Direct Computation of DFT :-

(a) Computational complexity of DFT :

By defn DFT is given by

$$X(K) = \sum_{n=0}^{N-1} x(n)\, W_N^{Kn} \quad , \quad k = 0, 1, \ldots N-1$$

where $x(n) \rightarrow$ i/p sequence which can be real or complex

$W_N \rightarrow$ is twiddle factor which is complex (Also known as phase factor)

$\therefore$ computation of $X(K)$ involves the multiplication and summation of complex numbers.

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

$\rightarrow$ $(N-1)$ complex addition for one value of K

$\rightarrow$ (N) complex multiplications for one value of K

$\rightarrow$ No of complex multiplications required for calculating $X(K)$ for $K = 0, 1, 2 \ldots N-1 \Rightarrow \underline{N \times N = N^2}$

$\rightarrow$ No of complex additions required for calculating $X(K)$ for $K = 0, 1, 2, \ldots N-1 \Rightarrow \underline{(N-1)N = N^2 - N}$

Ex: if we want to evaluate 1024 point DFT $N = 1024$

(i) complex multiplications $= N^2 = (1024)^2 = 1 \times 10^6$.

(ii) — " — Addition $= N(N-1) = 1024 \times 1024$

$= \underline{1 \times 10^6}$.

(b) Need for efficient computation of DFT :

Let us assume that processor executes one complex multiplication and one complex addition in 1 micro second.

If $N = 1024$ then time taken will be

$$t = (1024 \times 10^{-6} + 1024 \times 10^{-6}) = 2 \text{ seconds}$$

→ This 2 seconds is large time & hence there is a need for efficient computation of DFT

$$\boxed{\text{Time} = \left[(\text{complex multiplication}) \times (\text{Time for one multiplication})\right. \\ \left. + (\text{complex addition} \times \text{Time for one addition})\right]}$$

Properties of phase Factor (W_N) :-

"N" - pt DFT of sequence $x(n)$ is given as

$$X(k) = \sum_{n=0}^{N-1} x(n)\, W_N^{kn} \qquad k = 0, 1, 2 \cdots N-1$$

where W_N → twiddle factor (or) phase factor

& $W_N = e^{-j\frac{2\pi}{N}}$

→ This twiddle factor exhibits symmetry and periodicity properties.

(i) Periodicity property of W_N :

$$\boxed{W_N^{k+N} = W_N^{k}}$$

proof : wkt $W_N = e^{-j\frac{2\pi}{N}}$.

$$W_N^{k+N} = e^{-j\frac{2\pi}{N}(k+N)}$$

$$= e^{-j\frac{2\pi k}{N}} \cancel{e^{-j\frac{2\pi N}{N}}}^{1}$$

$$= e^{-j\frac{2\pi k}{N}}$$

$$= \left[e^{-j\frac{2\pi}{N}}\right]^{k} = W_N^k$$

ie W_N^k is periodic with period "N"

$$W_s^{k+s} = W_s^k.$$

(ii) <u>Symmetry Property</u> :-

$$\boxed{W_N^{k+N/2} = -W_N^k} \quad \& \quad \boxed{W_N^2 = W_{N/2}}$$

Proof : WKt $W_N = e^{-j\frac{2\pi}{N}}$

$$W_N^{k+N/2} = e^{-j\frac{2\pi}{N}[k+N/2]}$$

$$= e^{-j\frac{2\pi}{N}k} \underbrace{e^{-j\frac{2\pi}{\cancel{N}}\frac{\cancel{N}}{2}}}_{-1}$$

$$= e^{-j\frac{2\pi}{N}k}(-1)$$

$$= -e^{-j\frac{2\pi}{N}k}$$

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

$$W_N = e^{-j\frac{2\pi}{N}}$$

Replace N by N/2.

$$W_{N/2} = e^{-j\frac{2\pi}{N}\times 2}$$

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

* The Direct computation of DFT doesnot use these properties of W_N. Then FFT algorithms use these properties of W_N to reduce calculation of DFT

FFT Algorithms and their calculations & classification :-

* FFT = Fast Fourier transform
* efficient computation of the DFT (FFT algorithms)
* uses the properties of twiddle factor for efficient computation of DFT.
* FFT algorithms are based on two basic methods.

(1) Divide and conquer approach

→ N-point DFT is divided successively to 2-point DFT's to reduce calculations

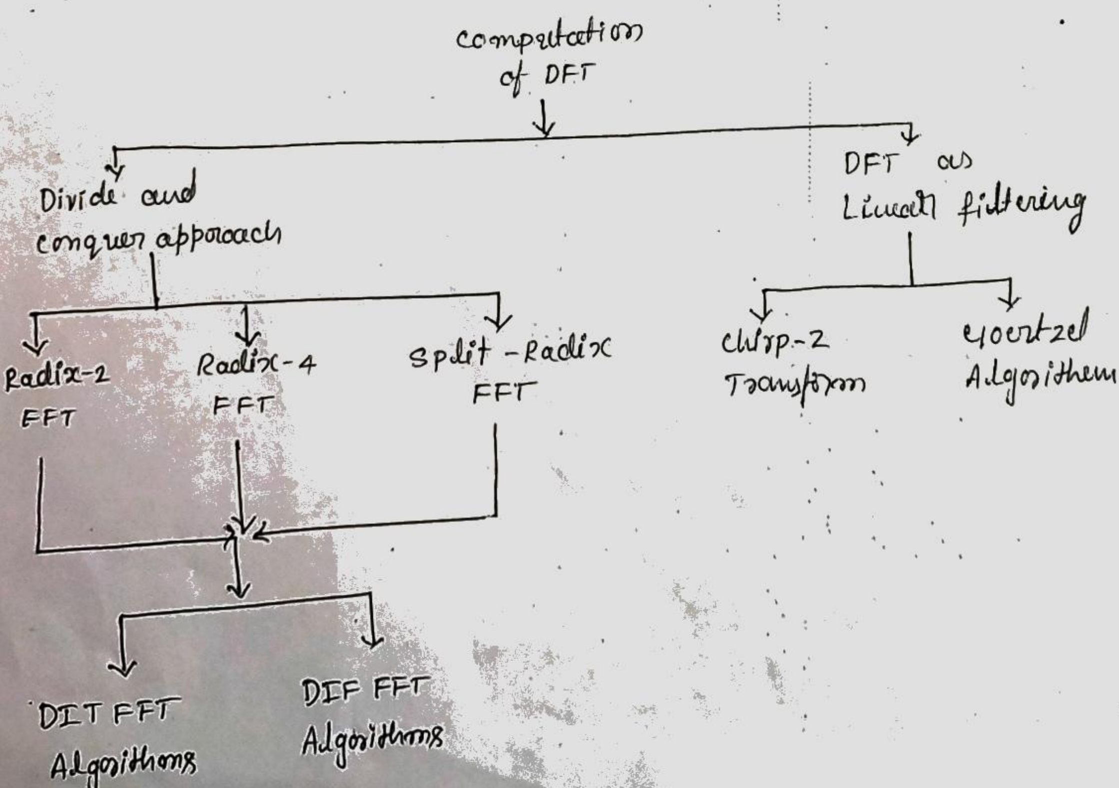
→ Radix-2, Radix-4 decimation in time, decimation in frequency are developed.

(2) Based on linear filtering

→ There are two algorithms

(i) Goertzel algorithm

(ii) Chirp-z transform algorithm.



* Advantages of FFT Algorithms :-

(1) Can be used to compute DFT as well as IDFT very efficiently

(2) computation complexity is greatly reduced compared to direct computation.

(3) As the length of DFT increases, then computation time reduces.

(4) Storage requirement of FFT is "2N" which is very small.

(5) Real time implementation is possible only because of FFT algorithm.

FFT Algorithms :-

* In the direct computation of N-point DFT, number of complex multiplications and additions required are N^2 and $N(N-1)$ respectively.

* The no of complex multiplications can be reduced if the periodicity and symmetric property of phase factor are utilized

(i) Periodicity property $W_N^{k+N} = W_N^k$

(ii) symmetry property $W_N^{k+N/2} = -W_N^k$

* There are different ways of computing FFTs one of the Most efficient method is the "Radix-2 FFT" algorithms.

Based on the divide-and-conquer approach.

→ In this method there are two ways of computing the FFT

(1) Decimation in time FFT Algorithm (DIT-FFT)

(2) Decimation in frequency FFT Algorithm (DIF-FFT)

* To apply the Radix-2 FFT Algorithm number of the sample points in the i/p sequence must be $N = 2^v$

where $v \rightarrow$ is an integer.

(1) Radix-2 DIT-FFT Algorithm :-

Consider a sequence $x(n)$ of length "N", $N = 2^v$

→ The i/p sequence is divided into 2 sequence $f_1(n)$ and $f_2(n)$ each of length $N/2$ where $f_1(n)$ and $f_2(n)$ consists of even and odd samples of $x(n)$ respectively.

$$\left.\begin{array}{l} f_1(n) = x_1(2n) \\ f_2(n) = x_2(2n+1) \end{array}\right\} \quad 0 \le n \le \frac{N}{2} - 1$$

By defn $\quad X(K) = DFT[x(n)] = \sum_{n=0}^{N-1} x(n) W_N^{Kn} \qquad 0 \le K \le N-1$

$$X(K) = \sum_{n(even)} x(n) W_N^{Kn} + \sum_{n(odd)} x(n) W_N^{Kn}$$

$$= \sum_{n=0}^{\frac{N}{2}-1} x(2n) W_N^{K2n} + \sum_{n=0}^{N/2-1} x(2n+1) W_N^{K(2n+1)}$$

$$= \sum_{n=0}^{N/2-1} f_1(n) W_{N/2}^{\frac{Kn2}{2}} + \sum_{n=0}^{N/2-1} f_2(n) \cdot W_N^{\frac{2Kn}{2}} W_N^{K}$$

$$= \sum_{n=0}^{N/2-1} f_1(n) \cdot W_{N/2}^{Kn} + W_N^{K} \sum_{n=0}^{N/2-1} f_2(n) \cdot W_{N/2}^{Kn}$$

$$X(K) = F_1(K) + W_N^K F_2(K) \qquad 0 \le K \le \frac{N}{2} - 1$$

$$X(K + N/2) = F_1(K + N/2) + W_N^{K+N/2} F_2(K + N/2) \qquad 0 \le K \le \frac{N}{2} - 1$$

* $F_1(K)$ & $F_2(K)$ are $N/2$ point DFT and they are periodic with period $N/2$

$$\therefore \quad F_1(K+N/2) = F_1(K)$$

$$F_2(K+N/2) = F_2(K)$$

$$W_N^{K+N/2} = -W_N^K$$

$$W_N^{K+N/2} = W_N^K \cdot W_N^{N/2}$$

$$= W_N^K \cdot e^{-j\frac{2\pi}{N}\times N/2}$$

$$= W_N^K \cdot (\cos\pi - j\sin\pi)$$

$$= W_N^K(-1) = -W_N^K$$

$$\boxed{\begin{aligned} X(K) &= F_1(K) + W_N^K F_2(K) \\ X(K+N/2) &= F_1(K) - W_N^K F_2(K) \end{aligned}} \qquad 0 \le K \le \frac{N}{2}-1$$

* Direct computation of $F_1(K)$ requires $(N/2)^2$ complex multiplication
* ———— 〃 ———— $F_2(K)$ ———— 〃 ————
* $\frac{N}{2}$ additional complex multiplications requires to compute $W_N^K F_2(K)$

$$\therefore \quad X(K) \text{ requires} \Rightarrow \left(\frac{N}{2}\right)^2 + \left(\frac{N}{2}\right)^2 + \frac{N}{2}$$

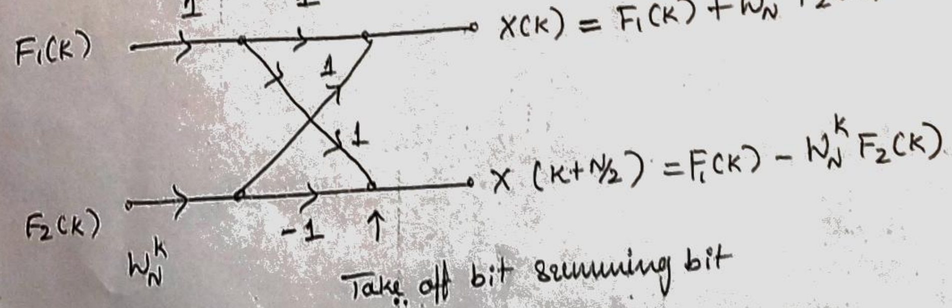
$$= 2(N/2)^2 + N/2 \quad \text{complex multiplication}$$

* The No of complex additions required to compute the N-point DFT is

$$\underbrace{\frac{N}{2}\left(\frac{N}{2}-1\right)}_{F_1(K)} + \underbrace{\frac{N}{2}\left(\frac{N}{2}-1\right)}_{F_2(K)} + N = \frac{N^2}{2}$$

$$=$$

* To compute $X(K)$ from $F_1(K)$ & $F_2(K)$ "Butterfly diagram" is used



Ex : for $N = 8$

$$\left.\begin{aligned} X(K) &= F_1(K) + W_8^K F_2(K) \\ X(K+4) &= F_2(K) - W_8^K F_2(K) \end{aligned}\right\} \quad 0 \le K \le 3$$

$k=0$; $\cancel{X}(0) = F_1(0) + W_8^0 F_2(0)$

$X(4) = F_1(0) - W_8^0 F_2(0)$

$K = \cancel{1}$, $X(1) = F_1(1) + W_8^1 F_2(1)$

$X(5) = F_1(1) - W_8^1 F_2(1)$

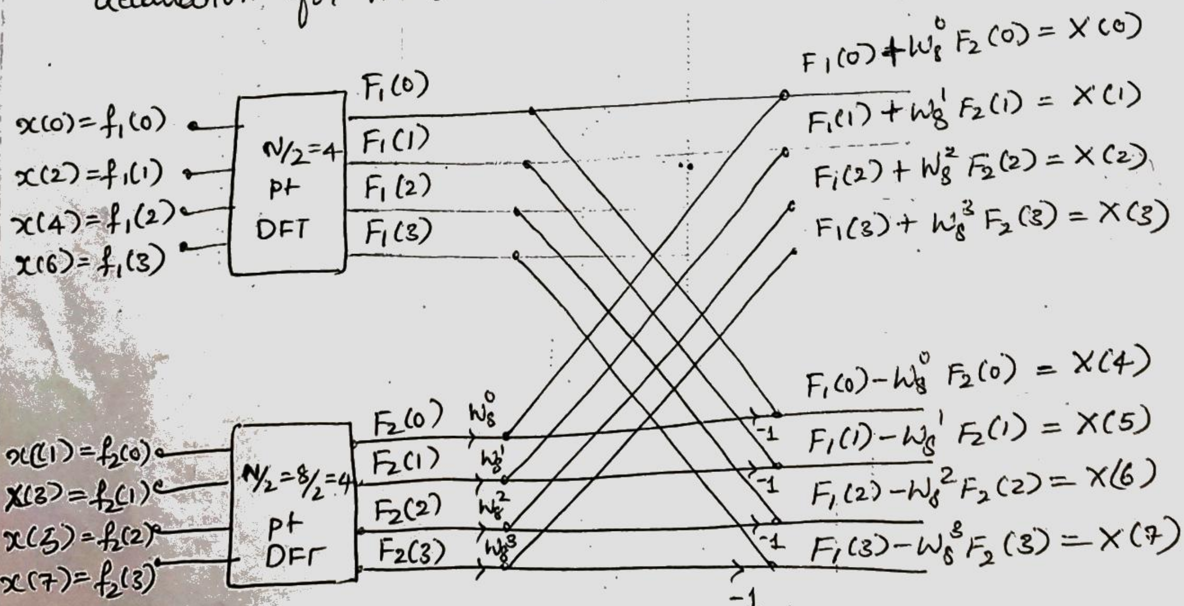
$K=2$, $X(2) = F_1(2) + W_8^2 F_2(2)$

$X(6) = F_1(2) - W_8^2 F_2(2)$

$k=3$, $X(3) = F_1(3) + W_8^3 F_2(3)$

$X(7) = F_1(3) - W_8^3 F_2(3)$

→ The signal flow graph / flow diagram at the end of first decimation for $N = 8$ is as shown below.



WKT $\quad F_1(K) = \sum_{n=0}^{N-1} f_1(n)\, W_{N/2}^{Kn} , \quad 0 \le K \le N/2 - 1$

→ $f_1(n)$ is split into two sequences $g_1(n)$ & $g_2(n)$ consisting of even and odd samples of $f_1(n)$ respectively.

