UNIT-II

Fast Fourier Transforms: Fast Fourier Transforms (FFT) - Radix-2 decimation in time and decimation in frequency FFT algorithms, inverse FFT and FFT for Composite N.

Discrete Fourier Transform (DFT) is used for data analysis of repeated data. The DFT helps to identify and quantify the cyclic (Repeating) phenomenon. The computation of DFT involves the multiplication of a matrix by a vector.

DFT pair is given by

$$X(k) = \sum_{n=0}^{N-1} x(n)\, e^{\frac{-j2\pi kn}{N}} \quad ; \; k = 0 \text{ to } N-1$$

$$= \sum_{n=0}^{N-1} x(n)\, W_N^{kn} \quad ; \; k = 0 \text{ to } N-1 \qquad \longrightarrow (1)$$

$W_N^{kn} = e^{\frac{-j2\pi}{N}}$ is called twiddle factor.

IDFT is given by

$$x(n) = \frac{1}{N} \sum_{k=0}^{N-1} X(k)\, W_N^{-kn} \quad ; \; n = 0 \text{ to } N-1 \qquad \longrightarrow (2)$$

Both x(n) and X(k) are complex.

Computation of DFT & IDFT includes number of operations (additions and multiplications).

Eq (1) can be represented as

$$X(k) = \sum_{n=0}^{N-1} \left[Re(x(n)) + j\, Im\, [x(n)] \right] \left[Re(W_N^{nk}) + j Im \left(W_N^{nk} \right) \right]$$

(or)

$$= \sum_{n=0}^{N-1} \left[x_R(n) + j x_I(n) \right] \left[\cos\left(\frac{2\pi kn}{N}\right) - j \sin\left(\frac{2\pi kn}{N}\right) \right]$$

$$= \sum_{n=0}^{N-1} \left[x_R(n) \cos\left(\frac{2\pi kn}{N}\right) + x_I(n) \sin\left(\frac{2\pi kn}{N}\right) \right]$$

$$+ j \sum_{n=0}^{N-1} \left[x_I \cos\left(\frac{2\pi kn}{N}\right) - x_R(n) \sin\left(\frac{2\pi kn}{N}\right) \right] \quad \text{--- (3)}$$

From eq (3) Computing $X(k)$ involves 4N real multiplications i.e., for each value of 'k' there are 4- Multiplication terms. Since this multiplication must be extended to $n = 0, 1, 2, --(N-1)$ (totally N terms), hence totally 4N real multiplications.

Similarly each of the four sum of N terms requires N-1 real additions and two more to combine real part and imaginary part i.e., $4(N-1) + 2 = 4N - 2$ additions.

Direct Computation of $X(k)$ for N different values of k, of a sequence $x(n)$ requires $4N^2$ real multiplication and $N(4N-2)$ real additions. Alternatively there are N^2 Complex multiplications and $N(N-1)$ Complex additions.

As N increased or is large, the no of arithmetic operations (Multiplications and additions) required to Compute DFT by direct Method becomes Complex. i.e direct evaluation of DFT on the basis of it's definition (equation ①) entails a large amount of redundancy and through some properties/strategies, a huge reduction of computational complexity can be achieved. These strategies are collectively known as Fast Fourier Transforms (FFT's) algorithms for computing DFT.

These algorithms are also called Cooley-Tukey FFT algorithm. FFT algorithms exploits the two basic properties of twiddle factor.

$$\boxed{\begin{array}{ll} \text{Symmetry property :} & W_N^{k+\frac{N}{2}} = -W_N^k \\ \text{Periodicity property :} & W_N^{k+N} = W_N^k \end{array}} \quad \text{---} (4)$$

FFT algorithms are based on the fundamental principle of decomposing the computation of DFT of a sequence of length 'N' in to successively smaller DFT's. There are basically two types of FFT algorithms

(i) Decimation in Time (DIT-FFT)

(ii) Decimation in Frequency (DIF-FFT)

The efficiency of DFT algorithm can be improved based on the divide and conquer approach. This approach holds good only if the number 'N' of data points is not a prime.

* Decimation in Time FFT algorithm (DIT-FFT)

DIT-FFT algorithm is based on splitting (decimating) $x(n)$ into smaller sequences and finding $X(k)$ from DFTs of these decimated sequences.

$$N = r^{v}$$

If $r = 2$, radix-2 $\quad N = 2^{v}$, v is even integer.

Let $x(n)$ be a sequence of length N, where N is power of 2 (Radix-2 algorithm). Decimate this sequence in to two sequences of length $\frac{N}{2}$-points, where one sequence consists of even-indexed values of $x(n)$ and other sequence consists of odd-indexed sequence values.

i.e
$$x_e(n) = x(2n) \quad \text{for } n = 0, 1, 2 \cdots \left(\tfrac{N}{2}-1\right)$$
$$x_o(n) = x(2n+1) \quad \text{for } n = 0, 1, 2 \cdots \left(\tfrac{N}{2}-1\right)$$

N-point DFT is given by

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

Separating $x(n)$ in to even and odd indexed values of $x(n)$

$$X(k) = \sum_{n\text{-even}} x(n)\, W_N^{kn} + \sum_{n\text{-odd}} x(n)\, W_N^{kn} \quad ; \; k = 0 \text{ to } N-1$$

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

$; \; k = 0 \text{ to } N-1$

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

$$= \sum_{n=0}^{\frac{N}{2}-1} x_e(n) W_N^{2kn} + W_N^k \sum_{n=0}^{\frac{N}{2}-1} x_o(n) W_N^{2kn} \quad \text{—(5)}$$

$$W_N^{2kn} = \left(W_N^2\right)^{kn}$$

$$W_N^2 = e^{-\frac{j2\pi \cdot 2}{N}} = e^{-\frac{j2\pi}{N/2}} = W_{N/2}$$

i.e $\quad W_N^{2kn} = W_{N/2}^{nk}$

$$\Rightarrow X(k) = \underbrace{\sum_{n=0}^{\frac{N}{2}-1} x_e(n) W_{N/2}^{kn}}_{\text{N/2- point DFT of even indexed sequence}} + W_N^k \underbrace{\sum_{n=0}^{\frac{N}{2}-1} x_o(n) W_{N/2}^{kn}}_{\text{N/2- point odd indexed sequence}} \quad \text{—(6)}$$

$$X(k) = X_e(k) + W_N^k X_o(k) \quad \text{—(7)}$$

$k = 0$ to $N-1$

Eventhough index k ranges from 0 to $N-1$, each sum is computed only for $k=0$ to $\frac{N}{2}-1$ since $X_e(k)$ and $X_o(k)$ are periodic in k with period $\frac{N}{2}$.

i.e

$$X_e(k) = X_e\left(k+\frac{N}{2}\right)$$

$$X_o(k) = X_o\left(k+\frac{N}{2}\right)$$

$$W_N^{k+\frac{N}{2}} = -W_N^k$$

$\therefore$ Equation (7) becomes

$$X(k) = X_e(k) + W_N^k X_o(k)$$

for $k = 0$ to $\frac{N}{2} - 1$

and for $k \geq \frac{N}{2}$

$$X\left(k+\frac{N}{2}\right) = X_e\left(k+\frac{N}{2}\right) + W_N^{k+\frac{N}{2}} X_o\left(k+\frac{N}{2}\right)$$

$$X(k) = X_e(k) - W_N^k X_o(k)$$

$k = 0$ to $\frac{N}{2} - 1$

(or)

$$X\left(k+\frac{N}{2}\right) = X_e(k) - W_N^k X_o(k) \quad k = 0 \text{ to } \frac{N}{2} - 1$$

$$\therefore \boxed{\begin{aligned} X(k) &= X_e(k) + W_N^k X_o(k) \quad ; k = 0 \text{ to } \tfrac{N}{2} - 1 \\ X\left(k+\tfrac{N}{2}\right) &= X_e(k) - W_N^k X_o(k) \quad ; k = 0 \text{ to } \tfrac{N}{2} - 1 \end{aligned}}$$

for $N = 8$;

$k = 0$

$$X(0) = X_e(0) + W_8^0 X_o(0)$$

$$X(1) = X_e(1) + W_8^1 X_o(0)$$

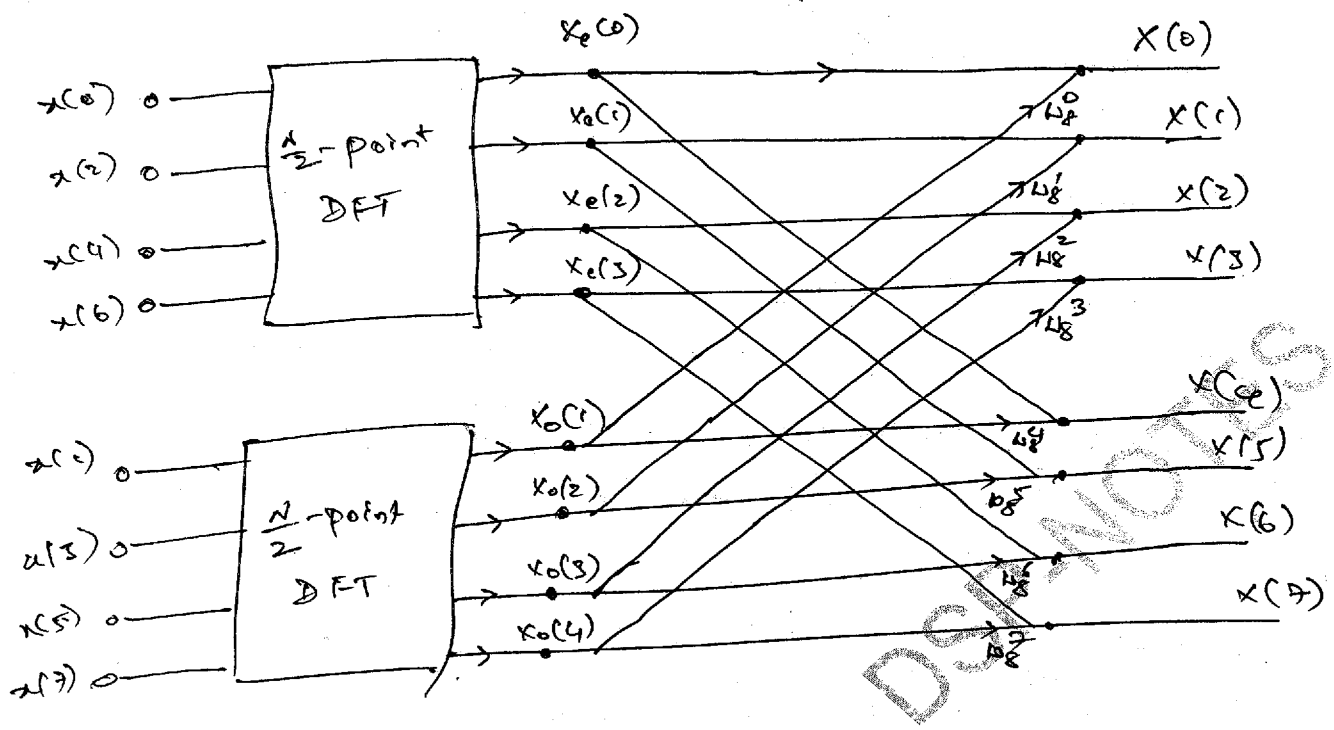
$$X(2) = X_e(2) + W_8^2 X_o(2)$$

$$X(3) = X_e(3) + W_8^3 X_o(3)$$

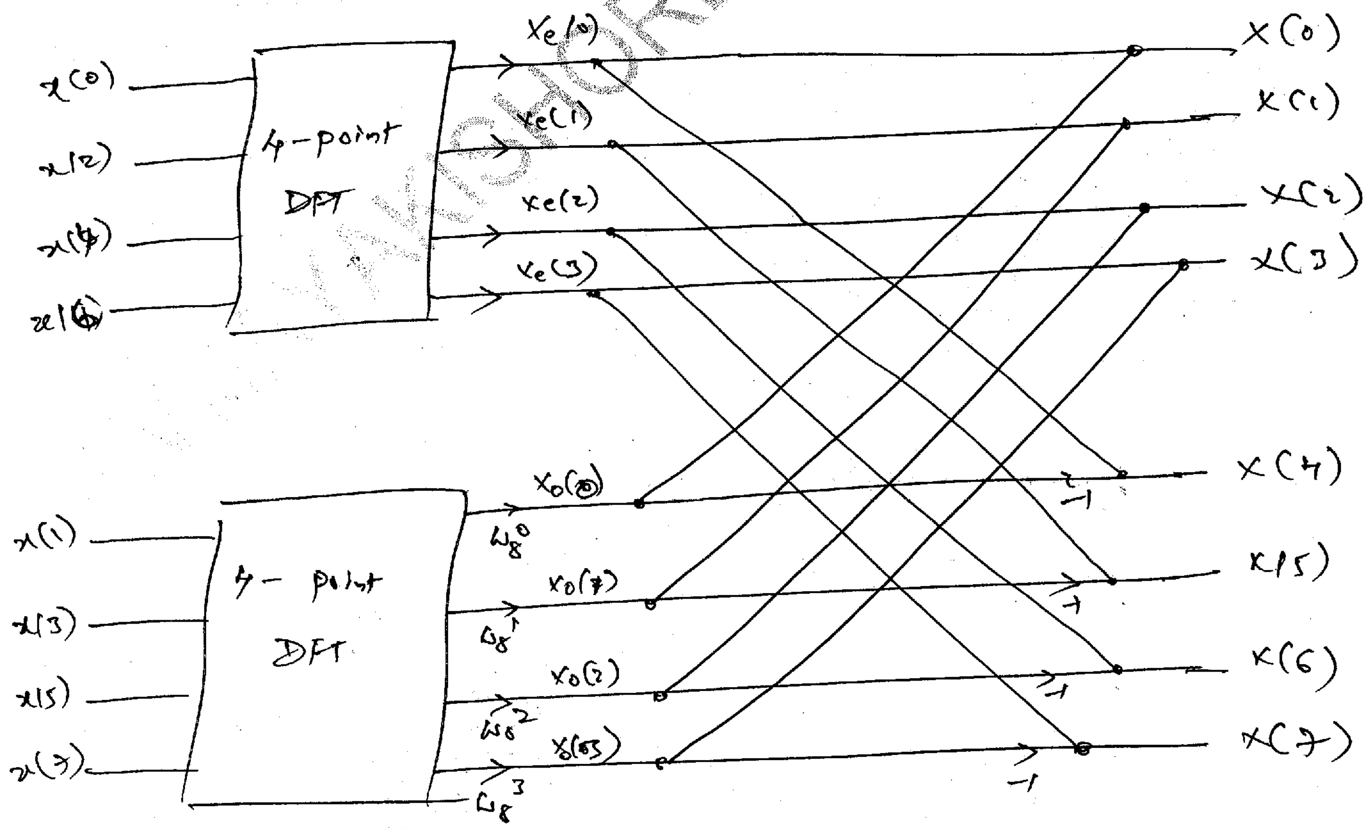
using periodicity property.

