{2}} = \frac{1}{4}\left(X(0)e^{j0} + X(1)e^{j\frac{\pi}{2}} + X(2)e^{j\pi} + X(3)e^{j\frac{3\pi}{2}}\right)$$
$$= \frac{1}{4}(X(0) + jX(1) - X(2) - jX(3))$$
$$= \frac{1}{4}(10 + j(-2 + j2) - (-2) - j(-2 - j2)) = 2$$

for $n = 2$

$$x(2) = \frac{1}{4}\sum_{k=0}^{3} X(k)e^{jk\pi} = \frac{1}{4}(X(0)e^{j0} + X(1)e^{j\pi} + X(2)e^{j2\pi} + X(3)e^{j3\pi})$$
$$= \frac{1}{4}(X(0) - X(1) + X(2) - X(3))$$
$$= \frac{1}{4}(10 - (-2 + j2) + (-2) - (-2 - j2)) = 3$$

and for $n = 3$

$$x(3) = \frac{1}{4}\sum_{k=0}^{3} X(k)e^{j\frac{k\pi 3}{2}} = \frac{1}{4}\left(X(0)e^{j0} + X(1)e^{j\frac{3\pi}{2}} + X(2)e^{j3\pi} + X(3)e^{j\frac{9\pi}{2}}\right)$$
$$= \frac{1}{4}(X(0) - jX(1) - X(2) + jX(3))$$
$$= \frac{1}{4}(10 - j(-2 + j2) - (-2) + j(-2 - j2)) = 4$$

- **We can define the frequency resolution as the frequency step between two consecutive DFT coefficients to measure how fine the frequency domain presentation is and achieve**

$$\Delta f = \frac{f_s}{N} \text{ (Hz)}.$$

For example 1, we can Determine the frequency resolution by

$$\Delta f = \frac{f_s}{N} = \frac{10}{4} = 2.5\,\text{Hz}$$

8.3 Fast Fourier Transform

FFT is a very efficient algorithm in computing DFT coefficients and can reduce a very large amount of computational complexity (multiplications). Consider the digital sequence x(n) consisting of 2^m samples, where m is a positive integer—the number of samples of the digital sequence x(n) is a power of 2, N = 2, 4, 8, 16, etc. If x(n) does not contain 2^m samples, then *we simply append it with zeros* until the number of the appended sequence is equal to an integer of a power of 2 data points.

The number of points $N = 2^m$, where the stages $m = \log_2 N$.

In this section, we focus on two formats. One is called the decimation in- frequency algorithm, while the other is the decimation-in-time algorithm. They are referred to as the radix-2 FFT algorithms.

8.3.1 Method of Decimation-in-Frequency (Reduced DIF FFT)

Beginning with the definition of DFT :

$$X(k) = \sum_{n=0}^{N-1} x(n) W_N^{kn} \text{ for } k = 0, 1, \ldots, N-1, \qquad (8.1)$$

Where, $W_N = e^{-j2\pi / N}$ is the twiddle factor, and N = 0, 2, 4, 8, 16, …..Equation (8.1) can be expanded as:

$$X(k) = x(0) + x(1) W_N^k + \ldots + x(N-1) W_N^{k(N-1)}. \qquad (8.2)$$

If we split equation (8.2):

$$\begin{aligned} X(k) = & x(0) + x(1) W_N^k + \ldots + x\left(\frac{N}{2} - 1\right) W_N^{k(N/2-1)} \\ & + x\left(\frac{N}{2}\right) W^{kN/2} + \ldots + x(N-1) W_N^{k(N-1)} \end{aligned}$$

Then we can rewrite as a sum of the following two parts:

$$X(k) = \sum_{n=0}^{(N/2)-1} x(n) W_N^{kn} + \sum_{n=N/2}^{N-1} x(n) W_N^{kn}. \qquad (8.3)$$

Modifying the second term in Equation (8.3) yields:

$$X(k) = \sum_{n=0}^{(N/2)-1} x(n) W_N^{kn} + W_N^{(N/2)k} \sum_{n=0}^{(N/2)-1} x\left(n + \frac{N}{2}\right) W_N^{kn}.$$

