Frequency analysis of signal using DFT, FFT algorithm, Application of FFT algorithm.

2020 7a) Define DFT and IDFT equation. (2)

DFT

The discrete Fourier transform (DFT) is the primary transform used for numerical computation in digital signal processing. It is very widely used for spectrum analysis, fast convolution, and many other applications. The DFT transforms N discrete-time samples to the same number of discrete frequency samples, and is defined as

$$X(k) = \sum_{n=0}^{N-1} x(n) e^{-\left(i \frac{2\pi nk}{N}\right)}$$

The DFT is widely used in part because it can be computed very efficiently using fast Fourier transform (FFT) algorithms.

IDFT

The inverse DFT (IDFT) transforms N discrete-frequency samples to the same number of discrete-time samples. The IDFT has a form very similar to the DFT,

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

and can thus also be computed efficiently using FFTs.

7(b) Define symmetry property of DFT equation

Ans: Symmetry property of real value of x(n)

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

- Let $k = N - k$:

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

- But $e^{-j2\pi n} = 1$ *for* $n = 0,\ 1,\ 2,\ \ldots$
- Therefore

$$X(N-k) = \sum_{n=0}^{N-1} x(n)e^{j2\pi kn/N} = X^*(k) \quad \textit{complex conjugate of } X(k)$$

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

- When N is even, $|Xk|$ is symmetric about N/2.
- The phase, Xk, has odd symmetry about N/2.

7 c) Define DFT leakage. (3)

DFT leakage happens due to inappropriate sampling of the signals. The DFT is vulnerable to spectral leakage. Spectral leakage occurs when a non-integer number of periods of a signal is sent to the DFT. Spectral leakage lets a single-tone signal be spread among several frequencies after the DFT operation. This makes it hard to find the actual frequency of the signal.

Frequency Bins: An N point sequence will produce an N point frequency response, each of these points is called a frequency bin. Exactly where these frequency bins lie in terms of actual values (e.g. 4356 Hz) depends on the number of points you have in your sequence, as well as the sampling frequency. The more points, the closer the bins lie together, the higher the sampling frequency the further apart the bins are.

Leakage between Frequency Bins

If we take the sine wave example, you would expect to see just one sample at the frequency of the signal. However, if we get this discontinuity then the frequency of the signal won't be lying on any bin, it falls between the two bins closest to it.

DFTs are constrained to operate on a finite set of N input values sampled at a sample rate of f5, to produce an N-point transform whose discrete outputs are associated with the individual analytical frequencies $f_{analysis}(m)$, with

$$f_{analysis}(m) = \frac{m\,fs}{N} \text{ where } m = 0, 1, \ldots, N-1. \text{ ------ (i)}$$

The DFT produces correct results only when the input data sequence contains energy precisely at the analysis frequencies given in (i). at integral multiples of our fundamental frequency f_s/N. If the input has a signal component at some intermediate frequency between our analytical frequencies of mf_s/N, say 1.5fs /N, this input signal will show up to some degree in all of the N Output analysis frequencies of our DFT! Assume we're taking a 64-point D.FT of the sequence indicated by the dots in Figure 1(a). The sequence is a sine wave with exactly three cycles contained in our N = 64 samples.

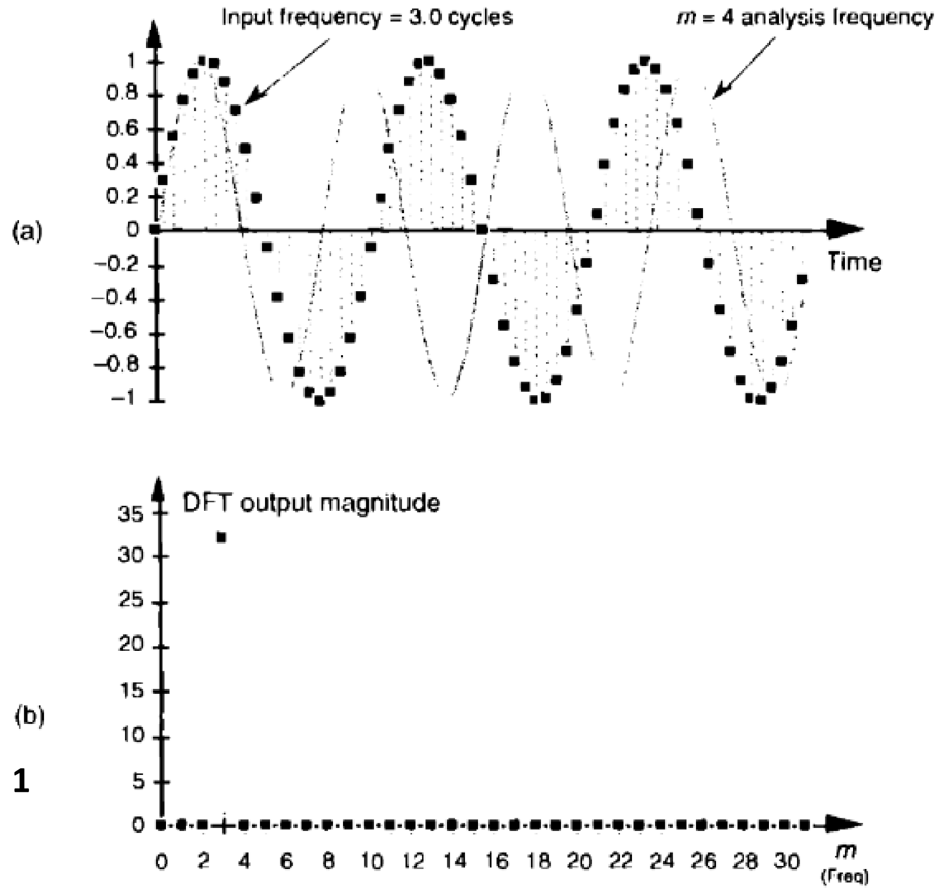


Figure 3-7 64-point DFT: (a) input sequence of three cycles and the *m* = 4 analysis frequency sinusoid; (b) DFT output magnitude.