$$\left\{\begin{aligned} X(4) &= X_e(4) + W_8^4 X_o(4) = X_e(0) - W_8^1 X_o(0) \\ X(5) &= X_e(5) + W_8^5 X_o(5) = X_e(1) - W_8^2 X_o(1) \\ X(6) &= X_e(6) + W_8^6 X_o(6) = X_e(2) - W_8^3 X_o(2) \\ X(7) &= X_e(7) + W_8^7 X_o(7) = X_e(3) - W_8^4 X_o(3) \end{aligned}\right.$$

8-point DIT-FFT algorithm after first decimation is shown below.



(or)



Computations

For N-point DFT direct evaluation.

Complex multiplications - N^2

Complex additions - $N(N-1)$

For $\frac{N}{2}$ - point DFT

Complex multiplications - $\left(\frac{N}{2}\right)^2$

Complex additions $\frac{N}{2}\left(\frac{N}{2}-1\right)$

$X_e(k)$ and $X_o(k)$ are two $\frac{N}{2}$-point DFTs

i.e for $X_e(k)$ Complex multiplications and additions.

are $\left(\frac{N}{2}\right)^2$ and $\frac{N}{2}\left(\frac{N}{2}-1\right)$

'lly for $X_o(k)$ Complex multiplications and additions

are $\left(\frac{N}{2}\right)^2$ and $\frac{N}{2}\left(\frac{N}{2}-1\right)$

$X_e(k)$ and $W_N^k X_o(k)$ must be added to obtain $X(k)$

$\therefore$ Total Complex multiplications after 1st level decomposition

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

Total Complex additions after 1st level decomposition

$$\frac{N}{2}\left(\frac{N}{2}-1\right) + \frac{N}{2}\left(\frac{N}{2}-1\right) + N = \frac{N^2}{2} \cong N + \frac{N^2}{2}$$

* If we decompose N- point DFT into two $\frac{N}{2}$ - point DFTs the no of Computations are reduced by factor 2.

2^{nd} level decomposition

If $\frac{N}{2}$ is even, $x_e(n)$ and $x_o(n)$ may again be decimated.

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

$$= \sum_{n=odd} x_e(n)\, W_{N/2}^{kn} + \sum_{n=even} x_e(n)\, W_{N/2}^{kn}$$

$$= \underbrace{\sum_{n=0}^{\frac{N}{4}-1} x_e(2n)\, W_{N/4}^{kn}}_{\frac{N}{4}\text{- point DFT}} + W_{N/2}^{k} \underbrace{\sum_{n=0}^{\frac{N}{2}-1} x_e(2n+1)\, W_{N/4}^{kn}}_{\frac{N}{4}\text{- point DFT}}$$

Ily
$$X_o(k) = \sum_{n=0}^{\frac{N}{4}-1} x_o(2n)\, W_{N/4}^{kn} + W_{N/2}^{k} \sum_{n=0}^{\frac{N}{4}-1} x_o(2n+1)\, W_{N/4}^{kn}$$

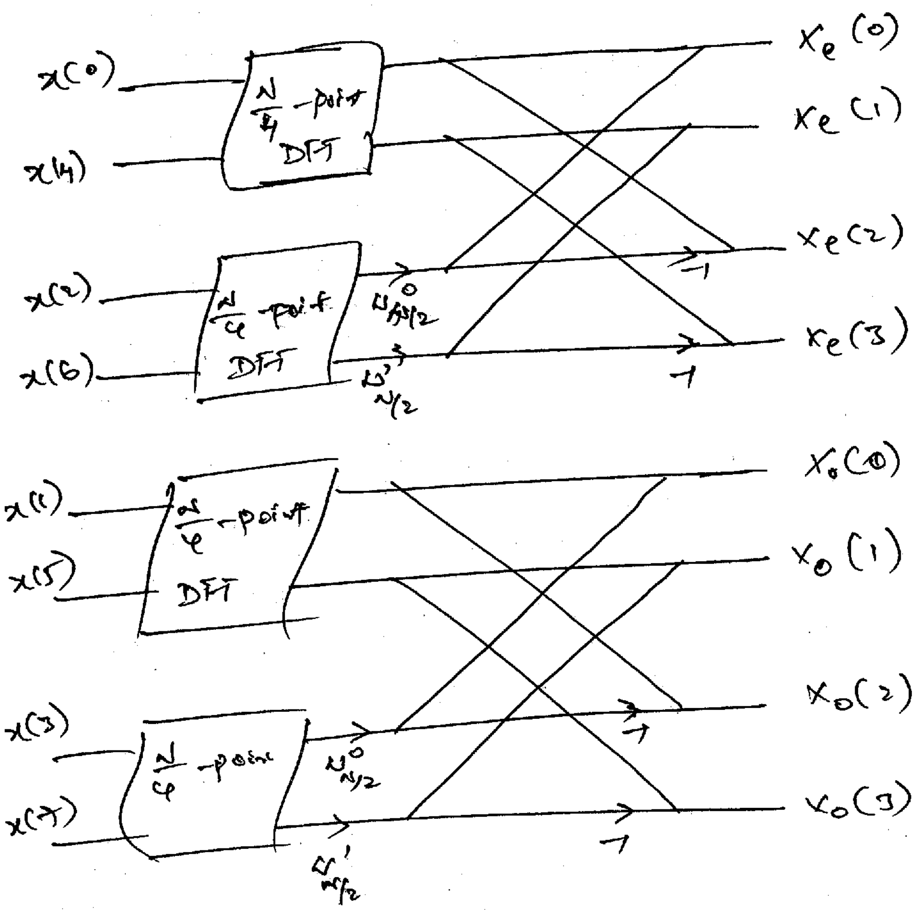
$$\Rightarrow \boxed{\begin{aligned} X_e(k) &= X_{ee}(k) + W_{N/2}^{k} X_{eo}(k) \\ X_e\left(k+\tfrac{N}{4}\right) &= X_{ee}(k) - W_{N/2}^{k} X_{eo}(k) \end{aligned}} \quad k = 0 \text{ to } \frac{N}{4}-1$$

and

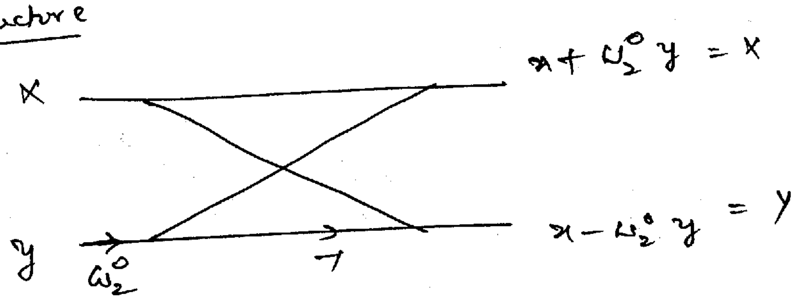
$$\boxed{\begin{aligned} X_o(k) &= X_{oe}(k) + W_{N/2}^{k} X_{oo}(k) \\ X_o\left(k+\tfrac{N}{4}\right) &= X_{oe}(k) - W_{N/2}^{k} X_{oo}(k) \end{aligned}} \quad k = 0 \text{ to } \frac{N}{4}-1$$

The decimation may be continued until there are only two-point DFTs.

II^{nd} level decimation for $N=8$



Two point DFT structure is shown below known as <u>Butterfly structure</u>



Each butterfly involves one complex multiplication and two complex additions.

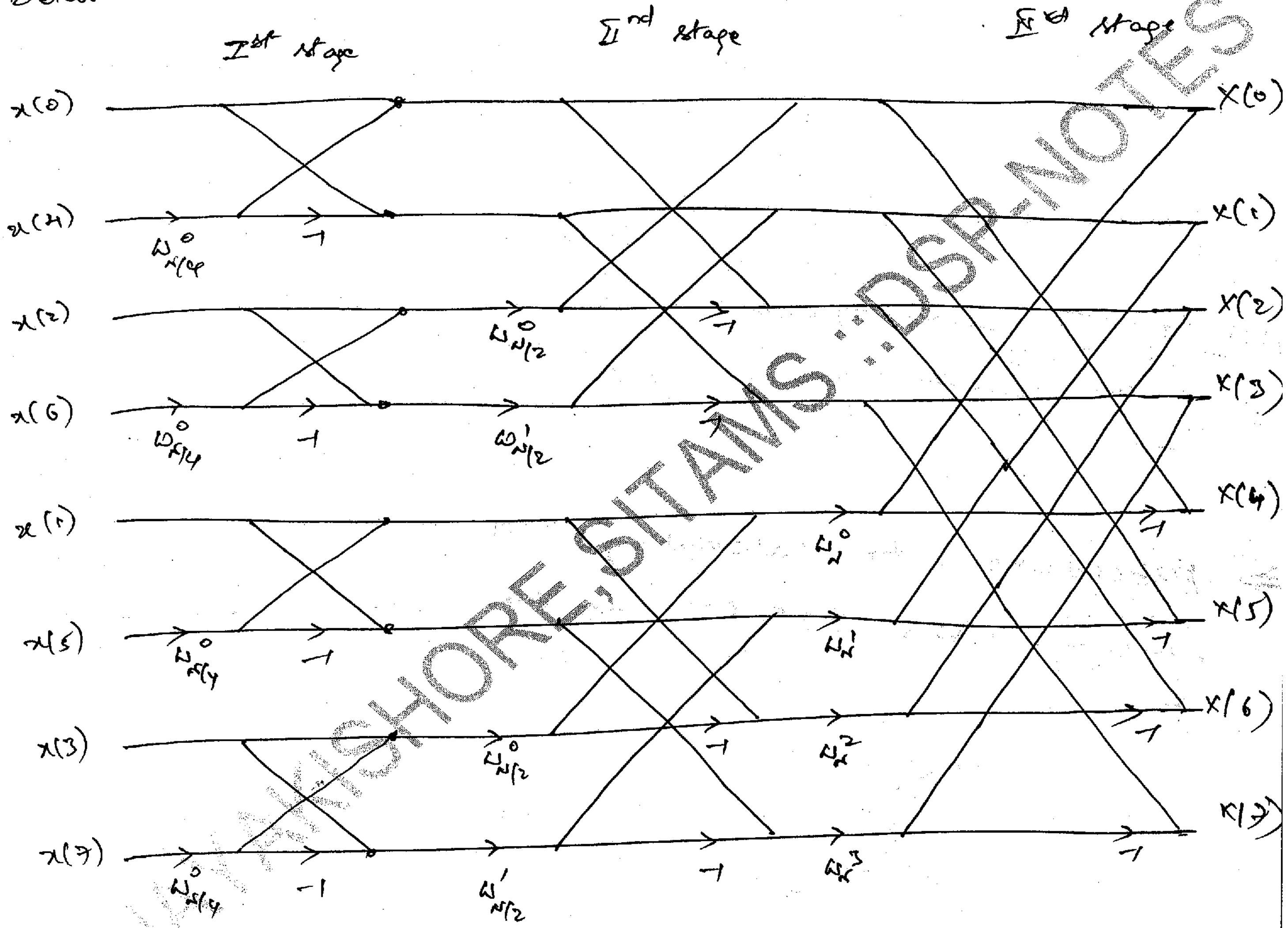
In radix-2 algorithm, there are $\frac{N}{2}$ butterfly per stage of computation process and $\log_2 N$ stages.

