



وزارة التعليم العالي والبحث العلمي

جامعة ديالى

كلية العلوم

قسم علوم الحاسبات



Fast Fourier Transform (FFT)

Ass. Prof. Dr. Yahiea Al Naiemy

Fast Fourier Transform (FFT)

Fast Fourier Transform (FFT) is a very efficient algorithm in computing DFT coefficients and can reduce a very large amount of computational complexity (multiplications). Without loss of generality, we consider the digital sequence $x(n)$ consisting of $2m$ 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 $2m$ 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.

- In this lecture, we focus on two formats. One is called the Decimation In-Frequency (DIF) algorithm, while the other is the Decimation-In-Time (DIT) algorithm. They are referred to as the radix-2 FFT algorithms.

Method of Decimation-in-Frequency (DIF)

We begin with the definition of DFT

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

The above equation can be expanded as

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

Again, if we split the above equation into

$$X(k) = x(0) + x(1)W_N^k + \dots + x\left(\frac{N}{2} - 1\right)W_N^{k(N/2-1)} \\ + x\left(\frac{N}{2}\right)W_N^{kN/2} + \dots + x(N-1)W_N^{k(N-1)}$$

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

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

Modifying the second term

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

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

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

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

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

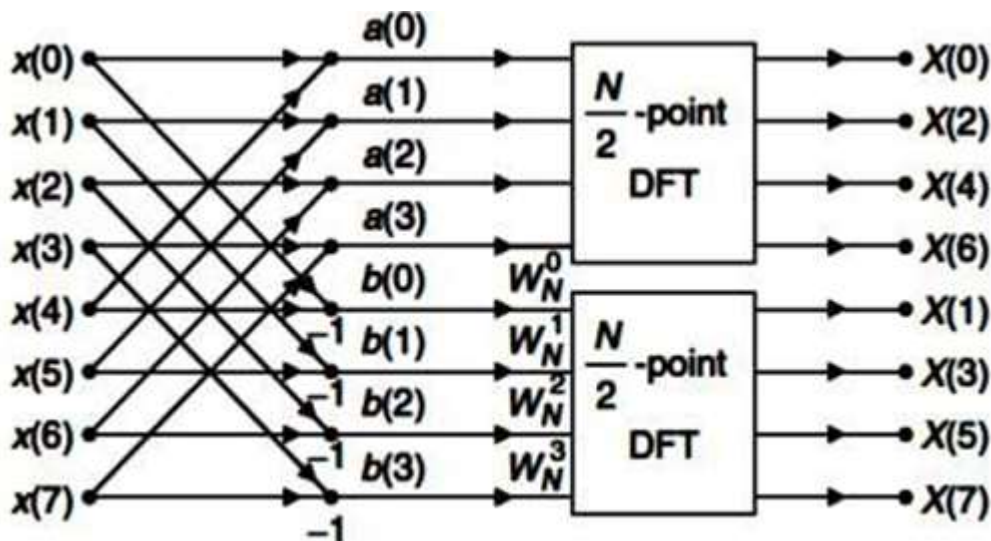
$$X(2m+1) = \sum_{n=0}^{(N/2)-1} b(n) W_N^n W_{N/2}^{mn} = \text{DFT}\{b(n) W_N^n \text{ with } (N/2) \text{ points}\},$$

Where $a(n)$ and $b(n)$ are introduced and expressed as

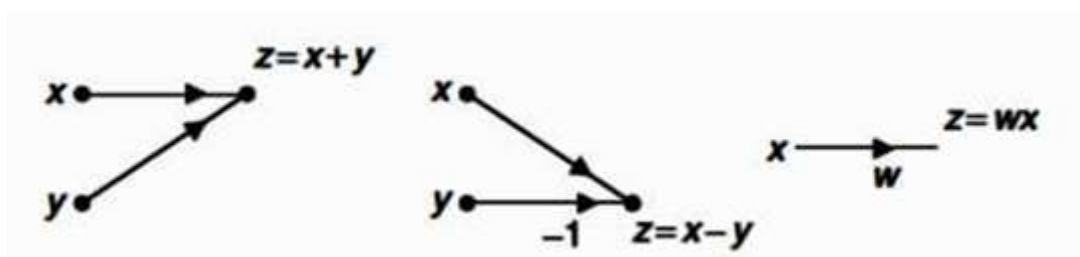
$$a(n) = x(n) + x\left(n + \frac{N}{2}\right), \text{ for } n = 0, 1, \dots, \frac{N}{2} - 1$$

