JAYPEE INSTITUTE OF INFORMATION TECHNOLOGY
Electronics and Communication Engineering
Digital Signal Processing (15B11EC413)
Tutorial Sheet: IV

Q1. [CO1] Assume that a complex multiply takes 1 μs and that the time to compute a DFT is determined by the time it takes to perform all of the multiplications.
a) How much time does it take to compute a 1024-point DFT directly?
b) How much time is required if an FFT is used?
c) Repeat parts (a) and (b) for a 4096- point DFT.

Sol.

(*a*) Including possible multiplications by ± 1, computing an N-point DFT directly requires N^2 complex multiplications. If it takes 1 μs per complex multiply, the direct evaluation of a 1024-point DFT requires

$$t_{DFT} = (1024)^2 \cdot 10^{-6}\text{ s} \approx 1.05\text{ s}$$

(*b*) With a radix-2 FFT, the number of complex multiplications is approximately $(N/2)\log_2 N$ which, for $N = 1024$, is equal to 5120. Therefore, the amount of time to compute a 1024-point DFT using an FFT is

$$t_{FFT} = 5120 \cdot 10^{-6}\text{ms} = 5.12\text{ ms}$$

(*c*) If the length of the DFT is increased by a factor of 4 to $N = 4096$, the number of multiplications necessary to compute the DFT directly increases by a factor of 16. Therefore, the time required to evaluate the DFT directly is

$$t_{DFT} = 16.78\text{ s}$$

If, on the other hand, an FFT is used, the number of multiplications is

$$2{,}048 \cdot \log_2 4{,}096 = 24{,}576$$

and the amount of time to evaluate the DFT is

$$t_{FFT} = 24.576\text{ ms}$$

Q2. [CO1] The DFT of a sequence x[n] that has $N = 2^v$ can be calculated using two algorithms. Algorithm A computes the DFT by direct evaluation and take N^2 seconds to run. Algorithm B implements DIT-FFT and takes $5N\log_2 N$ seconds to run. What is the shortest sequence N such that algorithm B runs faster than algorithm A.

Sol. N = 32

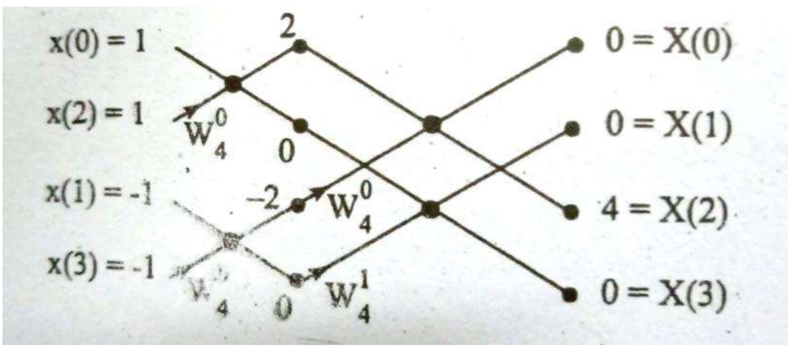
N	Algo A (N^2 seconds)	Algo B ($5N \log_2 N$ seconds)
2	4	10
4	16	40
8	64	120
16	256	320
32	**1024**	**800**

Q3. [CO1] Find the DFT of a following sequences using DIT-FFT algorithm.

