DIGITAL SIGNAL PROCESSING

UNIT-III
Discrete Fourier Transform
Fast Fourier Transform

1. Discrete Fourier Transform (DFT)

- Computation of N-Point DFT of a Sequence
- Computation of N-Point IDFT of a Sequence
- Relation between N-Point DFT & DTFT of a Sequence
- Properties of Phase Factor or Twiddle Factor

2. Properties of DFT

- Linear Property
- Periodic Property
- Time Shifting Property
- Frequency Shifting Property
- Time Reversal property
- Conjugate Property
- Parsevalls Theorem
- Time Convolution Theorem
- Frequency Convolution Theorem

3. Convolution

- Linear Convolution
- Circular Convolution
- Linear Convolution through Circular Convolution
- Response of discrete LTI system through circular convolution
- Circular Convolution through DFT-IDFT
- Linear Convolution through DFT-IDFT

4. Fast Fourier Transform (FFT)

- Direct Computation of N-Point DFT
- Indirect Computation of N-Point DFT (FFT)
- Comparison b/n Direct Computation of N-Point DFT & FFT
- Decimation In Time (DIT) radix–2 FFT Algorithm
- Decimation In Frequency (DIF) radix–2 FFT Algorithm
- Comparison b/n DIT radix–2 FFT and DIF radix–2 FFT
- Inverse FFT

5. Descriptive Questions

6. Quiz Questions

Discrete Fourier Transform (DFT):

Discrete Fourier Transform (DFT) is a mathematical tool, which is used to compute the discrete frequency domain from a finite duration sequence.

(A)Computation of N-Point DFT of a Sequence:

If x(n) is a finite duration time domain sequence with a duration of N samples over the range 0≤n≤N–1, then the N-Point Discrete Fourier Transform (DFT) of a sequence x(n) can be computed from the formula.

$$N\text{-Point } DFT[\, x(n) \,] = X(k) = \sum_{n=0}^{N-1} x(n)\, W_N^{nk}$$

Where,

X(k) : Frequency domain sequence and it is a periodic with a period of N samples.

W_N^{nk} : Phase factor or twiddle factor and it can be defined as $W_N^{nk} = e^{-j2\pi \frac{nk}{N}}$

(B)Computation of N-Point IDFT of a Sequence:

If X(k) is a finite duration frequency domain sequence with a duration of N samples over the range 0≤k≤N–1, then the N-Point Inverse Discrete Fourier Transform (IDFT) of a sequence X(k) can be computed from the formula.

$$N\text{-Point } IDFT[\, X(k) \,] = x(n) = \frac{1}{N} \sum_{k=0}^{N-1} X(k)\, W_N^{-nk}$$

Where,

x(n) : Time domain sequence and it is a periodic with a period of N samples.

W_N^{-nk} : Phase factor or twiddle factor and it can be defined as $W_N^{-nk} = e^{j2\pi \frac{nk}{N}}$

W_N^{nk} and W_N^{-nk} are complex conjugate pairs

(C)Relation between N-Point DFT and DTFT of a Sequence:

From the basic definition of N-Point DFT of a sequence x(n)

$$N-Point\,DFT[\,x(n)\,] = \sum_{n=0}^{N-1} x(n)\, W_N^{nk}$$

$$= \sum_{n=0}^{N-1} x(n)\, e^{-j2\pi\frac{nk}{N}}$$

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

$$= \sum_{n=0}^{N-1} x(n)\, e^{-j\omega n}, \omega = \frac{2\pi k}{N}$$

$$= DTFT[x(n)]$$

$$N-Point\,DFT[\,x(n)\,] = DTFT[x(n)], with, \omega = \frac{2\pi k}{N}$$

(D)Properties of Phase Factor or Twiddle Factor:

Property 1 : $W_N^0 = W_{-N}^0 = 1$

Property 2 : $W_N^N = W_{-N}^N = W_N^{-N} = W_{-N}^{-N} = 1$

Property 3 : $W_N^{nN} = W_1^n = 1, n = 0, \pm1, \pm2, \pm3, ...$

Property 4 : $W_N^a = W_N^{a-nN} = W_N^{a+nN} = 1, n = 0, \pm1, \pm2, \pm3, ...$

Ex : $W_4^9 = W_4^{9-4} = W_4^5 = W_4^{5-4} = W_4^1$

: $W_4^{-9} = W_4^{-9+4} = W_4^{-5} = W_4^{-5+4} = W_4^{-1} = W_4^{-1+4} = W_4^3$

Property 5 : $W_N^{nk} = W_{N/k}^{n} = W_{N/n}^{k}$

Ex : $W_{16}^{12} = W_4^3$

Property 6 : $W_N^{n_1k}\, W_N^{n_2k} = W_N^{(n_1+n_2)k}$

Property 7 : $\left| W_N^{nk} \right| = 1$

Property 8 : W_N^{nk} and W_N^{-nk} are complex conjugate pairs

Useful Formulas:

- $W_2^0 = e^{-j2\pi\frac{0}{2}} = e^0 = 1 = W_4^0 = W_8^0 = W_{16}^0$
- $W_2^1 = e^{-j2\pi\frac{1}{2}} = e^{-j\pi} = Cos\pi - jSin\pi = -1 - j0 = -1$
- $W_2^2 = e^{-j2\pi\frac{2}{2}} = e^{-j2\pi} = Cos2\pi - jSin2\pi = 1 - j0 = 1 = W_4^4 = W_8^8 = W_{16}^{16}$
- $W_2^3 = W_2^{3-2} = W_2^1 = -1$ and $W_2^4 = W_2^{4-2} = W_2^2 = 1$
- $W_2^5 = W_2^{5-2} = W_2^3 = W_2^{3-2} = W_2^1 = -1$
- $W_4^1 = e^{-j2\pi\frac{1}{4}} = e^{-j\pi/2} = Cos(\pi/2) - jSin(\pi/2) = 0 - j1 = -j$
- $W_4^2 = e^{-j2\pi\frac{2}{4}} = e^{-j\pi} = Cos\pi - jSin\pi = -1 - j0 = -1$
- $W_4^3 = e^{-j2\pi\frac{3}{4}} = e^{-j3\pi/2} = Cos(3\pi/2) - jSin(3\pi/2) = 0 - j(-1) = j$
- $W_4^4 = e^{-j2\pi\frac{4}{4}} = e^{-j2\pi} = Cos(2\pi) - jSin(2\pi) = 1 - j0 = 1$
- $W_4^5 = W_4^{5-4} = W_4^1 = -j$ and $W_4^6 = W_4^{6-4} = W_4^2 = -1$
- $W_4^7 = W_4^{7-4} = W_4^3 = j$ and $W_4^8 = W_4^{8-4} = W_4^4 = 1$
- $W_4^9 = W_4^{9-4} = W_4^5 = W_4^{5-4} = W_4^1 = -j$
- $W_8^1 = e^{-j2\pi\frac{1}{8}} = e^{-j\pi/4} = Cos(\pi/4) - jSin(\pi/4) = \frac{1}{\sqrt{2}} - j\frac{1}{\sqrt{2}}$
- $W_8^2 = W_4^1 = -j$ and $W_8^4 = W_2^1 = -1$ and $W_8^6 = W_8^{6-8} = W_8^{-2} = (-j)^* = j$
- $W_8^3 = e^{-j2\pi\frac{3}{8}} = e^{-j3\pi/4} = Cos(3\pi/4) - jSin(3\pi/4) = -\frac{1}{\sqrt{2}} - j\frac{1}{\sqrt{2}}$
- $W_8^5 = W_8^{5-8} = W_8^{-3} = \left(-\frac{1}{\sqrt{2}} - j\frac{1}{\sqrt{2}}\right)^* = -\frac{1}{\sqrt{2}} + j\frac{1}{\sqrt{2}}$
- $W_8^7 = W_8^{7-8} = W_8^{-1} = \left(\frac{1}{\sqrt{2}} - j\frac{1}{\sqrt{2}}\right)^* = \frac{1}{\sqrt{2}} + j\frac{1}{\sqrt{2}}$

Example 1: Compute the 4-point DFT of a sequence x(n)={1,2,3,4}

From the basic definition of N-Point DFT of a sequence x(n)

$$N-\text{Point}\,DFT[\,x(n)\,] = \sum_{n=0}^{N-1} x(n)\,W_N^{nk}$$

Given sequence x(n)={1,2,3,4} and N=4

$$X(k) = \sum_{n=0}^{3} x(n)\,W_4^{nk}$$

$$= x(0)\,W_4^{0k} + x(1)\,W_4^{1k} + x(2)\,W_4^{2k} + x(3)\,W_4^{3k}$$

$$= 1 + 2W_4^{k} + 3W_4^{2k} + 4W_4^{3k}$$

Compute 4 samples of X(k) by substituting k=0,1,2,3

$$k=0 \Rightarrow X(0) = 1 + 2W_4^{0} + 3W_4^{0} + 4W_4^{0} = 1+2+3+4 = 10$$

$$k=1 \Rightarrow X(1) = 1 + 2W_4^{1} + 3W_4^{2} + 4W_4^{3} = 1 + 2(-j) + 3(-1) + 4(j) = -2 + 2j$$

$$k=2 \Rightarrow X(2) = 1 + 2W_4^{2} + 3W_4^{4} + 4W_4^{6} = 1 + 2(-1) + 3(1) + 4(-1) = -2$$

$$k=3 \Rightarrow X(3) = 1 + 2W_4^{3} + 3W_4^{6} + 4W_4^{9} = 1 + 2(j) + 3(-1) + 4(-j) = -2 - 2j$$

$$4-\text{Point}\,DFT[\,x(n)\,] = X(k) = \{10, -2+2j, -2, -2-2j\}$$

Magnitude Spectrum of X(k)

$$|X(k)| = \{|10|, |-2+2j|, |-2|, |-2-2j|\}$$

$$= \{10, \sqrt{(-2)^2 + 2^2}, 2, \sqrt{(-2)^2 + (-2)^2}\}$$

$$= \{10, \sqrt{8}, 2, \sqrt{8}\}$$

$$= \{10, 2\sqrt{2}, 2, 2\sqrt{2}\}$$

Magnitude Spectrum: $|X(k)| = \{10, 2\sqrt{2}, 2, 2\sqrt{2}\}$

Phase Spectrum of X(k)

$$\angle X(k) = \left\{Tan^{-1}\left(\frac{0}{10}\right), Tan^{-1}\left(\frac{2}{-2}\right), Tan^{-1}\left(\frac{0}{-2}\right), Tan^{-1}\left(\frac{-2}{-2}\right)\right\}$$

$$= \{0^0, 180^0 - 45^0, 180^0, 180^0 + 45^0\}$$

$$= \{0^0, 135^0, 180^0, 225^0\}$$

Phase Spectrum : $\angle X(k) = \{0^0, 135^0, 180^0, 225^0\}$

Example 2: Compute the 4-point IDFT of a sequence X(k)={10,-2+2j,-2,-2-2j}

From the basic definition of N-Point IDFT of a sequence X(k)

$$N-Point\ IDFT[\ X(k)\ ] = \frac{1}{N}\sum_{k=0}^{N-1} X(k)\, W_N^{-nk}$$

Given sequence X(k)={10,-2+2j,-2,-2-2j} and N=4

$$x(n) = \frac{1}{4}\sum_{k=0}^{3} X(k)\, W_4^{-nk}$$

$$= \frac{1}{4}\left(X(0)\, W_4^{-n0} + X(1)\, W_4^{-n1} + X(2)\, W_4^{-n2} + X(3)\, W_4^{-n3} \right)$$

$$= \frac{1}{4}\left(10 + (-2+2j)W_4^{-n} - 2W_4^{-2n} + (-2-2j)W_4^{-3n} \right)$$

Compute 4 samples of x(n) by substituting n=0,1,2,3

$$n=0 \Rightarrow x(0) = \frac{1}{4}\left(10 + (-2+2j)W_4^{-0} - 2W_4^{-0} + (-2-2j)W_4^{-0} \right)$$

$$\Rightarrow x(0) = \frac{1}{4}\left(10 + (-2+2j) - 2 + (-2-2j)\right) = \frac{1}{4}(4) = 1$$

$$n=1 \Rightarrow x(1) = \frac{1}{4}\left(10 + (-2+2j)W_4^{-1} - 2W_4^{-2} + (-2-2j)W_4^{-3} \right)$$

$$\Rightarrow x(1) = \frac{1}{4}\left(10 + (-2+2j)(j) - 2(-1) + (-2-2j)(-j)\right)$$

$$\Rightarrow x(1) = \frac{1}{4}\left(10 + -2j - 2 + 2 + 2j - 2\right) = \frac{1}{4}(8) = 2$$

$$n=2 \Rightarrow x(2) = \frac{1}{4}\left(10 + (-2+2j)W_4^{-2} - 2W_4^{-4} + (-2-2j)W_4^{-6} \right)$$

$$\Rightarrow x(2) = \frac{1}{4}\left(10 + (-2+2j)(-1) - 2(1) + (-2-2j)(-1)\right)$$

$$\Rightarrow x(2) = \frac{1}{4}\left(10 + 2 - 2j - 2 + 2 + 2j\right) = \frac{1}{4}(12) = 3$$

$$n=3 \Rightarrow x(3) = \frac{1}{4}\left(10 + (-2+2j)W_4^{-3} - 2W_4^{-6} + (-2-2j)W_4^{-9} \right)$$

$$\Rightarrow x(3) = \frac{1}{4}\left(10 + (-2+2j)(-j) - 2(-1) + (-2-2j)(j)\right)$$

$$\Rightarrow x(3) = \frac{1}{4}\left(10 + 2j + 2 + 2 - 2j + 2\right) = \frac{1}{4}(16) = 4$$

4-Point IDFT [X(k)] = {1,2,3,4}

Properties of DFT:

(A)Linear Property:

If $x_1(n)$, $x_2(n)$ are two finite duration sequences with the equal duration of N samples and

N-Point DFT[$x_1(n)$] = $X_1(k)$,

N-Point DFT[$x_2(n)$] = $X_2(k)$, then

N-Point DFT[a $x_1(n)$ + b $x_2(n)$] = a $X_1(k)$ + b $X_2(k)$ is called linear property of DFT.

Where a and b are arbitrary constants.

Proof:

From the basic definition of N-Point DFT of a sequence

$$\text{N-Point DFT}[\,x(n)\,] = \sum_{n=0}^{N-1} x(n)\, W_N^{nk}$$

Replace x(n) by 'a $x_1(n)$ + b $x_2(n)$'

$$\begin{aligned}
\text{N-Point DFT}[\,a\,x_1(n) + b\,x_2(n)\,] &= \sum_{n=0}^{N-1} [\,a\,x_1(n) + b\,x_2(n)\,]\, W_N^{nk} \\
&= \sum_{n=0}^{N-1} \left[a\,x_1(n)\, W_N^{nk} + b\,x_2(n)\, W_N^{nk} \right] \\
&= \sum_{n=0}^{N-1} \left[a\,x_1(n)\, W_N^{nk} \right] + \sum_{n=0}^{N-1} \left[x_2(n)\, W_N^{nk} \right] \\
&= a \sum_{n=0}^{N-1} x_1(n)\, W_N^{nk} + b \sum_{n=0}^{N-1} x_2(n)\, W_N^{nk} \\
&= a\,\text{N-Point DFT}[\,x_1(n)\,] + b\,\text{N-Point DFT}[\,x_2(n)\,] \\
&= a\,X_1(k) + b\,X_2(k)
\end{aligned}$$

(B)Periodic Property:

(a) If x(n) is a finite duration sequence with a duration of N samples and

N-Point DFT[x(n)] = X(k), then the sequence X(k) is periodic with a period of N samples,

i.e X(N + k) = X(k).

Proof:

From the basic definition of N-Point DFT of a sequence

$$\text{N-Point DFT}[\,x(n)\,] = \sum_{n=0}^{N-1} x(n)\, W_N^{nk}$$

$$X(k) = \sum_{n=0}^{N-1} x(n)\, W_N^{nk}$$

Replace k with N + k

$$X(N+k) = \sum_{n=0}^{N-1} x(n)\, W_N^{n(N+k)}$$

$$X(N+k)=\sum_{n=0}^{N-1} x(n)\,W_N^{(nN+nk)}$$
$$=\sum_{n=0}^{N-1} x(n)\,W_N^{nN}\,W_N^{nk}$$
$$=\sum_{n=0}^{N-1} x(n)\;(1)\,W_N^{nk}$$
$$=\sum_{n=0}^{N-1} x(n)\,W_N^{nk}$$
$$=X(k)$$

(b) If X(k) is a finite duration sequence with a duration of N samples and

N-Point IDFT[X(k)] = x(n), then the sequence x(n) is periodic with a period of N samples,

i.e x(N +n) = x(n).

Proof:

From the basic definition of N-Point IDFT of a sequence

$$N\text{-Point IDFT}[\,X(k)\,]=\frac{1}{N}\sum_{k=0}^{N-1} X(k)\,W_N^{-nk}$$
$$x(n)=\frac{1}{N}\sum_{k=0}^{N-1} X(k)\,W_N^{-nk}$$

Replace n with N + n

$$x(N+n)=\frac{1}{N}\sum_{k=0}^{N-1} X(k)\,W_N^{-(N+n)k}$$
$$=\frac{1}{N}\sum_{k=0}^{N-1} X(k)\,W_N^{-(Nk+nk)}$$
$$=\frac{1}{N}\sum_{k=0}^{N-1} X(k)\,W_N^{-Nk}\,W_N^{-nk}$$
$$=\frac{1}{N}\sum_{k=0}^{N-1} X(k)\;(1)\,W_N^{-nk}$$
$$=\frac{1}{N}\sum_{k=0}^{N-1} X(k)\,W_N^{-nk}$$
$$=N-\text{Point IDFT}[X(k)]$$
$$=x(n)$$

(C)Time Shifting Property:

If x(n) is finite duration sequence with a duration of N samples and

N-Point DFT[x(n)] = X(k), then

DFT[$x(n-n_0)$] = $W_N^{n_0k}X(k)$ is called time shifting or circular time shift property of DFT.

Where, n_0 is constant.

Proof:

From the basic definition of N-Point DFT of a sequence

$$DFT[\ x(n)\ ] = \sum_{n=0}^{N-1} x(n)\ W_N^{nk}$$

Replace x(n) with $x(n - n_0)$

$$DFT[\ x(n-n_0)\ ] = \sum_{n=0}^{N-1} x(n-n_0)\ W_N^{nk}, \qquad \text{Let } n - n_0 = m$$

$$= \sum_{m=-n_0}^{N-1-n_0} x(m)\ W_N^{(n_0+m)k}$$

$$= \sum_{m=0}^{N-1} x(m)\ W_N^{n_0k+mk}$$

$$= \sum_{m=0}^{N-1} x(m)\ W_N^{n_0k}\ W_N^{mk}$$

$$= W_N^{n_0k} \sum_{m=0}^{N-1} x(m)\ W_N^{mk}$$

$$= W_N^{n_0k}\ X(k)$$

(D)Frequency Shifting Property:

If x(n) is a finite duration sequence with a duration of N samples and N-Point DFT[x(n)] = X(k),

then N-Point DFT [$W_N^{-nk_0}$ x(n)] = $X(k - k_0)$ is called frequency shifting property.