$\therefore$ total no. of complex multiplications $- \frac{N}{2} \log_2 N$

complex additions $- N \log_2 N$

$\Rightarrow$ Regarding memory requirement of structure, the complex from x and y produce complex X and Y. These are stored in the same location of x and y. Hence for entire process, fixed amount of memory is sufficient. The o/p is placed in i/p location itself, so computation is also known as <u>Inplace computation</u>

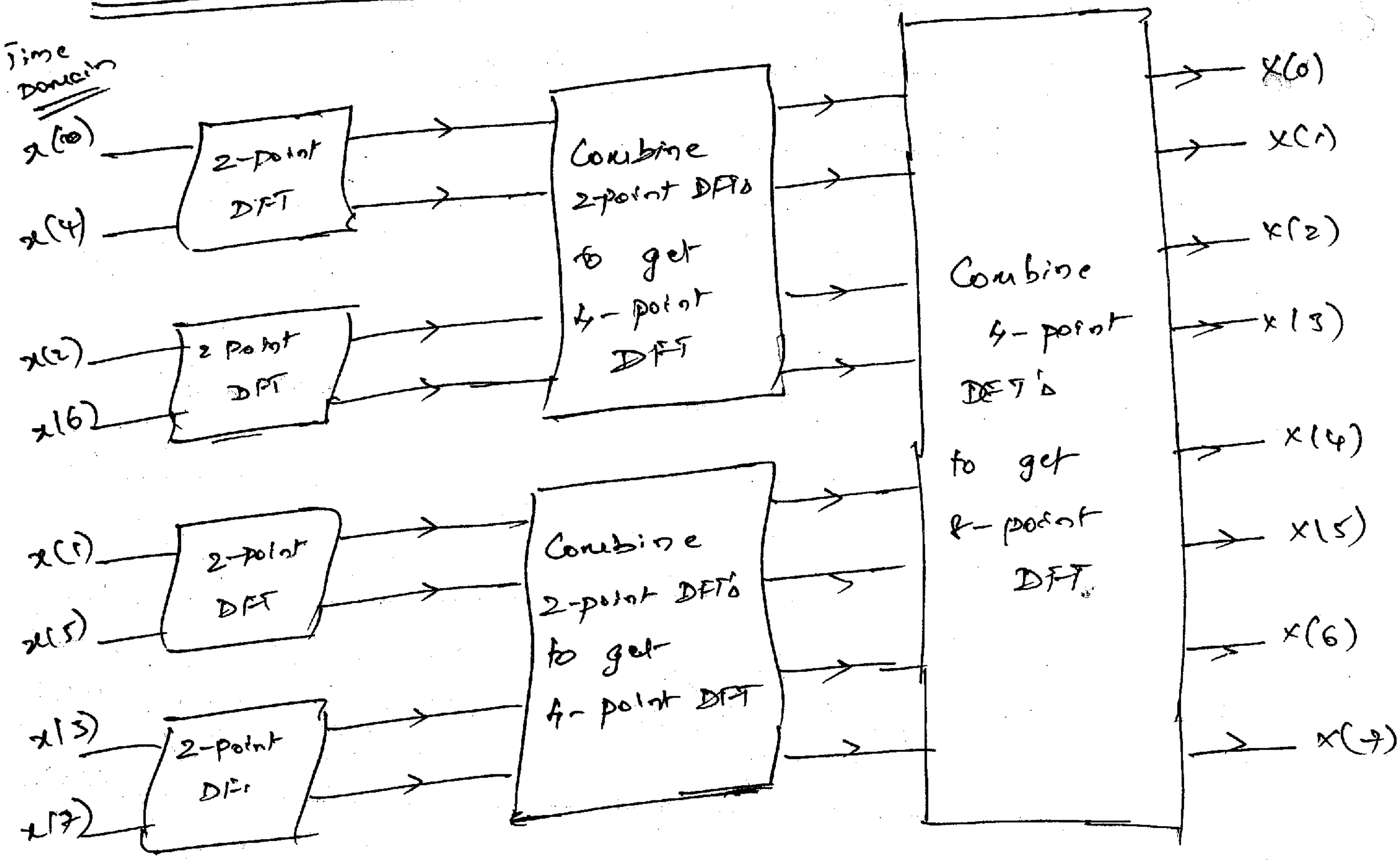
Complete N-point DIT-FFT DFT structure (N=8) is shown below.



When data are decimated at each level, the shuffling takes place in the order form. Results correspond to a bit-reversed indexing of the original sequence. (Bit Reversed structure)

000	x(0)	x(0)	x(0)	000
001	x(1)	x(2)	x(4)	100
010	x(2)	x(4)	x(2)	010
011	x(3)	x(6)	x(6)	110
100	x(4)	x(1)	x(1)	001
101	x(5)	x(3)	x(5)	101
110	x(6)	x(5)	x(3)	011
111	x(7)	x(7)	x(7)	111

DIT - Radix-2 FFT :-



⇒ Procedure for Calculating DFT of a given sequence using DIT FFT algorithm.

1. Draw the decimation in time FFT structure
2. Calculate the weight parameters W.
3. I/p the data sequence in bit-reversed order to 1st stage.
4. Obtain the values at every stage and propogate it to the last stage to obtain DFT.

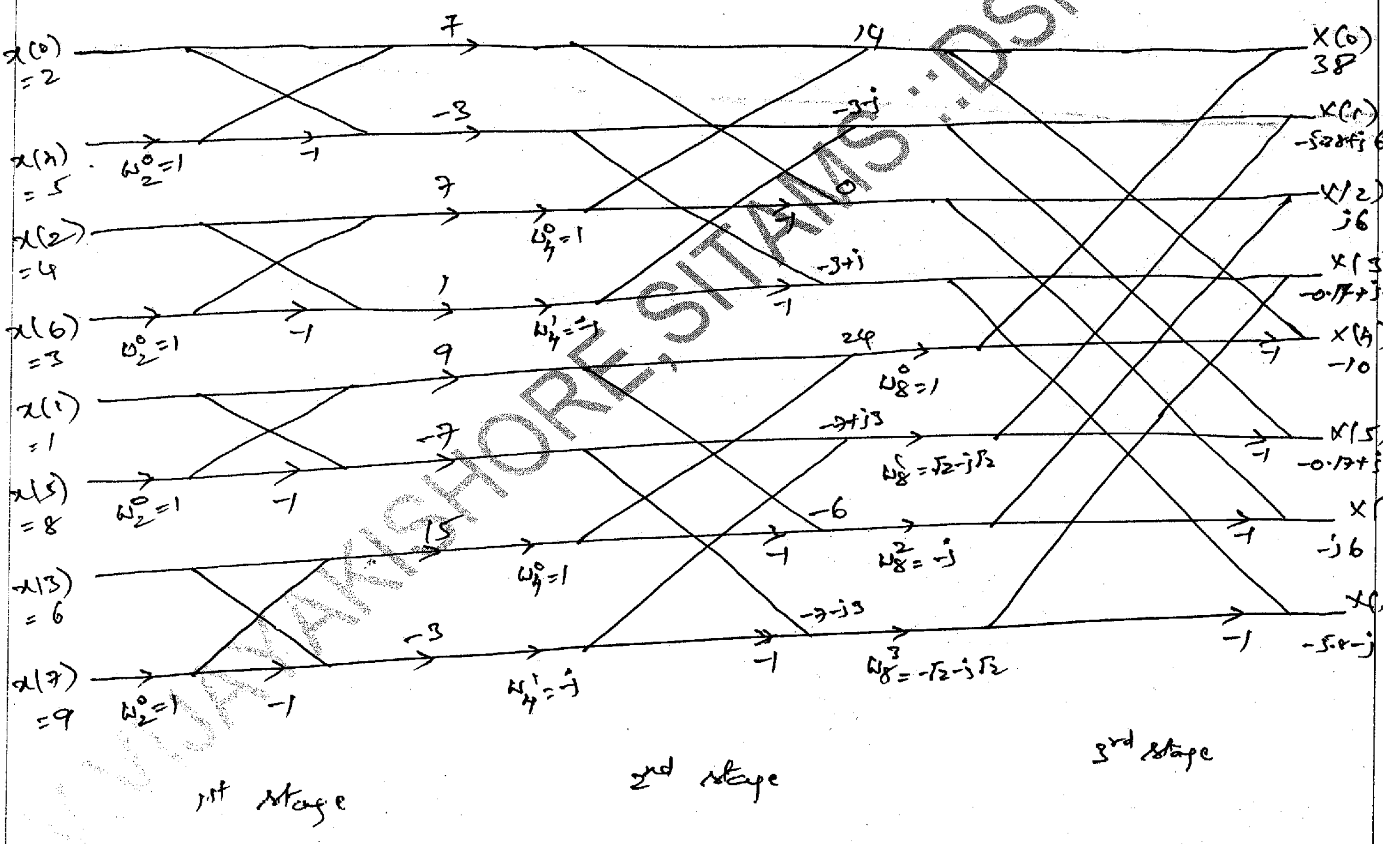
Prob: Determine the DFT of the given data sequence $x(n) = \{2, 1, 4, 6, 5, 8, 3, 9\}$
0 1 2 3 4 5 6 7

Sol:

$N = 8$

In DIT-FFT structure, the i/p data sequence must be given in bit-reversal fashion. The o/p DFT of $x(n)$ is in correct order.

Bit reversal order is $x'(n) = \{2, 5, 4, 3, 1, 8, 6, 9\}$



$X(k) = \{38, -5.828+j6.07, j6, -0.172+j8.07, -10, -0.172-j8.07, -j6, -5.828-j607\}$

16-point DIT-FFT :-

Indexes

$x(n) = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15\}$

1st level decimation $\left(\frac{N}{2}\right)$

even indexed — $\{0, 2, 4, 6, 8, 10, 12, 14\}$

odd indexed — $\{1, 3, 5, 7, 9, 11, 13, 15\}$

2nd level decimation $\left(\frac{N}{4}\right)$

even index — $\{0, 4, 8, 12\}$ odd index — $\{2, 6, 10, 14\}$

even index — $\{1, 5, 9, 13\}$ odd index — $\{3, 7, 11, 15\}$

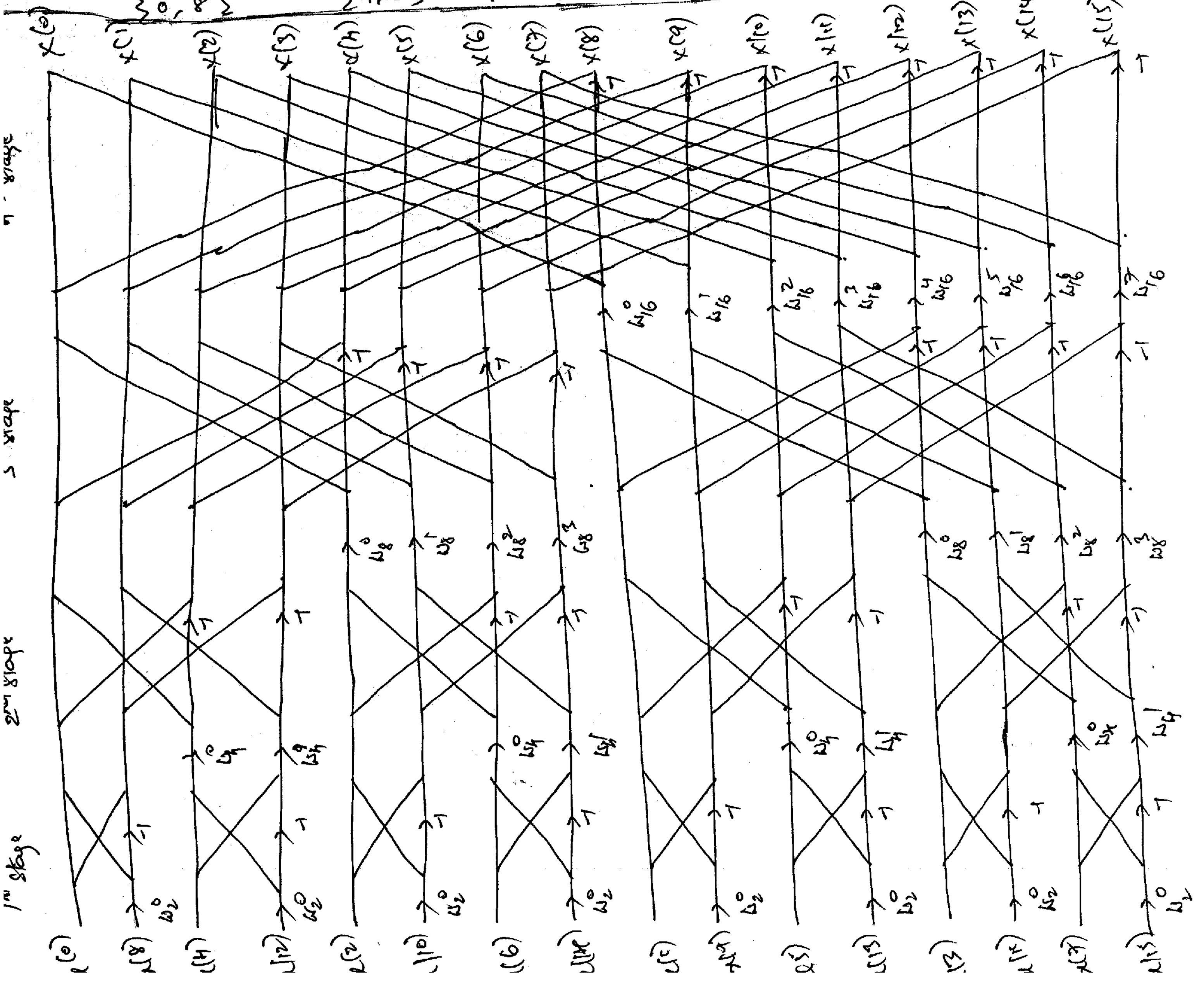
3rd level $\left(\frac{N}{8}\right)$ decimation

even index — $\{0, 8\}$ odd index — $\{4, 12\}$

even index — $\{2, 10\}$ odd index — $\{6, 14\}$

even index — $\{1, 9\}$ odd index — $\{5, 13\}$

even index — $\{3, 11\}$ odd index — $\{7, 15\}$



Decimation in Frequency FFT algorithm (DIF- FFT)

DIF -FFT is another method derived by decimating o/p sequence $X(k)$ into smaller subsequences.