Figure 1 (b) shows the first half of the DFT of the input sequence and indicates that the sequence has an average value of zero (X(0) = 0) and no signal components at any frequency other than the m = 3

Frequency. The dots in Figure 2(a) show an input sequence having 3.4 cycles over our N = 64 samples. Because the input sequence does not have an integral number of cycles over our 64-sample interval, input energy has leaked into all the other DFT output bins as shown in Figure 2(b).

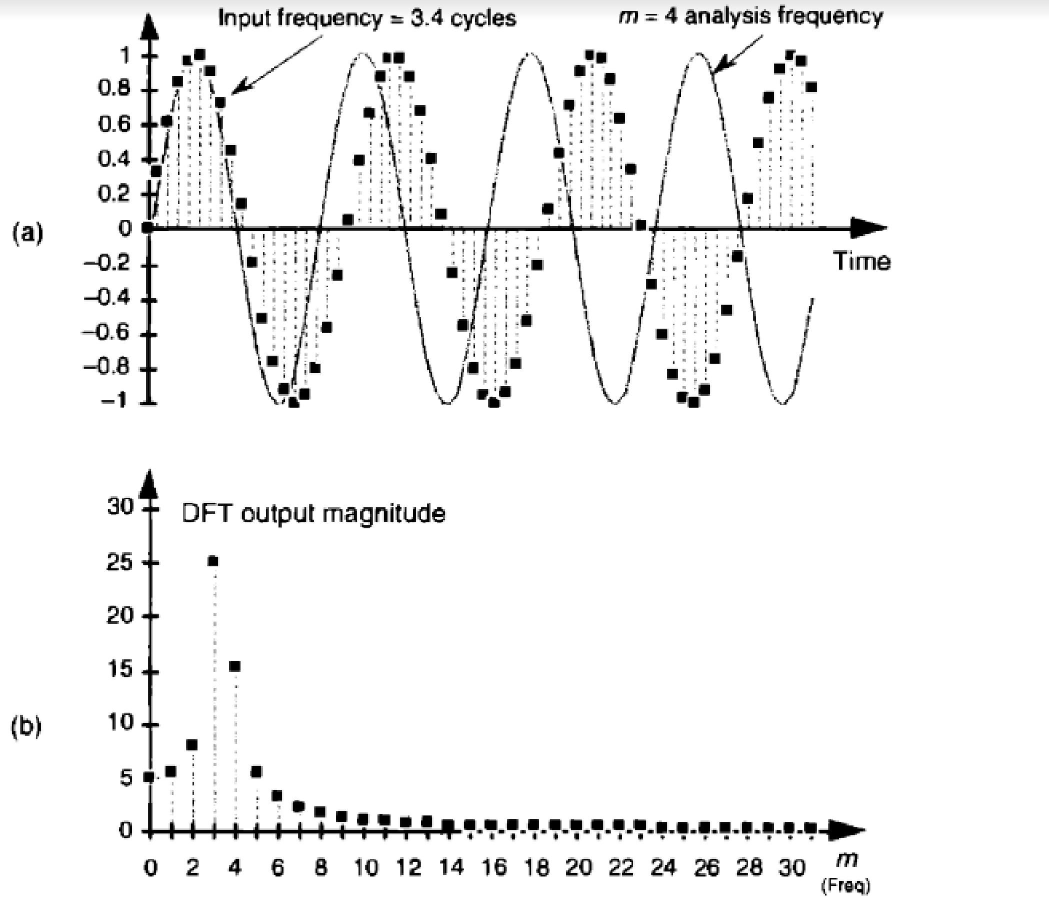


Figure 3-8 64-point DFT: (a) 3.4 cycles input sequence and the $m = 4$ analysis frequency sinusoid; (b) DFT output magnitude.

The m= 4 bin, for example, is not zero because the sum of the products of the input sequence and the m = 4 analysis frequency is no longer zero. This is leakage. it causes any input signal whose frequency is not exactly at a DFT bin center to leak into all of the other DFT output bins. Moreover, leakage is an unavoidable fact of life when we perform the DFT on real-world finite-length time sequences.

2020

8 a) Derive 4 point FFT equation from DFT equation upto twiddle factor formation. (3)

Answer: We know,

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

FFT requires use of W notation

$$F_r = \sum_{k=0}^{N_0-1} f_k\, W_{N_0}^{kr}$$

where $W_{N_0} = e^{-j\frac{2\pi}{N_0}} = e^{-j\Omega_0}$

Consider the N_0 point Data Sequence f_k (and assume N_0 is even).