Proof:

From the basic definition of N-Point DFT of a sequence

$$N\text{-Point } DFT[\ x(n)\ ] = \sum_{n=0}^{N-1} x(n)\ W_N^{nk}$$

Replace x(n) by $W_N^{-nk_0}$ x(n)

$$N\text{-Point } DFT[\ W_N^{-nk_0}\ x(n)\ ] = \sum_{n=0}^{N-1} W_N^{-nk_0}\ x(n)\ W_N^{nk}$$

$$= \sum_{n=0}^{N-1} x(n)\ \ W_N^{nk} W_N^{-nk_0}$$

$$= \sum_{n=0}^{N-1} x(n)\ \ W_N^{n(k-k_0)}$$

$$= DFT[\ x(n)\ ] \text{ at } k = k - k_0.$$

$$= X(k - k_0)$$

(E)Time Reversal Property:

If x(n) is a finite duration sequence with a duration of N samples and N-Point DFT[x(n)] = X(k), then N-Point DFT[x(N – n)] = X(N – k) is called time reversal property of DFT

Proof:

From the basic definition of N-Point DFT of a sequence

$$\text{N-Point DFT}[\ x(n)\ ] = \sum_{n=0}^{N-1} x(n)\ W_N^{nk}$$

Replace x(n) by x(N – n)

$$\text{N-Point DFT}[\ x(N-n)\ ] = \sum_{n=0}^{N-1} x(N-n)\ W_N^{nk}\ \text{, Let } N-n=m$$

$$= \sum_{m=N}^{1} x(m)\ W_N^{(N-m)k}$$

$$= \sum_{m=N}^{1} x(m)\ W_N^{Nk}\ W_N^{-mk}$$

$$= \sum_{m=N}^{1} x(m)\ (1)\ W_N^{-mk}$$

$$= \sum_{m=N}^{1} x(m)\ W_N^{-mk}\ (1)$$

$$= \sum_{m=0}^{N-1} x(m)\ W_N^{-mk}\ W_N^{mN}$$

$$= \sum_{m=0}^{N-1} x(m)\ W_N^{m(N-k)}$$

$$= \text{DFT}[\ x(n)\ ] \text{ at } k = N-k.$$

$$= X(N-k)$$

(F)Conjugate Property:

If x(n) is a finite duration sequence with a duration of N samples and N-Point DFT[x(n)] = X(k), then

(a)N-Point DFT[x*(n)] = X*(N – k)

(b)N-Point DFT[x*(N-n)] = X*(N + k) is called conjugate property of DFT.

Proof:

(a) From the basic definition of N-Point DFT

$$\text{N-Point DFT}[\ x(n)\ ] = \sum_{n=0}^{N-1} x(n)\ W_N^{nk}$$

Replace x(n) by x*(n)

$$\text{N-Point DFT}[\,x^*(n)\,] = \sum_{n=0}^{N-1} x^*(n)\, W_N^{nk}$$

$$= \sum_{n=0}^{N-1} x^*(n) \left[W_N^{-nk} \right]^*$$

$$= \sum_{n=0}^{N-1} \left[x(n)\, W_N^{-nk} \right]^*$$

$$= \left[\sum_{n=0}^{N-1} x(n)\,(1)\, W_N^{-nk} \right]^*$$

$$= \left[\sum_{n=0}^{N-1} x(n)\,(W_N^{nN})\, W_N^{-nk} \right]^*$$

$$= \left[\sum_{n=0}^{N-1} x(n)\; W_N^{n(N-k)} \right]^*$$

$$= [\,\text{N-Point DFT}\,[\,x(n)\,]\text{ at } k = N-k\,]^*$$

$$= [\,X(N-k)\,]^*$$

$$= X^*(N-k)$$

(b) From the basic definition of N-Point DFT

$$\text{N-Point DFT}[\,x(n)\,] = \sum_{n=0}^{N-1} x(n)\, W_N^{nk}$$

Replace x(n) by x*(N-n)

$$\text{N-Point DFT}[\,x^*(N-n)\,] = \sum_{n=0}^{N-1} x^*(N-n)\, W_N^{nk} \quad ,\text{Let } N-n = m$$

$$= \sum_{m=N}^{1} x^*(m)\, W_N^{(N-m)k}$$

$$= \sum_{m=1}^{N} x^*(m)\, W_N^{Nk}\;\; W_N^{-mk}$$

$$= \sum_{m=0}^{N-1} x^*(m)\,(1)\;\; W_N^{-mk}$$

$$= \sum_{m=N}^{1} x^*(m) \left[W_N^{mk} \right]^*$$

$$= \sum_{m=0}^{N-1} \left[x(m)\; W_N^{mk} \right]^*$$

$$= \left[\sum_{n=0}^{N-1} x(n)\, W_N^{nk} \right]^*$$

$$= \left[\sum_{n=0}^{N-1} x(n)\,(1)\, W_N^{nk} \right]^*$$

$$= \left[\sum_{n=0}^{N-1} x(n)\,(W_N^{nN})\, W_N^{nk} \right]^*$$

$$\text{N-Point DFT}[\, x^*(N-n)\,] = \left[\sum_{n=0}^{N-1} x(n)\; W_N^{n(N+k)}\right]^*$$
$$= [\,\text{N-Point DFT } [x(n)] \text{ at } k = N+k\,]^*$$
$$= [\, X(N+k)\,]^*$$
$$= X^*(N+k)$$

(G)Parsevalls Theorem:

If x(n) is a finite duration sequence with a duration of N samples and N-Point DFT[x(n)] = X(k), then the total Energy under the signal x(n) can be computed from the following relation is called Parsevalls theorem.

$$\text{Total Energy (E)} = \sum_{n=0}^{N-1} |x(n)|^2 = \frac{1}{N}\sum_{k=0}^{N-1} |X(k)|^2$$

Proof:

We know that,

$$\text{Total Energy (E)} = \sum_{n=0}^{N-1} |x(n)|^2$$
$$= \sum_{n=0}^{N-1} x(n)\,[x(n)]^*$$
$$= \sum_{n=0}^{N-1} x(n)\,[\text{IDFT}[\, X(k)\,]\,]^*$$
$$= \sum_{n=0}^{N-1} x(n)\left[\frac{1}{N}\sum_{k=0}^{N-1} X(k)\, W_N^{-nk}\right]^*$$
$$= \sum_{n=0}^{N-1} x(n)\frac{1}{N}\sum_{k=0}^{N-1} X^*(k)\, W_N^{nk}$$

Change the order of two sums

$$= \frac{1}{N}\sum_{k=0}^{N-1} X^*(k)\sum_{n=0}^{N-1} x(n) W_N^{nk}$$
$$= \frac{1}{N}\sum_{k=0}^{N-1} X^*(k)\,\text{DFT}[\, x(n)\,]$$
$$= \frac{1}{N}\sum_{k=0}^{N-1} X^*(k)\, X(k)$$
$$= \frac{1}{N}\sum_{k=0}^{N-1} |X(k)|^2$$

(H)Time Convolution Theorem:

If $x_1(n)$, $x_2(n)$ are two finite duration sequences with the equal duration of N samples and

N-Point DFT[$x_1(n)$] = $X_1(k)$,

N-Point DFT[$x_2(n)$] = $X_2(k)$, then

N-Point DFT[$x_1(n) \otimes x_2(n)$] = $X_1(k)\ X_2(k)$ is called time convolution theorem.

i.e., Convolution in time domain leads to multiplication in frequency domain.

Proof:

From the basic definition of N-Point DFT

$$\text{N-Point DFT}[\,x(n)\,] = \sum_{n=0}^{N-1} x(n)\, W_N^{nk}$$

Replace x(n) by $x_1(n) \otimes x_2(n)$

$$\text{N-Point DFT}[\,x_1(n) \otimes x_2(n)] = \sum_{n=0}^{N-1} [\,x_1(n) \otimes x_2(n)]\ \ W_N^{nk}$$

$$= \sum_{n=0}^{N-1} \sum_{m=0}^{N-1} [\,x_1(m)\, x_2(n-m)]\ \ W_N^{nk}$$

Change the order of summation

$$\text{N-Point DFT}[\,x_1(n) \otimes x_2(n)] = \sum_{m=0}^{N-1} \left[x_1(m)\ \left(\sum_{n=0}^{N-1} x_2(n-m)]\ \ W_N^{nk} \right) \right]$$

$$= \sum_{m=0}^{N-1} [x_1(m)\ \ (DFT(x_2(n-m))]$$

$$= \sum_{m=0}^{N-1} \left[x_1(m)\ \left(W_N^{mk}\ \ X_2(K) \right) \right]$$

$$= X_2(K) \sum_{m=0}^{N-1} \left[x_1(m)\, W_N^{mk} \right]$$

$$= X_2(K)\ \ X_1(K)$$

$$= X_1(K) X_2(K)$$

N-Point DFT[$x_1(n) \otimes x_2(n)$] = $X_1(k)\ X_2(k)$

(I)Frequency Convolution Theorem:

If $x_1(n)$, $x_2(n)$ are two finite duration sequences with the equal duration of N samples and

N-Point DFT[$x_1(n)$] = $X_1(k)$,

N-Point DFT[$x_2(n)$] = $X_2(k)$, then

N-Point DFT[$x_1(n)\, x_2(n)$] = [$X_1(k) \otimes X_2(k)$] / N is called frequency convolution theorem.

i.e., Convolution in frequency domain leads to multiplication in time domain.

Proof:

From the basic definition of N-Point IDFT of a sequence

$$\text{N-Point IDFT}[\,X(k)\,] = \frac{1}{N}\sum_{k=0}^{N-1} X(k)\, W_N^{-nk}$$

Replace X(k) by $X_1(k) \otimes X_2(k)$

$$\text{N-Point IDFT}[\,X_1(k)\otimes X_2(k)\,] = \frac{1}{N}\sum_{k=0}^{N-1} [\,X_1(k)\otimes X_2(k)\,]\, W_N^{-nk}$$

$$= \frac{1}{N}\sum_{k=0}^{N-1}\left(\sum_{m=0}^{N-1} [\,X_1(m) X_2(k-m)\,]\right) W_N^{-nk}$$

Change the order of two sums

$$\text{N-Point IDFT}[\,X_1(k)\otimes X_2(k)\,] = \frac{1}{N}\sum_{m=0}^{N-1}\left(X_1(m) \sum_{k=0}^{N-1} X_2(k-m)\, W_N^{-nk}\right)$$

$$= \frac{1}{N}\sum_{m=0}^{N-1}\left(X_1(m)\, N\left(\frac{1}{N}\sum_{k=0}^{N-1} X_2(k-m)\, W_N^{-nk}\right)\right)$$

$$= \frac{1}{N}\sum_{m=0}^{N-1}\left(X_1(m)\, N\left(\text{IDFT}(X_2(k-m))\right)\right)$$

$$= \frac{1}{N}\sum_{m=0}^{N-1}\left(X_1(m)\, N\left(W_N^{-mn}\, x_2(n)\right)\right)$$

$$= N\, x_2(n)\ \frac{1}{N}\sum_{m=0}^{N-1}\left(X_1(m)\, W_N^{-mn}\right)$$

$$= N\, x_1(n)\ x_2(n)$$

$$\text{N-Point DFT}\,[\,x_1(n)\ x_2(n)\,] = \frac{X_1(k)\otimes X_2(k)}{N}$$

Convolution:

Convolution is an operation, which is used in almost all signal processing applications to analyze signals and systems in both the time and frequency domain. Convolution is a special operation, which includes four different operations, namely

- Folding,
- Shifting,
- Multiplication and
- Summation in the case of discrete time signals or
  Integration in the case of continues time signals.

Convolution in continuous time domain can be defined as

$$x_1(t) * x_2(t) = \int_{-\infty}^{\infty} x_1(\tau)\, x_2(t-\tau)\, d\tau$$

Where, $x_1(t)$ and $x_2(t)$ are two continuous time signals

Convolution in discrete time domain can be defined as

$$x_1(n) * x_2(n) = \sum_{m=-\infty}^{\infty} x_1(m)\, x_2(n-m)$$

Where, $x_1(n)$ and $x_2(n)$ are two discrete time signals

There are two types of convolution procedures

- Linear Convolution
- Circular Convolution

(A)Linear Convolution:

If $x_1(n)$ and $x_2(n)$ are two finite duration sequences with the duration of N_1 samples $[0 \le n \le N_1 - 1]$ and N_2 samples $[0 \le n \le N_2 - 1]$, then the linear convoluted sequence can be defined as

$$x(n) = x_1(n) * x_2(n) = \sum_{m=0}^{N-1} x_1(m)\, x_2(n-m)$$

Where

N_1 : Duration of $x_1(n)$, $0 \le n \le N_1 - 1$

N_2 : Duration of $x_2(n)$, $0 \le n \le N_2 - 1$

N : Duration of linear convoluted sequence $x(n)$, $0 \le n \le N - 1$ ($N= N_1 + N_2 - 1$)

- Duration of $x_1(n)$ or $x_2(n)$ and linear convoluted sequence $x(n)$ are different.
- Linear convolution is also known as aperiodic convolution.
- DFT does not support linear convolution.
- Response of the LSI system can be computed directly from the linear convolution.

Linear convoluted sequence can be computed from

- Graphical method
- Tabular method
- Matrix method

Example:

Compute the linear convoluted sequence $x(n) = x_1(n) * x_2(n)$ using (a)Graphical method (b)Tabular method (c)Matrix method. Given $x_1(n) = \{1,2,3,4\}$ and $x_2(n) =\{5,6,7,8,9\}$

(a)Graphical method

Given sequences

$x_1(n) = \{1,2,3,4\}$ with duration $N_1=4$ Samples and

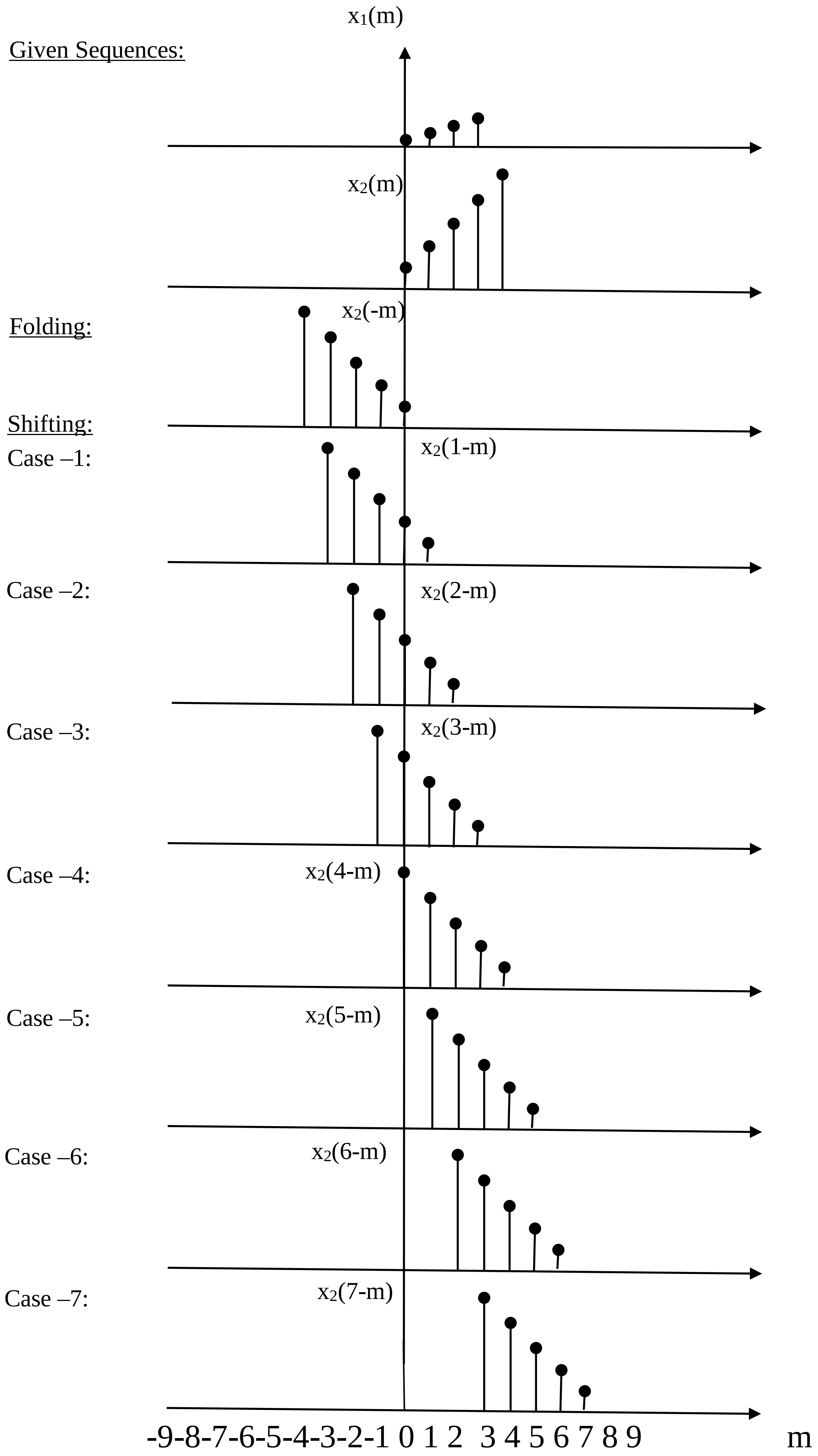
$x_2(n) =\{5,6,7,8,9\}$ with duration $N_2=5$ Samples

Duration of the linear convoluted sequence $x(n)$, $N= N_1 + N_2 - 1 =4+5-1=8$

i.e Compute 8 samples of $x(n)$ through the following procedure

- Draw the graphical representation of given sequences $x_1(m)$ and $x_2(m)$**.**
- Take the folding form of $x_2(m)$ to get $x_2(-m)$.
- Shift the folding sequence $x_2(-m)$ and draw $x_2(1-m)$ to $x_2(7-m)$
- Finally apply linear convolution formula to get the linear convoluted sequence.

$$x(n) = x_1(n) * x_2(n) = \sum_{m=0}^{N-1} x_1(m)\, x_2(n-m)$$


Given Sequences:
x1(m)
x2(m)
Folding:
x2(-m)
Shifting:
Case –1:
x2(1-m)
Case –2:
x2(2-m)
Case –3:
x2(3-m)
Case –4:
x2(4-m)
Case –5:
x2(5-m)
Case –6:
x2(6-m)
Case –7:
x2(7-m)
-9-8-7-6-5-4-3-2-1 0 1 2 3 4 5 6 7 8 9
m

Compute 8 samples of x(n) by substituting n=0 to 7 in $x(n) = x_1(n) * x_2(n) = \sum_{m=0}^{N-1} x_1(m)\, x_2(n-m)$

$$n = 0 \Rightarrow x(0) = \sum_{m=0}^{7} x_1(m)\, x_2(-m) = 1\times 5 = 5$$

$$n = 1 \Rightarrow x(1) = \sum_{m=0}^{7} x_1(m)\, x_2(1-m) = 1\times 6 + 2\times 5 = 6 + 10 = 16$$

$$n = 2 \Rightarrow x(2) = \sum_{m=0}^{7} x_1(m)\, x_2(2-m) = 1\times 7 + 2\times 6 + 3\times 5 = 7 + 12 + 15 = 34$$

$$n = 3 \Rightarrow x(3) = \sum_{m=0}^{7} x_1(m)\, x_2(3-m) = 1\times 8 + 2\times 7 + 3\times 6 + 4\times 5 = 8 + 14 + 18 + 20 = 60$$

$$n = 4 \Rightarrow x(4) = \sum_{m=0}^{7} x_1(m)\, x_2(4-m) = 1\times 9 + 2\times 8 + 3\times 7 + 4\times 6 = 9 + 16 + 21 + 24 = 70$$

$$n = 5 \Rightarrow x(5) = \sum_{m=0}^{7} x_1(m)\, x_2(5-m) = 2\times 9 + 3\times 8 + 4\times 7 = 18 + 24 + 28 = 70$$

$$n = 6 \Rightarrow x(6) = \sum_{m=0}^{7} x_1(m)\, x_2(6-m) = 3\times 9 + 4\times 8 = 27 + 32 = 59$$

$$n = 7 \Rightarrow x(7) = \sum_{m=0}^{7} x_1(m)\, x_2(7-m) = 4\times 9 = 36$$

Linear Convoluted Sequence $x(n) = x_1(n) * x_2(n)$ = {5, 16, 34, 60, 70, 70, 59, 36}

(b)Tabular method:

Given sequences

$x_1(n)$ = {1,2,3,4} with duration N_1=4 Samples and

$x_2(n)$ ={5,6,7,8,9} with duration N_2=5 Samples

Duration of the linear convoluted sequence x(n), N= $N_1 + N_2 - 1$ =4+5-1=8

i.e Compute 8 samples of x(n) through the following procedure

- Draw a table and represent the samples of $x_1(m)$ and $x_2(m)$ in a table**.**
- Take the folding form of $x_2(m)$ and represent the samples of $x_2(-m)$ in a table
- Shift the folding sequence $x_2(-m)$ and represent the samples of $x_2(1-m)$ to $x_2(7-m)$ in a table
- Apply multiplication operation to get the samples of $x_1(m)\, x_2(-m)$ to $x_1(m)\, x_2(7-m)$.
- Finally apply summation operation to get the linear convoluted sequence.

$$x(n) = x_1(n) * x_2(n) \;=\; \sum_{m=0}^{N-1} x_1(m)\, x_2(n-m)$$

	m	**-4**	**-3**	**-2**	**-1**	**0**	**1**	**2**	**3**	**4**	
Given Sequences	**$x_1(m)$**					1	2	3	4		
	$x_2(m)$					5	6	7	8	9	
Folding	**$x_2(-m)$**	9	8	7	6	5	0	0	0		
Shifting	**$x_2(1-m)$**		9	8	7	6	5	0	0		
	$x_2(2-m)$			9	8	7	6	5	0		
	$x_2(3-m)$				9	8	7	6	5		
	$x_2(4-m)$					9	8	7	6		
	$x_2(5-m)$					0	9	8	7		
	$x_2(6-m)$					0	0	9	8		
	$x_2(7-m)$					0	0	0	9		**Sum**
Multiplication	**$x_1(m)\, x_2(-m)$**					5	0	0	0	=	**5**
	$x_1(m)\, x_2(1-m)$					6	10	0	0	=	**16**
	$x_1(m)\, x_2(2-m)$					7	12	15	0	=	**34**
	$x_1(m)\, x_2(3-m)$					8	14	18	20	=	**60**
	$x_1(m)\, x_2(4-m)$					9	16	21	24	=	**70**
	$x_1(m)\, x_2(5-m)$					0	18	24	28	=	**70**
	$x_1(m)\, x_2(6-m)$					0	0	27	32	=	**59**
	$x_1(m)\, x_2(7-m)$					0	0	0	36	=	**36**

Linear Convoluted Sequence $x(n) = x_1(n) * x_2(n)$ = {5, 16, 34, 60, 70, 70, 59, 36}

(c)Matrix method:

Given sequences

$x_1(n)$ = {1,2,3,4} with duration N_1=4 Samples and

$x_2(n)$ ={5,6,7,8,9} with duration N_2=5 Samples

Duration of the linear convoluted sequence x(n), N= $N_1 + N_2 – 1$ =4+5-1=8

i.e Compute 8 samples of x(n) through the following procedure

- Pad $x_1(m)$ with '$N_2 – 1$=4' number of zeros to get a length of 8 samples.

