COMPUTATION OF DFT USING FFT ALGORITHM

FAST FOURIER TRANSFORM (FFT)

The Fast Fourier transform is a method for computing the discrete Fourier transform with reduced number of calculations. The Computational efficiency is achieved if we adopt a divide and conquer approach. This approach is based on the decomposition of an N point DFT into successively smaller DFTs.

Radix-2 FFT

In an N-point sequence if N can be expressed as $N=2^m$ then the sequence can be dissipated into 2-point sequences. For each 2-point sequence, 2-point DFT can be computed. From the result of 2-point DFT the 4-point DFT can be calculated. This FFT algorithm is called radix-2 FFT.In computing N-point DFT requires 'm' number of stages of computation $N=2^m$

Number of Calculations in N-point DFT:

$$X(K) = \sum_{n=0}^{N-1} x(n) e^{\frac{-j2\pi kn}{N}}$$

For k=0, 1, 2 ….N-1

$$X(k) = x(0)\, e^{0} + x(1)\, e^{\frac{-j2\pi k}{N}} + x(2)\, e^{\frac{-j4\pi k}{N}} + x(3)\, e^{\frac{-j6\pi k}{N}} + \ldots + x(N-1)\, e^{\frac{-j2(N-1)\pi k}{N}}$$

From the above equation we can say that

The numbers of calculations to calculate X (k) for one values of k are,

N number of Complex multiplications and

N-1number of Complex additions.

The X (k) is a sequence consisting of N complex numbers.

Therefore, the number of calculations to calculate all the N complex numbers of the X (k) are,

$N\times N = N^2$ number of complex multiplications and

$N\times (N-1) = N(N-1)$ number of complex additions

Hence, in direct computations of N point DFT, the total numbers of complex additions are N (N-1) and total number of complex multiplications are N^2.

Number of Calculations in Radix-2 FFT:

In radix 2 FFT, $N=2^m$ and so there will be m stages of computations, where $m=\log_2 N$, with each stage having N/2 butterflies.

The number of calculations in one butterflies are

1. Number of complex multiplications and

2. Number of complex additions.

There are N/2 butterflies in each stage.

Therefore, number of calculations in one stage is,

$\frac{N}{2} \times 1 = \frac{N}{2}$ Complex multiplications

$\frac{N}{2} \times 2 = N$ Complex additions.

The N-point DFT involves m stages of computations.Therefore, the number of calculations for m stages are,

$M \times \frac{N}{2} = \log_2 N \times \frac{N}{2} = \frac{N}{2} \log_2 N$ complex multiplications and

$m \times N = \log_2 N \times N = N \log_2 N$ complex additions

Phase or twiddle factor:

By the definition of DFT, the N point DFT is given by

$$X(k) = \sum_{n=0}^{N-1} x(n) e^{\frac{-j2\pi nk}{N}} \qquad \text{for } k=0,1,2,3\ldots\ldots\ldots\ldots N-1$$

To simplify the notation it is desirable to define the complex valued phase factor W_N which is an N^{th} root of unity as,

$$W_N = e^{-j2\pi}$$

The phase value of -2π of W can be multiplied by any integer and it is represented as prefix in W.For example multiplying -2π by k can be represented as W^k.

$$e^{-j2\pi k} \quad \Rightarrow W^k$$

The phase value -2π of W can be divided by any integer and it is represented as suffix in W.For example dividing -2π by N can be represented as W_N.

$$e^{-j2\pi \div N} = e^{-j2\pi \times \frac{1}{N}} \quad \Rightarrow W_N$$

$$e^{\frac{-j2\pi nk}{N}} = (e^{-j2\pi})^{\frac{nk}{N}}$$

The equation of N point DFT using phase factor can be written as

$$X(k) = \sum_{n=0}^{N-1} x(n)\, W_N^{kn} \; ; \; \text{for } k = 0,1,2,\ldots\ldots N-1$$

DECIMATION IN TIME (DIT) RADIX 2 FFT:

Decimation in Time (DIT) Radix 2 FFT algorithm converts the time domain N point sequence x(n) to a frequency domain N-point sequence X(k).In Decimation in Time algorithm the time domain sequence x(n) is decimated and smaller point DFT are performed. The results of smaller point DFTs are combined to get the result of N-point DFT.

In DIT radix -2 FFT the time domain sequence is decimated into 2-point sequences. For each 2-point sequence,2 -point DFT can be computed. From the result of 2-point DFT the 4-point DFT can be calculated. From the result of 4-point DFT the 8-point DFT can be calculated. This process is continued until we get N point DFT.This FFT algorithm is called radix-2 FFT.

In decimation in time algorithm the N point DFT can be realized from two numbers of N/2 point DFTs, The N/2 point DFT can be calculated from two numbers of N/4-point DFTs and so on.