$N = 2^{v}$;

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

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

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

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

$$= \sum_{n=0}^{\frac{N}{2}-1} x(n)\, W_N^{nk} + (-1)^k \sum_{n=0}^{\frac{N}{2}-1} x(n+\tfrac{N}{2})\, W_N^{kn} \qquad \because W_N^{\frac{N}{2}k} = (-1)^k$$

$$= \sum_{n=0}^{\frac{N}{2}-1} \left[x(n) + (-1)^k\, x(n+\tfrac{N}{2}) \right] W_N^{kn} \quad \text{—(8)}$$

Now decimating $X(k)$ into odd and even indexed samples.

* Even decimation *

$$X(2k) = \sum_{n=0}^{\frac{N}{2}-1} \left(x(n) + x(n+\tfrac{N}{2}) \right) W_N^{2kn} \quad ; \quad k = 0 \text{ to } \tfrac{N}{2}-1$$

$$= \sum_{n=0}^{\frac{N}{2}-1} \left(x(n) + x(n+\tfrac{N}{2}) \right) W_{N/2}^{nk} \quad ; \quad k = 0 \text{ to } \tfrac{N}{2}-1 \quad \text{—(9)}$$

* Odd decimation *

$$X(2k+1) = \sum_{n=0}^{\frac{N}{2}-1} \left[x(n) - x(n+\tfrac{N}{2}) \right) W_N^{n(2k+1)} \quad ; \quad k = 0 \text{ to } \tfrac{N}{2}-1$$

$$= \sum_{n=0}^{\frac{N}{2}-1} \left\{ \left[\left(x(n) - x(n+\tfrac{N}{2}) \right) W_N^{n} \right] W_{N/2}^{kn} \right. \quad ; \quad k = 0 \text{ to } \tfrac{N}{2}-1 \quad \text{—(10)}$$

Eqns (9) and (10) can be redefined as

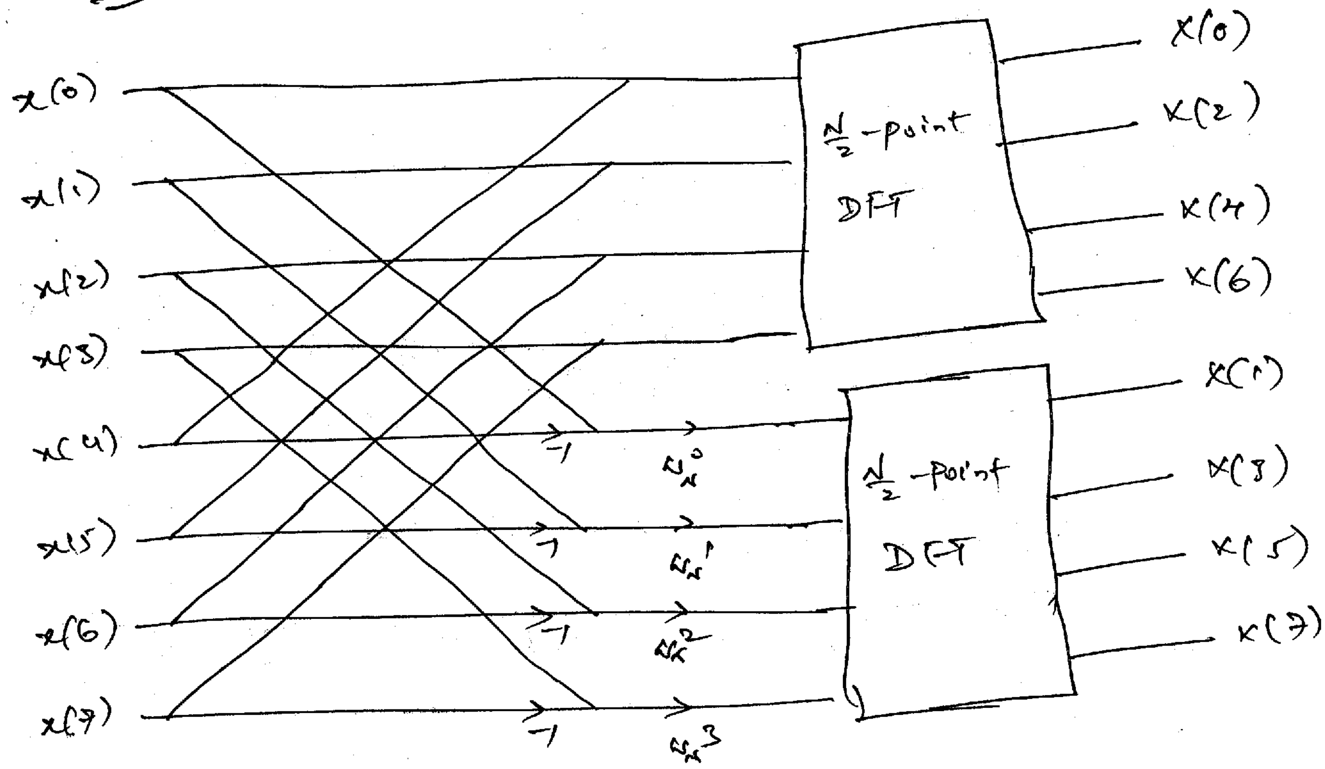
$$X(2k) = \sum_{n=0}^{\frac{N}{2}-1} g_1(n)\, W_{N/2}^{nk} \quad ; \quad k = 0 \text{ to } \frac{N}{2}-1$$

$$g_1(n) = x(n) + x\left(n+\frac{N}{2}\right) \qquad -(11)$$

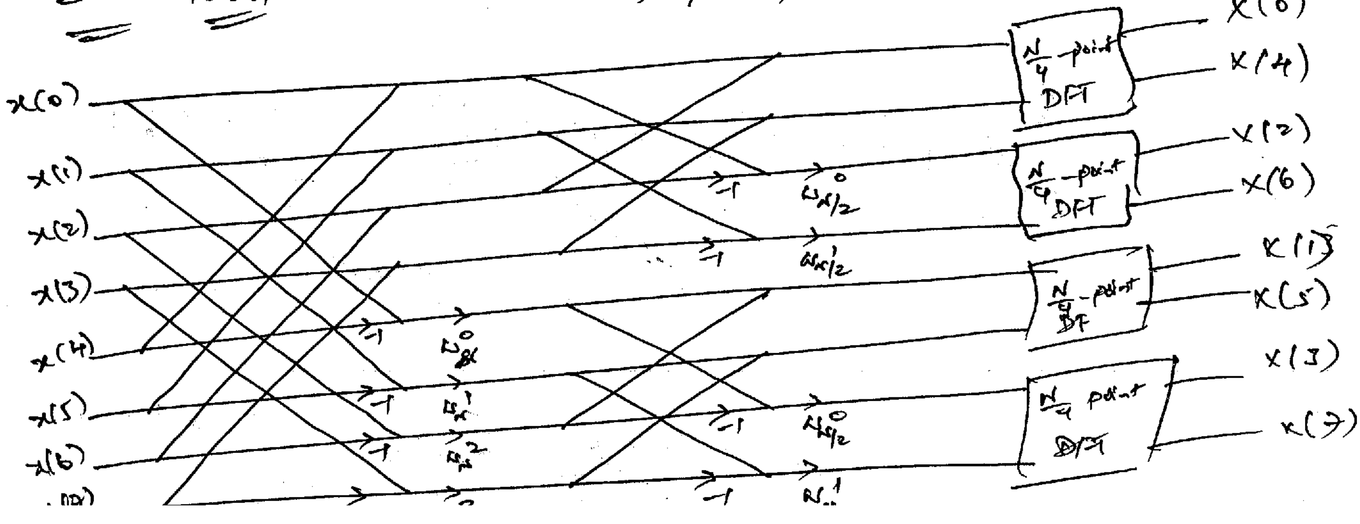
$$X(2k+1) = \sum_{n=0}^{\frac{N}{2}-1} g_2(n)\, W_{N/2}^{nk} \quad ; \quad k = 0 \text{ to } \frac{N}{2}-1$$