Recall $W_N^{N/2} = e^{-j\frac{2\pi(N/2)}{N}} = e^{-j\pi} = -1$; then we have

$$X(k) = \sum_{n=0}^{(N/2)-1} \left(x(n) + (-1)^k x\left(n + \frac{N}{2}\right)\right) W_N^{kn}.$$

Now letting k = 2m as an even number achieves:

$$X(2m) = \sum_{n=0}^{(N/2)-1} \left(x(n) + x\left(n + \frac{N}{2}\right)\right) W_N^{2mn},$$

While substituting k = 2m + 1 as an odd number yields:

$$X(2m+1) = \sum_{n=0}^{(N/2)-1} \left(x(n) - x\left(n + \frac{N}{2}\right) \right) W_N^n W_N^{2mn}.$$

Using the fact that $W_N^2 = e^{-j\frac{2\pi \times 2}{N}} = e^{-j\frac{2\pi}{(N/2)}} = W_{N/2}$, it follows that

$$X(2m) = \sum_{n=0}^{(N/2)-1} a(n) W_{N/2}^{mn} = DFT\{a(n)\ with\ (N/2)\ \text{points}\}$$

$$X(2m+1) = \sum_{n=0}^{(N/2)-1} b(n) W_N^n W_{N/2}^{mn} = DFT\{b(n) W_N^n\ with\ (N/2)\ \text{points}\},$$

Where, a(n) and b(n) are introduced and expressed as:

$$a(n) = x(n) + x\left(n + \frac{N}{2}\right), \text{ for } n = 0,1 \ldots, \frac{N}{2} - 1$$
$$b(n) = x(n) - x\left(n + \frac{N}{2}\right), \text{ for } n = 0,1, \ldots, \frac{N}{2} - 1.$$

$$DFT\{x(n)\ with\ N\ \text{points}\} = \begin{cases} DFT\{a(n)\ with\ (N/2)\ \text{points}\} \\ DFT\{b(n) W_N^n\ with\ (N/2)\ \text{points}\} \end{cases}$$

Figure 8.1(a) illustrates the block diagram of N-point DIF FFT. Fig. 8.1(b) illustrates **reduced** DIF FFT computation for the eight-point DFT, where there are 12 complex multiplications as compared with the eight-point DFT with 64 complex multiplications. For a data length of N, the number of complex multiplications for DFT and FFT, respectively, are determined by:

Complex multiplications of DFT = N^2

Complex multiplications of FFT (*With Reduction*) = $(N / 2) \log_2 (N)$

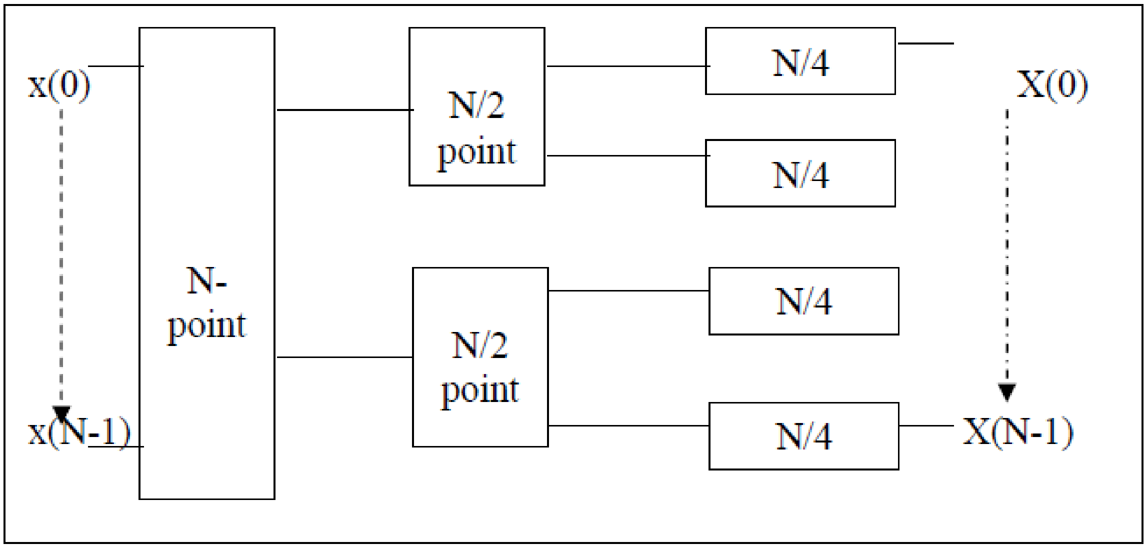


Fig. 8.1(a) Block diagram of DIF FFT

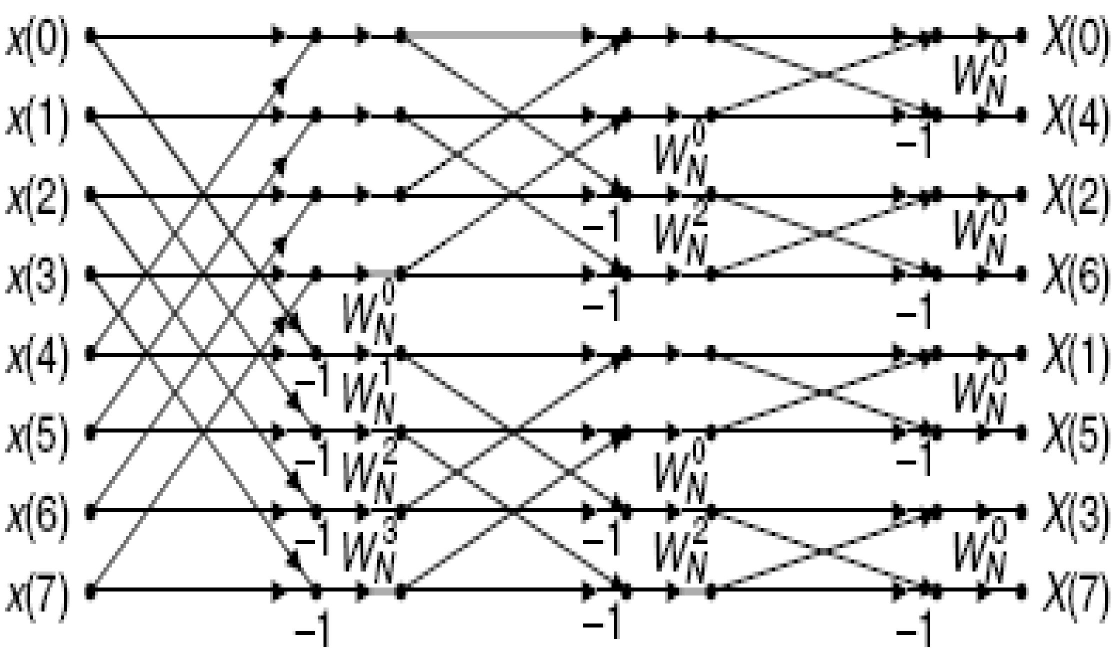


Fig. 8.1(b) The eight-point FFT (total twelve multiplications).

Reduced DIF FFT

The inverse FFT is defined as:

$$x(n) = \frac{1}{N}\sum_{k=0}^{N-1} X(k)W_N^{-kn} = \frac{1}{N}\sum_{k=0}^{N-1} X(k)\tilde{W}_N^{kn}, \text{ for } k = 0, 1, \ldots, N-1.$$