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

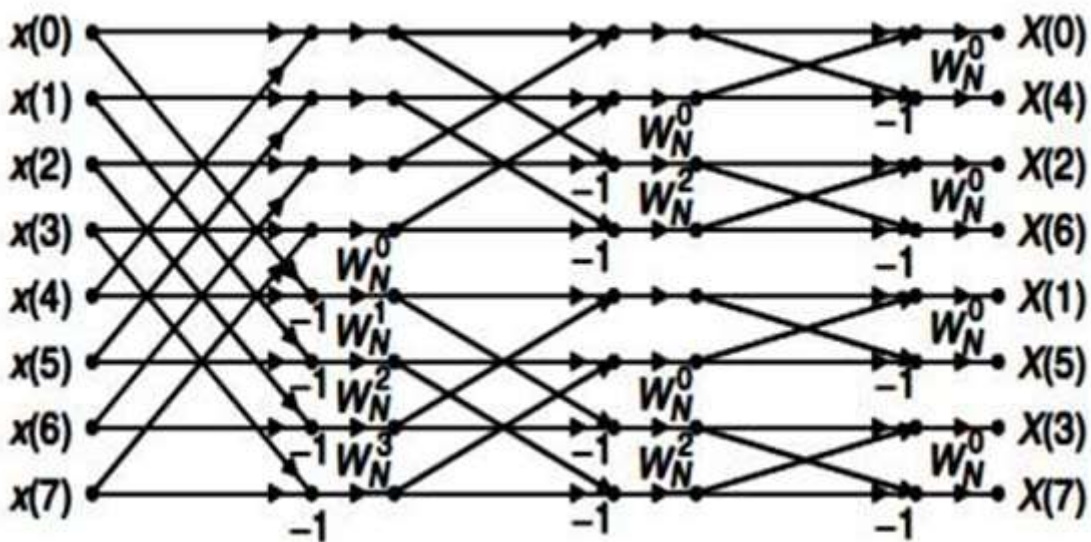
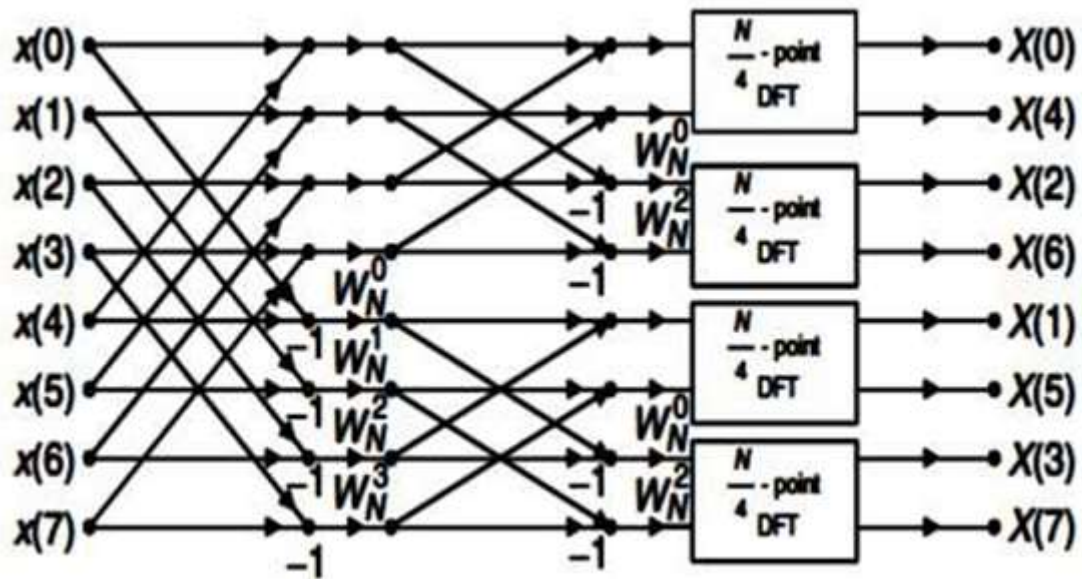
The computation process can be illustrated in the figure below