$$g_2(n) = x(n) - x\left(n+\frac{N}{2}\right) \qquad -(12)$$

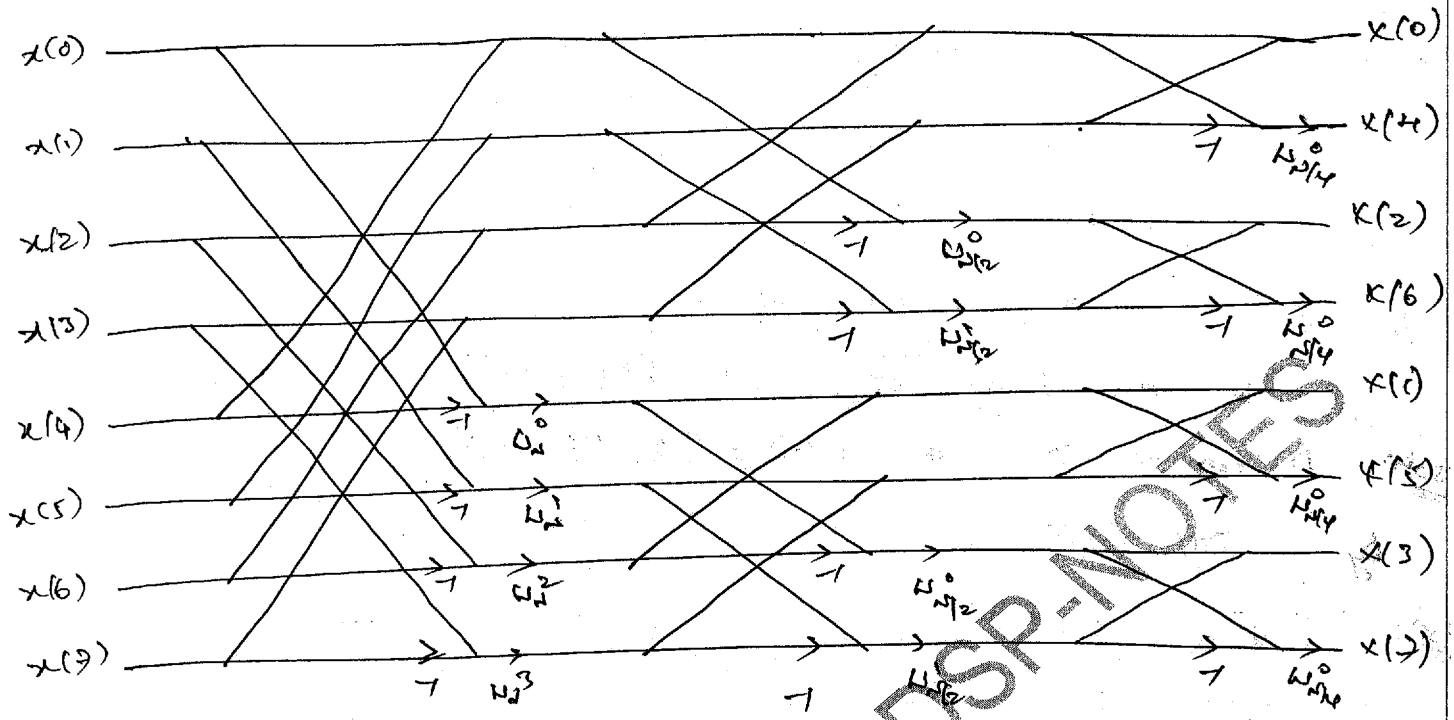
I st level decimation for N=8



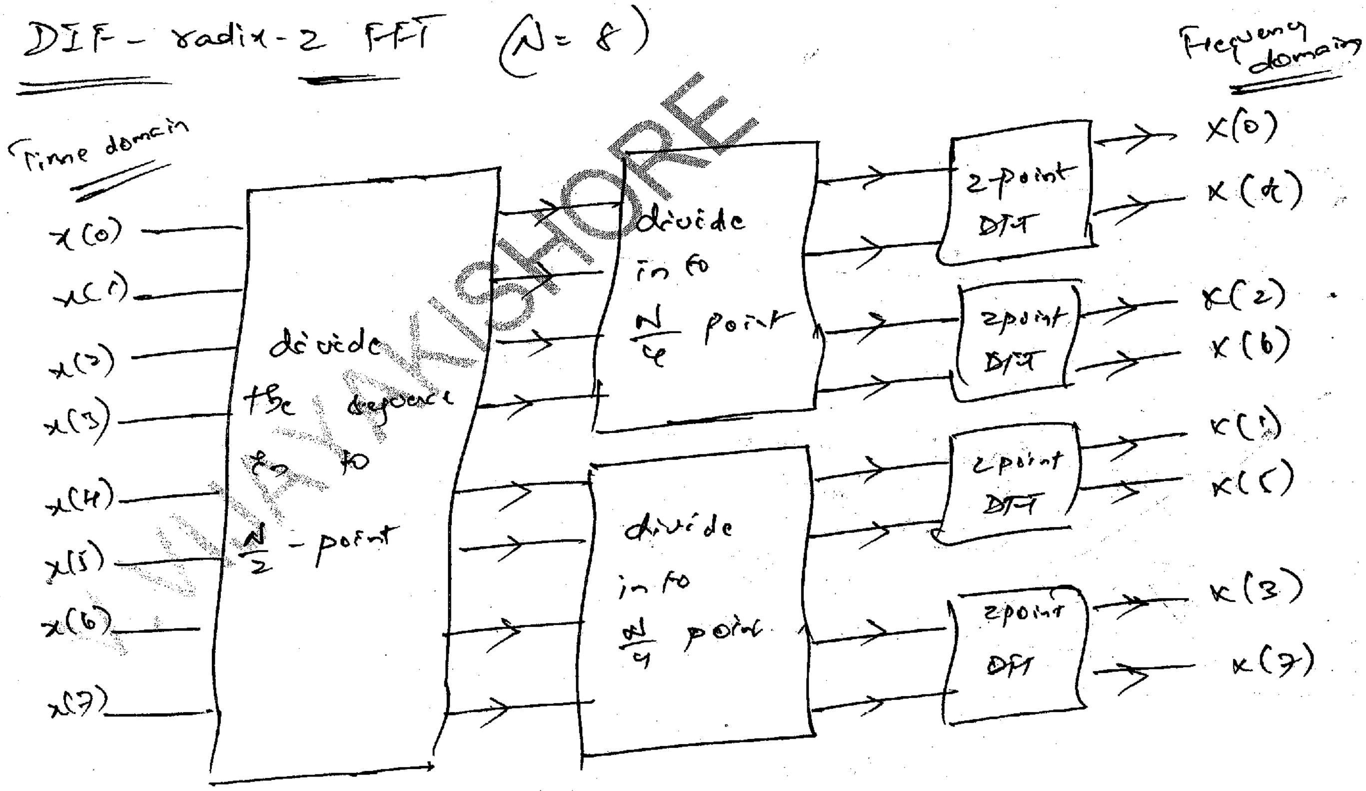
II nd level decimation in frequency



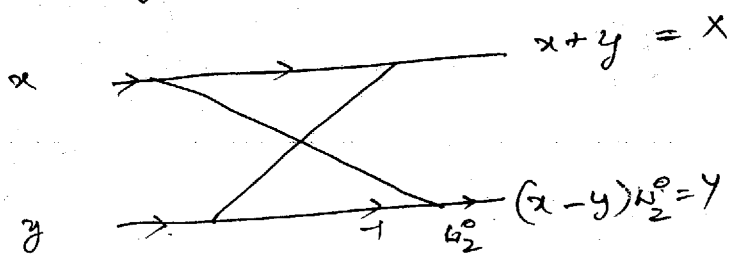
Complete DIF-FFT structure for N=8



DIF - radix-2 FFT (N = 8)



Basic Butterfly structure for DIF-FFT

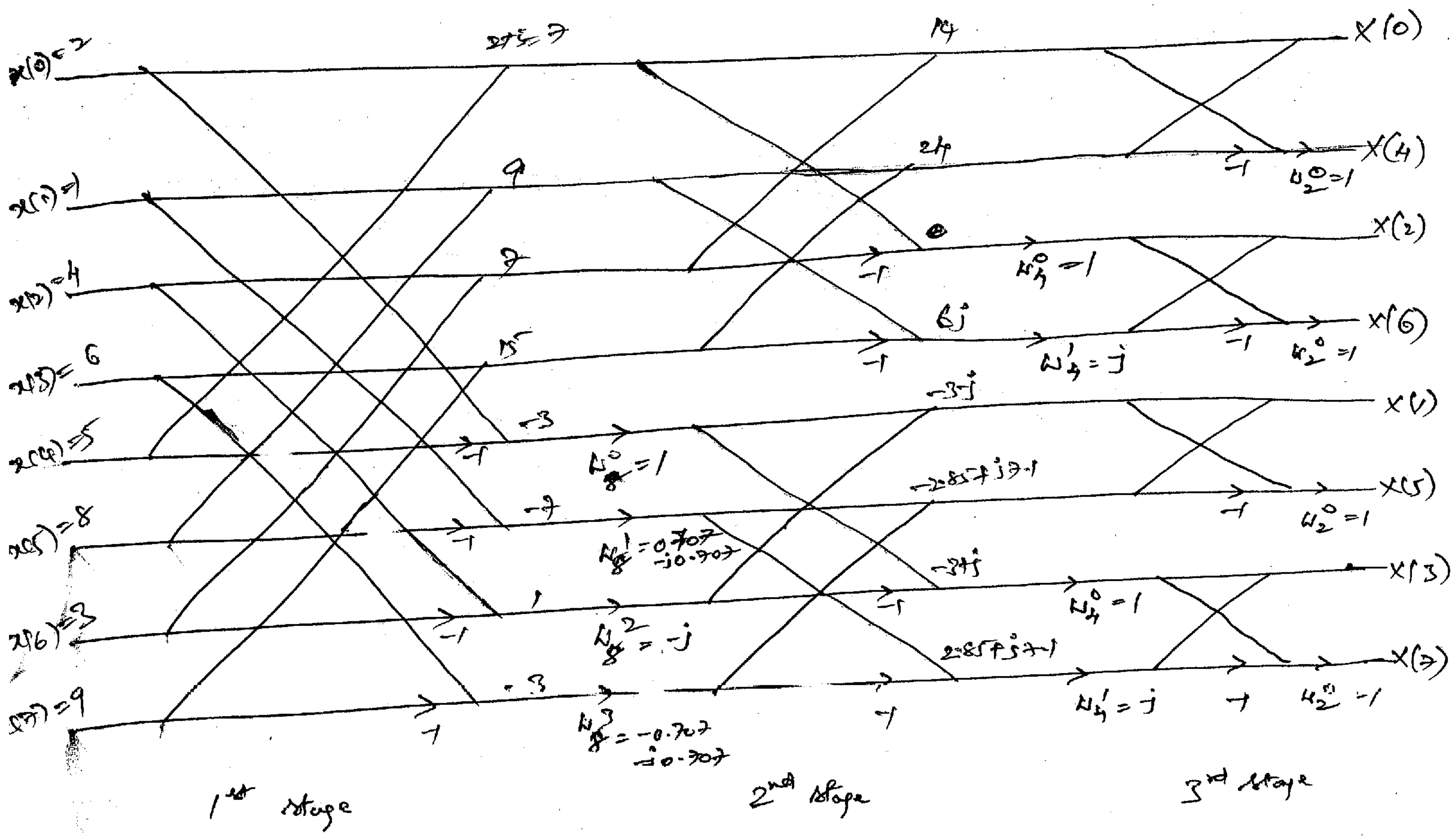


Procedure for calculating DFT of a given sequence using DIF-FFT algorithm

1. Draw the decimation in frequency FFT structure
2. Calculate the weight parameters
3. Input the data sequence to the 1st stage
4. Obtain the values at every stage and propogate it to the last stage to obtain DFT.

Prob: Determine DFT of $x(n) = \{2, 1, 4, 6, 5, 8, 3, 9\}$ by DIF - FFT

Sol^n: $x(n) = \{2, 1, 4, 6, 5, 8, 3, 9\}$ $N = 8$



$X(k) = \{38, -5.82+j6.1, j6, -0.142+j8.1, -10, -0.142-j8.1, -6j, -5.82-j6.1\}$

Comparision b/n DIT and DIF FFT algorithms

DIT	DIF
1. Time domain sequence $x(n)$ is decimated	1. Frequency domain sequence $X(k)$ is decimated.
2. I/p sequence $x(n)$ is in bit reversal order and o/p DFT sequence $X(k)$ is in natural order	2. I/p sequence $x(n)$ is in natural order and DFT sequence $X(k)$ is in bit-reversal order
3. Complex multiplication is done before complex addition & subtraction	3. Complex multiplication is done after complex addition & subtraction.

⇒ Similarities b/n DIT and DIF

1. Both algorithms require same no. of complex multiplications $\left(\frac{N}{2}\log_2^N\right)$ and complex additions $(N\log_2^N)$
2. Both algorithms can be used to compute DFT & IDFT with same no. of stages $= \log_2^N$
3. Both of them uses bit reversal algorithm.
4. Reduced memory requirement by storing the data o/ps in the original location (i.e in place computations).

⇒ Applications of FFT algorithms

1. Linear filtering
2. Correlation
3. Spectrum analysis
4. Efficient means to compute DFT

16-point DIF-FFT

i/p $x(n)$ in natural order

DFT $X(k)$ is in Bit reversal order

(Refer Page No: for indexes of $X(k)$ same as $x(n)$)

Inverse Fast Fourier Transform (IFFT)

IFFT can be obtained from same structures which are used to calculate FFT using DIT-FFT and DIF-FFT.

$$DFT[x(n)] = X(k) = \sum_{n=0}^{N-1} x(n)\, e^{\frac{-j2\pi kn}{N}} \quad ; k = 0 \text{ to } N-1$$

$$IDFT[X(k)] = x(n) = \frac{1}{N} \sum_{k=0}^{N-1} X(k)\, e^{\frac{j2\pi kn}{N}} \quad ; n = 0 \text{ to } N-1$$

$$N \cdot x(n) = \sum_{k=0}^{N-1} X(k)\, W_N^{-kn} \quad ; n = 0 \text{ to } N-1$$

apply conjugation on both sides.

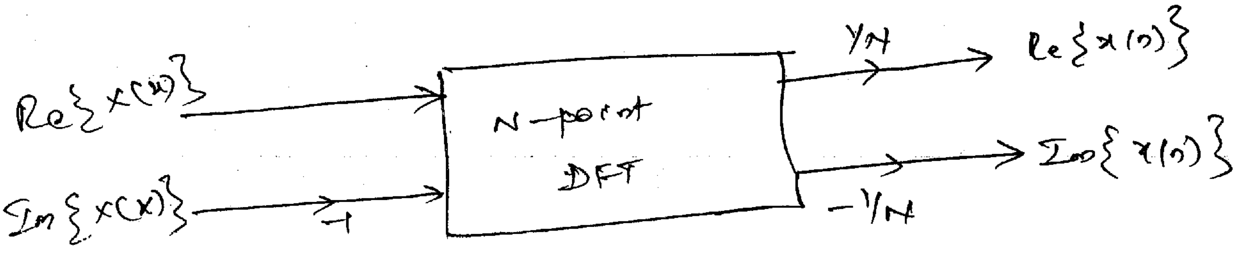
$$N \cdot x^*(n) = \sum_{k=0}^{N-1} X^*(k)\, W_N^{kn} \quad ; n = 0 \text{ to } N-1$$

The above equation can be assumed as DFT of signal $X^*(k)$ and can be computed using any one of the FFT algorithms.

The desired IDFT $x(n)$ is then obtained as

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

i.e Given an N-point DFT $X(k)$, we first form its complex conjugate sequence $X^*(k)$, then compute N-point DFT of $X^*(k)$ and obtain complex conjugate of DFT computed and finally divide each sample by N.

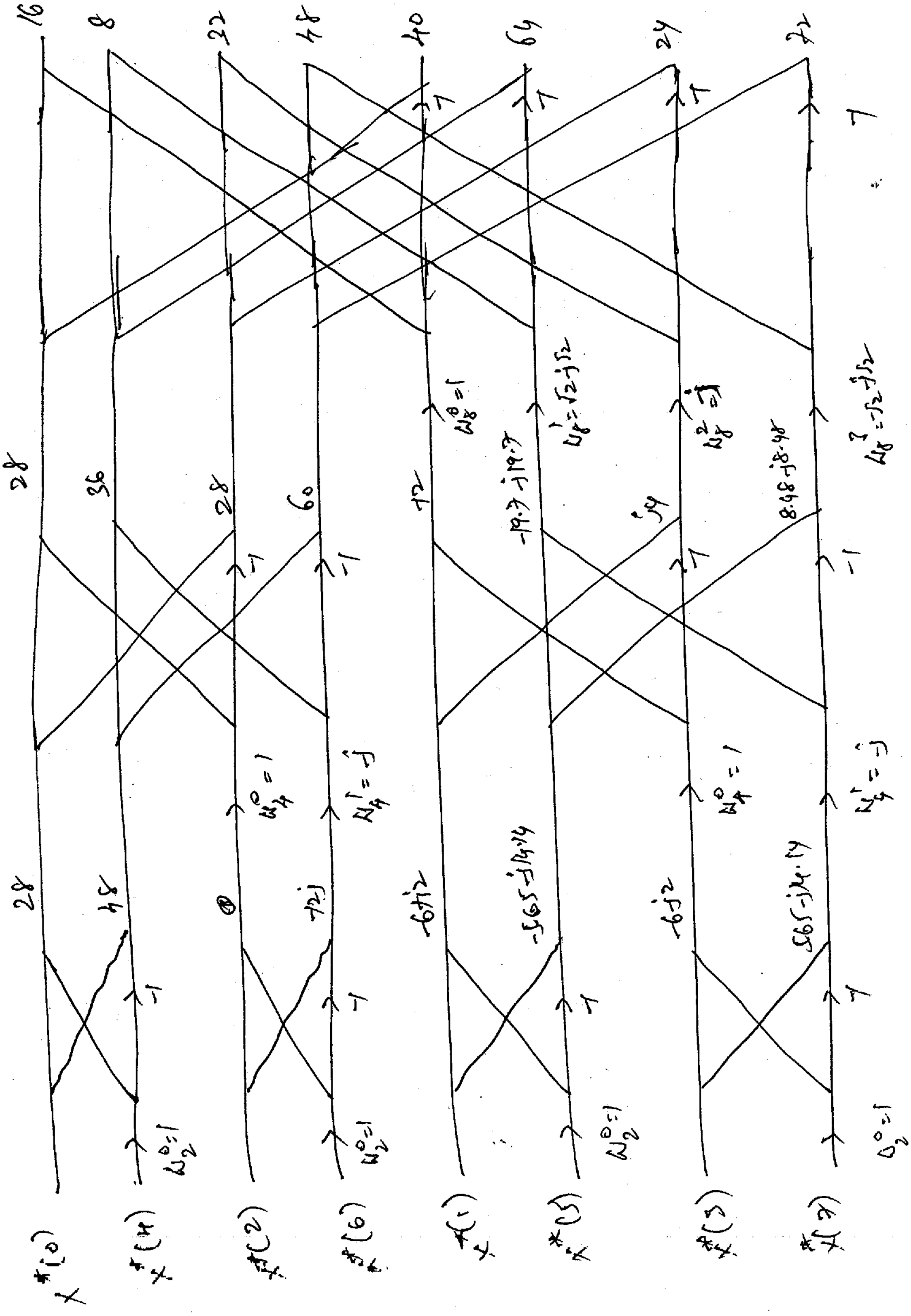


Prob: Calculate the IFFT for given co-efficients

$X(k) = \{38, -5.82+j6.07, j6, -0.172+j8.07, -10, -0.172-j8.07, -j6, -5.82-j6.07\}$ using DIT FFT structure