Divide Into two $\frac{N_0}{2}$ Sequence.

$$\underbrace{f_0, f_2, f_4, \ldots, f_{N_0-2}}_{g_k},\ \underbrace{f_1, f_3, f_5, \ldots f_{N_0-1}}_{h_k}$$

We can rewrite In FFT expression as,

$$F_r = \sum_{k=0}^{N_0/2-1} f_{2k}\, W_{N_0}^{2kr} + \sum_{k=0}^{\frac{N_0}{2}-1} f_{2k+1}\, W_{N_0}^{(2k+1)r}$$

$$= \sum_{k=0}^{\frac{N_0}{2}-1} f_{2k} \left(W_{N_0}^{2}\right)^{kr} + W_{N_0}^{r} \sum_{k=0}^{\frac{N_0}{2}-1} f_{2k+1} \left(W_{N_0}^{2}\right)^{kr}$$

$$= \sum_{k=0}^{\frac{N_0}{2}-1} f_{2k}\, W_{\frac{N_0}{2}}^{kr} + W_{N_0}^{r} \cdot \sum_{k=0}^{\frac{N_0}{2}-1} f_{2k+1}\, W_{\frac{N_0}{2}}^{kr}$$

$$= \sum_{k=0}^{\frac{N_0}{2}-1} g_k\, W_{\frac{N_0}{2}}^{kr} + W_{N_0}^{r} \sum_{k=0}^{\frac{N_0}{2}-1} h_k\, W_{\frac{N_0}{2}}^{kr}$$

$$= G_r + W_{N_0}^{r} H_r \qquad (1)$$

Here, $\frac{N_0}{2}$ sequeceses g_k and h_k

G_r and H_r are their $\frac{N_0}{2}$-point DFTs

An N_0-point DFT is always N_0 periodic

$\Rightarrow$ G_r and H_r Are $\frac{N_0}{2}$ periodic

Since $N_0/2$ periodic

$$G_{r+\frac{N_0}{2}} = G_r \qquad (2)$$

$$H_{r+\frac{N_0}{2}} = H_r \qquad (3)$$

Also, we compute

$$W_{N_0}^{r+\frac{N_0}{2}} = W_{N_0}^{\frac{N_0}{2}} W_{N_0}^{r} = e^{-j\pi} W_{N_0}^{r} = -W_{N_0}^{r} \quad (4)$$

Putting (2), (3) & (4) in (1) we get,

$$F_{r+\frac{N_0}{2}} = G_r - W_{N_0}^{r} H_r$$

The first $\frac{N_0}{2}$ points can be computed with

$$F_r = G_r + W_{N_0}^{r} H_r \quad 0 \leq r \leq \frac{N_0}{2} - 1$$

The Last $\frac{N_0}{2}$ points are computed with

$$F_{r+\frac{N_0}{2}} = G_r - W_{N_0}^{r} H_r ; \; 0 \leq r \leq \frac{N_0}{2} - 1$$

The product $W_{N_0}^{r} H_r$ only has to be computed Once. Here we can get two points in the DFT using just one multiplication.

Given G_r and H_r, to Compute F_r requires

$N_0/2$ Complex Multiplication

N_0 Complex Addition

$$F_r = G_r + W_{N_0}^{r} H_r \quad 0 \leq r \leq \frac{N_0}{2} - 1$$

$$F_{r+\frac{N_0}{2}} = G_r - W_{N_0}^{r} H_r \quad 0 \leq r \leq \frac{N_0}{2} - 1$$

8(c)Determine 4 point DFT x(n)={1,0,1,1}

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

A.F.F: $X(k) = \sum_{0}^{N-1} x(n) \cdot e^{-j \frac{2\pi}{N} k(n)}$

$X(k) = \sum_{0}^{3} x(n)\, e^{-j \frac{2\pi}{4} k(n)}$ —— 1

$= \sum_{0}^{3} x(n)\, e^{-j \frac{\pi}{2} k(n)}$

$K = \{0, 1, 2, 3\}$

If $K=0$

$= \sum_{0}^{3} x(n) e^{-0} = \sum_{0}^{3} x(n)$

$= x(0) + x(1) + x(2) + x(3)$

$= 1 + 0 + 1 + 1$

$= 3$

If $K=1$ $\sum_{0}^{3} x(n) \cdot e^{-j \frac{\pi}{2} 1 n} = \sum_{0}^{3} x(n)\, e^{-j \frac{\pi}{2} n}$

$= x(0) \cdot e^{-0} + x(1) \cdot e^{-j \frac{\pi}{2} 1} + x(2) \cdot e^{-j\pi} + x(3) \cdot e^{-j \frac{\pi}{2} \cdot 3}$

$= 1.1 + 0 + 1 e^{-j\pi} + 1 . e^{-j \frac{\pi}{2} \cdot 3}$

$= 1 + (-1) + j$

$= j$

If $k=2 = \sum_{0}^{3} x(n).e^{-J\frac{\pi}{2}.2n}$

$= \sum_{0}^{3} x(n) e^{-J\pi n}$

$= x(0).e^{-0} + 0 + x(2) e^{-J2\pi} + x(3).e^{-J3\pi}$

$= 1.1 + 0 + e^{-J2\pi} + e^{-J3\pi}$

$= 1 + 0 + J - 1$

$= J$

If $k=3 = \sum_{0}^{3} x(n)e^{-J\frac{\pi}{2}.3n} = \sum_{0}^{3} x(n)e^{-J\frac{3\pi}{2}n}$

$= x(0).e^{-0} + 0 + x(2).e^{-J\frac{3\pi}{2}.2} + x(3).e^{-J\frac{3\pi}{2}.3}$

$= 1.1 + 0 + e^{-J3\pi} + e^{-J\frac{9\pi}{2}}$

$= 1 + 0 - 1 - J$

$= -J$

~~$x(K) = \{J, J, -J\}$~~

$x(K) = \{3, J, J, -J\}$

8(c) What is twiddle factor?Describe its importance in FFT equation

Ans: A twiddle factor, in fast Fourier transform (FFT) algorithms, is any of the trigonometric constant coefficients that are multiplied by the data in the course of the algorithm.