As shown in this figure, there are three graphical operations, which are illustrated in the figure below



If we continue the process described by the above figure, we obtain the block diagrams shown in following two figures



Complex multiplications of DFT = N^2 , and

Complex multiplications of FFT = $\frac{N}{2} \log_2 (N)$.

Next, the index (bin number) of the eight-point DFT coefficient $X(k)$ becomes 0, 4, 2, 6, 1, 5, 3, and 7, respectively, which are not in the natural order. This can be fixed by index matching. Index matching between the input sequence and the output frequency bin number by applying reversal bits is described in table shown below

Input Data	Index Bits	Reversal Bits	Output Data
$x(0)$	000	000	$X(0)$
$x(1)$	001	100	$X(4)$
$x(2)$	010	010	$X(2)$
$x(3)$	011	110	$X(6)$
$x(4)$	100	001	$X(1)$
$x(5)$	101	101	$X(5)$
$x(6)$	110	011	$X(3)$
$x(7)$	111	111	$X(7)$

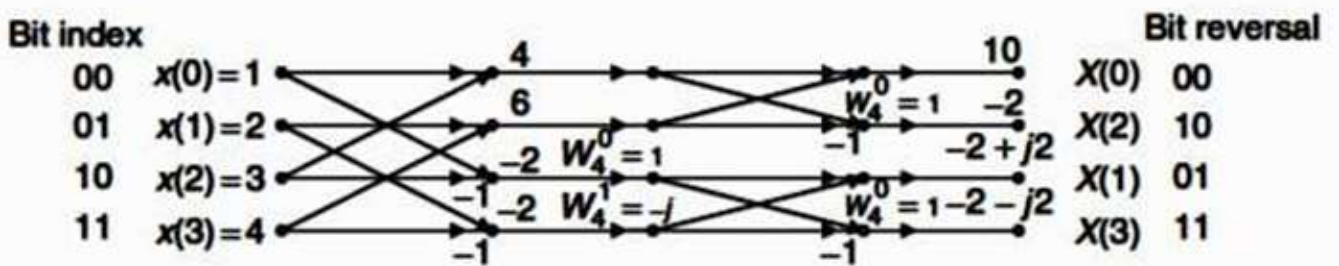
Example:

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

- a. Evaluate its DFT $X(k)$ using the Decimation-in-Frequency DIF FFT method.
- b. Determine the number of complex multiplications.

Solution:

- a. Using the FFT block diagram shown in the figure below



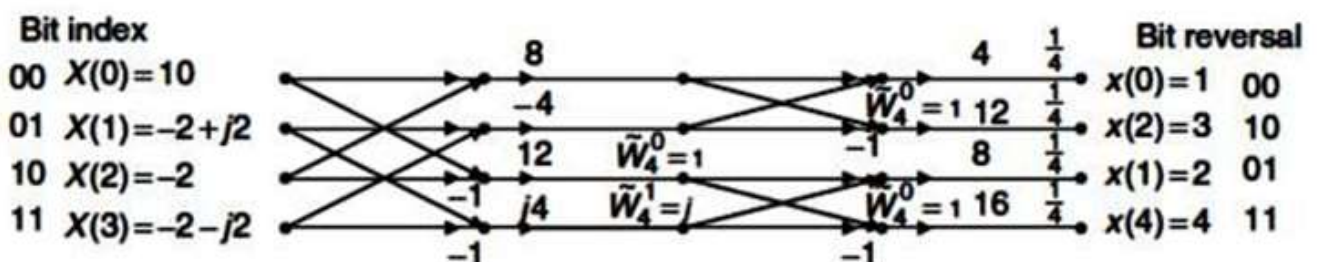
- b. From above figure, the number of complex multiplications is *four*, which can also be determined by

$$\frac{N}{2} \log_2(N) = \frac{4}{2} \log_2(4) = 4.$$

Example: Given the DFT sequence $X(k)$ for $0 \leq k \leq 3$ computed in previous example.