$g_1(n) = f_1(2n)$ & $g_2(n) = f_1(2n+1)$ $\qquad 0 \le n \le \frac{N}{4} - 1$

$$\therefore F_1(K) = \sum_{n(\text{even})} f_1(n) W_{N/2}^{Kn} + \sum_{n(\text{odd})} f_1(n) W_{N/2}^{Kn}$$

$$= \sum_{n=0}^{\frac{N}{4}-1} f_1(2n) W_{N/2}^{K2n} + \sum_{n=0}^{\frac{N}{4}-1} f_1(2n+1) W_{N/2}^{K(2n+1)}$$

$$F_1(K) = \sum_{n=0}^{N/4-1} g_1(n) W_{N/4}^{Kn} + \sum_{n=0}^{N/4-1} g_2(n) W_{N/4}^{Kn} W_{N/2}^{K}$$

$$= \sum_{n=0}^{N/4-1} g_1(n) W_{N/4}^{Kn} + W_{N/2}^{K} \sum_{n=0}^{N/4-1} g_2(n) W_{N/4}^{Kn}$$

$$\boxed{F_1(K) = e_1(K) + W_{N/2}^{K} e_2(K)} \quad , \quad 0 \le K \le \frac{N}{4} - 1$$

$e_1(K)$ & $e_2(K)$ are $N/4$ point DFT and they are periodic with period $N/4$

$$e_1(K + \tfrac{N}{4}) = e_1(K) \quad , \quad e_2(K + \tfrac{N}{4}) = e_2(K) \qquad 0 \le K \le N/4$$

& $W_{N/2}^{(K+N/4)} = - W_{N/2}^{K}$

$$\boxed{\therefore \; F_1(K + N/4) = e_1(K) - W_{N/2}^{K} e_2(K)}$$

Illly $f_2(n)$ is split into two sequences $h_1(n)$ & $h_2(n)$ consisting of even and odd samples of $f_2(n)$ respectively.

$$h_1(n) = f_2(2n) \quad , \quad h_2(n) = f_2(2n+1) \qquad 0 \le n \le \frac{N}{4} - 1$$

wkt $\quad F_2(K) = \sum_{n=0}^{N/2-1} f_2(n) W_{N/2}^{Kn}$

$$\therefore F_2(K) = \sum_{n(\text{even})} f_2(n) W_{N/2}^{Kn} + \sum_{n(\text{odd})} f_2(n) W_{N/2}^{Kn} \qquad 0 \le n \le \frac{N}{2} - 1$$

$$F_2(K) = \sum_{n=0}^{N/4-1} f_2(2n)\, W_{N/2}^{2Kn} + \sum_{n(odd)}^{N/4-1} f_2(2n+1)\, W_{N/2}^{K(2n+1)}$$

$$= \sum_{n=0}^{\frac{N}{4}-1} h_1(n)\, W_{N/4}^{Kn} + W_{N/2}^{K} \sum_{n=0}^{\frac{N}{4}-1} h_2(n)\, W_{N/4}^{Kn} \qquad 0 \le K \le \frac{N}{4}-1$$

$$F_2(K) = H_1(K) + W_{N/2}^{K}\, H_2(K)$$

$$\boxed{\begin{aligned} F_2(K) &= H_1(K) + W_{N/2}^{K}\, H_2(K) \\ F_2(K+N/4) &= H_1(K) - W_{N/2}^{K}\, H_2(K) \end{aligned}} \qquad 0 \le K \le \frac{N}{4}-1$$

where $\quad H_1(K) = \sum_{n=0}^{\frac{N}{4}-1} h_1(n)\, W_{N/4}^{Kn}$

$$H_2(K) = \sum_{n=0}^{\frac{N}{4}-1} h_2(n)\, W_{N/4}^{Kn}$$

∴ The number of complex multiplications required to compute the N-point DFT at the end of II decimation is

$$= \underset{\substack{G_1(K),\, G_2(K) \\ H_1(K),\, H_2(K)}}{4\left(\frac{N}{4}\right)^2} + \underset{\substack{F_1(K) \\ F_2(K)}}{2\left(N/4\right)} + \underset{\substack{G_1(K) \\ G_2(K)}}{N/2} \qquad 2\left(\frac{N}{4}\right)$$

$$2\left(\frac{N}{2}\right)^2 + \frac{N}{2}$$

$$4\left(\frac{N}{4}\right)^2 + 2\,\frac{N}{4}$$

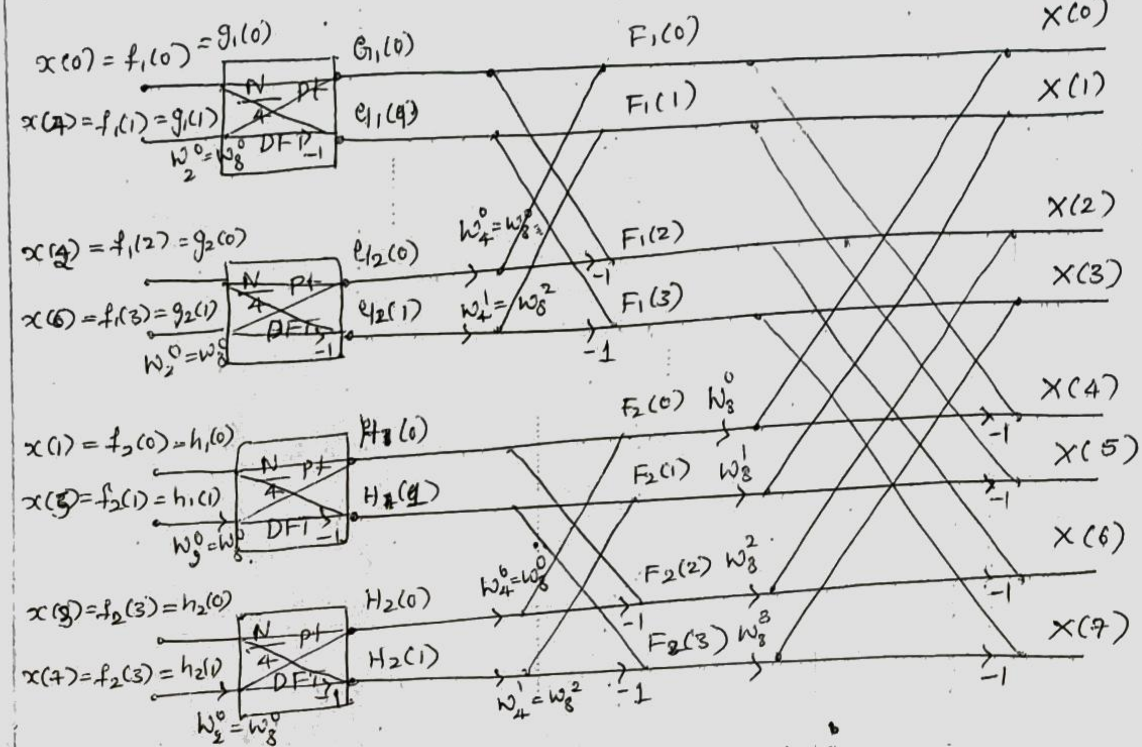
$$= \underline{\underline{\frac{N^2}{4} + N}}$$

* No. of complex additions required to compute the N-point DFT at the end of II decimation.

$$= 4\left[\frac{N}{4}\left(\frac{N}{4}-1\right)\right] + 2\left[\frac{N}{2}\right] + N = \underline{\underline{\frac{N^2}{4} + N}}$$

$$\cancel{4}\,\frac{N^2}{\cancel{16}\,4} - 4\,\frac{N}{4} + N + N$$

The flow diagram at the end of II decimation for N=8 is



* NO of Butterfly diagram in each stage $= N/2$

* NO of Decimations $= \vartheta = \log_2 N$ (or) NO of stages.

* NO of complex Multiplications required to compute the N-pt DFT $= \frac{N}{2} \cdot \vartheta = N/2 \times \log_2 N$

* NO of complex additions required $= N \cdot \vartheta = N \log_2 N$

* The speed factor $=$ $\dfrac{\text{NO of Multiplication required for Direct computation}}{\text{NO of} \text{ — " — } \text{FFT Algorithm}}$

$$= \frac{N^2}{\frac{N}{2} \cdot \vartheta} = \frac{N^2}{\frac{N}{2} \log_2 N}$$

* Comparison of computational complexity for the Direct Computation of the DFT Versus the FFT Algorithm.

No of pt N	complex multiplication in direct comput N^2	complex Mult in FFT Algorithm $\frac{N}{2}\log_2 N$	speed Improvement factor.
4	16	4	4.0
8	64	12	5.3
16	256	32	8.0
32	1024	80	12.8
64	4096	192	21.3
128	16,384	448	36.6
256	65,536	1024	64.0
512	262,144	2,304	113.8
1024	10,48,576	5,120	204.8

Input combination : DIF - FFT

$x(n) = x(0) \quad x(1) \quad x(2) \quad x(3) \quad x(4) \quad x(5) \quad x(6) \quad x(7)$

→ grouping as even & odd sequence

	even				odd			
	$x(0)$	$x(2)$	$x(4)$	$x(6)$	$x(1)$	$x(3)$	$x(5)$	$x(7)$
At even sequence	0	1	2	3	0	1	2	3
positions	$F_1(0)$	$F_1(1)$	$F_1(2)$	$F_1(3)$	$F_2(0)$	$F_2(1)$	$F_2(2)$	$F_2(3)$

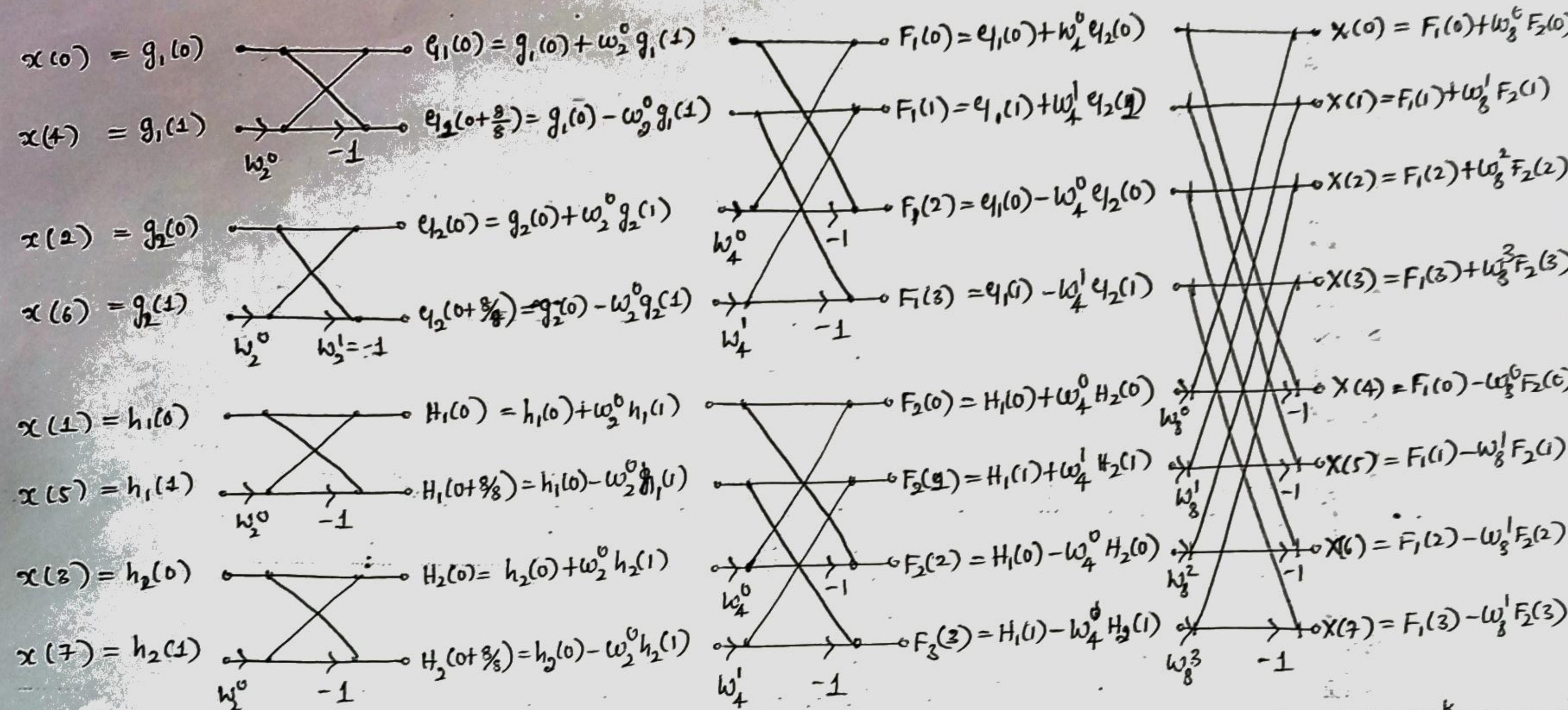
→ grouping as even & odd.

even: $\underbrace{x(0) \quad x(4)}_{g_1(n)}$ odd: $\underbrace{x(2) \quad x(6)}_{g_2(n)}$

$G_1(0) \quad G_1(1) \quad G_2(0) \quad G_2(1)$

even: $\underbrace{x(1) \quad x(5)}_{h_1(n)}$ odd: $\underbrace{x(3) \quad x(7)}_{h_2(n)}$

$H_1(0) \quad H_1(1) \quad H_2(0) \quad H_2(1)$



$$G_1(K) = g_1(0) + W_{N/4}^{K} g_1(1)$$

$$G_1(K+N/8) = g_1(0) - W_{N/4}^{K} g_1(1)$$

$$0 \leq K \leq N/8$$

$$K = 0$$

$$F_1(K) = G_1(K) + W_{N/2}^{K} G_2(K)$$

$$F_1(K+N/4) = G_1(K) - W_{N/2}^{K} G_2(K)$$

$$0 \leq K \leq N/4$$

$$N = 8$$

$$K = 0, 1$$

$$X(K) = F_1(K) + W_N^{K} F_2(K)$$

$$X(K+N/2) = F_1(K) - W_N^{K} F_2(K)$$

$$0 \leq K \leq N/2$$

$$N = 8$$

$$K = 0, 1, 2, 3$$

(*) compute the 4-pt DFT of the sequence $x(n) = (1,0,1,0)$ using DIT-FFT radix-2 algorithm

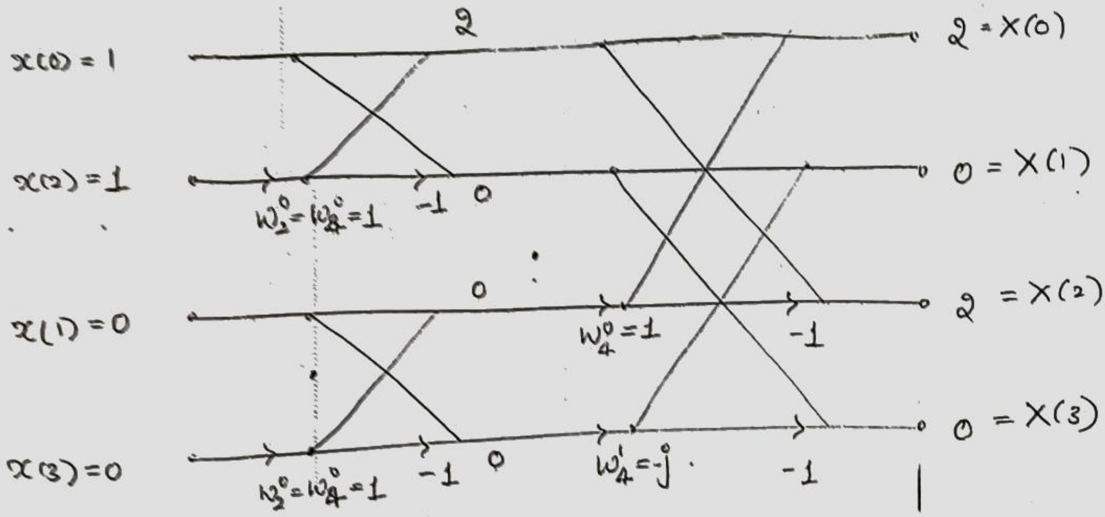
Sol^n

$W_4^0 = 1$

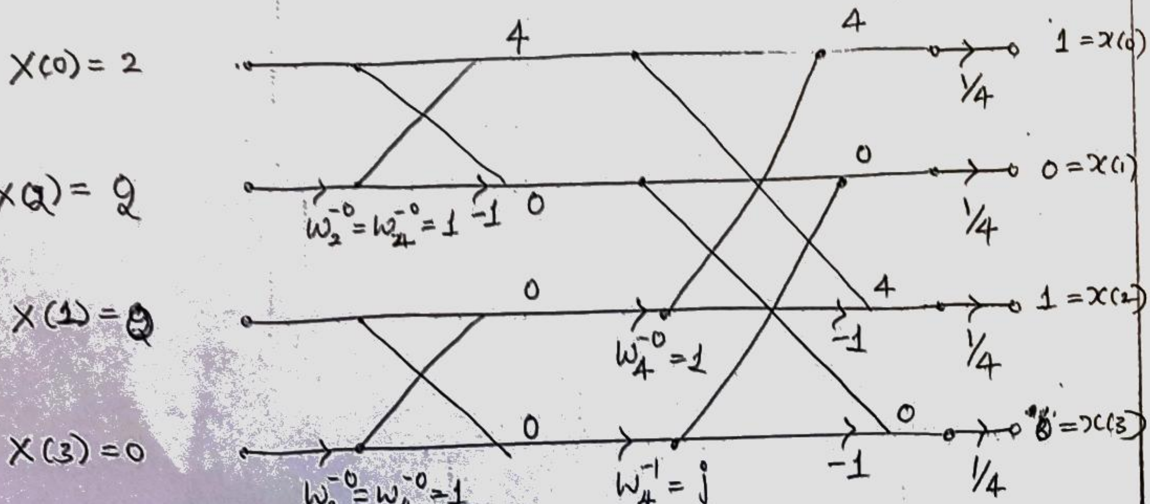
$W_4^1 = -j$

$W_4^{-0} = (W_4^0)^* = 1$

$W_4^{-1} = (W_4^1)^* = j$



IDFT $\Rightarrow$ $x(n) = \frac{1}{N}\sum_{K=0}^{N-1} X(K)\, W_N^{-Kn}$ $\quad 0 \leq n \leq N-1$