  $x_1(n)$ = {1,2,3,4,0,0,0,0}

- Pad $x_2(m)$ with '$N_1 – 1$=3' number of zeros to get a length of 8 samples.

  $x_2(n)$ ={5,6,7,8,9,0,0,0}

- Now represent $x_1(m)$ and $x_2(n-m)$ in the form of matrices and finally compute the linear convoluted sequence.

$$x(n) = x_1(n) * x_2(n) = \sum_{m=0}^{N-1} x_1(m)\, x_2(n-m)$$

$$= \begin{bmatrix}1 & 2 & 3 & 4 & 0 & 0 & 0 & 0\end{bmatrix}\begin{bmatrix}5 & 6 & 7 & 8 & 9 & 0 & 0 & 0\\0 & 5 & 6 & 7 & 8 & 9 & 0 & 0\\0 & 0 & 5 & 6 & 7 & 8 & 9 & 0\\0 & 0 & 0 & 5 & 6 & 7 & 8 & 9\\9 & 0 & 0 & 0 & 5 & 6 & 7 & 8\\8 & 9 & 0 & 0 & 0 & 5 & 6 & 7\\7 & 8 & 9 & 0 & 0 & 0 & 5 & 6\\6 & 7 & 8 & 9 & 0 & 0 & 0 & 5\end{bmatrix}$$

$$= [1\times5 \quad 1\times6+2\times5 \quad 1\times7+2\times6+3\times5 \quad 1\times8+2\times7+3\times6+4\times5$$
$$1\times9+2\times8+3\times7+4\times6 \quad 2\times9+3\times8+4\times7 \quad 3\times9+4\times8 \quad 4\times9]$$
$$= [5 \quad 6+10 \quad 7+12+15 \quad 8+14+18+20$$
$$9+16+21+24 \quad 18+24+28 \quad 27+32 \quad 36]$$
$$= [5 \quad 16 \quad 34 \quad 60 \quad 70 \quad 70 \quad 59 \quad 36]$$

Linear Convoluted Sequence x(n) = $x_1(n) * x_2(n)$ = {5, 16, 34, 60, 70, 70, 59, 36}

(B)Circular Convolution:

If $x_1(n)$ and $x_2(n)$ are two finite duration sequences with the equal duration of N samples [$0 \le n \le N-1$], then the circular convoluted sequence can be defined as

$$x(n) = x_1(n) * x_2(n) = \sum_{m=0}^{N-1} x_1(m)\, x_2(n-m)$$

Where, N is the duration of $x_1(n)$ or $x_2(n)$ or circular convoluted sequence x(n), $0 \le n \le N - 1$.

- Duration of $x_1(n)$, $x_2(n)$ and circular convoluted sequence x(n) is same.
- Circular convolution is also known as periodic convolution.
- DFT support circular convolution.
- Response of the LSI system can't be computed directly from the circular convolution.

Circular convoluted sequence can be computed from

- Graphical method
- Tabular method
- Matrix method
- Circular method

Example:

Compute the circular convoluted sequence $x(n) = x_1(n) * x_2(n)$ using (a)Graphical method (b)Tabular method (c)Matrix method (d)Circular method. Given $x_1(n)$={1,2,3,4} and $x_2(n)$=5,6,7,8}

(a)Graphical method

Given sequences

$x_1(n)$ = {1,2,3,4} with duration N=4 Samples and

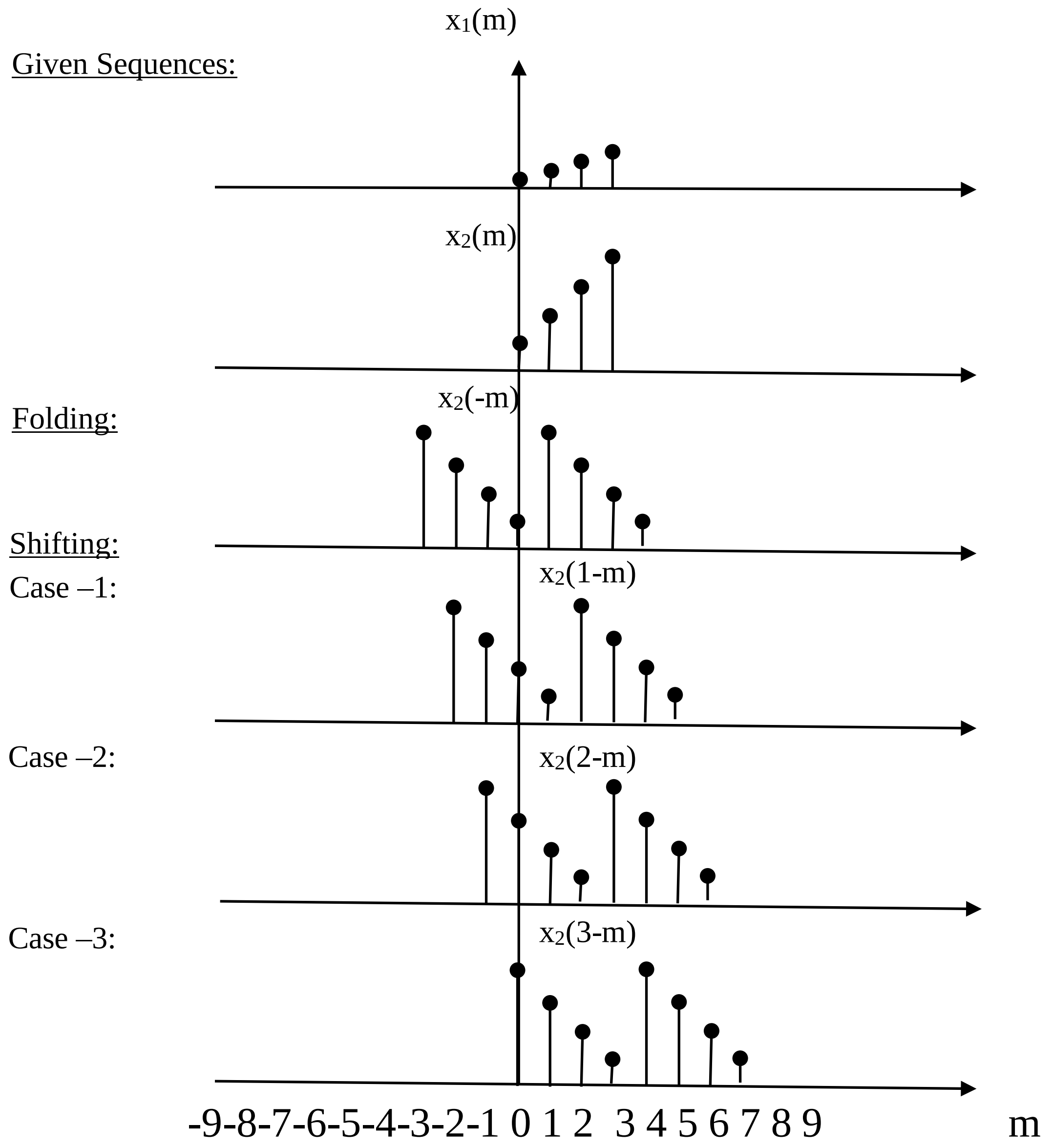
$x_2(n)$ ={5,6,7,8} with duration N=4 Samples

Duration of the circular convoluted sequence x(n), N= 4 samples

i.e., compute 4 samples of x(n) through the following procedure

- Draw the graphical representation of given sequences $x_1(m)$ and $x_2(m)$**.**
- Take the folding form of $x_2(m)$ to get $x_2(-m)$ with the periodic extension.
- Shift the folding sequence $x_2(-m)$ and draw $x_2(1-m)$ to $x_2(3-m)$.
- Finally apply circular convolution formula to get the circular convoluted sequence.

$$x(n) = x_1(n) * x_2(n) = \sum_{m=0}^{N-1} x_1(m)\, x_2(n-m)$$



Compute 4 samples of x(n) by substituting n=0 to 4 in $x(n) = x_1(n) * x_2(n) = \sum_{m=0}^{N-1} x_1(m)\, x_2(n-m)$

$$n = 0 \Rightarrow x(0) = \sum_{m=0}^{3} x_1(m)\, x_2(-m) = 1\times5 + 2\times8 + 3\times7 + 4\times6 = 5 + 16 + 21 + 24 = 66$$

$$n = 1 \Rightarrow x(1) = \sum_{m=0}^{3} x_1(m)\, x_2(1-m) = 1\times6 + 2\times5 + 3\times8 + 4\times7 = 6 + 10 + 24 + 28 = 68$$

$$n = 2 \Rightarrow x(2) = \sum_{m=0}^{3} x_1(m)\, x_2(2-m) = 1\times7 + 2\times6 + 3\times5 + 4\times8 = 7 + 12 + 15 + 32 = 66$$

$$n = 3 \Rightarrow x(3) = \sum_{m=0}^{3} x_1(m)\, x_2(3-m) = 1\times8 + 2\times7 + 3\times6 + 4\times5 = 8 + 14 + 18 + 20 = 60$$

Circular Convoluted Sequence $x(n) = x_1(n) * x_2(n) = \{66, 68, 66, 60\}$

(b)Tabular method:

Given sequences

$x_1(n)$ = {1,2,3,4} with duration N = 4 Samples and

$x_2(n)$ ={5,6,7,8} with duration N = 4 Samples

Duration of the circular convoluted sequence x(n), N = 4

i.e Compute 4 samples of x(n) through the following procedure

- Draw a table and represent the samples of $x_1(m)$ and $x_2(m)$ in a table**.**
- Take the folding form of $x_2(m)$ and represent $x_2(-m)$ in a table with periodic extension.
- Shift the folding sequence $x_2(-m)$ and represent the samples of $x_2(1-m)$ to $x_2(3-m)$ in a table
- Apply multiplication operation to get the samples of $x_1(m)$ $x_2(-m)$ to $x_1(m)$ $x_2(3-m)$.
- Finally apply summation operation to get the circular convoluted sequence.

$$x(n) = x_1(n) * x_2(n) \;=\; \sum_{m=0}^{N-1} x_1(m)\, x_2(n-m)$$

	m	**-4**	**-3**	**-2**	**-1**	**0**	**1**	**2**	**3**	**4**	
Given Sequences	**x_1(m)**					1	2	3	4		
	x_2(m)					5	6	7	8		
Folding	**x_2(-m)**		8	7	6	5	8	7	6		
Shifting	**x_2(1-m)**			8	7	6	5	8	7		
	x_2(2-m)				8	7	6	5	8		
	x_2(3-m)					8	7	6	5		Sum
Multiplication	**x_1(m) x_2(-m)**					5	16	21	24	=	**66**
	x_1(m) x_2(1-m)					6	10	24	28	=	**68**
	x_1(m) x_2(2-m)					7	12	15	32	=	**66**
	x_1(m) x_2(3-m)					8	14	18	20	=	**60**

Circular Convoluted Sequence x(n) = x_1(n) ∗ x_2(n) = {66, 68, 66, 60}

(c)Matrix method:

Given sequences

$x_1(n)$ = {1,2,3,4} with duration N = 4 Samples and

$x_2(n)$ ={5,6,7,8} with duration N = 5 Samples

Duration of the circular convoluted sequence x(n), N = 4 Samples

i.e., compute 4 samples of x(n) by representing $x_1(m)$ and $x_2(n-m)$ in the form of matrices and finally compute the circular convoluted sequence.

$$x(n) = x_1(n) * x_2(n) = \sum_{m=0}^{N-1} x_1(m)\, x_2(n-m)$$

$$= \begin{bmatrix} 1 & 2 & 3 & 4 \end{bmatrix} \begin{bmatrix} 5 & 6 & 7 & 8 \\ 8 & 5 & 6 & 7 \\ 7 & 8 & 5 & 6 \\ 6 & 7 & 8 & 5 \end{bmatrix}$$

$$= [1\times5 + 2\times8 + 3\times7 + 4\times6 \quad 1\times6 + 2\times5 + 3\times8 + 4\times7$$
$$= 1\times7 + 2\times6 + 3\times5 + 4\times8 \quad 1\times8 + 2\times7 + 3\times6 + 4\times5]$$
$$= [5+16+21+24 \quad 6+10+24+28 \quad 7+12+15+32 \quad 8+14+18+20]$$
$$= [66 \quad 68 \quad 66 \quad 60]$$

Circular Convoluted Sequence $x(n) = x_1(n) * x_2(n) = \{66, 68, 66, 60\}$

(d)Circular method:

Given sequences

$x_1(n) = \{1,2,3,4\}$ with duration N = 4 Samples and

$x_2(n) = \{5,6,7,8\}$ with duration N = 4 Samples

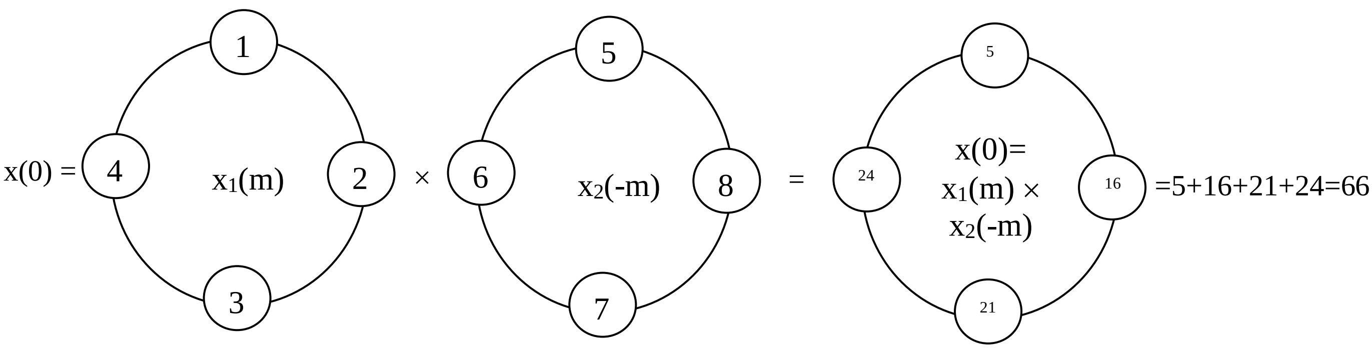
Duration of the circular convoluted sequence, x(n) is N = 4

i.e., compute 4 samples of x(n) through the following procedure

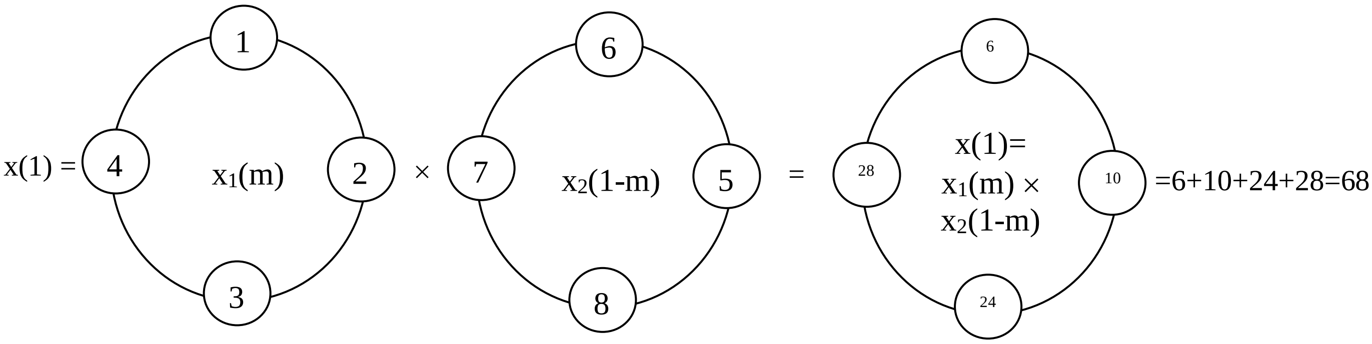
- Draw a circle and represent samples of $x_1(m)$ on a circle in clockwise direction.
- Draw a circle and represent samples of $x_2(m)$ on a circle in anticlockwise direction to get $x_2(-m)$. It is folding operation.
- Rotate the circle $x_2(-m)$ by $2\pi/N=2\pi/4=\pi/2$ radians in clockwise direction to get $x_2(1-m)$ and repeat the process to $x_2(2-m)$ and $x_2(3-m)$. It is shifting operation.
- Now multiply circles $x_1(m)$ and $x_2(n-m)$ and add corresponding samples to get x(0), x(1), x(2) and x(3). It includes multiplication and summation operations.

$$x(n) = x_1(n) * x_2(n) = \sum_{m=0}^{N-1} x_1(m)\, x_2(n-m)$$

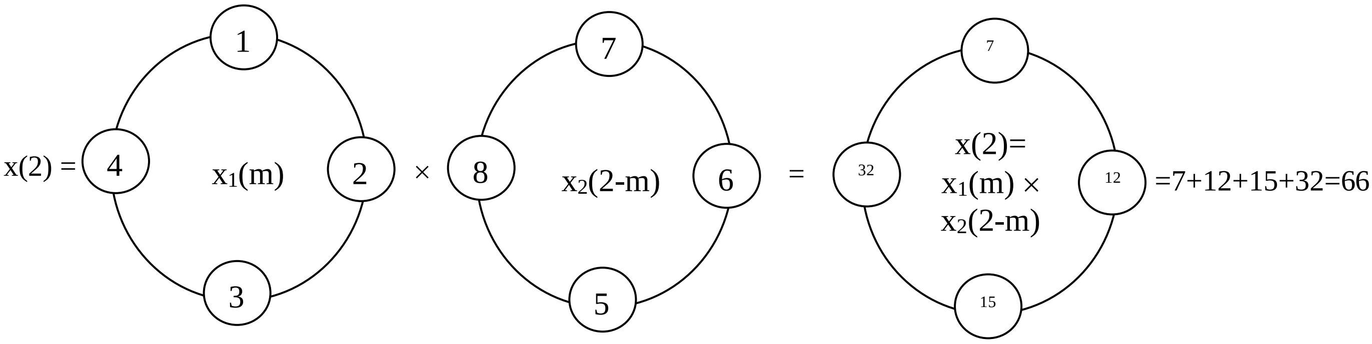
Case 1 : $x(0) = \sum_{m=0}^{3} x_1(m)\, x_2(-m)$



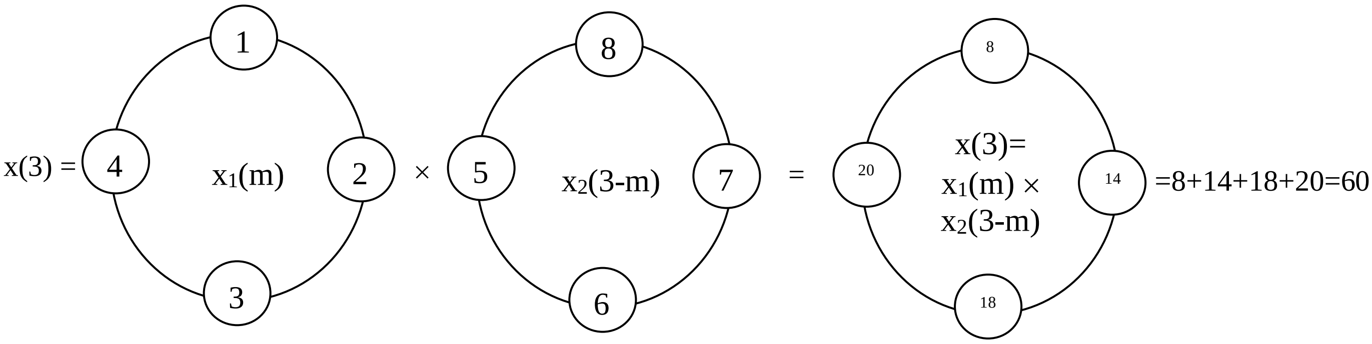
Case 2 : $x(1) = \sum_{m=0}^{3} x_1(m)\, x_2(1-m)$



Case 3 : $x(2) = \sum_{m=0}^{3} x_1(m)\, x_2(2-m)$



Case 4 : $x(3) = \sum_{m=0}^{3} x_1(m)\, x_2(3-m)$



Circular Convoluted Sequence $x(n) = x_1(n) * x_2(n) = \{66, 68, 66, 60\}$

(C)Linear Convolution through Circular Convolution:

Here, the task is to compute the linear convoluted sequence by using circular convolution method. If $x_1(n)$ and $x_2(n)$ are two finite duration sequences with the durations of N_1 and N_2 number of samples, then the duration of linear convoluted sequence is $N = N_1 + N_2 – 1$. Procedure to compute the linear convoluted sequence through circular convolution is given below.