Sol: Given $X(k) = \{38, -5.82+j6.07, j6, -0.172+j8.07, -10, -0.172-j8.07, -j6, -5.82-j6.07\}$

$X^*(k) = \{38, -5.82-j6.07, -j6, -0.172-j8.07, -10, -0.172+j8.07, +j6, -5.82+j6.07\}$



$x(n) = \frac{1}{8}\{16, 8, 32, 48, 40, 64, 24, 72\}$

$x(n) = \{2, 1, 4, 6, 5, 8, 3, 9\}$

Repeat above problem using DIF-FFT

⇒ FFT for Composite N (or) Composite Radix FFT :-

A Composite or mixed radix is used when 'N' is not a power of 2 (or) when N is a Composite number which has more than one prime factor. Because it is not always possible to work with sequences whose length is power of 2. Efficient DIT and DIF algorithms exists to Compute DFT even if N is not a power of 2.

N can be written as $N = m_1 m_2 \cdots m_r$

If $N = m_1 N_1$, where $N_1 = m_2 m_3 \cdots m_r$, the i/p sequence $x(n)$ can be separated into m_1 subsequences of N_1 elements each. Then the DFT can be written as

$$X(k) = \sum_{n=0}^{N_1-1} x(nm_1) W_N^{nm_1 k} + \sum_{n=0}^{N_1-1} x(nm_1+1) W_N^{(nm_1+1)k}$$

$$+ \dots + \sum_{n=0}^{N_1-1} x(nm_1 + m_1 - 1) W_N^{(nm_1+m-1)k}$$

Prob:- Develop a Radix-3 DIF FFT algorithm for evaluating the DFT for $N=9$.

Sol: $N = 9 = 3 \cdot 3$; $m_1 = 3$; $N_1 = 3$

$\therefore X(k)$ according to above equation.

$$X(k) = \sum_{n=0}^{2} x(3n) W_9^{3nk} + \sum_{n=0}^{2} x(3n+1) W_N^{(3n+1)k} + \sum_{n=0}^{2} x(3n+2) W_N^{(3n+2)k}$$

$$= X_1(k) + W_9^{k} X_2(k) + W_9^{2k} X_3(k)$$

$$X_1(k) = \sum_{n=0}^{2} x(3n) W_9^{3nk} = x(0) + x(3) W_9^{3k} + x(6) W_9^{6k}$$

$$X_2(k) = \sum_{n=0}^{2} x(3n+1) W_9^{(3n)k} = x(1) + x(4) W_9^{3k} + x(7) W_9^{6k}$$

$$X_3(k) = \sum_{n=0}^{2} x(3n+2) W_9^{3nk} = x(2) + x(5) W_9^{3k} + x(8) W_9^{6k}$$

$$\boxed{X_i(k+N_1) = X_i(k)}$$

$$X(0) = X_1(0) + W_9^0 X_2(0) + W_9^0 X_3(0)$$

$$X(1) = X_1(1) + W_9^1 X_2(1) + W_9^2 X_3(1)$$

$$X(2) = X_1(2) + W_9^2 X_2(2) + W_9^4 X_3(2)$$

$$X(3) = X_1(0) + W_9^3 X_2(0) + W_9^6 X_3(0)$$

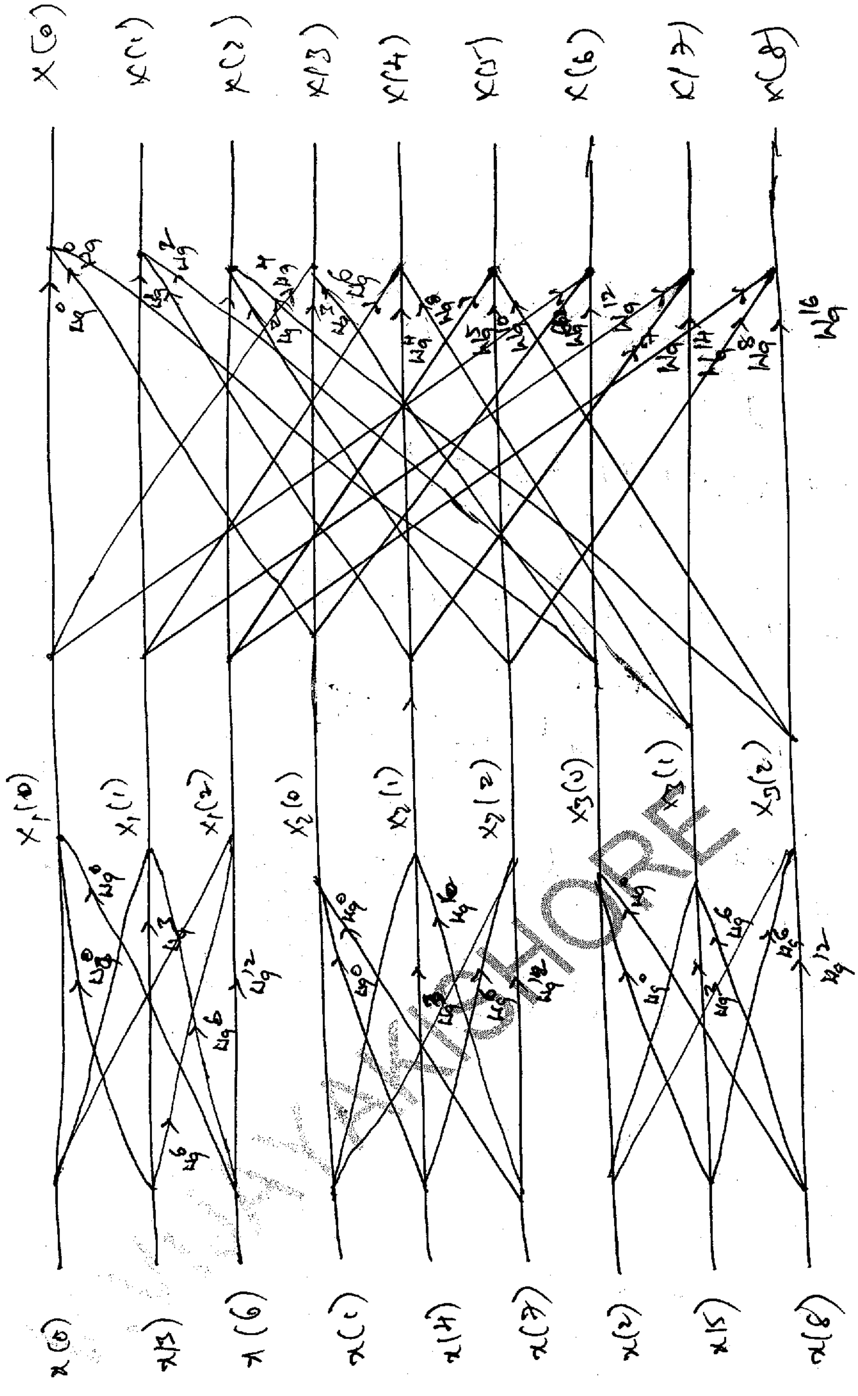
$$X(4) = X_1(1) + W_9^4 X_2(1) + W_9^8 X_3(1)$$

$$X(5) = X_1(2) + W_9^5 X_2(2) + W_9^{10} X_3(2)$$

$$X(6) = X_1(0) + W_9^6 X_2(0) + W_9^{12} X_3(0)$$

$$X(7) = X_1(1) + W_9^{7} X_2(1) + W_9^{14} X_3(1)$$

$$X(8) = X_1(2) + W_9^{8} X_2(2) + W_9^{16} X_3(2)$$



* Develop a Radix-3 DIF-FFT algorithm for evaluating the DFT for $N=9$.

Soln: $N = 9 = 3 \cdot 3$

$$X(k) = \sum_{n=0}^{8} x(n)\, W_9^{nk}$$

$$X(k) = \sum_{n=0}^{2} x(n)\, W_9^{nk} + \sum_{n=3}^{5} x(n)\, W_9^{nk} + \sum_{n=6}^{8} x(n)\, W_9^{nk}$$

$$= \sum_{n=0}^{2} x(n)\, W_9^{nk} + \sum_{n=0}^{2} x(n+3)\, W_9^{(n+3)k} + \sum_{n=0}^{2} x(n+6)\, W_9^{(n+6)k}$$

$$= \sum_{n=0}^{2} \left[x(n) + x(n+3)\, W_9^{3k} + x(n+6)\, W_9^{6k} \right] W_9^{nk}$$

$$X(3k) = \sum_{n=0}^{2} \left(x(n) + x(n+3)\, W_9^{9k} + x(n+6)\, W_9^{18k} \right) W_9^{nk}$$

$$= \sum_{n=0}^{2} \left(x(n) + x(n+3) + x(n+6) \right) W_9^{3nk} \qquad \left(\because W_N^{Nk} = 1 \right)$$

$$= \sum_{n=0}^{2} f(n)\, W_9^{3nk}$$

$$X(3k+1) = \sum_{n=0}^{2} \left(x(n) + x(n+3)\, W_9^{3} + x(n+6)\, W_9^{6} \right) W_9^{n} \cdot W_9^{3nk}$$

$$= \sum_{n=0}^{2} g(n)\, W_9^{n}\, W_9^{3nk}$$

$$X(3k+2) = \sum_{n=0}^{2} \left(x(n) + x(n+3)\, W_9^{6} + x(n+6)\, W_9^{3} \right) W_9^{2n} \cdot W_9^{3n}$$