Let x (n) be N sample sequence, we can decimate x (n) into two sequences of N/2 samples. Let the two sequences be f_1 (n) and f_2 (n).Let f_1 (n) consists of even numbered samples of x (n) and f_2(n) consists of odd numbered samples of x(n).

$f_1(n) = x(2n)$ for n=0,1,2,3………… $\frac{N}{2} - 1$

$f_2(n) = x(2n+1)$ for n=0,1,2,3………… $\frac{N}{2} - 1$

Let X (k) = N-point DFT of x (n)

F_1 (k) = N/2 point DFT of f_1(n)

F_2 (k) = N/2 point DFT of f_2(n)

By definition of DFT the N/2 point DFT of f_1(n) and f_2(n) are given by

$$F_1(k) = \sum_{n=0}^{\frac{N}{2}-1} f_1(n) W_{N/2}^{kn};$$

$$F_2(k) = \sum_{n=0}^{\frac{N}{2}-1} f_2(n) W_{N/2}^{kn}$$

Now-point DFT X (k), in terms of N/2 point DFTs F_1 (k) and F_2 (k) is given by

$$X(k) = F_1(k) + W_N^k\, F_2(k), \text{ where, } k = 0,1,2,\ldots\ldots(N-1)$$

Having performed the decimation in time once, we can repeat the process for each of the sequences f_1 (n) and f_2 (n).Thus $f_1(n)$ would result in the two N/4 point sequences and $f_2(n)$ would result in another two N/4 point sequences.

Let the decimated N/4 point sequences of f_1 (n) be V_{11} (n) and V_{12} (n).

$V_{11}(n) = f_1(2n)$; for n= 0, 1, 2,….. $\frac{N}{4} - 1$

$V_{12}(n) = f_1(2n+1)$; for n= 0,1,2,…..$\frac{N}{4} - 1$

Let the decimated N/4 point sequences of $f_2(n)$ be $V_{21}(n)$ and $V_{22}(n)$.

$V_{21}(n) = f_1(2n)$; for n= 0,1,2,…..$\frac{N}{4} - 1$

$V_{22}(n) = f_1(2n+1)$; for n= 0,1,2,…..$\frac{N}{4} - 1$

Let $V_{11}(k)$ = N/4 point DFT of $V_{11}(n)$;

$V_{12}(k)$ = N/4 point DFT of $V_{12}(n)$

$V_{21}(k)$ = N/4 point DFT of $V_{21}(n)$

$V_{22}(k)$ = N/4 point DFT of $V_{22}(n)$

Then like earlier analysis we can show that,

$$F_1(k) = V_{11}(k) + W^k_{N/2} V_{12}(k) \text{ ; for } k = 0,1,2,3,\ldots\ldots\frac{N}{2} - 1$$

$$F_2(k) = V_{21}(k) + W^k_{N/2} V_{22}(k) \text{ ; for } k = 0,1,2,3,\ldots\ldots\frac{N}{2} - 1$$

Hence the N/2 point DFTs are obtained from the results of N/4 point DFTs.

The decimation of the data sequence can be repeated again and again until the resulting sequences are reduced to 2-point sequences.

Flow graph for 8 point DFT using radix 2 DIT FFT

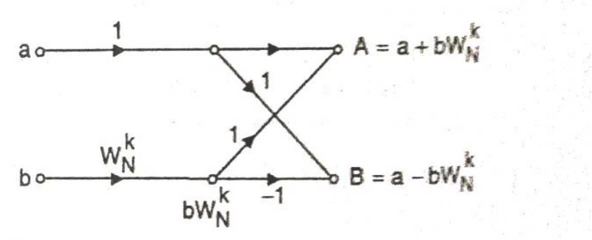


Fig.Basic Butterfly computation

In each computation two complex numbers "a" and "b" are considered. The complex number "b" is multiplied by a phase factor "W_N^k "The product "b W_N^k" is added to complex number “a” to form new complex number “A”. The product “b W_N^k” is subtracted from complex number “a” to form new complex number”B”.

The input sequence is 8 point sequence.Therefore, N $=8=2^3=r^m$.Here r=2 and m=3.The sequence x (n) is arranged in bit reversed order and then decimated into two sample sequences.