- Pad the sequence $x_1(n)$ with "$N_2 – 1$" number of zeroes to get the length of $N_1 + N_2 – 1$.
- Pad the sequence $x_2(n)$ with "$N_1 – 1$" number of zeroes to get the length of $N_1 + N_2 – 1$.
- Now compute the circular convolution of $x_1(n)$ and $x_2(n)$ is called linear convoluted sequence.

Example:

Compute the linear convoluted sequence $x(n) = x_1(n) * x_2(n)$ through circular convolution using matrix method. Given $x_1(n) = \{1,2,3,4\}$ and $x_2(n) =\{5,6,7,8,9\}$

Given sequences

$x_1(n) = \{1,2,3,4\}$ with duration $N_1=4$ Samples and

$x_2(n) =\{5,6,7,8,9\}$ with duration $N_2=5$ Samples

Duration of the linear convoluted sequence x(n), $N= N_1 + N_2 – 1 =4+5-1=8$

i.e Compute 8 samples of x(n) through the following procedure

- Pad $x_1(n)$ with '$N_2 – 1=4$' number of zeros to get a length of 8 samples.

  $x_1(n) = \{1,2,3,4,0,0,0,0\}$

- Pad $x_2(n)$ with '$N_1 – 1=3$' number of zeros to get a length of 8 samples.

  $x_2(n) =\{5,6,7,8,9,0,0,0\}$

- Now represent $x_1(m)$ and $x_2(n-m)$ in the form of matrices and finally compute the linear convoluted sequence.

$$x(n) = x_1(n) * x_2(n) = \sum_{m=0}^{N-1} x_1(m)\, x_2(n-m)$$

$$= \begin{bmatrix} 1 & 2 & 3 & 4 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 5 & 6 & 7 & 8 & 9 & 0 & 0 & 0 \\ 0 & 5 & 6 & 7 & 8 & 9 & 0 & 0 \\ 0 & 0 & 5 & 6 & 7 & 8 & 9 & 0 \\ 0 & 0 & 0 & 5 & 6 & 7 & 8 & 9 \\ 9 & 0 & 0 & 0 & 5 & 6 & 7 & 8 \\ 8 & 9 & 0 & 0 & 0 & 5 & 6 & 7 \\ 7 & 8 & 9 & 0 & 0 & 0 & 5 & 6 \\ 6 & 7 & 8 & 9 & 0 & 0 & 0 & 5 \end{bmatrix}$$

$$= [1\times5 \quad 1\times6+2\times5 \quad 1\times7+2\times6+3\times5 \quad 1\times8+2\times7+3\times6+4\times5$$
$$1\times9+2\times8+3\times7+4\times6 \quad 2\times9+3\times8+4\times7 \quad 3\times9+4\times8 \quad 4\times9]$$
$$= [5 \quad 6+10 \quad 7+12+15 \quad 8+14+18+20 \quad 9+16+21+24 \quad 18+24+28 \quad 27+32 \quad 36]$$
$$= [5 \quad 16 \quad 34 \quad 60 \quad 70 \quad 70 \quad 59 \quad 36]$$

Linear Convoluted Sequence $x(n) = x_1(n) * x_2(n) = \{5, 16, 34, 60, 70, 70, 59, 36\}$

(D)Response of discrete LSI System through Circular Convolution:

Here, the task is to compute the response of discrete LSI system by using circular convolution method. If x(n) is input sequence and h(n) is impulse response of a discrete LSI system with the durations of N_1 and N_2 number of samples, then the duration of the response of discrete LSI system is $N = N_1 + N_2 - 1$. Procedure to compute the response of discrete LSI system through circular convolution is given below.

- Pad the input sequence x(n) with "$N_2 - 1$" number of zeroes to get a length of $N_1 + N_2 - 1$.
- Pad the impulse response h(n) with "$N_1 - 1$" number of zeroes to get a length of $N_1 + N_2 - 1$.
- Now compute the circular convolution of input x(n) and impulse response h(n) is called the response of discrete LSI system.

Example:

Compute the response of LSI system through circular convolution using matrix method. Given input sequence x(n) = {1,2,3,4} and impulse response h(n) ={5,6,7,8,9}

Given   Input sequence x(n)= {1,2,3,4} with duration N_1=4 Samples and

Impulse response h(n) ={5,6,7,8,9} with duration N_2=5 Samples

Duration of response of LSI system y(n), $N= N_1 + N_2 - 1$ =4+5-1=8

i.e Compute 8 samples of y(n) through the following procedure

- Pad x(n) with '$N_2 - 1$=4' number of zeros to get a length of 8 samples.
  x(n) = {1,2,3,4,0,0,0,0}
- Pad h(n) with '$N_1 - 1$=3' number of zeros to get a length of 8 samples.
  h(n) ={5,6,7,8,9,0,0,0}
- Now represent x(m) and h(n-m) in the form of matrices and finally compute the response of LSI system.

$$y(n) = x(n) * h(n) = \sum_{m=0}^{N-1} x(m)\, h(n-m)$$

$$= \begin{bmatrix} 1 & 2 & 3 & 4 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 5 & 6 & 7 & 8 & 9 & 0 & 0 & 0 \\ 0 & 5 & 6 & 7 & 8 & 9 & 0 & 0 \\ 0 & 0 & 5 & 6 & 7 & 8 & 9 & 0 \\ 0 & 0 & 0 & 5 & 6 & 7 & 8 & 9 \\ 9 & 0 & 0 & 0 & 5 & 6 & 7 & 8 \\ 8 & 9 & 0 & 0 & 0 & 5 & 6 & 7 \\ 7 & 8 & 9 & 0 & 0 & 0 & 5 & 6 \\ 6 & 7 & 8 & 9 & 0 & 0 & 0 & 5 \end{bmatrix}$$

$$= [1\times5 \quad 1\times6+2\times5 \quad 1\times7+2\times6+3\times5 \quad 1\times8+2\times7+3\times6+4\times5 \quad 1\times9+2\times8+3\times7+4\times6 \quad 2\times9+3\times8+4\times7 \quad 3\times9+4\times8 \quad 4\times9]$$

$$= [5 \quad 6+10 \quad 7+12+15 \quad 8+14+18+20 \quad 9+16+21+24 \quad 18+24+28 \quad 27+32 \quad 36]$$

$$= [5 \quad 16 \quad 34 \quad 60 \quad 70 \quad 70 \quad 59 \quad 36]$$

Response of LSI system y(n) = x(n) ∗ h(n) = {5, 16, 34, 60, 70, 70, 59, 36}

(E)Circular Convolution through DFT-IDFT:

Here, the task is to compute the circular convoluted sequence by DFT-IDFT. If $x_1(n)$ and $x_2(n)$ are two finite duration sequences with the equal duration of N number of samples, then the duration of a circular convoluted sequence is N. Procedure to compute the circular convoluted sequence through DFT-IDFT method is given below.

Step-I : Compute the N –Point DFT [$x_1(n)$], i.e $X_1(k)$

Step-II : Compute the N –Point DFT [$x_2(n)$], i.e $X_2(k)$.

Step-III : Now determine N samples of X(k), such that $X(k) = X_1(k). X_2(k)$.

Step-IV : Compute the N –Point IDFT [X(k)]=x(n). It is the circular convolution of $x_1(n)$ and $x_2(n)$.

Example:

Compute the circular convoluted sequence $x(n) = x_1(n) * x_2(n)$ using DFT-IDFT method.

Given $x_1(n) = \{1,2,3,4\}$ and $x_2(n) = \{5,6,7,8\}$

Step-I: Compute the 4 –Point DFT [$x_1(n)$], i.e $X_1(k)$

From the basic definition of N-Point DFT of a sequence x(n)

$$N-Point\,DFT[\,x(n)\,] = \sum_{n=0}^{N-1} x(n)\,W_N^{nk}$$

Given sequence $x_1(n)=\{1,2,3,4\}$ and N=4

$$X_1(k) = \sum_{n=0}^{3} x_1(n)\,W_4^{nk}$$

$$= x_1(0)\,W_4^{0k} + x_1(1)\,W_4^{1k} + x_1(2)\,W_4^{2k} + x_1(3)\,W_4^{3k}$$

$$= 1 + 2W_4^{k} + 3W_4^{2k} + 4W_4^{3k}$$

Compute 4 samples of $X_1(k)$ by substituting k=0,1,2,3

$$k=0 \Rightarrow X_1(0) = 1 + 2W_4^0 + 3W_4^0 + 4W_4^0 = 1+2+3+4 = 10$$

$$k=1 \Rightarrow X_1(1) = 1 + 2W_4^1 + 3W_4^2 + 4W_4^3 = 1 + 2(-j) + 3(-1) + 4(j) = -2 + 2j$$

$$k=2 \Rightarrow X_1(2) = 1 + 2W_4^2 + 3W_4^4 + 4W_4^6 = 1 + 2(-1) + 3(1) + 4(-1) = -2$$

$$k=3 \Rightarrow X_1(3) = 1 + 2W_4^3 + 3W_4^6 + 4W_4^9 = 1 + 2(j) + 3(-1) + 4(-j) = -2 - 2j$$

$$\boxed{4-Point\,DFT[\,x_1(n)\,] = X_1(k) = \{10, -2+2j, -2, -2-2j\}}$$

Step-II: Compute the 4 –Point DFT [$x_2(n)$], i.e $X_2(k)$

From the basic definition of N-Point DFT of a sequence x(n)

$$N-\text{Point}\, DFT[\, x(n)\,] = \sum_{n=0}^{N-1} x(n)\, W_N^{nk}$$

Given sequence $x_2(n)=\{5,6,7,8\}$ and N=4

$$X_2(k) = \sum_{n=0}^{3} x_2(n)\, W_4^{nk}$$

$$= x_2(0)\, W_4^{0k} + x_2(1)\, W_4^{1k} + x_2(2)\, W_4^{2k} + x_2(3)\, W_4^{3k}$$

$$= 5 + 6W_4^{k} + 7W_4^{2k} + 8W_4^{3k}$$

Compute 4 samples of $X_2(k)$ by substituting k=0,1,2,3

$$k=0 \Rightarrow X_2(0) = 5 + 6W_4^0 + 7W_4^0 + 8W_4^0 = 5+6+7+8 = 26$$

$$k=1 \Rightarrow X_2(1) = 5 + 6W_4^1 + 7W_4^2 + 8W_4^3 = 5 + 6(-j) + 7(-1) + 8(j) = -2+2j$$

$$k=2 \Rightarrow X_2(2) = 5 + 6W_4^2 + 7W_4^4 + 8W_4^6 = 5 + 6(-1) + 7(1) + 8(-1) = -2$$

$$k=3 \Rightarrow X_2(3) = 5 + 6W_4^3 + 7W_4^6 + 8W_4^9 = 5 + 6(j) + 7(-1) + 8(-j) = -2-2j$$

$$\boxed{4-\text{Point}\, DFT[\, x_2(n)\,] = X_2(k) = \{26, -2+2j, -2, -2-2j\}}$$

Step-III : Now determine 4 samples of X(k), such that X(k) = X_1(k). X_2(k).

$$X(k) = X_1(k) X_2(k)$$

$$= \{10, -2+2j, -2, -2-2j\} \times \{26, -2+2j, -2, -2-2j\}$$

$$= \{10\times 26, (-2+2j)^2, -2\times -2, (-2-2j)^2\}$$

$$= \{260, -8j, 4, 8j\}$$

Step-IV : Compute the 4 –Point IDFT [X(k)], i.e $N-\text{Point}\, IDFT[\, X(k)\,] = \frac{1}{N}\sum_{k=0}^{N-1} X(k)\, W_N^{-nk}$

$$x(n) = \frac{1}{4}\sum_{k=0}^{3} X(k)\, W_4^{-nk}$$

$$= \frac{1}{4}\left(X(0)\, W_4^{-n0} + X(1)\, W_4^{-n1} + X(2)\, W_4^{-n2} + X(3)\, W_4^{-n3} \right)$$

$$= \frac{1}{4}\left(260 + (-8j)W_4^{-n} + 4W_4^{-2n} + (8j)W_4^{-3n} \right)$$

$$= 65 - 2jW_4^{-n} + W_4^{-2n} + 2jW_4^{-3n}$$

Compute 4 samples of x(n) by substituting n=0,1,2,3

$$n=0 \Rightarrow x(0) = 65-2jW_4^{-0}+W_4^{-0}+2jW_4^{-0}=65-2j(1)+1+2j(1)=65-2j+1+2j=66$$

$$n=1 \Rightarrow x(1) = 65-2jW_4^{-1}+W_4^{-2}+2jW_4^{-3}=65-2j(j)+(-1)+2j(-j)=65+2-1+2=68$$

$$n=2 \Rightarrow x(2) = 65-2jW_4^{-2}+W_4^{-4}+2jW_4^{-6}=65-2j(-1)+1+2j(-1)=65+2j+1-2j=66$$

$$n=3 \Rightarrow x(3) = 65-2jW_4^{-3}+W_4^{-6}+2jW_4^{-9}=65-2j(-j)+(-1)+2j(j)=65-2-1-2=60$$

Circular convolution of $x_1(n)$ and $x_2(n)$ is x(n) = {66, 68, 66, 60}

(F)Linear Convolution through DFT-IDFT:

Here, the task is to compute the linear convoluted sequence by using DFT-IDFT method. If $x_1(n)$ and $x_2(n)$ are two finite duration sequences with the durations of N_1 and N_2 number of samples, then the duration of linear convoluted sequence is $N = N_1 + N_2 - 1$. Procedure to compute the linear convoluted sequence through DFT-IDFT method is given below.

Step-I : Pad the sequence $x_1(n)$ with "$N_2 - 1$" number of zeroes to get the length of $N_1 + N_2 - 1$.

Step-II : Pad the sequence $x_2(n)$ with "$N_1 - 1$" number of zeroes to get the length of $N_1 + N_2 - 1$.

Step-III : Compute the N –Point DFT [$x_1(n)$], i.e $X_1(k)$.

Step-IV : Compute the N –Point DFT [$x_2(n)$], i.e $X_2(k)$.

Step-V : Now determine N samples of X(k), such that $X(k) = X_1(k). X_2(k)$.

Step-VI : Compute the N –Point IDFT [X(k)] i.e x(n). It is the linear convolution of $x_1(n)$ and $x_2(n)$.

Example:

Compute the linear convoluted sequence $x(n) = x_1(n) * x_2(n)$ using DFT-IDFT method.

Given $x_1(n)$ = {1,2} and $x_2(n)$ = {3,4,5}

Given sequences

$x_1(n)$ = {1,2} with duration N_1=2 Samples and

$x_2(n)$ ={3,4,5} with duration N_2=3 Samples

Duration of the linear convoluted sequence x(n), $N= N_1 + N_2 - 1$ =2+3-1=4

i.e Compute 4 samples of x(n) through the following procedure

Step-I : Pad $x_1(n)$ with '$N_2 - 1=2$' number of zeros to get a length of 4 samples.

$x_1(n) = \{1,2,0,0\}$

Step-II : Pad $x_2(n)$ with '$N_1 - 1=1$' number of zero to get a length of 4 samples.