$$= \sum_{n=0}^{2} h(n)\, W_9^{2n}\, W_9^{3nk}$$

Iiiwal

$f(0) = x(0) + x(3) + x(6)$

$f(1) = x(1) + x(4) + x(7)$

$f(2) = x(2) + x(5) + x(8)$

$g(0) = x(0) + x(3)\, W_9^{3} + x(6)\, W_9^{6}$

$g(1) = x(1) + x(4)\, W_9^{3} + x(7)\, W_9^{6}$

$g(2) = x(2) + x(5)\, W_9^{3} + x(8)\, W_9^{6}$

$h(0) = x(0) + x(3)\, W_9^6 + x(6)\, W_9^3$

$h(1) = x(1) + x(4)\, W_9^6 + x(7)\, W_9^3$

$h(2) = x(2) + x(5)\, W_9^6 + x(8)\, W_9^3$

III

$X(0) = f(0) + f(1) + f(2)$

$X(3) = f(0) + f(1)\, W_9^3 + f(2)\, W_9^6$

$X(6) = f(0) + f(1)\, W_9^6 + f(2)\, W_9^3$

$X(3k+1)$

$X(1) = g(0) + g(1)\, W_9^1 + g(2)\, W_9^2$

$X(4) = g(0) + g(1)\, W_9\, W_9^3 + g(2)\, W_9^2\, W_9^{\cancel{6}}$

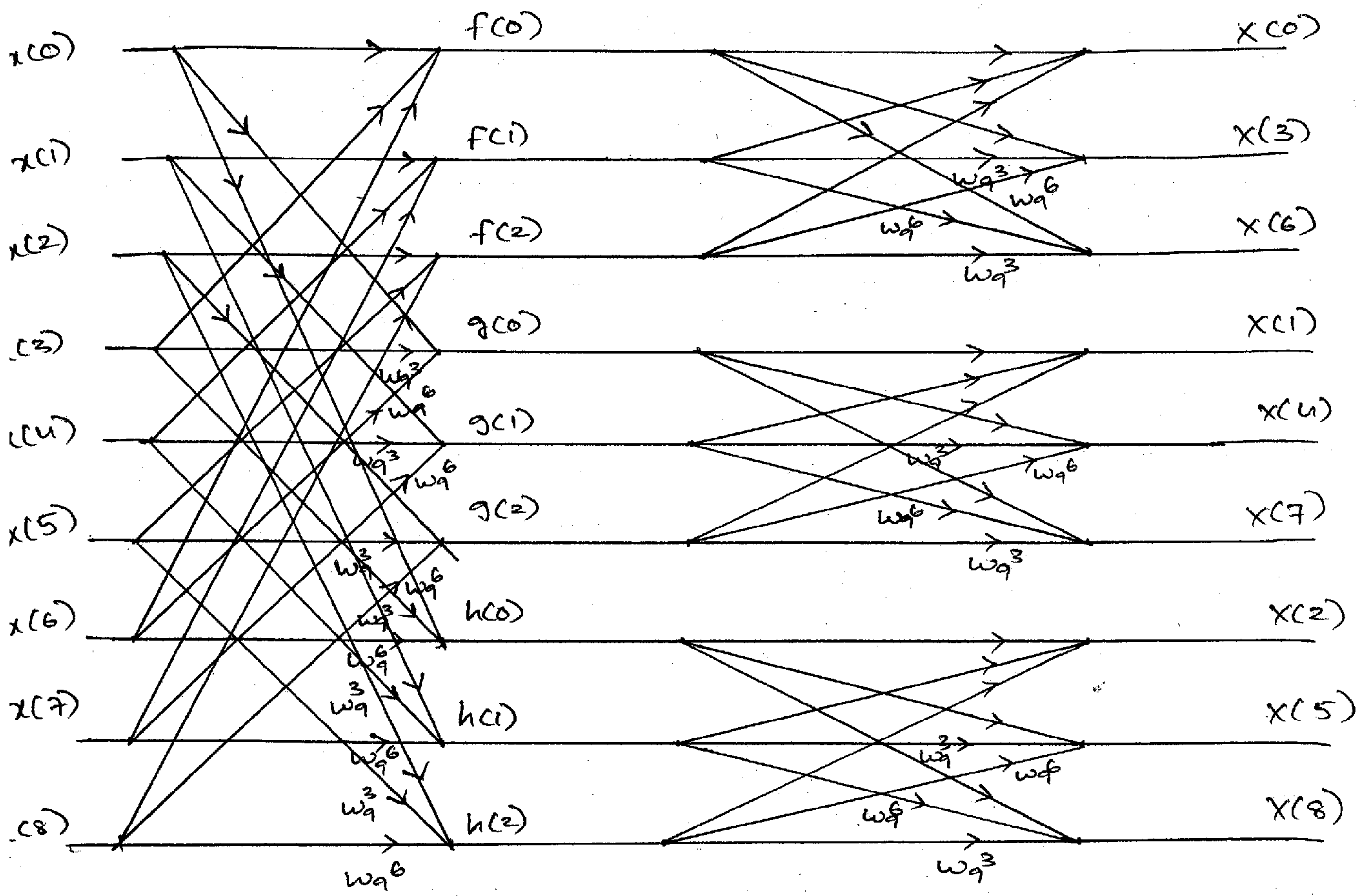
$X(7) = g(0) + g(1)\, W_9\, W_9^6 + g(2)\, W_9^2\, W_9^{32}$

$X(3k+2)$

$X(2) = h(0) + h(1)\, W_9^2 + h(2)\, W_9^4$

$X(5) = h(0) + h(1)\, W_9^2\, W_9^3 + h(2)\, W_9^4\, W_9^6$

$X(8) = h(0) + h(1)\, W_9^2\, W_9^6 + h(2)\, W_9^4\, W_9^3$



→ Develop DIT FFT algorithms for decomposing the DFT for N=6 and draw the flow diagrams for (a) $N = 2.3$ and (b) $N = 3.2$.

When $N = m_1 N_1$, the DFT can be written as

$$X(k) = \sum_{n=0}^{N_1-1} x(nm_1) W_N^{nm_1k} + \sum_{n=0}^{N_1-1} x(nm_1+1) W_N^{(nm_1+1)k} + \cdots$$

$$\cdots + \sum_{n=0}^{N_1-1} x(nm_1+m_1-1) W_N^{(nm_1+m_1-1)k} \longrightarrow (1)$$

(a) $N = 6 = 2.3$ where $m_1 = 2$ and $N_1 = 3$

then eq(1) becomes

$$X(k) = \sum_{n=0}^{2} x(2n) W_6^{2nk} + \sum_{n=0}^{2} x(2n+1) W_6^{(2n+1)k}$$

$$= \sum_{n=0}^{2} x(2n) W_6^{2nk} + W_6^{k} \sum_{n=0}^{2} x(2n+1) W_6^{2nk} \longrightarrow (2)$$

We know that $X_i(k+N_1) = X_i(k)$

Here $N_1 = 3$

$\Rightarrow X_i(k+3) = X_i(k) \longrightarrow (3)$

let $X_1(K) = \sum_{n=0}^{2} x(2n) W_6^{2nK}$

Now, $\quad = x(0) + x(2) W_6^{2K} + x(4) W_6^{4K}$

$X_1(0) = x(0) + x(2) + x(4)$

$X_1(1) = x(0) + x(2) W_6^{2} + x(4) W_6^{4}$

$X_1(2) = x(0) + x(2) W_6^{4} + x(4) W_6^{8}$ $\qquad W_6^8 = W_6^6 \times W_6^2 = W_6^2$

$\quad = x(0) + x(2) W_6^{4} + x(4) W_6^{2}$ $\qquad [W_6^6 = 1]$

let $X_2(K) = \sum_{n=0}^{2} x(2n+1) W_6^{2nK}$

$\quad = x(1) + x(3) W_6^{2K} + x(5) W_6^{4K}$

NOW, $X_2(0) = x(1) + x(3) + x(5)$

$X_2(1) = x(1) + x(3) W_6^{2} + x(5) W_6^{4}$

$X_2(2) = x(1) + x(3) W_6^{4} + x(5) W_6^{8}$

$\quad = x(1) + x(3) W_6^{4} + x(5) W_6^{2}$

eq② can be written as

$$X(K) = X_1(K) + W_6^K X_2(K)$$

$X(0) = X_1(0) + X_2(0)$

$X(1) = X_1(1) + W_6 X_2(1)$ $\qquad$ [∴ from ③]

$X(2) = X_1(2) + W_6^2 X_2(2)$

$X(3) = X_1(3) + W_6^3 X_2(3) = X_1(0) + W_6^3 X_2(0)$

$X(4) = X_1(4) + W_6^4 X_2(4) = X_1(1) + W_6^4 X_2(1)$

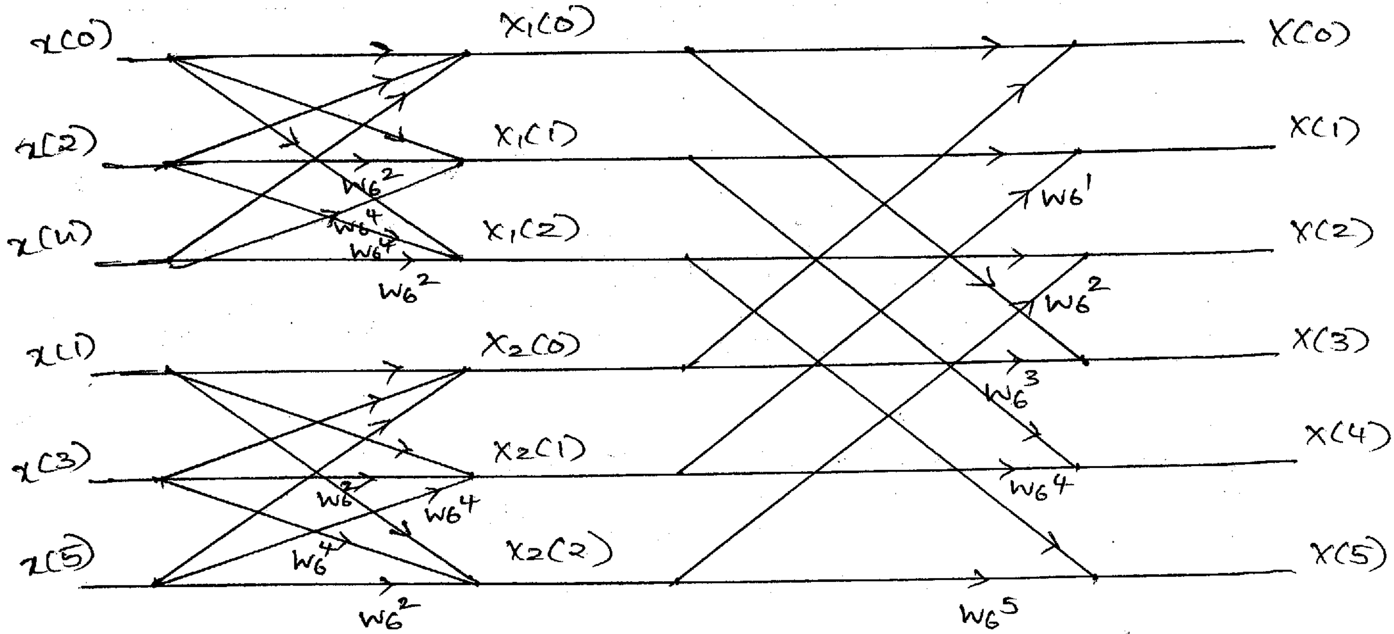
$X(5) = X_1(5) + W_6^5 X_2(5) = X_1(2) + W_6^5 X_2(2)$

$\cancel{X(6) = X_1(6) + W_6^6 X_2(6) = X_1(0) + W_6^3 X_2(0)}$

$\cancel{X(7) = X_1(7) + W_6^7 X_2(7) = X_1(1) + W_6^7}$

$\cancel{X(8) = X_1(8) + W_6^8 X_2(8)}$

flow diagram for $N=6=2\cdot3$



(b) $N=6=3\times2$ where $m_1=3$ and $N_1=2$

then eq① becomes

$$X(k)=\sum_{n=0}^{1}x(3n)W_6^{3nk}+\sum_{n=0}^{1}x(3n+1)W_6^{(3n+1)k}+\sum_{n=0}^{1}x(3n+2)W_6^{(3n+2)k}$$

$$=\sum_{n=0}^{1}x(3n)W_6^{3nk}+W_6^{k}\sum_{n=0}^{1}x(3n+1)W_6^{3nk}+W_6^{2k}\sum_{n=0}^{1}x(3n+2)W_6^{3k} \rightarrow ②$$

We know that $X_i(k+N_1)=X_i(k)$

Here $N_1=2$

$$X_i(k+2)=X_i(k) \longrightarrow ③$$

let $X_1(k)=\sum_{n=0}^{1}x(3n)W_6^{3nk}$

$$=x(0)+x(3)W_6^{3k}$$

Now, $X_1(0)=x(0)+x(3)$

$X_1(1)=x(0)+x(3)W_6^{3}$

let $X_2(k)=\sum_{n=0}^{1}x(3n+1)W_6^{3nk}=x(1)+x(4)W_6^{3k}$

$X_2(0)=x(1)+x(4)$

$X_2(1)=x(1)+x(4)W_6^{3}$

let $X_3(k) = \sum_{n=0}^{1} x(3n+2)\, W_6^{3nk} = x(2) + x(5)\, W_6^{3k}$

Now, $X_3(0) = x(2) + x(5)$

$X_3(1) = x(2) + x(5)\, W_6^{3}$

eq ② can be written as

$$X(k) = X_1(k) + W_6^{k} X_2(k) + W_6^{2k} X_3(k)$$

$X(0) = X_1(0) + X_2(0) + X_3(0)$

$X(1) = X_1(1) + W_6^{1} X_2(1) + W_6^{2} X_3(1)$

$X(2) = X_1(2) + W_6^{2} X_2(2) + W_6^{4} X_3(2)$

$= X_1(0) + W_6^{2} X_2(0) + W_6^{4} X_3(0)$

$X(3) = X_1(3) + W_6^{3} X_2(3) + W_6^{6} X_3(3)$

$= X_1(1) + W_6^{3} X_2(1) + W_6^{6} X_3(1)$

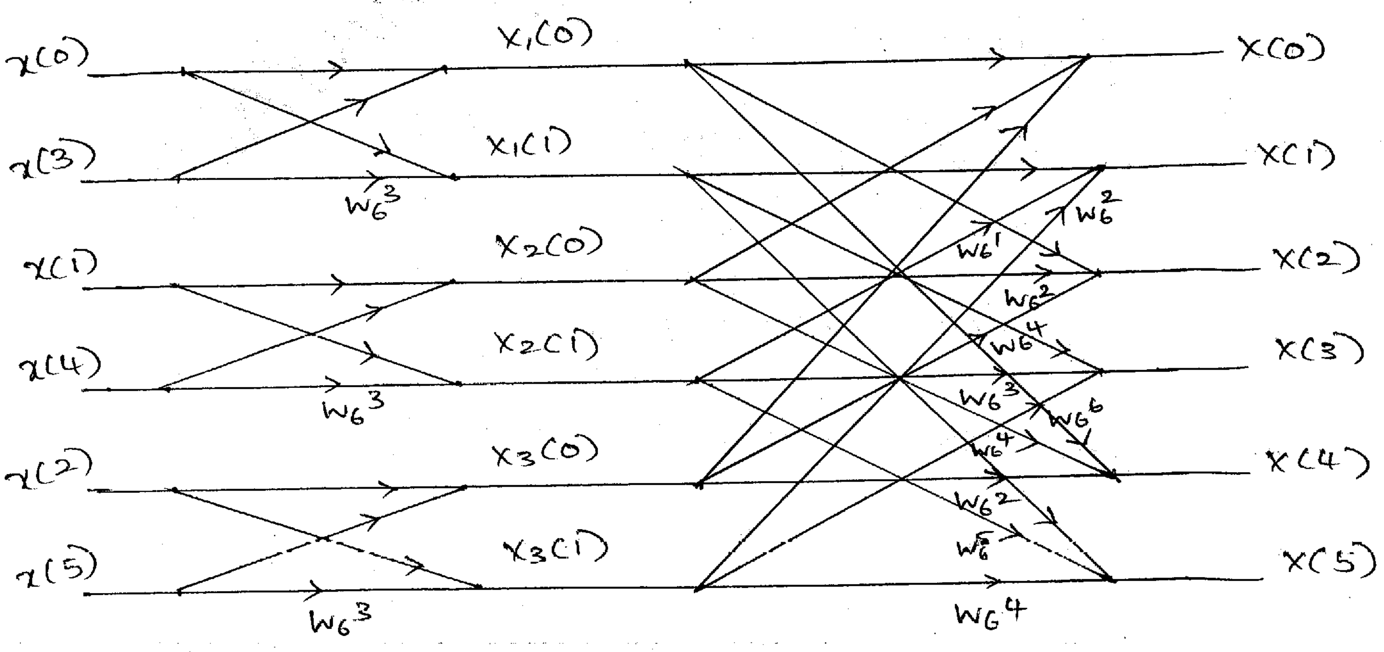
$X(4) = X_1(4) + W_6^{4} X_2(4) + W_6^{8} X_3(4)$

$= X_1(0) + W_6^{4} X_2(0) + W_6^{2} X_3(0)$

$X(5) = X_1(5) + W_6^{5} X_2(5) + W_6^{10} X_3(5)$

$= X_1(1) + W_6^{5} X_2(1) + W_6^{4} X_3(1)$

flow diagram for $N = 6 = 3 \cdot 2$



→ Develop a DIF FFT algorithm for decomposing the DFT for N=6 and draw the flow diagrams for a) N= 3.2 and b) N= 2.3.

