

UNIT II

Frequency Analysis of Signals and Systems-II



DISCRETE FOURIER TRANSFORM (DFT)

- The DFT provides uniformly spaced samples of the Discrete-Time Fourier Transform (DTFT)
- DFT definition:

$$X[k] = \sum_{n=0}^{N-1} x[n] e^{-j \frac{2\pi nk}{N}} \quad x[n] = \frac{1}{N} \sum_{k=0}^{N-1} X[k] e^{j \frac{2\pi nk}{N}} \rightarrow \text{CB, TB}$$

- Requires N^2 complex multiplications and $N(N-1)$ complex additions

$$N=4, N=8$$



FASTER DFT COMPUTATION?

- Take advantage of the symmetry and periodicity of the complex exponential (let $W_N = e^{-j2\pi/N}$)
 - **symmetry:** $W_N^{k[N-n]} = W_N^{-kn} = (W_N^{kn})^*$
 - **periodicity:** $W_N^{kn} = W_N^{k[n+N]} = W_N^{[k+N]n}$
- Note that two length $N/2$ DFTs take less computation than one length N DFT: $2(N/2)^2 < N^2$
- Algorithms that exploit computational savings are collectively called *Fast Fourier Transforms*



DFT AND FFT

- FFT is an algorithm to convert a time domain signal to DFT efficiently.
- FFT is not unique. Many algorithms are available.
- Each algorithm has merits and demerits.
- In each algorithm, depending on the sequence needed at the output, the input is regrouped.
- The groups are decided by the number of samples.
- Algorithms having number of samples 2^N , where N is an integer is most preferred.
- Radix-x: here 'x' represents number of samples in each group made at the first stage. They are generally equal.
- We shall study radix-2 ALGORITHM.



RADIX-2: DIT OR, DIF

- Radix-2 is the first FFT algorithm. It was proposed by Cooley and Tukey in 1965.
- Though it is *not* the efficient algorithm, it lays foundation for time-efficient DFT calculations.
- The algorithms appear either in
 - (a) **Decimation In Time (DIT)**, or,
 - (b) **Decimation In Frequency (DIF)**.
- DIT and DIF, both yield same complexity and results. They are complementary.
- We shall stress on 8 to radix 2 DIT FFT.



OTHER POPULAR ALGORITHMS

Besides many, the popular algorithms are:

- Goertzel algorithm
- Chirp Z algorithm
- Index mapping algorithm
- Split radix in prime number algorithm.



FFT COMPUTATIONAL EFFICIENCY

| Number of Data Points N | Direct Computation | | FFT algorithm | |
|----------------------------|---|--------------------------------------|---|---|
| | <i>Complex Multiplications</i> N^2 | <i>Complex Additions</i> $N(N-1)$ | <i>Complex Multiplications</i> $(N/2)\log_2 N$ | <i>Complex Additions</i> $N\log_2 N$ |
| 16 | 256 | 240 | 32 | 64 |
| 64 | 4096 | 4032 | 192 | 384 |
| 128 | 16384 | 16256 | 448 | 896 |
| 256 | 65536 | 65280 | 1024 | 2048 |
| 512 | 262144 | 261632 | 2304 | 4608 |
| 1024 | 1048576 | 1047552 | 5120 | 10240 |
| 2048 | 4194304 | 4192256 | 11264 | 22528 |
| 4096 | 16777216 | 16773120 | 24576 | 49152 |
| 8192 | 67108864 | 67100672 | 53248 | 106496 |



DECIMATION-IN-TIME ALGORITHM (DIT-FFT)

| N | Direct computation | | DIT FFT algorithm | | Improvement in processing speed for multiplication |
|-----|---------------------------------|-------------------------------|--|----------------------------------|--|
| | Complex Multiplication N^2 | Complex Addition $N^2 - N$ | Complex Multiplication $N/2 \log_2 N$ | Complex Addition $N \log_2 N$ | |
| 8 | 64 | 52 | 12 | 24 | 5.3 times |
| 16 | 256 | 240 | 32 | 64 | 8 times |
| 256 | 65536 | 65280 | 1024 | 2048 | 64 times |



FFT

- A fast Fourier transform (FFT) is an algorithm that samples a signal over a period of time (or space) and divides it into its frequency components. An FFT computes the DFT and produces exactly the same result as evaluating the DFT definition directly; the most important difference is that an FFT is much faster

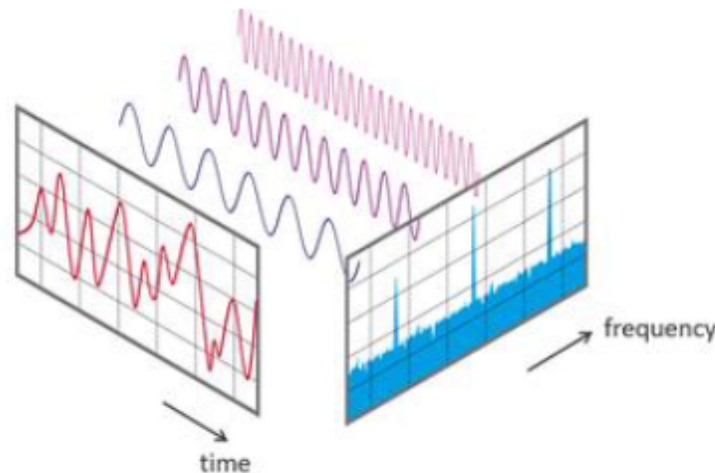


Figure 1. View of a signal in the time and frequency domain



THE PROCESS OF DECIMATION

$$16 \begin{matrix} < \\ 8 \\ < \\ 8 \end{matrix}$$

- First step of process of decimation is splitting a sequence in smaller sequences.
- A sequence of 15 can be splitted in five sequences of threes or three sequences of fives.
- A sequence of 16 numbers can be splitted in 2 sequences of 8.
Further,
 - ✓ Each sequence of 8 can be splitted in two sequences of 4;
 - ✓ Subsequently each sequence of 4 can be splitted in two sequences of two;
 - ✓ There can be various combinations and varied complexities.



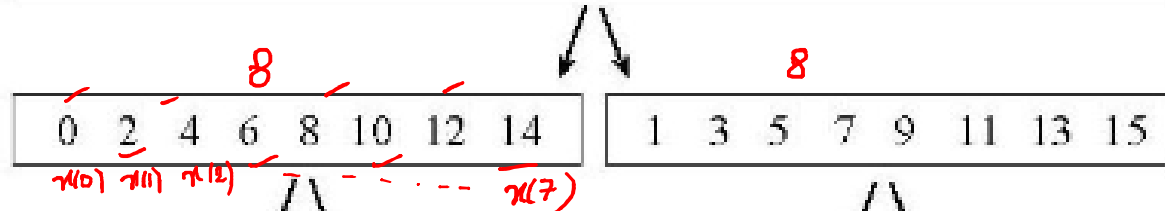
DECIMATION

$x(0), x(4), \dots$

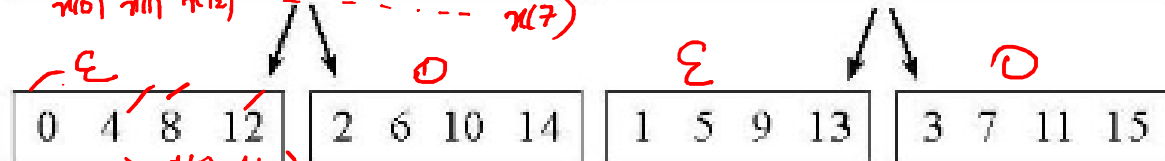
1 signal of 16 points



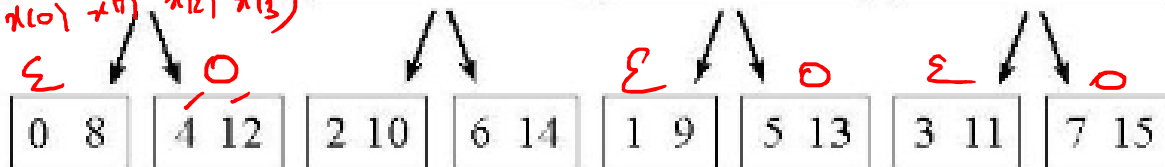
2 signals of 8 points



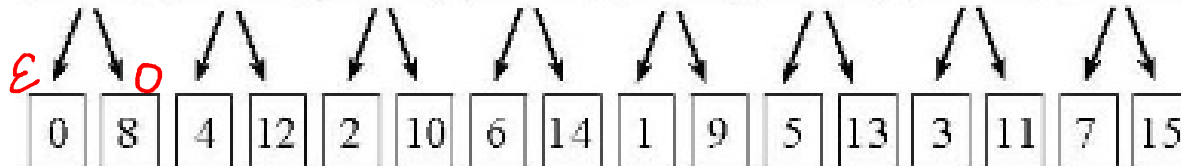
4 signals of 4 points



8 signals of 2 points



16 signals of 1 point



DECIMATION-IN-TIME FFT ALGORITHMS –RADIX-2

- Makes use of both periodicity and symmetry
- Consider special case of N an integer power of 2
- Separate $x[n]$ into two sequence of length $N/2$
 - Even indexed samples in the first sequence
 - Odd indexed samples in the other sequence

$$\begin{aligned} X[k] &= \sum_{n=0}^{N-1} x[n] e^{-j(2\pi/N)kn} \\ &= \sum_{\substack{n \text{ even} \\ \text{Dr.S.KALAIIVANI}}} x[n] e^{-j(2\pi/N)kn} + \sum_{\substack{n \text{ odd} \\ =}} x[n] e^{-j(2\pi/N)kn} \end{aligned}$$



DECIMATION-IN-TIME FFT ALGORITHMS

$$N < \begin{matrix} N/2 \\ N/2 \end{matrix}$$

$$X[k] = \sum_{\substack{n \text{ even} \\ =}} x[n] e^{-j(2\pi/N)kn} + \sum_{\substack{n \text{ odd} \\ =}} x[n] e^{-j(2\pi/N)kn}$$

◆ Substitute variables $n=2r$ for n even and $n=2r+1$ for odd

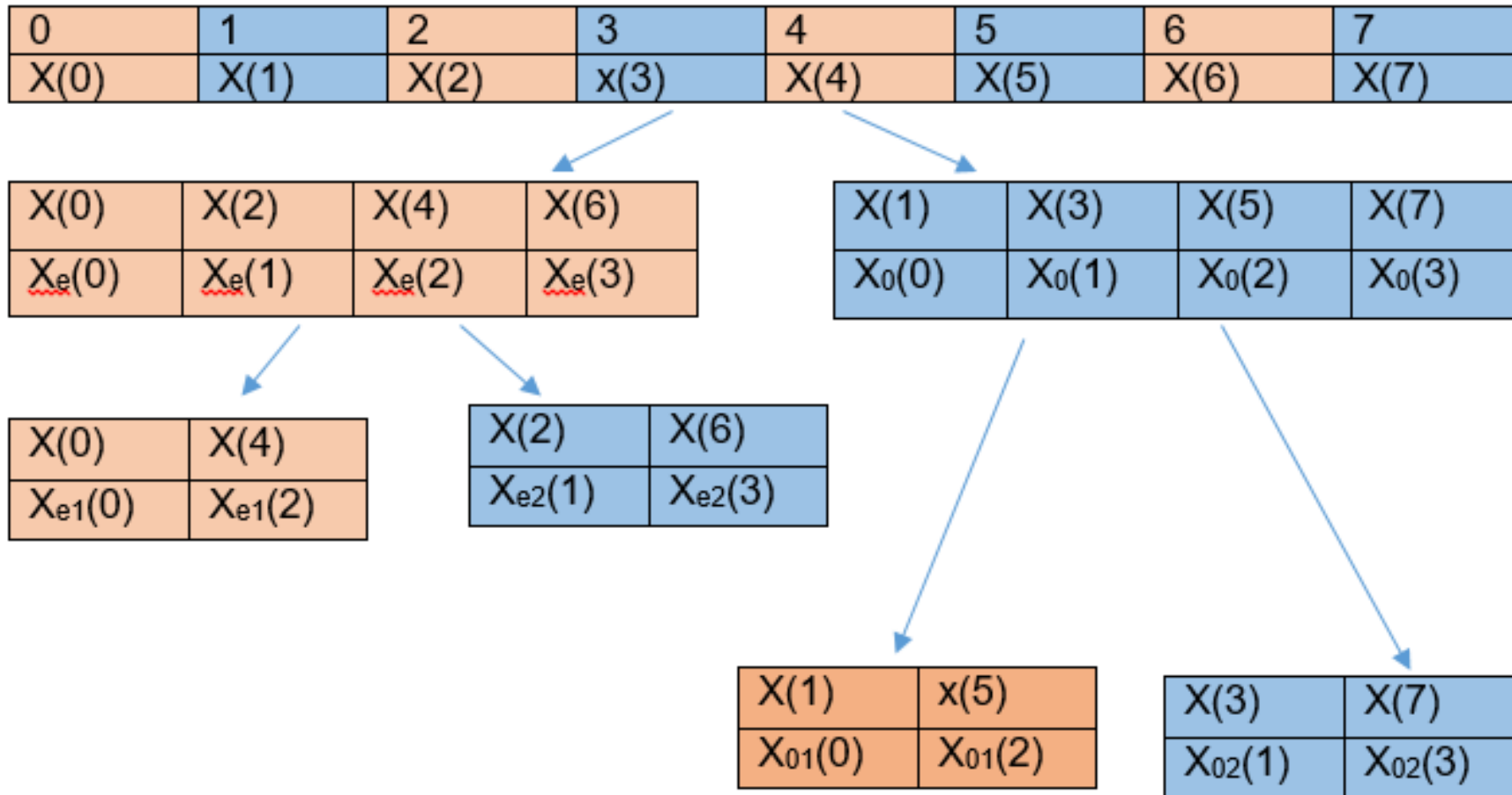
$$X[k] = \sum_{r=0}^{N/2-1} x[2r] W_N^{2rk} + \sum_{r=0}^{N/2-1} x[2r+1] W_N^{(2r+1)k}$$

$$= \sum_{r=0}^{N/2-1} x[2r] W_{N/2}^{rk} + W_N^k \sum_{r=0}^{N/2-1} x[2r+1] W_{N/2}^{rk}$$