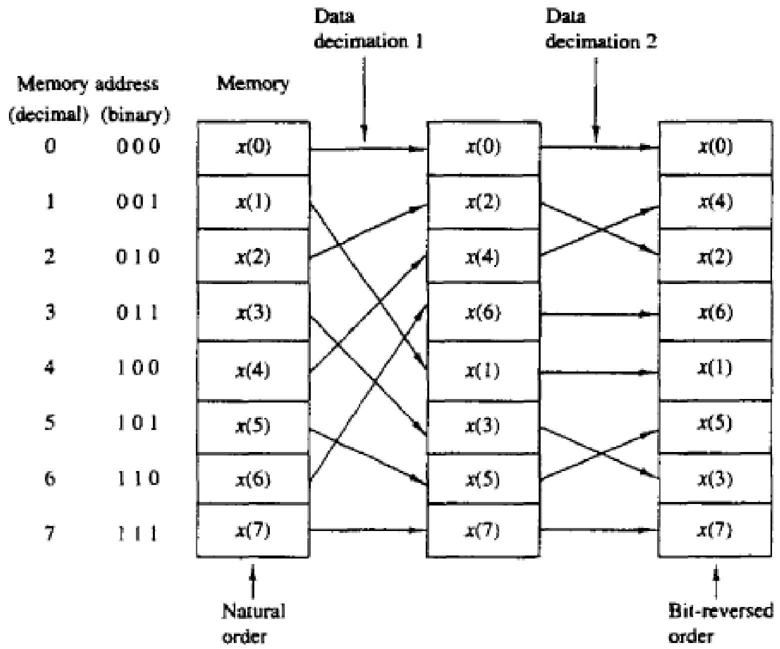


Fig.Bit reversed order

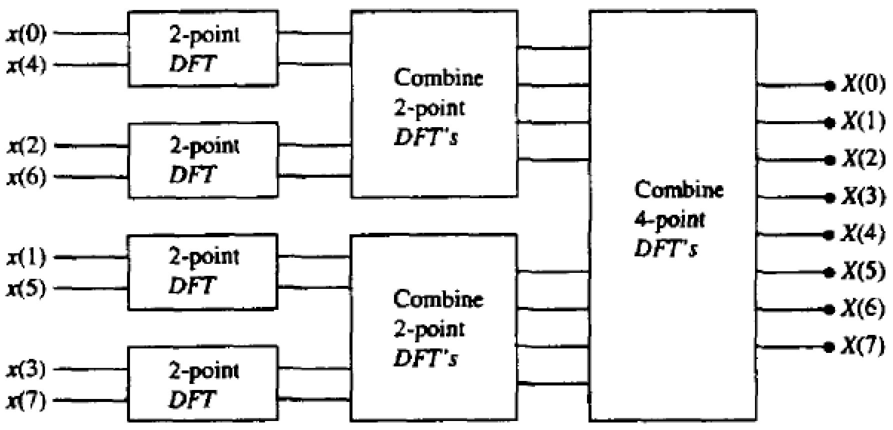


Fig . Three stages in the computation of an N = 8 point

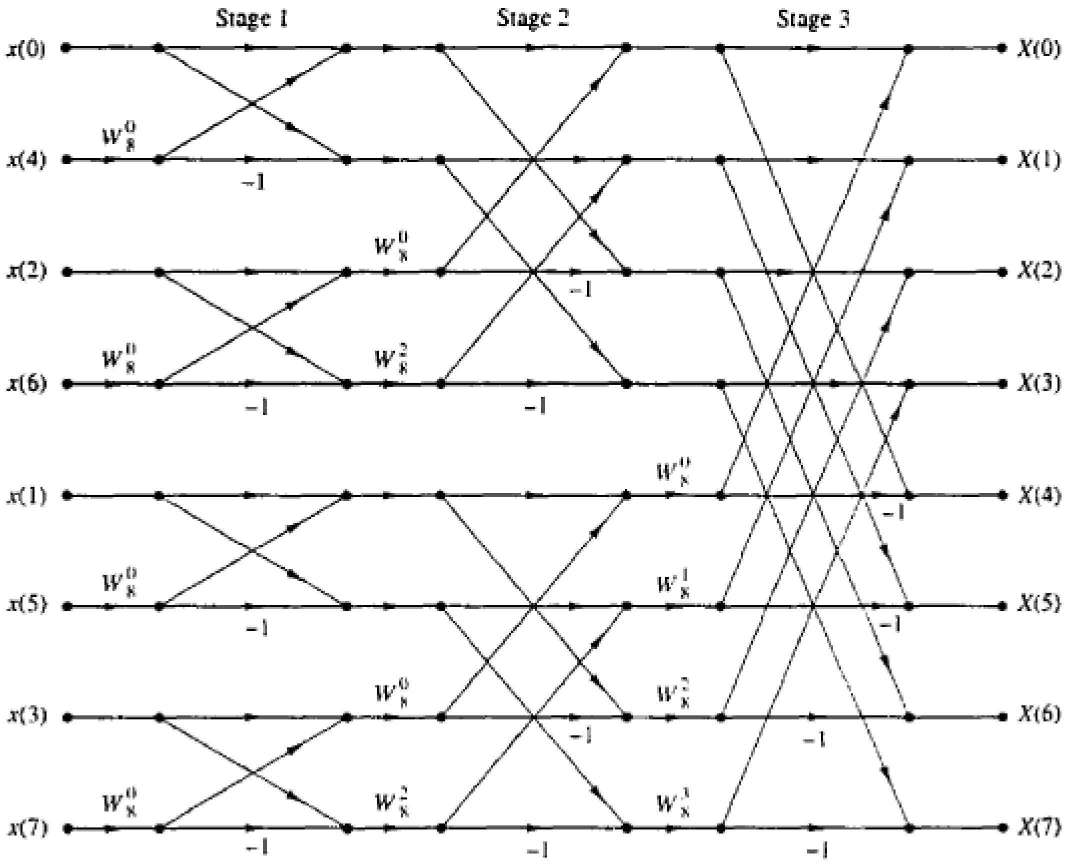


Fig . Eight point Decimation In Time-FFT

DECIMATION IN FREQUENCY (DIF) RADIX 2 FFT:

In Radix-2 decimation-in-frequency (DIF) FFT algorithm, original sequence s(n) is decomposed into two subsequences as first half and second half of a sequence. There is no need of reordering (shuffling) the original sequence as in Radix-2 decimation-in-time (DIT) FFT algorithm.

In this algorithm the N-point time domain sequence is converted into two numbers of N/2 sequences. Then each N/2 point sequence is converted into two numbers of N/4 point sequences. Thus we get four numbers of N/4 point sequences. This process is continued until we get N/2 numbers of 2-point sequences.