Evaluate its inverse DFT $x(n)$ using the Decimation-in-Frequency FFT (DIF FFT) method

Solution:



Method of Decimation-in-Time FFT(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.

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

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

Using the relation

$$W_N^2 = W_{N/2},$$

It follows that

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

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

Define new functions as

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

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

Note that

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

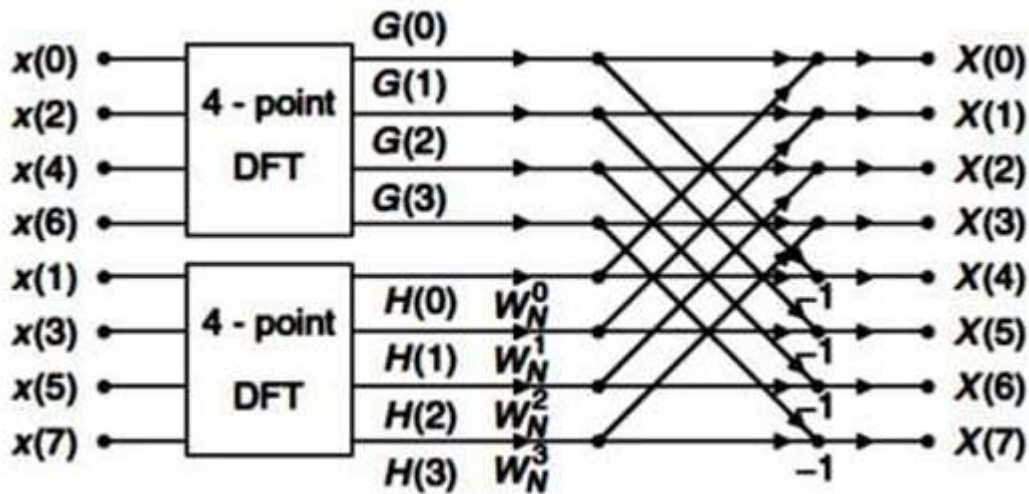
$$H(k) = H\left(k + \frac{N}{2}\right), \text{ for } k = 0, 1, \dots, \frac{N}{2} - 1.$$