(*) Use the 8-point radix-2 DIT-FFT algorithm to find the DFT of the sequence. $x(n) = \{0.707, 1, 0.707, 0, -0.707, -1, -0.707, 0\}$

$\quad\quad 0 \quad 1 \quad 2 \quad 3 \quad 4 \quad 5 \quad 6 \quad 7$

Sol^n

$W_2^0 = W_4^0 = W_8^0 = 1$

$= W_8^1 = 0.7071 - j0.7071$

$W_4^1 = W_8^2 = -j$

$W_8^3 = -0.7071 - j0.7071$

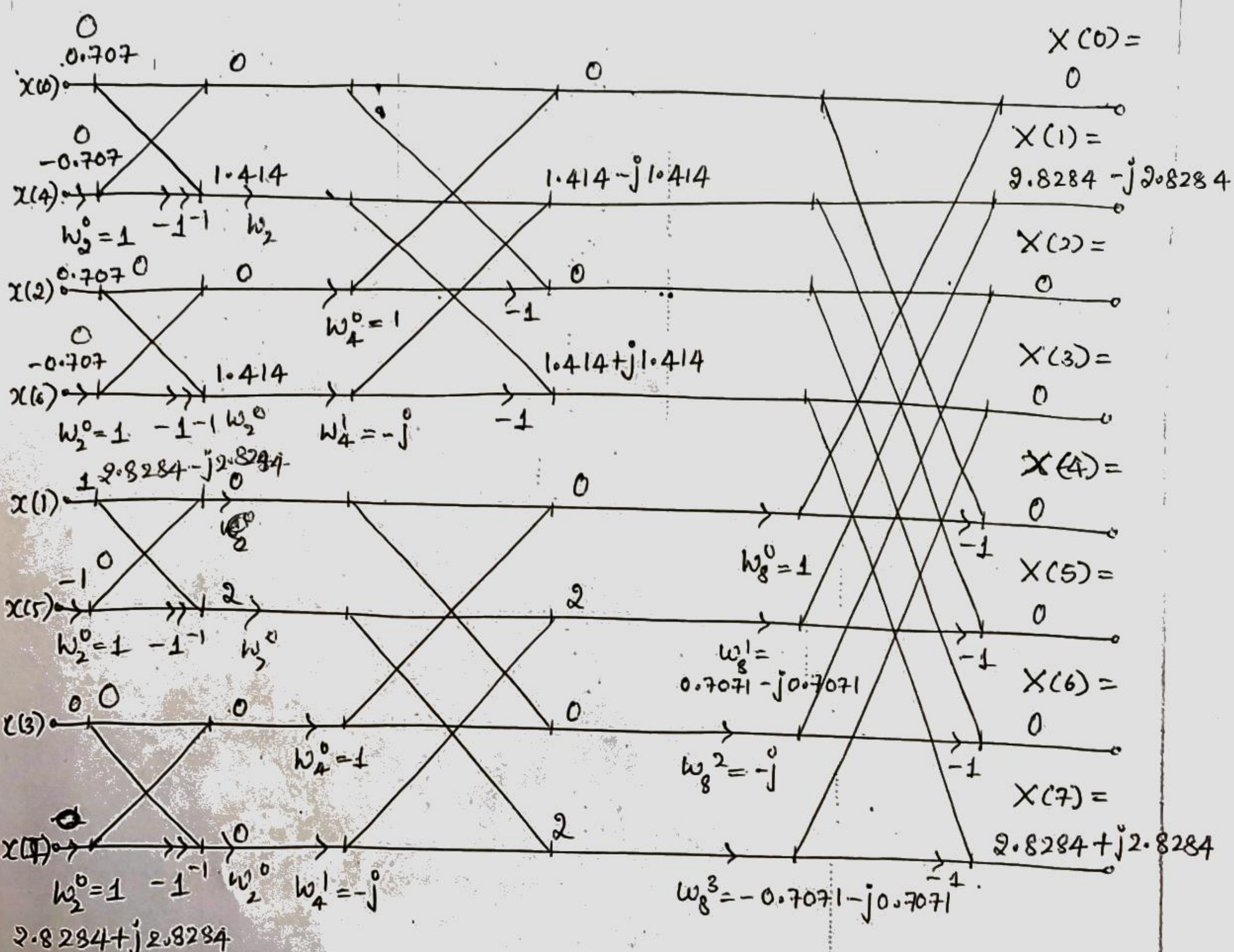
$X(N-K) = X^*(K)$

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

$X(5) = X^*(8-5) = X^*(3)$

$X(6) = 0$

$X(7) = 2.8284 + j2.8284$



(*) First five points of the eight point DFT of a real valued sequence is given by $X(0) = 0$, $X(1) = 2+j2$, $X(2) = -4j$, $X(3) = 2-2j$, $X(4) = 0$. Determine the remaining pts. Hence find the original sequence $x(n)$ using Decimation in frequency FFT algorithm.

Radix - 2 DIF - FFT Algorithm :-

Consider a finite length sequence $x(n)$ having N-samples

$N = 2^{v}$ where $v \rightarrow$ Integer.

* In this method the frequency spectrum $X(k)$ is split into smaller and smaller sequences till we get a sequence of length 1.

By def^n $X(k) = DFT\{x(n)\}$

$$= \sum_{n=0}^{N-1} x(n)\, W_N^{kn} \qquad 0 \le k \le N-1$$

→ The time domain sequence $x(n)$ is split into 2 sequences in the natural order

$n = N/2$

$N/2 = n + N/2$

$n = 0$

$n + N/2 = N-1$

$n = N-1-\frac{N}{2}$

$n = N/2 - 1$

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

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

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

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

$$\left(\because W_N^{kN/2} = (-1)^k\right)$$

* Splitting the sequence $X(k)$ into two sequences $F_1(k)$ & $F_2(k)$ of length $N/2$ consisting of even & odd samples of $X(k)$ respectively

Even samples of $X(K) = X(2K) = F_1(K)$

odd — " —————— $= X(2K+1) = F_2(K)$

Eq^n ① will be

$$F_1(K) = X(2K) = \sum_{n=0}^{\frac{N}{2}-1} \left[x(n) + (-1)^{2K} x(n+N/2)\right] W_N^{2Kn}$$

$$= \sum_{n=0}^{\frac{N}{2}-1} \underbrace{\left[x(n) + x(n+N/2)\right]}_{f_1(n)} W_{N/2}^{Kn} \qquad 0 \le K \le \frac{N}{2}-1$$

$$\left[\because (-1)^{2K} = 1, \quad \& \; N \Rightarrow N/2, \quad n \Rightarrow N/2 \right.$$

$$= \sum_{n=0}^{\frac{N}{2}-1} f_1(n) W_{N/2}^{Kn} \qquad 0 \le K \le \frac{N}{2}-1$$

where $f_1(n) = x(n) + x(n+N/2)$

$$\boxed{\therefore \; F_1(K) = X(2K) = \sum_{n=0}^{\frac{N}{2}-1} f_1(n) W_{N/2}^{Kn}} \qquad 0 \le K \le \frac{N}{2}-1 \longrightarrow ②$$

Illy.

$$F_2(K) = X(2K+1) = \sum_{n=0}^{\frac{N}{2}-1} \left[x(n) - x(n+N/2)\right] W_N^{2Kn} W_N^n$$

$$= \sum_{n=0}^{\frac{N}{2}-1} \left\{ \left[x(n) - x(n+N/2)\right] W_N^n \right\} W_{N/2}^{Kn}$$

$$\boxed{F_2(K) = X(2K+1) = \sum_{n=0}^{\frac{N}{2}-1} f_2(n) W_{N/2}^{Kn}} \qquad 0 \le K \le \frac{N}{2}-1 \longrightarrow ③$$

where $f_2(n) = \left[x(n) - x(n+N/2)\right] W_N^n \qquad 0 \le n \le \frac{N}{2}-1$

For $N=8 \Rightarrow$

$$\left. \begin{aligned} f_1(n) &= x(n) + x(n+8/2) \\ f_2(n) &= x(n) - x(n+8/2) W_8^n \end{aligned} \right\} 0 \le n \le \frac{8}{2}-1$$

$$\Rightarrow \left. \begin{aligned} f_1(n) &= x(n) + x(n+4) \\ f_2(n) &= x(n) - x(n+4) W_8^n \end{aligned} \right\} 0 \le n \le 3$$

when $n=0$, $f_1(0) = x(0) + x(4)$

$$f_2(0) = [x(0) - x(4)]\, W_8^0$$

$n=1$, $f_1(1) = x(1) + x(5)$

$$f_2(1) = [x(1) - x(5)]\, W_8^1$$

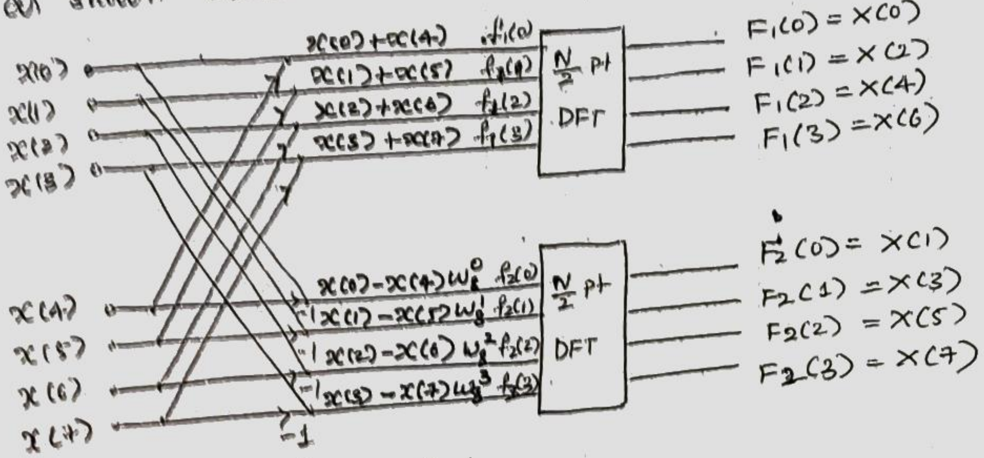
$n=2$, $f_1(2) = x(2) + x(6)$

$$f_2(2) = [x(2) - x(6)]\, W_8^2$$

$n=3$, $f_1(3) = x(3) + x(7)$

$$f_2(3) = [x(3) - x(7)]\, W_8^2 .$$

The flow diagram for $N=8$, at the end of 1^{st} decimation is as shown below.



wkt $$F_1(K) = X(2K) = \sum_{n=0}^{\frac{N}{2}-1} f_1(n)\, W_{N/2}^{Kn}, \quad 0 \le K \le \frac{N}{2}-1$$

* The time domain sequence $f_1(n)$ is split into two sequences in the natural order

$$F_1(K) = \sum_{n=0}^{\frac{N}{4}-1} f_1(n)\, W_{N/2}^{Kn} + \sum_{n=N/4}^{\frac{N}{2}-1} f_1(n)\, W_{N/2}^{Kn}$$

$$= \sum_{n=0}^{\frac{N}{4}-1} f_1(n)\, W_{N/2}^{Kn} + \sum_{n=0}^{\frac{N}{4}-1} f_1(n+N/4)\, W_{N/2}^{K(n+N/4)}$$

$$= \sum_{n=0}^{\frac{N}{4}-1} f_1(n) W_{N/2}^{kn} + \sum_{n=0}^{\frac{N}{4}-1} f_1(n+\tfrac{N}{4}) W_{N/2}^{kn} \underbrace{W_{N/2}^{kN/4}}_{(-1)^K}.$$

$$= \sum_{n=0}^{\frac{N}{4}-1} \left[f_1(n) + (-1)^k f_1(n+N/4)\right] W_{N/2}^{kn} \quad . \quad 0 \leq K \leq \tfrac{N}{2}-1$$

→ $F_1(K)$ is split into two sequences $G_1(K)$ & $G_2(K)$ each of length $N/4$ consisting of even and odd samples of $F_1(k)$ respectively.

$G_1(k)$ = even samples of $F_1(K) = F_1(2K) = X(4K)$

$G_2(K)$ = odd samples of $F_1(k) = F_1(2K+1) = X(4K+2)$

$$G_1(K) = F_1(2K) = X(4K) = \sum_{n=0}^{\frac{N}{4}-1} \left[f_1(n) + (-1)^{2K} f_1(n+\tfrac{N}{4})\right] W_{N/2}^{2kn}$$

$$G_1(k) = \sum_{n=0}^{\frac{N}{4}-1} \left[f_1(n) + f_2(n+\tfrac{N}{4})\right] W_{N/4}^{kn} \qquad 0 \leq K \leq \tfrac{N}{4}-1$$

$$\boxed{G_1(K) = X(4K) = \sum_{n=0}^{\frac{N}{4}-1} g_1(n) W_{N/4}^{kn}} \qquad 0 \leq K \leq \tfrac{N}{4}-1$$

where $g_1(n) = f_1(n) + f_1(n+N/4) \qquad 0 \leq n \leq \frac{N}{4}-1$

$$G_2(K) = F_1(2K+1) = X(4K+2)$$

$$= \sum_{n=0}^{\frac{N}{4}-1} \left[f_1(n) + (-1)^{2K+1} f_1(n+\tfrac{N}{4})\right] W_{N/4}^{(2K+1)n}$$

$$= \sum_{n=0}^{\frac{N}{4}-1} \left\{\left[f_1(n) - f_1(n+\tfrac{N}{4})\right] W_{N/2}^{n}\right\} W_{N/4}^{kn}$$

$$G_2(K) = X(4K+2) = \sum_{n=0}^{\frac{N}{4}-1} g_2(n) W_{N/4}^{kn} \qquad 0 \leq K \leq \tfrac{N}{4}-1$$

Illy $F_2(K) = X(2K+1)$

$$= \sum_{n=0}^{\frac{N}{4}-1} \left[f_2(n) + (-1)^K f_2(n + N/4) \right] W_{N/2}^{Kn} \qquad 0 \le K \le \frac{N}{2} - 1$$

$F_2(K)$ is split into two sequences $H_1(K)$ & $H_2(K)$ of length $\frac{N}{4}$ consisting of even and odd samples of $F_2(K)$ respectively

$$H_1(K) = F_2(2K) = X(4K+1)$$

$$H_2(K) = F_2(2K+1) = X(4K+3)$$

$$H_1(K) = \sum_{n=0}^{\frac{N}{4}-1} \left[f_2(n) + f_2(n + \tfrac{N}{4}) \right] W_{N/4}^{Kn}, \qquad 0 \le K \le \frac{N}{4} - 1.$$

$$H_1(K) = \sum_{n=0}^{\frac{N}{4}-1} h_1(n) \cdot W_{N/4}^{Kn} \qquad 0 \le K \le \frac{N}{4} - 1.$$

where $h_1(n) = f_2(n) + f_2(n + N/4) \qquad 0 \le n \le \frac{N}{4} - 1$

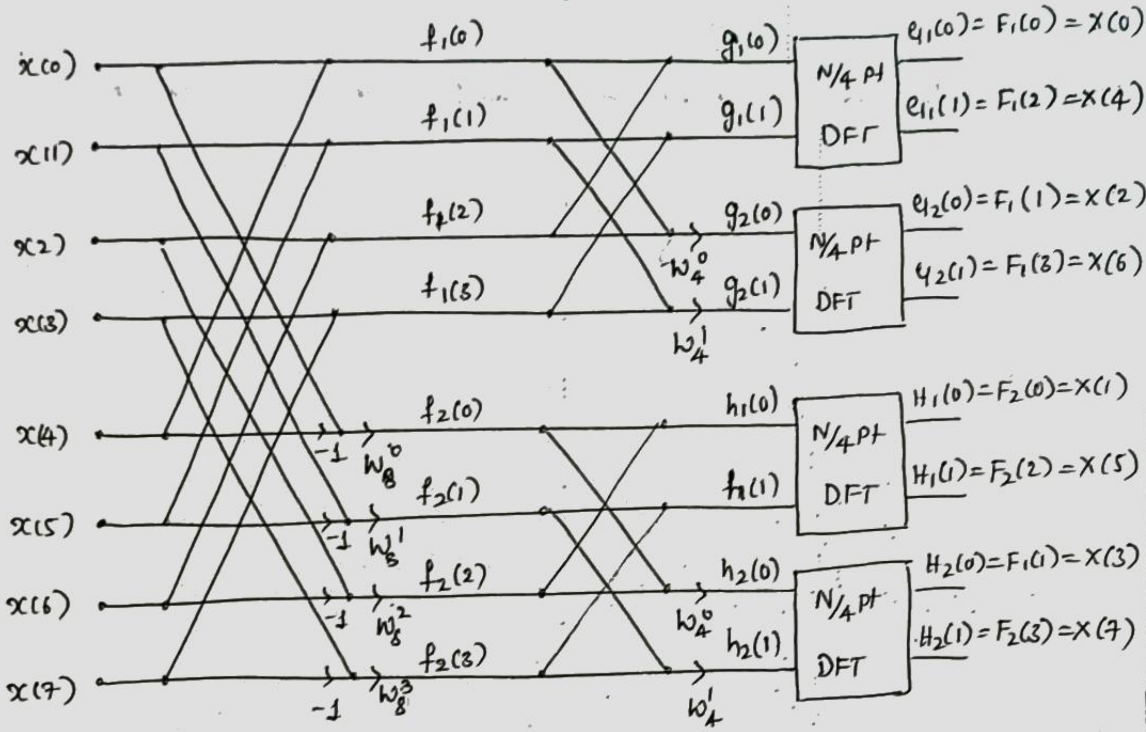
$$H_2(K) = \sum_{n=0}^{\frac{N}{4}-1} \left[f_2(n) - f_2(n + N/4) \right] W_{N/2}^{(2K+1)n}$$

$$= \sum_{n=0}^{\frac{N}{4}-1} \left\{ \left[f_2(n) - f_2(n + \tfrac{N}{4}) \right] W_{N/2}^{n} \right\} W_{N/4}^{Kn}$$

where $h_2(n) = \left[f_2(n) - f_2(n + \tfrac{N}{4}) \right] W_{N/4}^{n} \qquad 0 \le n \le \frac{N}{4} - 1$

	STAGE 1	STAGE 2	STAGE 3
	$f_1(n)$, $f_2(n)$	$g_1(n), g_2(n)$, $h_1(n), h_2(n)$	$G_1(K)$ $G_2(K)$, $H_1(K)$ $H_2(K)$
N = 8	4+4	2+2+2+2	2+2+2+2.