Importance of FFT equation: FFT is so important because

The FFT is a very efficient algorithm technique based on the DFT but with fewer computations required. The FFT is one of the most commonly used operations in digital signal processing to provide a frequency spectrum analysis [1–6].It is an important measurement method in the science of audio and acoustics measurement. It converts a signal into individual spectral components and thereby provides frequency information about the signal.

2019

5. (b) Explain DFT leakage with example. (3) --------------------same as 2020 7(c)

5(c) Find 4 point DFT of following sequence x(n)={1,0,-1,0}

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

A.T.F :

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

$$= \sum_{0}^{3} x(n)e^{-j\frac{2\pi}{4}kn} \quad \text{——①} \quad 0 \le k \le N-1$$

$k = 0, 1, 2, 3$

If $k=0$

$$x(0) = \sum_{0}^{3} x(n) e^{-0} \Rightarrow x(0) = \sum_{0}^{3} x(n).1 \Rightarrow x(0) = \sum_{0}^{3} x(n)$$

$= x(0) + x(1) + x(2) + x(3)$

$= 1 + 0 + (-1) + 0$

$= 1 - 1$

$= 0$

If $k = 1$.

$$x(1) = \sum_{0}^{3} x(n)e^{-j\frac{2\pi}{4}1.n} \Rightarrow x(1) = \sum_{0}^{3} x(n)e^{-j\frac{2\pi}{4}n}$$

$$= x(0).e^{-0} + x(1).e^{-j\frac{2\pi}{4}.1} + x(2).e^{-j\frac{2\pi}{4}.2} + x(3).e^{-j\frac{2\pi}{4}.3}$$

$= 1.1 + 0 + (-1).(-1) + 0$

$= 1 + 1$

$= 2$

If $k=2$

$$x(2) = \sum_{0}^{3} x(n) e^{-j\frac{2\pi}{4}2n}$$

$$= x(0).e^{-0} + x(1).e^{-j\frac{2\pi}{4}.2.1} + x(2).e^{-j\frac{2\pi}{4}.2.2} + x(3)e^{-j\frac{2\pi}{4}.2.3}$$

$$= 1.1 + 0 + (-1).e^{-j2\pi} + 0$$

$$= 1 - 1$$

$$= 0$$

If $k=3$

$$x(3) = \sum_{0}^{3} x(n) e^{-j\frac{2\pi}{4}3n}$$

$$= x(0).e^{-0} + 0 + x(2)e^{-j\frac{2\pi}{4}3.3} + 0$$

$$= 1.1 + 0 + (-1)e^{-j\frac{18\pi}{4}} + 0$$

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

$$= 1 + j$$

$$x(k) = \{0, 2, 0, 1+j\}$$

7(a) Why FFT is needed?

Ans: FFT helps in converting the time domain in frequency domain which makes the calculations easier as we always deal with various frequency bands in communication system another very big advantage is that it can convert the discrete data into a contionousdata type available at various frequencies. The Fast Fourier Transform (FFT) is commonly used to transform an image between the spatial and frequency domain . Unlike other domains such as Hough and Radon, the FFT method preserves all original data. Plus, FFT fully transforms images into the frequency domain, unlike time-frequency or wavelet transforms.

7.(b) Describe the algorithm of decimation in time (DIT) of FFT. (3)

Answer:

Decimation in time

The radix-2 decimation-in-time algorithm rearranges the discrete Fourier transform (DFT) equation into two parts: a sum over the even-numbered discrete-time indices $n = [0, 2, 4, \ldots, N-2]$ and a sum over the odd-numbered indices $n = [1, 3, 5, \ldots, N-1]$ as in Equation:

$$\begin{aligned}
X(k) &= \sum_{n=0}^{N-1} x(n) e^{-\left(i\frac{2\pi nk}{N}\right)} \\
&= \sum_{n=0}^{\frac{N}{2}-1} x(2n) e^{-\left(i\frac{2\pi\times(2n)k}{N}\right)} + \sum_{n=0}^{\frac{N}{2}-1} x(2n+1) e^{-\left(i\frac{2\pi(2n+1)k}{N}\right)} \\
&= \sum_{n=0}^{\frac{N}{2}-1} x(2n) e^{-\left(i\frac{2\pi nk}{\frac{N}{2}}\right)} + e^{-\left(i\frac{2\pi k}{N}\right)} \sum_{n=0}^{\frac{N}{2}-1} x(2n+1) e^{-\left(i\frac{2\pi nk}{\frac{N}{2}}\right)} \\
&= \mathrm{DFT}_{\frac{N}{2}}\left[\left[x(0), x(2), \ldots, x(N-2)\right]\right] + W_N^k\, \mathrm{DFT}_{\frac{N}{2}}\left[\left[x(1), x(3), \ldots, x(N-1)\right]\right]
\end{aligned}$$

The mathematical simplifications in Equation reveal that all DFT frequency outputs $X(k)$ can be computed as the sum of the outputs of two length-$\frac{N}{2}$ DFTs, of the even-indexed and odd-indexed discrete-time samples, respectively, where the odd-indexed short DFT is multiplied by a so-called **twiddle factor** term $W_N^k = e^{-\left(i\frac{2\pi k}{N}\right)}$. This is called a **decimation in time** because the time samples are rearranged in alternating groups, and a **radix-2** algorithm because there are two groups. Figure graphically illustrates this form of the DFT computation, where for convenience the frequency outputs of the length-$\frac{N}{2}$ DFT of the even-indexed time samples are denoted $G(k)$ and those of the odd-indexed samples as $H(k)$. Because of the periodicity with $\frac{N}{2}$ frequency samples of these length-$\frac{N}{2}$ DFTs, $G(k)$ and $H(k)$ can be used to compute **two** of the length-N DFT frequencies, namely $X(k)$ and $X\left(k+\frac{N}{2}\right)$, but with a different twiddle factor. This reuse of these short-length DFT outputs gives the FFT its computational savings.