$x_2(n) =\{3,4,5,0\}$

Step-III: Compute the 4 –Point DFT $[x_1(n)]$, i.e $X_1(k)$

From the basic definition of N-Point DFT of a sequence x(n)

$$N-Point\,DFT[\,x(n)\,] = \sum_{n=0}^{N-1} x(n)\,W_N^{nk}$$

Given sequence $x_1(n)=\{1,2,0,0\}$ and N=4

$$X_1(k) = \sum_{n=0}^{3} x_1(n)\,W_4^{nk}$$

$$= x_1(0)\,W_4^{0k} + x_1(1)\,W_4^{1k} + x_1(2)\,W_4^{2k} + x_1(3)\,W_4^{3k}$$

$$= 1 + 2W_4^{k}$$

Compute 4 samples of $X_1(k)$ by substituting k=0,1,2,3

$$k = 0 \Rightarrow X_1(0) = 1 + 2W_4^{0} = 1 + 2 = 3$$

$$k = 1 \Rightarrow X_1(1) = 1 + 2W_4^{1} = 1 + 2(-j) = 1 - 2j$$

$$k = 2 \Rightarrow X_1(2) = 1 + 2W_4^{2} = 1 + 2(-1) = 1 - 2 = -1$$

$$k = 3 \Rightarrow X_1(3) = 1 + 2W_4^{3} = 1 + 2(j) = 1 + 2j$$

$$\boxed{4-Point\,DFT[\,x_1(n)\,] = X_1(k) = \{3, 1-2j, -1, 1+2j\}}$$

Step-IV: Compute the 4 –Point DFT $[x_2(n)]$, i.e $X_2(k)$

From the basic definition of N-Point DFT of a sequence x(n)

$N-Point\,DFT[\,x(n)\,] = \sum_{n=0}^{N-1} x(n)\,W_N^{nk}$, given sequence $x_2(n)=\{3,4,5,0\}$ and N=4

$$X_2(k) = \sum_{n=0}^{3} x_2(n)\,W_4^{nk}$$

$$= x_2(0)\,W_4^{0k} + x_2(1)\,W_4^{1k} + x_2(2)\,W_4^{2k} + x_2(3)\,W_4^{3k}$$

$$= 3 + 4W_4^{k} + 5W_4^{2k}$$

Compute 4 samples of $X_2(k)$ by substituting k=0,1,2,3

$$k=0 \Rightarrow X_2(0) = 3+4W_4^0+5W_4^0 = 3+4+5 = 12$$

$$k=1 \Rightarrow X_2(1) = 3+4W_4^1+5W_4^2 = 3+4(-j)+5(-1) = 3-4j-5 = -2-4j$$

$$k=2 \Rightarrow X_2(2) = 3+4W_4^2+5W_4^4 = 3+4(-1)+5(1) = 3-4+5 = 4$$

$$k=3 \Rightarrow X_2(3) = 3+4W_4^3+5W_4^6 = 3+4(j)+5(-1) = 3+4j-5 = -2+4j$$

$$\boxed{4-Point\,DFT[\,x_2(n)\,] = X_2(k) = \{12,-2-4j,4,-2+4j\}}$$

Step-V : Now determine 4 samples of X(k), such that X(k) = X_1(k). X_2(k).

$$\begin{aligned} X(k) &= X_1(k)X_2(k) \\ &= \{3,1-2j,-1,1+2j\}\times\{12,-2-4j,4,-2+4j\} \\ &= \{3\times 12,(1-2j)(-2-4j),-1\times 4,(1+2j)(-2+4j)\} \\ &= \{36,-10,-4,-10\} \end{aligned}$$

Step-VI : Compute the 4 –Point IDFT [X(k)], i.e $N-Point\,IDFT[\,X(k)\,] = \frac{1}{N}\sum_{k=0}^{N-1} X(k)\,W_N^{-nk}$

$$\begin{aligned} x(n) &= \frac{1}{4}\sum_{k=0}^{3} X(k)\,W_4^{-nk} \\ &= \frac{1}{4}\left(X(0)\,W_4^{-n0} + X(1)\,W_4^{-n1} + X(2)\,W_4^{-n2} + X(3)\,W_4^{-n3} \right) \\ &= \frac{1}{4}\left(36+(-10)W_4^{-n}+(-4)W_4^{-2n}+(-10)W_4^{-3n} \right) \\ &= 9-(5/2)W_4^{-n}-W_4^{-2n}-(5/2)W_4^{-3n} \end{aligned}$$

Compute 4 samples of x(n) by substituting n=0,1,2,3

$$n=0 \Rightarrow x(0) = 9-(5/2)W_4^{-0}-W_4^{-0}-(5/2)W_4^{-0} = 9-(5/2)(1)-1-(5/2)(1) = 9-6 = 3$$

$$n=1 \Rightarrow x(1) = 9-(5/2)W_4^{-1}-W_4^{-2}(5/2)W_4^{-3} = 9-(5/2)(j)-(-1)-(5/2)(-j) = 9+1 = 10$$

$$n=2 \Rightarrow x(2) = 9-(5/2)W_4^{-2}-W_4^{-4}-(5/2)W_4^{-6} = 9-(5/2)(-1)-1-(5/2)(-1) = 9+4 = 13$$

$$n=3 \Rightarrow x(3) = 9-(5/2)W_4^{-3}-W_4^{-6}-(5/2)W_4^{-9} = 9-(5/2)(-j)-(-1)-(5/2)(j) = 9+1 = 10$$

Linear convolution of x_1(n) and x_2(n) is x(n) = {3, 10, 13, 10}

Fast Fourier Transform (FFT):

Fast Fourier Transform (FFT) is a method or algorithm, which is used to compute Discrete Fourier Transform (DFT) and Inverse Discrete Fourier Transform (IDFT) with

- Reduced number of calculations in terms of complex additions and multiplications.
- Increased percentage of saving in the evaluation of complex additions and multiplications.
- Increased speed of FFT method over DFT computation.

(A) Direct Computation of N-Point DFT:

We know that the computation of N–Point DFT of a finite duration sequence x(n)

$$X(k) = \sum_{n=0}^{N-1} x(n)\, W_N^{nk}$$

$$X(k) = x(0)\, W_N^{0k} + x(1)\, W_N^{1k} + x(2)\, W_N^{2k} + \ldots\ldots\ldots\ldots\ldots\ldots\ldots + x(N-1)\, W_N^{(N-1)k}$$

N samples of X(k) can be computed by substituting k = 0, 1, 2, …….. N – 1.

$$k = 0 \Rightarrow X(0) = x(0)\, W_N^{0} + x(1)\, W_N^{0} + x(2)\, W_N^{0} + \ldots\ldots\ldots\ldots\ldots\ldots\ldots + x(N-1)\, W_N^{0}$$

$$k = 1 \Rightarrow X(1) = x(0)\, W_N^{0} + x(1)\, W_N^{1} + x(2)\, W_N^{2} + \ldots\ldots\ldots\ldots\ldots\ldots\ldots + x(N-1)\, W_N^{N-1}$$

$$k = 2 \Rightarrow X(2) = x(0)\, W_N^{0} + x(1)\, W_N^{2} + x(2)\, W_N^{4} + \ldots\ldots\ldots\ldots\ldots\ldots\ldots + x(N-1)\, W_N^{(N-1)2}$$

.

.

.

$$k = N-1 \Rightarrow X(N-1) = x(0)\, W_N^{0} + x(1)\, W_N^{N-1} + x(2)\, W_N^{2(N-1)} + \ldots\ldots\ldots + x(N-1)\, W_N^{(N-1)(N-1)}$$

In each line of above N-point DFT computation involves 'N – 1' number of complex additions and 'N' number of complex multiplications, such lines are 'N', therefore the direct computation of N-Point DFT involves

- Number of Complex Additions (A) = N(N – 1).
- Number of Complex Multiplications (B) = N×N = N^2.

Ex: Direct computation of 10-Point DFT involves

- Number of Complex Additions (A) = 10(10 – 1) = 10(9) = 90.
- Number of Complex Multiplications (B) = 10×10 = 10^2 = 100.

Direct computation of N-Point DFT involves more number of calculations, which are in the form of complex additions and complex multiplications. To reduce above number of calculations, we can use indirect computation of N-Point DFT is called Fast Fourier Transform (FFT).

(B) Indirect Computation of N-Point DFT (FFT):

We know that the direct computation of N-Point DFT involves more number of calculations, which are in the form of complex additions and complex multiplications. To reduce those calculations, we can use indirect computation of N-Point DFT is called Fast Fourier Transform (FFT).

In FFT method, N can be expressed as

$$N = r^{m}$$

Where,

r : Radix number (minimum value is 2).

m : Required number of stages to compute N-Point DFT in FFT method

Indirect computation of N-Point DFT (FFT method) involves

- Number of Complex Additions (a) = $Nm = N \log_2 N$
- Number of Complex Multiplications (b) = $(N/2)m = (N/2) \log_2 N$

In FFT method, the total computation is divided into 'm' number of stages and all those 'm' stages are cascaded to compute N-point DFT, where the total computation consists of

- Two N/2-point DFT's
- Four N/4-point DFT's
- Eight N/8-point DFTs and so on.

  This process will continue until we get 2-Point DFT, which is called radix number.

The process of converting N-point DFT into smaller point DFTs is known as decimation. The decimation process may starts from time domain or from frequency domain, based on the decimation process, FFT algorithms are classified into two types.

- Decimation in Time (DIT) radix-2 FFT algorithm
- Decimation in Frequency (DIF) radix-2 FFT algorithm

Note :
We can develop DIT radix-3 FFT or DIF radix-3 FFT algorithms, where the radix number r=3.

(C)Comparison b/n Direct Computation of N-Point DFT & FFT:

No. of points	Direct Computation of N-Point DFT involves		Indirect Computation of N-Point DFT involves (DIT radix-2 FFT / DIF radix-2 FFT)			Percentage of saving due to		Speed of FFT due to	
N	Complex Additions $A=N(N-1)$	Complex Multiplications $B=N^2$	No. of Stages $m=\text{Log}_2N$	Complex Additions $a=N.m$	Complex Multiplications $b=(N/2)m$	Complex Additions $(1- a/A) \times 100$	Complex Multiplications $(1-b/B) \times 100$	Complex Additions A / a	Complex Multiplications B / b
4	12	16	2	8	4	33.33%	75.00%	1.50	4.00
8	56	64	3	24	12	57.14%	81.25%	2.33	5.33
16	240	256	4	64	32	73.33%	87.50%	3.75	8.00
32	992	1,024	5	160	80	83.87%	92.19%	6.20	12.80
64	4,032	4,096	6	384	192	90.48%	95.31%	10.50	21.33
128	16,256	16,384	7	896	448	94.49%	97.27%	18.14	36.57
256	65,280	65,536	8	2,048	1,024	96.86%	98.44%	31.86	64.00
512	2,61,632	2,62,144	9	4,608	2,304	98.24%	99.12%	56.78	113.78
1024	10,47,552	10,48,576	10	10,240	5,120	99.02%	99.51%	102.3	204.8

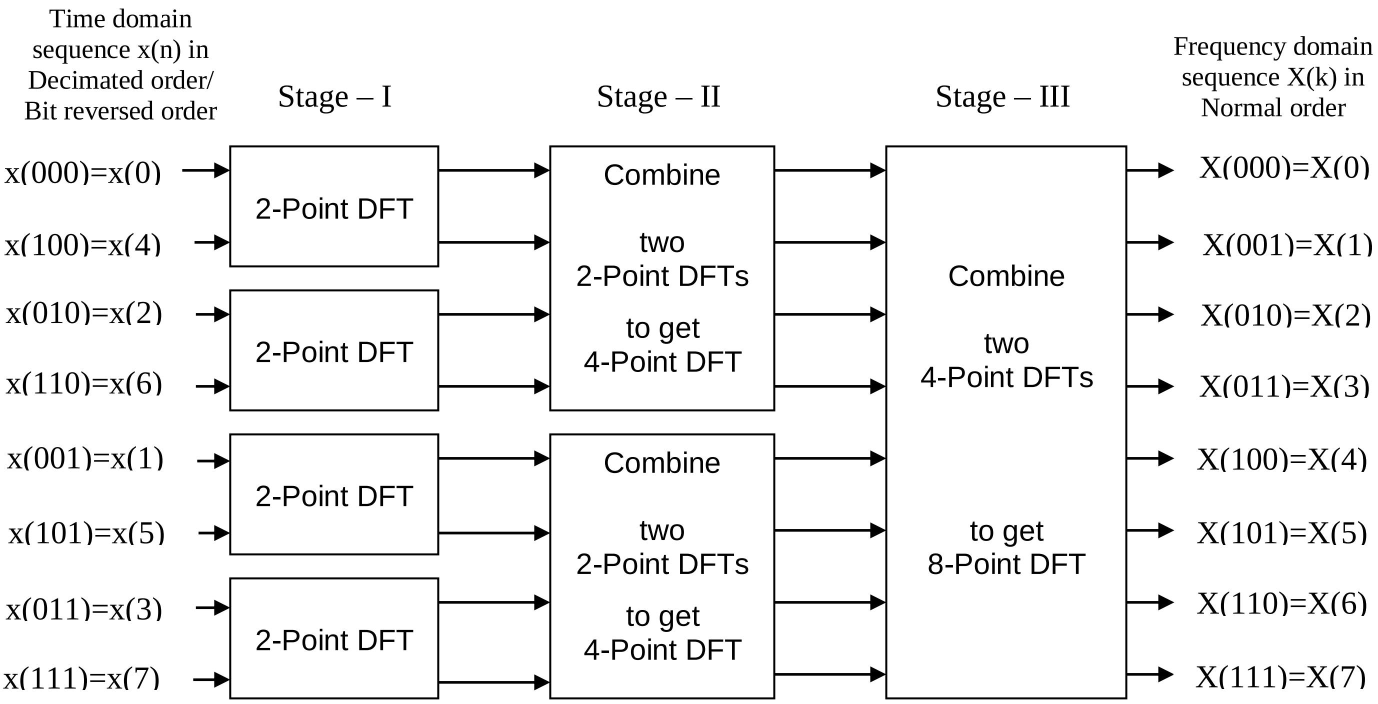
Advantages of FFT method over direct computation of DFT

- Number of complex additions and complex multiplications are reduced.
- Percentage saving due to complex additions and complex multiplications increases.
- Speed of FFT method increases.

(D)Decimation in Time (DIT) radix–2 FFT Algorithm:

In DIT radix-2 FFT, the time domain sequence x(n) is decimated and it is in bit reversed order and frequency domain sequence X(k) is in normal order. In the first stage of DIT radix–2 FFT, the time domain N-point sequence x(n) is decimated into possible number of 2-point sequences. In the second stage, compute the possible number of 4-point DFTs by combining two 2-point DFTs. In the third stage, compute possible number of 8-point DFTs by combining two 4-point DFTs. This process will continue until we get the N-point DFT.

Ex: Three stage computational structure for DIT radix-2 FFT as shown below



Step-I:

From the basic definition of N-point DFT of a sequence x(n)

$$X(k)=\sum_{n=0}^{N-1} x(n)\, W_N^{nk}$$

Separate even and odd locations of n by replacing n with 2n and 2n + 1

$$=\sum_{2n=0,2,4..}^{N-2} x(2n)\, W_N^{2nk}+\sum_{2n+1=1,3,5..}^{N-1} x(2n+1)\, W_N^{(2n+1)k}$$

$$=\sum_{n=0,1,2..}^{\frac{N}{2}-1} x(2n)\, W_N^{2nk}+\sum_{n=1,2,3..}^{\frac{N}{2}-1} x(2n+1)\, W_N^{2nk} W_N^{k}$$

$$=\sum_{n=0}^{\frac{N}{2}-1} x(2n)\, W_{N/2}^{nk}+W_N^{k}\sum_{n=0..}^{\frac{N}{2}-1} x(2n+1)\, W_{N/2}^{nk}$$

$$=\frac{N}{2}-\text{point DFT}\,[\,x(2n)\,]+W_N^{k}\,\frac{N}{2}-\text{point DFT}\,[\,x(2n+1)\,]$$

$$X(k)=I_1(k)+W_N^{k}\, I_2(k), X(N+k)=X(k)$$

Where, X(k) is periodic with a period of N samples, $I_1(k)$ and $I_2(k)$ are periodic with a period of N/2 samples. In step-1, our task is to compute N-point DFT of x(n) by combining two N/2-Point DFTs.

$$I_1(k) = \sum_{n=0}^{\frac{N}{2}-1} x(2n)\, W_{N/2}^{nk} \quad \text{-------------}1, \quad I_1(N/2+k) = I_1(k)$$

$$I_2(k) = \sum_{n=0..}^{\frac{N}{2}-1} x(2n+1)\, W_{N/2}^{nk} \quad \text{-----------}2, \quad I_2(N/2+k) = I_2(k)$$

Ex: For N=8, we have to compute 8 samples of X(k) by substituting k=0,1,2,3,4,5,6,7.

$$X(0) = I_1(0) + W_8^0\, I_2(0)$$

$$X(1) = I_1(1) + W_8^1\, I_2(1)$$

$$X(2) = I_1(2) + W_8^2\, I_2(2)$$

$$X(3) = I_1(3) + W_8^3\, I_2(3)$$

$$X(4) = I_1(4) + W_8^4\, I_2(4) = I_1(0) - W_8^0\, I_2(0)$$

$$X(5) = I_1(5) + W_8^5\, I_2(5) = I_1(1) - W_8^1\, I_2(1)$$

$$X(6) = I_1(6) + W_8^6\, I_2(6) = I_1(2) - W_8^2\, I_2(2)$$

$$X(7) = I_1(7) + W_8^7\, I_2(7) = I_1(3) - W_8^3\, I_2(3)$$

$I_1(4+k) = I_1(k)$

$I_1(4) = I_1(0)$, $I_1(5) = I_1(1)$,

$I_1(6) = I_1(2)$, $I_1(7) = I_1(3)$.

$I_2(4+k) = I_2(k)$

$I_2(4) = I_2(0)$, $I_2(5) = I_2(1)$,

$I_2(6) = I_2(2)$, $I_2(7) = I_2(3)$.

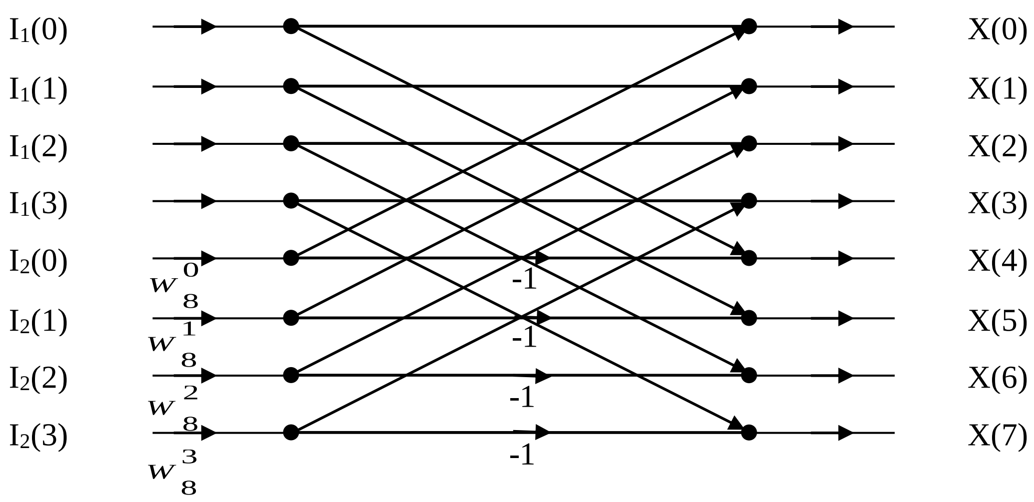
$$W_8^4 = W_8^{4+0} = W_8^4 W_8^0 = -W_8^0$$

$$W_8^5 = W_8^{4+1} = W_8^4 W_8^1 = -W_8^1$$

$$W_8^6 = W_8^{4+2} = W_8^4 W_8^2 = -W_8^2$$

$$W_8^7 = W_8^{4+3} = W_8^4 W_8^3 = -W_8^3$$

Signal flow graph of step-I or stage-III (Butterfly Structure)



Step-II:

Equation 1 $\Rightarrow I_1(k) = \sum_{n=0}^{\frac{N}{2}-1} x(2n)\, W_{N/2}^{nk}$

Replace n with 2n and 2n + 1

$$I_1(k) = \sum_{n=0}^{\frac{N}{4}-1} x(2(2n))\, W_{N/2}^{2nk} + \sum_{n=0}^{\frac{N}{4}-1} x(2(2n+1))\, W_{N/2}^{(2n+1)k}$$

$$= \sum_{n=0}^{\frac{N}{4}-1} x(4n)\, W_{N/2}^{2nk} + \sum_{n=0}^{\frac{N}{4}-1} x(4n+2))\, W_{N/2}^{2nk}\; W_{N/2}^{k}$$

$$= \sum_{n=0}^{\frac{N}{4}-1} x(4n)\, W_{N/4}^{nk} + W_{N/2}^{k} \sum_{n=0..}^{\frac{N}{4}-1} x(4n+2)\, W_{N/4}^{nk}$$

$$= \frac{N}{4} - \text{Point DFT}\,[\,x(4n)\,] + W_{N/2}^{k}\; \frac{N}{4} - \text{Point DFT}\,[\,x(4n+2)\,]$$

$$I_1(k) = I_3(k) + W_{N/2}^{k}\; I_4(k)$$

Where, $I_1(k)$ is periodic with a period of N/2 samples, $I_3(k)$ and $I_4(k)$ are periodic with a period of N/4 samples. Here, we have to compute N/2-point DFT of $I_1(k)$ by combining two N/4-Point DFTs.

$$I_3(k) = \sum_{n=0}^{\frac{N}{4}-1} x(4n)\, W_{N/4}^{nk} \quad ----------- 3, \quad I_3(N/4+k) = I_3(k)$$

$$I_4(k) = \sum_{n=0..}^{\frac{N}{4}-1} x(4n+2)\, W_{N/4}^{nk} \quad --------- 4, \quad I_4(N/4+k) = I_4(k)$$

Equation 2 $\Rightarrow I_2(k) = \sum_{n=0}^{\frac{N}{2}-1} x(2n+1)\, W_{N/2}^{nk}$