It can be shown that the N-point DFT of x(n) can be realized from two numbers of N/2 point DFTs.The N/2 point DFTs can be realized from two numbers of N/4 point DFTs and so on. The decimation is continued up to 2-point DFTs.

Flow graph for 8 point DFT using Radix-2 DIF FFT

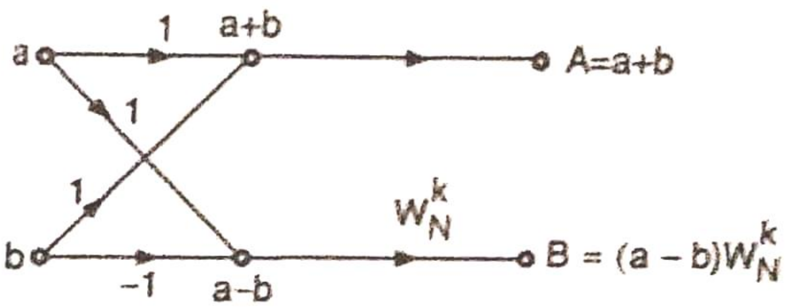


Fig .Basic Butterfly Computation

In each computation two complex numbers “a” and “b” are considered.

The sum of the two complex numbers is computed which forms a new complex number “A”.

Then subtract complex number “b” from “a” to get the term “a-b”.The difference term “a-b”is multiplied with the phase factor “W_N^k “ to form a new complex number “B”.

Let x (n) and X(k) be N-point DFT pair.

Let G_1 (k) and G_2 (k) be two numbers of N/2 point sequences obtained by the decimation of X (k).

Let G_1 (k) be N/2 point DFT of $g_1(n)$ and $G_2(k)$ be N/2 point DFT of $g_2(n)$.

Now, the N point DFT X(k) can be obtained from the two numbers of N/2 point DFTs of $G_1(k)$ and $G_2(k)$ as shown below.