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

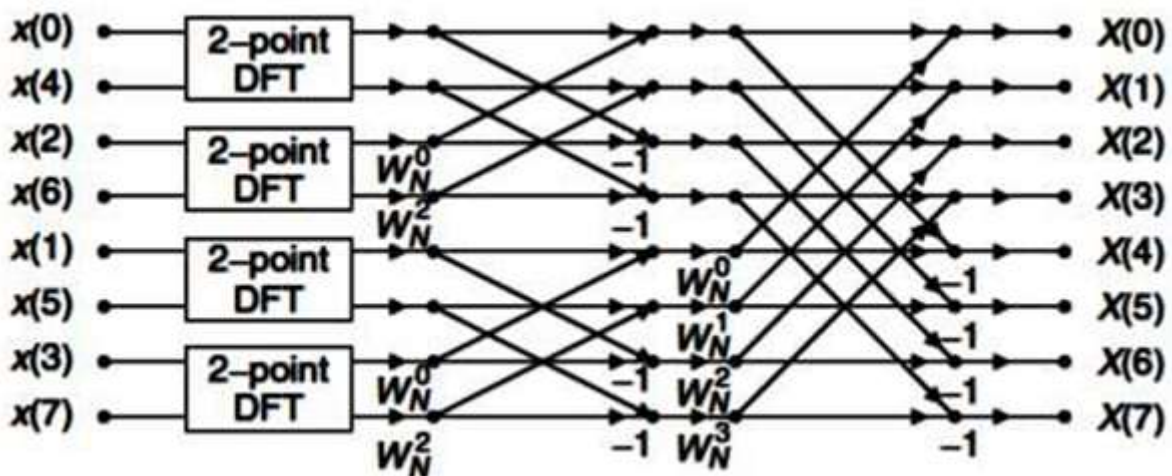
$$W_N^{(N/2+k)} = -W_N^k.$$

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

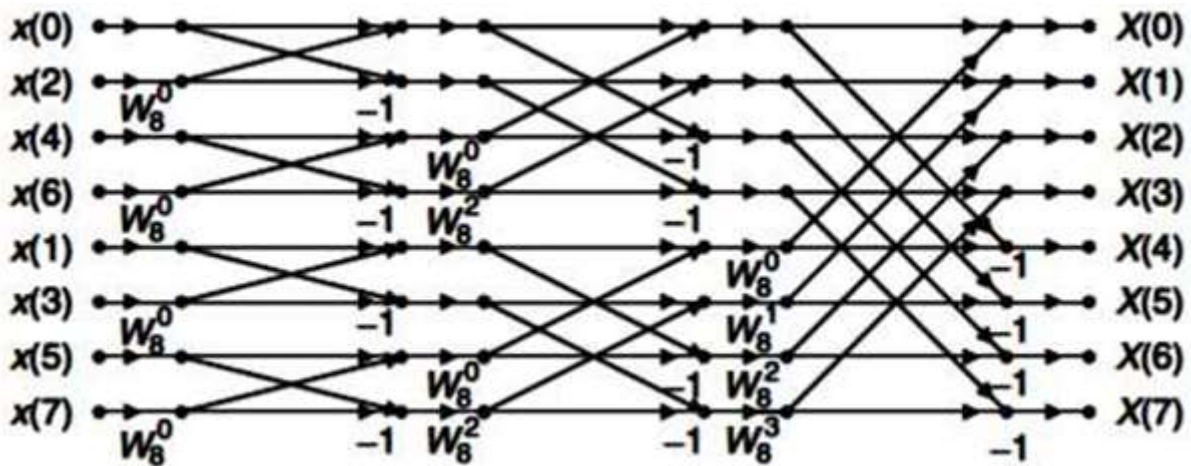
If we perform backward iterations, we can obtain the FFT algorithm. The procedure is illustrated in the figure below, the block diagram for the eight-point FFT algorithm.



From a further iteration, we obtain the figure shown below.



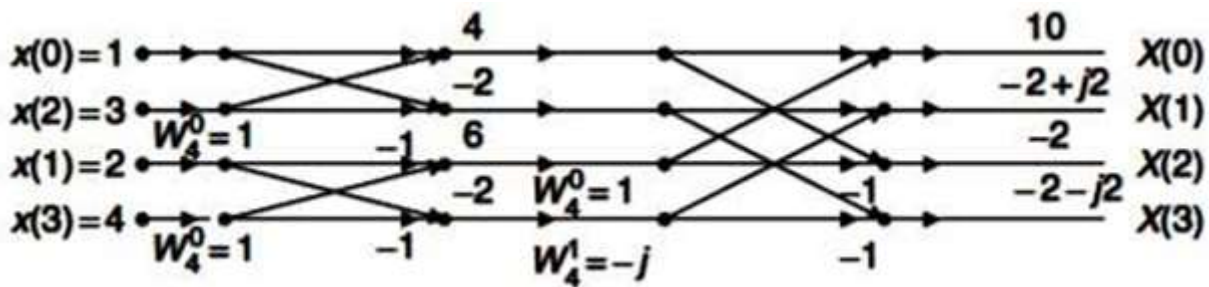
Finally, after three recursions, we end up with the block diagram in the figure below



Example:-

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 $X(k)$ using the Decimation-in-Time FFT (DIT FFT) method.

Solution:



Example:-

Given the DFT sequence $X(k)$ for $0 \leq k \leq 3$ computed in previous example, evaluate its inverse DFT $x(n)$ using the Decimation-in-Time FFT (DIT FFT) method

Solution:-