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

◆ $G[k]$ and $H[k]$ are the $N/2$ -point DFT's of each subsequence





DECIMATION-IN-TIME FFT ALGORITHMS

$$X[k] = \sum_{r=0}^{N/2-1} x[2r]W_{N/2}^{rk} + W_N^k \sum_{r=0}^{N/2-1} x[2r+1]W_{N/2}^{rk}$$

$$= G[k] + W_N^k H[k] \quad e^{-j\frac{2\pi}{N}2rk} = e^{-j\frac{2\pi}{N/2}rk} = W_{N/2}^{rk}$$

$$k = 0, 1, \dots, \frac{N-1}{2} \quad \longrightarrow \quad k = 0, 1, \dots, N$$

$$G\left[k + \frac{N}{2}\right] = G[k] \quad H\left[k + \frac{N}{2}\right] = H[k]$$

◆ G[k] and H[k] are the N/2-point DFT's of each subsequence

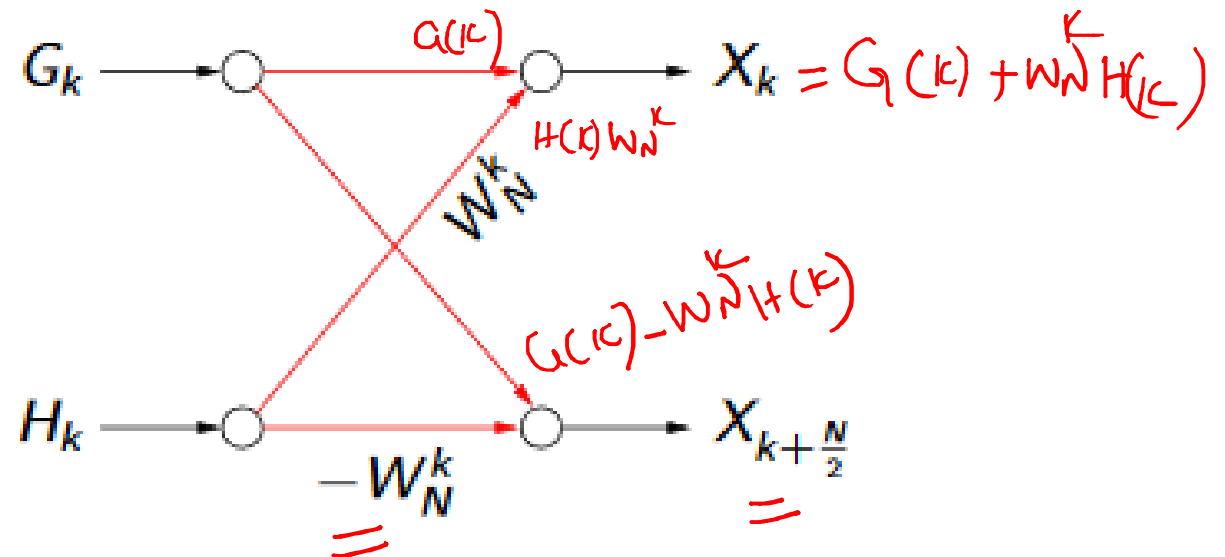


DECIMATION-IN-TIME FFT ALGORITHMS

- The $N/2$ point DFTs $\{G_k\}$ and $\{H_k\}$ are periodic with period $N/2$
 - $G_{k+\frac{N}{2}} = G_k$
 - $H_{k+\frac{N}{2}} = H_k$
- $W_N^{k+\frac{N}{2}} = -W_N^k$
- Hence, if $X_k = G_k + W_N^k H_k$, then $X_{k+\frac{N}{2}} = G_k - W_N^k H_k$
 - $W_N^k H_k$ needs to be computed only once for $k = 0$ to $\frac{N}{2} - 1$
- Thus, the multiplication overhead due to the twiddle factors is only $\frac{N}{2}$



DECIMATION-IN-TIME FFT ALGORITHMS



- $X_k = G_k + W_N^k H_k$
- $X_{k+\frac{N}{2}} = G_{k+\frac{N}{2}} + W_N^{k+\frac{N}{2}} H_{k+\frac{N}{2}}$
 $= G_k - W_N^k H_k$

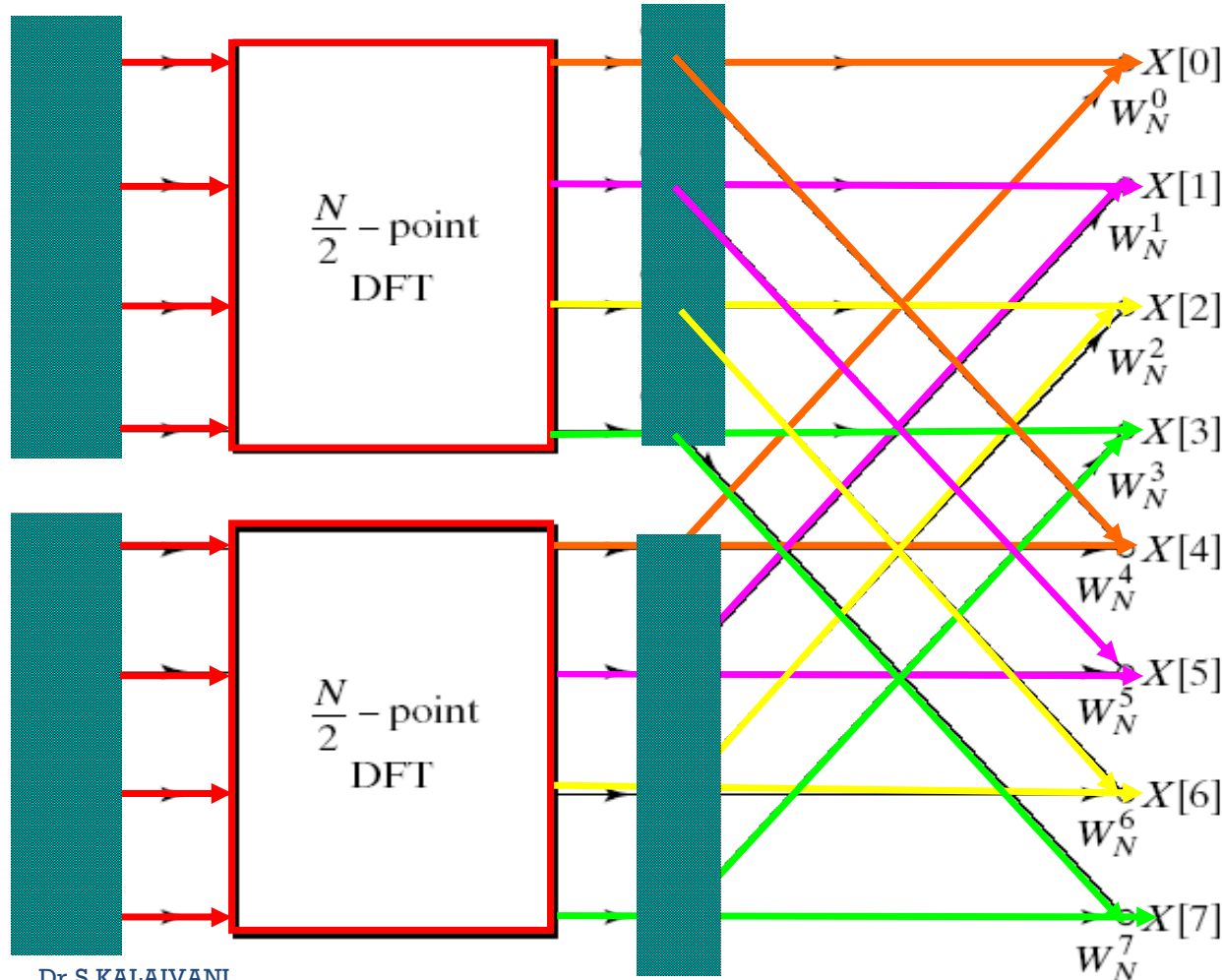


8-POINT DFT USING DECIMATION-IN-TIME

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

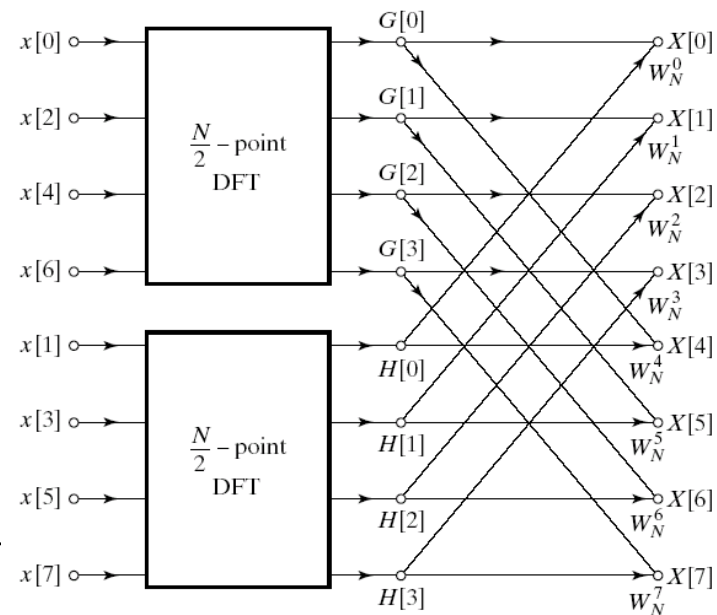
$$X[k] = G[k] + W_N^k H[k]$$

$$X[k+N/2] = G[k] - W_N^k H[k]$$



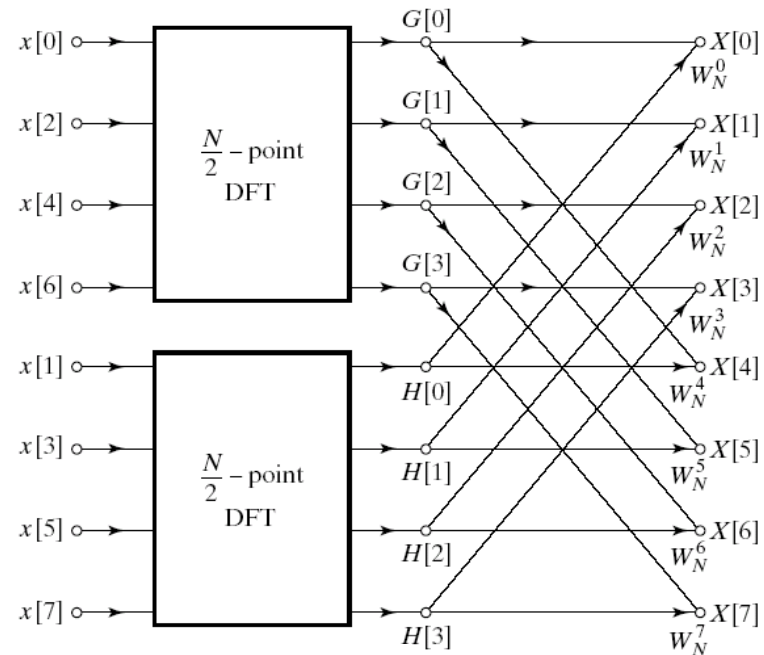
COMPUTATIONAL COMPLEXITY

- Two $N/2$ -point DFTs
 - $2(N/2)^2$ complex multiplications
 - $2(N/2)^2$ complex additions
- Combining the DFT outputs
 - N complex multiplications
 - N complex additions
- Total complexity
 - $N^2/2+N$ complex multiplication
 - $N^2/2+N$ complex additions
 - More efficient than direct DFT

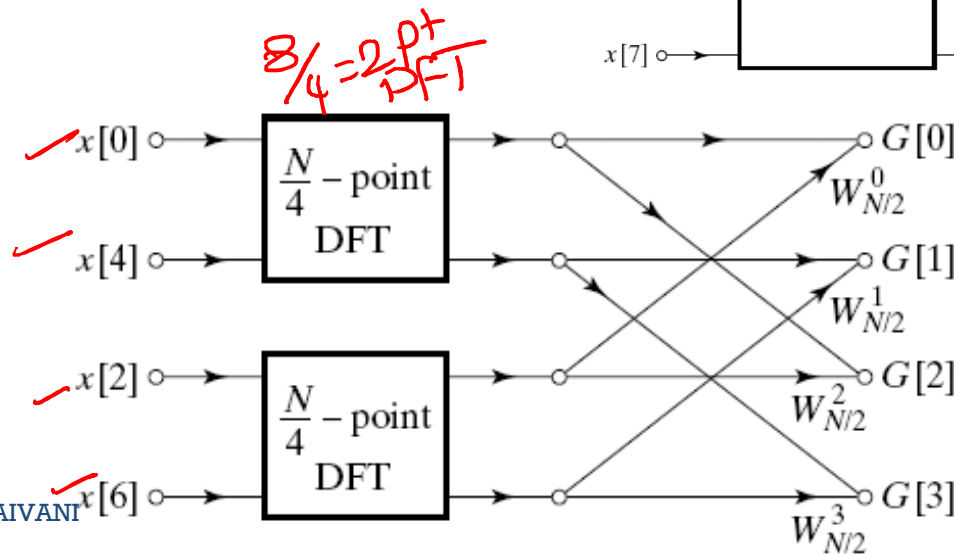


DECIMATION-IN-TIME FFT ALGORITHMS

- Repeat same process ,
Divide $N/2$ -point DFTs into
 - Two $N/4$ -point DFTs
 - Combine outputs