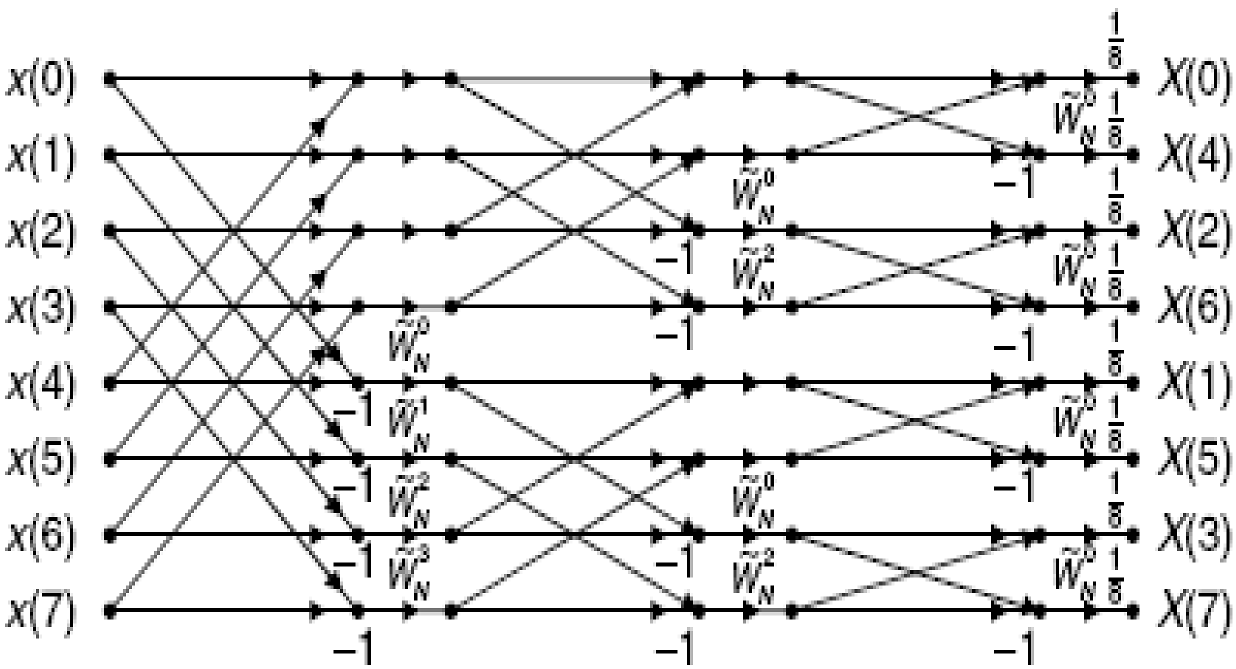


Fig. 8.2 Block diagram for the inverse of eight-point FFT.

Reduced DIF IFFT

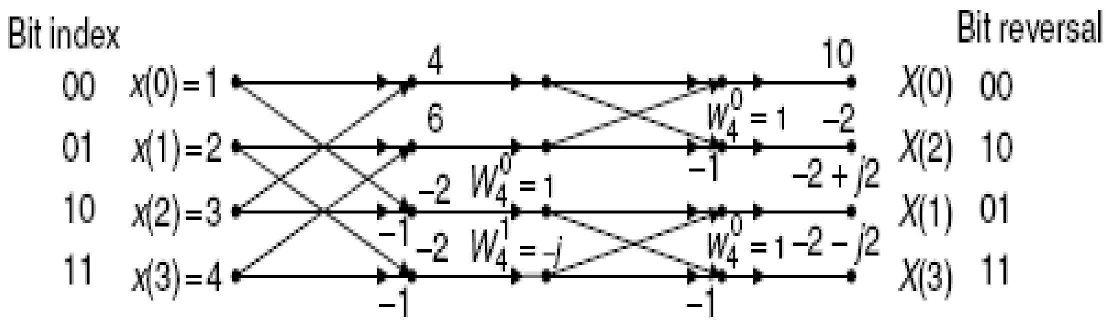
- The twiddle factor $\boldsymbol{W_N}$ is changed to be $\boldsymbol{\widetilde{W_N} = W_N^{-1}}$, and the sum is multiplied by a factor of 1/N. Hence, the inverse FFT block diagram is achieved as shown in Fig. 8.2

Example (3): Given a sequence x(n) for $0 \le n \le 3$, where x(0) = 1, x(1) = 2, x(2) = 3, and x(3) = 4,

a. Evaluate its DFT X(k) using the decimation-in-frequency FFT method.
b. Determine the number of complex multiplications.