The flow diagram at the end of second decimation for $N=8$ is as shown below.



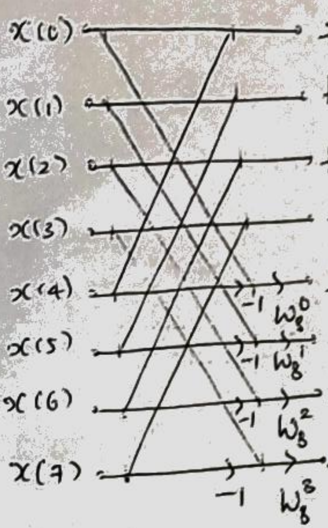
* NO of Butterfly diagrams in each case $= N/2$
* NO of decimation $= \nu = \log_2 N$
* NO of complex multiplication and additions required to compute N-pt DFT are $\frac{N}{2}\nu$ & $N\nu$ respectively.

OBSERVATION :-

(1) I/P is in normal order

(2) O/P is in bit-reversed order.

Stage-1



$f_1(0) = x(0) + x(4)$

$f_1(1) = x(1) + x(5)$

$f_1(2) = x(2) + x(6)$

$f_1(3) = x(3) + x(7)$

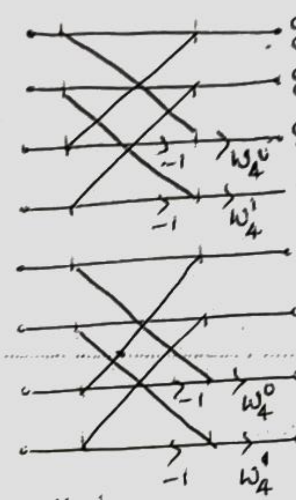
$f_2(0) = (x(0) - x(4))\, W_8^0$

$f_2(1) = (x(1) - x(5))\, W_8^1$

$f_2(2) = (x(2) - x(6))\, W_8^2$

$f_2(3) = (x(3) - x(7))\, W_8^3$

Stage-2



$g_1(0) = f_1(0) + f_1(2)$

$g_1(1) = f_1(1) + f_1(3)$

$g_2(0) = (f_1(0) - f_1(2))\, W_4^0$

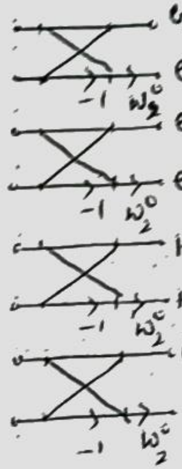
$g_2(1) = (f_1(1) - f_1(3))\, W_4^1$

$h_1(0) = h_1(0) + h_1(2)$

$h_1(1) = h_1(1) + h_1(3)$

$h_2(0) = (h_1(0) - h_1(2))\, W_4^0$

$h_2(0) = (h_1(1) - h_1(3))\, W_4^1$



$e_1(0) = g_1(0) + g_1(1) = X(0)$

$e_1(1) = (g_1(0) - g_1(1))\, W_2^0 = X(4)$

$e_2(0) = g_2(0) + g_2(1) = X(2)$

$e_2(0) = (g_2(0) - g_2(1))\, W_2^0 = X(6)$

$H_1(0) = (h_1(0) + h_1(1)) = X(1)$

$H(1) = (h_1(0) - h_1(1))\, W_2^0 = X(5)$

$H_2(0) = h_1(0) + h_1(1) = X(3)$

$H_2(1) = (h_1(0) - h_1(1))\, W_2^0 = X(7)$

$X(K) \Rightarrow F_1(K)$ & $F_2(K)$

e $X(2K)$ o $X(2K+1)$

$$F_1(K) = \sum_{n=0}^{N/2-1} f_1(n)\, W_{N/2}^{Kn}$$

$$F_2(K) = \sum_{n=0}^{N/2-1} f_2(n)\, W_{N/2}^{Kn}$$

$f_1(n) = x(n) + x(n+N/2)$

$f_2(n) = [x(n) - x(n+N/2)]\, W_N^n$

$n = 0, 1, 2, 3$

$$F_1(K) = \sum_{n=0}^{N/4-1} \left[f_1(n) + (-1)^K f_1\left(n+\frac{N}{4}\right)\right] W_{N/2}^K$$

$e_1(K)$ e $X(4K)$ $e_2(K)$ o $X(4K+2)$

$$e_1(K) = \sum_{n=0}^{N/4-1} g_1(n)\, W_{N/4}^{Kn} \qquad n = 0, 1.$$

$$e_2(K) = \sum_{n=0}^{N/4-1} g_2(n)\, W_{N/4}^{Kn}$$

$g_1(n) = f_1(n) + f_1(n+N/4)$

$g_2(n) = f_1(n) - f_1(n+N/4)\, W_{N/2}^n$

|||y. $h_1(n) = f_2(n) + f_2(n+N/4)$

$h_2(n) = f_2(n) - f_2(n+N/4)\, W_{N/2}^n$

$$e_1(K) = \sum_{n=0}^{N/4-1} g_1(n)\, W_{N/4}^{Kn}$$

$$e_2(K) = \sum_{n=0}^{N/4-1} g_2(n)\, W_{N/4}^{Kn}$$

$$H_1(K) = \sum_{n=0}^{N/4-1} h_1(n)\, W_{N/4}^{Kn}$$

$$H_2(K) = \sum_{n=0}^{N/4-1} h_2(n)\, W_{N/4}^{Kn}$$

$e_1(K) = X(4K)$, $e_2(K) = X(4K+2)$

$H_1(K) = X(4K+1)$

$K = 0, 1$ $H_2(K) = X(4K+3)$

11 (a)

$x(n) \longrightarrow X(K)$

by splitting into Natural 2 sets

$$X(K) = \sum_{n=0}^{N/2-1} \left(x(n) + (-1)^K x(n+N/2)\right) W_N^{Kn}$$

split into even & odd

even

$K = 2K$

$F_1(K) = X(2K)$

$$= \sum_{n=0}^{N/2-1} \underbrace{[x(n) + x(n+N/2)]}_{f_1(n)} W_{N/2}^{Kn}$$

$K = 2K+1$

$F_2(K) = X(2K+1)$

$$= \sum_{n=0}^{N/2-1} \underbrace{(x(n) - x(n+N/2))W_N^n}_{f_2(n)} W_{N/2}^{Kn}$$

Split it as a natural 2 sets

$$\therefore F_1(K) = \sum_{n=0}^{N/4-1} [f_1(n) + (-1)^K f_1(n+N/4)] W_{N/2}^{Kn}$$

split it as a natural 2 sets

$F_2(K) = X(2K+1)$

$$= \sum_{n=0}^{N/4-1} \left[f_2(n) + (-1)^K f_2\left(n+\frac{N}{4}\right)\right] W_{N/2}^{Kn}$$

even

$K = 2K$

$G_1(K) = F_1(2K)$

$= X(4K)$

$$= \sum_{n=0}^{N/4-1} \underbrace{\left(f_1(n) + f_2\left(n+\frac{N}{4}\right)\right)}_{g_1(n)} W_{N/4}^{Kn}$$

odd

$K = 2K+1$

$G_2(K) = F_1(2K+1) = X(4K+2)$

$$= \sum_{n=0}^{N/4-1} \left[\underbrace{\left(f_1(n) - f_2(n+N/4)\right) W_{N/2}^{n}}_{g_2(n)}\right] W_{N/4}^{Kn}$$

even

$k = 2K$

ie $4K+1$

$H_1(K) = F_2(2K)$

$= X(4K+1)$

$$H_1(K) = \sum_{n=0}^{N/4-1} \left[\underbrace{f_2(n) + f_2\left(n+\frac{N}{4}\right)}_{h_1(n)}\right] W_{N/4}^{Kn}$$

odd

$K = 2K+1$

$= 2(2K+1)+1$

$= 4K+3$

$H_2(K) = F_2(2K+1)$

$= X(4K+3)$

$$H_2(K) = \sum_{n=0}^{N/4-1} \left[\underbrace{\left(f_2(n) - f_2(n+N/4)\right) W_{N/2}^{n}}_{h_2(n)}\right] W_{N/4}^{Kn}$$

(*) Find the 8-point DFT of a real sequence $x(n) = (1,2,2,2,1,0,0,0)$ using decimation-in-frequency FFT algorithm.

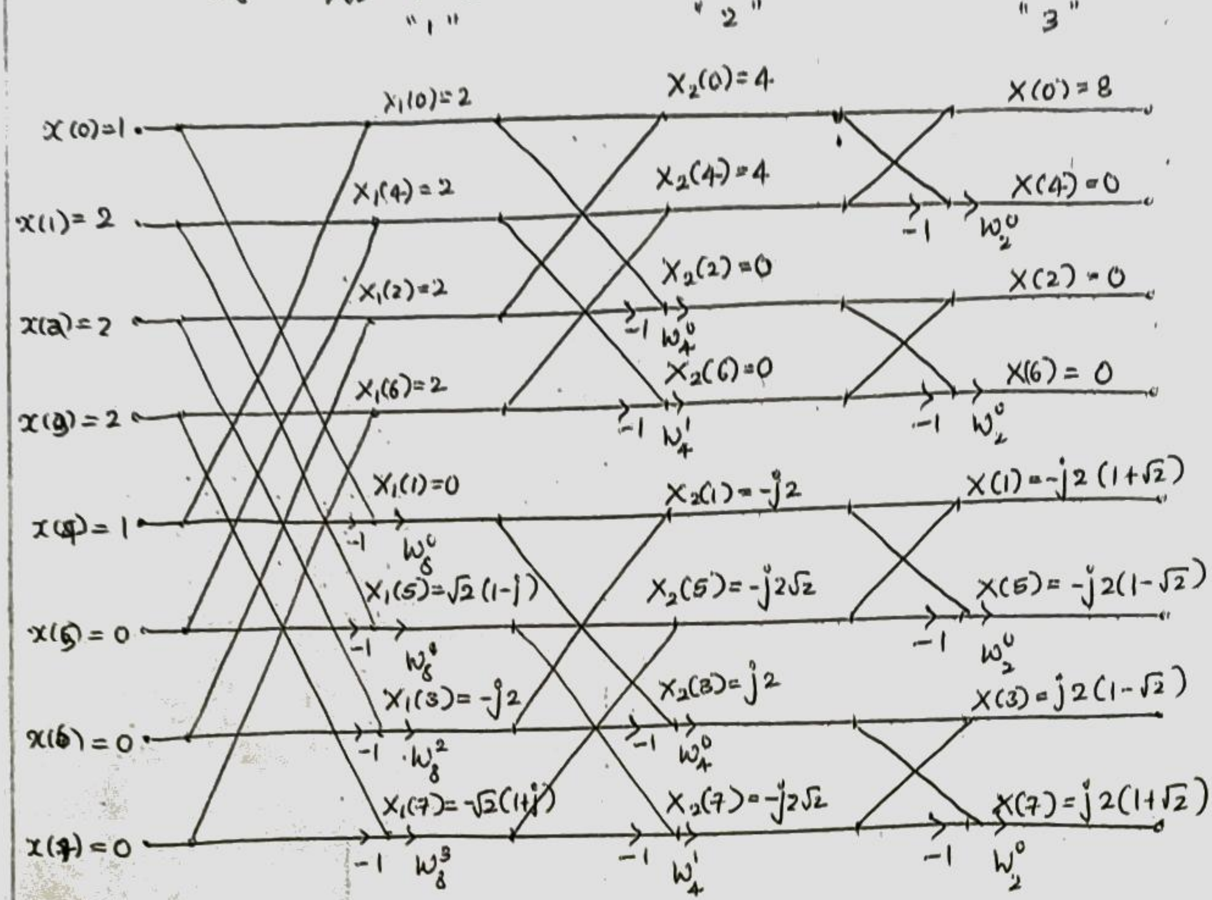
Sol^n:

$W_8^0 = 1$

$W_8^1 = \frac{1}{\sqrt{2}} - j\frac{1}{\sqrt{2}}$

$W_8^2 = -j$

$W_8^3 = -\frac{1}{\sqrt{2}} - j\frac{1}{\sqrt{2}}$

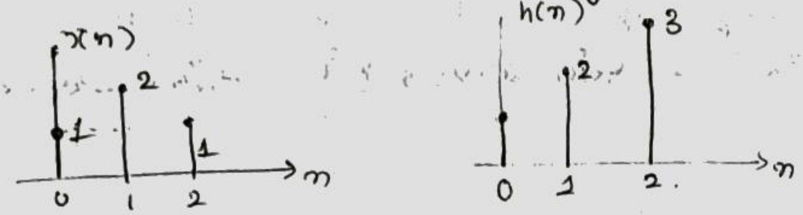


(*) Find the convolution of $x(n)$ & $h(n)$ given in below figure using

(a) The time - domain convolution operation

(b) The DFT & zero - padding

(c) The radix - 2 FFT and zero - padding.