Replace n with 2n and 2n + 1

$$I_2(k) = \sum_{n=0}^{\frac{N}{4}-1} x(2(2n)+1)\, W_{N/2}^{2nk} + \sum_{n=0}^{\frac{N}{4}-1} x(2(2n+1)+1)\, W_{N/2}^{(2n+1)k}$$

$$= \sum_{n=0}^{\frac{N}{4}-1} x(4n+1)\, W_{N/2}^{2nk} + \sum_{n=0}^{\frac{N}{4}-1} x(4n+3)\, W_{N/2}^{2nk}\; W_{N/2}^{k}$$

$$= \sum_{n=0}^{\frac{N}{4}-1} x(4n+1)\, W_{N/4}^{nk} + W_{N/2}^{k} \sum_{n=0..}^{\frac{N}{4}-1} x(4n+3)\, W_{N/4}^{nk}$$

$$= \frac{N}{4} - \text{point DFT}\,[\,x(4n+1)\,] + W_{N/2}^{k}\; \frac{N}{4} - \text{point DFT}\,[\,x(4n+3)\,]$$

$$I_2(k) = I_5(k) + W_{N/2}^{k}\; I_6(k)$$

Where, $I_2(k)$ is periodic with a period of N/2 samples, $I_5(k)$ and $I_6(k)$ are periodic with a period of N/4 samples. Here, we have to compute N/2-point DFT of $I_2(k)$ by combining two N/4-Point DFTs.

$$I_5(k) = \sum_{n=0}^{\frac{N}{4}-1} x(4n+1)\, W_{N/4}^{nk} \quad ---------5, \quad I_5(N/4+k) = I_5(k)$$

$$I_6(k) = \sum_{n=0..}^{\frac{N}{4}-1} x(4n+3)\, W_{N/4}^{nk} \quad ----------6, \quad I_6(N/4+k) = I_6(k)$$

Ex: For N=8, we have to compute 4 samples from $I_1(k)$ and another 4 samples of from $I_2(k)$ by substituting k=0,1,2,3.

$$I_1(0) = I_3(0) + W_4^0\ I_4(0)$$

$$I_1(1) = I_3(1) + W_4^1\ I_4(1)$$

$$I_1(2) = I_3(2) + W_4^2\ I_4(2) = I_3(0) - W_4^0\ I_4(0)$$

$$I_1(3) = I_3(3) + W_4^3\ I_4(3) = I_3(1) - W_4^1\ I_4(1)$$

$$I_2(0) = I_5(0) + W_4^0\ I_6(0)$$

$$I_2(1) = I_5(1) + W_4^1\ I_6(1)$$

$$I_2(2) = I_5(2) + W_4^2\ I_6(2) = I_5(0) - W_4^0\ I_6(0)$$

$$I_2(3) = I_5(3) + W_4^3\ I_6(3) = I_5(1) - W_4^1\ I_6(1)$$

$I_3(2+k) = I_3(k)$

$I_3(2) = I_3(0)$, $I_3(3) = I_3(1)$,

$I_4(2+k) = I_4(k)$

$I_4(2) = I_4(0)$, $I_4(3) = I_4(1)$,

$I_5(2+k) = I_5(k)$

$I_5(2) = I_5(0)$, $I_5(3) = I_5(1)$,

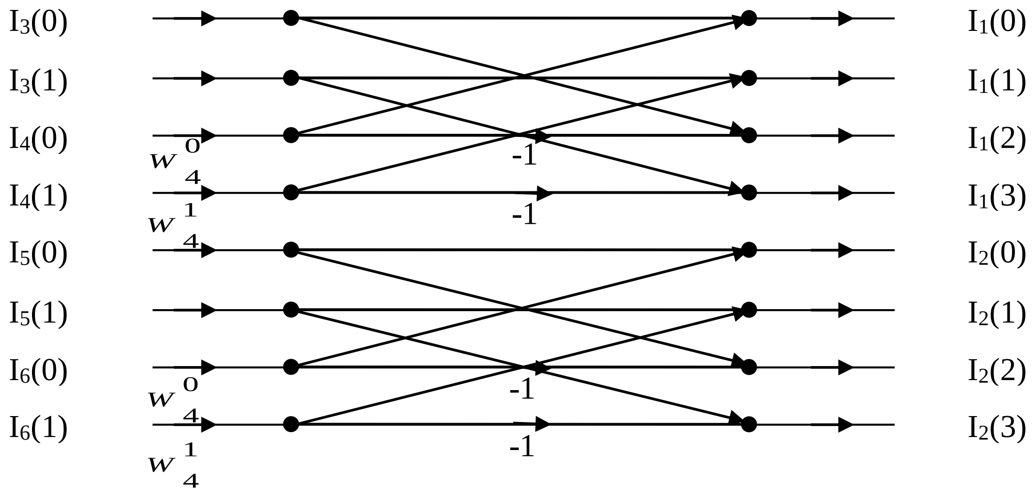
$I_6(2+k) = I_6(k)$

$I_6(2) = I_6(0)$, $I_6(3) = I_6(1)$,

$$W_4^2 = W_4^{2+0} = W_4^2 W_4^0 = -W_4^0$$

$$W_4^3 = W_4^{2+1} = W_4^2 W_4^1 = -W_4^1$$

Signal flow graph of step-II or stage-II (Butterfly Structure)



Step-III:

$I_3(k)$, $I_4(k)$, $I_5(k)$ and $I_6(k)$ are N/4-Point DFTs, if N=8, then $I_3(k)$, $I_4(k)$, $I_5(k)$ and $I_6(k)$ are 2-Point DFTs. Hence, in step-III no need to go for decimation process, because we have reached the 2-Point DFT i.e radix-2 for N=8.

$$\text{Equation 3} \Rightarrow I_3(k) = \sum_{n=0}^{\frac{N}{4}-1} x(4n)\, W_{N/4}^{nk} = \sum_{n=0}^{1} x(4n)\, W_2^{nk} = x(0) + W_2^{k}\, x(4)$$

$$\Rightarrow I_3(0) = x(0) + W_2^{0}\, x(4) \text{ and } I_3(1) = x(0) + W_2^{1}\, x(4) = x(0) - W_2^{0}\, x(4)$$

$$\text{Equation 4} \Rightarrow I_4(k) = \sum_{n=0..}^{\frac{N}{4}-1} x(4n+2)\, W_{N/4}^{nk} = \sum_{n=0}^{1} x(4n+2)\, W_2^{nk} = x(2) + W_2^{k}\, x(6)$$

$$\Rightarrow I_4(0) = x(2) + W_2^{0}\, x(6) \text{ and } I_4(1) = x(2) + W_2^{1}\, x(6) = x(2) - W_2^{0}\, x(6)$$

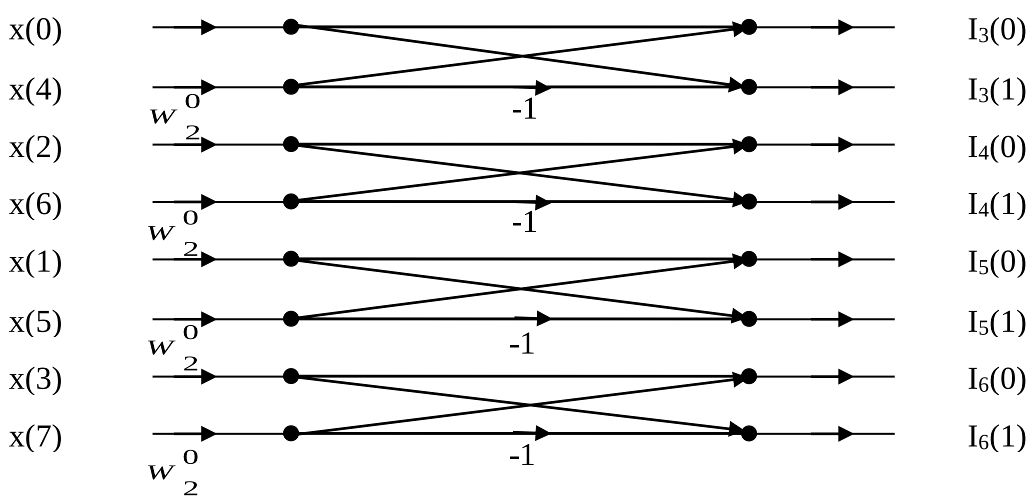
$$\text{Equation 5} \Rightarrow I_5(k) = \sum_{n=0}^{\frac{N}{4}-1} x(4n+1)\, W_{N/4}^{nk} = \sum_{n=0}^{1} x(4n+1)\, W_2^{nk} = x(1) + W_2^{k}\, x(5)$$

$$\Rightarrow I_5(0) = x(1) + W_2^{0}\, x(5) \text{ and } I_5(1) = x(1) + W_2^{1}\, x(5) = x(1) - W_2^{0}\, x(5)$$

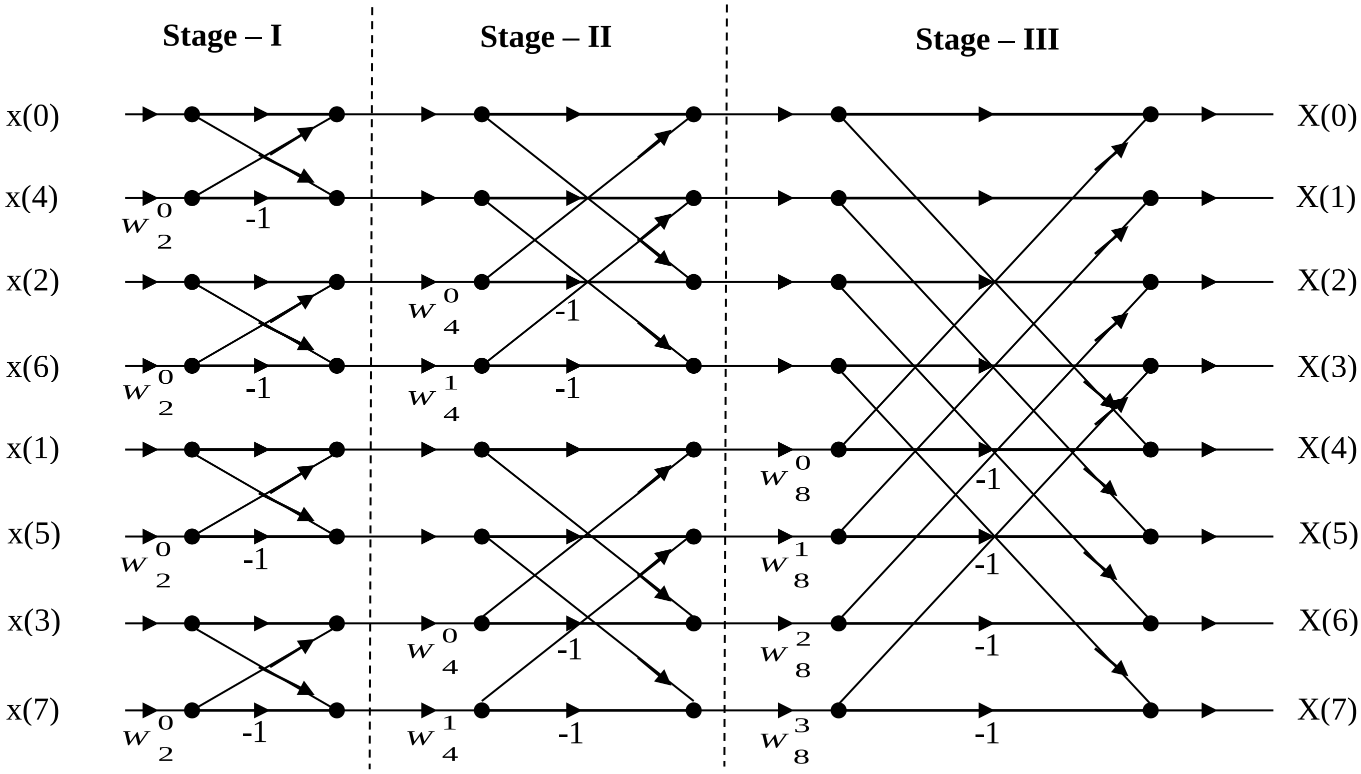
$$\text{Equation 6} \Rightarrow I_6(k) = \sum_{n=0..}^{\frac{N}{4}-1} x(4n+3)\, W_{N/4}^{nk} = \sum_{n=0}^{1} x(4n+3)\, W_2^{nk} = x(3) + W_2^{k}\, x(7)$$

$$\Rightarrow I_6(0) = x(3) + W_2^{0}\, x(7) \text{ and } I_6(1) = x(3) + W_2^{1}\, x(7) = x(3) - W_2^{0}\, x(7)$$

Signal flow graph of step-III or stage-I (Butterfly Structure)



Signal flow graph of 8-Point DIT radix-2 FFT (Butterfly Structure)



	Stage-I	Stage-II	Stage-III	Total
Number of Complex Adders	8	8	8	24
Number of Complex Multiplications	4	4	4	16
Number of Butterflies	4	2	1	7
Different Twiddle Factors	W_2^0	W_4^0, W_4^1	W_8^0, W_8^1, W_8^2, W_8^3	7

Ex: Compute 8-Point DFT of a sequence x(n)={1,1,1,1,0,0,0,0} using DIT radix-2 FFT algorithm.

Compute 8 samples of X(k) by assuming $I_1(k)$ and $I_2(k)$ are outputs of stage-I and stage-II

Output of stage-I:

$$I_1(0) = x(0) + W_2^0\, x(4) = 1 + 1\times 0 = 1 + 0 = 1$$

$$I_1(1) = x(0) - W_2^0\, x(4) = 1 - 1\times 0 = 1 - 0 = 1$$

$$I_1(2) = x(2) + W_2^0\, x(6) = 1 + 1\times 0 = 1 + 0 = 1$$

$$I_1(3) = x(2) - W_2^0\, x(6) = 1 - 1\times 0 = 1 - 0 = 1$$

$$I_1(4) = x(1) + W_2^0\, x(5) = 1 + 1\times 0 = 1 + 0 = 1$$

$$I_1(5) = x(1) - W_2^0\, x(5) = 1 - 1\times 0 = 1 - 0 = 1$$

$$I_1(6) = x(3) + W_2^0\, x(7) = 1 + 1\times 0 = 1 + 0 = 1$$

$$I_1(7) = x(3) - W_2^0\, x(7) = 1 - 1\times 0 = 1 - 0 = 1$$

$I_1(k) = \{1, 1, 1, 1, 1, 1, 1, 1\}$

Output of stage-II:

$$I_2(0) = I_1(0) + W_4^0\, I_1(2) = 1 + 1\times 1 = 1 + 1 = 2$$

$$I_2(1) = I_1(1) + W_4^1\, I_1(3) = 1 + (-j)\times 1 = 1 - j$$

$$I_2(2) = I_1(0) - W_4^0\, I_1(2) = 1 - 1\times 1 = 1 - 1 = 0$$

$$I_2(3) = I_1(1) - W_4^1\, I_1(3) = 1 - (-j)\times 1 = 1 + j$$

$$I_2(4) = I_1(4) + W_4^0\, I_1(6) = 1 + 1\times 1 = 1 + 1 = 2$$

$$I_2(5) = I_1(5) + W_4^1\, I_1(7) = 1 + (-j)\times 1 = 1 - j$$

$$I_2(6) = I_1(4) - W_4^0\, I_1(6) = 1 - 1\times 1 = 1 - 1 = 0$$

$$I_2(7) = I_1(5) - W_4^1\, I_1(7) = 1 - (-j)\times 1 = 1 + j$$

Output of stage-III:

$$X(0) = I_2(0) + W_8^0\, I_2(4) = 2 + 1\times 2 = 2 + 2 = 4$$

$$X(1) = I_2(1) + W_8^1\, I_2(5) = 1 - j + \left(\frac{1}{\sqrt{2}} - j\frac{1}{\sqrt{2}}\right)\times(1 - j) = 1 - j(1 + \sqrt{2})$$

$$X(2) = I_2(2) + W_8^2\, I_2(6) = 0 + (-j)\times 0 = 0$$

$$X(3) = I_2(3) + W_8^3\, I_2(7) = 1 + j + \left(-\frac{1}{\sqrt{2}} - j\frac{1}{\sqrt{2}}\right) \times (1 + j) = 1 + j(1 - \sqrt{2})$$

$$X(4) = I_2(0) - W_8^0\, I_2(4) = 2 - 1 \times 2 = 2 - 2 = 0$$

$$X(5) = I_2(1) - W_8^1\, I_2(5) = 1 - j - \left(\frac{1}{\sqrt{2}} - j\frac{1}{\sqrt{2}}\right) \times (1 - j) = 1 - j(1 - \sqrt{2})$$

$$X(6) = I_2(2) - W_8^2\, I_2(6) = 0 - (-j) \times 0 = 0$$

$$X(7) = I_2(3) - W_8^3\, I_2(7) = 1 + j - \left(-\frac{1}{\sqrt{2}} - j\frac{1}{\sqrt{2}}\right) \times (1 + j) = 1 + j(1 + \sqrt{2})$$

Note:

If x(n) is real, then X(N-k)=X*(k) ⇒ X(8-k)=X*(k)

k = 1 ⇒ X(7) = X*(1)

k = 2 ⇒ X(6) = X*(2)

k = 3 ⇒ X(5) = X*(3)

k = 4 ⇒ X(4) = X*(4)

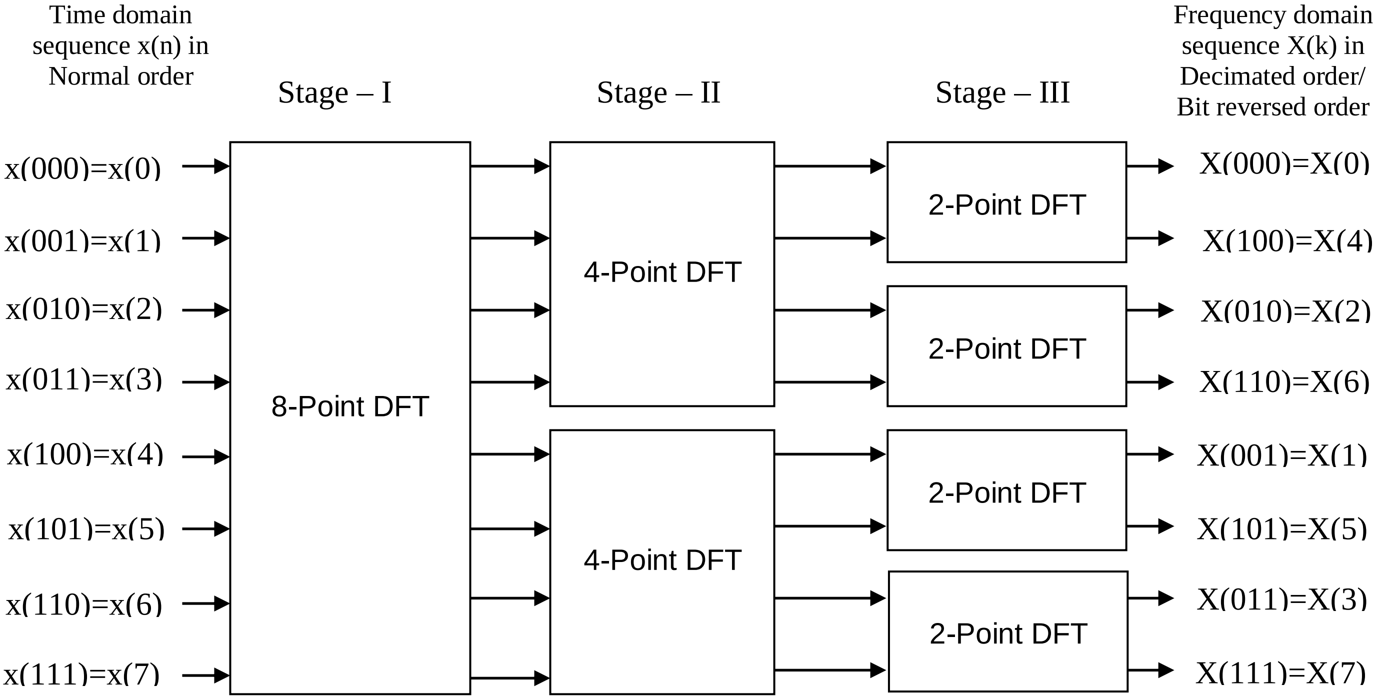
8-Point DFT of x(n) = {1, 1, 1, 1, 0, 0, 0, 0} is

$$X(k) = \{4, 1 - j(1+\sqrt{2}), 0, 1 + j(1-\sqrt{2}), 0, 1 - j(1-\sqrt{2}), 0, 1 + j(1+\sqrt{2})\}$$

(E)Decimation in Frequency (DIF) radix–2 FFT Algorithm:

In DIF radix-2 FFT, the frequency domain sequence X(k) is decimated into possible number of 2-Point sequences and time domain sequence x(n) is in normal order. In this algorithm N-point time domain sequence is converted into two N/2-Point sequences, then each N/2-Point sequence is converted into two N/4-Point sequences. This process will continue until we get the 2-point DFT.