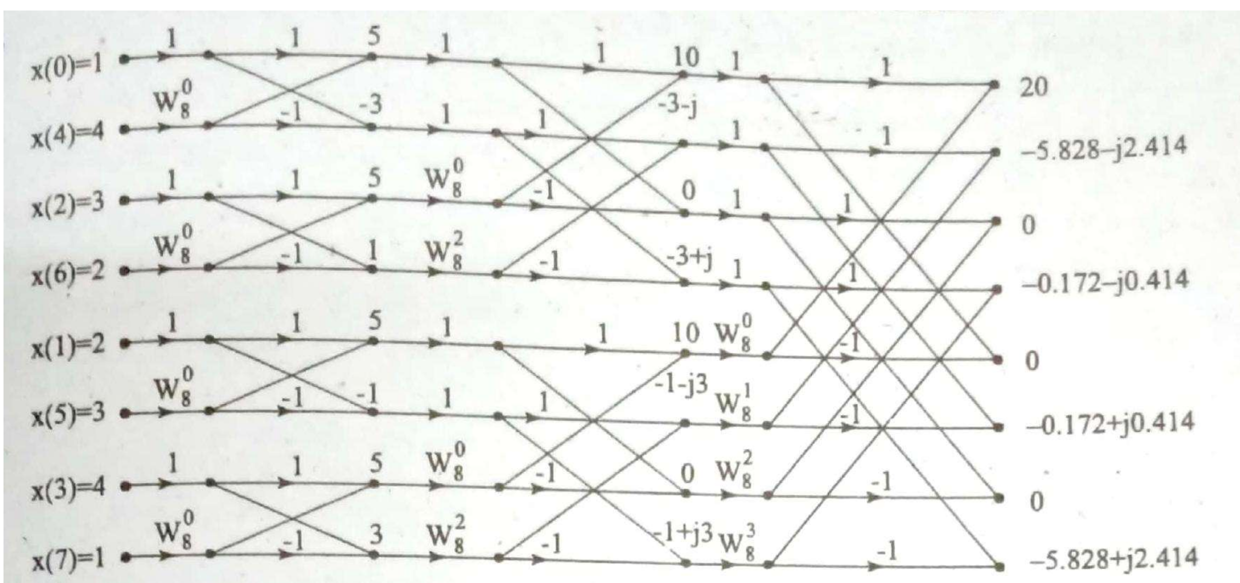
a) x[n] = {1-1,1,-1} b) x[n] = {1,2,3,4,4,3,2,1}

Sol.

a) $W_4^0 = 1, W_4^1 = -j,$



b)



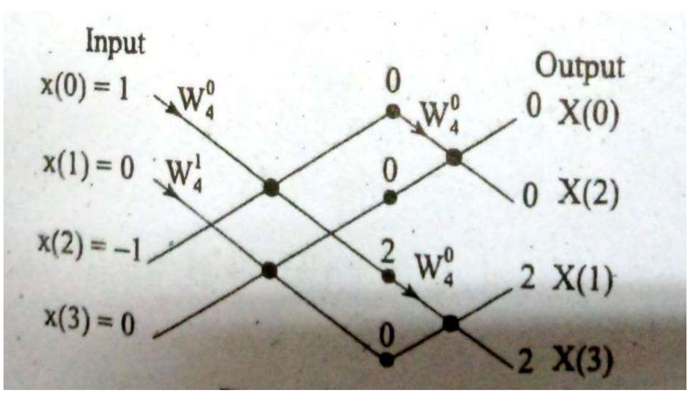
Input	Output of stage 1 (S_1)	Output of stage 2 (S_2)	Output
1	$1+4=5$	$5+5=10$	$10+10=20$
4	$1-4=-3$	$-3+(-j)1=-3-j$	$-3-j+(0.707-j0.707)(-1-3j)$ $=-5.828-j2.414$
3	$3+2=5$	$5-5=0$	0
2	$3-2=1$	$-3-(-j)1=-3+j$	$(-3+j)+(-0.707-j0.707)(-1+3j)$ $=-0.172-j0.414$
2	$2+3=5$	$5+5=10$	$10-10=0$
3	$2-3=-1$	$-1+(-j)3=-1-3j$	$-3-j-(0.707-j0.707)(-1-3j)$ $=-0.172+j0.414$
4	$4+1=5$	$5-5=0$	0
1	$4-1=3$	$-1-(-j)3=-1+3j$	$(-3+j)-(-0.707-j0.707)(-1+3j)$ $=-5.828+j2.414$

Q4. [CO1] Find the DFT of a following sequences using DIF-FFT algorithm.