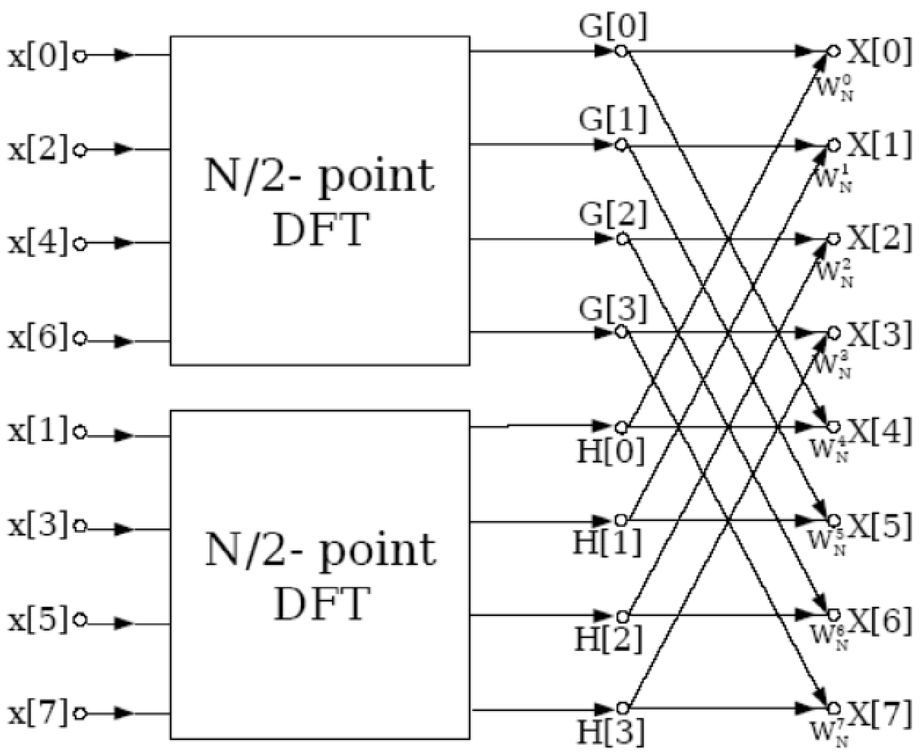
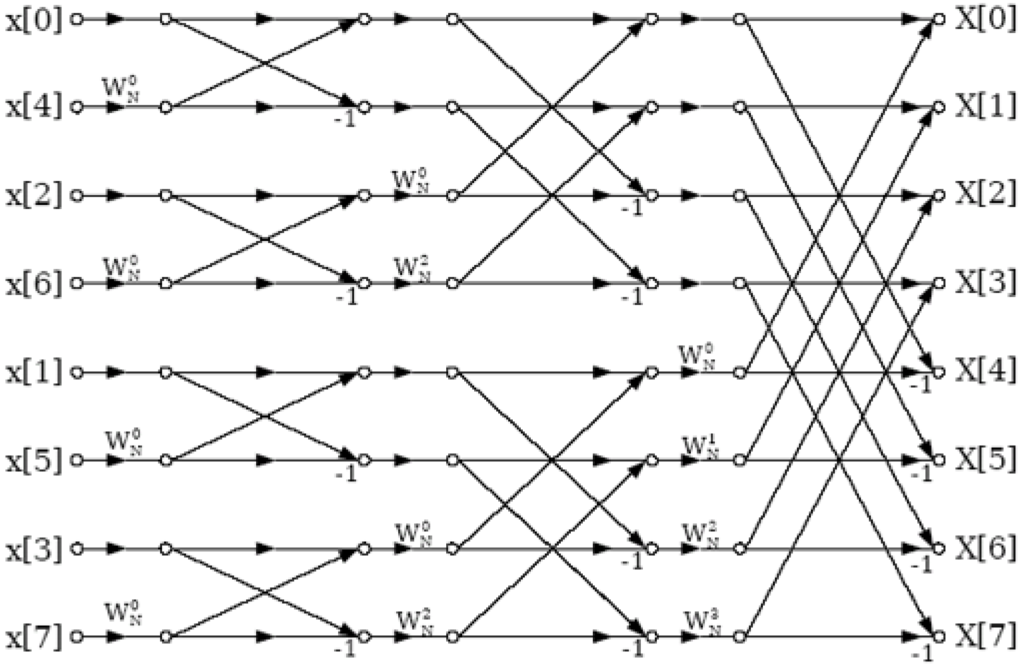


Figure 1. Decimation in time of a length-N DFT into two length-$\frac{N}{2}$ DFTs followed by a combining stage.