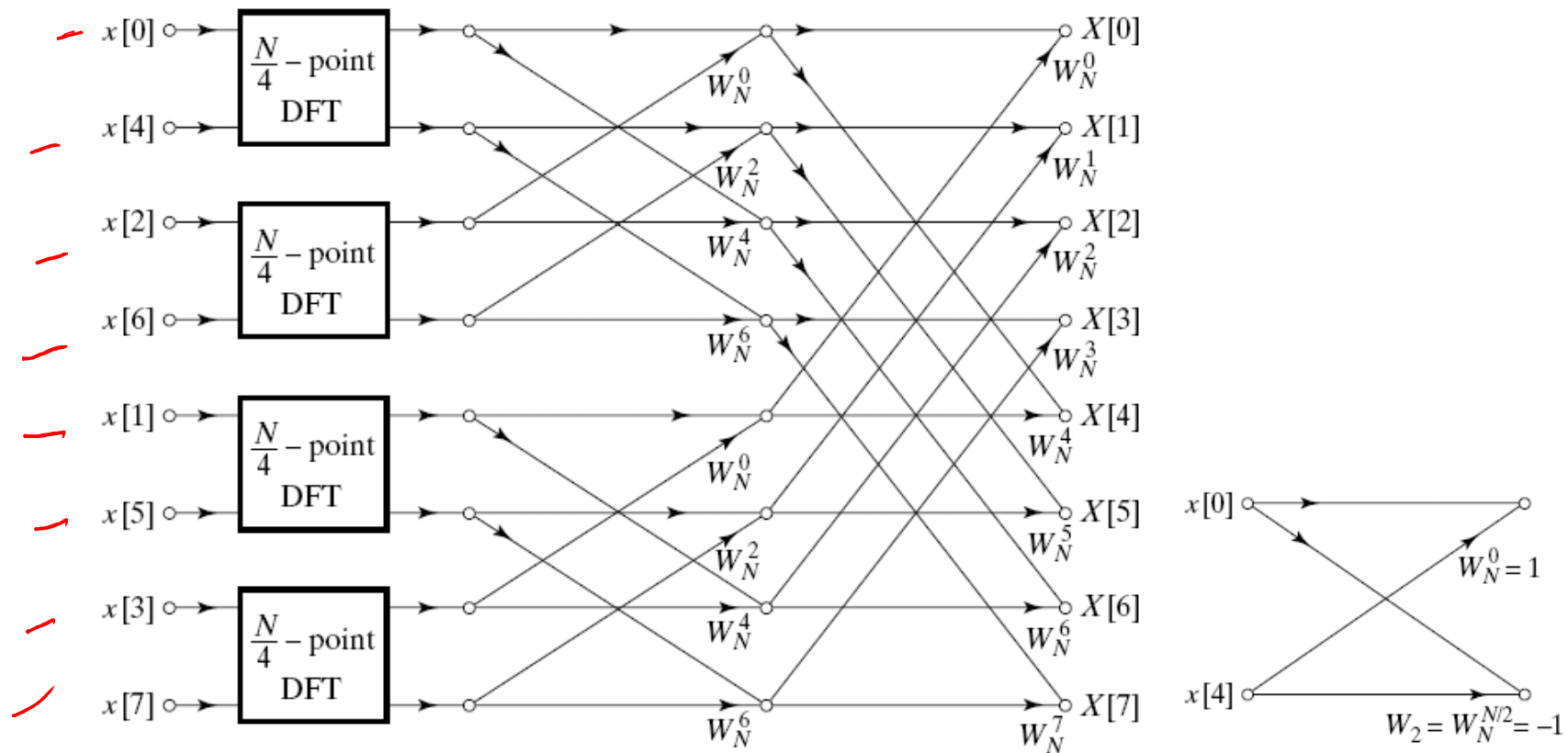


$N=8$



DECIMATION-IN-TIME FFT ALGORITHMS

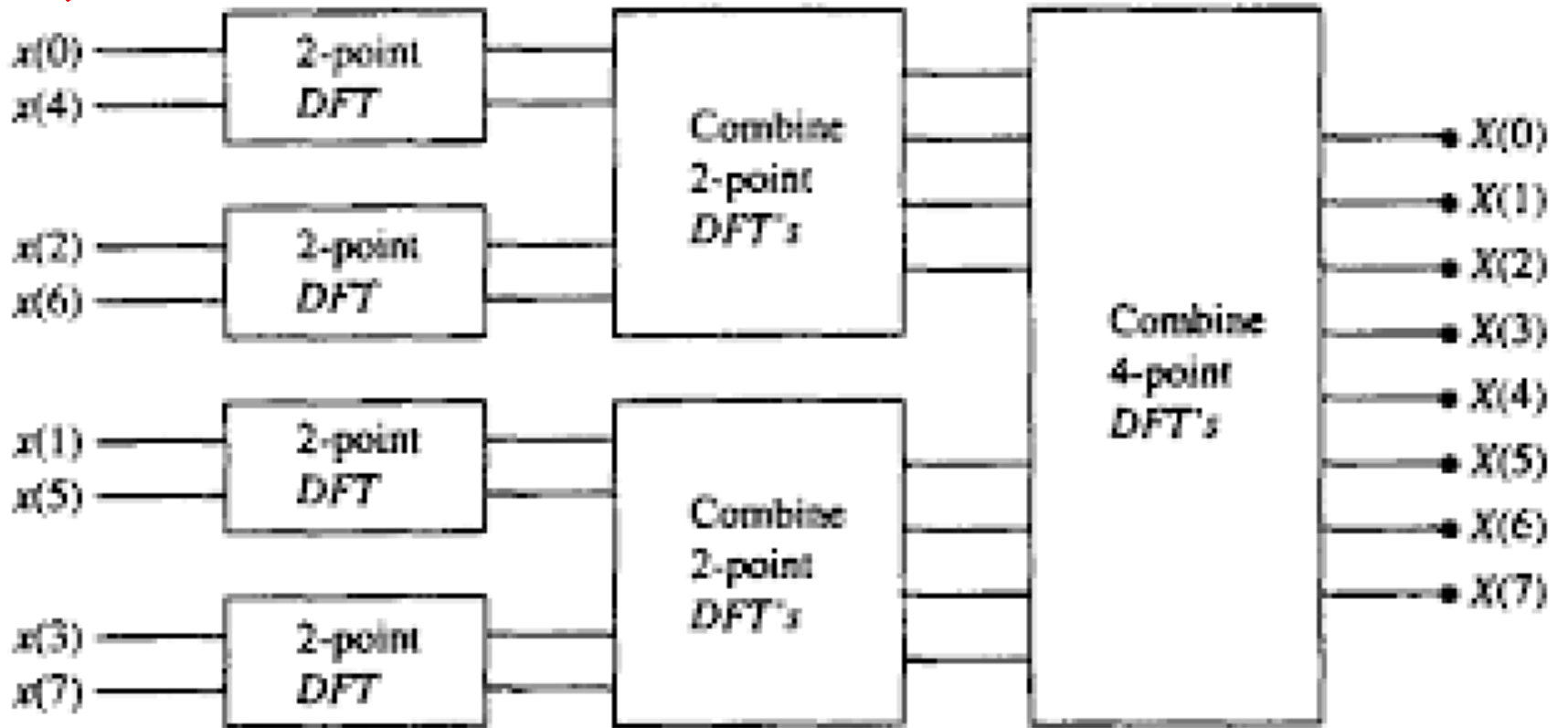
- After two steps of decimation in time



◆ Repeat until we're left with two-point DFT's

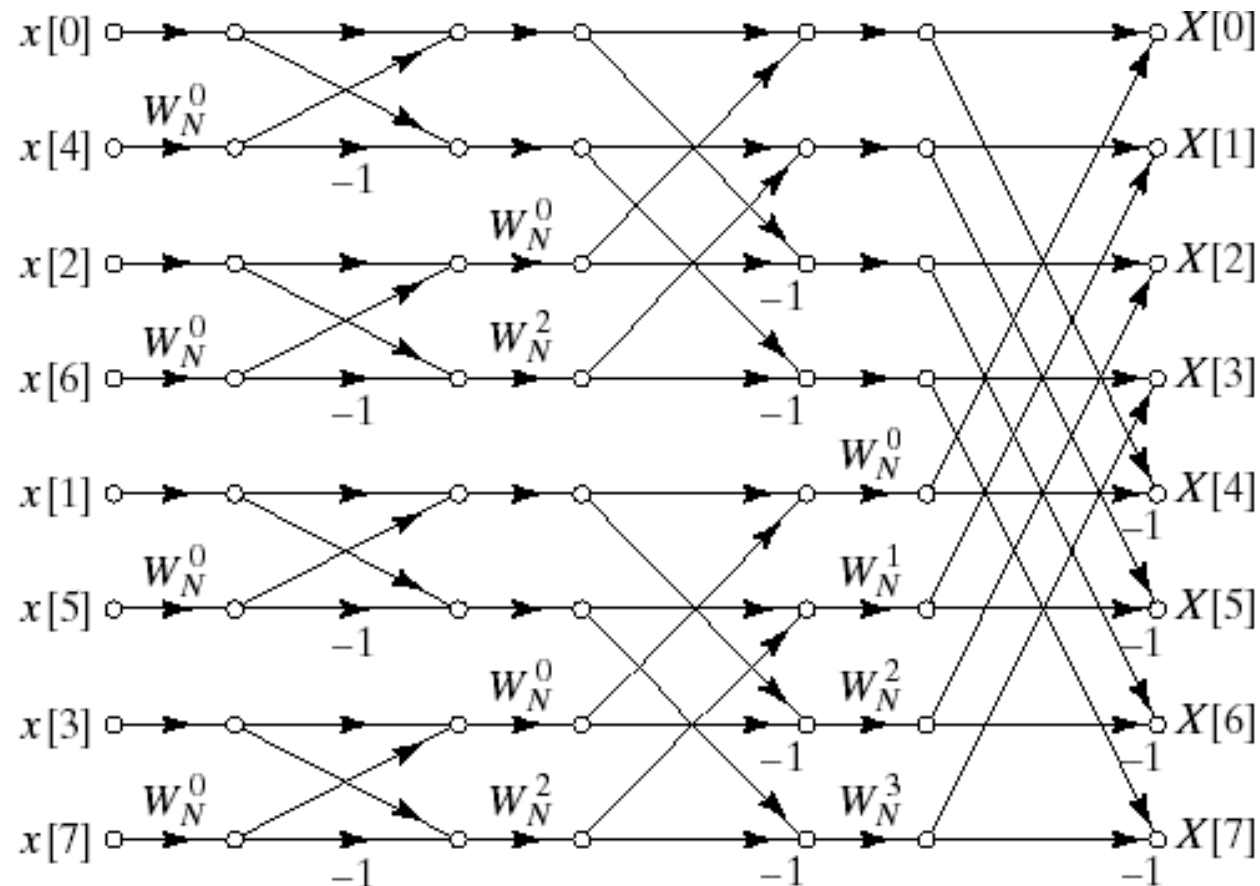


DIT-FFT



DECIMATION-IN-TIME FFT ALGORITHMS

- Final flow graph for 8-point decimation in time



◆ Complexity:

DSP_WINTER 2023

Dr.S.KALAIVANI

◆ $(N \log_2 N)/2$ complex multiplications and $N \log_2 N$ additions



RELATIONSHIP BETWEEN EXPONENTIAL FORMS AND TWIDDLE FACTORS (W) FOR PERIODICITY = N

| Sr. No. | Exponential form | Symbolic form |
|---------|---|------------------------|
| 01 | $e^{-j2\pi n/N} = e^{-j2\pi(n+N)/N}$ | $W_N^n = W_N^{n+N}$ |
| 02 | $e^{-j2\pi(n+N/2)/N} = -e^{-j2\pi n/N}$ | $W_N^{n+N/2} = -W_N^n$ |
| 03 | $e^{-j2\pi k} = e^{-j2\pi Nk/N} = 1$ | $W_N^{N+K} = 1$ |
| 04 | $e^{-j2(2\pi/N)} = e^{-j2\pi/(N/2)}$ | $W_N^2 = W_{N/2}$ |



VALUES OF VARIOUS TWIDDLES W_N^N FOR LENGTH $N=8$

| n | $W_N^n = e^{(-j2\pi)(n/N)}$ | Remarks |
|---|--------------------------------------|------------------|
| 0 | 1 = 1 \angle 0 | W_N^0 |
| 1 | $(1-j)/\sqrt{2}$ = 1 \angle -45° | W_N^1 |
| 2 | -j = 1 \angle -90° | W_N^2 |
| 3 | $-(1+j)/\sqrt{2}$ = 1 \angle -135° | W_N^3 |
| 4 | -1 = 1 \angle -180° | $W_N^4 = -W_N^0$ |
| 5 | $-(1-j)/\sqrt{2}$ = 1 \angle -225° | $W_N^5 = -W_N^1$ |
| 6 | j = 1 \angle 90° | $W_N^6 = -W_N^2$ |
| 7 | $(1+j)/\sqrt{2}$ = 1 \angle 45° | $W_N^7 = -W_N^3$ |



DIT –INPUT SEQUENCE ORDER

- Recall that, for $N = 8$, the first split requires the data to be arranged as follows:

$x_0, x_2, x_4, x_6, x_1, x_3, x_5, x_7$

- In the second and final split, the data appear in the following order:

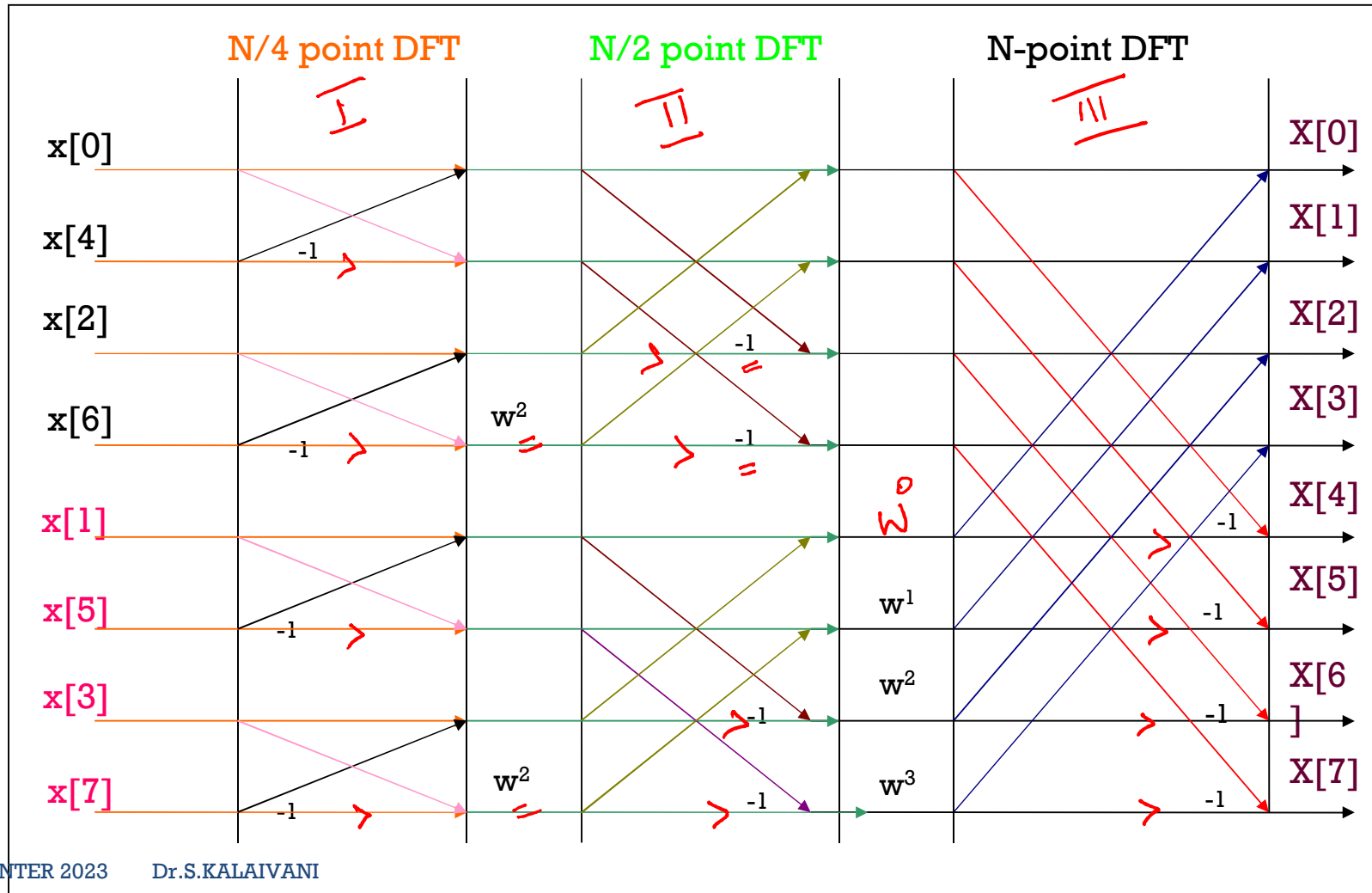
$x_0, x_4, x_2, x_6, x_1, x_5, x_3, x_7$