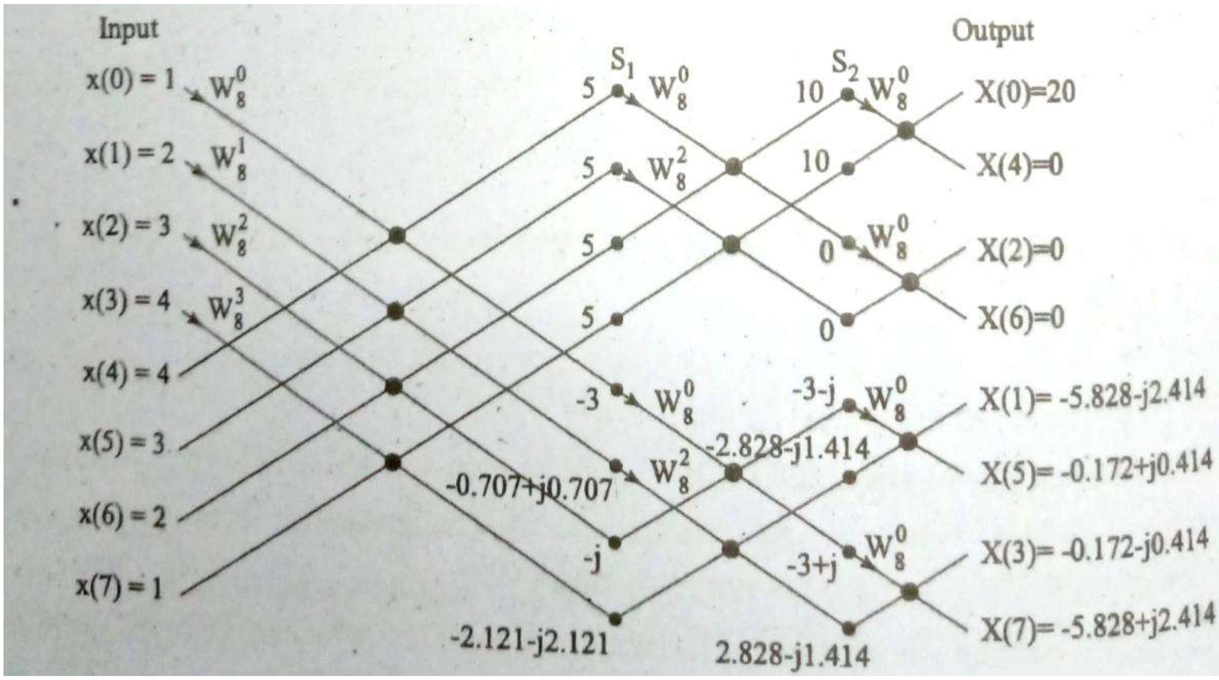
a) x[n] =cos(nπ/2), n=0,1,2,3　　　　　　　　b) x[n] = {1,2,3,4,4,3,2,1}

Sol.

a) Substituting n = 0,1,2,3 in cos(nπ/2), x[n] ={ 1, 0, -1, 0}



b)



Input	S_1	S_2	Output
1	$1+4=5$	$5+5=10$	$10+10=20$
2	$2+3=5$	$5+5=10$	$10-10=0$
3	$3+2=5$	$5-5=0$	0
4	$4+1=5$	$(5-5)W_8^2=0$	0
4	$(1-4)W_8^0=-3$	$[-3+(-j)]=-3-j$	$-3-j-2.828-j1.414$ $=-5.828-j2.414$
3	$(2-3)(0.707-j0.707)$ $=-0.707+j0.707$	$-0.707+j0.707+$ $(-2.121-j2.121)$ $=-2.828-j1.414$	$-3-j+2.828+j1.414$ $=-0.172+j0.414$
2	$(3-2)(-j)=-j$	$-3-(-j)=-3+j$	$-3+j+2.828-j1.414$ $=-0.172-j0.414$
1	$(4-1)(-0.707-j0.707)$ $=-2.121-j2.121$	$(-0.707+j0.707+2.121$ $+j2.121)(-j)$ $=2.828-j1.414$	$-3+j-2.828+j1.414$ $=-5.828+2.414j$