Sol^n From figure

$$x(n) = (\underset{\uparrow}{1}, 2, 1) \quad \& \quad h(n) = (\underset{\uparrow}{1}, 2, 3)$$

(a) The time - domain convolution operation.

$$x(n) = \delta(n) + 2\delta(n-1) + \delta(n-2)$$

$$h(n) = \delta(n) + 2\delta(n-1) + 3\delta(n-2)$$

$$y(n) = x(n) * h(n)$$

$$= [\delta(n) + 2\delta(n-1) + \delta(n-2)] * [\delta(n) + 2\delta(n-1) + 3\delta(n-2)]$$

$$= \delta(n) + 2\delta(n)\delta(n-1) + 3\delta(n)\delta(n-2)$$

$$+ 2\delta(n)\delta(n-1) + 4\delta(n-1)\delta(n-1) + 6\delta(n-1)\delta(n-2)$$

$$+ \delta(n)\delta(n-2) + 2\delta(n-1)\delta(n-2) + 3\delta(n-2)\delta(n-2)$$

$$= \delta(n) + 4\delta(n-1) + 8\delta(n-2) + 8\delta(n-3) + 3\delta(n-4)$$

$$y(n) = \{1, 4, 8, 8, 3\}$$

(b) The DFT & zero - padding (zero - padding for convolution length)

N = length of $(x(n))$ + length of $h(n) - 1$

$= 3 + 3 - 1 = 5$

$\therefore$ $x(n) = (1, 2, 1, 0, 0)$

$h(n) = (1, 2, 3, 0, 0)$

$$\text{DFT}(x(n)) = X(K) = \sum_{n=0}^{N-1} x(n)\, W_N^{Kn} \qquad n = 0, 1 \cdots N-1$$

0 to 4

$$X(K) = x(0)\,W_N^0 + x(1)\,W_N^K + x(2)\,W_N^{2K} + \cancelto{0}{x(3)\,W_N^{3K}} + \cancelto{0}{x(4)\,W_N^{4K}}$$

$$X(0) = 1 + 2W_5^0 + 1W_5^0 = 1+2+1 = 4$$

$$X(1) = 1 + 2W_5^1 + 1W_5^2 = 1+2(0.3090-j0.9510) + (-0.809-j0.5877)$$

$$X(2) = 1 + 2W_5^2 + W_5^4 =$$

$$X(3) = 1 + 2W_5^3 + W_5^6 =$$

$$X(4) = 1 + 2W_5^4 + W_5^8 =$$

$$X(5) = 1 + 2W_5^5 + W_5^{10} =$$

$$\mathrm{DFT}\{h(n)\} = H(K) = \sum_{n=0}^{N-1} h(n)\,W_N^{Kn}$$

$$H(K) = h(0)\,W_N^0 + h(1)\,W_N^K + h(2)\,W_N^{2K} + \cancelto{0}{h(3)\,W_N^{3K}} + \cancelto{0}{h(4)\,W_N^{4K}}$$

$$H(0) = 1 + 2 + 3 = 6$$

$$H(1) = 1 + 2W_5^1 + 3W_5^2 =$$

$$H(2) = 1 + 2W_5^2 + 3W_5^4 =$$

$$H(3) = 1 + 2W_5^3 + 3W_5^6 =$$

$$H(4) = 1 + 2W_5^4 + 3W_5^8 =$$

$$H(5) = 1 + 2W_5^5 + 3W_5^{10} =$$

$$y(n) = x_1(n) \; \textcircled{N}_N \; h_1(n)$$

$$Y(K) = X_1(K) \cdot H_1(K)$$

$$Y(K) = (1, 4, 8, 8, 3)$$

(c) The radix - 2 FFT ~~seg~~ zeropadding.

zero padding ~~fs~~should be for convolved length

$x_2(n) = (1, 2, 1, 0, 0, 0, 0, 0)$

$h_2(n) = (1, 2, 3, 0, 0, 0, 0, 0)$

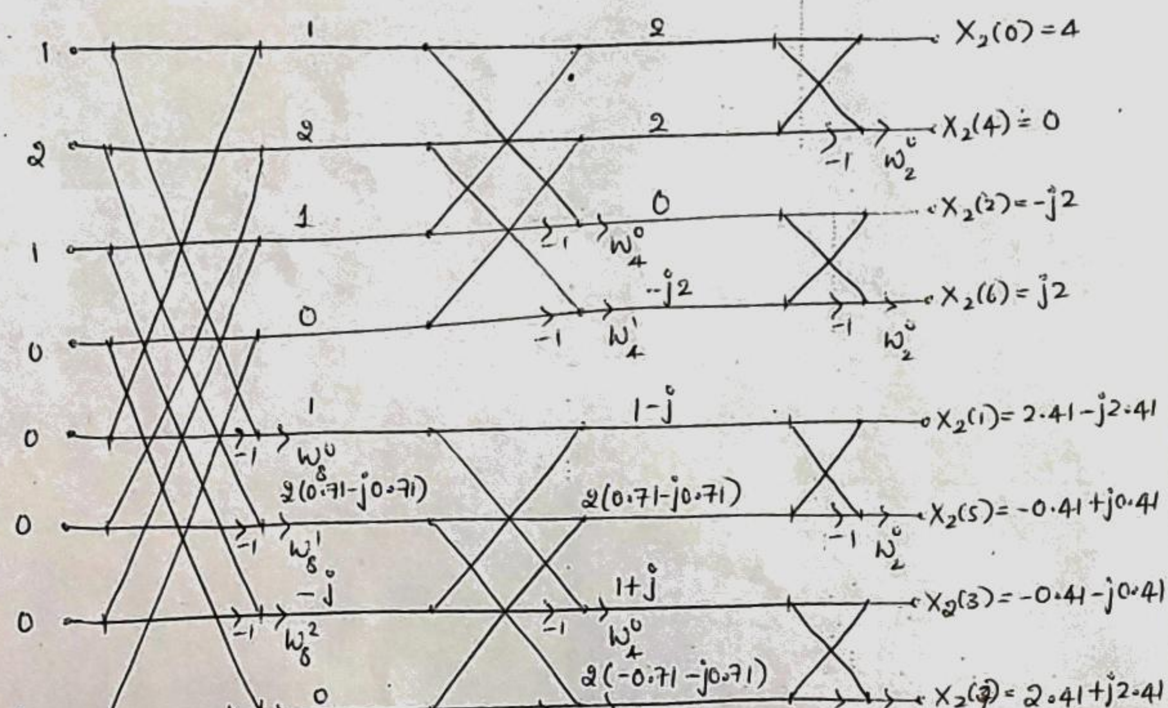
need to find out $X_2(K)$ & $H_2(K)$ using Radix-2 ~~DIT~~ DIF-FFT

$W_8^0 = 1 = W_4^0 = \omega_2^0$

$W_8^1 = \frac{1}{\sqrt{2}} - j\frac{1}{\sqrt{2}}$

$\omega_8^2 = 1 - j = \omega_8^2$

$\omega_8^3 = -1/\sqrt{2} - j\,1/\sqrt{2}$



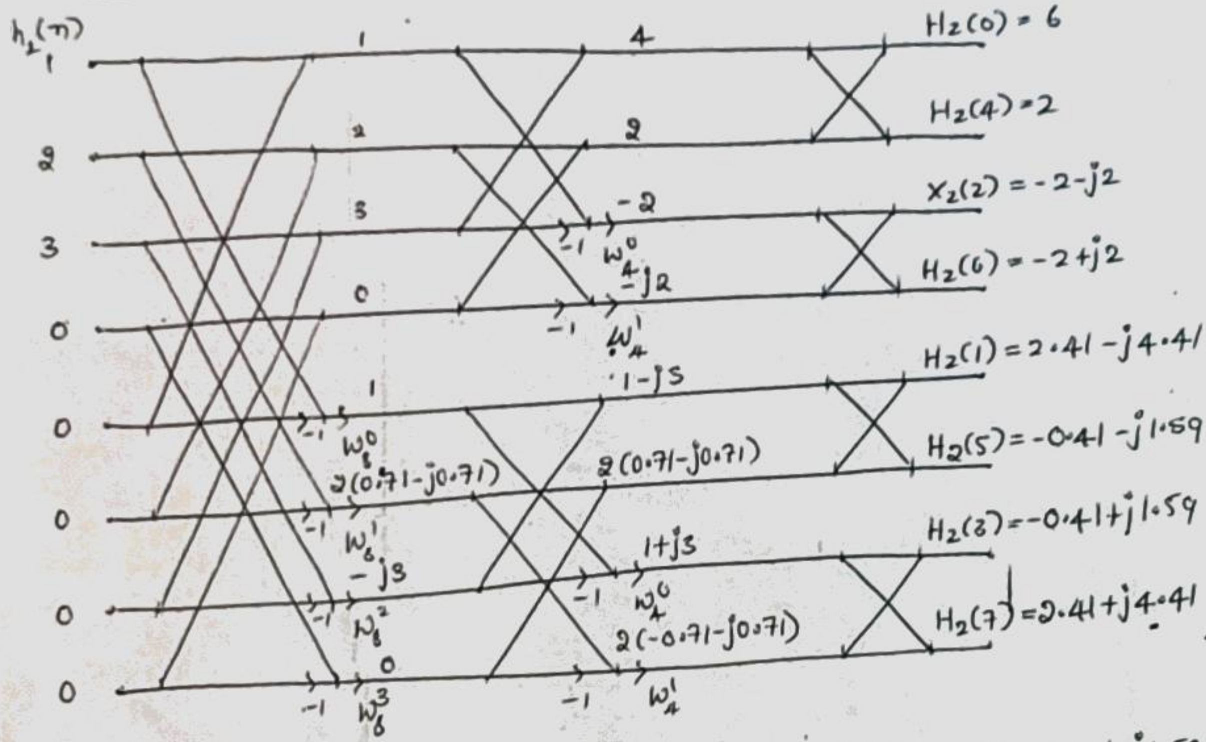
$X_2(K) = \{4,$ ~~2.41~~ $2.41 - j2.41, -j2, -0.41 - j0.41, 0, -0.41 + j0.41, j2, 2.41 + j2.41\}$

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

$X(0) = X^*(8-0) = x^*(8)$

$X(1) = X^*(8-1) = x^*(7)$

$H_2(K) = ?$

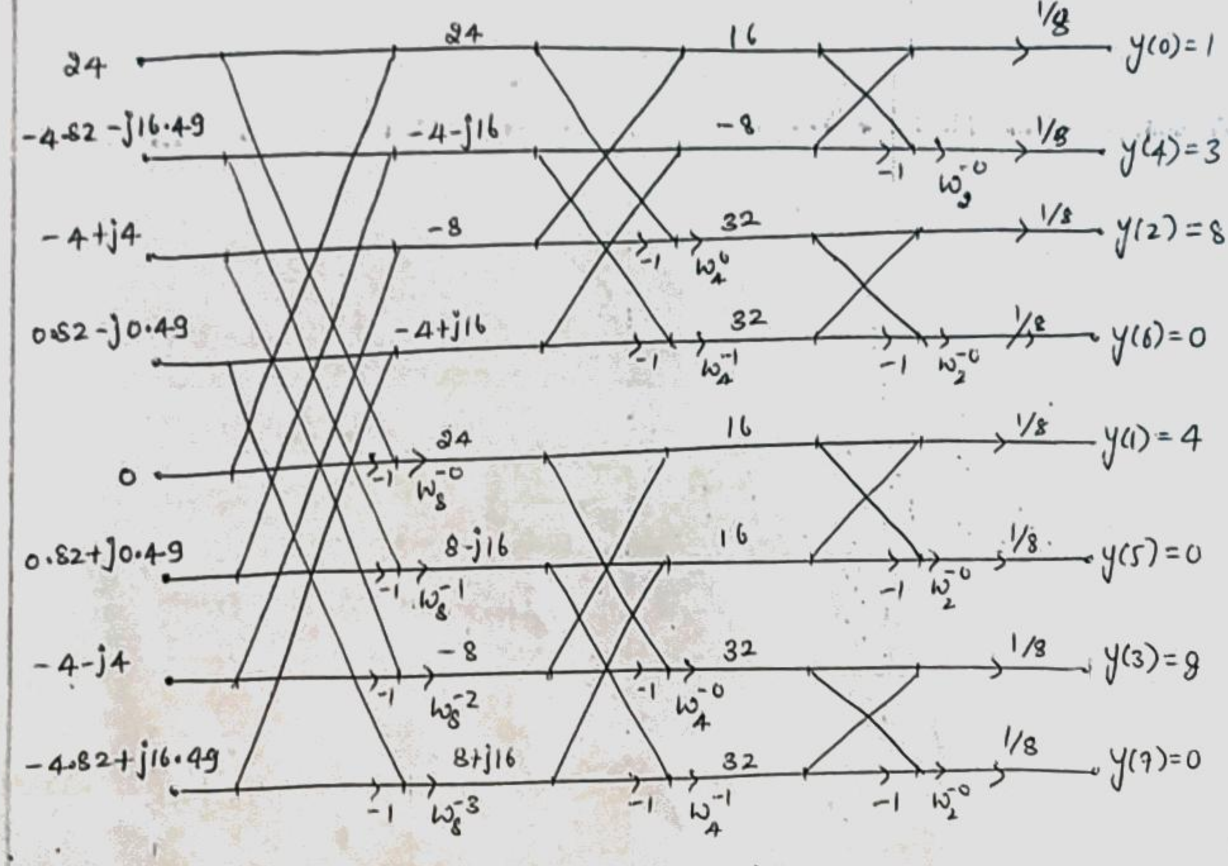


$H_2(K) = \{6,\ 2.41-j4.41,\ -2-j2,\ -0.41+j1.59,\ 2,\ -0.41-j1.59,$
$-2+j2,\ 2.41+j4.41\}$

$H_2(K) \cdot X_2(K) = Y_2(K)$

$X_2(K)$	$H_2(K)$	$Y_2(K)$
4	6	24
$2.41-j2.41$	$2.41-j4.41$	$-4.82-j16.49$
$-j2$	$-2-j2$	$-4+j4$
$-0.41-j0.41$	$-0.41+j1.59$	$0.82-j0.49$
0	2	0
$-0.41+j0.41$	$-0.41-j1.59$	$0.82+j0.49$
$j2$	$-2+j2$	$-4-j4$
$2.41+j2.41$	$2.41+j4.41$	$-4.82+j16.49$

IDFT $\{Y_2(k)\} = y(n)$



$$y(n) = (\dot{1}, 4, 8, 8, 3, 0, 0, 0)$$

$$y(n) = (1, 4, 8, 8, 3)$$

DIT - FFT	DIF - FFT
→ The time domain sequence is decimated	→ The DFT X(K) is decimated
→ I/P sequence is to be given in bit reversal order	→ The DFT at o/p is in bit reversed order.
→ First calculates 2-pt DFT and combines them	→ Decimates the sequence step by step to 2-pt sequence & calculates DFT
→ Suitable for calculating IDFT	→ Suitable for calculating DFT

Inverse Radix-2 FFT algorithms :-

* Inverse DFT can be computed with the help of FFT algorithms. Such algorithms are also called as Inverse radix-2 FFT algorithms

1. Inverse - DIT-FFT algorithms :-

wkt $x(n) = \frac{1}{N}\sum_{K=0}^{N-1} X(K) W_N^{-Kn}$

By spliting this formula into two summation.

$$x(n) = \frac{1}{N}\left[\sum_{K=0}^{N/2-1} X(K) W_N^{-Kn} + \sum_{K=N/2}^{N-1} X(K) W_N^{-Kn}\right]$$

By rearranging the second summation

$N/2 + K = N/2$

$K = 0$

$K + N/2 = N-1$

$K = N-1-N/2$

$K = N/2 - 1$

$$x(n) = \frac{1}{N}\left[\sum_{K=0}^{N/2-1} X(K) W_N^{-Kn} + \sum_{K=0}^{N/2-1} X(K+N/2) W_N^{-n(K+N/2)}\right]$$

$$= \frac{1}{N}\left[\sum_{K=0}^{N/2-1} X(K) W_N^{-Kn} + W_N^{-nN/2}\sum_{K=0}^{N/2-1} X(K+N/2) W_N^{-Kn}\right]$$

$$W_N^{-N/2} = -1 \Rightarrow (-1)^n$$

$$x(n) = \frac{1}{N}\left[\sum_{K=0}^{N/2-1} X(K) W_N^{-Kn} + (-1)^n \sum_{K=0}^{N/2-1} X(K+N/2) W_N^{-Kn}\right]$$

$$= \frac{1}{N}\left[\sum_{K=0}^{N/2-1}\left[X(K) + (-1)^n X(K+N/2)\right] W_N^{-Kn}\right]$$