- The final order is said to be in “**bit reversed**” form:

| Original | Binary Form | Reversed Form | Final |
|----------|-------------|---------------|-------|
| 0 | 000 | 000 | 0 |
| 1 | 001 | 100 | 4 |
| 2 | 010 | 010 | 2 |
| 3 | 011 | 110 | 6 |
| 4 | 100 | 001 | 1 |
| 5 | 101 | 101 | 5 |
| 6 | 110 | 011 | 3 |
| 7 | 111 | 111 | 7 |



~~N=8~~ N=8-POINT RADIX-2 DIT-FFT:



TWIDDLE FACTOR for $N=8$

$$W_8^0 = 1$$

$$W_8^4 = -1$$

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

$$W_8^5 = -\frac{1}{\sqrt{2}} + j\frac{1}{\sqrt{2}}$$

$$W_8^2 = -j$$

$$W_8^6 = j$$

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

$$W_8^7 = \frac{1}{\sqrt{2}} + j\frac{1}{\sqrt{2}}$$



PROBLEM 1

- Find 8-point DFT of the sequence

$$x[n] = [1, 1, 1, 1, 0, 0, 0, 0]$$

using DIT-FFT algorithm



$$x[n] = [1, 1, 1, 1, 0, 0, 0, 0]$$

$$X[k] = G[k] + W_N^k H[k]$$

$$X[k+N/2] = G[k] - W_N^k H[k]$$

To compute the output of first stage:

$$X_1(0) = 1 + 0 \times W_8^0 = 1$$

$$X_1(1) = 1 - 0 \times W_8^0 = 1$$

$$X_1(2) = 1 + 0 \times W_8^0 = 1$$

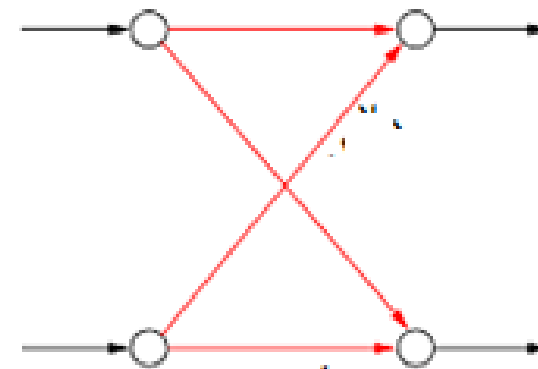
$$X_1(3) = 1 - 0 \times W_8^0 = 1$$

$$X_1(4) = 1 + 0 \times W_8^0 = 1$$

$$X_1(5) = 1 - 0 \times W_8^0 = 1$$

$$X_1(6) = 1 + 0 \times W_8^0 = 1$$

$$X_1(7) = 1 - 0 \times W_8^0 = 1$$



$$x[n] = [1, 1, 1, 1, 0, 0, 0, 0]$$

$$X[k] = G[k] + W_N^k H[k]$$

$$X[k+N/2] = G[k] - W_N^k H[k]$$

To compute the output of second stage:

$$X_2(0) = 1 + W_8^0 \times 1 = 2$$

$$X_2(1) = 1 + 1 \times W_8^2 = 1 - j$$

$$X_2(2) = 1 - W_8^0 \times 1 = 0$$

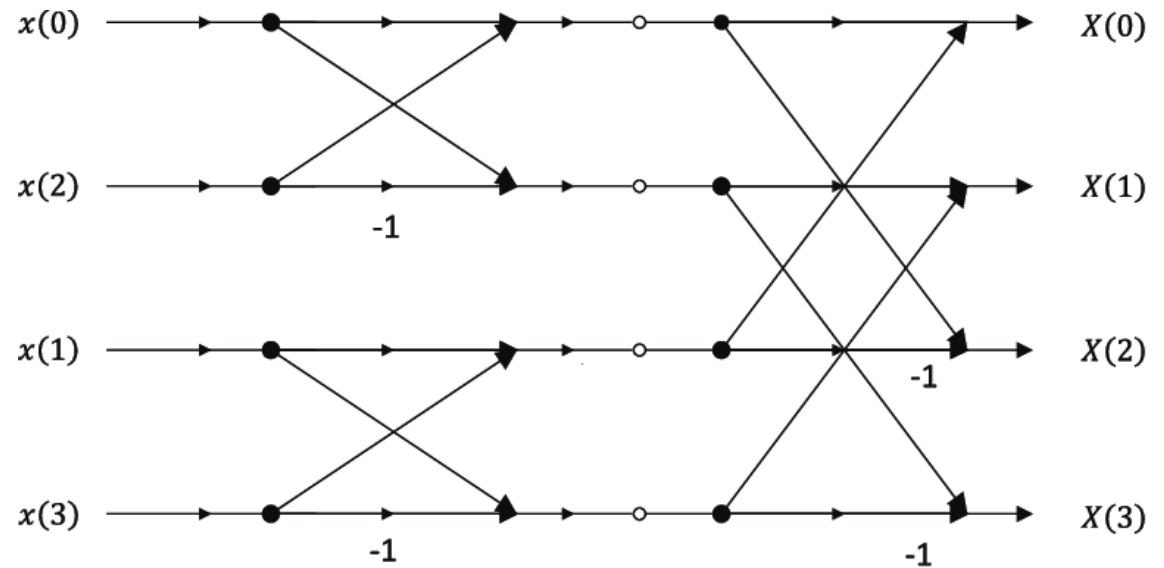
$$X_2(3) = 1 - 1 \times W_8^2 = 1 + j$$

$$X_2(4) = 1 + W_8^0 \times 1 = 2$$

$$X_2(5) = 1 + W_8^2 \times 1 = 1 - j$$

$$X_2(6) = 1 - W_8^0 \times 1 = 0$$

$$X_2(7) = 1 - W_8^2 \times 1 = 1 + j$$



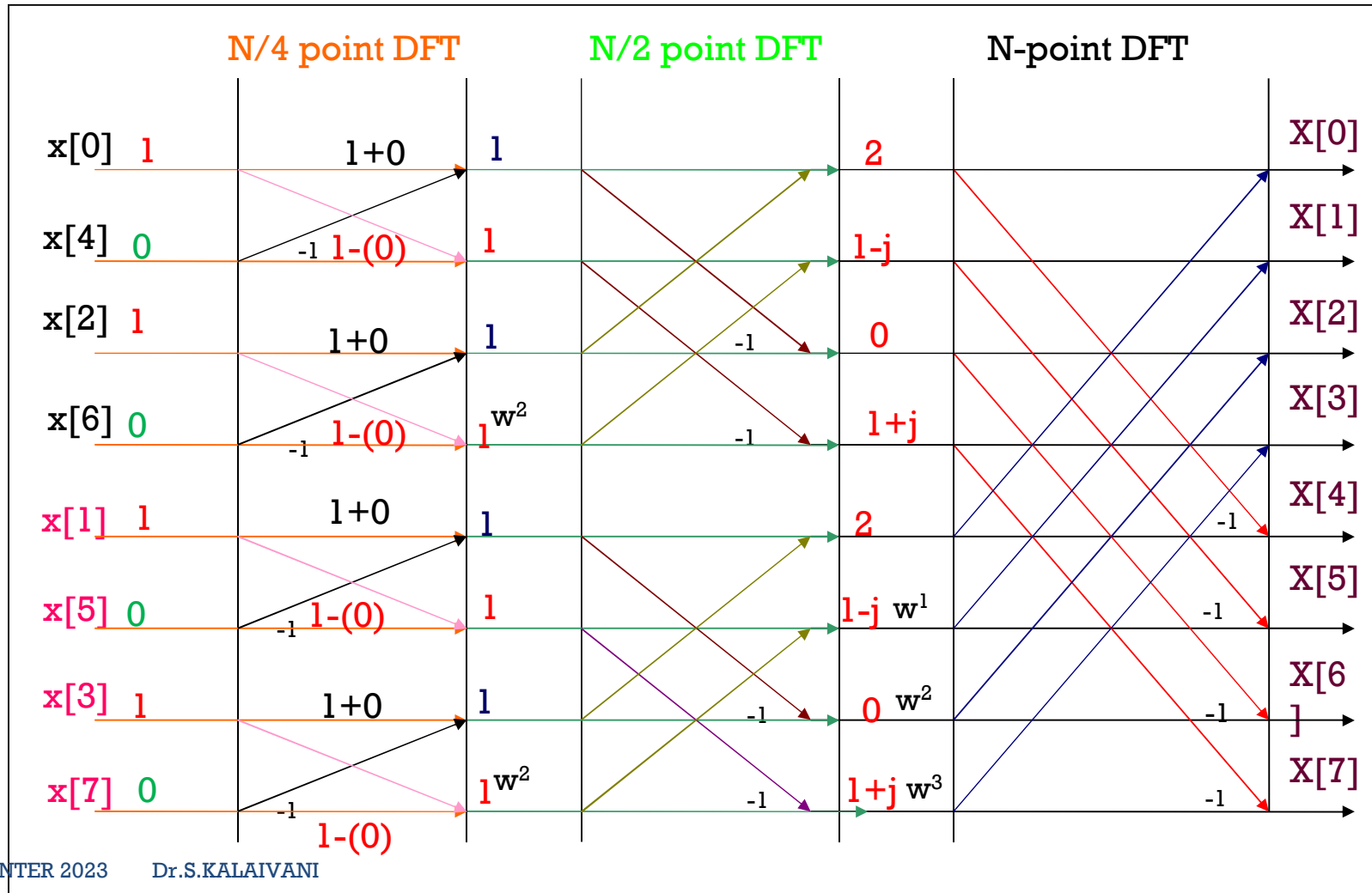
$$x[n] = [1, 1, 1, 1, 0, 0, 0, 0]$$

$$W_8^0 = 1$$

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

$$W_8^2 = -j$$

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



To compute the output of third stage

$$X_3(0) = X(0) = 2 + 2W_8^0 = 4$$

$$X_3(1) = X(1) = 1 - j + (1 - j)W_8^1 = 1 - j2.41$$

$$X_3(2) = X(2) = 0 + 0 \times W_8^2 = 0$$

$$X_3(3) = X(3) = (1 + j) + (1 + j)W_8^3 = 1 - j0.414$$

$$X_3(4) = X(4) = 2 - 2W_8^0 = 0$$

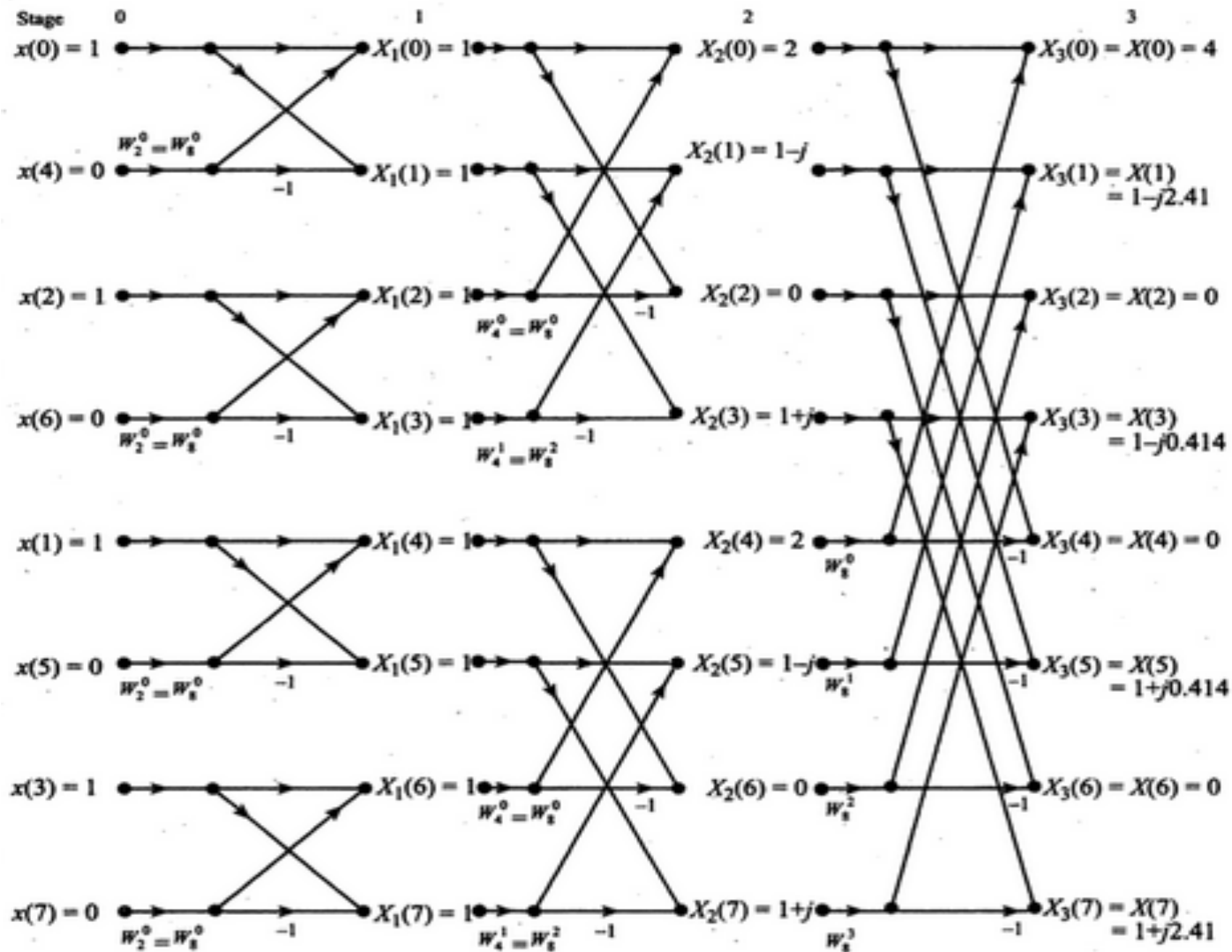
$$X_3(5) = X(5) = (1 - j) - (1 - j)W_8^1 = 1 + j0.414$$

$$X_3(6) = X(6) = 0 - 0 \times W_8^2 = 0$$

$$X_3(7) = X(7) = (1 + j) - (1 + j)W_8^3 = 1 + j2.414$$

Since $x(n)$ is a real sequence, we find that the symmetry property: $X(k) = X^*(8 - k)$ is satisfied.