Q5. [CO1] Suppose that we would like to find the N-point DFT of a sequence where N is a power of 3, $N = 3^v$.

a) Develop a radix-3 decimation-in-time FFT algorithm, and draw the corresponding flowgraph for N = 9.

b) How many multiplications are required for a radix-3 FFT?

c) Can the computations be performed in place?

Sol.

(*a*) A radix-3 decimation-in-time FFT may be derived in exactly the same way as a radix-2 FFT. First, $x(n)$ is decimated by a factor of 3 to form three sequences of length $N/3$:

$$f(n) = x(3n) \qquad n = 0, 1, \ldots, \frac{N}{3} - 1$$

$$g(n) = x(3n+1) \qquad n = 0, 1, \ldots, \frac{N}{3} - 1$$

$$h(n) = x(3n+2) \qquad n = 0, 1, \ldots, \frac{N}{3} - 1$$

Expressing the N-point DFT in terms of these sequences, we have

$$X(k) = \sum_{n=0,3,6,\ldots} x(n)W_N^{nk} + \sum_{n=1,4,5,\ldots} x(n)W_N^{nk} + \sum_{n=2,5,7,\ldots} x(n)W_N^{nk}$$

$$= \sum_{l=0}^{\frac{N}{3}-1} f(l)W_N^{3lk} + \sum_{l=0}^{\frac{N}{3}-1} g(l)W_N^{(3l+1)k} + \sum_{l=0}^{\frac{N}{3}-1} h(l)W_N^{(3l+2)k}$$

Since $W_N^{3lk} = W_{N/3}^{lk}$, then

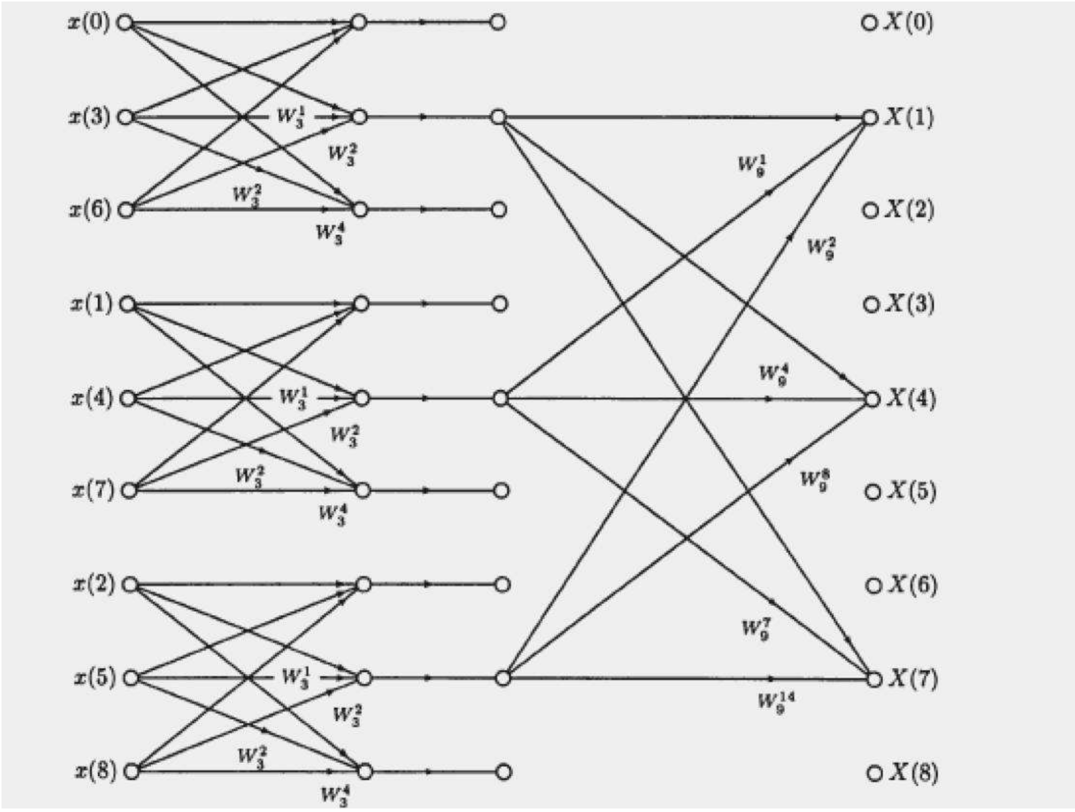
$$X(k) = \sum_{l=0}^{\frac{N}{3}-1} f(l)W_{N/3}^{lk} + W_N^k \sum_{l=0}^{\frac{N}{3}-1} g(l)W_{N/3}^{lk} + W_N^{2k} \sum_{l=0}^{\frac{N}{3}-1} h(l)W_{N/3}^{lk}$$