$$X(k)\,|_{k=even} = G_1(k)$$

$$X(k)\,|_{k=odd} = G_2(k)$$

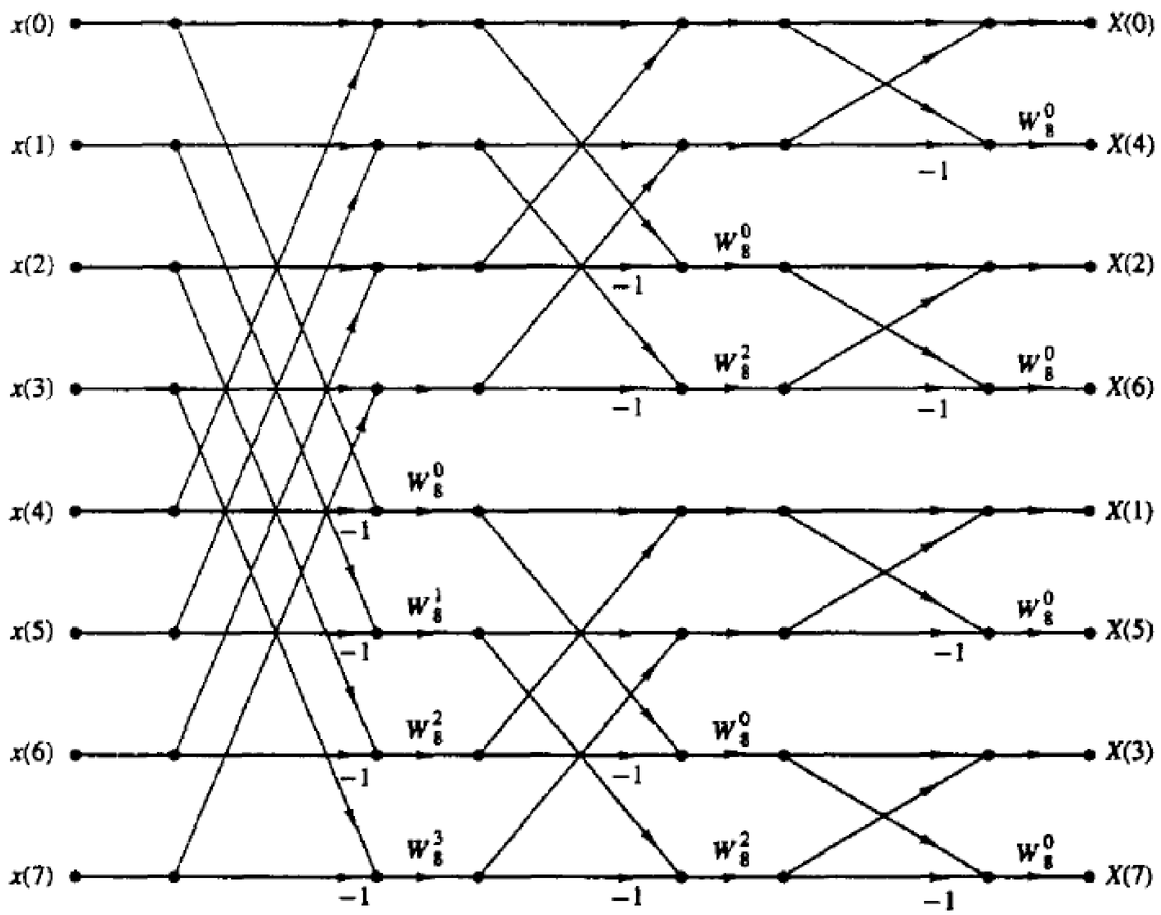


Fig.N=8 point decimation in frequency FFT algorithm.

In the next stage of decimation the N/2 point frequency domain sequence G_1 (k) is decimated into two numbers of N/4 point sequences D_{11} (k) and D_{12} (k), and G_2 (k) is decimated into two numbers of N/4 point sequences D_{21} (k) and D_{22} (k).

Let D_{11} (k) and D_{12} (k) be two numbers of N/4 point sequences obtained by the decimation of G_1 (k).

Let D_{11} (k) be N/4 point DFT of $d_{11}(n)$,and $D_{12}(k)$ be N/4 point DFT of $d_{12}(n)$.

Let D_{21} (k) and D_{22} (k) be two numbers of N/4 point sequences obtained by the decimation of G_2 (k).

Let D_{21} (k) be N/4 point DFT of d_{21} (n) and D_{22} (k) be N/4 point DFT of d_{22} (n).

Now, N/2 point DFTs can be obtained from two numbers of N/4 point DFTs as shown below.

$$G_1(k)\,|_{k=even} = D_{11}(k)$$

$$G_1(k)\,|_{k=odd} = D_{12}(k)$$

$$G_2(k)\,|_{k=even} = D_{21}(k)$$

$$G_2(k)\,|_{k=odd} = D_{22}(k)$$

The decimation of the frequency domain sequence can be continued until the resulting sequences are reduced to 2-point sequences. The entire process of decimation involves,m stages of decimation where $m = \log_2 N$.The computation of the N-point DFT via the decimation in frequency FFT algorithm requires $(N/2)\log_2 N$ Complex multiplications and $N\log_2 N$ complex additions.

1.Compute an 8 point DFT of the sequence using DIT and DIF-FFT algorithm.

x(n)=(1, 2 , 3 , 2 , 1 , 0).