Solution:

$$W_4^0 = e^{-j\frac{2\pi}{4}(0)} = 1 \; and \quad W_4^1 = e^{-j\frac{2\pi}{4}(1)} = -j$$



b) The number of complex multiplications is four

H.W Determine the eight-point DFT of the signal

$$x(n) = \{1, 1, 1, 1, 1, 1, 0, 0\}$$

$$\begin{aligned} X(k) &= \sum_{n=0}^{7} x(n) e^{-j\frac{2\pi}{8}kn} \\ &= \{6, -0.7071 - j1.7071, 1 - j, 0.7071 + j0.2929, 0, 0.7071 - j0.2929, 1 + j, \\ &\quad -0.7071 + j1.7071\} \end{aligned}$$

H.W

Given a sequence $x(n)$ for $0 \le n \le 3$, where $x(0) = 1$, $x(1) = 1$, $x(2) = -1$, and $x(3) = 0$, compute its DFT $X(k)$.

8.3.2 Method of Decimation-in-Time (Reduced DIT FFT):

In this method, we split the input sequence x(n) into the even indexed x(2m) and x(2m + 1), each with N data points. Then Equation (8.1) becomes:

$$X(k) = \sum_{m=0}^{(N/2)-1} x(2m) W_N^{2mk} + \sum_{m=0}^{(N/2)-1} x(2m+1) W_N^k W_N^{2mk},$$

for $k = 0, 1, \ldots, N - 1.$

Using $W_N^2 = W_{N/2}$ it follows that:

$$X(k) = \sum_{m=0}^{(N/2)-1} x(2m)W_{N/2}^{mk} + W_N^k \sum_{m=0}^{(N/2)-1} x(2m+1)W_{N/2}^{mk}, \quad \text{....(aa)}$$

for $k = 0, 1, \ldots, N-1.$

Define new functions as:

$$G(k) = \sum_{m=0}^{(N/2)-1} x(2m)W_{N/2}^{mk} = DFT\{x(2m)\ with\ (N/2)\ \text{points}\}$$

$$H(k) = \sum_{m=0}^{(N/2)-1} x(2m+1)W_{N/2}^{mk} = DFT\{x(2m+1)\ with\ (N/2)\ \text{points}\}.$$

$$G(k) = G\left(k + \frac{N}{2}\right), \text{ for } k = 0, 1, \ldots, \frac{N}{2} - 1$$

$$H(k) = H\left(k + \frac{N}{2}\right), \text{ for } k = 0, 1, \ldots, \frac{N}{2} - 1.$$

Substituting above Equations into Equation (aa) yields the first half frequency bins