Note that the first term is the $N/3$-point DFT of $f(n)$, the second is W_N^k times the $N/3$-point DFT of $g(n)$, and the third is W_N^{2k} times the $N/3$-point DFT of $h(n)$,

$$X(k) = F(k) + W_N^k G(k) + W_N^{2k} H(k)$$

We may continue decimating by factors of 3 until we are left with only 3-point DFTs. The flowgraph for a 9-point decimation-in-time FFT is shown in Fig. 7-11. Only one of the 3-point butterflies is shown in the second stage in order to allow for the labeling of the branches. The complete flowgraph is formed by replicating this 3-point butterfly up by one node, and down by one node, and changing the branch multiplies to their appropriate values.

Flowgraph for a 9-point DIT FFT



(*b*) If $N = 3^\nu$, then there are ν stages in the radix-3 FFT. The general form of each 3-point butterfly, shown in the second stage of the flowgraph in Fig. 7-11, requires six multiplies (some require fewer if we do not consider multiplications by ± 1). Since there are $N/3$ butterflies in each stage, then the total number of multiplications is

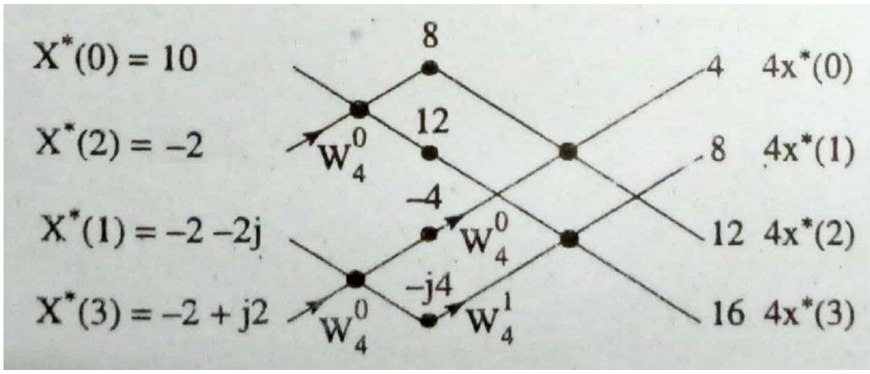
$$6N \log_3 N$$

(*c*) Yes, the computations may be performed in place.

Q6. [CO1] Find the IDFT of the following sequence using DIT-FFT algorithm.

X[k] = {10, -2 + j2, - 2, -2 - j2}

Sol.