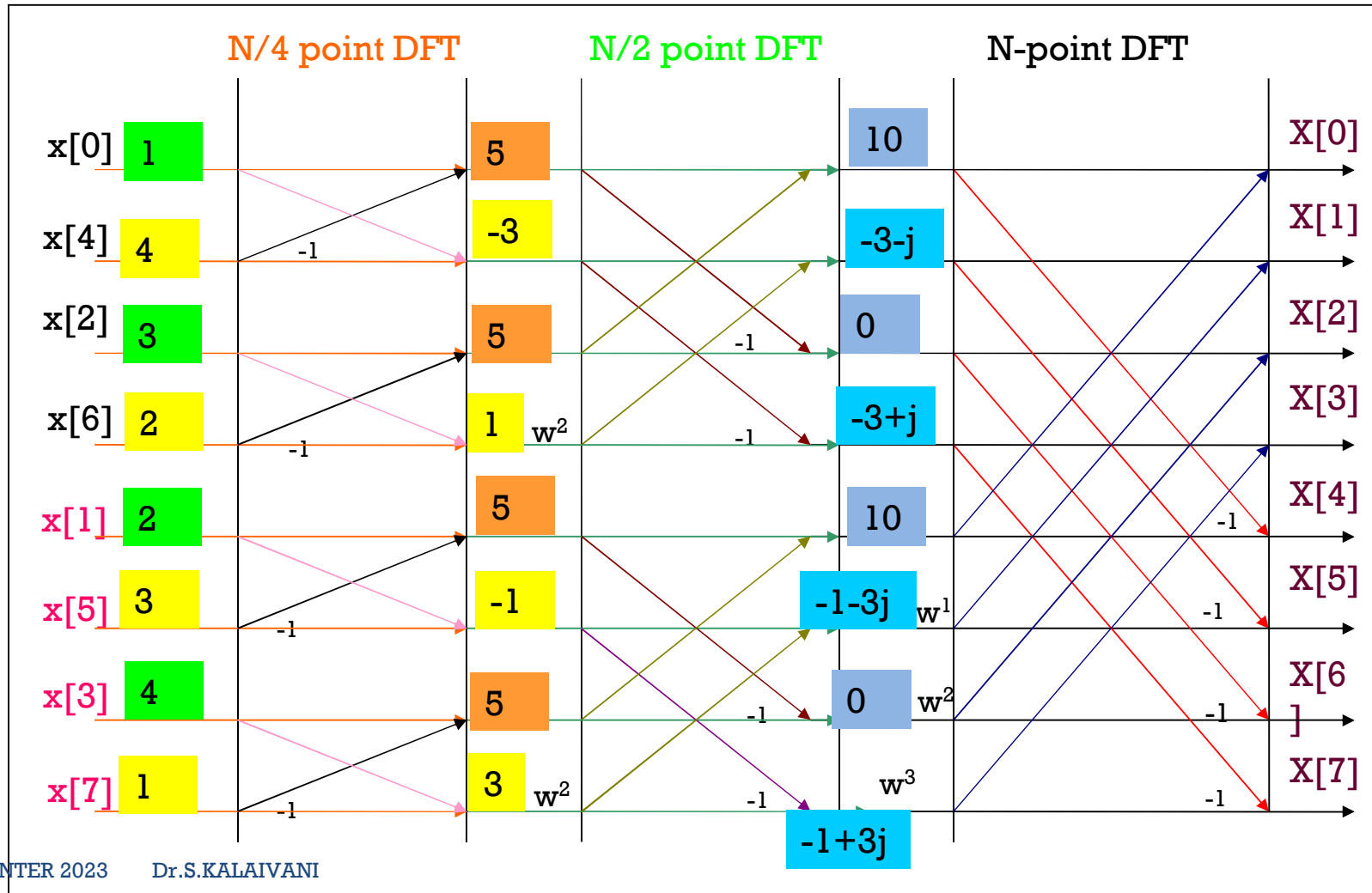


PROBLEM-2

- Find 8-point DFT of the sequence
 $x[n] = [1, 2, 3, 4, 4, 3, 2, 1]$
using DIT-FFT algorithm



$$x[n] = [1, 2, 3, 4, 4, 3, 2, 1]$$



SOL 2

To compute the output of first stage

$$X_1(0) = 1 + 4 \times W_8^0 = 5$$

$$X_1(1) = 1 - 4 \times W_8^0 = -3$$

$$X_1(2) = 3 + 2 \times W_8^0 = 5$$

$$X_1(3) = 3 - 2 \times W_8^0 = 1$$

$$X_1(4) = 2 + 3 \times W_8^0 = 5$$

$$X_1(5) = 2 - 3 \times W_8^0 = -1$$

$$X_1(6) = 4 + 1 \times W_8^0 = 5$$

$$X_1(7) = 4 - 1 \times W_8^0 = 3$$

To compute the output of the second stage

$$X_2(0) = 5 + 5 \times W_8^0 = 10$$

$$X_2(1) = -3 + 1 \times W_8^2 = -3 - j$$

$$X_2(2) = 5 - 5 \times W_8^0 = 0$$

$$X_2(3) = -3 - 1 \times W_8^2 = -3 + j$$

$$X_2(4) = 5 + 5 \times W_8^0 = 10$$

$$X_2(5) = -1 + 3 \times W_8^2 = -1 - j3$$

$$X_2(6) = 5 - 5 \times W_8^0 = 0$$

$$X_2(7) = -1 - 3 \times W_8^2 = -1 + j3$$



CONTD.,

To compute the output of the third stage

$$X_3(0) = X(0) = 10 + 10 \times W_8^0 = 20$$

$$X_3(1) = X(1) = -3 - j + (-1 - 3j)W_8^1 = -5.828 - j2.414$$

$$X_3(2) = X(2) = 0 + 0 \times W_8^2 = 0$$

$$X_3(3) = X(3) = -3 + j + (-1 + 3j)W_8^3 = -0.171 - j0.414$$

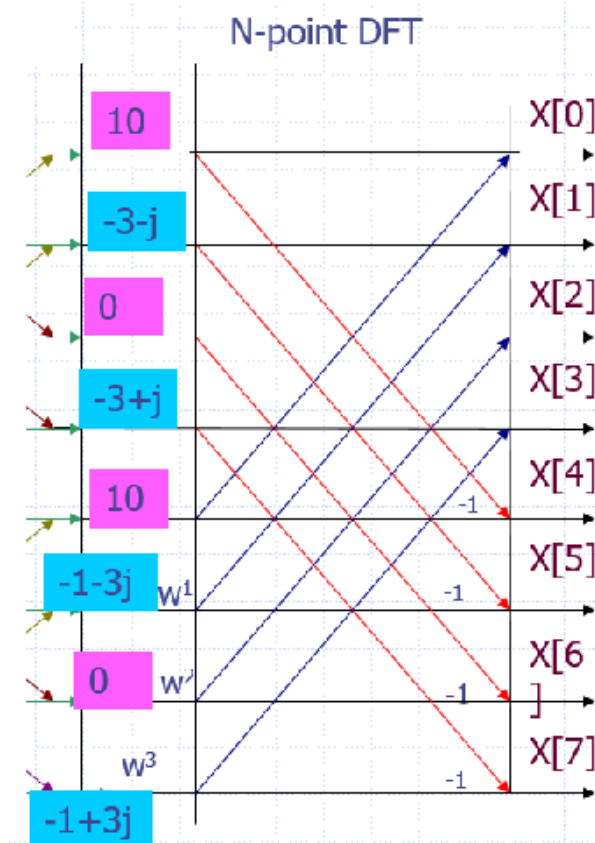
$$X_3(4) = X(4) = 10 - 10 \times W_8^0 = 0$$

$$X_3(5) = X(5) = -3 - j - (-1 - 3j)W_8^1 = -0.171 + j0.414$$

$$X_3(6) = X(6) = 0 - 0 \times W_8^2 = 0$$

$$X_3(7) = X(7) = -3 + j - (-1 + 3j)W_8^3 = -5.828 + j2.414$$

Since $x(n)$ is a real sequence, we find that the symmetry condition: $X(k) = X^*(8 - k)$ is satisfied.



DECIMATION IN FREQUENCY (DIF) ALGORITHM

UNIT-II



DIF

Decimation in Frequency

The discrete Fourier transform can be found using

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

$X(k)$ can be expressed as

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

Let $n = n + N/2$

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



CONTD.,

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

$W_N^{kn} \cdot W_N^{kN/2}$
 $=$

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

But

$$W_N^{(N/2)k} = e^{2\pi k(N/2)/N} = e^{\pi k} = \underline{\underline{(-1)^k}}$$

Then

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



DIT - n'
DIF - k'

CONTD.,

$x(n)$
DIT

$X(k)$
DIF

If $k = 2m$ or an even number

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

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

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

Then $X(2m)$ is $N/2$ -point DFT for $a(n)$

Hence, even values of $X[k]$, ie., $X[0]$, $X[2]$, $X[4]$ and $X[6]$ for $N=8$, can be determined from the above equation.



CONTD.,

- To find odd values of $X[k]$, **let $k = 2r + 1$**

$$\begin{aligned} X[2r+1] &= \sum_{n=0}^{N-1} x[n] W_N^{n(2r+1)} = \sum_{n=0}^{N/2-1} x[n] W_N^{n(2r+1)} + \sum_{n=N/2}^{N-1} x[n] W_N^{n(2r+1)} \\ &= \sum_{n=0}^{N/2-1} x[n] W_N^{n(2r+1)} + \sum_{n=0}^{N/2-1} x[n + N/2] W_N^{(n+N/2)(2r+1)} \end{aligned}$$

$$W_N^{(n+\frac{N}{2})(2r+1)} = W_N^{(n)(2r+1)} W_N^{(\frac{N}{2})(2r+1)}$$

$$W_N^{\frac{N}{2}(2r+1)} = W_N^{Nr} W_N^{N/2} = -1$$



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

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

$$W_N^{n(2r+1)} = W_N^{2rn} W_N^n = W_{N/2}^{rn} W_N^n$$

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

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

$$b(n) = x(n) - x(n + N/2)$$

$X(2m+1)$ is $N/2$ -point DFT for $b(n) W_N^n$



CONTD.,

$$\begin{aligned} X[2r] &= \sum_{n=0}^{N/2-1} (x[n] + x[n + N/2]) W_{N/2}^{nr} \\ &\quad \underline{\quad} \\ &= \sum_{n=0}^{N/2-1} g(n) W_{N/2}^{nr} \end{aligned}$$

- $g[0] = x[0] + x[4]$
- $g[1] = x[1] + x[5]$
- $g[2] = x[2] + x[6]$
- $g[3] = x[3] + x[7]$