Now, let us split $x(n)$ into even & odd number of samples

even $\Rightarrow 2n$

odd $\Rightarrow 2n+1$

$$x(2n) = \frac{1}{N}\left[\sum_{K=0}^{N/2-1}\left[X(K) + (-1)^{2n} X(K+N/2)\right] W_N^{-2Kn}\right]$$

$$= \frac{1}{N}\left[\sum_{K=0}^{N/2-1}\left[X(K) + (-1)^{2n} X(K+N/2)\right] W_{N/2}^{-Kn}\right]$$

$$= \frac{1}{2}\left[\frac{1}{N/2}\sum_{K=0}^{N/2-1}\underbrace{\left[X(K) + X(K+N/2)\right]}_{\to\, g_1(K)} W_{N/2}^{-Kn}\right]$$

$$x(2n+1) = \frac{1}{N}\left[\sum_{K=0}^{N/2-1}\left[X(K) + (-1)^{2n+1} X(K+N/2)\right] W_N^{-K(2n+1)}\right]$$

$$= \frac{1}{N}\left[\sum_{K=0}^{N/2-1}\left[X(K) - X(K+N/2)\right] W_N^{-2Kn} W_N^{-K}\right]$$

$$= \frac{1}{N}\left[\sum_{K=0}^{N/2-1}\left\{\left[X(K) - X(K+N/2)\right] W_N^{-K}\right\} W_{N/2}^{-Kn}\right]$$

$$= \frac{1}{2}\left[\frac{1}{N}\sum_{K=0}^{N/2-1}\underbrace{\left\{\left[X(K) - X(K+N/2)\right] W_N^{-K}\right\}}_{\to\, g_2(K)} W_{N/2}^{-Kn}\right]$$

$$g_1(K) = X(K) + X(K+N/2)$$

$$g_2(K) = \left(X(K) - X(K+N/2)\right) W_N^{-K}$$

To find i/p to 3stage DIT-FFT

$N = 8 \quad$ No of stages $\quad n = \log_2 N$

w.r.t $\quad f_1(n) = x(2n)$

$\qquad\qquad f_2(n) = x(2n+1)$

$$g_1(n) = f_1(2n) = x(2(2n)) = x(4n)$$

$$g_2(n) = f_1(2n+1) = x(2(2n+1)) = x(4n+2)$$

$$h_1(n) = f_2(2n) = x(2(2n)+1) = x(4n+1)$$

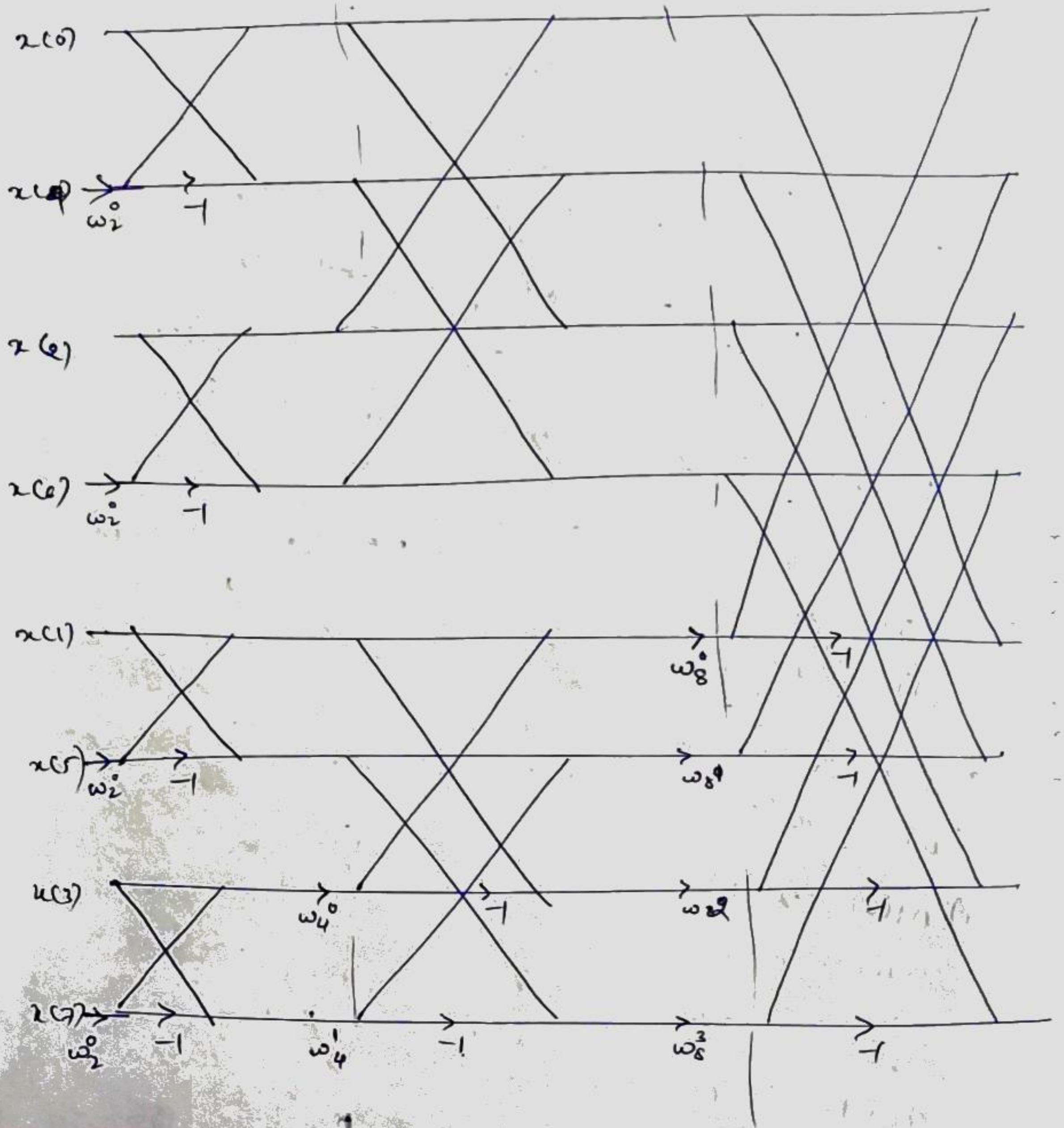
$$h_2(n) = f_2(2n+1) = x(2(2n+1)+1) = x(4n+3)$$

$n \to 0,1$ for g & h signals

	index	Normal/Natural repr
∴ $g_1(0) = x(0)$	000	000
$g_1(1) = x(4)$	100	001
$g_2(0) = x(2)$	010	010
$g_2(1) = x(6)$	110	011
$h_1(0) = x(1)$	001	100
$h_1(1) = x(5)$	101	101
$h_2(0) = x(3)$	011	110
$h_2(1) = x(7)$	111	111

ie i/p needs to be given in bit reversed order in DIT FFT algorithm.

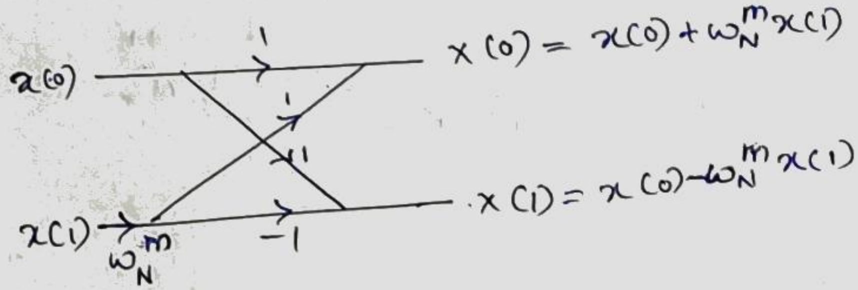
∴



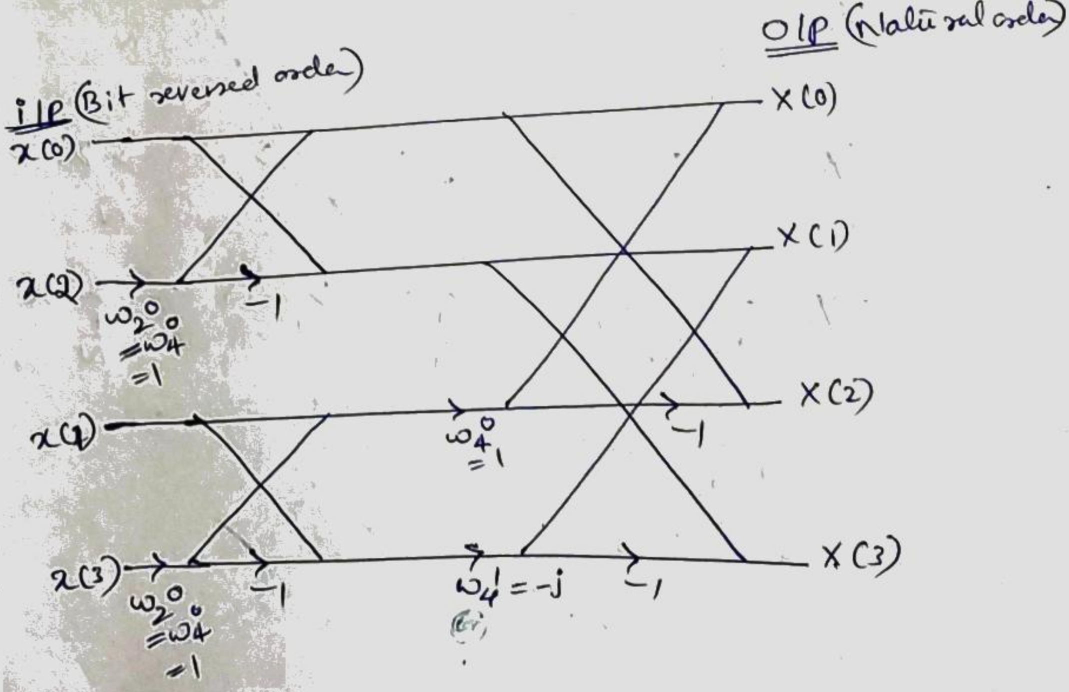
x(0)
x(2)
x(6)
x(1)
x(5)
x(3)
x(7)
ω_2^0
-1
ω_4^0
ω_4^1
ω_8^0
ω_8^3

DIT-FFT algorithm Butterfly structures

For Two-Point computation



For 4-point computation

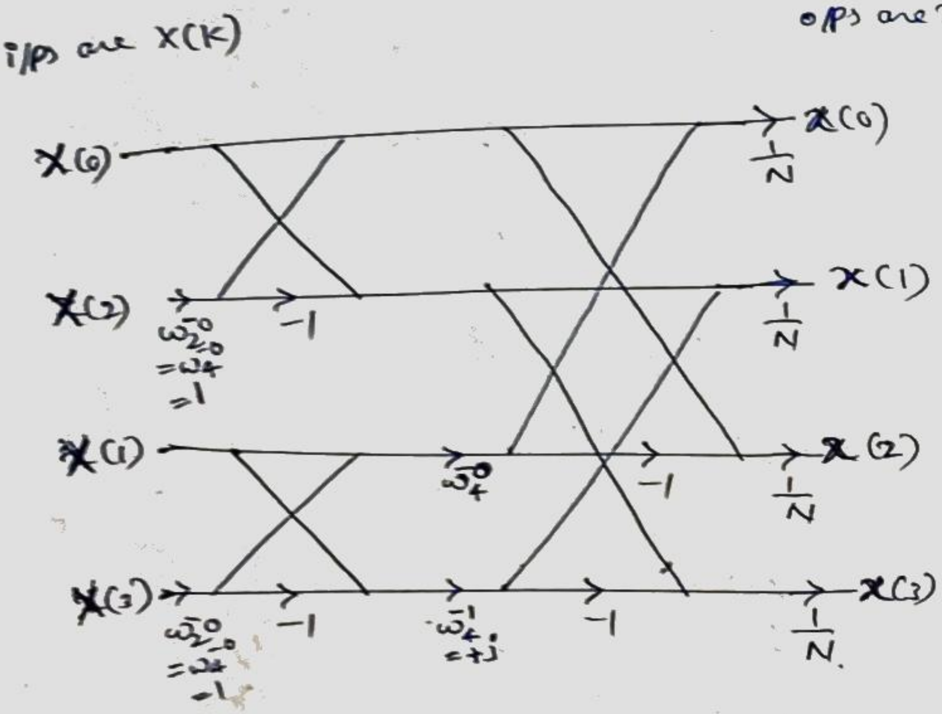


$W_4^0 = 1$

$W_4^1 = -j$

4 Point IDFT Computation

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



$$W_4^{-0} = W_4^{0} = 1$$

$$W_4^{-1} = j$$

$4∞x ╟► ‡@‼Idn.µ¿ỡⁿ●1◎Λ●$●+♥I◎^◎=◎}♥ⁿ●vy₠&@► ♦8¶z►pXtááñÉ¿►@ ►♠@‼↔˜Ç∞}▼

8-Point IDFT computation

* Replace all W_N^m by W_N^{-m} & multiply all o/p lines by $\frac{1}{N}$.
* i/ps are $X(K)$ in bitreversed order
* o/ps are $x(n)$ in natural order.

Number of computations

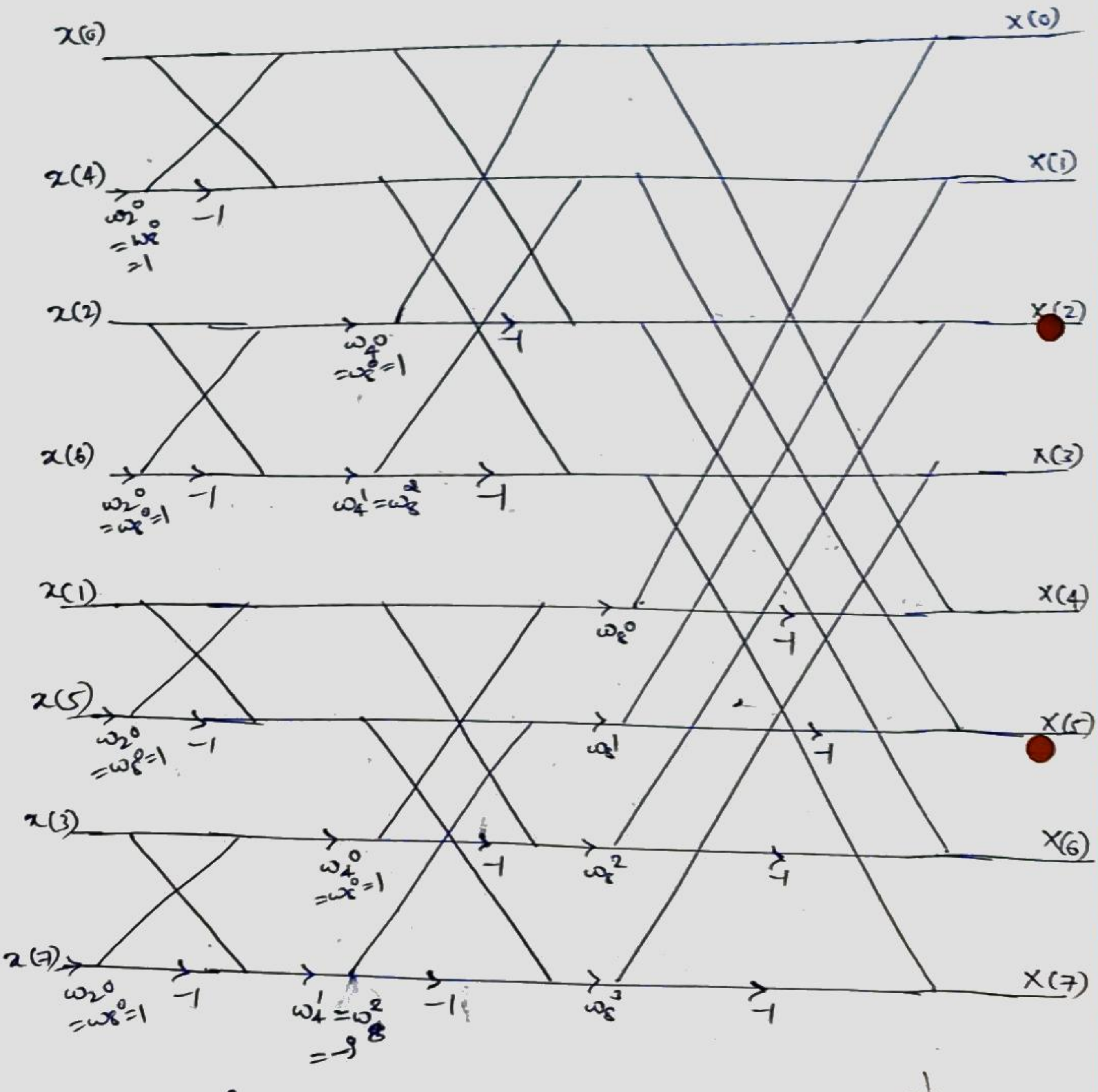
For N-point DFT.

* No of butterflies = No of butterflies in each stage $\times \nu = \frac{N}{2}\nu$
* No of stages $= \nu = \log_2 N$
* No of complex additions in each stage $= N$
* Total no of complex additions $= N \cdot$ no of stages $= N\nu$.
  $= N \log_2 N$
* No of complex multiplications in each stage $= \frac{N}{2}$
* Total number of complex multiplication $= \frac{N}{2}\nu$
  $= \frac{N}{2}\log_2 N$.
* Speed improvement factor.

$$= \frac{\text{No of mults for Direct computation of DFT}}{\text{No of multiplications in FFT algorithm.}}$$

$$= \frac{N^2}{\frac{N}{2}\log_2 N} = \frac{2N}{\log_2 N}$$

8-Point computation

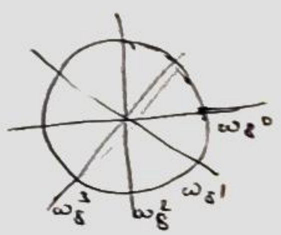


$$W_8^0 = 1$$

$$W_8^1 = +\frac{1}{\sqrt{2}} - \frac{j}{\sqrt{2}} = 0.7071 - j0.7071$$

$$W_8^2 = -j$$

$$W_8^3 = -\frac{1}{\sqrt{2}} - \frac{j}{\sqrt{2}} = -0.7071 - j0.7071$$



Speed improvement [illegible] total number of expressed in terms of total number of computations as

Speed improvement factor =

$$\frac{\text{Total No of computations using DFT}}{\text{Total no of computations using FFT}}$$

$$= \frac{(\text{No of additions} + \text{No of multiplications}) \text{ using DFT}}{(\text{No of additions} + \text{No of multiplications}) \text{ using FFT}}$$