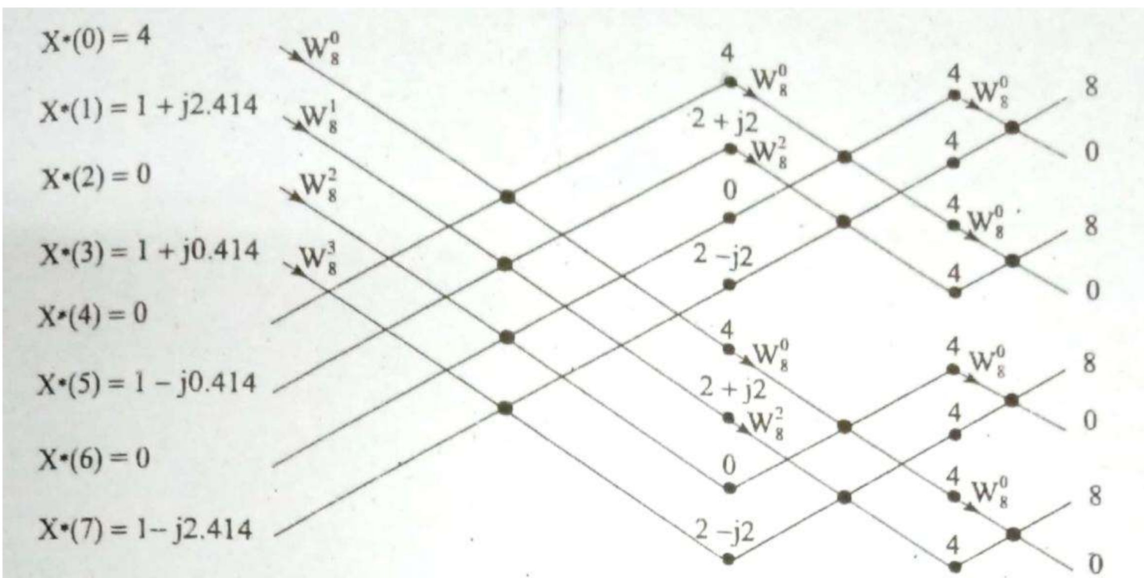


The ouyput Nx*[n] is in normal order. Therefore, x[n] = {1,2,3,4}.

Q7. [CO1] Find the IDFT of the following sequence using DIF-FFT algorithm.

X[k] = {4, 1 -j2.414, 0,1- j0.414, 0,1+ j0.414, 0, 1 + j2.414}

Sol.



The output Nx*[n] is in bit reversal order. Therefore, x[n] = {1,1,1,1,0,0,0,0}.

Q8. [CO1] Suppose that we have a number of eight-point decimation-in-time FFT chips. How could these chips be used to compute a 24-point DFT?

Sol.

A 24-point DFT is defined by

$$X(k) = \sum_{n=0}^{23} x(n) W_{24}^{nk}$$

Decimating $x(n)$ by a factor of 3, we may decompose this DFT into three 8-point DFTs as follows:

$$\begin{aligned} X(k) &= \sum_{n=0}^{7} x(3n) W_{24}^{3nk} + \sum_{n=0}^{7} x(3n+1) W_{24}^{(3n+1)k} + \sum_{n=0}^{7} x(3n+2) W_{24}^{(3n+2)k} \\ &= \sum_{n=0}^{7} x(n) W_8^{nk} + W_{24}^{k} \sum_{n=0}^{7} x(3n+1) W_8^{nk} + W_{24}^{2k} \sum_{n=0}^{7} x(3n+2) W_8^{nk} \end{aligned}$$

Therefore, if we form the three sequences

$$\begin{aligned} f(n) &= x(3n) \qquad & n &= 0, 1, 2, \ldots, 7 \\ g(n) &= x(3n+1) \qquad & n &= 0, 1, 2, \ldots, 7 \\ h(n) &= x(3n+2) \qquad & n &= 0, 1, 2, \ldots, 7 \end{aligned}$$

and use the 8-point FFT chips to find the DFTs $F(k)$, $G(k)$, and $H(k)$, the 24-point DFT of $x(n)$ may be found by combining the outputs of the 8-point FFTs as follows:

$$X(k) = F(k) + W_{24}^{k} G(k) + W_{24}^{2k} H(k)$$