↑

$\therefore x(n) = \{3, 2, 1, 0, 0, 0, 1, 2\}.$

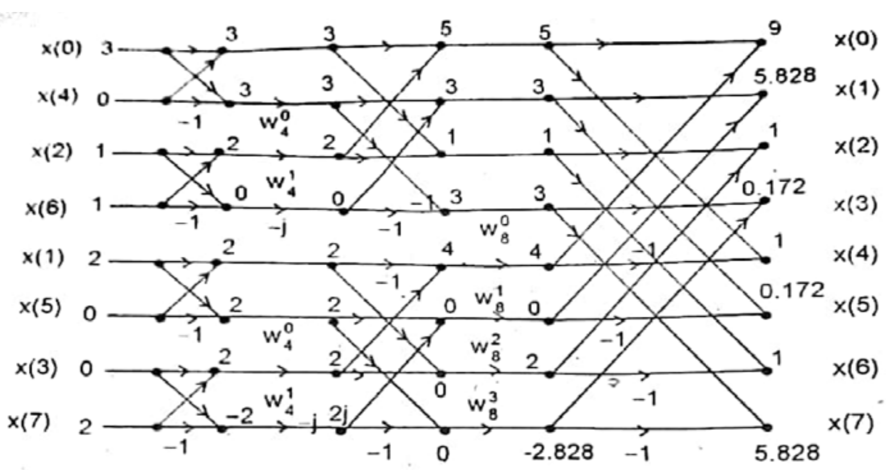
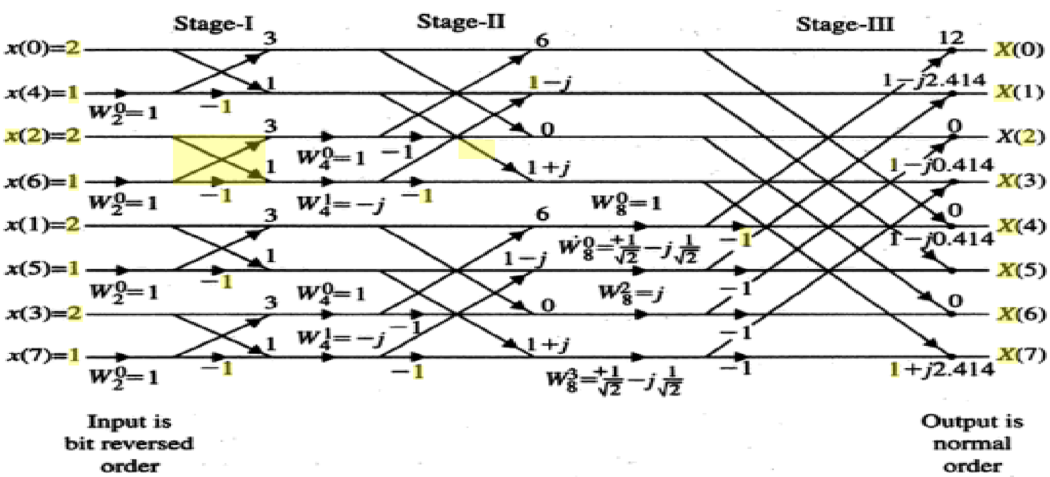


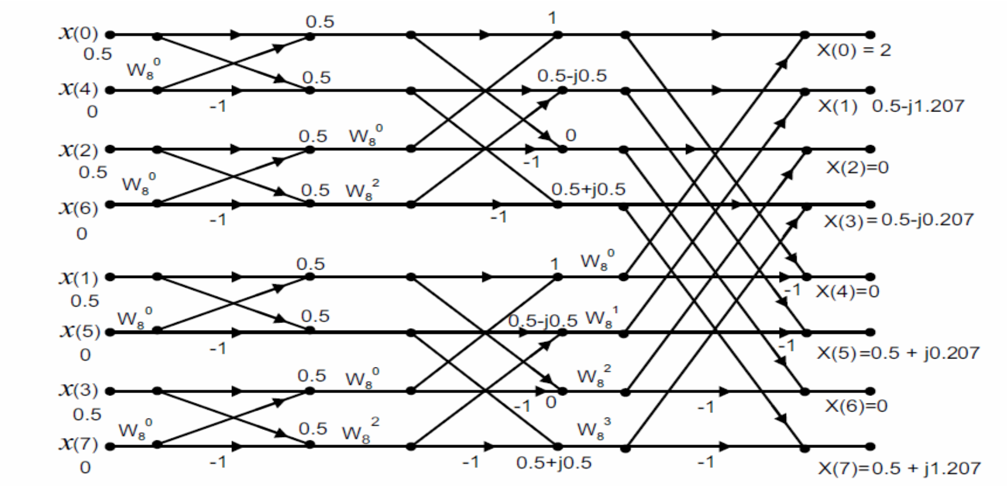
Fig.