The full radix-2 decimation-in-time decomposition illustrated in Figure using the simplified butterflies involves $M = \log_2 N$ stages, each with $\frac{N}{2}$ butterflies per stage. Each butterfly requires 1 complex multiply and 2 adds per butterfly. The total cost of the algorithm is thus

Computational cost of radix-2 DIT FFT

- $\frac{N}{2}\log_2 N$ complex multiplies
- $N\log_2$ N complex adds

This is a remarkable savings over direct computation of the DFT. For example, a length-1024 DFT would require 1048576 complex multiplications and 1047552 complex additions with direct computation, but only 5120 complex multiplications and 10240 complex additions using the radix-2 FFT, a savings by a factor of 100 or more. The relative savings increase with longer FFT lengths, and are less for shorter lengths.

Modest additional reductions in computation can be achieved by noting that certain twiddle factors, namely Using special butterflies for W_N^0, $W_N^{\frac{N}{2}}$, $W_N^{\frac{N}{4}}$, $W_N^{\frac{N}{8}}$, $W_N^{\frac{3N}{8}}$, require no multiplications, or fewer real multiplies than other ones. By implementing special butterflies for these twiddle factors as discussed in FFT algorithm and programming tricks,

7

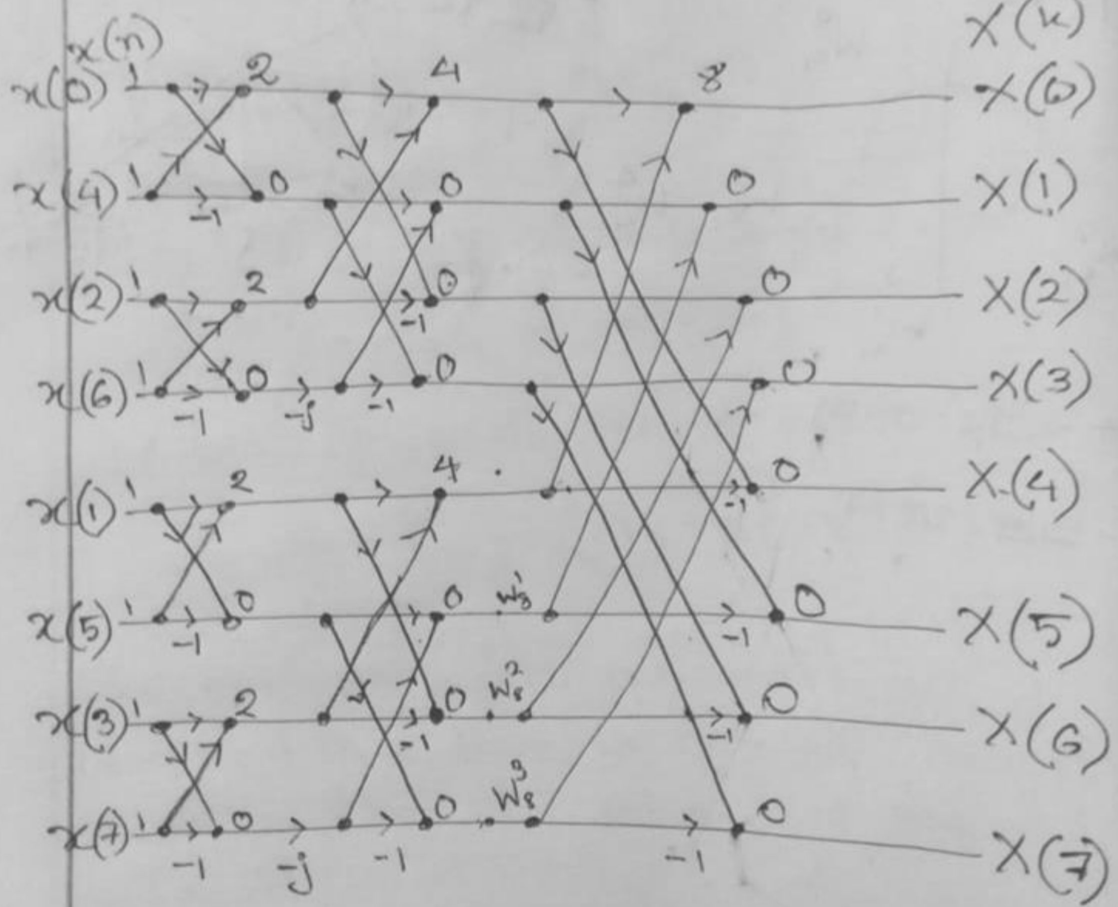
c) Compute the 8-point DFT of the sequence $x(n) = \begin{cases} 1 & 0 \le n \le 7 \\ 0 & otherwise \end{cases}$ 2.75

By using DIT (Decimation-In-Time), DIF (Decimation-In-Frequency) algorithm.