$$X(k) = G(k) + W_N^k H(k), \text{ for } k = 0, 1, \ldots, \frac{N}{2} - 1.$$

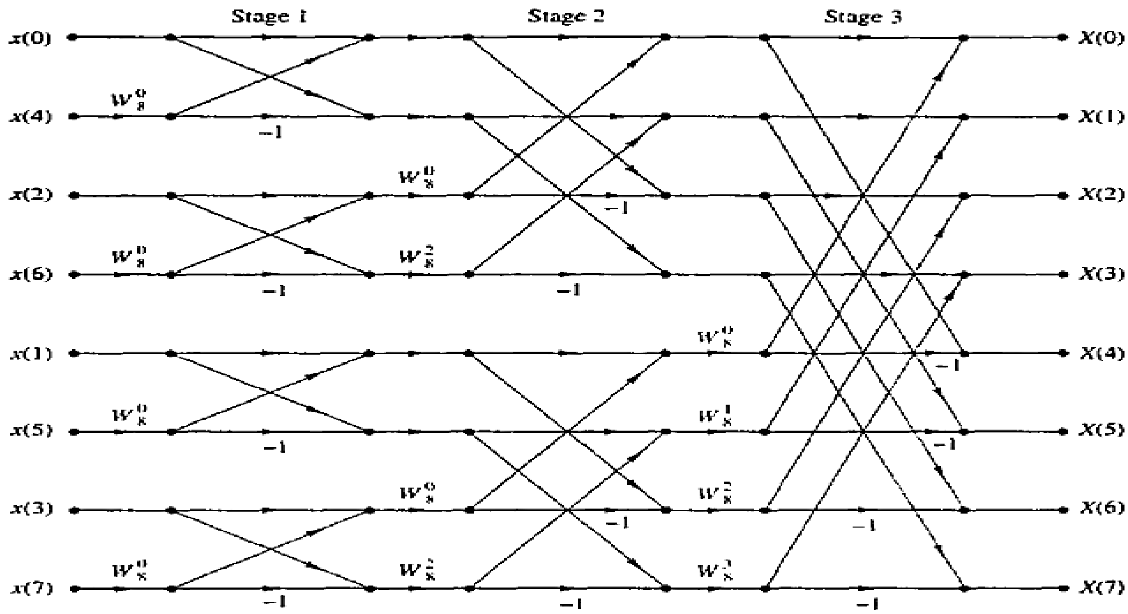
Considering the following fact

$$W_N^{(N/2+k)} = -W_N^k.$$

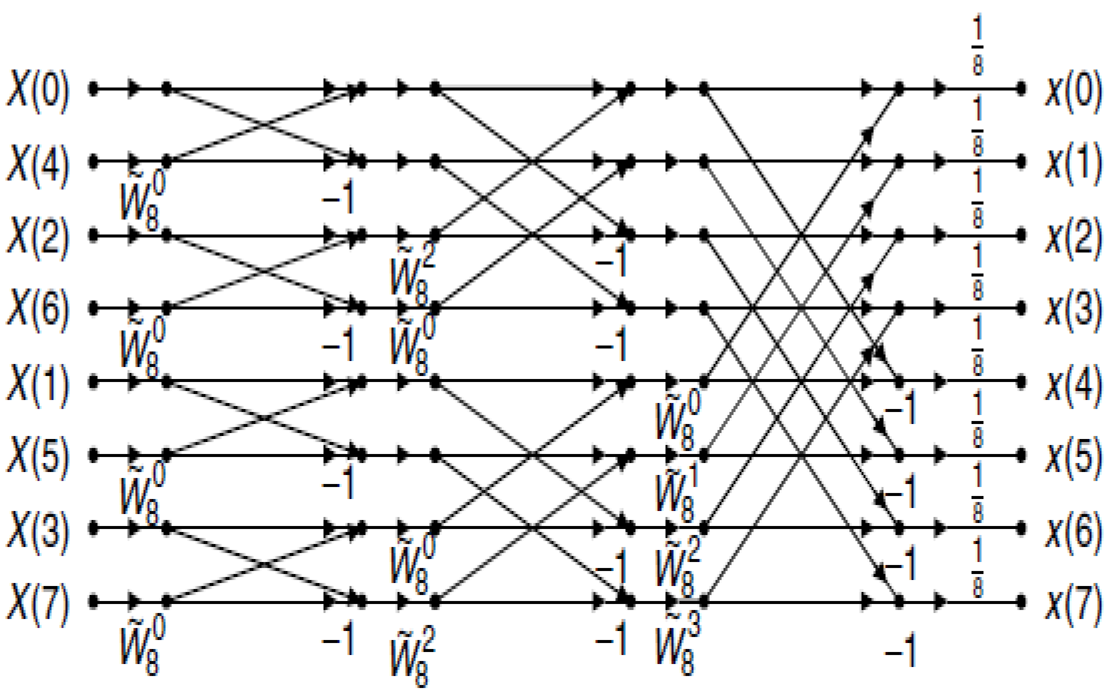
Then the second half of frequency bins can be computed as follows:

$$X\left(\frac{N}{2} + k\right) = G(k) - W_N^k H(k), \text{ for } k = 0, 1, \ldots, \frac{N}{2} - 1.$$