Ex: Three stage computational structure for DIF radix-2 FFT as shown below



Step-I:

From basic definition of N-point DFT of a sequence x(n)

$$X(k) = \sum_{n=0}^{N-1} x(n)\, W_N^{nk}$$

Split the range into two parts 0 to N/2 - 1and N/2 to N - 1.

$$= \sum_{n=0}^{N/2-1} x(n)\, W_N^{nk} + \sum_{n=N/2}^{N-1} x(n)\, W_N^{nk}$$

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

$$= \sum_{n=0}^{N/2-1} x(n)\, W_N^{nk} + \sum_{m=0}^{N/2-1} x(N/2+n)\, W_N^{Nk/2} W_N^{nk}$$

$$= \sum_{n=0}^{N/2-1} x(n)\, W_N^{nk} + \sum_{m=0}^{N/2-1} x(N/2+n)\, W_2^{k}\; W_N^{nk}$$

$$X(k) = \sum_{n=0}^{N/2-1} [\, x(n) + W_2^{k}\; x(N/2+n)\,]\, W_N^{nk}$$

Replace k with 2k for even locations

$$X(2k) = \sum_{n=0}^{N/2-1} [\, x(n) + W_2^{2k}\; x(N/2+n)\,]\, W_N^{n2k}$$

$$= \sum_{n=0}^{N/2-1} [\, x(n) + x(N/2+n)\,]\, W_{N/2}^{nk}$$

$$X(2k) = \sum_{n=0}^{N/2-1} g_1(n)\, W_{N/2}^{nk} \quad \text{-------------------------------------}\rightarrow \quad (1)$$

$$= \frac{N}{2} - \text{point DFT}\,[\, g_1(n)\,]$$

Where, $g_1(n) = x(n) + x(N/2 + n)$

Replace k by 2k +1 for odd location

$$X(2k+1) = \sum_{n=0}^{N/2-1} [\, x(n) + W_2^{2k+1}\; x(N/2+n)\,]\, W_N^{n(2k+1)}$$

$$= \sum_{n=0}^{N/2-1} [\, x(n) + W_2^{2k}\; W_2^{1}\; x(N/2+n)\,]\, W_N^{2nk}\, W_N^{n}$$

$$= \sum_{n=0}^{N/2-1} [\, x(n) - x(N/2+n)\,]\; W_N^{n}\;\; W_{N/2}^{nk}$$

$$X(2k+1) = \sum_{n=0}^{N/2-1} g_2(n)\, W_{N/2}^{nk} \quad \text{-------------------------------------}\rightarrow \quad (2)$$

$$= \frac{N}{2} - \text{point DFT}[\, g_2(n)\,]$$

Where, $g_2(n) = [\, x(n) - x(N/2 + n)\,]\, W_N^n$

Ex: For N =8, we have to compute 8 samples from $g_1(n)$ and $g_2(n)$.

$g_1(n) = x(n) + x(4 + n)$ $\Rightarrow g_1(0) = x(0) + x(4)$

$\Rightarrow g_1(1) = x(1) + x(5)$

$\Rightarrow g_1(2) = x(2) + x(6)$

$\Rightarrow g_1(3) = x(1) + x(7)$

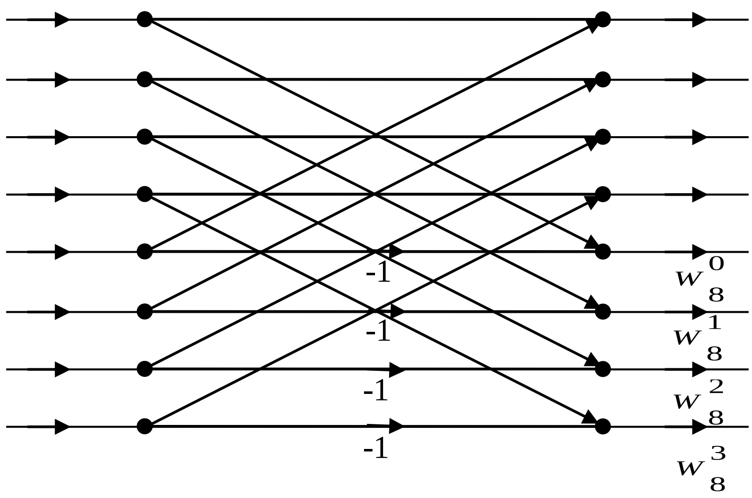
$g_2(n) = [\, x(n) - x(4 + n)\,]\, W_8^n \Rightarrow g_2(0) = [\, x(0) - x(4)\,]\, W_8^0$

$\Rightarrow g_2(1) = [\, x(1) - x(5)\,]\, W_8^1$

$\Rightarrow g_2(2) = [\, x(2) - x(6)\,]\, W_8^2$

$\Rightarrow g_2(3) = [\, x(3) - x(7)\,]\, W_8^3$

Signal flow graph of step-I or stage-I (Butterfly Structure)



Step-II:

From equation 1

$$X(2k) = \sum_{n=0}^{N/2-1} g_1(n)\, W_{N/2}^{nk}$$

Split the summation into two parts 0 to N/4 - 1 and N/4 to N - 1.

$$= \sum_{n=0}^{N/4-1} g_1(n)\, W_{N/2}^{nk} + \sum_{n=N/4}^{N/2-1} g_1(n)\, W_{N/2}^{nk}$$

$$= \sum_{n=0}^{N/4-1} g_1(n)\, W_{N/2}^{nk} + \sum_{n=0}^{N/4-1} g_1(N/4+n)\, W_{N/2}^{(N/4+n)k}$$

$$= \sum_{n=0}^{N/4-1} g_1(n)\, W_{N/2}^{nk} + \sum_{m=0}^{N/4-1} g_1(N/4+n)\, W_{N/2}^{Nk/4}\; W_{N/2}^{nk}$$

$$= \sum_{n=0}^{N/4-1} g_1(n)\, W_{N/2}^{nk} + \sum_{m=0}^{N/4-1} g_1(N/4+n)\, W_{2}^{k}\; W_{N/2}^{nk}$$

$$X(2k) = \sum_{n=0}^{N/4-1} [\, g_1(n) + W_{2}^{k}\; g_1(N/4+n)\,]\, W_{N/2}^{nk}$$

Replace k with 2k for even location

$$X(2.2k) = \sum_{n=0}^{N/4-1} [\, g_1(n) + W_{2}^{2k}\; g_1(N/4+n)\,]\, W_{N/2}^{n.2k}$$

$$X(4k) = \sum_{n=0}^{N/4-1} [\, g_1(n) + g_1(N/4+n)\,]\, W_{N/4}^{nk}$$

$$= \sum_{n=0}^{N/4-1} g_3(n)\; W_{N/4}^{nk} \quad \text{------------------------------------}\rightarrow (3)$$

$$X(4k) = \frac{N}{4} - \text{point DFT of } g_3(n)$$

Where. $g_3(n) = g_1(n) + g_1(N/4 + n)$

Replace k with 2k+1 for odd location

$$X(2(2k+1)) = \sum_{n=0}^{N/4-1} [\, g_1(n) + W_2^{2k+1}\, g_1(N/4+n)\,]\, W_{N/2}^{n(2k+1)}$$

$$X(4k+2) = \sum_{n=0}^{N/4-1} [\, g_1(n) + W_2^{2k} W_2^{1} g_1(N/4+n)\,]\, W_{N/2}^{2nk} W_{N/2}^{n}$$

$$= \sum_{n=0}^{N/4-1} [\, g_1(n) - g_1(N/4+n)\,]\, W_{N/2}^{n} W_{N/4}^{nk}$$

$$= \sum_{n=0}^{N/4-1} g_4(n)\; W_{N/4}^{nk} \text{-------------------------} \rightarrow (4)$$

$$X(4k) = \frac{N}{4} - \text{point DFT of } g_4(n)$$

Where, $g_4(n) = [\, g_1(n) - g_1(N/4 + n)\,]\, W_{N/2}^{\,n}$

From equation 2

$$X(2k+1) = \sum_{n=0}^{N/2-1} g_2(n)\, W_{N/2}^{nk}$$

Split summation into two parts 0 to N/4 - 1 and N/4 to N - 1.

$$= \sum_{n=0}^{N/4-1} g_2(n)\, W_{N/2}^{nk} + \sum_{n=N/4}^{N/2-1} g_2(n)\, W_{N/2}^{nk}$$

$$= \sum_{n=0}^{N/4-1} g_2(n)\, W_{N/2}^{nk} + \sum_{m=0}^{N/4-1} g_2(N/4+n)\, W_{N/2}^{(N/4+n)k}$$

$$= \sum_{n=0}^{N/4-1} g_2(n)\, W_{N/2}^{nk} + \sum_{m=0}^{N/4-1} g_2(N/4+n)\, W_{N/2}^{Nk/4}\; W_{N/2}^{nk}$$

$$= \sum_{n=0}^{N/4-1} g_2(n)\, W_{N/2}^{nk} + \sum_{n=0}^{N/4-1} g_2(N/4+n)\, W_2^{k}\; W_{N/2}^{nk}$$

$$X(2k+1) = \sum_{n=0}^{N/4-1} [\, g_2(n) + W_2^{k}\; g_2(N/4+n)\,]\, W_{N/2}^{nk}$$

Replace k with 2k for even location

$$X(2.2k+1) = \sum_{n=0}^{N/4-1} [\, g_2(n) + W_2^{2k}\; g_2(N/4+n)\,]\, W_{N/2}^{n.2k}$$

$$X(4k+1) = \sum_{n=0}^{N/4-1} [\, g_2(n) + g_2(N/4+n)\,]\, W_{N/4}^{nk}$$

$$= \sum_{n=0}^{N/4-1} g_5(n)\; W_{N/4}^{nk} \text{-----------------------------} \rightarrow (5)$$

$$X(4k+1) = \frac{N}{4} - \text{point DFT of } g_5(n)$$

Where, $g_5(n) = g_2(n) + g_2(N/4 + n)$

Replace k with 2k+1 for odd locations

$$X(2(2k+1)+1) = \sum_{n=0}^{N/4-1} [\, g_2(n) + W_2^{2k+1}\, g_2(N/4+n)\,]\, W_{N/2}^{n(2k+1)}$$

$$X(4k+3) = \sum_{n=0}^{N/4-1} [\, g_2(n) + W_2^{2k} W_2^{1} g_2(N/4+n)\,]\, W_{N/2}^{2nk} W_{N/2}^{n}$$

$$= \sum_{n=0}^{N/4-1} [\, g_2(n) - g_2(N/4+n)\,]\, W_{N/2}^{n} W_{N/4}^{nk}$$

$$= \sum_{n=0}^{N/4-1} g_5(n)\; W_{N/4}^{nk} \longrightarrow (6)$$

$$X(4k+3) = \frac{N}{4} - \text{point DFT of } g_6(n)$$

Where, $g_6(n) = [\, g_2(n) - g_2(N/4 + n)\,]\, W_{N/2}^{\ n}$

Ex: For N= 8, we have to compute 8 samples from $g_3(n)$, $g_4(n)$, $g_5(n)$ and $g_6(n)$

$g_3(n) = g_1(n) + g_1(2 + n)$ $\Rightarrow g_3(0) = g_1(0) + g_1(2)$

$\Rightarrow g_3(1) = g_1(1) + g_1(3)$

$g_4(n) = [g_1(n) - g_1(2 + n)]W_4^n$ $\Rightarrow g_4(0) = [\, g_1(0) - g_1(2)\,]W_4^0$

$\Rightarrow g_4(1) = [\, g_1(1) - g_1(3)\,]W_4^1$

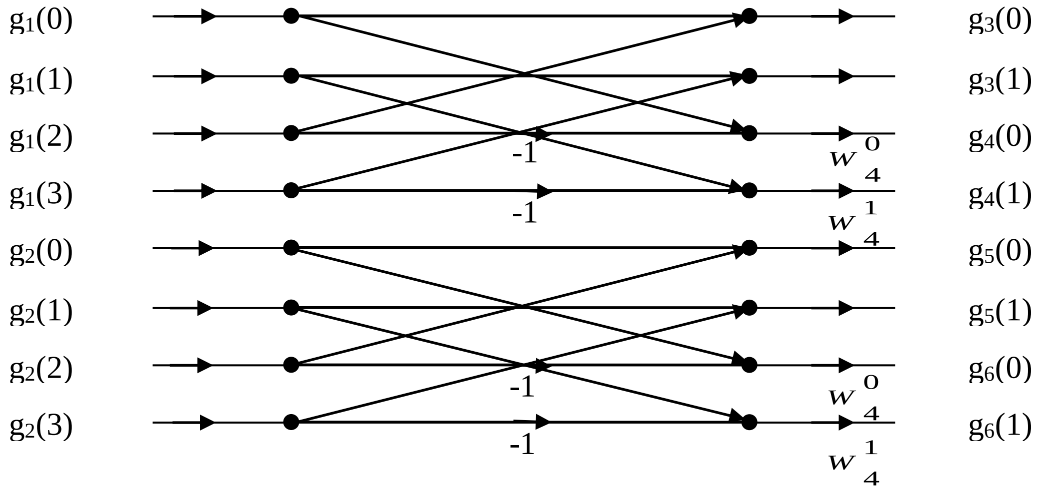
$g_4(n) = g_1(n) + g_1(2 + n)$ $\Rightarrow g_5(0) = g_2(0) + g_2(2)$

$\Rightarrow g_5(1) = g_2(1) + g_2(3)$

$g_5(n) = [g_1(n) - g_1(2 + n)]W_4^n$ $\Rightarrow g_6(0) = [\, g_2(0) - g_2(2)\,]W_4^0$

$\Rightarrow g_6(1) = [\, g_2(1) - g_2(3)\,]W_4^1$

Signal flow graph of step-II or stage-II (Butterfly Structure)



Step-III:

X(4k), X(4k+2), X(4k+1) and X(4k+3) are N/4-Point DFTs, if N=8, then X(4k), X(4k+2), X(4k+1) and X(4k+3) are 2-Point DFTs. Hence, in step-III no need to go for decimation process, because we have reached the 2-Point DFT i.e radix-2 for N=8.

From equation 3

$$X(4k) = \sum_{n=0}^{1} g_3(n)\ W_2^{nk} = g_3(0)\, W_2^{0k} + g_3(1)\, W_2^{1k} = g_3(0) + W_2^{k} g_3(1)$$

$\Rightarrow X(0) = g_3(0) + W_2^0\ g_3(2) = g_3(0) + g_3(2)$

$\Rightarrow X(4) = g_3(0) + W_2^1\ g_3(2) = [\ g_3(0) - g_3(2)\ ]\ W_2^0$

From equation 4

$$X(4k+2) = \sum_{n=0}^{1} g_4(n)\ W_2^{nk} = g_4(0)\, W_2^{0k} + g_4(1)\, W_2^{1k} = g_4(0) + W_2^{k} g_4(1)$$

$\Rightarrow X(2) = g_4(0) + W_2^0\ g_4(2) = g_4(0) + g_4(2)$

$\Rightarrow X(6) = g_4(0) + W_2^1\ g_4(2) = [\ g_4(0) - g_4(2)\ ]\ W_2^0$

From equation 5

$$X(4k+1) = \sum_{n=0}^{1} g_5(n)\ W_2^{nk} = g_5(0)\, W_2^{0k} + g_5(1)\, W_2^{1k} = g_5(0) + W_2^{k} g_5(1)$$

$\Rightarrow X(1) = g_5(0) + W_2^0\ g_5(2) = g_5(0) + g_5(2)$

$\Rightarrow X(5) = g_5(0) + W_2^1\ g_5(2) = [\ g_5(0) - g_5(2)\ ]\ W_2^0$

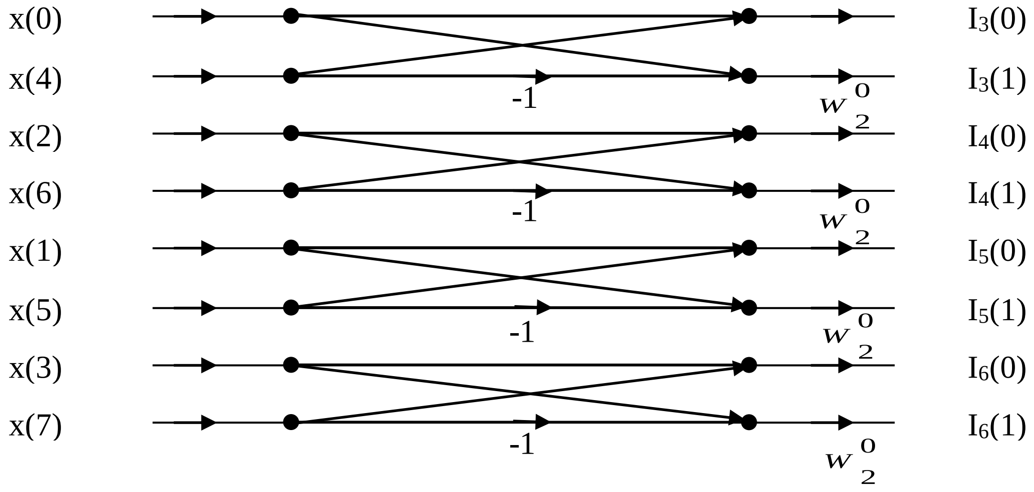
From equation 6

$$X(4k+3) = \sum_{n=0}^{1} g_6(n)\ W_2^{nk} = g_6(0)\, W_2^{0k} + g_6(1)\, W_2^{1k} = g_6(0) + W_2^{k} g_6(1)$$

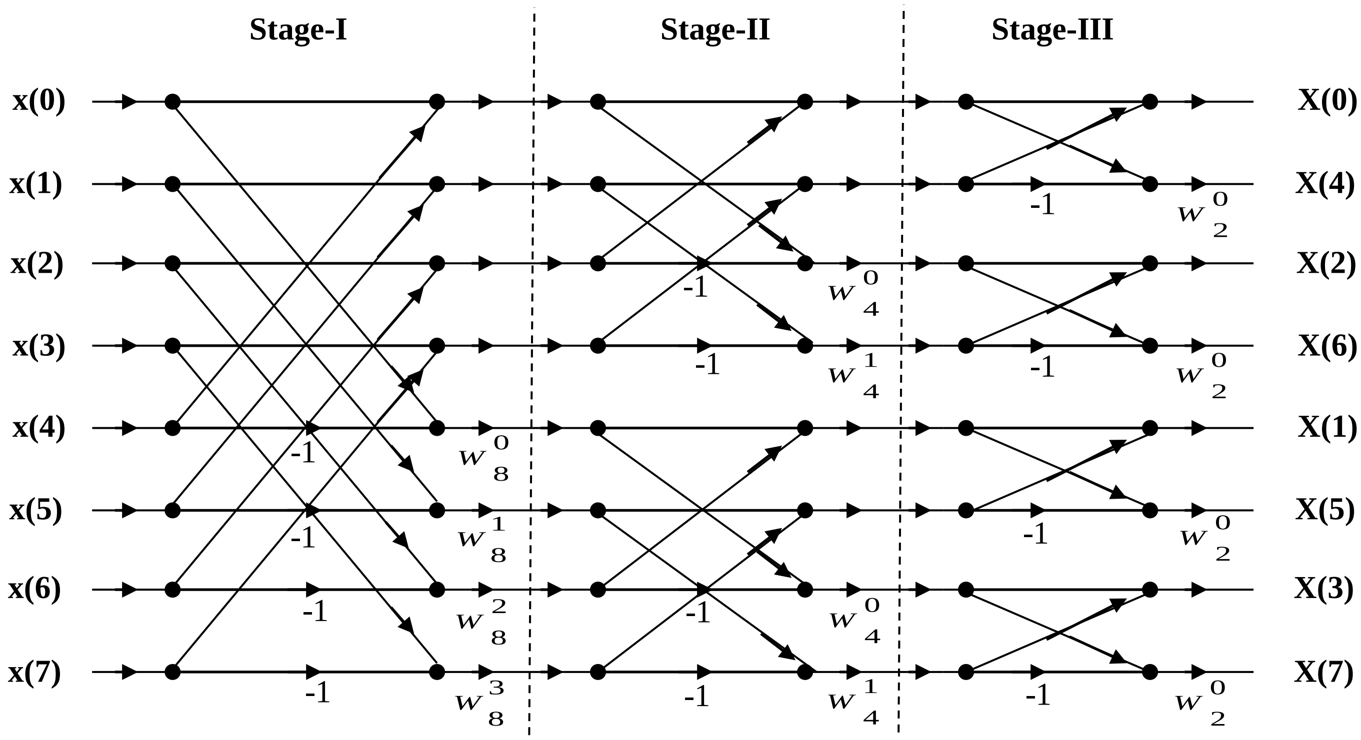
$\Rightarrow X(3) = g_6(0) + W_2^0\ g_6(2) = g_6(0) + g_6(2)$