$X(k) = \{9, 5.828, 1, 0.172, 1, 0.172, 1, 5.828\}.$

2. Find the DFT of the sequence x(n)=(2,2,2,2,1,1,1,1) using DIT FFT.



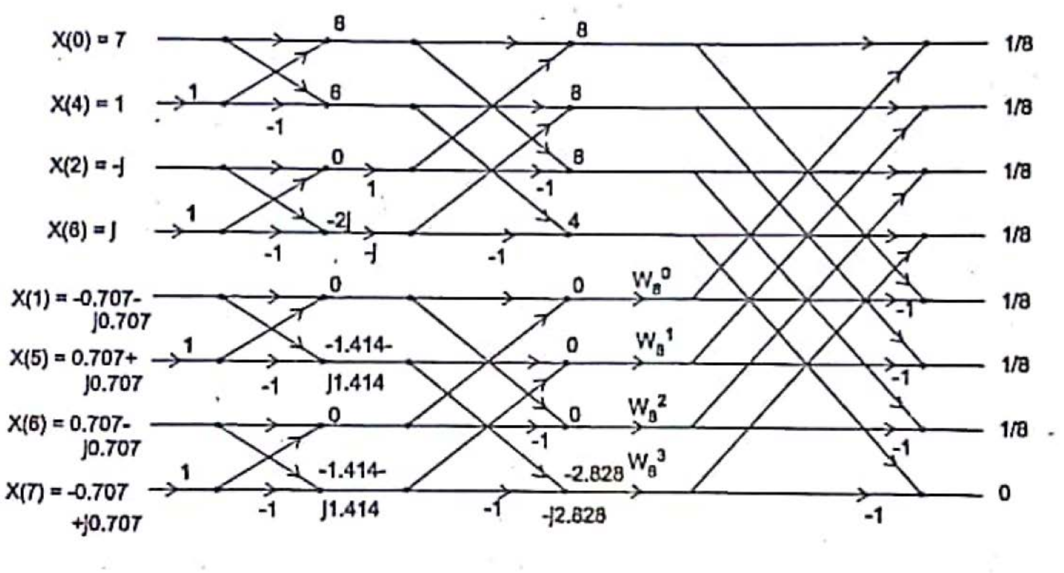
3. Compute the 8-Point DFT of the sequence x(n) = {1/2,1/2,1/2,1/2,0,0,0,0} by using the in-place radix-2 DIT FFT algorithm



$$X(k) = \{2,\ 0.5 - j1.207,\ 0, 0.5 - j0.207,\ 0,\ 0.5 + j0.207,\ 0,\ 0.5 + j1.207\}$$

4. Compute IDFT of the sequence

X(k) = {7, -0.707-j0.707,-j,0.707-j0.707,1,0.707 +j0.707,j, -0.707+j0.707} using DIT and DIF algorithms.



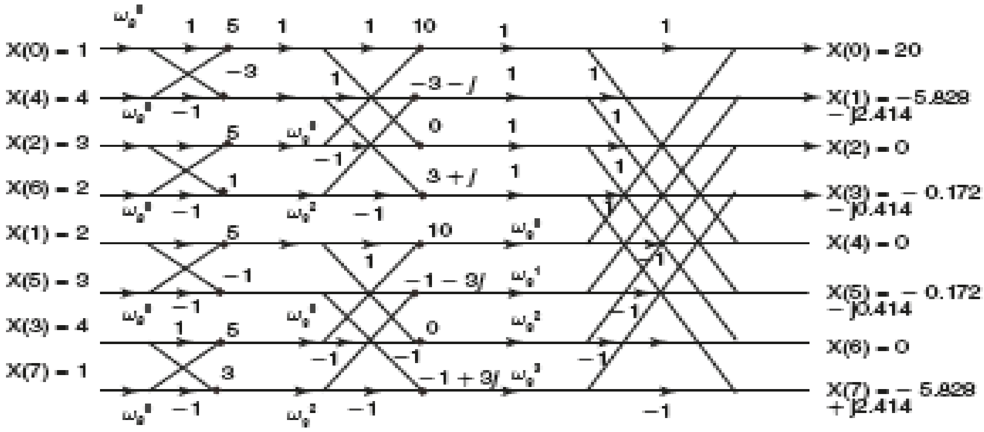
Ans: $x(n) = \{1, 1, 1, 1, 1, 1, 1, 0\}$

5.Determine the 8-point DFT using Decimation in Time FFT .

x(n) = {1,2,3,4,4,3,2,1 }

$\omega_8^0 = 1; \qquad \omega_8^1 = \left(e^{-j2\pi/8}\right)^1 = 0.707 - j0.707$