7(e) 8-point DFT of $x(n) = \begin{cases} 1 & 0 \le n \le 7 \\ 0 & \text{otherwise} \end{cases}$

Using DIT algorithms-

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

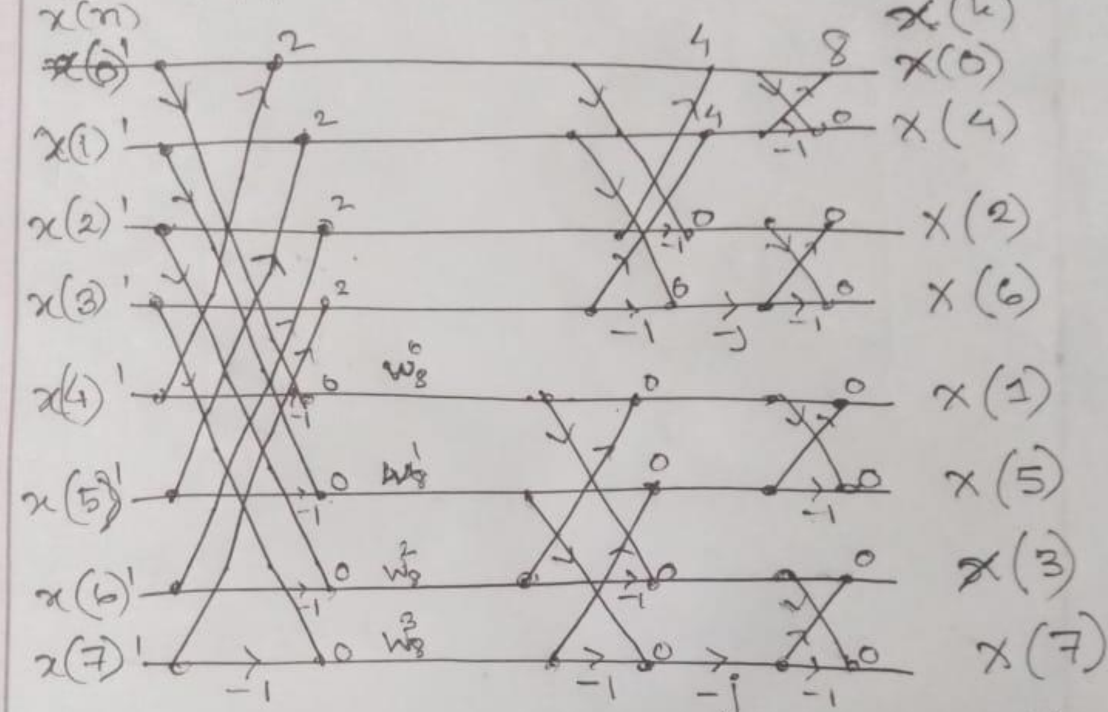


$\therefore X(k) = \text{DFT}\{x(n)\}$

$= \{8, 0, 0, 0, 0, 0, 0, 0\}$

$x(n) = \{1, 1, 1, 1; 1, 1, 1, 1\}$ $N = 8$

DIF-FFT



$X(k) = \{X(0), X(1), X(2), X(3), X(4), X(5), X(6), X(7)\}$

$= \{8, 0, 0, 0, 0, 0, 0, 0\}$

Ans.

2018

5 c) Define DFT equations. Fond 4 point DFT of the sequence x(n) = u(n). Draw amplitude, magnitude, phase, and power spectrum.

Answer:

6. a) Define DFT leakage. Explain with example (2.75) [same as 2020- 7. b)]

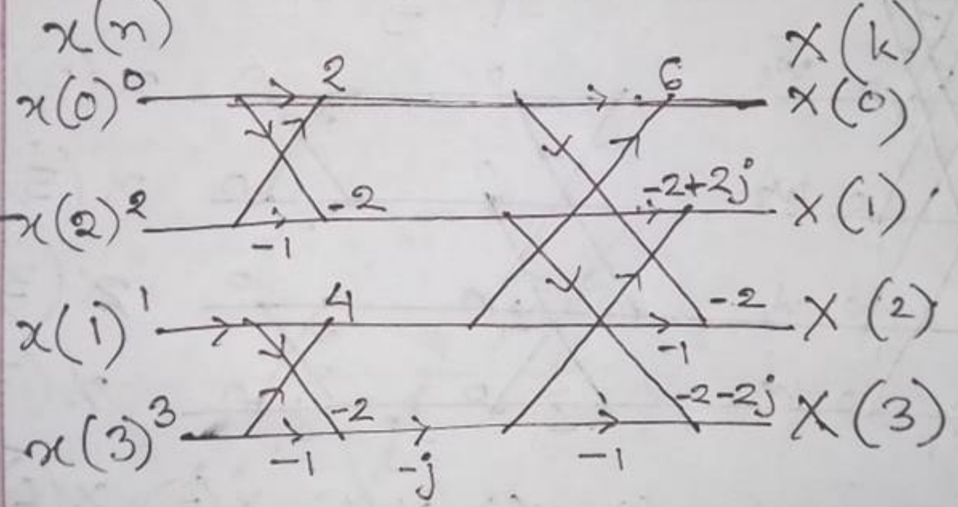
7(b) Compute 4 point DFT of a sequence x(n) = {0, 1, 2, 3} using DIT algorithm (3)

2018

7 (b) Compute 4 point DFT of a sequence $x(n) = \{0, 1, 2, 3\}$ using DIT algorithm. (3)