$\Rightarrow X(7) = g_6(0) + W_2^1\ g_6(2) = [\ g_6(0) - g_6(2)\ ]\ W_2^0$

Signal flow graph of step-III or stage-III (Butterfly Structure)



Signal flow graph of 8-Point DIF radix-2 FFT (Butterfly Structure)



	Stage-I	Stage-II	Stage-III	Total
Number of Complex Adders	8	8	8	24
Number of Complex Multiplications	4	4	4	16
Number of Butterflies	1	2	4	7
Different Twiddle Factors	W_8^0, W_8^1, W_8^2, W_8^3	W_4^0, W_4^1	W_2^0	7

Ex: Compute 8-Point DFT of a sequence x(n)={1,1,1,1,0,0,0,0} using DIF radix-2 FFT algorithm. Compute 8 samples of X(k) by assuming $I_1(k)$ and $I_2(k)$ are outputs of stage-I and stage-II

<u>Output of stage-I:</u>

$$I_1(0) = x(0) + x(4) = 1 + 0 = 1,\ I_1(1) = x(1) + x(5) = 1 + 0 = 1$$

$$I_1(2) = x(2) + x(6) = 1 + 0 = 1,\ I_1(3) = x(3) + x(7) = 1 + 0 = 1$$

$$I_1(4) = (x(0) - x(4))W_8^0 = (1-0)\times 1 = 1 - 0 = 1$$

$$I_1(5) = (x(1) - x(5))W_8^1 = (1-0)\times\left(\frac{1}{\sqrt{2}} - j\frac{1}{\sqrt{2}}\right) = \frac{1}{\sqrt{2}} - j\frac{1}{\sqrt{2}}$$

$$I_1(6) = (x(2) - x(6))W_8^2 = (1-0)\times(-j) = -j$$

$$I_1(7) = (x(3) - x(7))W_8^3 = (1-0)\times\left(-\frac{1}{\sqrt{2}} - j\frac{1}{\sqrt{2}}\right) = -\frac{1}{\sqrt{2}} - j\frac{1}{\sqrt{2}}$$

$$I_1(k) = \left\{1,1,1,1,\frac{1}{\sqrt{2}} - j\frac{1}{\sqrt{2}}, -j, -\frac{1}{\sqrt{2}} - j\frac{1}{\sqrt{2}}\right\}$$

Output of stage-II:

$$I_2(0)=I_1(0)+I_1(2)=1+1=2$$

$$I_2(1)=I_1(1)+I_1(3)=1+1=2$$

$$I_2(2)=\left(I_1(0)-I_1(2)\right)W_4^0=\left(1-1\right)1=0$$

$$I_2(3)=\left(I_1(1)-I_1(3)\right)W_4^1=\left(1-1\right)(-j)=0$$

$$I_2(4)=I_1(4)+I_1(6)=1+(-j)=1-j$$

$$I_2(5)=I_1(5)+I_2(7)=\frac{1}{\sqrt{2}}-j\frac{1}{\sqrt{2}}+\left(-\frac{1}{\sqrt{2}}-j\frac{1}{\sqrt{2}}\right)=-\sqrt{2}j$$

$$I_2(6)=\left(I_1(4)-I_1(6)\right)W_4^0=1-(-j)=1+j$$

$$I_2(7)=\left(I_1(5)-I_1(7)\right)W_4^1=\left(\frac{1}{\sqrt{2}}-j\frac{1}{\sqrt{2}}-\left(-\frac{1}{\sqrt{2}}-j\frac{1}{\sqrt{2}}\right)\right)=\sqrt{2}(-j)=-\sqrt{2}j$$

$$I_2(k)=\left\{2,2,0,0,1-j,-\sqrt{2}j,1+j,-\sqrt{2}j\right\}$$

Output of stage-III:

$$X(0)=I_2(0)+I_2(1)=2+2=4$$

$$X(4)=\left(I_2(0)-I_2(1)\right)W_2^0=\left(2-2\right)1=0$$

$$X(2)=I_2(2)+I_2(3)=0+0=0$$

$$X(6)=\left(I_2(2)-I_2(3)\right)W_2^0=\left(0-0\right)1=0$$

$$X(1)=I_2(4)+I_2(5)=1-j+\left(-\sqrt{2}j\right)=1-\left(1+\sqrt{2}\right)j$$

$$X(5)=\left(I_2(4)-I_2(5)\right)W_2^0=1-j-\left(-\sqrt{2}j\right)=1-\left(1-\sqrt{2}\right)j$$

$$X(3)=I_2(6)+I_2(7)=1+j+\left(-\sqrt{2}j\right)=1+\left(1-\sqrt{2}\right)j$$

$$X(7)=\left(I_2(6)-I_2(7)\right)W_2^0=1+j-\left(-\sqrt{2}j\right)=1+\left(1+\sqrt{2}\right)j$$

8-Point DFT of x(n) = {1, 1, 1, 1, 0, 0, 0, 0} is

$$X(k)=\{4,1-j(1+\sqrt{2}),0,1+j(1-\sqrt{2}),0,1-j(1-\sqrt{2}),0,1+j(1+\sqrt{2})\}$$

(F)Comparison b/n DIT radix–2 FFT and DIF radix–2 FFT:

DIT radix 2 FFT Method	DIF radix 2 FFT Method
The time domain sequence x(n) is decimated and it is in bit reversed order.	The frequency domain sequence X(k) is decimated and it is in bit reversed order.
The frequency domain sequence X(k) is in normal order.	The time domain sequence x(n) is in normal order.
In each stage of computation, the phase factors are multiplied before add and subtract operations.	In each stage of computation, the phase factors are multiplied after subtract operations.
For both the algorithms, the value of N should be expressed as $N = r^m$. where m: No. of stages and r: radix number	
Both the algorithms require same number complex additions and complex multiplications.	
For both the algorithms, required number of Complex additions = Nm = N $\log_2$ N. Complex multiplications = (N/2) m = (N/2) $\log_2$ N.	
Both the algorithms are used to compute DFT as well as IDFT.	

(G)Inverse FFT:

We know that the Fast Fourier Transform (FFT) is used to compute N-Point DFT[x(n)] by using DIT radix-2 FFT and DIF radix-2 FFT algorithms. Now, we can apply above algorithms to compute N-Point IDFT[X(k)] is called Inverse FFT.

From basic definition of N-Point IDFT of a sequence

$$x(n) = \frac{1}{N}\sum_{k=0}^{N-1} X(k)\, W_N^{-nk}$$

$$= \frac{1}{N}\left(\sum_{k=0}^{N-1}\left(X(k)\, W_N^{-nk}\right)^*\right)^*$$

$$= \frac{1}{N}\left(\sum_{k=0}^{N-1} X^*(k)\, W_N^{nk}\right)^*$$

$$= \frac{1}{N}\left(\ DFT\,(X^*(k))\ \right)^*$$

$$= \frac{y^*(n)}{N}$$

Where, y(n) = DFT[X*(k)]

Procedure to compute N-Point IDFT of a sequence X(k)

Step-I:

Take complex conjugate of given X(k), i.e X*(k).

Step-II:

Compute N-point DFT of X*(k) using DIT radix-2 FFT or DIF radix-2 FFT method,

i.e DFT[X*(k)] = y(n)

Step-III:

Now, take complex conjugate of y(n), i.e y*(n).

Step-IV:

Finally obtain the sequence x(n) by using x(n) = y*(n) / N.

Example: Compute the 4-point IDFT of a sequence X(k)={10,-2+2j,-2,-2-2j}by using
(a) DIT radix-2 FFT method (b) DIF radix-2 FFT method

(a) DIT radix-2 FFT method

Step-I:

Take the complex conjugate of given X(k), i.e X*(k).

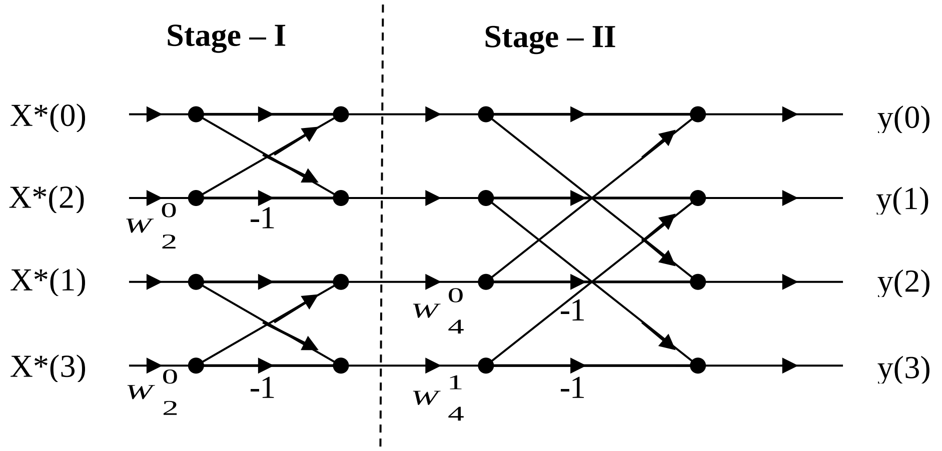
X*(k)={10, -2-2j, -2, -2+2j}

Step-II:

Compute 4-point DFT of X*(k) using DIT radix-2 FFT or DIF radix-2 FFT method,

i.e DFT[X*(k)] = y(n)

(a)Signal flow graph of 4-Point DIT radix-2 FFT (Butterfly Structure)



Compute 4 samples of y(n) by assuming I(k) is output of stage-I

<u>**Output of stage-I:**</u>

$$I(0) = X^*(0) + W_2^0\ X^*(2) = 10 + 1\times(-2) = 10 - 2 = 8$$

$$I(1) = X^*(0) - W_2^0\ X^*(2) = 10 - 1\times(-2) = 10 + 2 = 12$$

$$I(2) = X^*(1) + W_2^0\ X^*(3) = -2 - 2j + 1\times(-2 + 2j) = -2 - 2j - 2 + 2j = -4$$

$$I(3) = X^*(1) - W_2^0\ X^*(3) = -2 - 2j - 1\times(-2 + 2j) = -2 - 2j + 2 - 2j = -4j$$

$$I(k) = \{8, 12, -4, -4j\}$$

Output of stage-II:

$$y(0)=I(0)+W_4^0\, I(2)=8+1\times(-4)=8-4=4$$

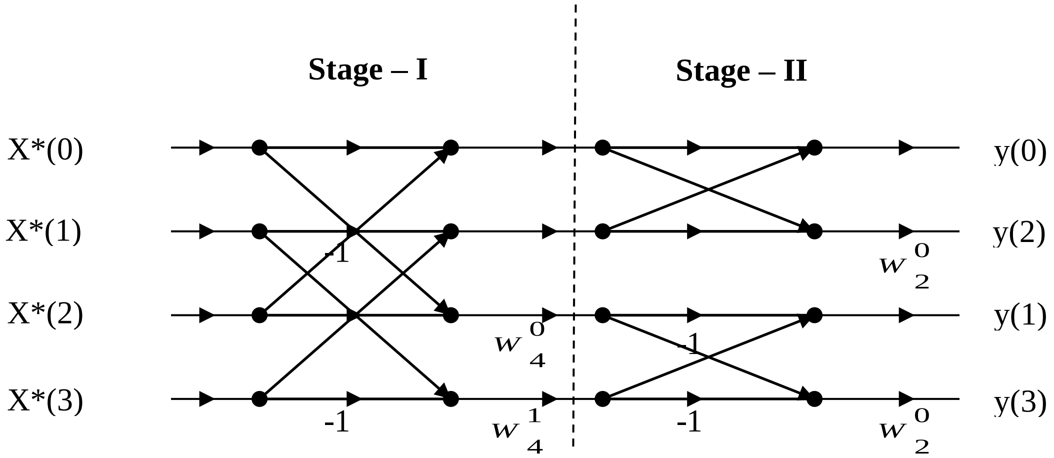
$$y(1)=I(1)+W_4^1\, I(3)=12+(-j)\times(-4j)=12-4=8$$

$$y(2)=I(0)-W_4^0\, I(2)=8-1\times(-4)=8+4=12$$

$$y(3)=I(1)-W_4^1\, I(3)=12-(-j)\times(-4j)=12+4=16$$

$$y(n)=\{4,8,12,16\}$$

(b)Signal flow graph of 4-Point DIF radix-2 FFT (Butterfly Structure)



Compute 4 samples of y(n) by assuming I(k) is output of stage-I

Output of stage-I:

$$I(0)=X*(0)+X*(2)=10-2=8,$$

$$I(1)=X*(1)+X*(3)=-2-2j+(-2+2j)=-4$$

$$I(2)=\left(X*(0)-X*(2)\right)W_4^0=\left(10-(-2)\right)\times 1=10+2=12$$

$$I(3)=\left(X*(1)-X*(3)\right)W_4^1=\left(-2-2j-(-2+2j)\right)\times(-j)=\left(-2-2j+2-2j\right)(-j)=-4$$

$$I(0)=\{8,-4,12,-4\}$$

Output of stage-II:

$$y(0) = I(0) + I(1) = 8 - 4 = 4$$

$$y(2) = (I(0) - I(1))W_2^0 = (8 - (-4)) \times 1 = 8 + 4 = 12$$

$$y(1) = I(2) + I(3) = 12 - 4 = 8$$

$$y(3) = (I(2) - I(3))W_2^0 = (12 - (-4)) \times 1 = 12 + 4 = 16$$

$$y(n) = \{4,8.12,16\}$$

Step-III:

Now, take complex conjugate of y(n), i.e y*(n).

$$y^*(n) = \{4,8,12,16\}$$

Step-IV:

Finally obtain the sequence x(n) by using x(n) = y*(n) / N.

$$x(n) = \frac{y^*(n)}{4} = \{1,2,3,4\}$$

Descriptive Questions:

1. Compute the 4-Point DFT of given sequences, hence obtain its magnitude and phase spectrum. (i)$x_1(n)=\{1,2,3,4\}$ (ii)$x_2(n)=\{1,2,3\}$.
2. Compute the 8-Point DFT of given sequence.
   (i)$x_1(n)=Cos(n\pi/2)$. $0\leq n\leq 7$. (ii)$x_2(n)=Cos(n\pi)$. $0\leq n\leq 7$.
3. Compute the 4-Point IDFT of sequences.
   (i)$X_1(k)=\{10,-2+2j,-2,-2-2j\}$ (ii)$X_2(k)=\{26,-2+2j,-2,-2-2j\}$.
4. Evaluate the sequence y(n), such the 4-Point DFT[y(n)]=Y(k)=$X^2(k)$ and 4-Point DFT[x(n)]=X(k). Given x(n)={1,2,3,4}.
5. Compute (i)2-Point DFT[x(n)] (ii)3-Point DFT[x(n)] (iii)4-Point DFT[x(n)].
   Given $x(n)=\delta(n-1)$.
6. Compute the linear convoluted sequence $x(n)=x_1(n)*x_2(n)$ using (i)Graphical method (ii)Tabular method. Given $x_1(n)=\{1,2,3,4\}$ and $x_2(n)=\{5,6,7,8,9\}$.
7. Evaluate the summation $\sum_{k=1}^{8} X(k)$, such that 4-Point DFT[x(n)]=X(k). Given x(n)={1,2,3,4}
8. Compute the circular convoluted sequence $x(n)=x_1(n)*x_2(n)$ using (i)Graphical method (ii)Tabular method (iii)Circular method (iv)Matrix method.
   Given $x_1(n)=\{1,2,3,4\}$ and $x_2(n)=\{5,6,7,8\}$.
9. Compute the linear convoluted sequence $x(n)=x_1(n)*x_2(n)$ using circular convolution.
   Given $x_1(n)=\{1,2,3,4\}$ and $x_2(n)=\{5,6,7,8,9\}$.
10. Compute the response of discrete LSI system having input x(n)={1,-1,-2} and impulse response h(n)={-2,1,0,1} using (i)Linear convolution-Tabular method
    (ii)Circular convolution-Matrix method.
11. Compute the circular convoluted sequence $x(n)=x_1(n)*x_2(n)$ using DFT-IDFT method.
    Given $x_1(n)=\{1,2,3,4\}$ and $x_2(n)=\{5,6,7,8\}$.
12. Compute the linear convoluted sequence $x(n)=x_1(n)*x_2(n)$ using DFT-IDFT method.
    Given $x_1(n)=\{1,2,3,4\}$ and $x_2(n)=\{5,6,7,8,9\}$.
13. Compute the response of discrete LSI system having input x(n)={1,2} and impulse response h(n)={3,4,5} using DFT-IDFT method.
14. Apply DIT radix-2 FFT algorithm and compute 8-Point DFT of sequences
    (i)x(n)={1,1,1,1,1,1,1,1} (ii)x(n)={1,0,1,0,1,0,1,0}
15. Apply DIF radix-2 FFT algorithm and compute 8-Point DFT of sequences
    (i)x(n)={1,-1,1,-1,1,-1,1,-1} (ii)x(n)={1,0,0,0,1,0,0,0}
16. Compute 8-Point IDFT of sequence X(k)={1,0,0,0,0,0,0,0} using
    (i)DIT radix-2 FFT algorithm (ii) DIF radix-2 FFT algorithm

Quiz Questions:

Q.No.	Question Description (Minimum 10 Questions)	Ans
1.	What is the magnitude of a phase factor W_{64}^{1}	1
2.	What is the duration of linear convoluted sequence $x(n) = x_1(n) * x_2(n)$, if N-2 is the duration of $x_1(n)$ and 2N is the duration of $x_2(n)$	3(N-1)
3.	A sequence $X(k) = 1, 0 \leq k \leq N-1$, and IDFT[X(k)] = x(n), then find x(0)	1
4.	A sequence $x(n) = 1, 0 \leq n \leq N-1$, and DFT[x(n)] = X(k), then find X(0)	N
5.	What is the percentage saving due to additions in the computation of 64 point radix–2 FFT (A)94.49 (B) 90.48 (C) 83.87 (D) 73.33	B
6.	The process of converting higher order point DFTs into lower order point DFTs called (A)FFT (B) IDFT (C) Decimation (D) Interpolation	C
7.	How many number of complex adders and complex multipliers required to compute 64-Point DFT of a sequence in direct DFT (A)4032,4096 (B) 992,1024 (C) 384,192 (D) 56,64	A
8.	If x(n) is real and 8-Point DFT[x(n)]=X(k), then which of the following are complex conjugate pairs (A)X(1)&X(7) (B) X(2)&X(6) (C) X(3)&X(5) (D) X(0)&X(4)	A,B,C
9.	The complex conjugate of phase factor W_{8}^{1} is (A) W_{8}^{-1} (B) W_{8}^{7} (C) W_{8}^{3} (D) W_{8}^{5}	A,B
10.	Linear and Circular convolution of x(n)={1}, y(n)={1} (A)Same (B) Different (C) {1} and {1,1} (D) {1,1} and {1}	A