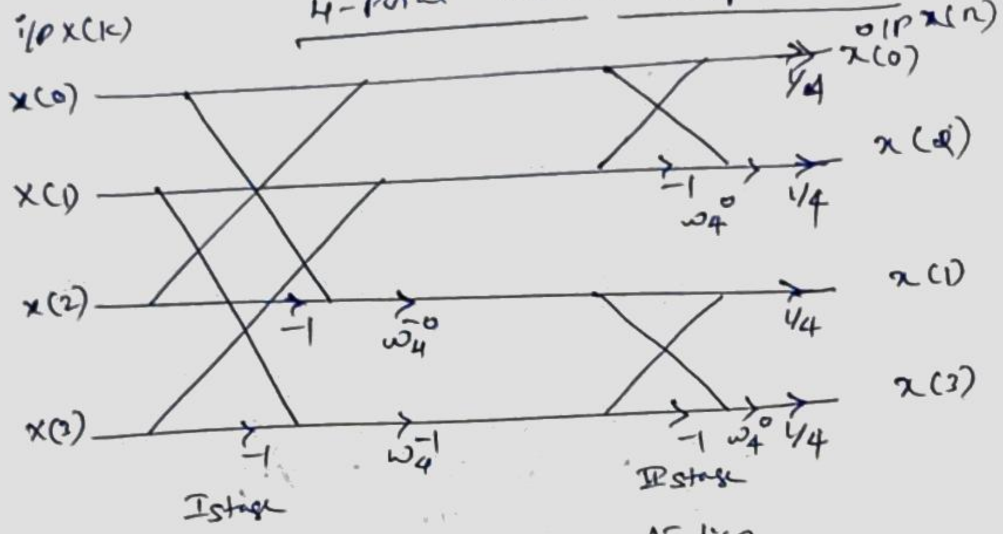
$$= \frac{N(N-1) + N^2}{N \log_2 N + \frac{N}{2} \log_2 N}$$

$$= \frac{N[2N-1]}{N\left[\log_2 N + \frac{1}{2}\log_2 N\right]}$$

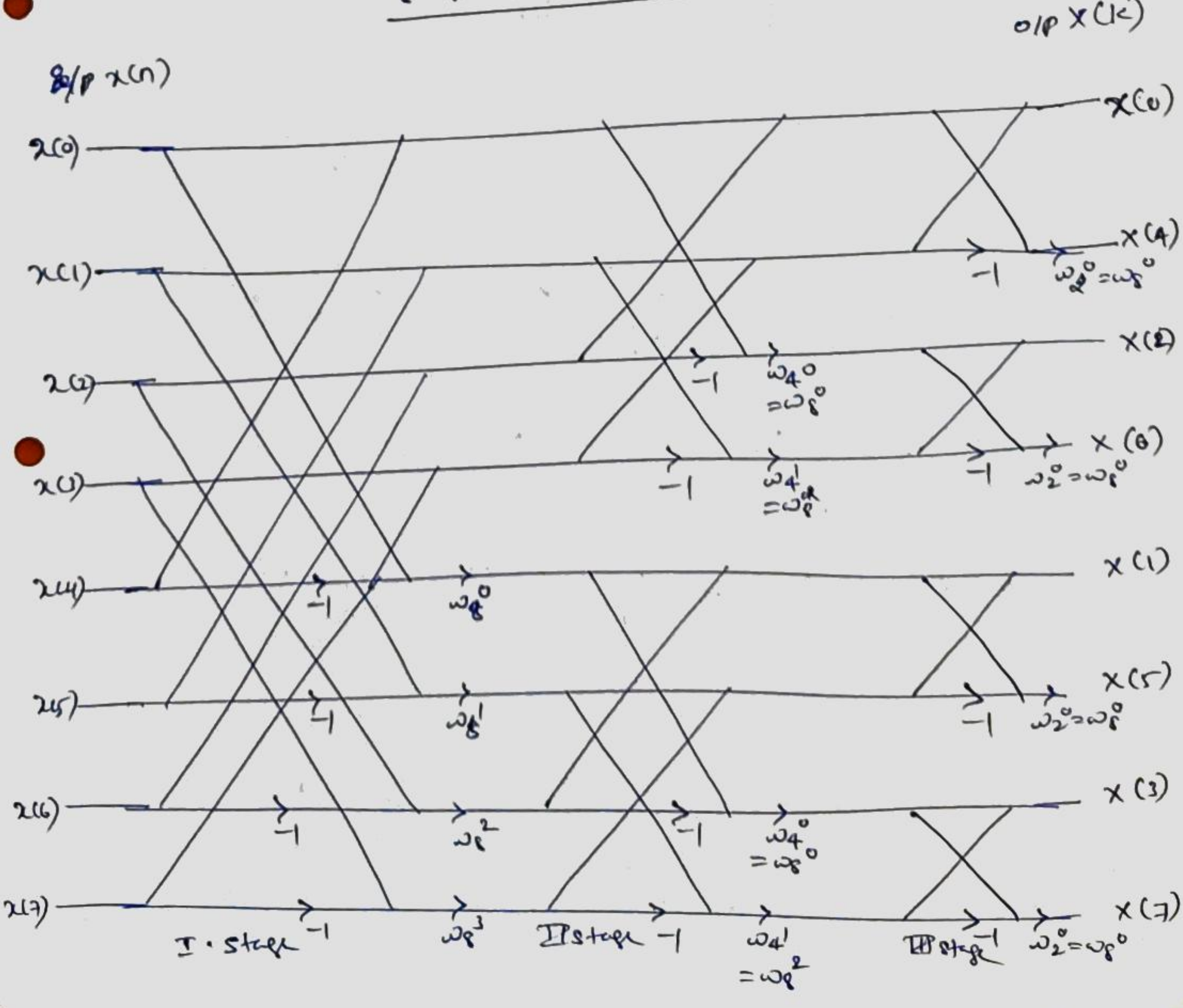
$$= \frac{2N-1}{\frac{3}{2}\log_2 N}$$

$$= \frac{2(2N-1)}{3\log_2 N}$$

4-Point IDFT computation



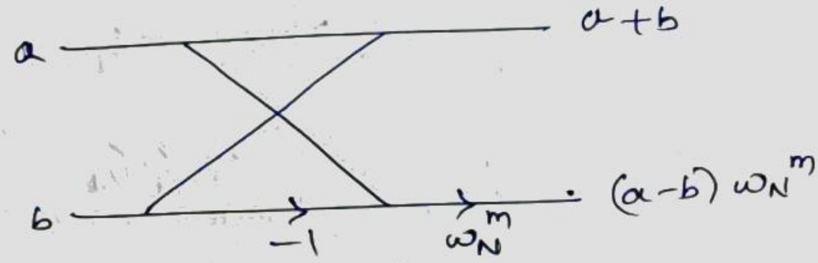
8-point computation



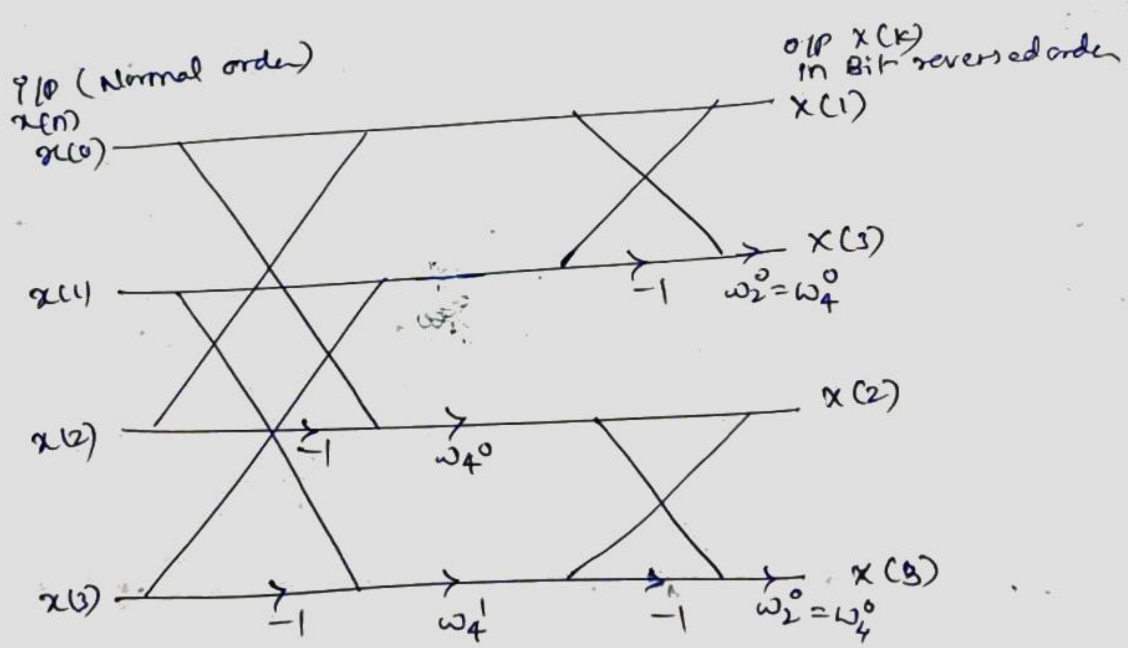
· Radix-2 DIF-FFT Algorithms

<u>Butterfly structures</u>

<u>2-point computation</u>



<u>4-point computation</u>



To find no. of real multiplications & additions in FFT algorithms & Direct computation.

W.K.T One complex multiplication needs
→ 4 real multiplications and
→ 3 real additions.

- One complex addition needs.
→ 2 real additions.

∴ Total number of additions & mults in computing N-DFT using direct method are

mults. → $N^2 \times 4$.

additions → $2 \times N(N-1) + 3N^2$ (in mults) $= 5N^2 - 2N$

∴ Total computations $= 4N^2 + 5N^2 - 2N$

$= 9N^2 - 2N$.

lly using FFT algorithm

$\text{Mults} = 4 \times \frac{N}{2} \log_2 N = 2N \log_2 N$

$\text{adds} = 2 \times N \log_2 N + 3 \frac{N}{2} \log_2 N = \frac{7N}{2} \log_2 N$

∴ speed improvement factor

$$= \frac{9N^2 - 2N}{\frac{7}{2} N \log_2 N}$$

$$= \frac{9N - 2}{\frac{7}{2} \log_2 N} \;//.$$

1) Find 4 point DFT's of using DITFFT algorithm

i) $x_1(n) = \{1 \; -1 \; 2 \; -2\}$

ii) $x_2(n) = \{1/2, 1/3, 3/2, 1/5\}$.

iii) $x_3(n) = \{1, -1\}$

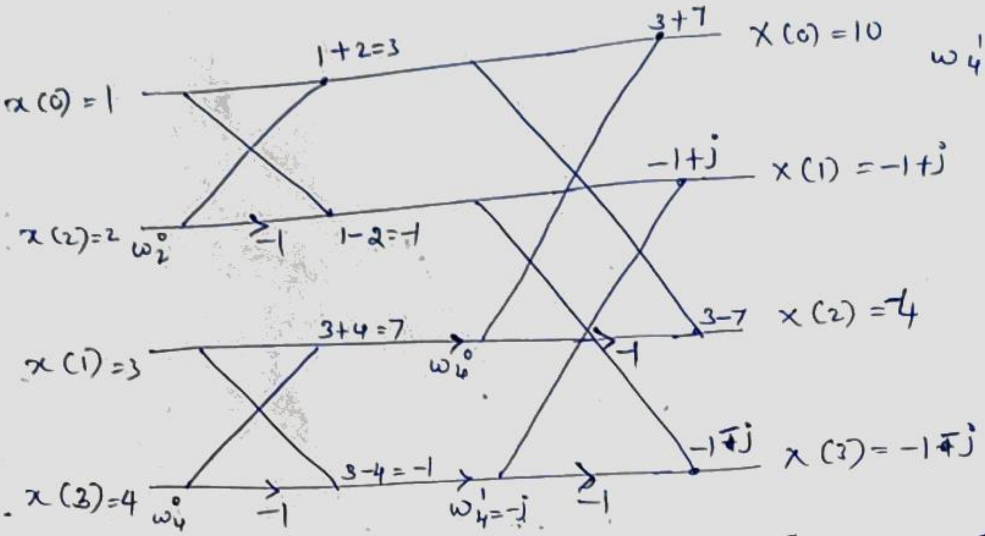
2) Find 8-point DFT of $x(n) = \{1, 2, -1, 2, 3\}$ using DITFFT algorithm and verify by finding IDFT of the same using DIT-IFFT algorithm.

3) Find response of system whose impulse response is $h(n) = \{1 \; 2 \; 1\}$ for the i/p $x(n) = \{1, 2\}$ using DIT-FFT algorithm

Problems on DIT FFT

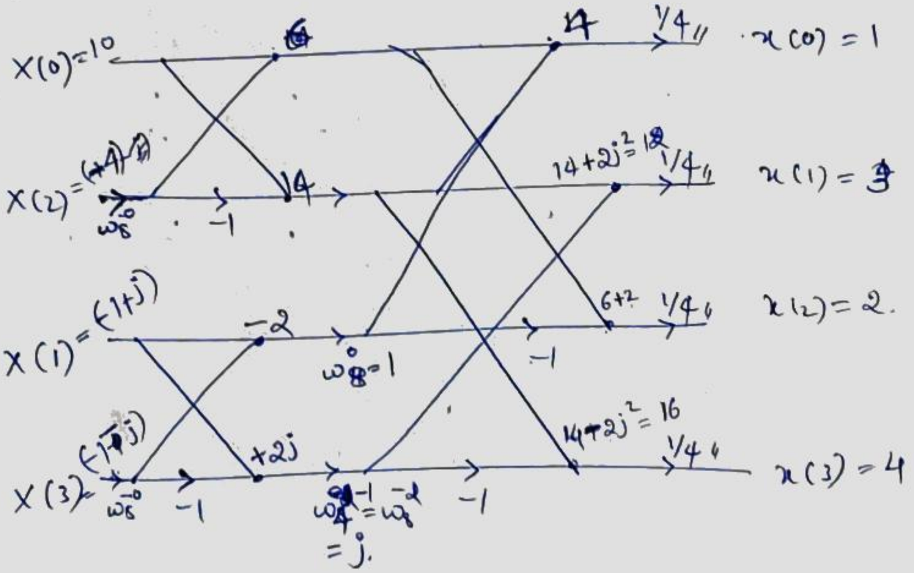
1) Find 4 point DFT of $x(n) = \{1, 3, 2, 4\}$

$N = 4$



$W_4^1 = e^{-j\frac{2\pi}{4}} = e^{-j\frac{\pi}{2}}$
$= -j$

2) Find 4 point IDFT of $X(k) = \{10, -1-j, 4, -1+j\}$



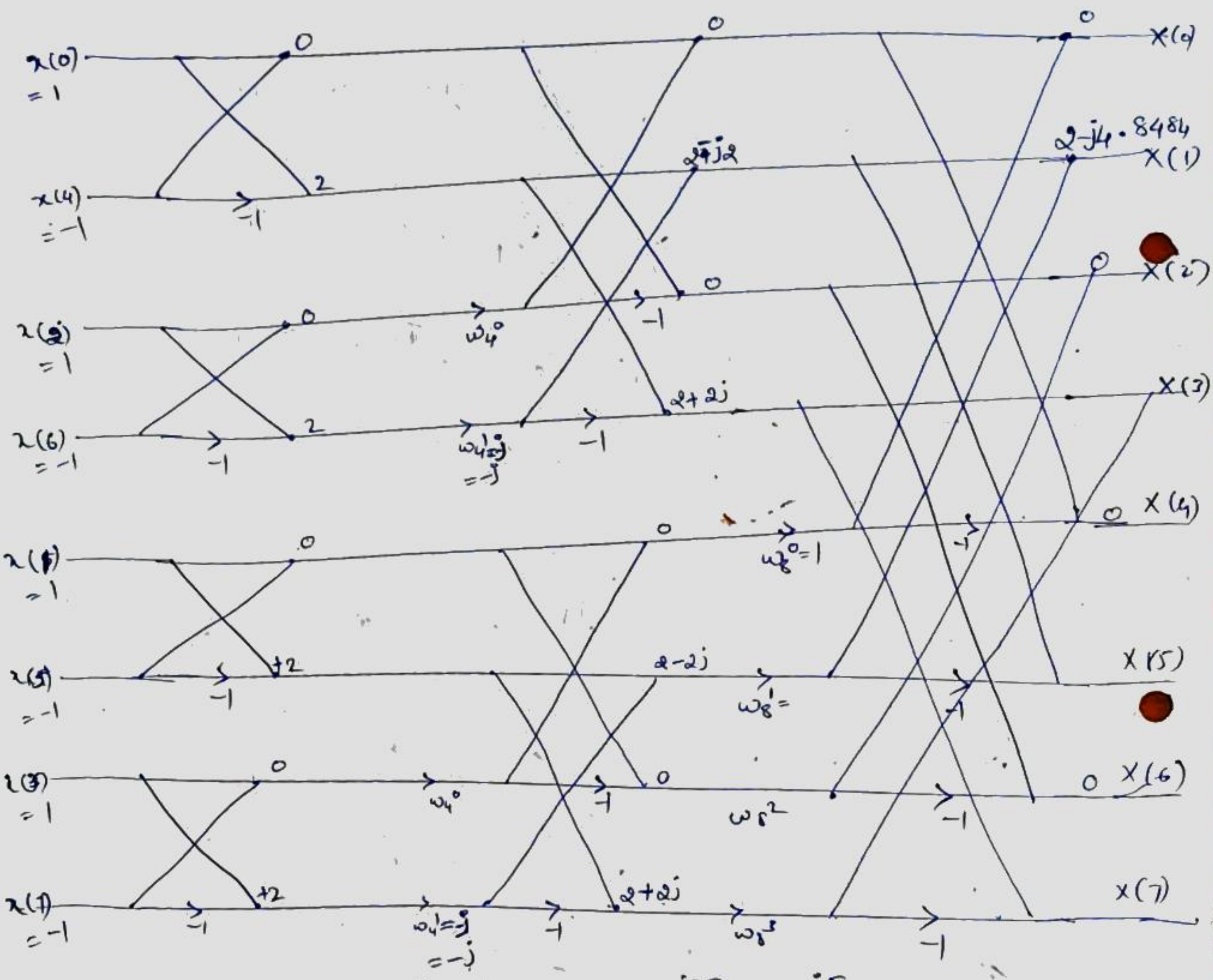
$W_4^{-1} = e^{-j\frac{2\pi}{4}\times -1}$
$= e^{j\frac{\pi}{2}} = 0 + j$
$= j$