$\omega_8^2 = -j; \qquad \omega_8^3 = -0.707 - j0.707$

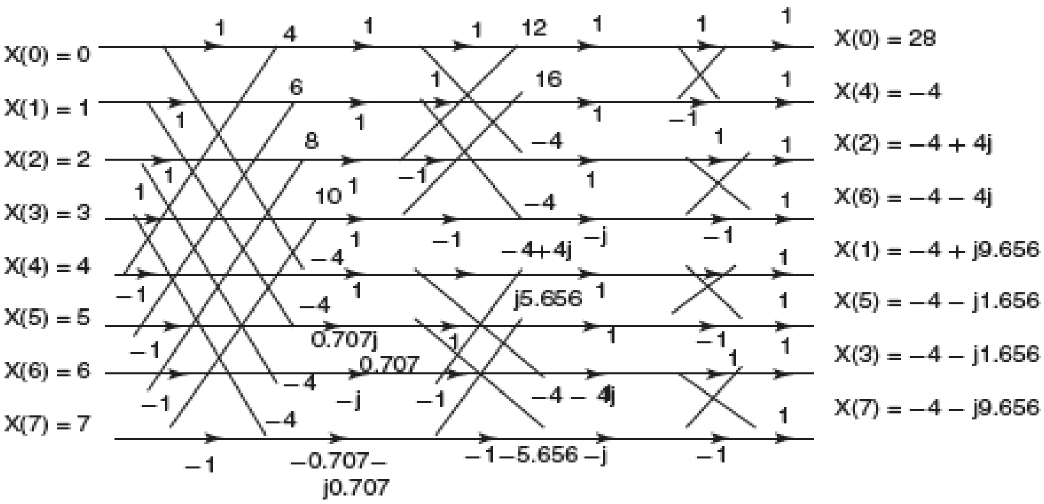


$\therefore \quad X(k) = \{20, -5.828 - j2.414, 0, -0.172 - j0.414, 0,$

$-0.172 + j0.414, 0, -5.828 + j2.414\}$

6. Determine the 8-point DFT using Radix- 2 DIF-FFT algorithm.

x(n) = {0,1,2,3,4,5,6,7}

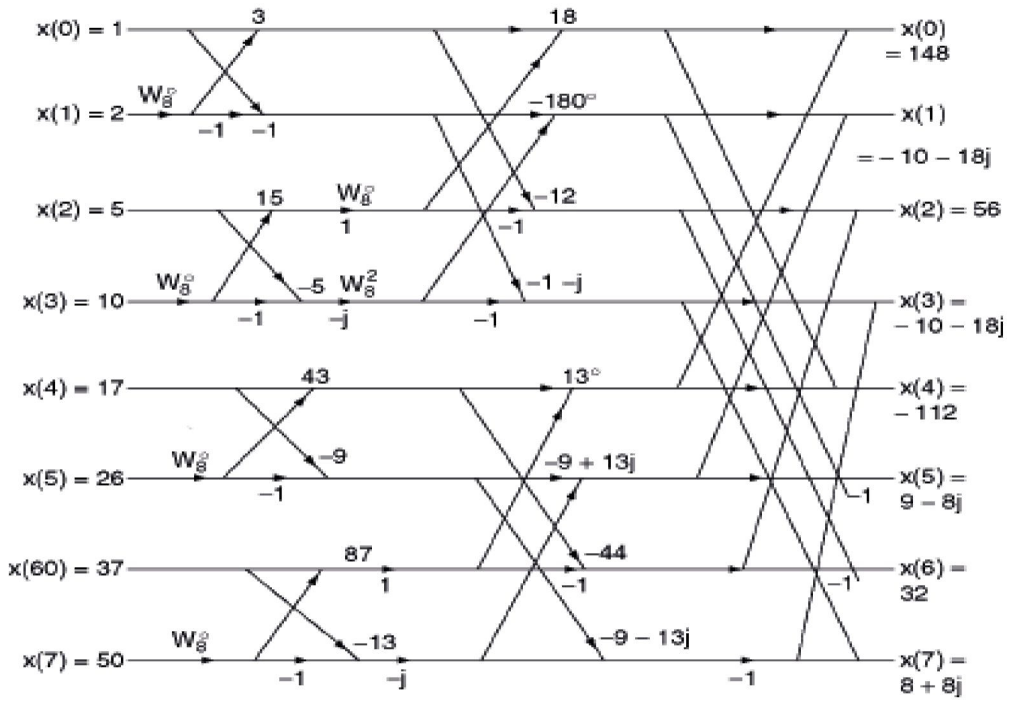


X(k) = {28, −4 + j9.656, −4 + 4j, −4 + j1.656, −4, −4 − j1.656, −4 − 4j, −4 − j9.656}

7.Compute the FFT of the of the sequence *x(n)*= n^2+1 for 0≤ n≤ N-1, where N=8 using DIT algorithm.

$$x(n) = n^2 + 1$$

$$x(n) = \{1, 2, 5, 10, 17, 26, 37, 50\}$$



$$X(k) = \{148, -10+18j, 56, -10-18j, -112, 9-8j, 32, 8+8j\}$$