(a) To develop DIF FFT algorithm for N= 3.2

N= 3.2 ⇒ 2 sequences of 3 elements each.

$$X(k) = \sum_{n=0}^{5} x(n) W_6^{nk} \qquad \left[X(k) = \sum_{n=0}^{N-1} x(n) W_N^{nk} \right]$$

$$= \sum_{n=0}^{2} x(n) W_6^{nk} + \sum_{n=3}^{5} x(n) W_6^{nk}$$

$$= \sum_{n=0}^{2} x(n) W_6^{nk} + \sum_{n=0}^{2} x(n+3) W_6^{(n+3)k}$$

$$= \sum_{n=0}^{2} x(n) W_6^{nk} + W_6^{3k} \sum_{n=0}^{2} x(n+3) W_6^{nk}$$

$$= \sum_{n=0}^{2} \left[x(n) + x(n+3) W_6^{3k} \right] W_6^{nk}$$

$$X(2k) = \sum_{n=0}^{2} \left[x(n) + x(n+3) W_6^{6k} \right] W_6^{2nk}$$

$$= \sum_{n=0}^{2} \left[x(n) + x(n+3) \right] W_6^{2nk} \qquad \left[W_6^{6k} = (W_6^{6})^{k} = 1^{k} = 1 \right]$$

$$X(2k+1) = \sum_{n=0}^{2} \left[x(n) + x(n+3) W_6^{3(2k+1)} \right] W_6^{(2k+1)n}$$

$$= \sum_{n=0}^{2} \left[x(n) + x(n+3) W_6^{6k} W_6^{3} \right] W_6^{(2k+1)n}$$

$$\boxed{W_N = e^{-j2\pi/N}}$$

$$W_6^{3} = \left[e^{-j2\pi/6} \right]^{3} = e^{-j2\pi/2} = \cancel{\cos\frac{\pi}{2}}$$

$$= e^{-j\pi} = \cos\pi - j\sin\pi = -1$$

$$\therefore X(2k+1) = \sum_{n=0}^{2} \left[x(n) - x(n+3) \right] W_6^{n} W_6^{2kn}$$

let $g(n) = x(n) + x(n+3)$, $h(n) = x(n) - x(n+3)$

Now,
$$\left.\begin{aligned} X(2k) &= \sum_{n=0}^{2} g(n) W_6^{2nk} \\ X(2k+1) &= \sum_{n=0}^{2} h(n) W_6^{n} \cdot W_6^{2nk} \end{aligned}\right\} \longrightarrow (1)$$

$g(0) = x(0) + x(3)$ $\qquad$ $h(0) = x(0) - x(3)$

$g(1) = x(1) + x(4)$ $\qquad$ $h(1) = x(1) - x(4)$

$g(2) = x(2) + x(5)$ $\qquad$ $h(2) = x(2) - x(5)$

Now,

$$X(0) = \sum_{n=0}^{2} g(n) W_6^0 = g(0) + g(1) + g(2)$$

$$X(2) = \sum_{n=0}^{2} g(n) W_6^{2n} = g(0) + g(1) W_6^2 + g(2) W_6^4$$

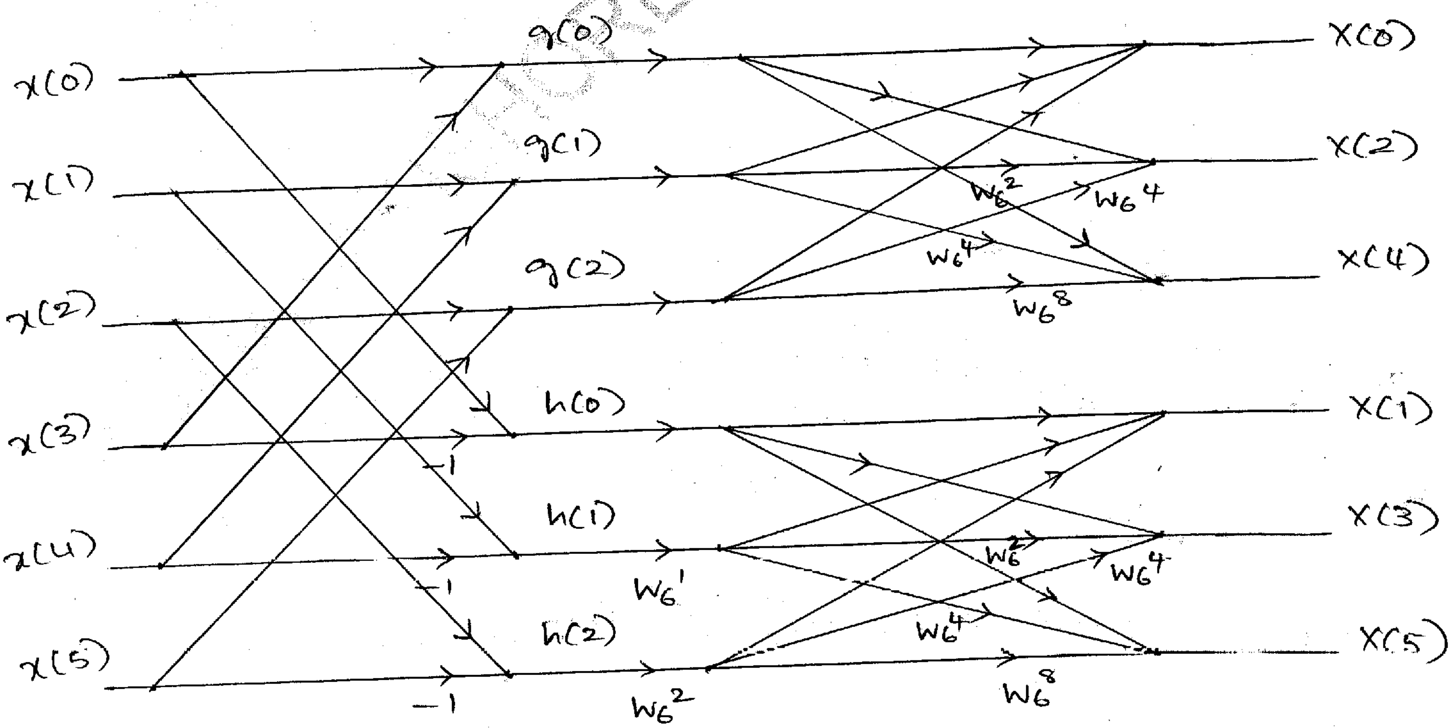
$$X(4) = \sum_{n=0}^{2} g(n) W_6^{4n} = g(0) + g(1) W_6^4 + g(2) W_6^8$$

$$X(1) = \sum_{n=0}^{2} h(n) W_6^{n} = h(0) + h(1) W_6^1 + h(2) W_6^2$$

$$X(3) = \sum_{n=0}^{2} h(n) W_6^{n} W_6^{2n} = h(0) + h(1) W_6^1 W_6^2 + h(2) W_6^2 W_6^4$$

$$X(5) = \sum_{n=0}^{2} h(n) W_6^{n} W_6^{4n} = h(0) + h(1) W_6^1 W_6^4 + h(2) W_6^2 W_6^8$$

Flow diagram :-



b) To develop DIF FFT algorithm for $N = 2.3$

$N = 2.3 \Rightarrow$ 3 sequences of 2 elements each.

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

$$\Rightarrow X(k) = \sum_{n=0}^{5} x(n) W_6^{nk} \quad [\because N=6]$$

$$= \sum_{n=0}^{1} x(n) W_6^{nk} + \sum_{n=2}^{3} x(n) W_6^{nk} + \sum_{n=4}^{5} x(n) W_6^{nk}$$

$$= \sum_{n=0}^{1} x(n) W_6^{nk} + \sum_{n=0}^{1} x(n+2) W_6^{(n+2)k} + \sum_{n=0}^{1} x(n+4) W_6^{(n+4)k}$$

$$= \sum_{n=0}^{1} \left[x(n) + x(n+2) W_6^{2k} + x(n+4) W_6^{4k}\right] W_6^{nk}$$

$$X(3k) = \sum_{n=0}^{1} \left[x(n) + x(n+2) W_6^{6k} + x(n+4) W_6^{12k}\right] W_6^{nk}$$

$$= \sum_{n=0}^{1} \left[x(n) + x(n+2) + x(n+4)\right] W_6^{nk} \quad \left[W_6^{6k} = 1;\ W_6^{12k} = 1\right]$$

$$X(3k+1) = \sum_{n=0}^{1} \left[x(n) + x(n+2) W_6^{2(3k+1)} + x(n+4) W_6^{4(3k+1)}\right] W_6^{n(3k+1)}$$

$$= \sum_{n=0}^{1} \left[x(n) + x(n+2) W_6^{6k} W_6^{2} + x(n+4) W_6^{12k} W_6^{4}\right] W_6^{n(3k+1)}$$

$$= \sum_{n=0}^{1} \left[x(n) + x(n+2) W_6^{2} + x(n+4) W_6^{4}\right] W_6^{n} W_6^{3nk}$$

$$X(3k+2) = \sum_{n=0}^{1} \left[x(n) + x(n+2) W_6^{2(3k+2)} + x(n+4) W_6^{4(3k+2)}\right] W_6^{n(3k+2)}$$

$$= \sum_{n=0}^{1} \left[x(n) + x(n+2) W_6^{6k} W_6^{4} + x(n+4) W_6^{12k} \cdot W_6^{8}\right] W_6^{n(3k+2)}$$

$$= \sum_{n=0}^{1} \left[x(n) + x(n+2) W_6^{4} + x(n+4) W_6^{8}\right] W_6^{2n} W_6^{3nk}$$

let $f(n) = x(n) + x(n+2) + x(n+4)$

$g(n) = x(n) + x(n+2) W_6^{2} + x(n+4) W_6^{4}$

$h(n) = x(n) + x(n+2) W_6^{4} + x(n+4) W_6^{8}$

$$h(n) = x(n) + x(n+2)W_6^4 + x(n+4)W_6^2 \quad [\because W_6^8 = W_6^2]$$

Now,

$$f(0) = x(0) + x(2) + x(4)$$

$$f(1) = x(1) + x(3) + x(5)$$

$$g(0) = x(0) + x(2)W_6^2 + x(4)W_6^4$$

$$g(1) = x(1) + x(3)W_6^2 + x(5)W_6^4$$

$$h(0) = x(0) + x(2)W_6^4 + x(4)W_6^2$$

$$h(1) = x(1) + x(3)W_6^4 + x(5)W_6^2$$

$X(3k)$, $X(3k+1)$ & $X(3k+2)$ can be written as

$$X(3k) = \sum_{n=0}^{1} f(n) W_6^{3nk}$$

$$X(3k+1) = \sum_{n=0}^{1} g(n) W_6^{n} W_6^{3nk}$$

$$X(3k+2) = \sum_{n=0}^{1} h(n) W_6^{2n} W_6^{3nk}$$

Now,

$$X(0) = \sum_{n=0}^{1} f(n) W_6^0 = f(0) + f(1)$$

$$X(1) = \sum_{n=0}^{1} g(n) W_6^{n} W_6^0 = g(0) + g(1)W_6^1$$

$$X(2) = \sum_{n=0}^{1} h(n) W_6^{2n} W_6^0 = h(0) + h(1)W_6^2$$

$$X(3) = \sum_{n=0}^{1} f(n) W_6^{3n} = f(0) + f(1)W_6^3$$

$$X(4) = \sum_{n=0}^{1} g(n) W_6^{n} W_6^{3n} = g(0) + g(1)W_6^1 W_6^3$$

$$X(5) = \sum_{n=0}^{1} h(n) W_6^{2n} W_6^{3n} = h(0) + h(1)W_6^2 W_6^3$$

Flow diagram