$x(n) = \{1 \quad 3 \quad 2 \quad 4\}$

Find 8-point DFT of i) $x(n) = \{1\ 2\ 3\ 4\ 5\ 6\ 7\ 8\}$

ii) $x(n) = \{1, 1, 1, 1, -1, -1, -1, -1\}$

ii) $X(K) = \{0, (2 - j4.8284), 0, (2 \pm 0.8284), 0, (2 \mp j.8284), 0, (2 + j4.8284)\}$



$$X(1) = 2 - j2 + (2 - j2)\left(\frac{1}{\sqrt{2}} - j\frac{1}{\sqrt{2}}\right)$$
$$= 2 - j2 + \sqrt{2} - j\sqrt{2} - j\sqrt{2} - \sqrt{2}$$
$$= 2 - j(2 + 2\sqrt{2})$$
$$= 2 - 4.8484j$$

$$w_8^1 = e^{-j\frac{2\pi}{8}} = e^{-j\frac{\pi}{4}} =$$
$$w_8^2 = e^{-j\frac{\pi}{2}} = -j$$
$$w_8^3 = e^{-j\frac{3\pi}{4}} =$$

MODULE - 2:

FFT Algorithms : USE of DFT in linear filtering, Direct computation of DFT, need for efficient computation of the DFT (FFT algorithms). Radix-2 FFT algorithm for the computation of DFT and IDFT - decimation in-time and decimation-in-frequency algorithms

USE of DFT in linear filtering :- (1)

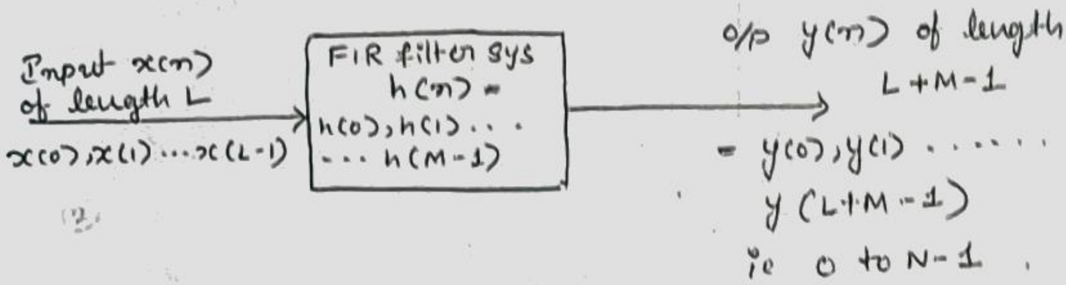
* The DFT provides a discrete frequency representation of a finite-duration sequence in the frequency domain.
  - → ∴ DFT use as a computational tool for linear sys analysis and especially for linear filtering.
* Because of continuous variable "ω", the computations cannot be done on a digital computer
  - → ∵ Computer can only store and perform computations on quantities at discrete frequencies.
* ∴ DFT does lend itself to computation on a digital computer.
  - → DFT can be used to perform linear filtering in the frequency domain.
  - → An alternative to time-domain convolution.
  - → more efficient than time-domain convolution ∵ of efficient algorithms.

* Linear filtering operation is implemented with the help of linear convolution.

* The o/p $y(n)$ is obtained by convolving impulse response $h(n)$ with input $x(n)$

→ Let the unit sample response of the LTI system be $h(n)$ of length "M". ie.
$h(0), h(1) \ldots\ldots h(M-1)$

→ i/p to the LTI system be $x(n)$ of length "L"
ie $x(0), x(1), x(2) \ldots\ldots x(L-1)$



$$\therefore y(n) = \sum_{K=-\infty}^{\infty} h(K)\,x(n-K) \longrightarrow (1)$$

$y(n)$ o/p of LTI sys, $y(n)$ is a linear convolution of $h(n)$ & $x(n)$

→ Consider the Fourier transform of the Eqⁿ (1)

$$Y(\omega) = F\left[\sum_{K=-\infty}^{\infty} h(K)\,x(n-K)\right] \longrightarrow (2)$$

→ w.k.t convolution property of Fourier transform is

$$F\{x_1(n) * x_2(n)\} = X_1(\omega)\,X_2(\omega)$$

$$\therefore Y(\omega) = H(\omega)\cdot X(\omega) \longrightarrow (3)$$

W k t $Y(K)$ can be obtained from $Y(\omega)$ as

$$Y(K) = Y(\omega)\Big|_{\omega = \frac{2\pi K}{N}} \quad , \quad K = 0,1,2 \cdots N-1$$

$$\therefore \; Y(K) = X(\omega)\cdot H(\omega)\Big|_{\omega = \frac{2\pi K}{N}} \quad , \quad K = 0,1,2 \cdots N-1$$

since $X(\omega)\Big|_{\omega = \frac{2\pi K}{N}} = X(K)$

$$H(\omega)\Big|_{\omega = \frac{2\pi K}{N}} = H(K)$$

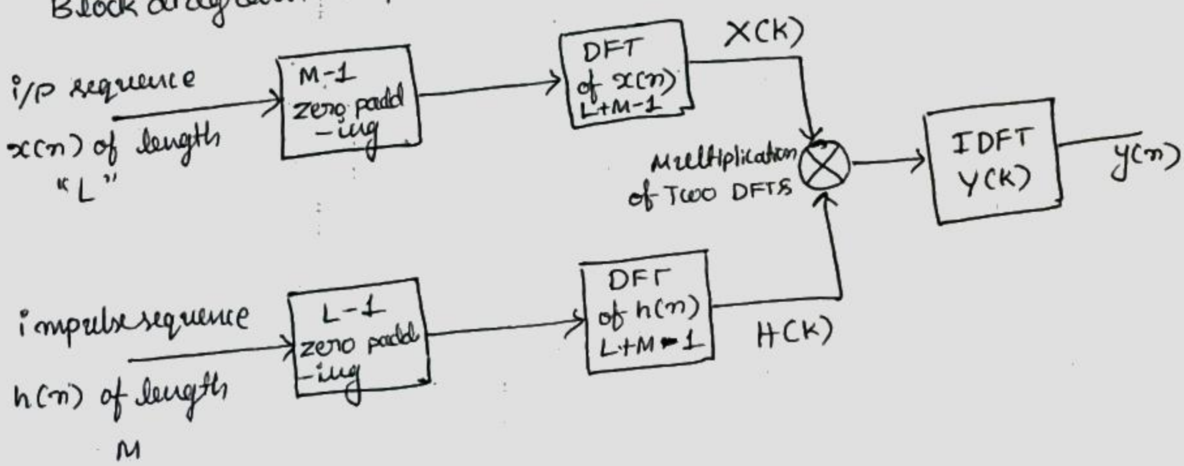
$$\therefore \; Y(K) = X(K)\cdot H(K) \qquad K = 0,1,2 \cdots N-1$$

* Multiplying "N" point DFT of $x(n)$ & $h(n)$, we get DFT $Y(K)$. This DFT represent $y(n)$ uniquely if $N \geq L+M-1$. Hence $y(n)$ can be obtained by taking IDFT of $Y(K)$

$$y(n) = \text{IDFT}\,\{Y(K)\}$$
$$= \text{IDFT}\,\{X(K)\cdot H(K)\}$$

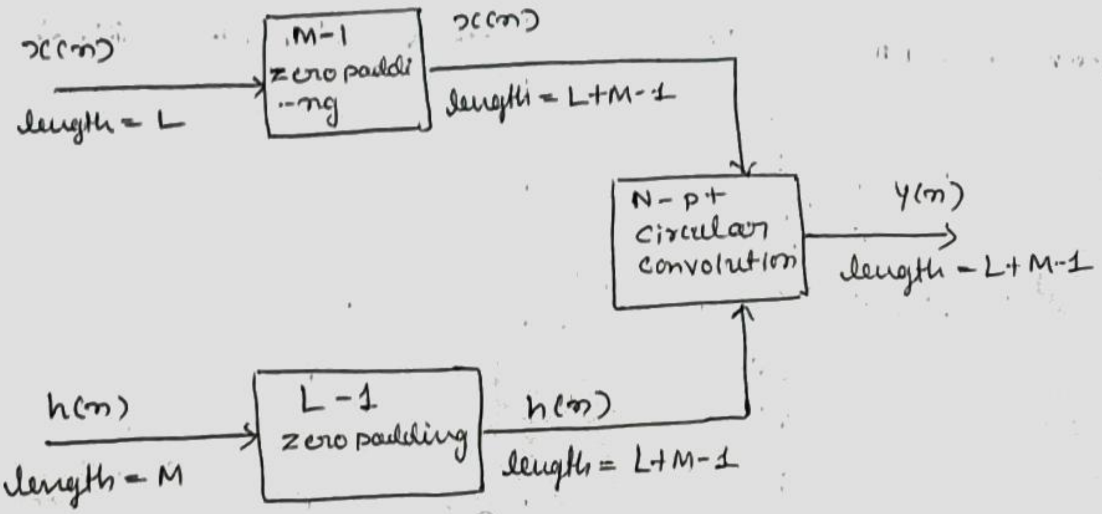
* Linear convolution can be implemented using DFT block diagram.

Block diagram representation by block diagram.



* Linear convolution implemented by circular convolution

$$y(n) = x(n) \circledast_N h(n) \Rightarrow X(K) \cdot H(K)$$



Linear filtering of long sequences

We know that the response of LTI sym is obtained by convolution of i/p and impulse response of the sym. ie $y(n) = x(n) * h(n)$

If any one of the ($x(n)$ or $h(n)$) is very much longer than the other, then it is very difficult to compute the linear convolution due to following reasons

1) The entire sequence should be available before convolution is performed, which introduces delay in getting the output

2) Large amounts of memory is required to store the sequences.

The above probs can be over come by dividing linear filtering of long seq. in to smaller sequences, as follows

1) The long seq is devided in to smaller sequences
2) Linear convolution of each section smaller sequence is performed
3) The output sequences convolutions are combined to get over all o/p seq.

There are two methods to achieve it

1) overlap add method
2) overlap save method.

Overlap - Add method

→ Let N_1 & N_2 be lengths of longer & smaller sequences respectively.

→ Let longer seq be devided in to sections of N_3 samples each (Normally the longer seq is devided in to sections of size same as that of smaller seq)

→ Perform linear convolution of each of sections with smaller seq. to produce N_2+N_3-1 length sequences

→ In this method the last N_2-1 samples of each output seq overlap with the first N_2-1 samples of next section.

→ Corresponding samples of overlapped region are added and non overlapped samples are retained to get final convolved seq of length N_1+N_2-1.

Ex. Find linear convolution of

$x(n) = \{1\ \ -1\ \ 2\ \ -2\ \ 3\ \ -3\ \ 4\ \ -4\}$

$h(n) = \{-1\ \ 1\}$

Sol: 1) here $N_1 = 8$, $N_2 = 2$

Let $N_3 = 3$.

2) Then $x_1(n) = \{1\ -1\ \ 2\}$ $\quad x_2(n) = \{-2\ \ 3\ -3\}$ $\quad \cancel{x_3(n) = \{\ \}}$

$x_3(n) = \{4\ \ -4\ \ 0\}$

3) Perform linear conv of $x_1(n)$ & $h(n)$

$y_1(n) = \{-1\ \ 2\ \ -3\ \ 2\} = x_1(n) * h(n)$

$y_2(n) = \{2\ \ -5\ \ 6\ \ -3\} = x_2(n) * h(n)$

$y_3(n) = \{-4\ \ 8\ \ -4\ \ 0\}$

4) Now $y(n)$ is obtained by writing ... as follows.

$y_3(n)$										
n	0	1	2	3	4	5	6	7	8	9
$y_1(n)$	-1	2	-3	2						
				2	-5	6	-3			
$y_2(n)$							-4	8	-4	0
$y_3(n)$										
$y(n)$	-1	2	-3	4	-5	6	-7	8	-4	0

$\therefore y(n) = \{-1 \quad 2 \quad -3 \quad 4 \quad -5 \quad 6 \quad -7 \quad 8 \quad -4\}$.

ex 2) Find $y(n)$ if $x(n) = \{1 \quad 2 \quad 3 \quad 4 \quad 5\}$ & $h(n) = \{1 \quad -1\}$ devide seq $x(n)$ in to 3 length seq

sol 1) Let $N_3 = 3$.

$\therefore x_1(n) = \{1 \quad 2 \quad 3\}$

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

2) $y_1(n) = x_1(n) * h(n)$

$= \{1 \quad 1 \quad 1 \quad -3\}$

	1	2	3	
1	1	2	3	
-1	-1	-2	-3	
	1	1	1	-3

3) $y_2(n) = x_2(n) * h(n)$

$= \{4 \quad 1 \quad -5 \quad 0\}$

	4	5	0	
1	4	5	0	
-1	-4	-5	0	
	4	1	-5	0

4) Find $y(n)$

Combine $y_1(n)$ & $y_2(n)$

n	0	1	2	3	4	5	6
$y_1(n)$	1	1	1	-3			
				4	1	-5	0
$y_2(n)$							
$y(n)$	$\{1$	1	1	1	1	-5 $\}$.	

Verification

	1	2	3	4	5
1	1	2	3	4	5
-1	-1	-2	-3	-4	-5

$\{1 \quad 1 \quad 1 \quad 1 \quad 1 \quad -5\}$ //.

eg 3: Find y(n) using DIT-FFT.

Overlap-save method.

Let N_1, N_2 & N_3 be lengths of longer, smaller & devided subseq resply.

1) In overlap save method result of linear convoluti-on is obtained by circular convolution method. hence each ~~seq~~ of smaller sub seq(s) and smaller (N_2) seq are converted to the length of (N_2+N_3-1) by padding (zeros)/ elements as follows.

i) The first (N_2-1) samples of a section is appended (padded) as last N_2-1 samples of previous section

or

The last (N_2-1) samples of a section are appended at last (N_2-1) samples of next section.

ii) The smaller seq is converted to (N_2+N_3-1) by padding (N_3-1) zeros.

2) Circular convolution of modified seq is perform-ed.

3) i) first (N_2-1) samples of circular conv are discard-ed & remaining are saved as final o/p

ii) The last (N_2-1) samples are discarded.

Solve ex using ○lar convolution

The problem can be
Find response of system with impulse response $h(n) = \{1 \ -1\}$ for the long sequence $x(n)$ given by $x(n) = \{1 \ 2 \ 3 \ 4 \ 5\}$. Use 4 point circular convolution for computation of convolution using overlap-add method.

Sol: Here $N_1 = 5$, $N_2 = 2$.

given $N_3 = ?$

To find N_3

given $L_1 = L_c = 4$ given

$\therefore L_c = N_2 + N_3 - 1$

ie $N_3 = L_c - N_2 + 1 = 4 - 2 + 1 = 3$

$\therefore$ Divide $x(n)$ in to subsequences of length 3 each.

ie $x_1(n) = \{1 \ 2 \ 3\}$ $\quad h(n) = \{1 \ -1\}$

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

* To find $y_1(n)$ using 4 point ○lar convol

$x_1(n) = \{1 \ 2 \ 3 \ 0\}$ $\quad h(n) = \{1 \ -1 \ 0 \ 0\}$

$$y_1(n) = \begin{bmatrix} 1 & 0 & 0 & -1 \\ -1 & 1 & 0 & 0 \\ 0 & -1 & 1 & 0 \\ 0 & 0 & -1 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 2 \\ 3 \\ 0 \end{bmatrix}$$

$y_1(n) = [h((n-k))_N][x_1(n)]$

ie $y_1(n) = [1 \ 1 \ 1 \ -3]$

* To find $y_2(n)$ using 4 point ○lar conv.

$x_2(n) = [4 \ 5 \ 0 \ 0]$ $\quad h(n) = [1 \ -1 \ 0 \ 0]$

$y_2(n) = [h((n-k))_N][x_2(n)]$

$$y_2(n) = \begin{bmatrix} 1 & 0 & 0 & -1 \\ -1 & 1 & 0 & 0 \\ 0 & -1 & 1 & 0 \\ 0 & 0 & -1 & 1 \end{bmatrix} \begin{bmatrix} 4 \\ 5 \\ 0 \\ 0 \end{bmatrix}$$

i.e. $y_2(n) = [4 \;\; 1 \;\; -5 \;\; 0]$

To find $y(n)$ using $y_1(n)$ & $y_2(n)$.

	0	1	2	3	4	5	6	7
$y_1(n)$	1	1	1	-2				
$y_2(n)$				4	-1	-5	0	
	1	1	1	1	$+1$	-5	0	

i.e. $y(n) = \{1, 1, 1, 1, 1, -5, 0\}$