Given, $x(n) = \{0, 1, 2, 3\}$ $N = 4$

$x(0)$ $x(1)$ $x(2)$ $x(3)$



$\therefore X(k) = \{6, -2+2j, -2, -2-2j\}$

∴ DIT FFT

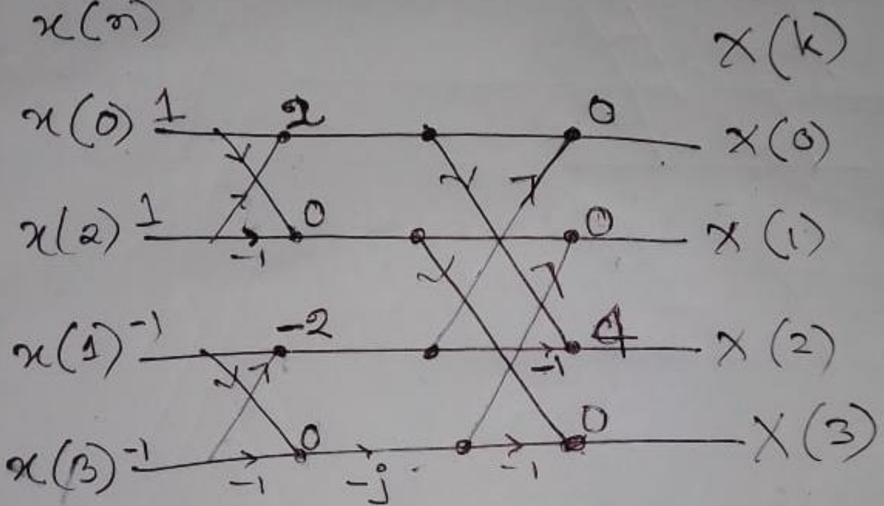
8 a) Define decimation-in-time algorithm. Draw the flow-graph of a two point DFT for a decimation-in-time decomposition (3) [Same as 2019 7-b)]

8. b) Compute DFT of a sequence x(n) = {1, -1, 1, -1} using DIT algorithm (3)

2017

8(b) Compute DFT of a sequence $x(n) = \{1, -1, 1, -1\}$ using DIT algorithm

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



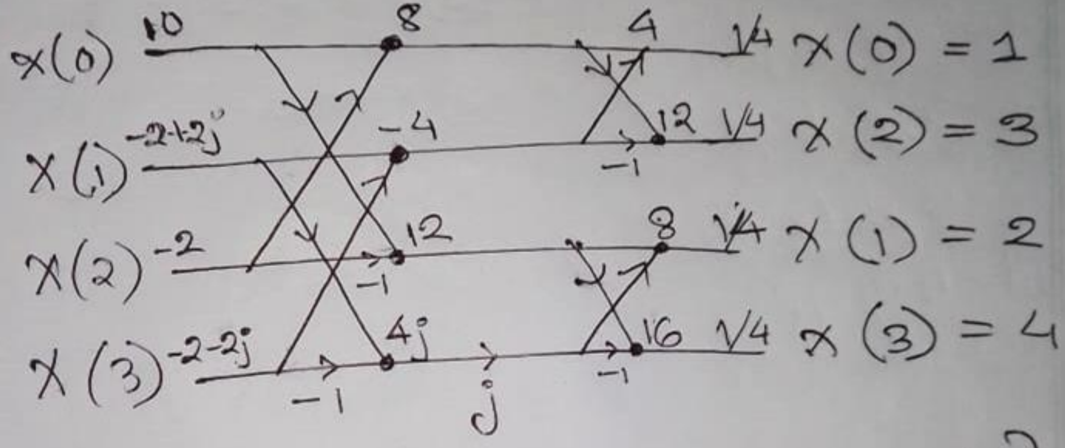
$\therefore X(K) = \{0, 0, 4, 0\}$

2017

8(c) Find the IDFT of the sequence x(n) = {10, -2+2j, -2, -2-2j} using DIT algorithm

2017

8 (c) Find the IDFT of the sequence $x(n) = \{10, -2+2j, -2, -2-2j\}$ using DIT algorithm 2.75



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

$= \{1, 2, 3, 4\}$

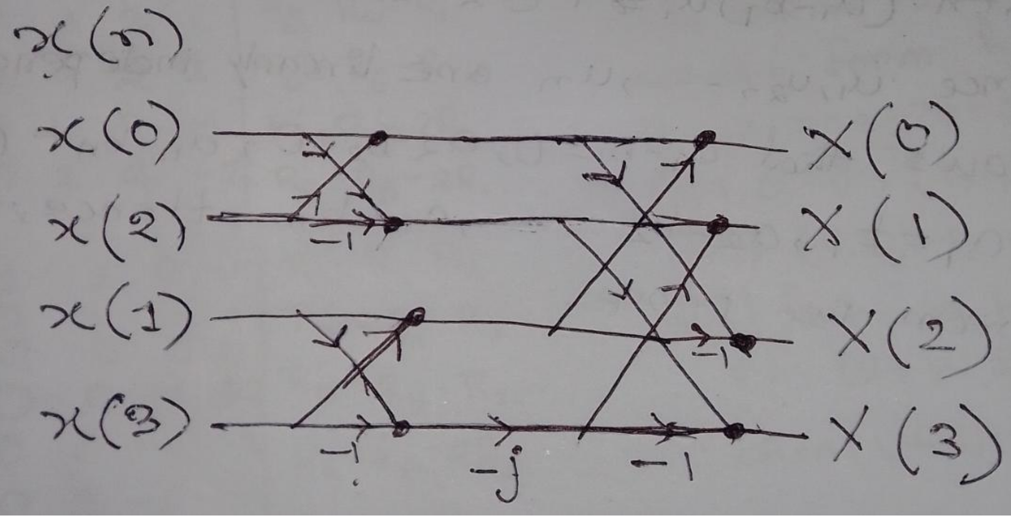
2016 - 7(c)

Draw the butterfly structure of FFT implementation for a 4-point DFT. (3)

Answer: We know FFT is of two types.

(1) Decimation in Time FFT.

The 4 point DFT for this type:-



(2) Decimation in Frequency FFT. The 4 point DFT butterfly structure:-