The block diagram for the eight-point DIT FFT algorithm is illustrated in Fig. below

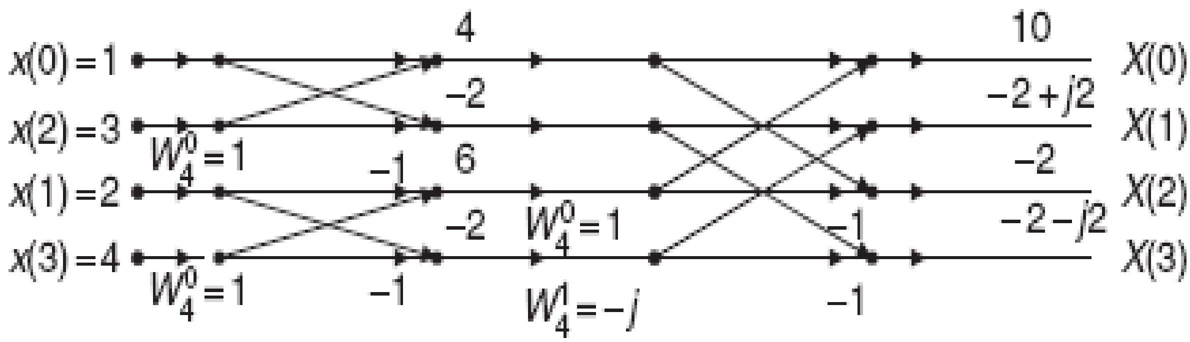


F IGURE 8.3 The eight-point FFT algorithm using decimation-in-time (twelve complex multiplications).



F IGURE 8.4 The eight-point IFFT using decimation-in-time.

Example(4): Given a sequence x(n) for $0 \le n \le 3$, where x(0) = 1, x(1) = 2, x(2) = 3, and x(3) = 4. Evaluate its DFT X(k) using the decimation-in-time FFT method.
Solution:

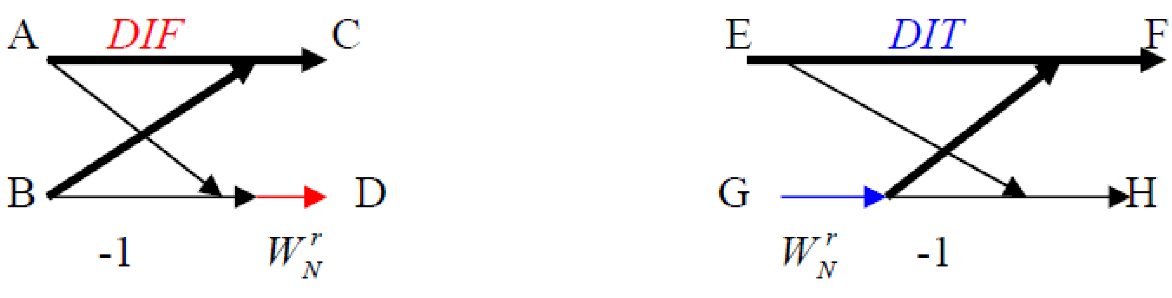


- **We can find FFT by matlab by using fft( )**

H.W Find DFT of the following sequence [1 -1 -1 -1 1 1 1 -1], using:
a) Reduced DIT FFT
b) Reduced DIF FFT

$$Ans: [0 \quad -\sqrt{2} + j(\sqrt{2}+2) \quad 2 - j2 \quad \sqrt{2} + j(\sqrt{2}-2) \quad 4 \quad \sqrt{2} - j(\sqrt{2}-2) \quad 2 + j2 \quad -\sqrt{2} - j(\sqrt{2}+2) \quad]$$

Note: The input sequence is in normal order index and the output frequency bin number is in reversal bits order. The *Butterfly structure* for DIF FFT and DIT FFT is shown below:



C = A+B, D = (A-B) W_N^r , F= E + W_N^r G, H= E - W_N^r G