CONT'D

$$X[2r+1] = \sum_{n=0}^{N/2-1} (x[n] - x[n + N/2]) W_N^n W_{N/2}^{rn}$$

$$= \sum_{n=0}^{N/2-1} h(n) W_N^n W_{N/2}^{rn}$$

$$h[0] = x[0] - x[4]$$

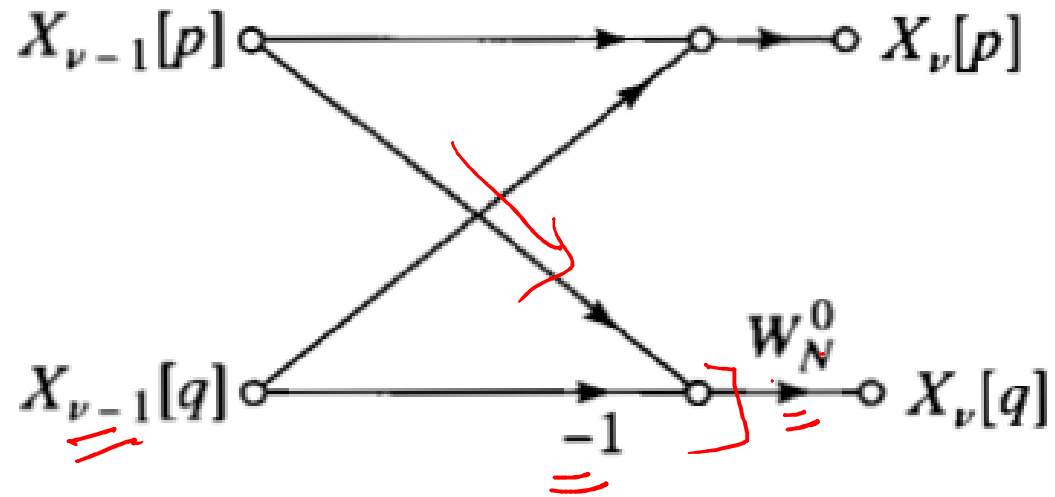
$$h[1] = x[1] - x[5]$$

$$h[2] = x[2] - x[6]$$

$$h[3] = x[3] - x[7]$$



2-point DIF - FFT

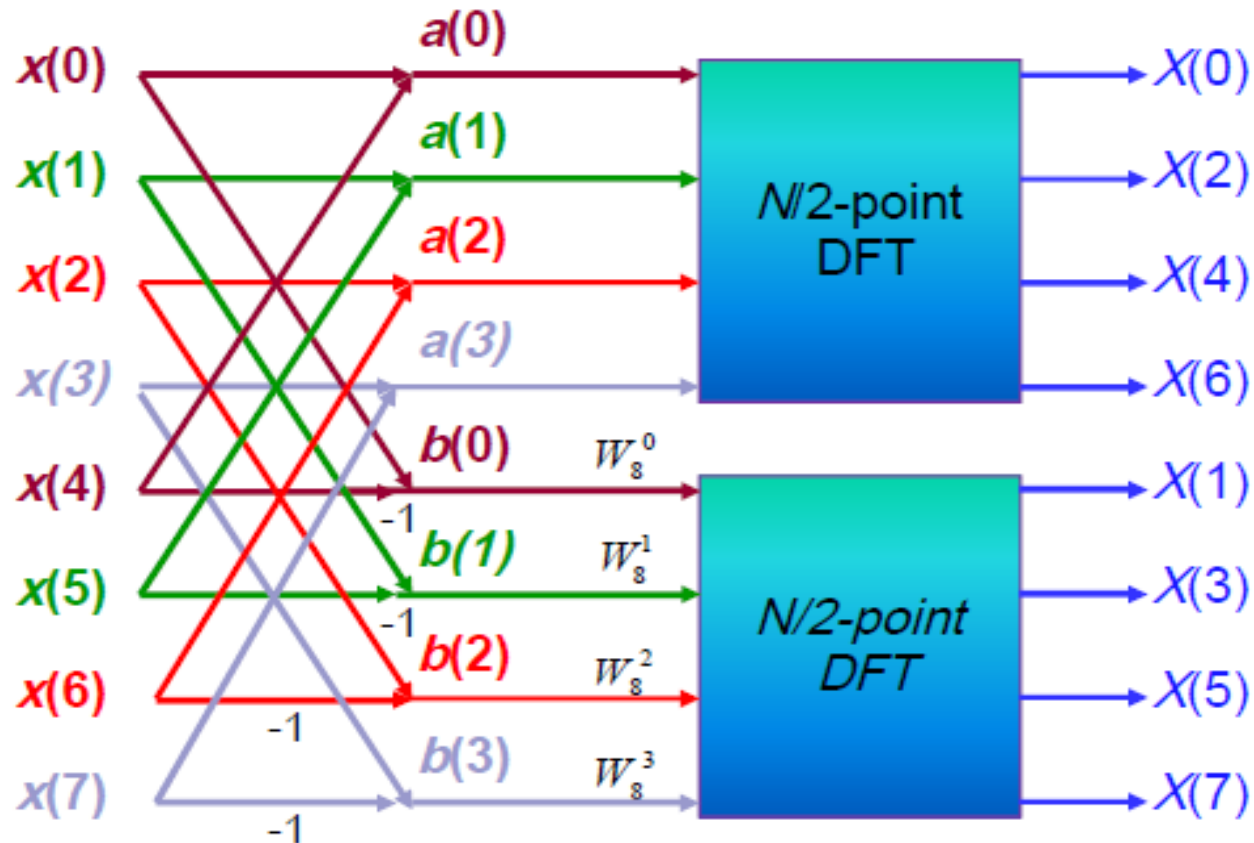


$$X_v(p) = X_{v-1}(p) + X_{v-1}(q)$$

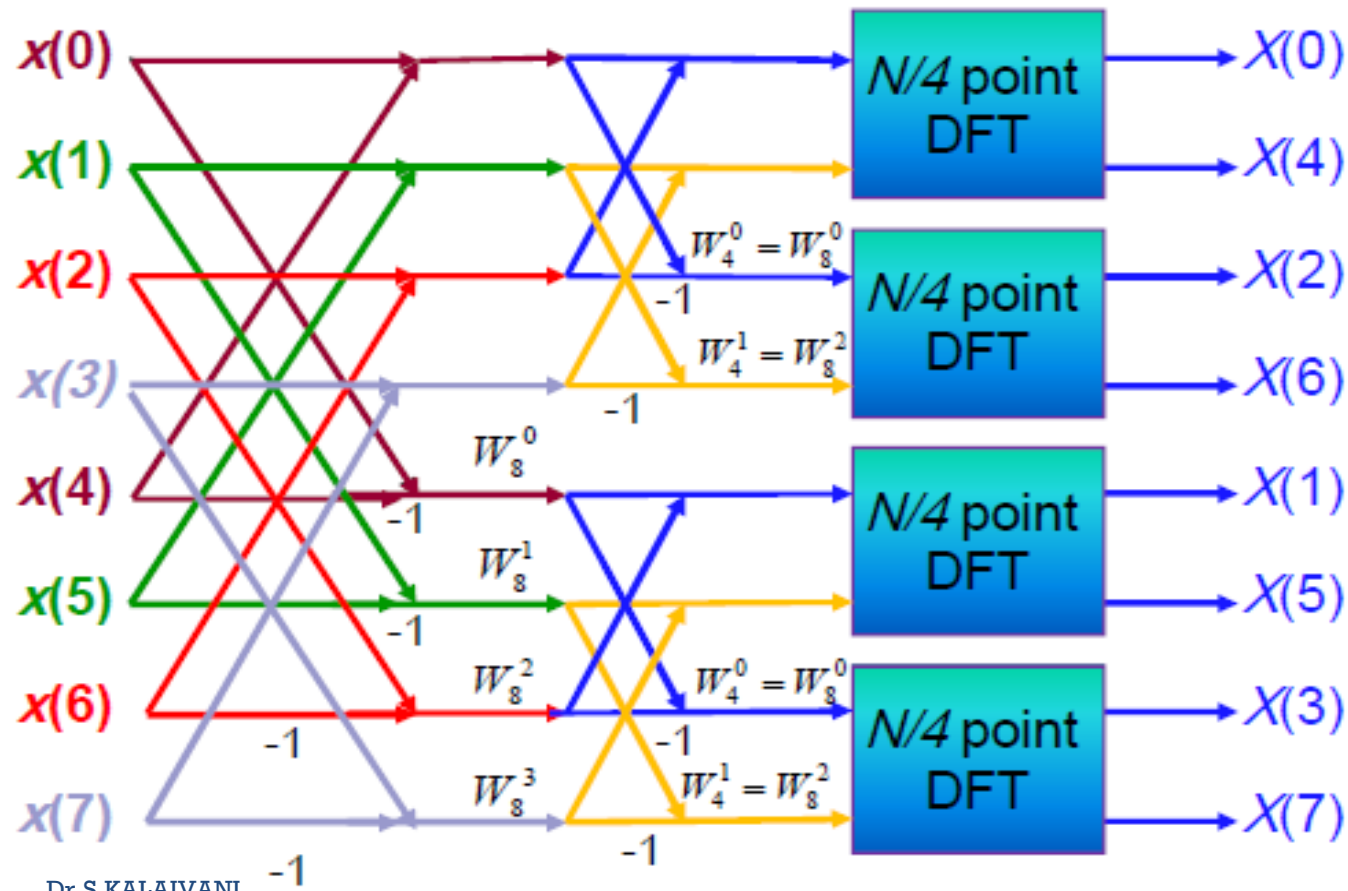
$$X_v(q) = \left[X_{v-1}(p) - X_{v-1}(q) \right] W_8^0 \quad \text{when } N = 8$$



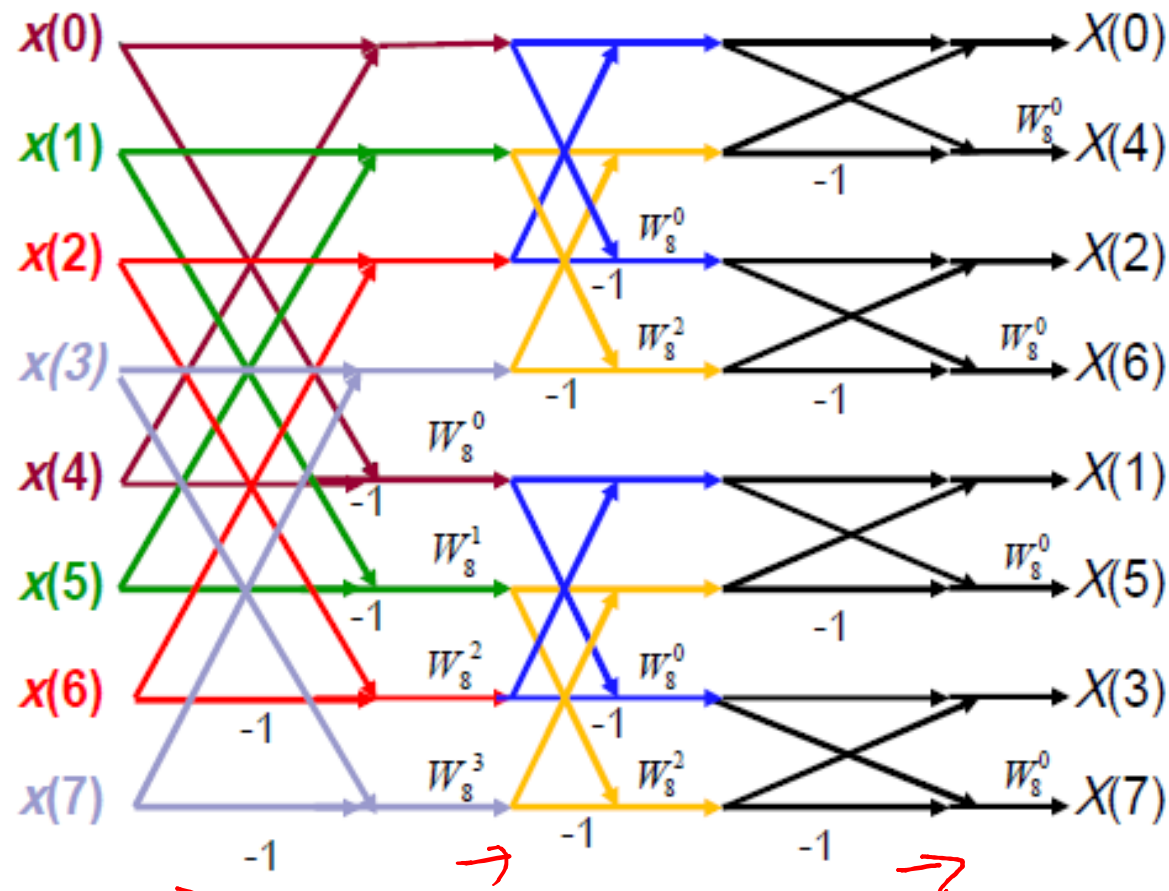
STAGE-I COMPUTATION



STAGE-II COMPUTATION

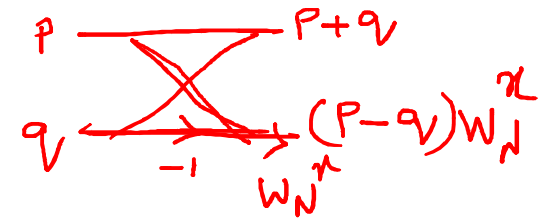
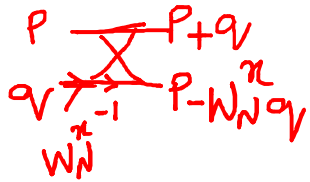


STAGE-III COMPUTATION



COMPARING DIT AND DIF STRUCTURES:

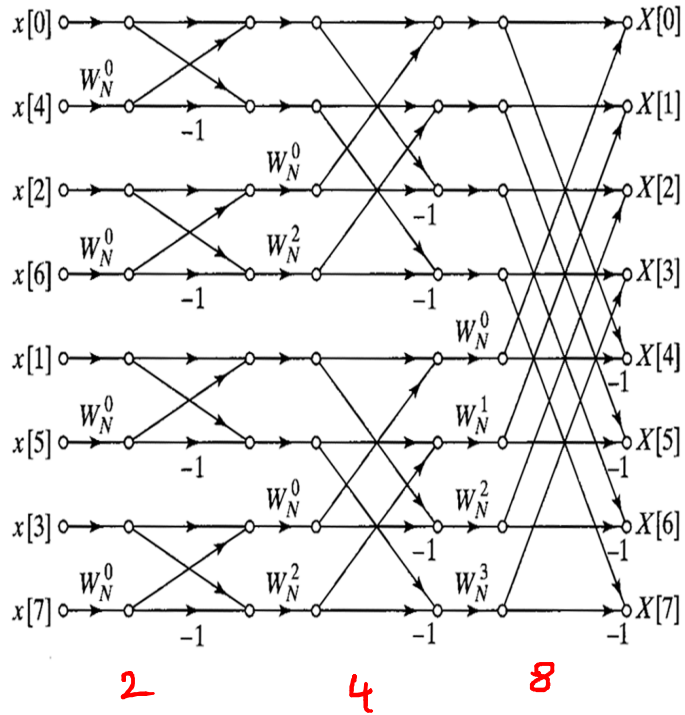
The DIF FFT is the *transpose* of the DIT FFT



'N'

DIT FFT structure:

1/P

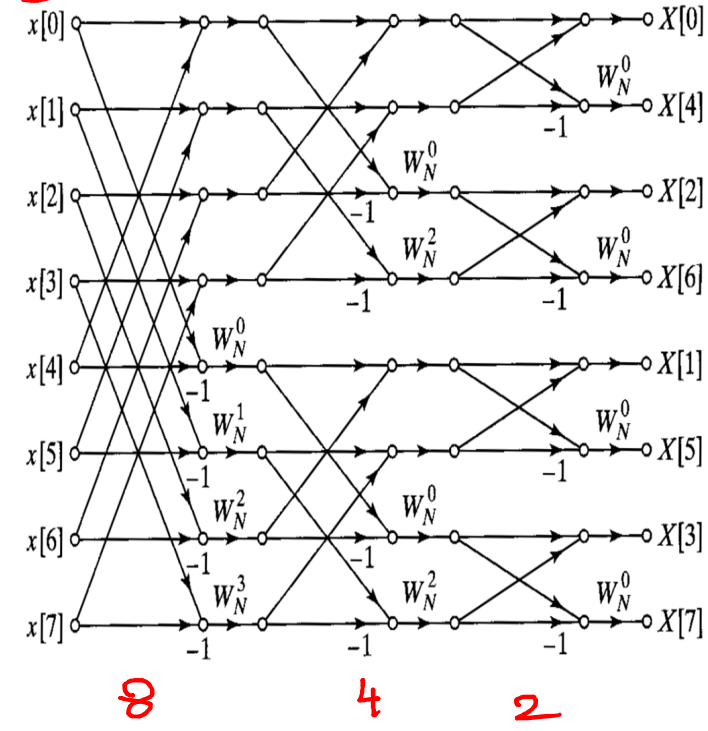


1/P

DIF FFT structure:

'K'

1/P



COMPUTATIONAL COMPLEXITY

Using the previous algorithm , the complex multiplications needed is only 12. While using the normal DFT would require 64 complex multiplications

In general

Complex multiplication of DFT is: N^2

Complex multiplication of FFT is $(N/2) \log_2(N)$

If $N = 1024$

Complex multiplication of DFT is: 1,048,576

Complex multiplication of FFT is: 5,120



NUMERICALS



EX.1

$$x(n) = \{1, 2, 3, 4\} = x(n) = \{1, 2, 3, 4, \underbrace{0, 0}_{\text{8-pt DFT}}\}$$

4-pt DFT 8-pt DFT

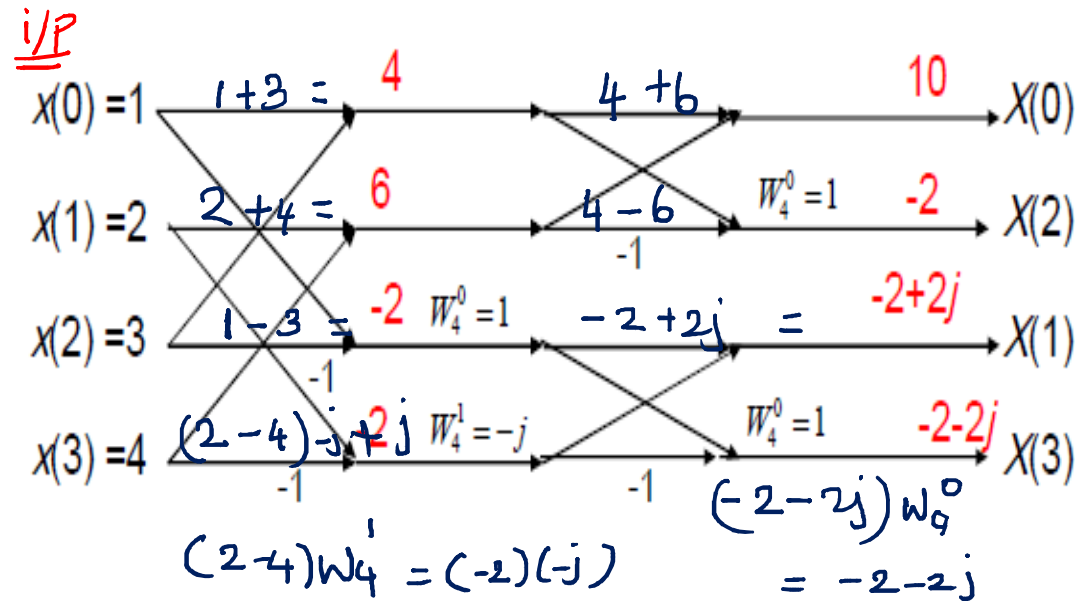
Example Given a sequence $x(n)$ where $x(0) = 1, x(1) = 2, x(2) = 3, x(3) = 4$ and $x(n) = 0$ elsewhere, find DFT for the first four points

solution

DIF-FFT

4-pt

4 - 2



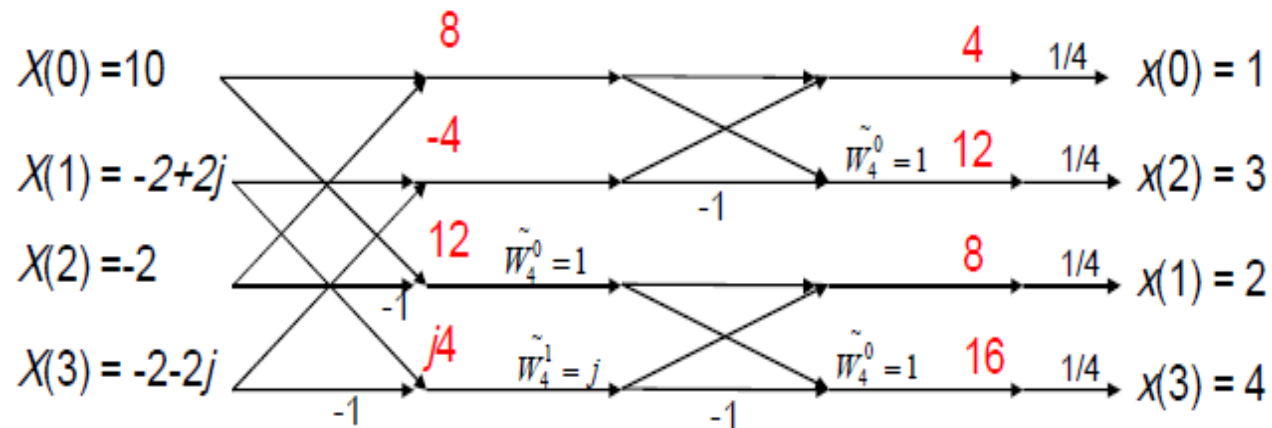
$$x[n] = \{1, 2, 3, 4\}$$

$$X[k] = \{10, -2+2j, -2, -2-2j\}$$

CONTD.,

- Verify your answer through inverse DIF-FFT algorithm

solution



EX.2

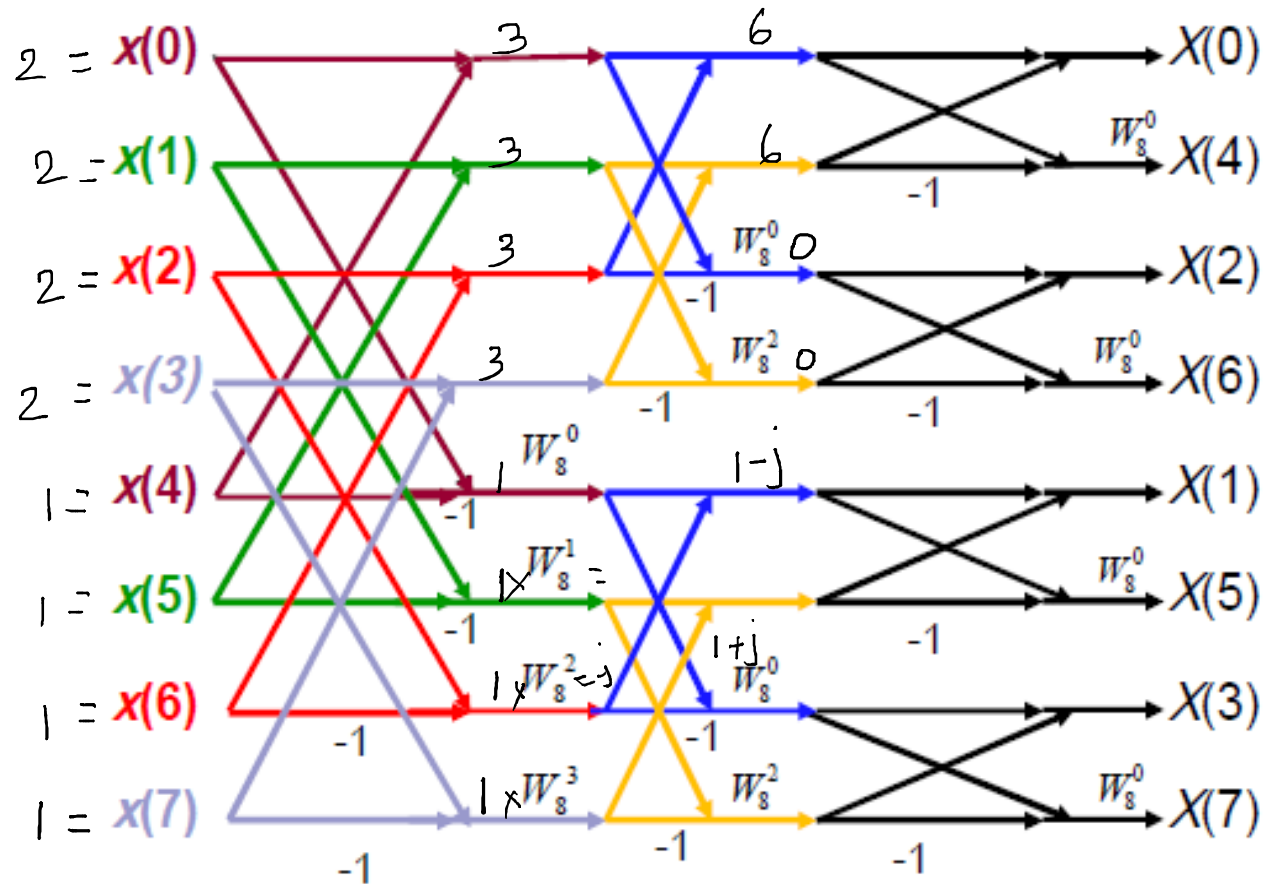
- The given sequence is:

$$x[n] = \{2, 2, 2, 2, 1, 1, 1, 1\}$$

Compute 8-point DFT by using DIF-FFT algorithm.



$$x[n] = \{2, 2, 2, 2, 1, 1, 1, 1\}$$



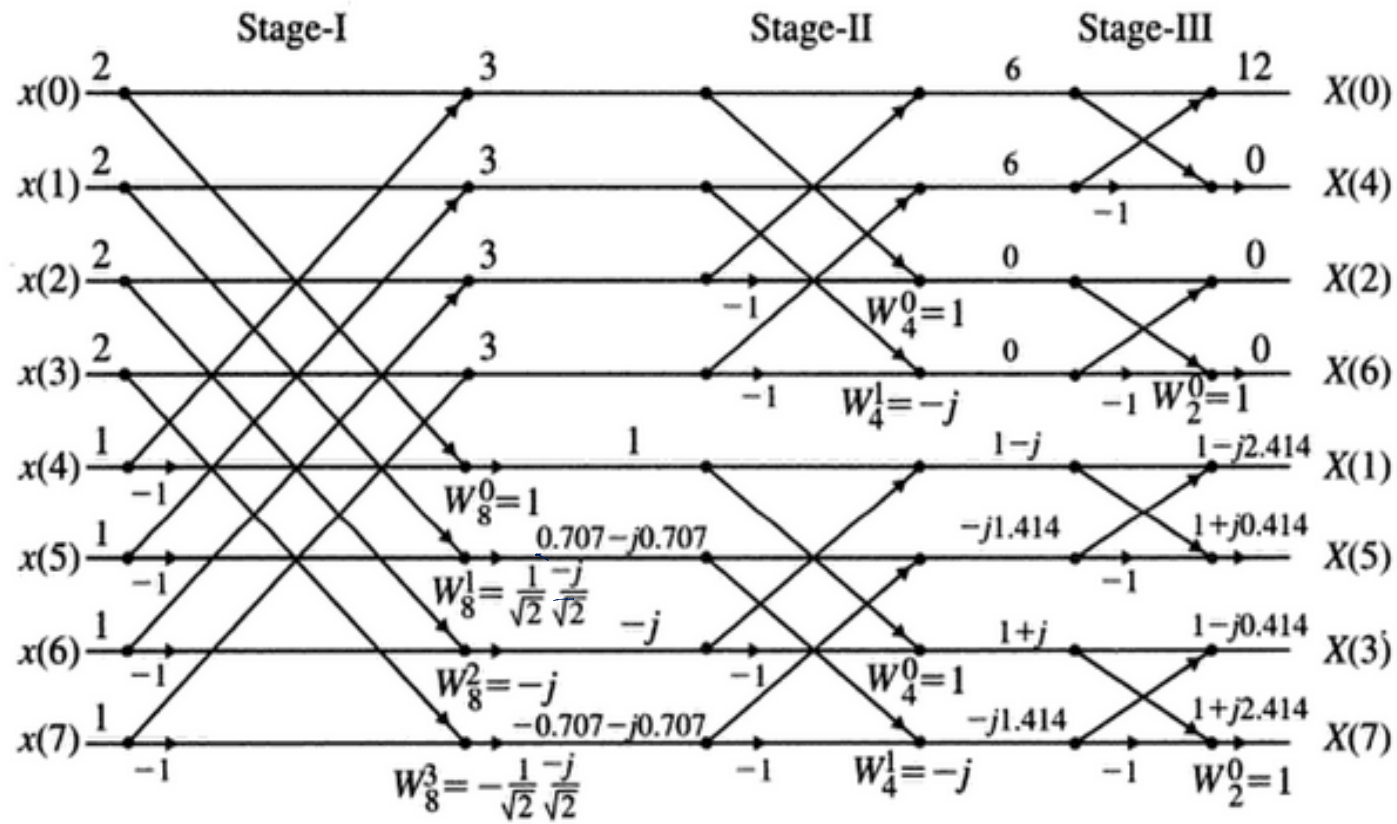
$$W_8^1 = \frac{1-j}{\sqrt{2}}$$

$$W_8^2 = -j$$

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



SOLUTION



Input is normal order

Output is bit reversed order



EX.3

$$\delta(n) = 1, n=0 \\ 0, n \neq 0$$

- Compute 8-point DFT of the given sequence $x[n] = \delta[n]$:

$$x[n] = \{1, 0, 0, 0, 0, 0, 0, 0\}$$

using (a) DFT (b) FFT algorithm

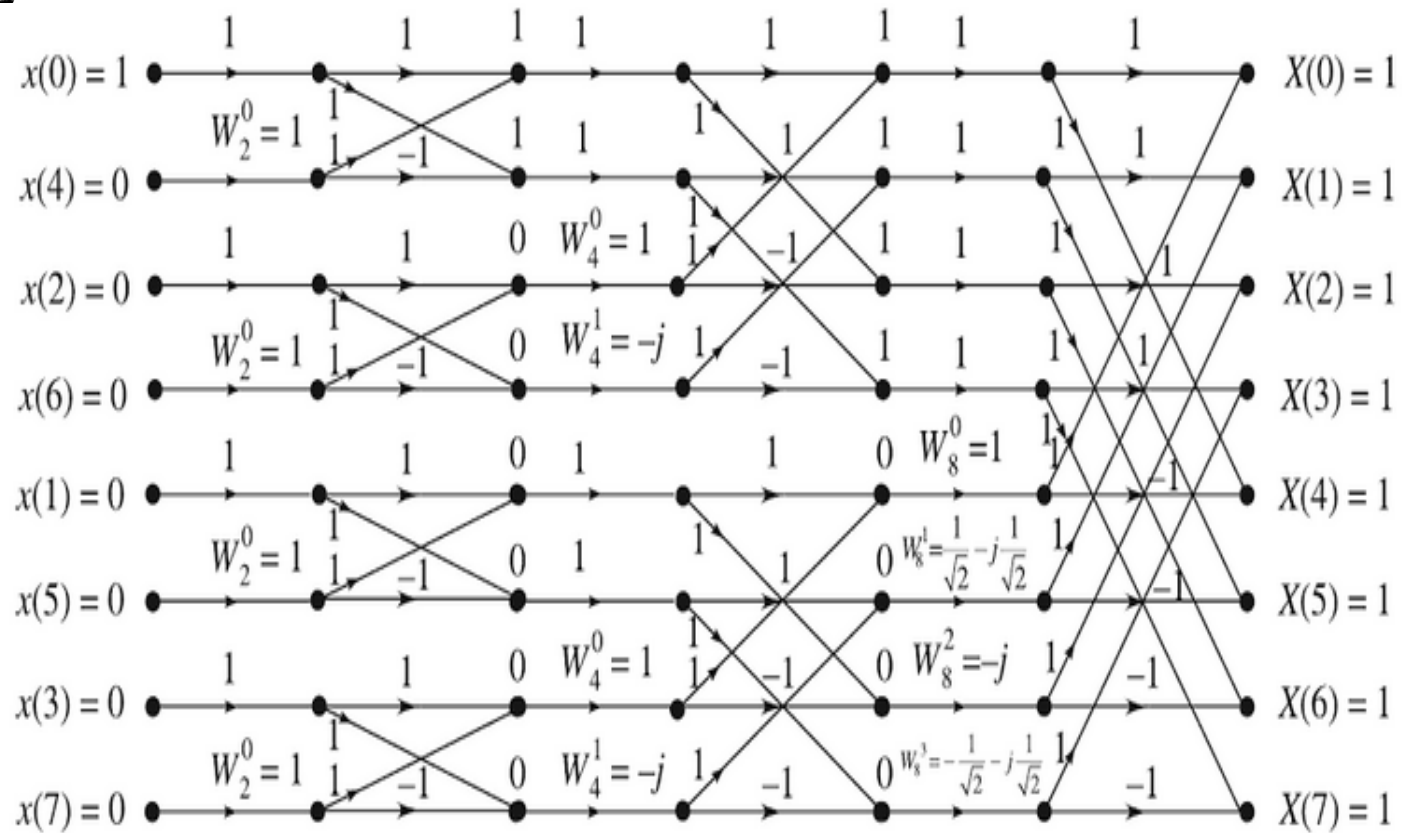
$$\sum_{n=0}^{N-1} x(n) W_N^{kn}$$

- DIT
- DIF

Solution: $X[k] = \{1, 1, 1, 1, 1, 1, 1, 1\}$



CONTD



EX.4

4, 8-P

$$\begin{aligned} \checkmark x(n) &= \{1, 2, 3, 0\} \\ \checkmark h(n) &= \{-1, -1, 0, 0\} \end{aligned}$$

- In an LTI system the input $x[n]=\{1,2,3\}$ and impulse response $h[n]=\{-1,-1\}$. Determine the response of the system by radix-2 algorithm.

DIT DIF

$$\underline{4\text{pt DFT}} \xrightarrow{x(n)} \underline{X(k)}$$

$$\underline{4\text{pt DFT}} \{h(n)\} \xrightarrow{\text{DIT/DIF}} \underline{H(k)}$$

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

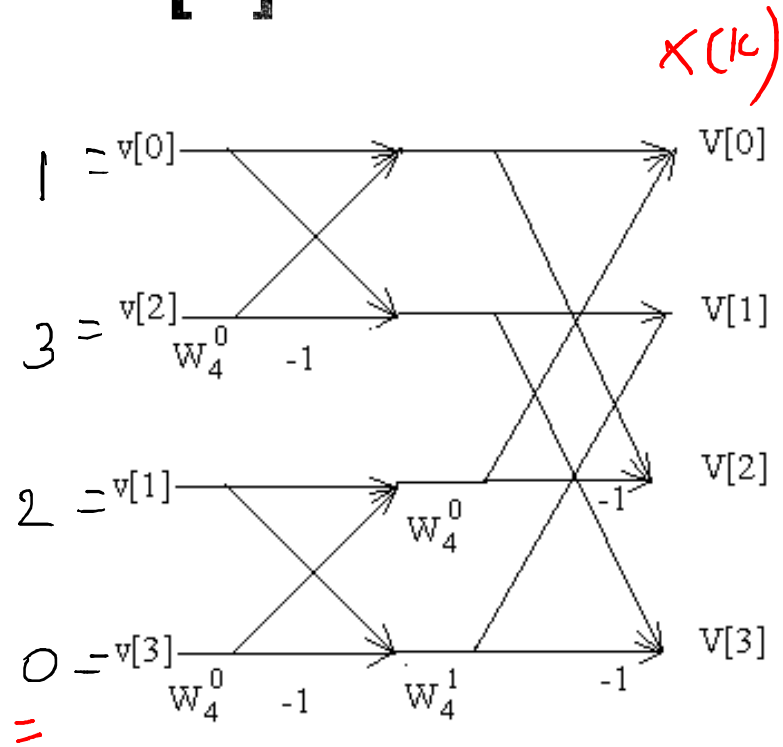
$$\checkmark Y(k) = \tilde{X}(k) \tilde{H}(k)$$

$$\underline{y(n)} = \underline{\text{IDFT}} \{ \underline{Y(k)} \}$$



$X[N] \rightarrow X[K]$

$$x[n] = \{1, 2, 3\}$$

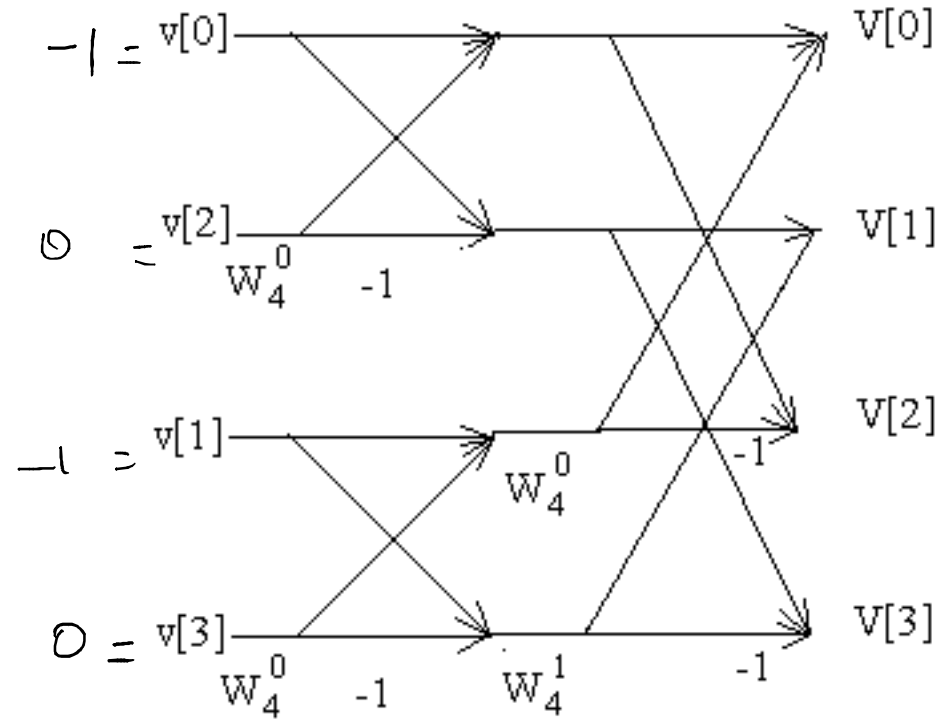


H[N] → H[K]

$$h[n] = \{-1, -1\}$$

DIT

H(1c)



$$Y(K) = X(K) * H(K)$$

- $$Y(k) = \{6, -2-2j, 2, -2+j2\} * \{-2, \cancel{1+j}, 0, -j\}$$

$$= \{-12, 4, 0, 4\} = 2+2j \}$$

$$6 \times -2 = -12$$

$$(-2-2j)(-1+j) = 4$$

$$2(0) = 0$$

$$(-2+j2)(0-j) = 4$$

$$Y(4) = X(4) H(4)$$

$$Y(0) = X(0) H(0)$$

$$Y(1) = X(1) H(1)$$

$$Y(2) = X(2) H(2)$$

$$Y(3) = X(3) H(3)$$

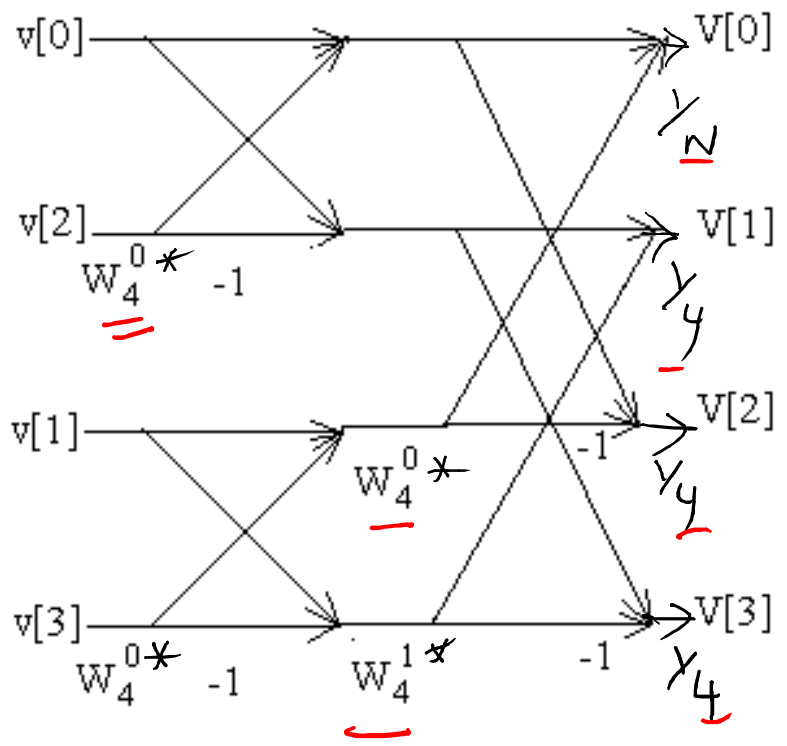


Y[K] → Y[N]

$$Y(k) = \{-12, 4, 0, 4\}^{2+2j}$$

\downarrow
 $y[n]$

$$x(n) = \frac{1}{N} \sum_{k=0}^{N-1} D_N^* X(k)$$



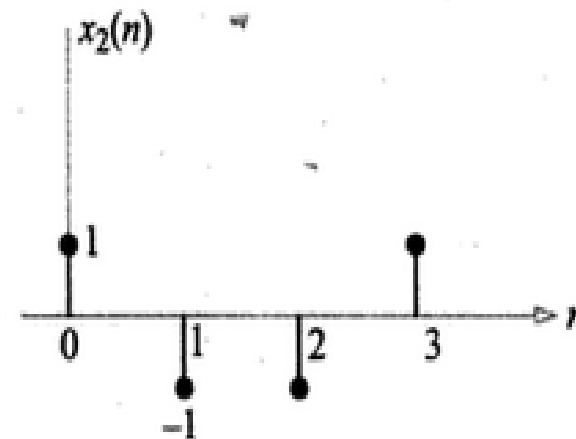
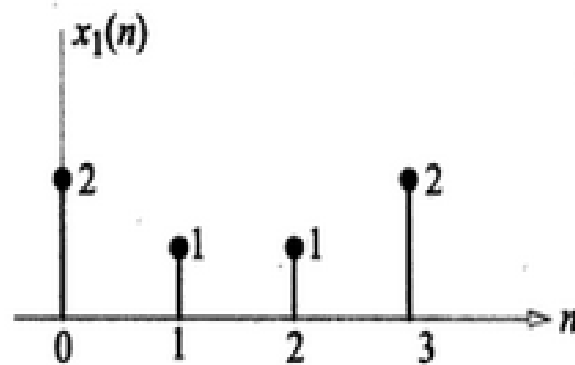
SOL:

- $X[k] = \{6, -2-j2, 2, -2+j2\}$
- $H[k] = \{-2, -1+j, 0, -1-j\}$
- $Y[k] = \{-12, 4, 0, 4\}$ ²⁺⁴
- $y[n] = \{-1, -3, -5, -3\}$ X



P.1

Given the sequences $x_1(n)$ and $x_2(n)$ below, compute the circular convolution $x_1(n) \circledast_N x_2(n)$ for $N = 4$. Use **DIT**-FFT algorithm.



SOL.1

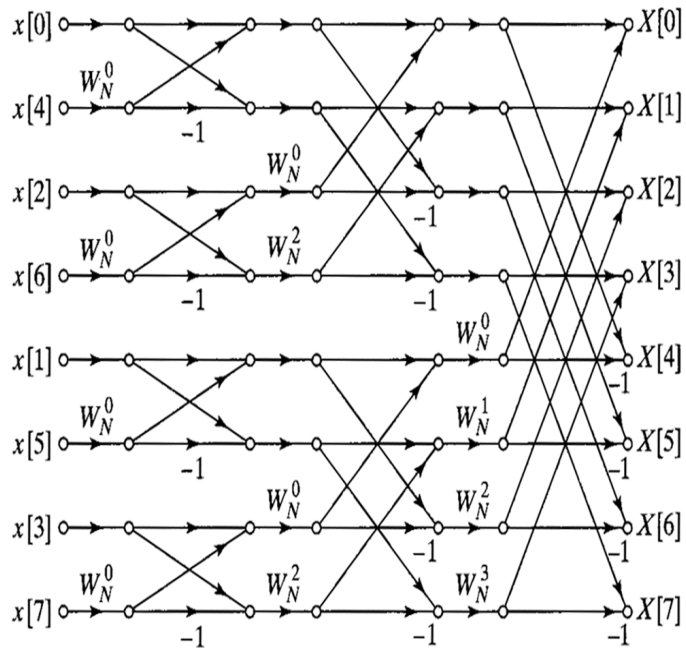
- $X1[k]=\{6, 1+j, 0, 1-j\}$
- $X2[k]=\{0, 2+2j, 0, 2-2j\}$
- $Y[k]=\{0, 4j, 0, -4j\}$
- $y[n]=\{0, -2, 0, 2\}$



COMPARING DIT AND DIF STRUCTURES:

The DIF FFT is the *transpose* of the DIT FFT

DIT FFT structure:



DIF FFT structure:

