UNIT-I - INTRODUCTION

PART A

1. Define Signal.

A Signal is defined as any physical quantity that varies with time, space or any other independent variables.

2. Define a system. (Nov/Dec 2014)

A System is a physical device (i.e., hardware) or algorithm (i.e., software) that performs an operation on the signal.

3. What are the steps involved in digital signal processing?

- Converting the analog signal to digital signal, this is performed by A/D converter
- Processing digital signal by a digital system.
- Converting the digital signal to analog signal, this is performed by D/A converter.

4. Give some applications of DSP? (Apr/May 2015)

- Speech processing – Speech compression & decompression for voice storage system
- Communication – Elimination of noise by filtering and echo cancellation.
- Bio-Medical – Spectrum analysis of ECG, EEG etc.

5. Write the classifications of DT Signals.

- Energy & Power signals
- Periodic & Aperiodic signals
- Even & Odd signals.

6. What is an Energy and Power signal?

Energy signal:

A finite energy signal is periodic sequence, which has a finite energy but zero average power.

Power signal:

An Infinite energy signal with finite average power is called a power signal.

7. What is Discrete Time Systems? (Nov/Dec 2014)

The function of discrete time systems is to process a given input sequence to generate output sequence. In practical discrete time systems, all signals are digital signals, and operations on such signals also lead to digital signals. Such discrete time systems are called digital filter.

8. Write the Various classifications of Discrete-Time systems. (Nov/Dec 2014)

Linear & Non linear system

Causal & Non Causal system

Stable & Un stable system

Static & Dynamic systems

9. Define Linear system

A system is said to be linear system if it satisfies super position and homogeneity principle. Let us consider x_1 (n) & x_2 (n) be the two input sequences & y_1 (n) & y_2 (n) are the responses respectively,

$$T[ax_1(n)+bx_2(n)] = ay_1(n)+b\,y_2(n)$$

10. Define Static & Dynamic systems

When the output of the system depends only upon the present input sample, then it is called static system, otherwise if the system depends past values of input then it is called dynamic system

11. Define causal system.

When the output of the system depends only upon the present and past input sample, then it is called causal system, otherwise if the system depends on future values of input then it is called non-causal system.

12. Define Shift-Invariant system.

If y(n) is the response to an input x(n), then the response to an input x(n) = x(n-k) is y(n,k). If y(n,k) = y(n – k) the above condition is said to be shift – invariant, otherwise it is shift variant.

13. Define impulse and unit step signal.

Impulse signal δ(n):

The impulse signal is defined as a signal having unit magnitude at n = 0 and zero for other values of n.

$$\delta(n) = 1 \text{ for } n = 0$$
$$= 0;\ n \neq 0$$

Unit step signal u(n):

The unit step signal is defined as a signal having unit magnitude for all values of $n \geq 0$

$$u(n) = 1 \text{ for } n \geq 0$$
$$= 0;\ n < 0$$

14. What are FIR and IIR systems?

The impulse response of a system with infinite number of samples is called IIR system & the impulse response of a system with finite number of samples is called FIR system.

15. What are the basic elements used to construct the block diagram of discrete time system?

The basic elements used to construct the block diagram of discrete time Systems are Adder, Constant multiplier &Unit delay element.

16. Define sampling theorem. (Nov/Dec 2011, May /June 2014)

A continuous time signal can be represented in its samples and recovered back if the sampling frequency $F_s \geq 2F_m$. Here 'Fs' is the sampling frequency and 'F_m' is the maximum frequency present in the signal.

17. Check the linearity and stability of sqrt x(n)

Since square root is nonlinear, the system is nonlinear.

As long as x(n) is bounded, its square root is bounded. Hence this system is stable.

18. Test whether the system governed by the relation $y(n) = \sum_{k=-\infty}^{n} x(k)$ is linear time invariant or not (NOV/DEC 2014)

A system is said to be time invariant if it satisfies the following condition

$$y(n, r) = y(n - r)$$

Therefore for the given $y(n) = \sum_{k=-\infty}^{n} x(k)$

Y(n,k) delay input by 'r' samples i.e $y(n, r) = \sum_{k=-\infty}^{n} x(k)$

Now delay output by 'r' samples i.e $y(n - r) = \sum_{k=-\infty}^{n-r} x(k)$ hence time invariant.

A system is said to be linear if it obeys superposition and homogeneity principle

$T[ax_1(n) + bx_2(n)] = ay_1(n) + b\,y_2(n)$. Therefore

For two input sequences the corresponding outputs are

$$y_1(n) = \sum_{k=-\infty}^{n} x_1(k) \text{ and } y_2(n) = \sum_{k=-\infty}^{n} x_2(k) \quad \rightarrow (a)$$

A linear combination of two input sequences results in the output

$$y_3(n) = T[ax_1(n) + bx_2(n)] = a\sum_{k=-\infty}^{n} x_1(k) + b\sum_{k=-\infty}^{n} x_2(k) \quad \rightarrow (b)$$

A linear combination of two outputs results in the output

$$ay_1(n) + by_2(n) = a\sum_{k=-\infty}^{n} x_1(k) + b\sum_{k=-\infty}^{n} x_2(k) \quad \rightarrow (c)$$

Since (b) = (c) hence linear. Therefore the given system is linear time variant.

19. What is aliasing? (Nov/Dec 2014)

When the sampling frequency is less than twice of the highest frequency content of the signal, then the high frequency signal will interfere with the low frequency and appears as low frequency known as aliasing.

20. Consider the analog signal $x(t) = 3\cos 50\pi t + 10\sin 300\pi t - \cos 100\pi t$. What is the Nyquist rate for the signal? (MAY/JUNE 2014, NOV?DEC 2013)

In the given signal $x(t) = 3\cos 50\pi t + 10\sin 300\pi t - \cos 100\pi t$ the frequencies are

$F_1 = 50\pi / 2\pi = 25\text{Hz}, F_2 = 300\pi / 2\pi = 150\text{Hz}, F_3 = 100\pi / 2\pi = 25\text{Hz}$. The maximum frequency Fm is 150Hz, therefore the Nyquist rate = 2 × Fm = 300 Hz.

21. Determine the fundamental period of the signal $x(n) = \cos\left(\frac{\pi 30n}{105}\right)$ (NOV/DEC 2014)

Given $x(n) = \cos\left(\frac{\pi 30n}{105}\right)$ and $\omega_0 = \left(\frac{\pi 30n}{105}\right), 2\pi f_0 = \left(\frac{\pi 30}{105}\right), f_0 = \frac{1}{7}$. Therefore the fundamental period is 7.

22. Determine whether x(n) = u(n) is a power signal or an energy signal? (MAY?JUNE 2013)

Given x(n) = u(n),

Energy signal $\sum_{n=-\infty}^{\infty} |x(n)|^2 = \sum_{n=-\infty}^{\infty} |u(n)|^2 = \sum_{n=0}^{\infty} 1 = 1+1+1+.....+\infty = \infty$

Power signal $P = \underset{N\to\infty}{\text{Lt}} \left(\frac{1}{2N+1}\right)^2 \sum_{n=-N}^{N} |x(n)|^2 = \underset{N\to\infty}{\text{Lt}} \left(\frac{1}{2N+1}\right)^2 \sum_{n=0}^{N} 1$

$$= \underset{N\to\infty}{\text{Lim}} \left(\frac{N+1}{2N+1}\right)^2 = \underset{N\to\infty}{\text{Lim}} \left(\frac{N\left(1+\frac{1}{N}\right)}{N\left(2+\frac{2}{N}\right)}\right)^2 = \frac{1}{2}$$

Therefore $P = \frac{1}{2} \,\&\, E = \infty$ hence the given signal is power signal.

23. What is an LTI system? (Nov/Dec 2012)

A system which obeys two of the basic property linearity and time invariance is called LTI system.

24. Define Nyquist rate (May/June 2012)

If the sampling frequency is ω_s, all the frequencies above $\omega_s/2$ (folding frequency) cause aliasing. This aliasing can be avoided if the input signal frequencies are below one half of the sampling frequencies. This frequency is called Nyquist frequency or Nyquist rate.

25. Differentiate between Energy and Power signals (Apr/May 2015)

Energy signals are time limited while power signals can exist over infinite time

Non periodic signals are energy signals while power signals are periodic

Power of an energy signal is zero and the energy of a power signals is infinite

26. What are the difference between discrete signal and digital signal? (Nov/Dec 2016)

S.No	Discrete signal	Digital Signal
1.	All discrete signals are not digital	All digital signals are discrete time signals
2.	Dependent variable should be discrete or continuous but independent variable should be discrete	Both dependent variable and independent variable should be discrete

27. If x(n) = x(n + 1) + x(n – 2)is the system casual? (NOV/DEC 2016)

Solution: x(0) = x(1) + x(-2) for n = 0

Since output depends upon future input, the system is not causal.

28. What is Nyquist rate of the analog signal? (Nov/Dec 2016)

When the sampling rate becomes exactly equal to 2B (i,e. $2F_{max}$) samples per second for the signal bandwidth of 'B' Hz, then it is called Nyquist rate. Then the nyquist interval is obtained as,

$$\text{Nysquist interval } = \frac{1}{2B} = \frac{1}{2F_{max}}$$

And

$$\text{Nyquist rate } = 2B = 2F_{max}$$

29. What is quantization error? (Nov/Dec 2016)

The difference between the actual analog value and quantized digital value due is called quantization error. This error is due either to rounding or truncation.

Thus the quantization error $e_q(n)$ is defined as,

Quantization error $e_q(n) = x_q(n) - x(n)$, where x(n) is discrete time signal and $x_q(n)$ is quantized signal by rounding.

PART B

1. For the discrete time systems given below (Nov/Dec 2013) (16)

$$\text{i)}\, y(n) = x(n)\cos(\omega_0 n) \qquad \text{(ii)}\, y(n) = x(n) + nx(n+1)$$
$$\text{(iii)}\, y(n) = x(2n) \qquad \text{(iv)}\, y(n) = \cos x(n)$$

Check whether these systems are a) Static or Dynamic b) Linear or Non – Linear c) Time Invariant or Time Variant d) Causal or Non Causal e) Stable or Unstable

Solution:

$$i)\ y(n) = x(n)\cos(\omega_0 n)$$

1. This is a static system since, the output of the system depends only upon the present input sample i.e, nth output sample depends upon nth input sample. Hence this is a static system.

2. We know that the given system is,

$$y(n) = T\{x(n)\} = x(n)\cos(\omega_0 n)$$

When the two inputs $x_1(n)\ \&\ x_2(n)$ are applied separately, the responses $y_1(n)\ \&\ y_2(n)$ will be,

$$\left.\begin{aligned} y_1(n) &= T\{x_1(n)\} = x_1(n)\cos(\omega_0 n) \\ y_2(n) &= T\{x_2(n)\} = x_2(n)\cos(\omega_0 n) \end{aligned}\right\} \quad \rightarrow (1)$$

The response of the system due to linear combination of inputs will be,

$$\begin{aligned} y_3(n) &= T\{a_1x_1(n) + a_2x_2(n)\} = [a_1x_1(n) + a_2x_2(n)]\cos(\omega_0 n) \\ &= a_1x_1(n)\cos(\omega_0 n) + a_2x_2(n)\cos(\omega_0 n) \qquad \rightarrow (2) \end{aligned}$$

Now the linear combination of the two outputs will be,

$$y_3(n) = a_1y_1(n) + a_2y_2(n)$$

$= a_1x_1(n)\cos(\omega_0 n) + a_2x_2(n)\cos(\omega_0 n)$ from equation (1)

From above equation and equation (2)

$$y_3(n) = y_3(n)$$

Hence the system is linear.

3. The system equation is,

$$y_3(n) = x(n)\cos(\omega_0 n)$$

The response of the system to delayed input will be,

$$\begin{aligned} y(n,k) &= T\{x(n-k)\} \\ &= x(n-k)\cos(\omega_0 n) \qquad \rightarrow (3) \end{aligned}$$

Now let us delay or shift the output y(n) by ‘k’ samples i.e,

$$y(n-k) = x(n-k)\cos[\omega_0(n-k)]$$

Here every ‘n’ is replaced by n – k. On comparing above equation with equation(3) we find that,

$y(n,k) \neq y(n-k)$ hence the system is shift variant.

4. In the given system, y(n) depends upon x(n), i.e present output depends upon present input. Hence the system is casual.

5. The given system equation is,

$$y(n) = x(n)\cos(\omega_0 n)$$

Here value of cos $(\omega_0 n)$ is always bounded. Hence as long as x(n) is bounded, y(n) is also bounded. Hence the system is stable.

This system is,

Static, linear, shift variant, casual and stable.

(ii) y(n) = x(n) +nx(n+1):

1. From the given equation it is clear that, the output depends upon the present input and next input. Hence system is dynamic.

2. The given system equation is,

$$y(n) = T\{x(n)\} = x(n) + nx(n+1) \qquad \rightarrow (4)$$

If we apply two inputs x_1 (n) and x_2(n) separately, then the outputs become,

$$\left.\begin{aligned} y_1(n) &= T\{x_1(n)\} = x_1(n) + nx_1(n+1) \\ y_2(n) &= T\{x_2(n)\} = x_2(n) + nx_2(n+1) \end{aligned}\right\} \rightarrow (4.a)$$

Response of the system to linea combination of inputs x_1(n) and x_2(n) will be,

$$\begin{aligned} y_3(n) &= T\{a_1x_1(n) + a_2x_2(n)\} \\ &= a_1[x_1(n) + nx_1(n+1)] + a_2[x_2(n) + nx_2(n+1)] \qquad \rightarrow (5) \end{aligned}$$

The linear combination of two outputs given by equation (4.a) will be,

$$\begin{aligned} y_3(n) &= a_1y_1(n) + a_2y_2(n) \\ &= a_1[x_1(n) + nx_1(n+1)] + a_2[x_2(n) + nx_2(n+1)] \end{aligned}$$

On comparing above equation with equation (4.a) we observe that,

$$y_3(n) = y_3(n)$$

Hence the system is linear.

3. The given system equation is

$$y(n) = T\{x(n)\} = x(n) + nx(n+1) \qquad \rightarrow (6)$$

Response of the system to delayed input will be,

$$y(n,k) = T\{x(n-k\}$$
$$= x(n-k) + nx(n-k+1) \qquad \rightarrow (7)$$

Now let us delay the output of equation (5) 'k' samples i.e,

$$y(n-k) = x(n-k) + (n-k)x(n-k+1)$$

Here we have replaced 'n' by 'n-k'. On comparing above equation with equation (7) we observe that,

$$y(n,k) \neq y(n-k)$$

Hence the system is shift variant.

4. The given system equation is,

$$y(n) = x(n) + nx(n+1)$$

Here observe that nth output sample depends upon (n+1)t hi.e next input sample.

That is the output sample depends upon future input. Hence the system is noncasual.

5. In the given system equation observe that as $n \rightarrow \infty, y(n) \rightarrow \infty$ even if x(n) is bounded. Hence the system unstable.

Thus the given system is,

Dynamic, linear, shift variant, non casual and stable

(iii) y(n) = x(2n)

1. By putting n = 1 in the given system equation,

Y(1) = x(2) similarly,

$$\begin{aligned} n = 2 &\Rightarrow y(2) = x(4) \\ n = 3 &\Rightarrow y(3) = x(6) \end{aligned} \quad \text{and so on.}$$

Thus the system needs to store the future input samples. Hence it requires memory.

Therefore the system is dynamic.

2. The given system equation is,

$$y(n) = T\{x(n)\} = x(2n) \qquad \rightarrow (8)$$

For two separate inputs $x_1(n)$ and $x_2(n)$ the system produces the response of

$$\left.\begin{aligned} y_1(n) = T\{x_1(n)\} = x_1(2n) \\ y_2(n) = T\{x_2(n)\} = x_2(2n) \end{aligned}\right\} \qquad \rightarrow (9)$$

The response of the system to linear combination of $x_1(n)$ and $x_2(n)$ will be.

$$y_3(n) = T\{a_1x_1(n) + a_2x_2(n)\}$$
$$= a_1x_1(2n) + a_2x_2(2n) \qquad \rightarrow (10)$$

Now the linear combination of two outputs given by equation (9) will be,

$$y_3(n) = a_1\, y_1(n) + a_2y_2(n)$$
$$= a_1x_1(2n) + a_2x_2(2n)$$

On comparing above equation with equation (10) we find that,

$$y_3(n) = y_3(n).$$

Hence the system is linear.

3. The given system equation is,

$$y(n) = T\{x(n)\} = x(2n) \qquad \rightarrow (11)$$

The response of the system to delayed input x(n – k) will be,

$$y(n,k) = T\{x(n-k)\} = x(2n-k) \qquad \rightarrow (12)$$

Now let us delay the output y(n) given by equation (11) by 'k' sample. This is obtained by replacing 'n' by n-k in equation (11) i.e,.

$$y(n-k) = x\left[2(n-k)\right]$$
$$y(n-k) = x\left(2n-2k\right)$$

On comparing above equation (12) we find tht,

$$y(n,k) \neq y(n-k)$$

Hence the system is shift variant.

4. The given system equation is,

$$y(n) = x(2n)$$

Here output depends upon future inputs i.e, n^{th} sample of output depends upon $(2n)^{th}$ sample of input. Clearly the system is non casual.

5. As long as x(n) is bounded, then x(2n) is also bounded. Hence output y(n) is also bounded. Therefore the system is stable.

Thus the given system is

Dynamic, linear, shift variant, non casual and stable

(iv) y(n) = cos [x(n)]:

1. A system is static if its output depends only upon the present input sample. Here since y(n) depends upon the cosine of x(n), i.e, present input sample, the system is static.

2. For two separate inputs the system produces the response of,

$$y_1(n) = T\{x_1(n)\} = \cos[x_1(n)]$$
$$y_2(n) = T\{x_2(n)\} = \cos[x_2(n)]$$

The response of the system to linear combination of two inputs will be,

$$y_3(n) = T\{a_1x_1(n) + a_2x_2(n)\} = \cos[a_1x_1(n) + a_2x_2(n)]$$

The linear combination of two outputs will be,

$$y_3(n) = a_1y_1(n) + a_2y_2(n) = a_1\cos[x_1(n)] + a_2\cos[x_2(n)]$$

Clearly $y_3(n) \neq y_3(n)$ hence system is non linear.

3. The system is said to be shift variant or time invariant if its. Characteristics do not change with shift of time origin. The given system is,

$$y(n) = T\{x(n)\} = \cos[x(n)] \qquad \rightarrow (13)$$

Let us delay the input by k samples. Then output will be,

$$y(n,k) = T\{x(n-k\} = \cos[x(n-k)] \qquad \rightarrow (14)$$

Now let us delay the output y(n) given by equation (13) by 'k' samples, i.e, y(n-k). This is equivalent to replacing n by n – k in equation (13) i.e,

$$y(n,k) = y(n-k)$$

This shows that the system is shift invariant.

4. The system is said to be casual if output depends upon past and present inputs only,

The output is given as,

$$y(n) = \cos[x(n)]$$

Here observe that nth sample of output depends upon nth sample of input x(n).

Hence the system is a casual system.

5. For any bounded value of x(n) the cosine function has bounded value. Hence y(n) has bounded value. Therefore the system is said to be BIBO stable.

Thus the given system is,

Static, nonlinear, shift invariant, casual and stable

2. Consider the signal $x_a(t) = 10 \cos 2\pi (1000)t + 5 \cos 2\pi (5000)t$ is to be sampled (16)

Determine the Nyquist rate for this signal.

If the signal is sampled at 4 Khz, will the signal be recovered from its samples?

Solution:-

(i) Consider the given signal

$$x_a(t) = 10\cos 2\pi(1000)t + 5\cos 2\pi(5000)t$$

Clearly, this signal contain two cosine waves

$$F_1 = 100\text{Hz and } A_1 = 10$$
$$f_2 = 5000\text{Hz and } A_2 = 5$$

Since the highst frequency in the given is $B = F_{max} = 5000$ Hz, the nyquist rate is given from equation (13) as,

Nyqusit rate $= 2F_{max} = 2 \times 500\text{Hz} = 10\text{kHz}$

(ii) This signal is sampled at F1 = 4000 Hz when the signal $x_a(t)$ is sampled, we get x(n) as,

$$\begin{aligned} x(n) &= 10\cos 2\pi \frac{F_1}{F_s} n + 5\cos 2\pi \frac{F_2}{F_s} n \\ &= 10\cos 2\pi \frac{1000}{4000} n + 5\cos 2\pi \frac{5000}{4000} n \\ &= 10\cos 2\pi \left(\frac{1}{4}\right) n + 5\cos 2\pi \left(\frac{5}{4}\right) n \end{aligned}$$

The second term can be rearranged in above equation as,

$$\begin{aligned} x(n) &= 10\cos 2\pi \left(\frac{1}{4}\right) n + 5\cos 2\pi \left(1 + \frac{1}{4}\right) n \\ &= 10\cos 2\pi \left(\frac{1}{4}\right) n + 5\cos\left(2\pi + 2\pi \frac{1}{4} n\right) \qquad \rightarrow (14) \end{aligned}$$

Here the second term can be simplified as,

$$\cos(2\pi nk + \phi) \qquad \text{for}, k = 1, 2, 3.....$$

Hence we can write equation (14) as,

$$\begin{aligned} x(n) &= 10\cos 2\pi \left(\frac{1}{4}\right) n + 5\cos 2\pi \left(\frac{1}{4}\right) n \\ &= 15\cos 2\pi \left(\frac{1}{4}\right) n \qquad \rightarrow (15) \end{aligned}$$

This is the sampled signal. observe that there is only one frequency $f = \frac{1}{4}$ in the sampled signal. when this signal is reconstructed we get,

$$x_a(t) = 15\cos 2\pi(1000)t \qquad \rightarrow (16) \text{ from equation (15)}$$

Here observe that the reconstructed signal contains only one frequency of 100 Hz and amplitude of F1 = 1000 Hz is increased. This shows that with the sampling rate 400 Hz, the signal is not recovered from its samples. This is because the sampling theorem is not satisfied. The nyquist rate as calculated (i) is 10 KHz. Hence minimum sampling frequency should be 10kHz to avoid aliasing.

3. Determine whether the following discrete time signals are periodic or not? If periodic, determine the fundamental period. (16)

a) $x(n) = \cos(0.01\pi n)$

b) $x(n) = \cos(3\pi n)$

c) $x(n) = \sin(3n)$

d) $x(n) = \cos\left(\frac{n}{8}\right)\cos\left(\frac{\pi n}{8}\right)$

Solution:-

The discrete time signal is periodic if its frequency 'f0' can be expressed as ratio of two integers (i.e) 'f

' is rational.

$$f_0 = \frac{k}{N}$$

Here 'k' is some integer and 'N' is fundamental period

(i) cos (0.01πn):

Here $\omega = 0.01\pi$

Since $\omega = 2\pi f$, we have $f = \frac{\omega}{2\pi}$

$$f = \frac{0.01\pi}{2\pi}$$
$$= \frac{1}{200}$$

Here K = 1 and N = 200. Thus 'f' is expressed as ratio of two integers. Hence given sequence is periodic. Here period , N = 200 samples. The relation $N = \frac{1}{f}$ holds only when k =1. Earlier we have considered all signals which have k = 1. Hence we have used the relation $N = \frac{1}{f}$. but this is not always true.

(ii) cos 3πn:

Here $\omega = 3\pi$

We have $f = \frac{\omega}{2\pi} = \frac{3\pi}{2\pi} = \frac{3}{2}$

We know that $f = \frac{k}{N}$ and if k and N are integers, then signal is periodic with period 'N'

Thus the given signal is periodic with period N = 2 samples.

(iii) sin (3n)

Here $\omega = 3\pi$

We have $f = \frac{\omega}{2\pi} = \frac{3}{2}$

Here k = 3 and N =2π which is not an integer. Thus 'f' cannot be expressed as the ratio of two integers, therefore this signal is non periodic.

(iv) $\cos\left(\frac{n}{8}\right)\cos\left(\frac{\pi n}{8}\right)$:

Here we can write given sequence as,

$\cos\omega_1 \cos\omega_2$

And

$$\omega_1 = \frac{1}{8}, \omega_2 = \frac{\pi}{8}$$

$$f_1 = \frac{\omega_1}{2\pi}$$
$$= \frac{1/8}{2\pi}$$
$$= \frac{1}{16\pi}$$

Here k1 = 1 and N1 = 16π which is not an integer. Hence f1 is not the ratio of two integers,

Therefore $\cos\left(\frac{n}{8}\right)$ is non periodic.

Now

$$f_2 = \frac{\omega_2}{2\pi}$$
$$= \frac{\pi/8}{2\pi}$$
$$= \frac{1}{16}$$

'f2' is the ratio of two integers with N2 = 16. Hence $\cos\left(\frac{\pi}{8}\right)$ is periodic. The given signal is thus the product of non periodic signal and periodic signal. hence the product signal $\cos\left(\frac{n}{8}\right)\cos\left(\frac{\pi n}{8}\right)$ is non periodic.

4. Explain in detail about analog to digital conversion with suitable block diagram and to reconstruct the analog signal. (May/June 2014) (16)

Solution:-

A/D conversion

Most of the signals are analog in nature when they are generated from the primary source. Analog signals are continuous in time and amplitude. The signals such as speech, video, rader, seismic signal, ECG, EMG etc signals are basically analog in nature. If these signals are to be processed digitally, they should be converted to digital form. Such conversion is performed by analog to digital (A?D) convertors. Thus A/D and D?A convertors are always used with DSP system.

It consists of three basic blocks a) Sampler b) Quantizer c) Encoder

Basic block diagram of A/D converter

SAMPLING

The continuous time signal xa(t) is converted to discrete time signal x(n) by sampling.

The sampler takes the samples at regular time intervals. The sampling interval is denoted by T.

Then continuous time variable 't' and discrete time variable 'n' are related as,

$$t = nT \quad n = 0,1,2,3..... \qquad \rightarrow (1)$$

This discrete time signal is discrete time signal x(n) is defined only at n = 0, 1, 2, 3... and its value is equal to x_a (nT). Here T is sampling interval

$$x_a(t) = x_a(nT) = x(n) \qquad n = 0,1,2,3.... \qquad \rightarrow (2)$$

The sampler is norrmally a switch and it chops off the incoming analog signal. it is illustrated in fig.1 as shown in the figure, the sampler switch switch operates at the sampling rate of Fs, which is given as,

$$\text{Sampling rate } F_s = \frac{1}{\text{sampling interval } T} \qquad \rightarrow (3)$$

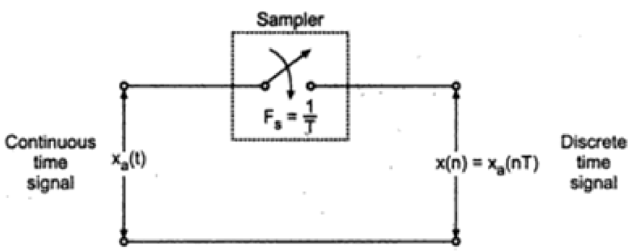


Fig.1 Sampler

Here 'Fs' is also called sampling frequency. Thus 'Fs' can be expressed as samples per second or in hertz. The sampling interval T is in seconds. Fig 2 shows the input signal xa (t) and its sampled version x(nT) or x(n). in this figure observe that x(n) takes the

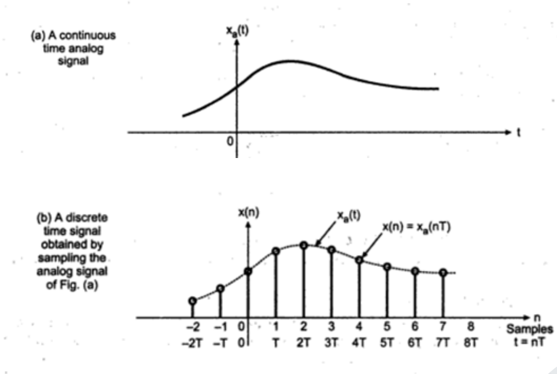


Fig. 2 Sampling of the analog signals

Amplitude of x_a (t) at the sampling instants t = nT. The values of 'n' can be only positive for real time processing. But 'n' is considered in the range $-\infty < n < \infty$ for mathematical simplicity. Hence we can rewrite equation. As,

$$x(n) = x_a(nT) \qquad -\infty < n < \infty \qquad \rightarrow (4)$$

Since t = nT and $F_s = \frac{1}{T}$ we can write an important relationship as,

$$t = nT = \frac{n}{F_s} \qquad \rightarrow (5)$$

This equation gives the relationship between time 't', samples 'n' and sampling frequency F_1.

SAMPLING THEOREM

It is the process of converting continuous time signal into a discrete time signal by taking samples of the continuous time signal at discrete time instants.

X[n]= Xa(t) where t= nTs = n/Fs ….(1)

When sampling at a rate of fs samples/sec, if k is any positive or negative integer, we cannot distinguish between the samples values of fa Hz and a sine wave of (fa+ kfs) Hz. Thus (fa + kfs) wave is alias or image of fa wave.

Thus Sampling Theorem states that if the highest frequency in an analog signal is Fmax and the signal is sampled at the rate fs > 2Fmax then x(t) can be exactly recovered from its sample values. This sampling rate is called Nyquist rate of sampling. The imaging or aliasing starts after Fs/2 hence folding frequency is fs/2. If the frequency is less than or equal to 1/2 it will be represented properly.

Example:

Case 1: $X1(t) = \cos 2\pi (10) t$ Fs= 40 Hz i.e t= n/Fs

$x1[n]= \cos 2\pi (n/4)= \cos (\pi /2)n$

Case 2: $X1(t) = \cos 2 \pi (50) t$ Fs= 40 Hz i.e t= n/Fs

$x1[n]= \cos 2 \pi (5n/4)= \cos 2 \pi (1+ ¼)n = \cos (\pi /2)n$

Thus the frequency 50 Hz, 90 Hz , 130 Hz … are alias of the frequency 10 Hz at the sampling rate of 40 samples/sec

QUANTIZATION

The process of converting a discrete time continuous amplitude signal into a digital signal by expressing each sample value as a finite number of digits is called quantization. The error introduced in representing the continuous values signal by a finite set of discrete value levels is called quantization error or quantization noise.

Example: $x[n] = 5(0.9)n\ u(n)$ where $0 < n < \infty$ & fs= 1 Hz

N	[n]	Xq [n] Rounding	Xq [n] Truncating	eq [n]
0	5	5.0	5.0	0
1	4.5	4.5	4.5	0
2	4.05	4.0	4.0	-0.05
3	3.645	3.6	3.6	-0.045
4	3.2805	3.2	3.3	0.0195

Quantization Step/Resolution : The difference between the two quantization levels is called quantization step. It is given by Δ = XMax – xMin / L-1 where L indicates Number of quantization levels.

CODING/ENCODING

Each quantization level is assigned a unique binary code. In the encoding operation, the quantization

sample value is converted to the binary equivalent of that quantization level. If 16 quantization levels are present, 4 bits are required. Thus bits required in the coder is the smallest integer greater than or equal to Log2 L. i.e b= Log2 LThus Sampling frequency is calculated as fs=Bit rate / b.

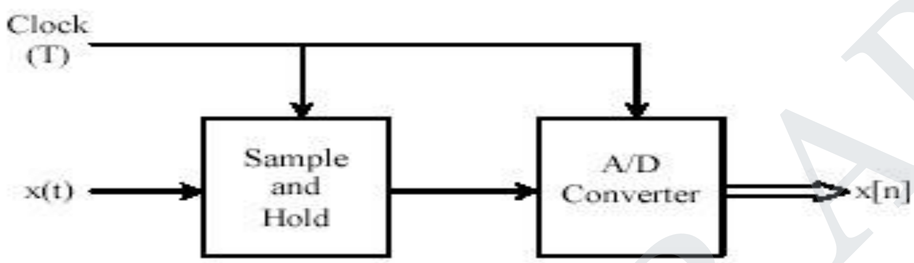
ANTI-ALIASING FILTER

When processing the analog signal using DSP system, it is sampled at some rate depending upon the bandwidth. For example if speech signal is to be processed the frequencies upon 3khz can be used. Hence the sampling rate of 6khz can be used. But the speech signal also contains some frequency components more than 3khz. Hence a sampling rate of 6khz will introduce aliasing. Hence signal should be band limited to avoid aliasing.

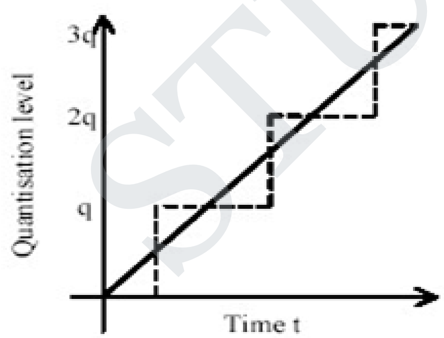
The signal can be band limited by passing it through a filter (LPF) which blocks or attenuates all the frequency components outside the specific bandwidth. Hence called as Anti aliasing filter or pre- filter as shown in block diagram.

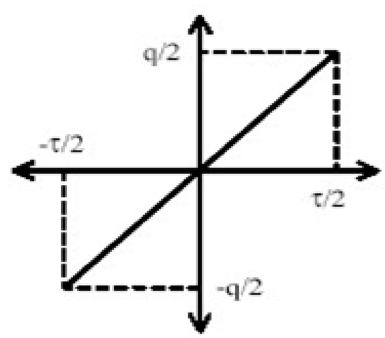
SAMPLE-AND-HOLD CIRCUIT

The sampling of an analogue continuous-time signal is normally implemented using a device called an analogue-to-digital converter (A/D). The continuous-time signal is first passed through a device called a sample-and-hold (S/H) whose function is to measure the input signal value at the clock instant and hold it fixed for a time interval long enough for the A/D operation to complete. Analogue-to-digital conversion is potentially a slow operation, and a variation of the input voltage during the conversion may disrupt the operation of the converter. The S/H prevents such disruption by keeping the input voltage constant during the conversion. This is schematically illustrated by Figure.



After a continuous-time signal has been through the A/D converter, the quantized output may differ from the input value. The maximum possible output value after the quantization process could be up to half the quantization level q above or q below the ideal output value. This deviation from the ideal output value is called the quantization error. In order to reduce this effect, we increases the number of bits.





Block diagram of a DSP system for processing of an analog singal

5. Find which of the following signals are energy signals and power signals.

(a) 4 (b) cost + cos2t (c) $e^{-2|t|}$ (d)$e^{j2\pi t}$ (Nov/Dec 2014) (16)

Solution:- Recall that the total energy of a signal f(t) is given by

$$E = \int_{-\infty}^{\infty} |f(t)|^2 dt$$

A signal is an "energy signal" if its total energy content is finite. If E = ∞, then if

$$\lim_{T\to\infty} \frac{1}{T} \int_{-T/2}^{T/2} |f(t)|^2 dt < \infty$$

The signal called is power signal.

Thus in our case:

(a)

$$\int_{-\infty}^{\infty} |4|^2 dt = \infty$$

Is NOT an energy signal, while,

$$\lim_{T\to\infty} \frac{1}{T} \int_{-T/2}^{T/2} |4|^2 dt = 16W$$

Shows that it is a power signal.

(b)

$$\int_{-\infty}^{\infty} |\cos t + \cos 2t|^2 dt = \infty$$

(Consider the integral as an area under the curve of the integrated). This shows that it is not an energy signal, while,

$$\lim_{T\to\infty} \frac{1}{T} \int_{-T/2}^{T/2} |\cos t + \cos 2t|^2 dt$$

$$= \lim_{T\to\infty} \frac{1}{T} \left[\int_{-T/2}^{T/2} \cos^2 t + \cos^2 2t + 2\cos t \cos 2t \, dt \right]$$

Using the identities (i) $\cos^2 t = \frac{1}{2}(1+\cos 2t)$, and (ii) 2cos 2t cost = cos (2t+t) + cos (2t-t) = cos 3t + cost, the value of the integral is computed as,

$$P = \frac{1}{2} + \frac{1}{2} = 1W$$

(c) Observe that,

$$\int_{-\infty}^{\infty}\left(e^{-2|t|}\right)^2 dt = \int_{-\infty}^{\infty} e^{-4|t|}dt = 2\int_{0}^{\infty} e^{-4t}dt = \frac{2}{-4}(0-1) = 0.5J$$

This is therefore an energy signal.

(d) $$\int_{-\infty}^{\infty}\left|e^{j2\pi t}\right|^2 dt = \int_{-\infty}^{\infty}|1|^2 dt = \infty \Rightarrow \text{Not an energy signal}$$

$$P = \frac{1}{T}\int_{-T/2}^{T/2}|1|^2 dt = 1W \Rightarrow \text{Power signal}$$

6. Determine if each of the following signals is periodic. If a signal is periodic, determine its fundamental period.

(a) $\cos\sqrt{3}t$ (b) $e^{(j2\pi-\pi/4)}$

(c) $\sin^2(2t)$ (d) $\sin(2t^2)$

Solution:- Recall that a signal f(t) is said to be periodic if there exists a number T such that for all t, f(t) = f(t+T). The number T is the period of the signal.

(a) $\cos\left(\sqrt{3}t\right)$ is periodic because a number T can be found such that $\cos\left(\sqrt{3}t\right) = \cos\left[\sqrt{3}\left(t+T\right)\right]$. This number is easily found to be $T = 2\pi/\sqrt{3}$ (note that we are referring to the smallest possible number here.)

(b) $$e^{\left[j\pi(t+T)-\pi/4\right]} = e^{j2\pi t}e^{-\pi/4}e^{j2\pi T} = e^{j2\pi t}e^{-\pi/4} \text{ (if T = 1)}$$

(Since $e^{j2\pi} = 1$). This means that the function is periodic with period T = 1.

(c)
$$\sin^2(2t) = \frac{1}{2}\left[1-\cos(4t)\right] \rightarrow (1)$$
$$\sin^2 2\left(t+T\right) = \frac{1}{2}\left[1-\cos 4\left(t+T\right)\right] \rightarrow (2)$$

If the function is periodic then (1) and (2) above must be equal ∡t. This happens if,
$\cos\left[4(t+T)\right] = \cos(4t), \text{or if } 4T = 2\pi, \text{or} = T = \pi/2$

(d) Given f(t) = $\sin(2t^2)$, recall that a sine function is periodic with period $2K\pi$. If f(t) is periodic, then

$$\sin\left(2t^2 + 2K\pi\right) = \sin\left[2\left(t+T\right)^2\right]$$

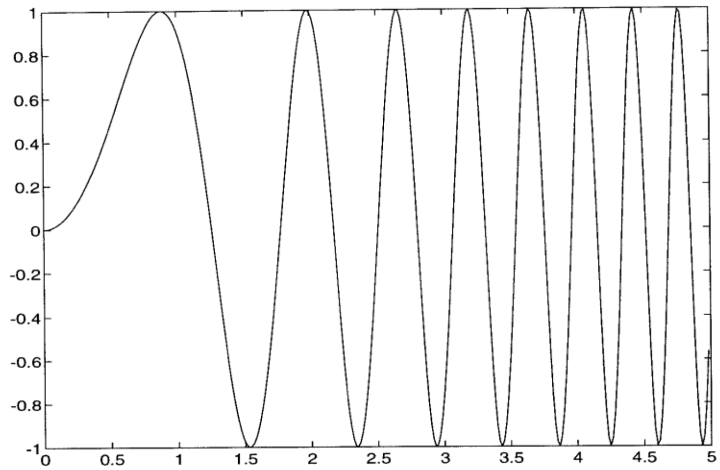


Figure 1: The function $f(t) = \sin(2t^2)$.

Where T is the period. This means that, $2t^2 + 2K\pi = 2(t+T)^2$. this simplifies to: $2T^2 + 4Tt = 2K\pi$. This means that f(t) is not periodic, since we cannot get a constant value for the period T from the above equation.

7. Find which of the following signals are energy signals and power signals.

$i) x(n) = \left(\frac{1}{3}\right)^n .u(n)$ (ii) $x(n) = e^{j\left(\frac{\pi}{2}n+\frac{\pi}{4}\right)}$ (iii) $x(n) = \sin\left(\frac{\pi}{4}n\right)$ (NOV/DEC 2014) (16)

Solution:-

(i) $x(n) = \left(\frac{1}{3}\right)^n .u(n)$

Energy signal: $\sum_{n=-\infty}^{\infty} |x(n)|^2 = \sum_{n=-\infty}^{\infty} \left|\left(\frac{1}{3}\right)^n\right|^2 = \sum_{n=-\infty} \left|\left(\frac{1}{3}\right)^2\right|^n$

$$= \sum_{n=-\infty}^{\infty} \left|\frac{1}{9}\right|^n$$

$$\therefore \sum_{n=0}^{\infty} (a)^n = \frac{1}{1-a}$$

Energy signal = $E = \frac{1}{1-\frac{1}{9}} = \frac{9}{8}$ Joules

To check for power, $P = \lim_{n\to\infty} \frac{1}{2N+1} \sum_{n=-N}^{N} |x(n)|^2$

$$= \lim_{n\to\infty} \frac{1}{2N+1} \sum_{n=0}^{N} \left|\frac{1}{9}\right|^{n}$$

$$= \lim_{n\to\infty} \frac{1}{2N+1} \sum_{n=0}^{N} \left[\frac{1-\left(\frac{1}{9}\right)^{N+1}}{1-\frac{1}{9}}\right] = 0$$

$$\therefore \sum_{n=0}^{N} (\alpha)^{n} = \left[\frac{1-a^{N+1}}{1-a}\right], a \neq 1$$

From the above result 'E' is finite and P = 0. Hence the above signal is energy signal.

(ii) $x(n) = e^{j\left(\frac{\pi}{2}n+\frac{\pi}{4}\right)}$

$$E = \sum_{n=-\infty}^{\infty} |x(n)|^{2}$$

$$\therefore \left|e^{j(\omega+\theta)}\right| = 1$$

$$E = \sum_{n=-\infty}^{\infty} \left|e^{j\left(\frac{\pi}{2}n+\frac{\pi}{4}\right)}\right|^{2} = \sum_{n=-\infty}^{\infty} |1|^{2} = \infty$$

To check for power, $P = \lim_{n\to\infty} \frac{1}{2N+1} \sum_{n=-N}^{N} |x(n)|^{2}$

$$P = \lim_{n\to\infty} \frac{1}{2N+1} \sum_{n=-N}^{N} \left|e^{j\left(\frac{\pi}{2}n+\frac{\pi}{4}\right)}\right|^{2}$$

$$= \lim_{n\to\infty} \frac{1}{2N+1} \sum_{n=-N}^{N} |1|^{2}$$

$$= \lim_{n\to\infty} \frac{1}{2N+1} 2N+1 = 1 \text{ Watt}$$

$$\therefore \sum_{n=-N}^{N} 1 - 2N+1$$

From the above results 'E' is ∞ and P = 1 wait. Hence the above signal is power signal.

(iii) $x(n) = \sin\left(\frac{\pi}{4}n\right)$

$$E = \sum_{n=-\infty}^{\infty} |x(n)|^{2} = \sum_{n=-\infty}^{\infty} \left|\sin\left(\frac{\pi}{4}n\right)\right|^{2} = \sum_{n=-\infty}^{\infty} \left|\sin^{2}\left(\frac{\pi}{4}n\right)\right|$$

$$= \sum_{n=-\infty}^{\infty} \left|\frac{\left(1-\cos 2\left(\frac{\pi}{4}n\right)\right)}{2}\right| = \sum_{n=-\infty}^{\infty} \left|\frac{\left(1-\cos\left(\frac{\pi}{2}n\right)\right)}{2}\right| = \infty$$

To check for power, $P = \lim_{n\to\infty} \frac{1}{2N+1} \sum_{n=-N}^{N} |x(n)|^2$

$$P = \lim_{n\to\infty} \frac{1}{2N+1} \sum_{n=-N}^{N} \left|\sin\left(\frac{\pi}{4}n\right)\right|^2 = \lim_{n\to\infty} \frac{1}{2N+1} \sum_{n=-N}^{N} \left|\sin^2\left(\frac{\pi}{4}n\right)\right|$$

$$P = \lim_{n\to\infty} \frac{1}{2N+1} \sum_{n=-N}^{N} \left|\left(\frac{1-\cos\left(\frac{\pi}{2}n\right)}{2}\right)\right| = \frac{1}{2}\lim_{n\to\infty} \frac{1}{2N+1} \sum_{n=-N}^{N} 1$$

From the above result 'E' is ∞ and P = 0.5 watt. Hence the above signal is power signal.

Unit – II

Discrete Time System Analysis

Part – A

1. Define DTFT

Let us consider the discrete time signal x(n). Its DTFT is denoted as X(ω). It is given as $X(\omega) = \sum_{n=-\infty}^{\infty} x(n)e^{-j\omega n}$

2. State the condition for existence of DTFT?

The conditions are, if x(n) is absolutely summable then $|x(n)| < \infty$. if x(n) is not absolutely summable then it should have finite energy for DTFT to exit.

3. List the properties of DTFT

Periodicity

Linearity

Time shift

Frequency shift

Scaling

Differentiation in frequency domain

Time reversal

Convolution

Multiplication in time domain

Parseval's theorem

4. What is the DTFT of unit sample?

The DTFT of unit sample is 1 for all values of ω.

5. Define Z transform (NOV/DEC 2013)

The Z transform of a discrete time signal x(n) is denoted by x(z) and is given by $X(z)=\sum_{n=-\infty}^{\infty} x(n)Z^{-n}$

6. Define ROC and its properties (MAY/JUNE 2014, NOV/DEC 2013, NOV/DEC 2012)

The values of z for which z – transform converges is called region of convergence (ROC). The z – transform is an infinite power series; hence it is necessary to mention the ROC along with Z – transform.

ROC of an LTI stable system contains unit circle

If x(n) is casual, then ROC is the entire unit circle.

7. Find Z transform of x(n) = {1,2,3,4}

Given x(n) = {1, 2, 3, 4}, $X(z)=\sum_{n=-\infty}^{\infty} x(n)Z^{-n} = 1+2Z^{-1}+3Z^{-2}+4z^{-3}$

8. State the convolution property of Z transform (NOV/DEC 2013)

The convolution property states that the convolution of two sequences in time domain is equivalent to multiplication of the sequences in Z domain.

If $Z\left[x_1(n)\right]=X_2(z)$ and $Z\left[x_2(n)\right]=X_2(z)$ then $Z\left[x_1(n)*x_2(n)\right]=X_2(z)X_2(z)$

9. What Z transform of (n-m)?

By time shifting property $Z\left[A\left(n-m\right)\right]=AZ^{-m}\sin Z\left[(n)\right]=1$

10. State final and initial value theorem (MAY/JUNE 2014)

Let x(n) be an one – sided signal defined in the range $0 \le n \le \infty$

If Z[x(n)] = X(z) then the initial value of x(n) is given by $x(0)=\lim_{z\to\infty} X(z)$

If Z[x(n)] = X(z), then the final value of x(n) is given by $x(\infty)=\lim_{z\to 1}\left(1-z^{-1}\right)X(z)$

11. List the methods of obtaining inverse Z transform.

Inverse Z transform can be obtained by using

Partial fraction expansion

Contour integration

Power series expansion

Convolution

12. Obtain the inverse Z transform of $X(z)=\frac{1}{z-a},|z|>|a|$

Given $X(z)=\frac{Z^{-1}}{\left(1-aZ^{-1}\right)}$ by time shifting property $x(n)=a^{n}u\left(n-1\right)$

13. List any four properties of Z – transform.

Linearity

Time shifting

Frequency shift or Frequency translation

Time reversal

14. What are the different methods of evaluating inverse Z – transform?

Partial fraction expansion

Power series expansion

Contour integration (Residue method)

15. What are the properties of convolution?

Commutative property $x(n)*h(n)=h(n)*x(n)$

Associative property $\left[x(n)*h_1(n)\right]*h_2(n)=x(n)*\left[h_1(n)*h_2(n)\right]$

Distributed property $x(n)*\left[h_1(n)+h_2(n)\right]=\left[x(n)*h_1(n)\right]+\left[x(n)*h_2(n)\right]\rightarrow$

16. Define discrete Fourier series representation for a periodic sequence (NOV/DEC 2014)

For x(n) is a periodic signal with period N, that is x[n+N]=x(n) as is continues time period signal we would like to represent x(n) in terms of discrete time complex exponential with period N, these signals are given by

$$e^{\frac{j2\pi nk}{N}}$$

17. Determine the Z – transform and ROC for the signal $x(n)=\delta(n-k)+\delta(n+k)$ (NOV/DEC2013, MAY/JUNE 2012)

Given $x(n)=\delta(n-k)+\delta(n+k)$

$$Z\left[\delta(n)\right]=1;Z\left[\delta(n-k)\right]=Z^{-k}X(z);Z\left[\delta(n+k)\right]=Z^{k}X(z)$$
$$Z\left[x(n)\right]=Z^{-k}X(z)+Z^{k}X(z)=\left(Z^{-k}+Z^{k}\right)X(z)$$

18. Given a difference equation y(n) = x(n) +3x(n – 1) +2y(n – 1). Determine the system function H(z)

(MAY/JUNE 2013)

Given y(n) = x(n) +3x(n – 1) +2y(n – 1). Taking Z transform on both sides

$$Y(z) = X(z) + 3z^{-1}X(z) + 2z^{-1}Y(z) \text{ and } Y(z)\left\{1-2z^{-1}\right\} = X(z)\left[1+3z^{-1}\right]$$

$$H(z) = \frac{Y(z)}{X(z)} = \frac{1+3z^{-1}}{1-2z^{-1}}$$

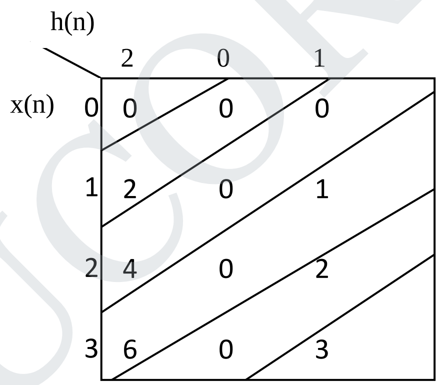
19. Find the stability of the system whose impulse response $h(n) = \left(\frac{1}{2}\right)^n u(n)$ (MAY/JUNE 2013)

Given $h(n) = \left(\frac{1}{2}\right)^n u(n)$ the condition for stability is $\sum_{n=-\infty}^{\infty} |h(n)| < \infty$

Therefore $\sum_{n=-\infty}^{\infty} |h(n)| = \sum_{n=-\infty}^{\infty} \left|\left(\frac{1}{2}\right)^n u(n)\right| = \sum_{n=0}^{\infty} \left(\frac{1}{2}\right)^n = 2$, hence stable.

20. Find the convolution for x(n) = {0, 1, 2, 3} and h(n) = {2, 0, 1} (MAY/JUNE 2012)

Solution:- y(n)= {0, 2, 4, 7, 2, 3}



21. Determine the Z – transform of $x(n) = a^n$ (APR/MAY 2015)

Let us assume given $x(n) = a^n$ is $x(n) = a^n u(n)$

Therefore

$$X(z) = \sum_{n=-\infty}^{\infty} x(n)Z^{-n} = X(z) = \sum_{n=-\infty}^{\infty} a^n u(n) Z^{-n}$$

$$= \sum_{n=0}^{\infty} a^n Z^{-n} = \frac{1}{1-A} = \frac{1}{1-az^{-1}}; \text{ROC}: |z| > a$$

PART B

1. Determine the z –transform and ROC of the signal

x(n)=[3(2^n) – 4(3^n)]u(n) (Nov/Dec 2014) (10)

Solution:- If we define the signals

$$x_1(n) = 2^n u(n)$$

And

$$x_2(n) = 3^n u(n)$$

Then x(n) can be written as

$$x(n) = 3x_1(n) - 4x_2(n)$$

$$X(z) = 3X_1(z) - 4x_2(z)$$

$$a^n u(n) \xleftrightarrow{z} \frac{1}{1-az^{-1}} \text{ROC}|z| > |a| \quad \rightarrow (1)$$

By setting a = 2 and a = 3 in equation 1, we obtain

$$x_1(n) = 2^n u(n) \xleftrightarrow{z} X_1(z) = \frac{1}{1-2z^{-1}} \quad \text{ROC}: |z| > 2$$
$$x_2(n) = 3^n u(n) \xleftrightarrow{z} X_2(z) = \frac{1}{1-3z^{-1}} \quad \text{ROC}: |z| > 3$$

The intersection of the ROC of X1(z) and X2(z) is $|z| > 3$. Thus the overall transform X(z) is

$$X(z) = \frac{3}{1-2z^{-1}} - \frac{4}{1-3z^{-1}} \quad \text{ROC}: |z| > 3$$

2. Determine the Z – transform of

a. $x(n) = a^n \cos\omega_0 n\, u(n)$ b. $x(n) = a^n \sin\omega_0 n\ u(n)$ **(NOV/Dec 2013) (16)**

Solution:-

a. $x(n) = a^n \cos\omega_0 n\ u(n)$

Let $x_1(n) = \cos(\omega_0 n)\ u(n)$

The Z – transform of $\cos(\omega_0 n)u(n)$ is given by

$$x_1(z) = \frac{1 - z^{-1}\cos\omega_0}{1 - 2z^{-1}\cos\omega_0 + z^{-2}}, \quad \text{ROC}: |z| > 1 \qquad \rightarrow (1)$$

Since $\cos(\omega_0 n)u(n) \overset{z}{\longleftrightarrow} \dfrac{1 - z^{-1}}{2z^{-1}\cos\omega_0 + z^{-2}}, \quad \text{ROC}: |z| > 1 \qquad \rightarrow (2)$

Now consider the given equation i.e,

$$\begin{aligned} x(n) &= a^n \cos(\omega_0 n)u(n) \\ &= a^n x_1(n) \\ x(z) &= Z\{a^n x_1(n)\} \qquad \rightarrow (3) \end{aligned}$$

Here let us use the scaling property of equation i.e,

$$Z\{a^n x(n)\} = x\left(\frac{z}{a}\right), \quad \text{ROC}: |\alpha| r_1 < |z| < |\alpha| r_2$$

Applying this property to equation (3) we have,

$$X(z) = X_1\left(\frac{z}{a}\right)$$

$X_1\left(\frac{z}{a}\right)$ is obtained by replacing z by $\frac{z}{a}$ in $X_1(z)$ of equation (2) i.e,

$$X(z) = \frac{1 - \left(\frac{z}{a}\right)^{-1}\cos\omega_0}{1 - 2\left(\frac{z}{a}\right)^{-1}\cos\omega_0 + \left(\frac{z}{a}\right)^{-2}} \quad \text{ROC}: |z| > 1|\alpha|$$

Thus we obtained the z – transform pair as,

$$a^n \cos(\omega_0 n)u(n) \overset{z}{\longleftrightarrow} \frac{1 - \left(\frac{z}{a}\right)^{-1}\cos\omega_0}{1 - 2\left(\frac{z}{a}\right)\cos\omega_0 + \left(\frac{z}{a}\right)^{-2}} \quad \text{ROC}: |z| > |\alpha| \qquad \rightarrow (4)$$

b. $x(n) = a^n \sin\omega_0 n \; u(n)$

Let $x_1(n) = \sin(\omega_0 n)u(n)$

The Z – transform of $\sin(\omega_0 n)u(n)$ is given by equation (2) i.e,

$$x_1(z) = \frac{z^{-1}\sin\omega_0}{1-2z^{-1}\cos\omega_0 + z^{-2}}, \quad \text{ROC}: |z| > 1 \qquad \rightarrow (5)$$

Since $\sin(\omega_0 n)u(n) \xleftrightarrow{z} \dfrac{z^{-1}\sin\omega_0}{1-2z^{-1}\cos\omega_0 + z^{-2}}, \quad \text{ROC}: |z| > 1 \qquad \rightarrow (6)$

Now consider the given equation i.e,

$$\begin{aligned} x(n) &= a^n \sin(\omega_0 n)u(n) \\ &= a^n x_1(n) \\ x(z) &= z\{a^n x_1(n)\} \qquad \rightarrow (7) \end{aligned}$$

Here let us use the scaling property of equation i.e,

$$Z\{a^n x(n)\} = X\left(\frac{z}{a}\right), \ \text{ROC}: |\alpha| r_1 < |z| < |\alpha| r_2$$

Applying this property to equation (7) we have

$$x(z) = X_1\left(\frac{z}{a}\right)$$

Here $x_1\left(\frac{z}{a}\right)$ is obtained by replacing z by X1 (z) of equation (5) i.e.,

$$x(z) = \frac{\left(\frac{z}{a}\right)^{-1}\sin\omega_0}{1-2\left(\frac{z}{a}\right)\cos\omega_0 + \left(\frac{z}{a}\right)^{-2}}, \ \text{ROC}: |z| > 1|\alpha|$$

Thus we have obtained the Z – transform pair as,

`

$$a^n \sin(\omega_0 n)u(n) \xleftrightarrow{z} \frac{\left(\frac{z}{a}\right)^{-1}\sin\omega_0}{1-2\left(\frac{z}{a}\right)^{-1}\cos\omega_0 + \left(\frac{z}{a}\right)^{-2}}, \ \text{ROC}: |z| > |\alpha| \qquad \rightarrow (8)$$

3. Determine Z transform for $x(n) = n\, a^n\, u(n)$ (8)

Solution:- Let $x_1(n) = a^n u(n)$

From equation the z- transform of a^n u(n) is given as

$$Z\left[a^n u(n)\right] = \frac{1}{1-az^{-1}}, \quad ROC: |z| > |\alpha|$$
$$X_1(z) = \frac{1}{1-az^{-1}}, \quad ROC: |z| > |\alpha|$$

Now the given function is,

$$x(n) = na^n u(n)$$
$$x(n) = nx_1(n) \qquad \text{since } x_1(n) = a^n u(n)$$
$$x(z) = Z\left[nx_1(n)\right] \qquad \rightarrow (9)$$

The differentiation in z – domain property of equation (2) states that,

$$Z\left[n\ x(n)\right] = -z\frac{d}{dz}x(z)$$

Applying this property to equation (9) we have

$$X(z) = -z\frac{d}{dz}X_1(z), \ ROC: |z| > |\alpha|$$
$$= -z\frac{d}{dz}\frac{1}{1-az^{-1}}$$
$$= \frac{az^{-1}}{\left(1-az^{-1}\right)^2}$$

Thus we have obtained the z – transform pair,

$$n\ a^n u(n) \xleftrightarrow{z} \frac{az^{-1}}{\left(1-az^{-1}\right)^2}, \ ROC: |z| > |\alpha| \qquad \rightarrow (10)$$

Z – transform of unit ramp sequence:

The unit ramp sequence is given as,

$$x(n) = n\ u(n)$$
$$X(z) = Z[n\ u(n)]$$

By putting a = 1 in equation (10) we can get the z – transform of unit ramp sequence.

Hence,

$$n\ u(n) \xleftrightarrow{z} \frac{z^{-1}}{\left(1-z^{-1}\right)},\ \ \text{ROC}: |z| > 1 \qquad \rightarrow (11)$$

4. Determine the z – transform of the signal

(a) $x_1(n) = \cos(\omega_0 n) u(n)$

Solution:- Using Eulers identify we have,

$$\cos\theta = \frac{e^{j\theta} + e^{-j\theta}}{2}$$

Hence

$$\cos(\omega_0 n) = \frac{e^{j\theta} + e^{-j\theta}}{2}$$
$$x(n) = \frac{1}{2}\left[e^{j\omega_0 n + e^{j\omega_0 n}}\right]u(n)$$
$$= \frac{1}{2}e^{j\omega_0 n}u(n) + \frac{1}{2}e^{-j\omega_0 n}u(n)$$

Taking Z – transform of above function,

$$X(z) = Z\left\{\frac{1}{2}e^{j\omega_0 n}u(n) + \frac{1}{2}e^{-j\omega_0 n}u(n)\right\}$$

Using the linearity property of equation (1) we can write above equation as,

$$X(z) = Z\left\{\frac{1}{2}e^{j\omega_0 n}u(n)\right\} + Z\left\{\frac{1}{2}e^{-j\omega_0 n}u(n)\right\}$$

Here let $a = e^{j\omega_0}$ & $b = e^{-j\omega_0}$, then we can write,

$$X(z) = \frac{1}{2}Z\left[a^n u(n)\right] + \frac{1}{2}Z\left[b^n u(n)\right] \qquad \rightarrow (1)$$

From equation we can write,

$$Z\left[a^n u(n)\right] = \frac{1}{1-az^{-1}} \quad ROC: |z| > |a| \qquad (2)$$

We have $\quad a = e^{j\omega_0} = \cos\omega_0 + j\sin\omega_0$

$$|a| = \sqrt{\cos^2\omega_0 + \sin^2\omega_0} = 1$$

Hence we can write equation (2) as,

$$Z\left[a^n u(n)\right] = \frac{1}{1-e^{j\omega_0}z^{-1}}, \quad ROC: |z| > 1 \qquad \to (3)$$

Similarly the z – transform of second term in equation (1) can be written as,

$$Z\left[b^n u(n)\right] = \frac{1}{1-bz^{-1}}, \quad ROC: |z| > |b| \text{ from equation} \qquad \to (4)$$

We have $\quad b = e^{-j\omega_0} = \cos\omega_0 - j\sin\omega_0$

$$|b| = \sqrt{\cos^2\omega_0 + \sin^2\omega_0} = 1$$

Hence we can write equation (4) as,

$$Z\left[b^n u(n)\right] = \frac{1}{1-e^{-j\omega_0}z^{-1}}, \quad ROC: |z| > 1 \qquad \to (5)$$

Putting the individual z – transform values from above equation and equation (3) in equation (1) we get,

$$X(z) = \frac{1}{2}\cdot\frac{1}{1-e^{j\omega_0}z^{-1}} + \frac{1}{2}\frac{1}{1-e^{-j\omega_0}z^{-1}}, \quad ROC: |z| > 1$$

$$= \frac{1}{2}\left\{\frac{1}{1-e^{j\omega_0}z^{-1}} + \frac{1}{1-e^{-j\omega_0}z^{-1}}\right\}$$

$$= \frac{1}{2}\left\{\frac{1-e^{-j\omega_0}z^{-1} + 1-e^{j\omega_0}z^{-1}}{1-e^{-j\omega_0}z^{-1} - e^{j\omega_0}z^{-1} + e^{j\omega_0}z^{-1}e^{-j\omega_0}z^{-1}}\right\}$$

$$= \frac{1}{2}\left\{\frac{2-z^{-1}\left(e^{j\omega_0} + e^{-j\omega_0}\right)}{1-z^{-1}\left(e^{j\omega_0} + e^{-j\omega_0}\right) + z^{-2}}\right\}$$

Here let us use Euler's identity i.e $2\cos\theta = e^{j\theta} + e^{-j\theta}$. Hence above equation can be further simplified as,

$$X(z) = \frac{1}{2}\left\{\frac{2-2z^{-1}\cos\omega_0}{1-2z^{-1}\cos\omega_0 + z^{-2}}\right\}$$

$$X(z) = \frac{1-z^{-1}\cos\omega_0}{1-2z^{-1}\cos\omega_0 + z^{-2}}, \quad ROC: |z| > 1$$

Thus we obtained the standard z – transform pair as,

$$\cos(\omega_0 n)u(n) \xleftrightarrow{z} \frac{1-z^{-1}\cos\omega_0}{1-2z^{-1}\cos\omega_0 + z^{-2}}, \ \text{ROC}: |z| > 1 \qquad \rightarrow (6)$$

(b) $x_2(n) = \sin(\omega_0 n).u(n)$

Solution:- Using Euler's identity we have

$$\sin\theta = \frac{e^{j\theta} - e^{-j\theta}}{2j}$$

$$\sin(\omega_0 n) = \frac{1}{2j}\left[e^{j\omega_0 n} - e^{-j\omega_0 n}\right]$$

$$x(n) = \frac{1}{2j}\left[e^{j\omega_0 n} - e^{-j\omega_0 n}\right]u(n)$$

$$= \frac{1}{2j}e^{j\omega_0 n}u(n) - \frac{1}{2}e^{-j\omega_0 n}u(n)$$

Taking Z – transform of above function,

$$X(z) = Z\left\{\frac{1}{2j}e^{j\omega_0 n}u(n) - \frac{1}{2j}e^{-j\omega_0 n}u(n)\right\}$$

$$= \frac{1}{2j}Z\left\{e^{j\omega_0 n}u(n)\right\} - \frac{1}{2j}Z\left\{e^{-j\omega_0 n}u(n)\right\} \ \text{By linearity property} \qquad \rightarrow (7)$$

In the previous example we have obtained z – transform of $e^{j\omega_0 n}u(n)$, i.e,

$$Z\left\{e^{j\omega_0 n}u(n)\right\} = \frac{1}{1-e^{j\omega_0 n}z^{-1}}, \ \text{ROC}: |z| > 1 \ \text{By equation (3)}$$

Similarly Z –transform of $e^{-j\omega_0 n}u(n)$ also we have obtained in pervious example i.e,

$$Z\left\{e^{-j\omega_0 n}u(n)\right\} = \frac{1}{1-e^{-j\omega_0 n}z^{-1}}, \ \text{ROC}: |z| > 1 \ \text{By equation (5)}$$

Putting the above two results in equation (7) we get,

$$x(z)=\frac{1}{2j}.\frac{1}{1-e^{j\omega_0 n}z^{-1}}-\frac{1}{2j}.\frac{1}{1-e^{-j\omega_0}z^{-1}},\quad ROC:|z|>1$$

$$=\frac{1}{2j}\left\{\frac{1}{1-e^{j\omega_0 n}z^{-1}}-\frac{1}{1-e^{-j\omega_0}z^{-1}}\right\}$$

$$=\frac{1}{2j}\left\{\frac{1-e^{-j\omega_0 n}z^{-1}-1e^{j\omega_0 n}z^{-1}}{1-e^{-j\omega_0 n}z^{-1}-e^{j\omega_0 n}z^{-1}+e^{-j\omega_0 n}z^{-1}.e^{j\omega_0 n}z^{-1}}\right\}$$

$$=\frac{1}{2j}\left[\frac{z^{-1}\left(e^{j\omega_0 n}-e^{-j\omega_0 n}\right)}{1-z^{-1}\left(e^{j\omega_0 n}+e^{-j\omega_0 n}\right)+z^{-2}}\right]$$

$$=\left\{\frac{z^{-1}\dfrac{e^{j\omega_0 n}-e^{-j\omega_0 n}}{2j}}{1-z^{-1}\left(e^{j\omega_0 n}-e^{-j\omega_0 n}\right)+z^{-2}}\right\}$$

Applying the Euler's identity to above equation we can write,

$$X(z)=\frac{z^{-1}\sin\omega_0}{1-2z^{-1}\cos\omega_0+z^{-2}},\quad ROC:|z|>1$$

Thus we have obtained the z – transform pair as,

$$\sin(\omega_0 n)u(n)\xleftrightarrow{z}\frac{z^{-1}\sin\omega_0}{1-2z^{-1}\cos\omega_0+z^{-2}},\quad ROC:|z|>1 \qquad \rightarrow(8)$$

5. Determine the inverse Z – transform of $x(z)=\dfrac{1}{1-1.5z^{-1}+0.5z^{-2}}$ if (16)

(i)$ROC:|z|>1$ (ii)$ROC:|z|<0.5$ (iii)$ROC:0.5<|z|<1$

Solution:- This is the same problem that we treated in example. The partial fraction expansion for X(z) was determined in example. The partial – fraction expression of X(z) yields.

$$x(z)=\frac{2}{1-z^{-1}}-\frac{1}{1-0.5z^{-1}} \qquad \rightarrow(1)$$

To invert X(z) we should apply for p1 = 1 and p2 = 0.5. however, this requires the specification of the corresponding ROC.

(a) In case when the ROC is $|z|>1$, the signal x(n) is casual and both terms in (1) are casual terms. According to, we obtain

$$x(n)=2(1)^n u(n)-(0.5)^n u(n)=(2-0.5^n)u(n) \qquad \rightarrow(2)$$

Which agrees with the results in example

(b) When the ROC is $|z|<0.5$, the signal x(n) is anticasual. Thus both terms in (1) result in anticasual components from we obtain

$$x(n)=\left[-2+(0.5)^n\right]u(-n-1) \qquad \rightarrow(3)$$

(c)In this case the ROC $0.5<|z|<1$ is a ring, which implies that the signal x(n) is two – sided. Thus one of the terms corresponds to a casual signal the other to an anticasual signal. obviously, the given ROC is the overlapping of the regions $|z|>0.5$ & $|z|<1$. Hence the pole p2 = 0.5 provides the casual part and the pole p1 = 1 the anticasual. Thus

$$x(n)=-2(1)^n u(-n-1)-(0.5)^n u(n) \qquad \rightarrow (4)$$

6. Determine the unit step response of the system described by the difference equation $y(n)=0.9y(n-1)-0.81y(n-2)+x(n)$ under the following initial conditions:

(a) $y(-1)=y(-2)=0$

(b) $y(-1)-y(-2)=1$

Solution:- The system function is

$$H(z)=\frac{1}{1-0.9z^{-1}+0.81z^{-2}}$$

This system has two complex – conjugate poles at

$$p_1=0.9e^{j\pi/3} \qquad p_2=0.9e^{-j\pi/3}$$

The z – transform of the unit step sequence is

$$X(z)=\frac{1}{1-z^{-1}}$$

Therefore

$$\begin{aligned} Y_{zs}(z) &= \frac{1}{\left(1-0.9e^{j\pi/3}z^{-1}\right)\left(1-0.9e^{-j\pi/3}z^{-1}\right)\left(1-z^{-1}\right)} \\ &= \frac{0.542-j0.049}{1-0.9e^{j\pi/3}z^{-1}}+\frac{0.542+j0.049}{1-0.9e^{-j\pi/3}z^{-1}}+\frac{1.099}{1-z^{-1}} \end{aligned}$$

And hence the zero – state response is

$$Y_{zs}(n)=\left[1.099+1.088(0.9)^n\cos\left(\frac{\pi}{3}n-5.2\right)\right]u(n)$$

(a) Since the initial condition are zero in this case, we conclude that y(n) = y_{zs}(n).

(b) For the initial condition y(-1) = y(-2) = 1, the additional component in the z – transform is

$$\begin{aligned} Y_{zs}(z) &= \frac{N_0(z)}{A(z)}=\frac{0.09-0.81z^{-1}}{1-0.9z^{-1}+0.81z^{-2}} \\ &= \frac{0.026+j0.4936}{1-0.9e^{j\pi/3}z^{-1}}+\frac{0.026-j0.4936}{1-0.9e^{-j\pi/3}z^{-1}} \end{aligned}$$

Consequently, the zero – input response is

$$Y_{zs}(n)=0.988(0.9)^n\cos\left(\frac{\pi}{3}n+87^\circ\right)u(n)$$

In this case the total response has the z- transform

$$\begin{aligned} Y(z) &= Y_{zs}(z) + Y_{zi}(z) \\ &= \frac{1.099}{1-z^{-1}} + \frac{0.568 + j0.445}{1-0.9e^{j\pi/3}z^{-1}} + \frac{0.568 - j0.445}{1-0.9e^{-j\pi/3}z^{-1}} \end{aligned}$$

The inverse transform yields the total response in the form

$$y(n) = 1.099u(n) + 1.44(0.9)^n \cos\left(\frac{\pi}{3}n + 38\right)u(n)$$

7. Determine the response $y(n), n \geq 0,$ of the system described by the second order difference equation.

$$y(n) - 3y(n-1) - 4y(n-2) = x(n) + 2x(n-1)$$

When the input sequence is

$$x(n) = 4^n u(n)$$

Solution:- We have already determined the solution to the homogeneous difference equation for this system in example for we have

$$y_n(n) = C_1(-1)^n + C_2(4)^n \qquad \rightarrow (1)$$

The particular solution to is assumed to be an exponential sequence of the same form as x(n). normally, we could assume a solution of the form

$$y_p(n) = K(4)^n u(n)$$

However, we observe that $y_p(n)$ is already contained in the homogeneous solution, so that this particular solution is redundant. Instead, we select the particular solution to be linearity independent of the terms contained in the homogenous solution. In fact, we treat this situation in the same manner as we have already treated multiple roots in the characteristics equation. Thus we assume that

$$y_p(n) = Kn(4)^n u(n) \qquad \rightarrow (2)$$

Upon substitution of (2) into we obtain

$$\begin{aligned} Kn(4)^n u(n) - 3K(n-1)(4)^{n-1}u(n-1) - 4K(n-2)(4)^{n-2}u(n-2) \\ = (4)^n u(n) + 2(4)^{n-1}u(n-1) \end{aligned}$$

To determine K, we evaluate this equation for any $n \geq 2$, where none of the unit step terms vanish. To simplify the arithmetic, we select n = 2, from which we obtain $K = \frac{6}{5}$. Therefore

$$y_p(n) = \frac{6}{5}n(4)^n u(n) \qquad \rightarrow (3)$$

$$\begin{aligned} y(1) &= 3y(0) + 4y(-1) + 6 \\ &= 13y(-1) + 12y(-2) + 9 \end{aligned}$$

The total solution to the difference equation is obtained by adding (1) to (3). Thus

$$y(n) = C_1(-1)^n + C_2(4)^n + \frac{6}{5}n(4)^n \quad n \geq 0 \qquad \rightarrow (4)$$

Where the constants C1 and C2 are determined such that the initial conditions are satisfied. To accomplish this, we return to from which we obtain

$$y(0) = 3y(-1) + 4y(-2) + 1$$

On the other hand, (4) evaluated at n = 0 and n = 1 yields

$$\begin{aligned} y(0) &= C_1 + C_2 \\ y(1) &= -C_1 + 4C_2 + \frac{24}{5} \end{aligned}$$

We can now equate these two sets of relations to obtain C1 and C2. In so doing, we have the response due to initial conditions y(-1) and y(-2) (the zero- input response). And the zero stare or forced response.

Since we have already solved for the zero – input response in example we can simplify the computations above by setting y(-1) = y(-2)= 0. Then we have

$$\begin{aligned} C_1 + C_2 &= 1 \\ -C_1 + 4C_2 + \frac{24}{5} &= 9 \end{aligned}$$

Hence $C_1 = \frac{1}{25}$ & $C_2 = \frac{26}{25}$ finally, we have the zero – state response to the forcing function $x(n) = (4)^n u(n)$ in the form

$$y_{zs}(n) = -\frac{1}{25}(-1)^n + \frac{26}{25}(4)^n + \frac{6}{5}n(4)^n \quad n \geq 0 \qquad \rightarrow (5)$$

The total response of the system, which includes the response to arbitrary initial conditions is the sum of equation.

8. The total response of the system, which includes the response to arbitrary initial conditions is the sum of equation.

Using z-transform determine the response $y(n)$ for $n \geq 0$ if (May'13, Nov'16) (16)

$$y(n)=\frac{1}{2}y(n-1)+x(n) \quad x(n)=\left(\frac{1}{3}\right)^n u(n), y(-1)=1$$

Solution:

$$Y^+(z)-0.5\left[z^{-1}Y^+(z)+1\right]=\frac{1}{1-\frac{1}{3}z^{-1}}$$

$$Y^+(z)=\frac{1.5-\frac{1}{6}z^{-1}}{\left(1-\frac{1}{3}z^{-1}\right)\left(1-0.5z^{-1}\right)}$$

$$=\frac{\frac{7}{2}}{1-0.5z^{-1}}-\frac{2}{1-\frac{1}{3}z^{-1}}$$

$$\text{hence } y(n)=\left[\frac{7}{2}(0.5)^n-2\left(\frac{1}{3}\right)^n\right]u(n)$$

9. Find the Z-transform of $x(n)=n^2u(n)$ (Apr/May 2015) (8)

$$X(z)=\sum_{n=0}^{\infty}n^2z^{-n}$$

$$=z^2\frac{d^2}{dz^2}\sum_{n=0}^{\infty}z^{-n}$$

$$=z^2\frac{d^2}{dz^2}\left[\frac{1}{1-z^{-1}}\right]$$

$$=-\frac{z^{-1}}{\left(1-z^{-1}\right)^2}+\frac{2z^{-1}}{\left(1-z^{-1}\right)^3}$$

$$=\frac{z^{-1}\left(1+z^{-1}\right)}{\left(1-z^{-1}\right)^3},|z|>1$$

10. Determine the step response of the system $y(n) = ay(n-1) + x(n)$ -1<α<1, when the initial condition is $y(-1) = 1$ (May/June 2012) (8)

Solution By taking the one side z tramsform of both sides, we obtained

$$Y^+(z) = \alpha\left[z^{-1}Y^+(z) + y(-1)\right] + X^+(z)$$

Upon substrution for y(-1) and X*(z) anfd solving for $Y^+(z)$. we can obtaine the result.

$$Y^+(z) = \frac{\alpha}{1-\alpha z^{-1}} + \frac{1}{(1-\alpha z^{-1})(1-z^{-1})}$$

By performing a partial fraction expansion and inverse transforming the result, we have

$$y(n) = \alpha^{n+1}u(n) + \frac{1-\alpha^{n+1}}{1-\alpha}u(n)$$
$$= \frac{1}{1-\alpha}\left(1-\alpha^{n+2}\right)u(n)$$

11. Find the z-transform of the following signals and plot its ROC.

(a) $x_1(n) = a^n u(n)$ (b) $x_2(n) = -a^n u(-n-1)$ (Nov/Dec 2014) (16)

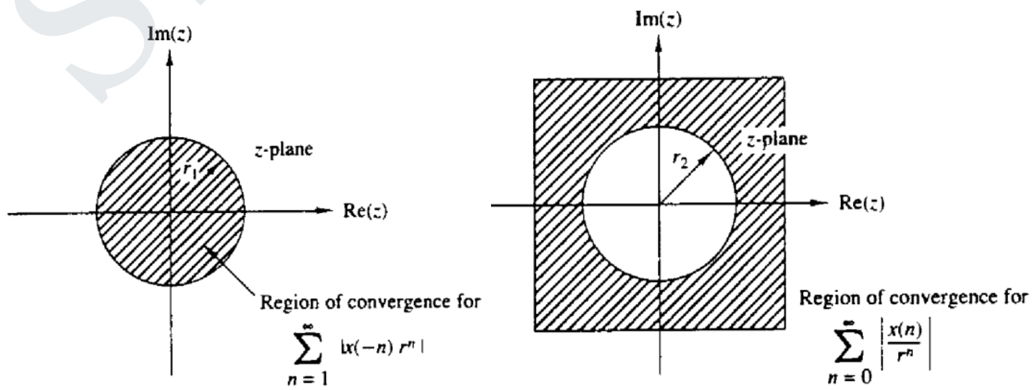
(a) Determine the z-transform of the signal

$$x(n) = a^n u(n) = \begin{cases} a^n & , n \geq 0 \\ 0 & , n < 0 \end{cases}$$

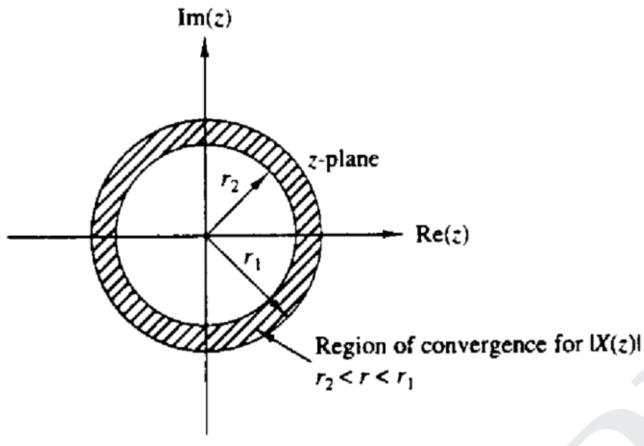
From the definition we have

$$X(z) = \sum_{n=0}^{\infty} a^n z^{-n} = \sum_{n=0}^{\infty}(az^{-1})^n$$

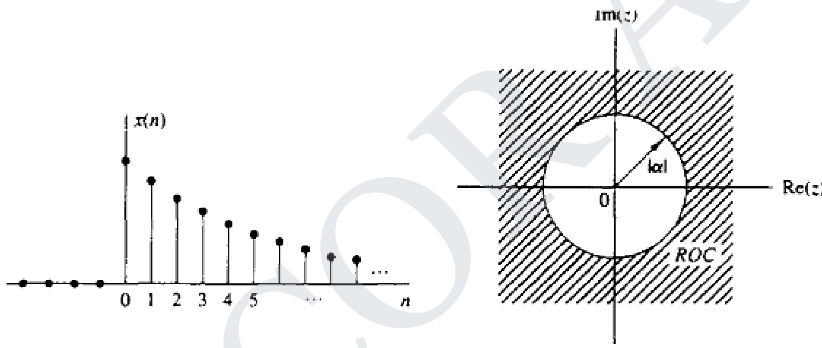
If $|az^{-1}| < 1$ or equivalently, $|z| > |a|$, this power serious converges to $1/(1-az^{-1})$



(a) (b)



Region of convergence for X(z) and its corresponding causal and anticausal components



(a) (b)

The exponential signal x(n) = a^nu(n) and its ROC

Thus we have thr z-transform pair

$$x(n) = a^n u(n) \xleftrightarrow{z} X(z) = \frac{1}{1-az^{-1}} \qquad \text{ROC } |z| = |a|$$

The ROC is the exterior of a circle having radius $|a|$. Graph if the signal x(n) and its corresponding ROC. Note that, in general a need not be real. If we send $a = 1$. we obtain the z-transform of the unit step signal.

$$x(n) = u(n) \xleftrightarrow{z} X(z) = \frac{1}{1-z^{-1}} \qquad \text{ROC } |z| = |1|$$

b) Determine the z-transform of the signal

$$x(n) = -a^n u(-n-1) = \begin{cases} 0 & n \ge 0 \\ -a^n & n \le -1 \end{cases}$$

Solution From the definition

$$X(z)=\sum_{n=-\infty}^{-1}\left(-a^{n}\right)z^{-n}=-\sum_{t=1}^{\infty}\left(za^{-1}\right)^{t}$$

When i= -n. using the formula

$$A+A^{2}+A^{3}+\ldots..=A\left(1+A+A^{2}+\ldots\right)=\frac{A}{1-A}$$

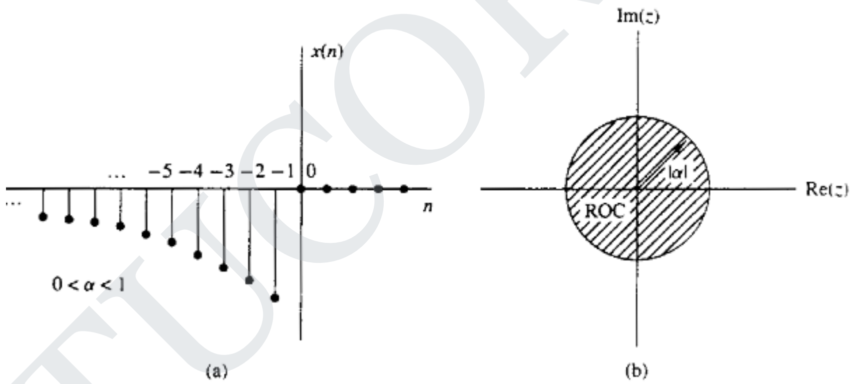
When $|A|<1$ gives

$$X(z)=\frac{a^{-1}z}{1-a^{-1}z}=\frac{1}{1-az^{-1}}$$

Provided that $|a^{-1}z|<1$ or equivalantly $|z|<|a|$ thus

$$x(n)=-a^{n}u(-n-1)\xleftrightarrow{z}X(z)=-\frac{1}{1-az^{-1}}\quad ROC|z|<|a|$$

The ROC is now the interior of a circle having radius $|a|$. This is show in fig.



The anticausal signal $x(n)=-a^{n}u(-n-1)$ **and its ROC**

Illustrate two very important issues. The first concern the uniqueness of the z transform. We see that the causual signal $\alpha^{n}u(n)$ and the anticasual signal $-a^{n}u(-n-1)$ have identical closed form expression for the z transform, that is $Z\left[a^{n}u(n)\right]=Z\{-au(-n-1)\}=\frac{1}{1-az^{-1}}$

This implies that a closed form expression for the z transform does not uniquely specify the signal in the time domain. The ambiguity can be resolved only if in addition to the closed form expression, the ROC is specified. In summery a discrete time signal x(n) is uniquely region of convergence of X(z). in the text the term z transform is used to refer to both the closed form expression and corresponding ROC. The point that ROC of causal signal is the exterior of a circle of some radius $r_2$while the ROC of an ant causal signal is the interior of a circle of some radius r_1. the following example consider a sequence that is nonzero for $\infty<n<\infty$.

12. State and prove the following properties of z-transform (16)

(i) Time shifting (ii) scaling in z domain (iii) Differentiation

Solution:

(i) Time shifting property

$$x(n) \xleftrightarrow{z} X(z) \quad \text{ROC}$$

$$x(n-k) \xleftrightarrow{z} z^{-k}X(z) \quad \text{atleast ROC expect } z=0\ (k>0) \text{ or } z=\infty(k<0)$$

Proof:

Let y(n)=x(n-k)

$$Y(z)=\sum_{n=-\infty}^{\infty} y(n)z^{-n}$$

$$=\sum_{n=-\infty}^{\infty} x(n-k)z^{-n}$$

$$\text{let} m=n-k$$

$$=\sum_{n=-\infty}^{\infty} x(m)z^{-(m+k)}=\sum_{n=-\infty}^{\infty} x(m)z^{-m}z^{-k}$$

$$=z^{-k}\left[\sum_{n=-\infty}^{\infty} x(m)z^{-m}\right]=z^{-k}X(z)$$

Time shifting Property ROC

$$x(n)=\delta(n) \xleftrightarrow{z} X(z)=1 \qquad \text{enter z plane}$$

Example for k = -1

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

$$y(n)=\delta(n+1) \xleftrightarrow{z} Y(z)=z \qquad \text{enter the z plane}$$

Example for k = 1

$$y(n)=x(n-(1))=n(n-1)=\delta(n-1)$$

$$y(n)=\delta(n-1) \xleftrightarrow{z} Y(z)=z^{-1} \qquad \text{enter the z plane}$$

ii) Scaling in the z domain

$$x(n) \xleftrightarrow{z} X(z) \quad \text{ROC } r_1<|z|<r_1$$

$$a^n x(n) \xleftrightarrow{z} X(a^{-1}z) \qquad \text{ROC}|a|\, r_1<|z|<|a|r_1$$

Scaling in the z-domain

Let y(n)= $a^n x(n)$

$$Y(z) = \sum_{n=-\infty}^{\infty} y(n) z^{-n}$$

$$= \sum_{n=-\infty}^{\infty} a^n x(n) z^{-n}$$

$$= \sum_{n=-\infty}^{\infty} x(n) \left(za^{-1}\right)^{-n}$$

$$= X\left(a^{-1}z\right)$$

$$\text{ROC}; r_1 < \left|a^{-1}z\right| = \left|\frac{z}{a}\right| r_2 \quad = |a| r_1 < |z| < |a| r_2$$

iii) Differentiation in z domain

This property state that if

$$x(n) \xleftrightarrow{z} X(z)$$

Then $$nx(n) \xleftrightarrow{z} -z\frac{d}{dz}X(z)$$

Proof: Consider the basic differentiation of z transform i.e.,

$$X(z) = \sum_{n=-\infty}^{\infty} x(n) z^{-n}$$

Diiferentiate both the side of above equation with respect to z we get,

$$\frac{d}{dz}X(z) = \sum_{n=-\infty}^{\infty} \frac{d}{dz}[x(n) z^{-n}]$$

$$= \sum_{n=-\infty}^{\infty} x(n) \frac{d}{dz}[z^{-n}]$$

$$= \sum_{n=-\infty}^{\infty} x(n)(-n) z^{-n-1}$$

$$= -\sum_{n=-\infty}^{\infty} x(n)(n) z^{-n} z^{-1}$$

$$= -z^{-1} \sum_{n=-\infty}^{\infty} \left[nx(n)\right] z^{-n}$$

$$= -z^{-1} Z\left[nx(n)\right] \text{ by definition of } z \text{ transform}$$

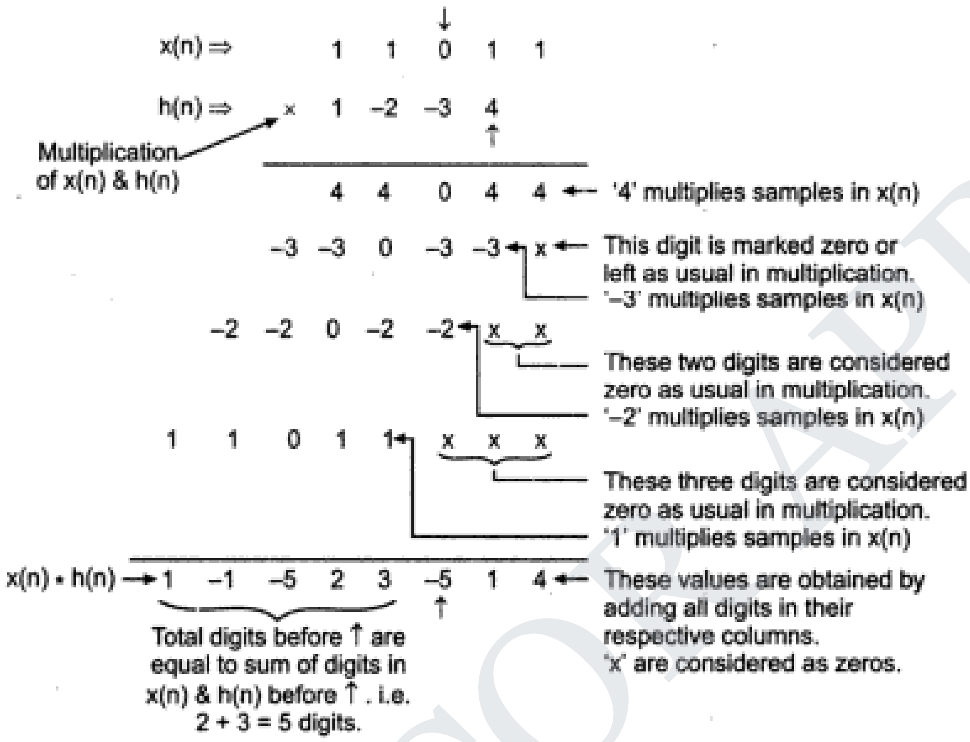
$$Z\left[nx(n)\right] == -z\frac{d}{dz}X(z)$$

Which is provided, the ROC of Z[nx(n)]is the same as that of X(z)

13. Find the linear convolution of $x(n) = \{1,1,\underset{\uparrow}{0},1,1\}$ with $h(x) = (1,-2,-3,\underset{\uparrow}{4})$ and verify the results of multiplication method with basic convolution equation and tabulation method. (Nov/Dec 2012) (16)

Solution:

The convolution of the two sequences is given by $y(n) = x(n) * h(n)$



Computation of convolutions using multiplication

In the sequence x(n) observe that there are 2 digits before the zero mark ↑ arrow. Similarly there are 3 digits before the zero mark ↑ arrow in h(n). Hence there will be 2+3=5 digits before the zero mark ↑ arrow in y(n). Thus the result of convolution is obtained as

$$y(n) = \left\{ 1, -1, -5, 2, 3, \underset{\uparrow}{-5}, 1, 4 \right\}$$

The basic convolution equation is given by

$$\sum y = \sum x . \sum h$$

Since y (n) = {1, -1, -5, 2, 3, $\underset{\uparrow}{-5}$, 1, 4}

$$\sum y = 1 - 1 - 5 + 2 + 3 - 5 + 1 + 4 = 0$$

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

Therefore $\sum x = 1 + 1 + 0 + 1 + 1 = 4$

and $h(x)=(1,-2,-3,4)$

$\sum h = 1-2-3+4=0$

Therefore $\sum x.\sum h = 4\cdot 0 = 0$

Thus $\sum y = \sum x.\sum h$ is proved.

ii) To verify result using $y(n)=\sum x(k)\cdot h(n-k)$

here let us compute the convention using the basic convolution equation. The given sequence are;

$x(n)=\{1,1,0,1,1\}$ and $h(x)=(1,-2,-3,4)$

These two sequence can be written as

$$\begin{array}{ll} x(-2)=1 & h(-3)=1 \\ x(-1)=1 & h(-2)=-2 \\ x(0)=0\leftarrow & h(-1)=-3 \\ x(1)=1 & h(0)=4\leftarrow \\ x(2)=1 & \end{array}$$

From above we can write can the following:

Lowest index of x(n) $\Rightarrow n_{xl}$=-2

Highest index of x(n) $\Rightarrow n_{xh}$=2

Lowest index of x(n) $\Rightarrow n_{hl}$=--3

Highest index of x(n) $\Rightarrow n_{hh}$=0

The convention equation is given by equation as

$$y(n)=\sum_{k=n_{xl}} x(k)h(n-k) \quad \text{and} \quad (n_{xl}+n_{hl})\le n\le (n_{xh}+n_{hh})$$

Putting values oh highest and lowest indicies in the above equation we get,

$$y(n)=\sum_{k=-2}^{2} x(k)h(n-k) \quad \text{and} \quad (-2-3)\le n\le (2+0)$$

$$y(n)=\sum_{k=-2}^{2} x(k)h(n-k) \quad \text{and} \quad -5\le n\le 2$$

Now let us compute y(n) for the complete range

n=-5 in equation 1 given y(-5)

$$y(-5)=\sum_{k=-2}^{2}x(k)h(-5-k)$$
$$=x(-2)h(-3)+x(-1)h(-4)+x(0)h(-5)+x(1)h(-6)+x(2)h(-7)$$
$$=(1\times 1)+(1\times 0)+(0\times 0)+(1\times 0)+(1\times 0)$$
$$\sin ce\, h(-4)=h(-5)=h(-6)=h(-7)=0$$
$$=1+0+0+0+0$$
$$=1 \qquad y(-5)=1$$

n= -4 in equation 1 given by y(-4)

$$y(-4)=\sum_{k=-2}^{2}x(k)h(-4-k)$$
$$=x(-2)h(-2)+x(-1)h(-3)+x(0)h(-4)+x(1)h(-5)+x(2)h(-6)$$
$$=(1\times -2)+(1\times 1)+(0\times 1)+(1\times 0)+(1\times 0)$$
$$\sin ce\, h(-4)=h(-5)=h(-6)=0$$
$$=-2+1+0+0+0$$
$$=-1 \qquad y(-4)=-1$$

n= -3 in equation 1 given by y(-3)

$$y(-3)=\sum_{k=-2}^{2}x(k)h(-3-k)$$
$$=x(-2)h(-1)+x(-1)h(-2)+x(0)h(-3)+x(1)h(-4)+x(2)h(-5)$$
$$=(1\times -3)+(1\times -2)+(0\times 1)+(1\times 0)+(1\times 0)$$
$$\sin ce\, h(-n)=h(-5)=0$$
$$=-3-2+0+0+0$$
$$=-5 \qquad y(-3)=1$$

n= -2 in equation 1 given by y(-2)

$$y(-2)=\sum_{k=-2}^{2}x(k)h(-2-k)$$
$$=x(-2)h(0)+x(-1)h(-1)+x(0)h(-2)+x(1)h(-3)+x(2)h(-4)$$
$$=(1\times 4)+(1\times -3)+(0\times -2)+(1\times 1)+(1\times 0)$$
$$\sin ce\, h(-n)=h(-4)=0$$
$$=4-3+0+1+0$$
$$=-2 \qquad y(-2)=2$$

n= -1 in equation 1 given by y(-1)

$$y(-1) = \sum_{k=-2}^{2} x(k)h(-1-k)$$
$$= x(-2)h(1) + x(-1)h(0) + x(0)h(-1) + x(1)h(-2) + x(2)h(-3)$$
$$= 0 + (1\times 4) + (0\times -3) + (1\times -2) + (1\times 1)$$
$$\text{since}\, h(1) = 0$$
$$= 0 + 4 + 0 - 2 + 1$$
$$= 3 \qquad y(-1) = 3$$

n= 0 in equation 1 given by y(0)

$$y(0) = \sum_{k=-2}^{2} x(k)h(-k)$$
$$= x(-2)h(2) + x(-1)h(1) + x(0)h(0) + x(1)h(-1) + x(2)h(-2)$$
$$= 0 + 0 + (0\times 4) + (1\times -3) + (1\times 2)$$
$$\text{since}\, h(2) = h(1) = 0$$
$$= 0 + 0 + 0 - 3 - 2$$
$$= -5 \qquad y(0) = -5$$

n= 1 in equation 1 given by y(1)

$$y(1) = \sum_{k=-2}^{2} x(k)h(1-k)$$
$$= x(-2)h(3) + x(-1)h(2) + x(0)h(1) + x(1)h(0) + x(2)h(-1)$$
$$= 0 + 0 + 0 + (1\times 4) + (1\times -3)$$
$$\text{since}\, h(3) = h(2) = h(1) = 0$$
$$= 0 + 0 + 0 + 4 - 3$$
$$= 1 \qquad y(1) = 1$$

n= 2 in equation 1 given by y(2)

$$y(2) = \sum_{k=-2}^{2} x(k)h(2-k)$$
$$= x(-2)h(4) + x(-1)h(3) + x(0)h(2) + x(1)h(1) + x(2)h(0)$$
$$= 0 + 0 + 0 + 0 + (1\times 4)$$
$$\text{since}\, h(4) = h(3) = h(2) = h(1) = 0$$
$$= 0 + 0 + 0 + 0 + 4$$
$$= 4 \qquad y(1) = 4$$

Thus the sequence y(n) obtained using basic computation of convolution is as following

y(n) = {1,-1,-5,2,3,-5,1,4}

iii) **Tabulation method**

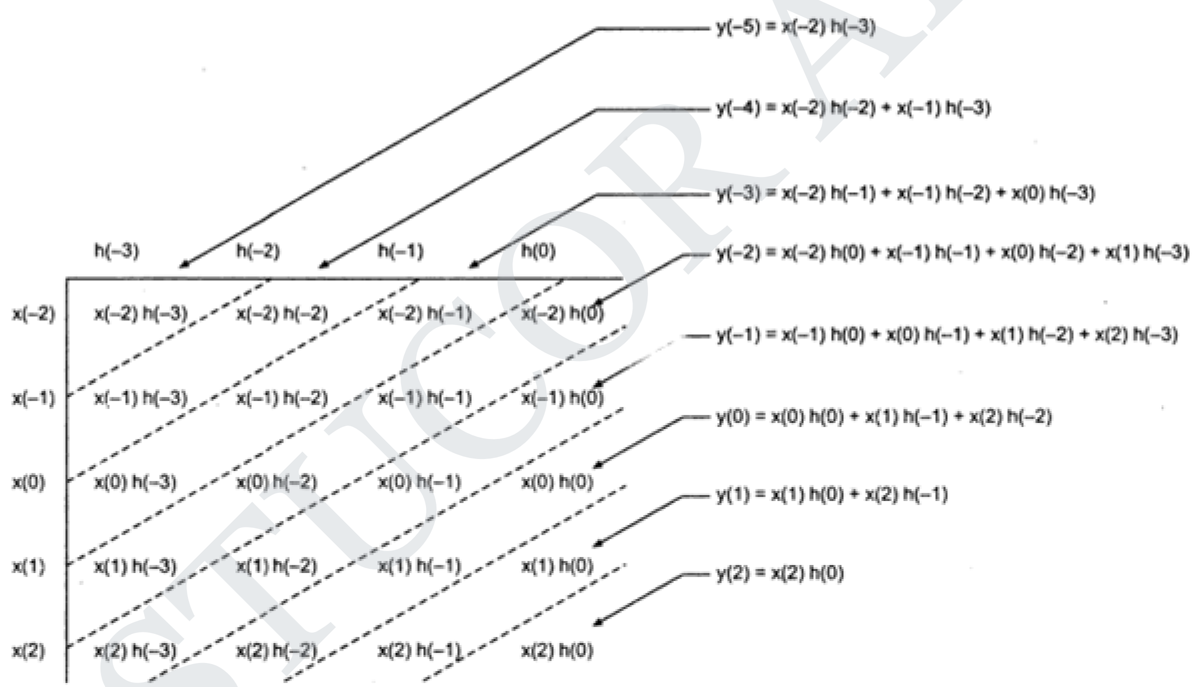
This example illustrates another easy method of computations of convolution

$$x(n) = \{1,1,0,1,1\} \quad h(x) = (1,-2,-3,4)$$

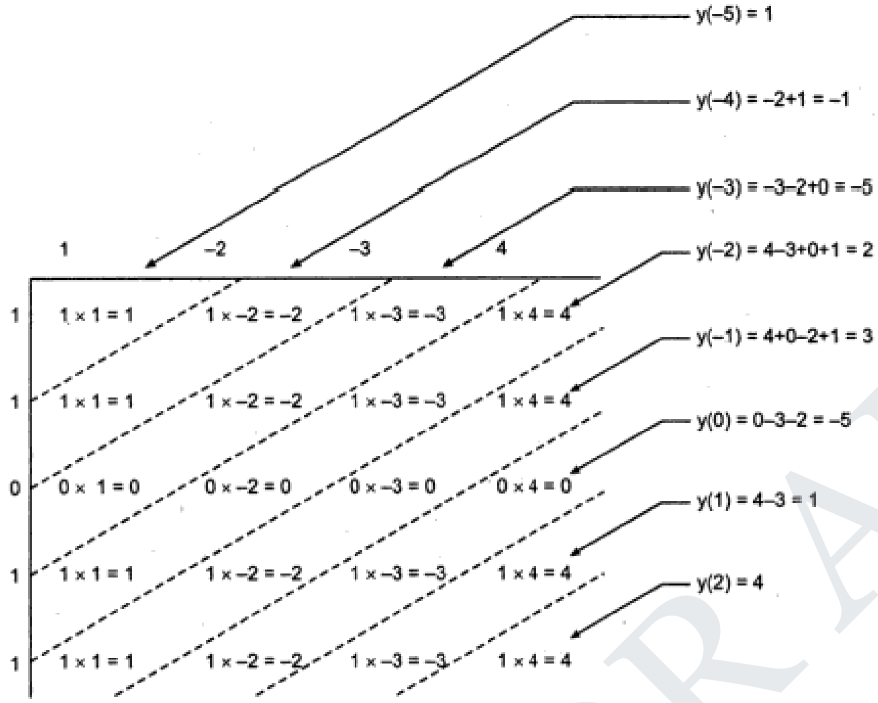
Sometimes this method is also called tabulation

The value of x(n) and h(n) can be written as following

$$\begin{array}{ll} x(-2)=1 & h(-3)=1 \\ x(-1)=1 & h(-2)=-2 \\ x(0)=0 \leftarrow & h(-1)=-3 \\ x(1)=1 & h(0)=4 \leftarrow \\ x(2)=1 & \end{array}$$



Computation of convolution using tabulation



Computation of convolution

Thus as shown in above fig h(-3), h(-2), h(-1) and h(0), from columns of table. And x(-2), x(-1), x(0), x(1), x(2) from, the row of the table. In the table the multiplication of x(n) and h(n)are written as shown. Then the multiplications are separately diagonally as shown by dotted line in fig

From given sequence x(n) and h(n) we have,

Lowest index of $x(n) \Rightarrow n_{xl} = -2$

Lowest index of $x(n) \Rightarrow n_{hl} = -3$

Hence lowest of $y(n) \Rightarrow n_{yl} = n_{xl} + n_{hl} = -2-3$

$$\therefore\ n_{yl} = -5$$

The dfirst element in y(n) will be y(-5). This element is equal to top left diaognal array. Ait contains only one multiplication i.e,

$$y(-5) = x(-2)\,h(-3)$$

the other diagonal array are sucessively y(-4), y(-3), y(-2),… as shown in fig finally the last element in the array is y(2) and it is the bottom right element in table

$$y(-2) = x(2)h(0)$$

Thus the complete sequance is

$$y(n) = \left\{ \begin{matrix} 1, -1, -5, 2, 3, -5, 1, 4 \\ \qquad\qquad\qquad\quad \uparrow \end{matrix} \right\}$$

14. Determine the response of the system whose input x(n) and unit sample response h(n) is given as follows:

$$x(n) = \begin{cases} \frac{1}{3}n & \text{for } 0 \le n \le 6 \\ 0 & \text{else here} \end{cases}$$

$$h(n) = \begin{cases} 1 & \text{for } -2 \le n \le 2 \\ 0 & \text{else here} \end{cases}$$

Sol : The input sequence x(n) is,

$$x(n) = \left\{ \begin{matrix} 0, \frac{1}{3}, \frac{2}{3}, \frac{3}{3}, \frac{4}{3}, \frac{5}{3}, \frac{6}{3} \\ \uparrow \qquad\qquad\qquad\qquad \end{matrix} \right\}$$

$$x(n) = \left\{ \begin{matrix} 0, \frac{1}{3}, \frac{2}{3}, 2, \frac{4}{3}, \frac{5}{3}, 2 \\ \uparrow \qquad\qquad\qquad\quad \end{matrix} \right\}$$

The unit sample response h(n) is $h(n) = \{1,1,1,1,1\}$

$$y(n) = \left\{ \begin{matrix} \frac{1}{3}, 1, 2, \frac{10}{3}, 5, 6, \frac{16}{3}, 5, \frac{11}{3}, 2 \\ \quad \uparrow \qquad\qquad\qquad\qquad\qquad \end{matrix} \right\}$$

x(n) ⇒				0 ↑	$\frac{1}{3}$	$\frac{2}{3}$	1	$\frac{4}{3}$	$\frac{5}{3}$	2	
h(n) ⇒					1	1	1 ↑	1	1		
					0	$\frac{1}{3}$	$\frac{2}{3}$	1	$\frac{4}{3}$	$\frac{5}{3}$	2
				0	$\frac{1}{3}$	$\frac{2}{3}$	1	$\frac{4}{3}$	$\frac{5}{3}$	2	×
			0	$\frac{1}{3}$	$\frac{2}{3}$	1	$\frac{4}{3}$	$\frac{5}{3}$	2	×	×
		0	$\frac{1}{3}$	$\frac{2}{3}$	1	$\frac{4}{3}$	$\frac{5}{3}$	2	×	×	×
	0	$\frac{1}{3}$	$\frac{2}{3}$	1	$\frac{4}{3}$	$\frac{5}{3}$	2	×	×	×	×
y(n) =	0	$\frac{1}{3}$	1 ↑	2	$\frac{10}{3}$	5	6	$\frac{16}{3}$	5	$\frac{11}{3}$	2

Computation of convolution

UNIT-III - DISCRETE FOURIER TRANSFORM AND COMPUTATION

PART A

1. Define DFT.

It is a finite duration discrete frequency sequence, which is obtained by sampling at N equally spaced points over the period extending from ω=0 to 2π. DFT is defined as $X(\omega) = x(n)e^{-j\omega n}$. Here x(n)is the discrete time sequence X(ω)is the Discrete Fourier Transform of x(n).

2. Define N point DFT.

The DFT of discrete sequence x(n) is denoted by X(K). It is given by, X(K)= $\sum_{k=1}^{N-1} x(n)e^{\frac{-j2\pi kn}{N}}$ .Since this summation is taken for N points, it is called as N-point DFT.

3. Define Twiddle factor

The Twiddle factor is defined as $W_N = e^{-j\frac{2\pi}{N}}$

4. What is DFT of unit impulse δ(n)? (May/June 2013)

The DFT of unit impulse δ(n) is X(K)= $\sum_{k=1}^{N-1} x(n)e^{\frac{-j2\pi kn}{N}} = \sum_{k=1}^{N-1} \delta(n)e^{\frac{-j2\pi kn}{N}} = 1$

5. Define circularly even sequence

A Sequence is said to be circularly even if it is symmetric about the point zero on the circle $x(N-n) = x(n),\ 1 \le n \le N-1$

6. Define circularly odd sequence

A Sequence is said to be circularly odd if it is anti symmetric about point $x(0)$ on the circle

7. Define circularly folded sequences (May/June 2014)

A circularly folded sequence is represented as $x((-n))N$. It is obtained by plotting $x(n)$ in clockwise direction along the circle

8. State circular convolution

This property states that multiplication of two DFT is equal to circular convolution of their sequence in time domain.

9. State parseval's theorem (Nov/Dec 2014)

Consider the complex valued sequences $x(n)$ and $y(n)$. If $x(n)y^*(n) = \frac{1}{N}\sum_{k=0}^{N-1} X(k)Y^*(k)$

10. List the properties of DFT.

Linearity, Periodicity, Circular symmetry, symmetry, Time shift, Frequency shift, complex conjugate, convolution, correlation and Parseval's theorem.

11. State Linearity property of DFT.

DFT of linear combination of two or more signals is equal to the sum of linear combination of DFT of individual signal.

12. When a sequence is called circularly even?

The N point discrete time sequence is circularly even if it is symmetric about the point zero on the circle.

13. What is the condition of a sequence to be circularly odd?

An N point sequence is called circularly odd it if is anti-symmetric about point zero on the circle.

14. Why the result of circular and linear convolution is not same?

Circular convolution contains same number of samples as that of x (n) and h (n), while in linear convolution, number of samples in the result (N) are,

$N=N_1+N_2-1$

Where N_1= Number of samples in x (n)

N_2=Number of samples in h (n)

15. What is circular time shift of sequence? (May/June 2014)

Shifting the sequence in time domain by '1' samples is equivalent to multiplying the sequence in frequency domain by W_N^{kl}

16. What is the disadvantage of direct computation of DFT?

For the computation of N-point DFT, N^2 complex multiplications and N[N-1] Complex additions are required. If the value of N is large than the number of computations will go into lakhs. This proves inefficiency of direct DFT computation.

17. What is the way to reduce number of arithmetic operations during DFT computation?

Number of arithmetic operations involved in the computation of DFT is greatly reduced by using different FFT algorithms as follows.

Radix-2 FFT algorithms.

Radix-2 Decimation in Time (DIT) algorithm.

Radix-2 Decimation in Frequency (DIF) algorithm.

Radix-4 FFT algorithm.

18. What is the computational complexity using FFT algorithm?

Complex multiplications = $N/2 \log_2 N$

Complex additions = $N \log_2 N$

19. How linear filtering is done using FFT?

Correlation is the basic process of doing linear filtering using FFT. The correlation is nothing but the convolution with one of the sequence, folded. Thus, by folding the sequence h (n), we can compute the linear filtering using FFT.

20. What is zero padding? What are its uses? (Nov/Dec 2014)

Let the sequence x (n) has a length L. If we want to find the N point DFT (N>L) of the sequence x (n). Then appending zeros in the given sequence is called as Zero padding. This is known as zero padding. The uses of padding a sequence with zeros are (i) We can get 'better display' of the frequency spectrum (ii) With zero padding, the DFT can be used in linear filtering.

21. Why FFT is needed?

The direct evaluation of the DFT using the formula requires N^2 complex multiplications and N (N-1) complex additions. Thus for reasonably large values of N (in order of 1000) direct evaluation of the DFT requires an inordinate amount of computation. By using FFT algorithms the number of computations can be reduced. For example, for an N-point DFT, The number of complex multiplications required using FFT is $N/2\log_2 N$. If N=16, the number of complex multiplications required for direct evaluation of DFT is 256, whereas using DFT only 32 multiplications are required.

22. What is the speed of improvement factor in calculating 64-point DFT of a sequence using direct computation and computation and FFT algorithms? Or

Calculate the number of multiplications needed in the calculation of DFT and FFT with 64-point sequence. The number of complex multiplications required using direct computation is N^2=642=4096. The number of complex multiplications required using FFT is $N/2 \log_2 N = 64/2\log_2 64$=192. Speed improvement factor = 4096/192=21.33

23. What is the main advantage of FFT?

FFT reduces the computation time required to compute discrete Fourier transform. Calculate the number of multiplications needed in the calculation of DFT using FFT algorithm with using FFT algorithm with 32-point sequence. For N-point DFT the number of complex multiplications needed using FFT algorithm is $N/2 \log_2 N$. For N=32, the number of the complex multiplications is equal to $32/2\log_2 32$=16*5=80.

24. What is FFT? What is its advantage (Nov/Dec 2012)

The fast Fourier transforms (FFT) is an algorithm used to compute the DFT. It makes use of the Symmetry and periodically properties of twiddles factor W^K_N to effectively reduce the DFT computation time. It is based on the fundamental principle of decomposing the computation of the DFT of a sequence of length N into successively smaller discrete Fourier transforms. The FFT algorithm provides speed-increase factors, when compared with direct computation of the DFT, of approximately 64 and 205 for 256-point and 1024-point transforms, respectively.

25. How many complex additions and complex multiplications are required to compute N-point DFT using radix-2 FFT?

The number of complex additions and complex multiplications required to compute N-point DFT using radix-2 FFT are $N \log_2 N$ and $N/2 \log_2 N$ respectively.

26. What is meant by radix-2 FFT?

The FFT algorithm is most efficient in calculating N-point DFT. If the number of output points N can be expressed as a power of 2, that is, $N=2^M$, where M is an integer, Then this algorithm is known as radix-2 FFT algorithm.

27. What is a decimation-in-time algorithm?

Decimation-in-time algorithm is used to calculate the DFT of a N-point Sequence. The idea is to break the N-point sequence into two sequences, the DFTs of which can be combined to give the DFT of the original N-point sequence. Initially the N-point sequence is divided into two N/2-point sequences $x_e(n)$ and $x_0(n)$, which have the even and odd members of x(n) respectively. The N/2 point DFTs of these two sequences are evaluated and combined to give the N point DFT. Similarly the N/2 point DFTs can be expressed as a combination of N/4 point DFTs. This process is continued till we left with 2-point DFT. This algorithm is called Decimation-in-time because the sequence x(n) is often splitted into smaller sub sequences.

28. What are the differences and similarities between DIF and DIT algorithms? (May/June 2014)

Differences:

For DIT, the input is bit reversal while the output is in natural order, whereas for DIF, the input is in natural order while the output is bit reversed.

The DIF butterfly is slightly different from the DIT butterfly, the difference being that the complex multiplication takes place after the add-subtract operation in DIF.

Similarities:

Both algorithms require same number of operations to compute the DFT. Bot algorithms can be done in place and both need to perform bit reversal at some place during the computation.

29. What are the applications of FFT algorithms?

- Linear filtering
- Correlation
- Spectrum analysis

30. What is a decimation-in-frequency algorithm?

In this the output sequence X (K) is divided into two N/2 point sequences and each N/2 point sequences are in turn divided into two N/4 point sequences.

31. Distinguish between DFT and DTFT.

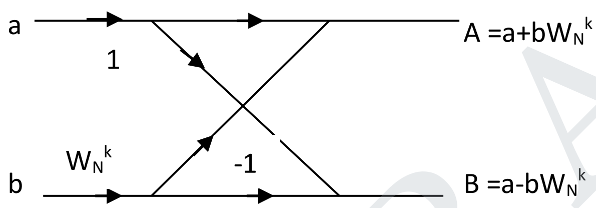
S.No	DFT	DTFT
1.	Obtained by performing sampling operation in both the time and frequency domains.	Sampling is performed only in time domain.

2.	Discrete frequency spectrum	Continuous function of ω

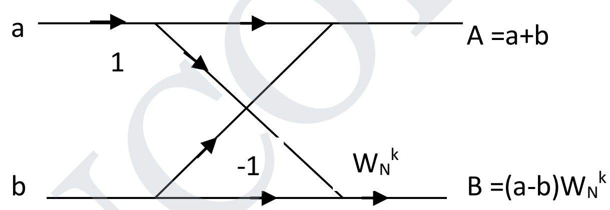
32. Distinguish between Fourier series and Fourier transform.

S.No.	Fourier Series	Fourier transform
1	Gives the harmonic content of a periodic time function.	Gives the frequency information for an aperiodic signal.
2.	Discrete frequency spectrum	Continuous frequency spectrum

33. Draw the butterfly diagram for decimation in time FFT algorithm. (Nov/Dec 2013, Apr/May 2015)



34. Draw the butterfly diagram for decimation in time FFT algorithm. (May/June 2013)



35. In eight point decimation in time (DIT), what is the gain of the signal path that goes from X(7) to X(2)? (Nov/Dec 2013)

Gain = $W^0_8 \cdot W^0_8(-1) \cdot W^2_8 = j$

PART B

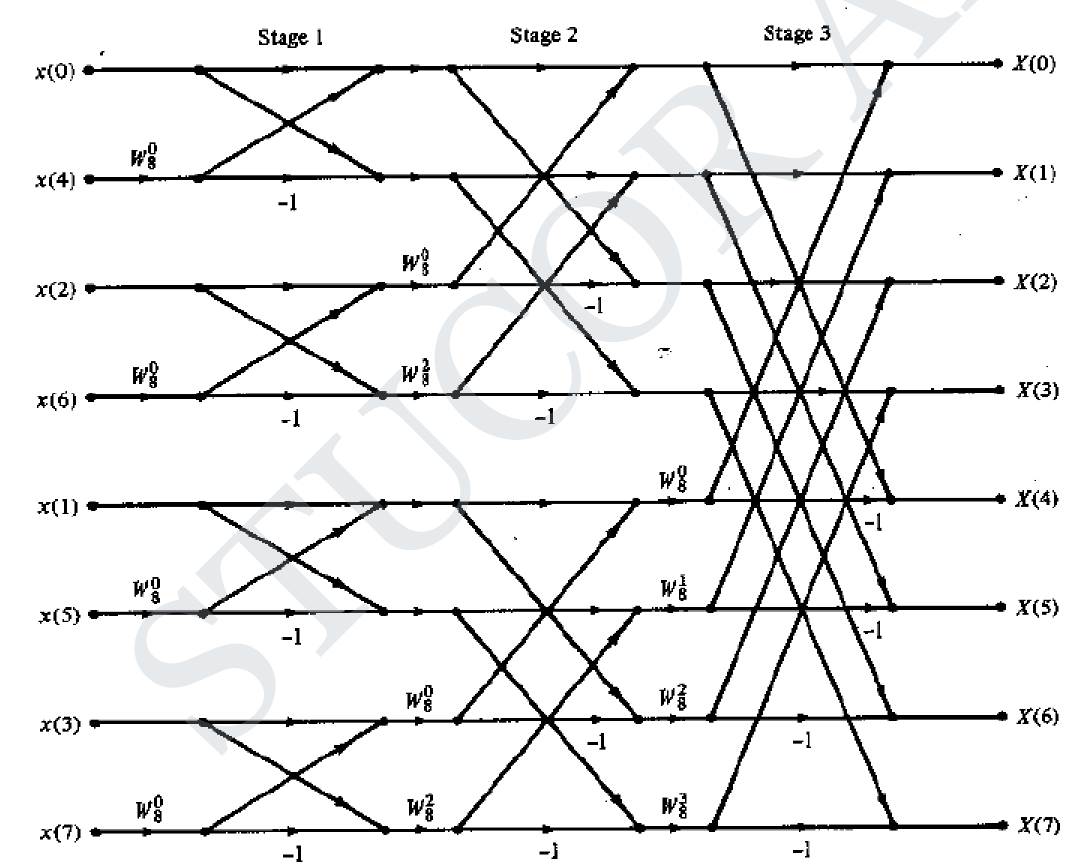
1. Obtain the 8 point DFT using the radix 2 DIT FFT algorithm for $x(n) = \{0,1,2,3,4,5,6,7\}$ (16)

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

$$(W_N)^K = \left(e^{\frac{-j2\pi}{N}}\right)^K$$

$$W_8^0 = 1$$
$$W_8^1 = 0.707 - j0.707$$
$$W_8^2 = -j$$
$$W_8^3 = -0.707 - j0.707$$

Using butterfly diagram



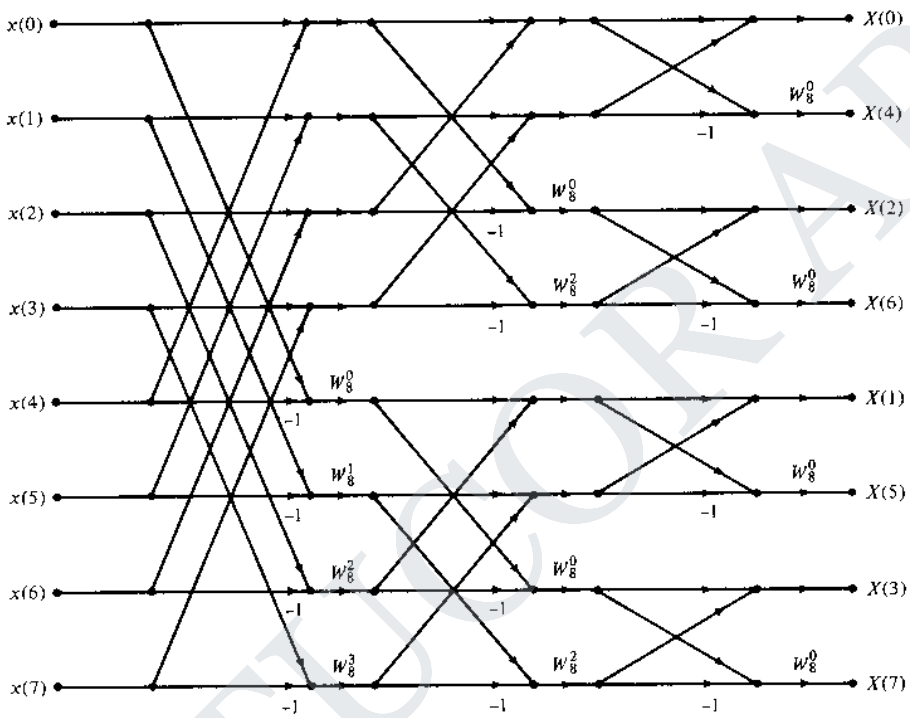
X(k) = {28, -4+j9.656, -4+j4, -4+j1.656, -4, -4-j1.656, -4-j4, -4-j9.656}

2.Given , X(k) = {28,-4+j9.656,-4+4j,-4+j1.656,-4,-4-j1.656,-4-4j,-4-j4,-4-j9.656}. Find x(n) using inverse DIT FFT algorithm. **(16)**

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

$$(W_N)^K = \left(e^{\frac{-j2\pi}{N}}\right)^K$$

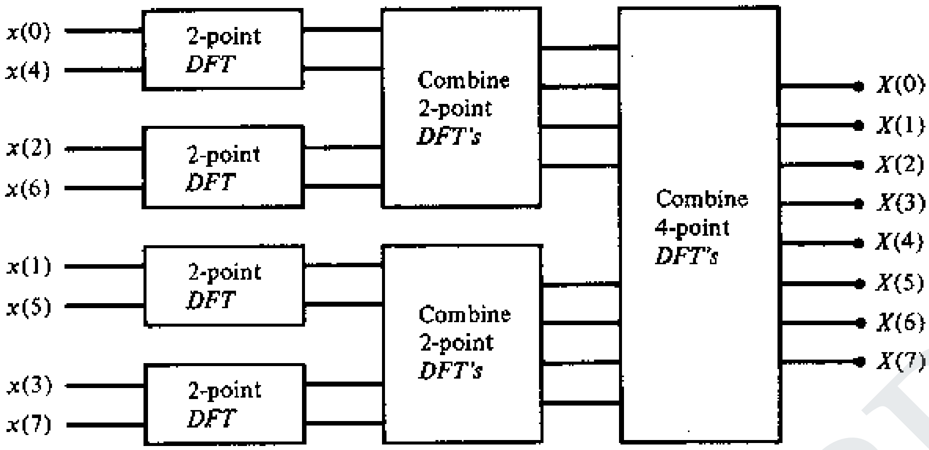
$W_8^0 = 1$
$W_8^1 = 0.707 - j0.707$
$W_8^2 = -j$
$W_8^3 = -0.707 - j0.707$



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

3. Derive and draw the radix -2 DIT algorithm for FFT of 8 points. (Nov/Dec 2012) (May/June 2012) (8)

Ans: Basic butterfly diagram,



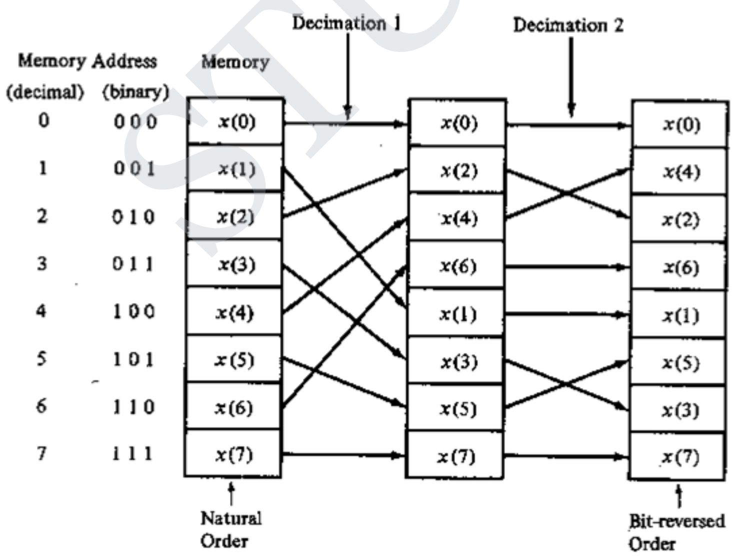
$$F_1(k) = F\{f_1(2n)\} + W_{N12}^{k} F\{f_1(2n+1)\} \qquad k = 0,1,\dots,\frac{N}{4}-1; \quad n = 0,1,\dots,\frac{N}{4}-1;$$

$$F_1\left(k+\frac{N}{4}\right) = F\{f_1(2n)\} + W_{N12}^{k} F\{f_1(2n+1)\} \quad k = 0,1,\dots,\frac{N}{4}-1; \quad n = 0,1,\dots,\frac{N}{4}-1;$$

$$F_2(k) = F\{f_2(2n)\} + W_{N12}^{k} F\{f_2(2n+1)\} \qquad k = 0,1,\dots,\frac{N}{4}-1; \quad n = 0,1,\dots,\frac{N}{4}-1;$$

$$F_2\left(k+\frac{N}{4}\right) = F\{f_2(2n)\} + W_{N12}^{k} F\{f_2(2n+1)\} \quad k = 0,1,\dots,\frac{N}{4}-1; \quad n = 0,1,\dots,\frac{N}{4}-1;$$

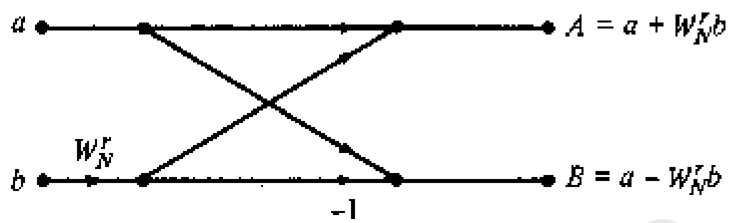
F (*) represent fourior transformer

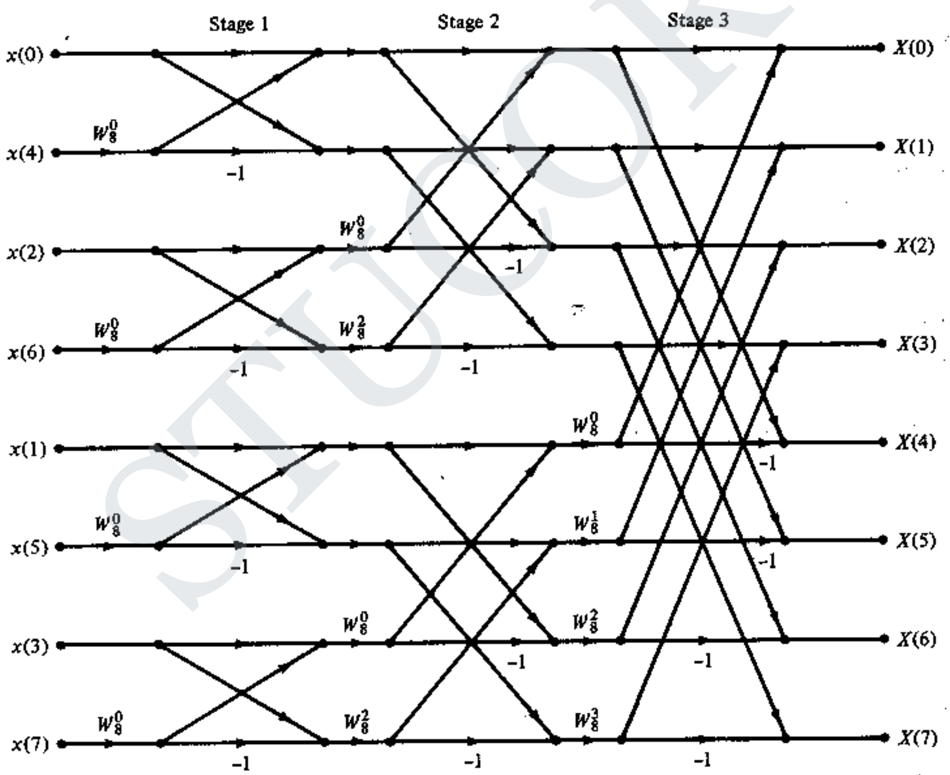


(a)

$(n_2n_1n_0)$	$\rightarrow$	$(n_0n_2n_1)$	$\rightarrow$	$(n_0n_1n_2)$
(0 0 0)	$\rightarrow$	(0 0 0)	$\rightarrow$	(0 0 0)
(0 0 1)	$\rightarrow$	(1 0 0)	$\rightarrow$	(1 0 0)
(0 1 0)	$\rightarrow$	(0 0 1)	$\rightarrow$	(0 1 0)
(0 1 1)	$\rightarrow$	(1 0 1)	$\rightarrow$	(1 1 0)
(1 0 0)	$\rightarrow$	(0 1 0)	$\rightarrow$	(0 0 1)
(1 0 1)	$\rightarrow$	(1 1 0)	$\rightarrow$	(1 0 1)
(1 1 0)	$\rightarrow$	(0 1 1)	$\rightarrow$	(0 1 1)
(1 1 1)	$\rightarrow$	(1 1 1)	$\rightarrow$	(1 1 1)

(b)

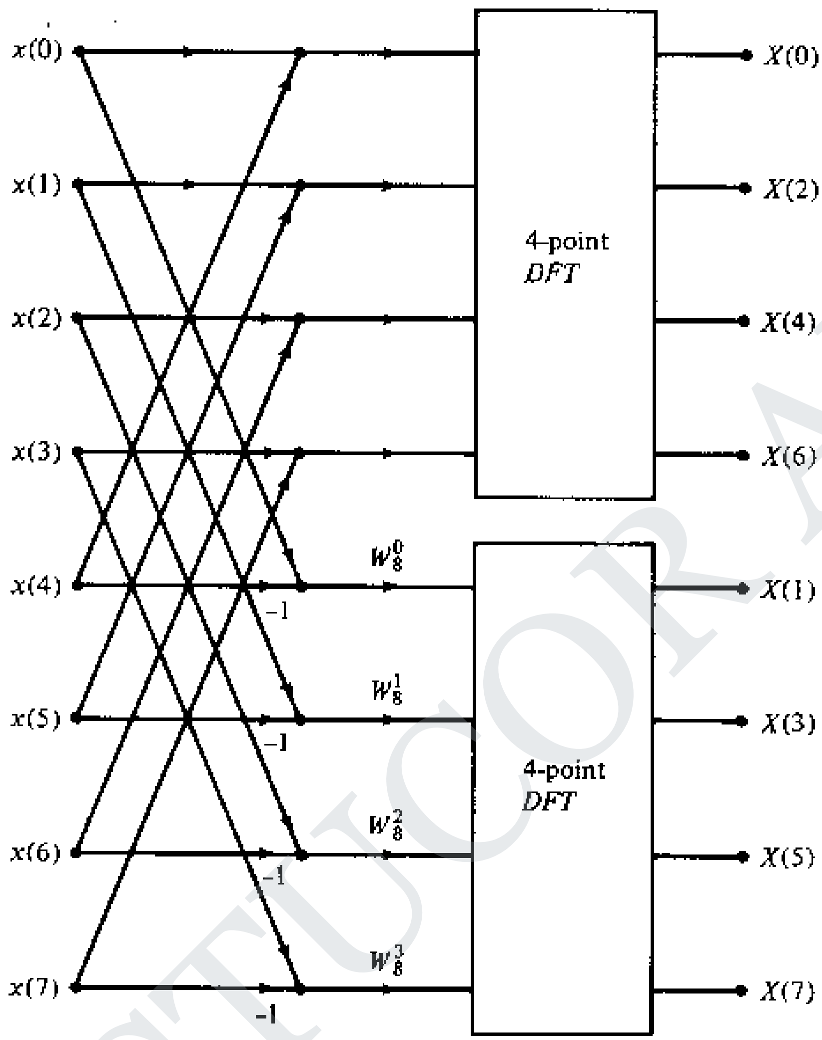


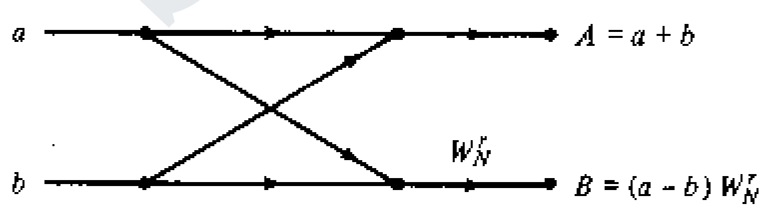


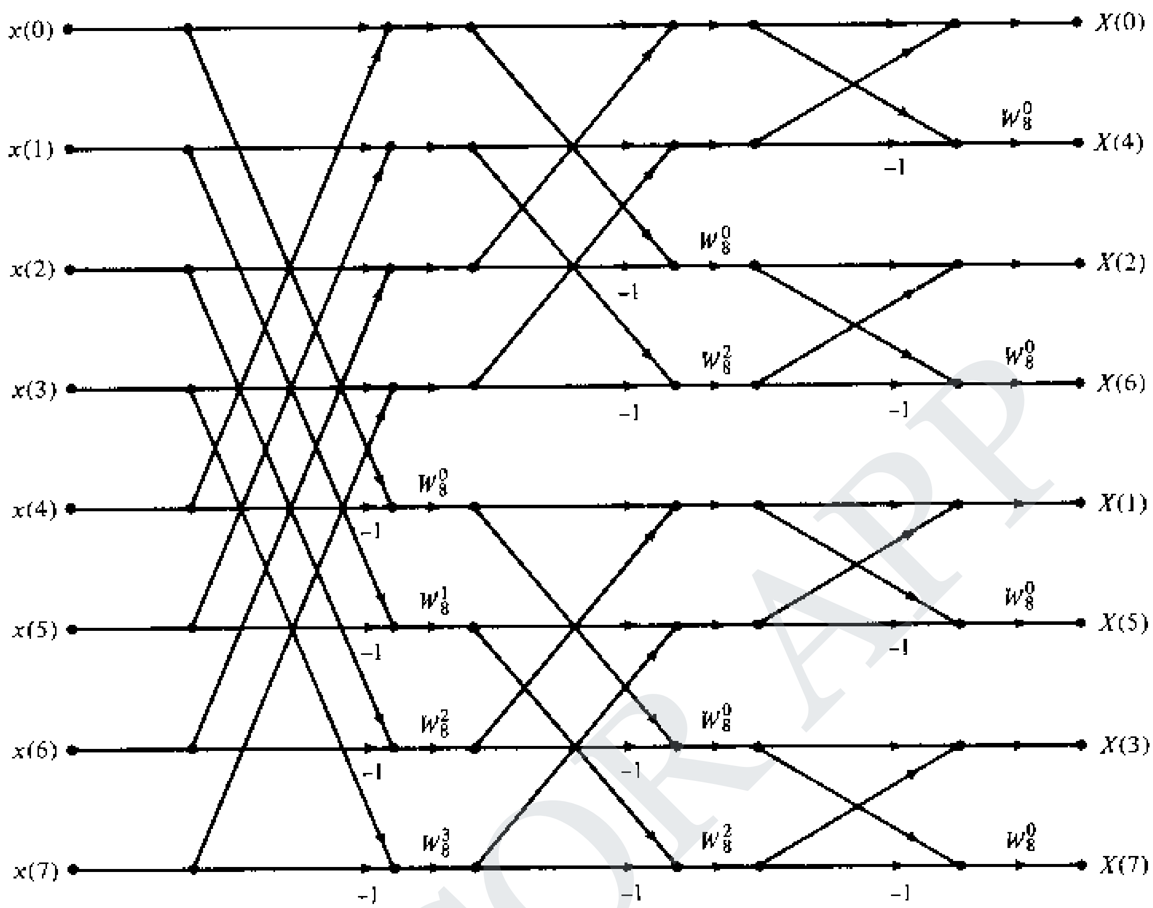
Complete butterfly diagram.

4. Derive and draw the radix -2 DIF algorithm for FFT of 8 points (Nov/Dec 2014) (Apr/May 2015) (8)

Ans: Basic butterfly diagram,







Complete butterfly diagram.

5. Explain the methods of filtering long data sequences.

- Overlap add method
- Overlap save method

Overlap Add Method

Use a block length of N=L+P-1

Append (L-1) zeros to h[n] and compute the N-length DFT H[k] once. It will be used for all blocks.

Start the input block index at 0.

Repeat the following:

Get the next length L sequence of input from x[n] starting at the block index

Append (P-1) zeros and compute the N-length DFT X[k].

Compute the N-point IDFT of (H[k]X[k]/N) to get a partial output sequence.

Overlap the last partial output sequence with the current output sequence by adding the last (P-1) outputs of the last partial output sequence to the first (P-1) outputs of the current partial output sequence. Output the first L outputs of the sum. Save the remaining (P-1) outputs for use with the next block of L input values

Overlap Save Method

Use a block length of N=L+P-1

Append (L-1) zeros to h[n] and compute the N-length DFT H[k] once. It will be used for all blocks.

Start the input block index at 0.

Initialize the current x[n] to all zeros.

Repeat the following:

Get the next length L sequence of input from x[n] starting at the block index

Store it into the last L locations of x[n].

Compute the N-length DFT X[k].

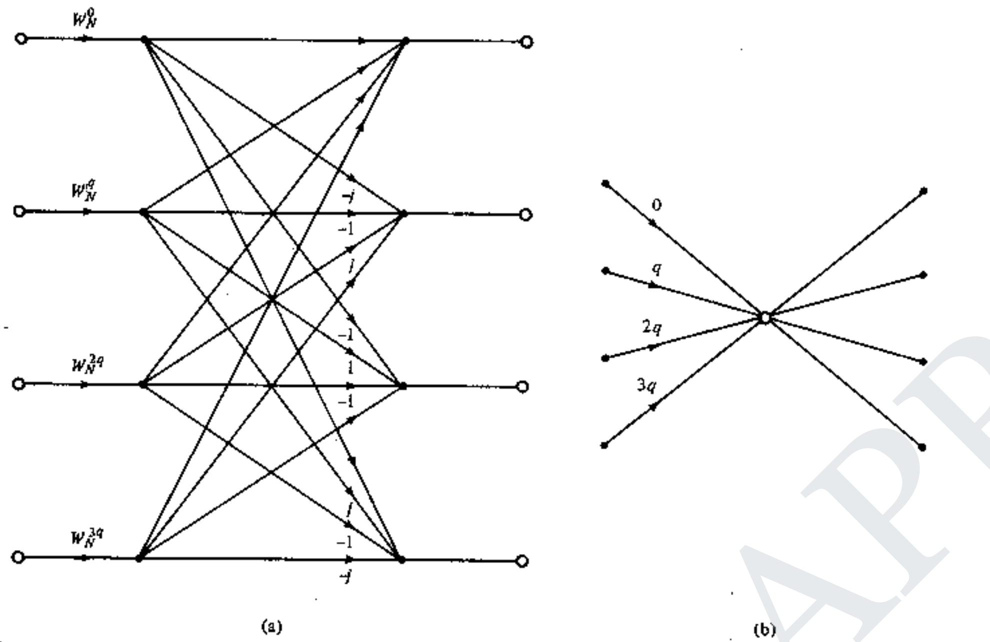
Compute the N-point IDFT of (H[k]X[k]/N) to get a temporary output sequence with

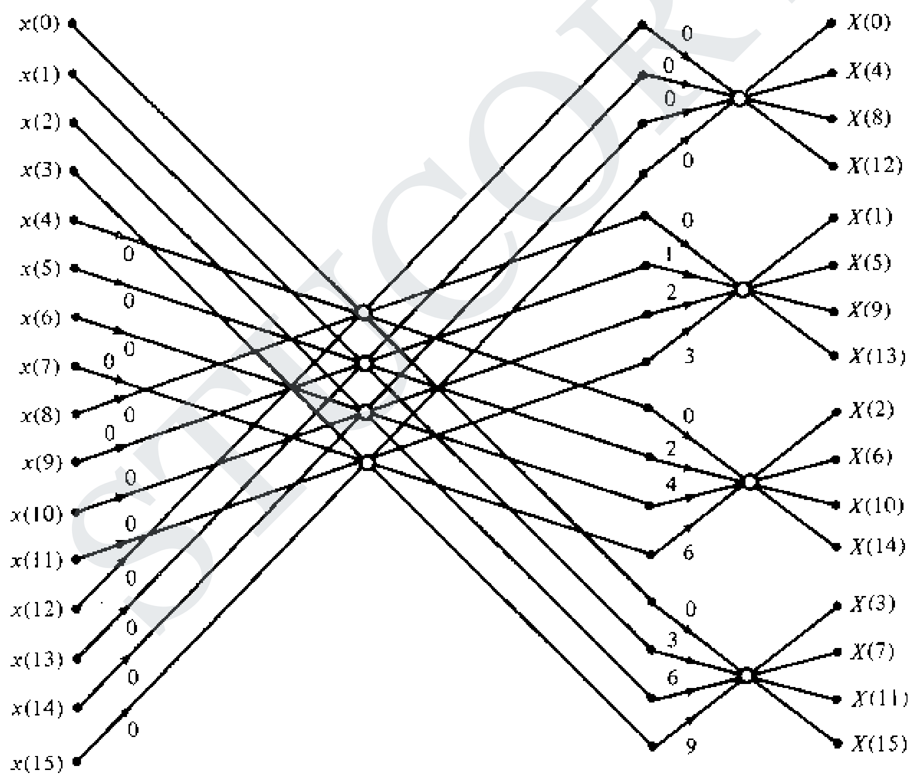
some valid results and some invalid results. Output the L valid results for linear convolution at the end of the temporary output sequence and discard the first (P-1) invalid results. (Refer to linear convolution example above.) Move the last (P-1) values in x[n] to the first [P-1] entries to be used again in the next Block computation

6. Derive 16 point radix –4 DIF-FFT algorithm.

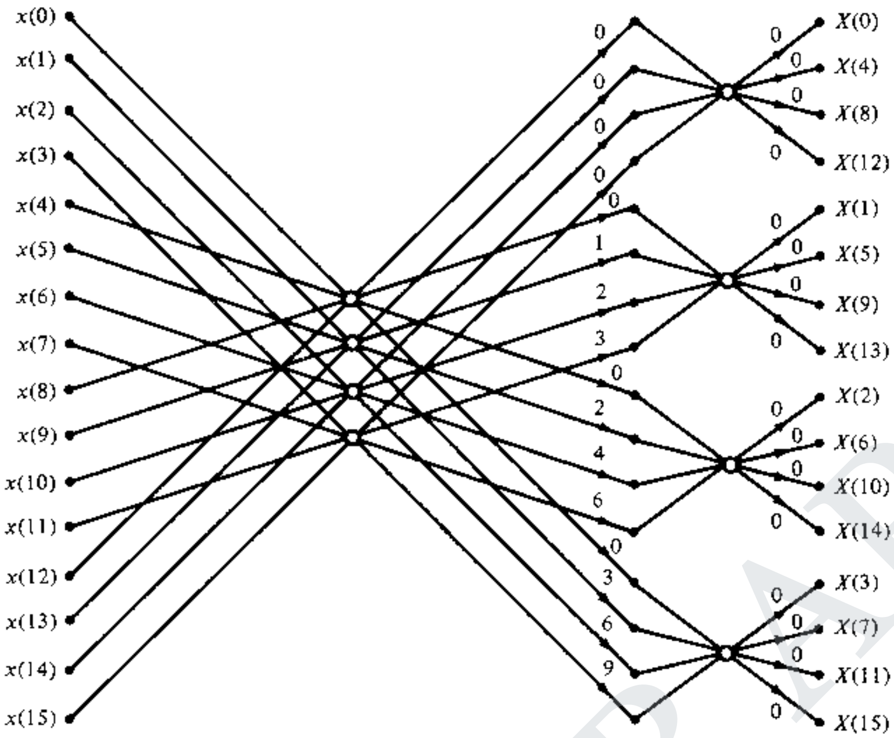
Ans: Basic butterfly diagram,

$$\begin{bmatrix} X(0,q) \\ X(1,q) \\ X(2,q) \\ X(3,q) \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & -j \\ 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & j \end{bmatrix} \begin{bmatrix} 1 & 0 & 1 & 0 \\ 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & -1 \end{bmatrix} \begin{bmatrix} W_N^0 F(0,q) \\ W_N^1 F(1,q) \\ W_N^2 F(2,q) \\ W_N^3 F(3,q) \end{bmatrix}$$





(c) 16-point radix-4 DIT algorithm with input in normal order and output in bit-reversed order



(d) 16-point, radix-4 DIF algorithm with input in normal order and output in bit-reversed order

Complete butterfly diagram

7. State and prove any five properties of DFT (Nov/Dec 2014) (May/June 2012) (8)

Solution:

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

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

Where W_N is defined as

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

$$x(n)\xleftrightarrow[N]{DFT}X(k)$$

Periodicity: If x(n) and X(k) are an N-point DFT pair, then

$$x(n+N)=x(n)\ \text{ for all n}$$
$$X(k+N)=X(k)\ \text{for all k}$$

We have evaluated DFT at m = 0, 1,,…, N-1. There after $(m \geq N)$, it shows periodicity.

For example

$$X(m) = X(N+m) = X(2N+m) = X(-N+m) = X(-2N+m) = X(-kN+m)$$

Where k is an integer.

Proof:

$$X(kN+m) = \sum_{n=0}^{N-1} x_n e^{\frac{-j2\pi n(kN+m)}{N}}$$
$$= \sum_{n=0}^{N-1} x_n e^{\frac{-j2\pi nm}{N}} e^{-j2\pi kn}$$

Both k and n are integers. Hence $e^{-j2\pi kn}=1$; Therefore from (1) we set

$$X(kN+m) = \sum_{n=0}^{N-1} x_n e^{\frac{-j2\pi nm}{N}} = X(m)$$

Linearity. If

$$x_1(n) \xleftrightarrow[N]{DFT} X_1(k)$$
$$x_2(n) \xleftrightarrow[N]{DFT} X_2(k)$$

Then for any real valued or complex valued constant a_1 and a_2,

$$a_1 x_1(n) + a_2 x_2(n) \xleftrightarrow[N]{DFT} a_1 X_1(k) + a_2 X_2(k)$$

Circular symmetries of a sequence. As we have seen, the N-point DFT of a finite duration sequence x(n), of the length $L \le N$, is equivalent to the N- point DFT of a periodic sequence x_p (n), of period N, which is obtained by periodically extending x(n), that is,

$$x_p(n) = \sum_{\ell=-\infty}^{\infty} x(n-\ell N)$$

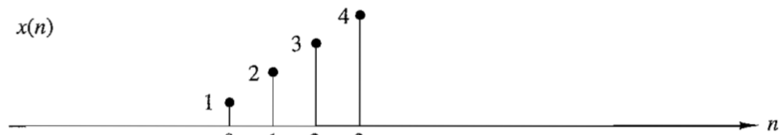
Now suppose that we shift the periodic sequence $x_p(n)$ by k units to the right. Thus we obtain another periodic sequence

$$x'_p(n) = x_p(n-k) = \sum_{\ell=-\infty}^{\infty} x(n-k-\ell N)$$

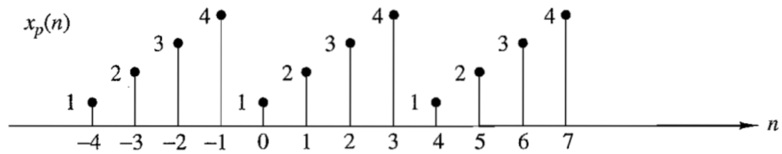
The finite duration sequence

$$x'(n) = \begin{cases} x'_p & 0 \le n \le -1 \\ 0 & \text{otherwise} \end{cases}$$

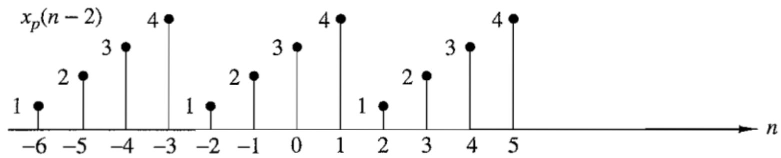
Is related to the original sequence x(n) by a circular shift. This relationship is illustrated in fig for N=4



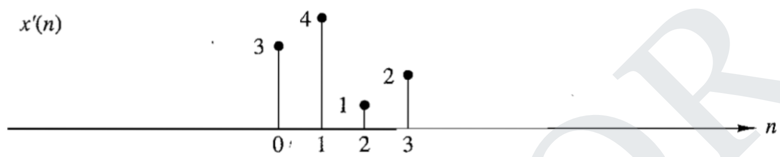
(a)



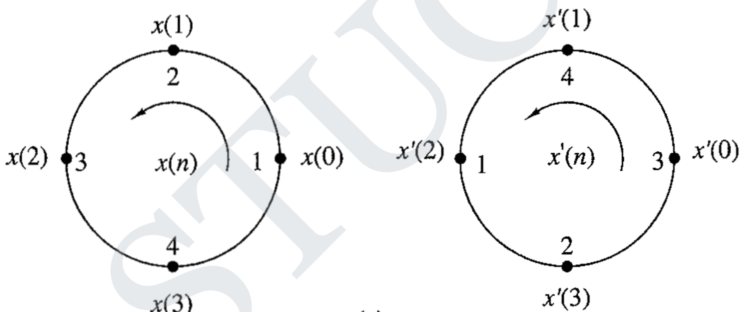
(b)



(c)



(d)



(e)

Circular shift of a sequence

In general, the circular shift of the sequence of the sequence can be represented as the index modulo N. thus we can write

$$x'(n) = x(n-k, \text{modulo } N)$$
$$= x(n-k)_N$$

For example, if k = 2 and N= 4, we have

$$x'(n) = x(2-k)_4$$

Which implies that

$$x'(0) = x(-2)_4 = x(2)$$
$$x'(1) = x(-1)_4 = x(3)$$
$$x'(2) = x(0)_4 = x(0)$$
$$x'(3) = x(1)_4 = x(1)$$

Hence x'(n) is simply x(n) shifted circulsrly by two units in time, where the counter clockwise direction has been arbitarly selected as the positive direction. Thus we concludes that a circular bshift an N point sequence is equivalent to a linear shift of its periodic extension and vise versa.

The inherent periodicity resulting from the arrguments of the N point sequence on the cicumference of a circle dictatates a different difinition of even and odd symmetry and time reversel of a sequence. An N-point sequence is called circularly even if it symmetric about the point zero on the circle. This implies that

$$x(N-n) = x(n) \quad 1 \le n \le N-1$$

An N-point sequence is called circularly even if it symmetric about the point zero on the circle. This implies that

$$x(N-n) = -x(n) \quad 1 \le n \le N-1$$

The timing reversal of an N-point sequence is attained by reversing its samples about the points zero on the circle. Thus the sequence $x(-n)_N$ is simple given as

$$x(-n)_N = x(N-n) \quad 0 \le n \le N-1$$

This time reverse is equlaent to plotting x(n) in a clockwise direction on a circle.

An equilent definition of even and odd sequence for the associated periodic sequence $x_p(n)$ is given as follows

even: $x_p(n) = x_p(-n) = x_p(N-n)$

odd: $x_p(n) = - x_p(-n) = - x_p(N-n)$

If the periodic sequence is complex valued, we have

Connjugated even: $x_p(n) = x'_p(N-n)$

Conjucated odd: $x_p(n) = - x^*_p(N-n)$

These relationships suggest that we decomposed the sequence $x_p(n)$ as

$$x_p(n) = x_{pe}(n) + x_{po}(n)$$

where

$$x_{pe}(n) = \frac{1}{2}\left[x_p(n) + x_p^*(N-n)\right]$$
$$x_{po}(n) = \frac{1}{2}\left[x_p(n) - x_p^*(N-n)\right]$$

Symmetry propoerties of the DFT. The summery properties for the DFT for the DFT can be obtaibed by applying the methodology previously used for the Fourier transform. Let us assume that the N- point sequence x(n) nand its DFT are both complex valued. Then the sequence can be expessed as

$$x(n) = x_R(n) + jx_1(n), \quad 0 \le n \le N-1$$
$$x(k) = X_R(k) + jX_1(k), \quad 0 \le n \le N-1$$

Substitute x(n) in DFT eqn. and X(k) in IDFT eqn. we obtain the following eqns.

$$X_R(k) = \sum_{n=0}^{N-1}\left[x_R(n)\cos\frac{2\pi kn}{N} + x_1(n)\sin\frac{2\pi kn}{N}\right]$$
$$X_1(k) = -\sum_{n=0}^{N-1}\left[x_R(n)\sin\frac{2\pi kn}{N} - x_1(n)\cos\frac{2\pi kn}{N}\right]$$
$$x_R(n) = \frac{1}{N}\sum_{n=0}^{N-1}\left[X_R(k)\cos\frac{2\pi kn}{N} - X_1(k)\sin\frac{2\pi kn}{N}\right]$$
$$x_1(k) = \frac{1}{N}\sum_{n=0}^{N-1}\left[X_R(k)\sin\frac{2\pi kn}{N} - X_1(k)\cos\frac{2\pi kn}{N}\right]$$

Multiplication of two sequences. If

$$x_1(n) \xleftrightarrow[N]{DFT} X_1(k)$$

And $\quad x_2(n) \xleftrightarrow[N]{DFT} X_2(k)$

Then $\quad x_1(n)x_2(n) \xleftrightarrow[N]{DFT} \frac{1}{N}X_1(k) \times X_2(k)$

This property is the duel of its proof following simply by interchanging the roles of time and frequency in the expression for the circular convolution of two sequences

Paraseval's theorem. For complex valued sequence x(n)and y(n), in general, if

$$x(n) \xleftrightarrow[N]{DFT} X(k)$$

and $\quad y(n) \xleftrightarrow[N]{DFT} Y(k)$

then $\quad \sum_{n=0} x(n)y^*(n) = \frac{1}{N}\sum_{k=0}^{N-1} X(k)Y^*(k)$

proof : the property following immediately from the circular correlation property we have

$$\sum_{n=0}^{N-1} x(n)y^*(n) = \overline{r}_{xy}(0)$$

And

$$\bar{r}_{xy}(l) = \frac{1}{N}\sum_{k=0}^{N-1}\bar{R}_{xy}(k)e^{\frac{j2\pi kl}{N}}$$
$$= = \frac{1}{N}\sum_{k=0}^{N-1}X(k)Y^*(k)e^{\frac{j2\pi kl}{N}}$$

Hence above equ following by evaluating the IDFT at $\ell = 0$

The expression in above equ is the general form of parseval's theorem. In the special case where y(n)=x(n), reduces to $\sum_{n=0}^{N-1}|x(n)|^2 = \frac{1}{N}\sum_{n=0}^{N-1}|x(k)|^2$

Which express the energy in the finite duration sequence x(n) in terms of the frequency components {X(k)}.

Circular correlation. In general for complex valued sequence x(n) and y(n), if

$$x(n) \overset{DFT}{\underset{N}{\longleftrightarrow}} X(k)$$

And

$$y(n) \overset{DFT}{\underset{N}{\longleftrightarrow}} Y(k)$$

Then

$$\bar{r}_{xy}(\ell) \overset{DFT}{\underset{N}{\longleftrightarrow}} \bar{R}_{xy}(k) = X(k)Y^*(k)$$

Where $\bar{r}_{xy}(\ell)$ is the (un normalized) circular cross correlation sequence, defined as

$$\bar{r}_{xy}(\ell) = \sum_{n=0}^{N-1}x(n)y^*((n-\ell))N$$

Proof we can write $\bar{r}_{xy}(\ell)$ as the circular convolution of x(n) with $y^*(-n)$, that is

$$\bar{r}_{xy}(\ell) = x(\ell) \times y^*(-\ell)$$

Then with the aid of the properties in the N-point DFT of $\bar{r}_{xy}(\ell)$ is

$$\bar{R}_{xy}(k) = X(k)Y^*(k)$$

In the special case where y(n) = x(n), we have the corresponding expression for the circular auto correction of x(n), $\bar{r}_{xx}(\ell) \overset{DFT}{\underset{N}{\longleftrightarrow}} \tilde{R}_{xx}(k) = |X(k)|^2$

Circular convolution

If $\quad x_1(n) \xleftrightarrow[N]{DFT} X_1(k)$

And $\quad x_2(n) \xleftrightarrow[N]{DFT} X_2(k)$

Then $\quad x_1(n) \times x_2(n) \xleftrightarrow[N]{DFT} X_1(k) X_2(k)$

Where $x_1(n) \times x_2(n)$ denotes the circlar convolution of the sequence $x_1(n)$ and $x_2(n)$

Properties of the DFT

Property	Time Domain	Frequency Domain
Notation	$x(n), y(n)$	$X(k), Y(k)$
Periodicity	$x(n) = x(n+N)$	$X(k) = X(k+N)$
Linearity	$a_1 x_1(n) + a_2 x_2(n)$	$a_1 X_1(k) + a_2 X_2(k)$
Time reversal	$x(N-n)$	$X(N-k)$
Circular time shift	$x((n-l))_N$	$X(k) e^{-j2\pi kl/N}$
Circular frequency shift	$x(n) e^{j2\pi ln/N}$	$X((k-l))_N$
Complex conjugate	$x^*(n)$	$X^*(N-k)$
Circular convolution	$x_1(n) \circledN x_2(n)$	$X_1(k) X_2(k)$
Circular correlation	$x(n) \circledN y^*(-n)$	$X(k) Y^*(k)$
Multiplication of two sequences	$x_1(n) x_2(n)$	$\frac{1}{N} X_1(k) \circledN X_2(k)$
Parseval's theorem	$\sum_{n=0}^{N-1} x(n) y^*(n)$	$\frac{1}{N} \sum_{k=0}^{N-1} X(k) Y^*(k)$

UNIT-IV - DESIGN OF DIGITAL FILTER

PART A

1. Define IIR filter?

Filter having Infinite Impulse Response is known as IIR filter.

2. What are the various methods to design IIR filters? (May/June 2013)

Approximation of derivatives

Impulse invariance (method to digitize an analog transfer function)

Bilinear transformation (method to digitize an analog transfer function)

3. Which of the methods do you prefer for designing IIR filters? Why?

Bilinear transformation is best method to design IIR filter, since there is no aliasing in it.

4. What is the main problem of bilinear transformation?

Frequency warping or nonlinear relationship is the main problem of bilinear transformation.

5. What is prewarping? Why it is employed? (May/June 2014, 2012, Nov/Dec 2012)

Prewarping is the method to compensate non linear compression at high frequency. When the desired magnitude response is piecewise constant over frequency, this compression can be compensated by introducing a suitable prescaling or prewarping the critical frequencies.

6. State the frequency relationship in bilinear transformation?

$$\Omega = \frac{2}{T}\tan\frac{\omega}{2}$$

7. Where the jΩ axis of s-plane is mapped in z-plane in bilinear transformation?

The jΩ axis of s-plane is mapped on the unit circle in z-plane in bilinear transformation

8. Where left hand side and right hand side are mapped in z-plane in bilinear transformation?

Left hand side -- Inside unit circle

Right hand side – Outside unit circle

9. What is the frequency response of Butterworth filter?

Butterworth filter has monotonically reducing frequency response.

10. Which filter approximation has ripples in its response?

Chebyshev approximation has ripples in its pass band or stop band.

11. Can IIR filter be designed without analog filters?

Yes. IIR filter can be designed using pole-zero plot without analog filters

12. What is the advantage of designing IIR Filters using pole-zero plots?

The frequency response can be located exactly with the help of poles and zeros.

13. Compare the digital and analog filter.

Digital filter	Analog filter
i) Operates on digital samples of the signal.	i) Operates on analog signals.
ii) It is governed by linear difference equation.	ii) It is governed by linear difference equation.
iii) It consists of adders, multipliers and delays implemented in digital logic.	iii) It consists of electrical components like resistors, capacitors and inductors.
iv) In digital filters the filter coefficients are designed to satisfy the desired frequency response.	iv) In analog filters the hardware parameters has to be tuned to satisfy the desired frequency response.

14. What are the advantages and disadvantages of digital filters?

Advantages of digital filters:

- High thermal stability due to absence of resistors, inductors and capacitors
- Increasing the length of the registers can enhance the performance characteristics like accuracy, dynamic range, stability and tolerance
- The digital filters are programmable
- Multiplexing and adaptive filtering are possible

Disadvantages of digital filters:

- The bandwidth of the discrete signal is limited by the sampling frequency.

- The performance of the digital filter depends on the hardware used to implement the filter.

15. What is impulse invariant transformation?

The transformation of analog filter to digital filter without modifying the impulse response of the filter is called impulse invariant transformation.

16. How analog poles are mapped to digital poles in impulse invariant transformation?

In impulse invariant transformation the mapping of analog to digital poles are as follows,

The analog poles on the left half of s-plane are mapped into the interior of unit circle in z-plane.

The analog poles on the imaginary axis of s-plane are mapped into the unit circle in the z-plane.

The analog poles on the right half of s-plane are mapped into the exterior of unit circle in z-plane.

17. What is the importance of poles in filter design? (May/June 2012)

The stability of a filter is related to the location of the poles. For a stable analog filter the poles should lie on the left half of s-plane. For a stable digital filter the poles should lie inside the unit circle in the z-plane.

18. Why an impulse invariant transformation is not considered to be one-to-one?

In impulse invariant transformation any strip of width $2\pi/T$ in the s-plane for values of s-plane in the range $(2k-1)/T \leq \Omega \leq (2k-1)\ \pi/T$ is mapped into the entire z-plane. The left half of each strip in s-plane is mapped into the interior of unit circle in z-plane, right half of each strip in s-plane is mapped into the exterior of unit circle in z-plane and the imaginary axis of each strip in s-plane is mapped on the unit circle in z-plane. Hence the impulse invariant transformation is many-to-one.

19. What is Bilinear transformation?

The bilinear transformation is conformal mapping that transforms the s-plane to z-plane. In this mapping the imaginary axis of s-plane is mapped into the unit circle in z-plane, The left half of s-plane is mapped into interior of unit circle in z-plane and the right half of s-plane is mapped into exterior of unit circle in z-plane. The Bilinear mapping is a one-to-one mapping.

20. How the order of the filter affects the frequency response of Butterworth filter.

The magnitude response of butterworth filter approaches the ideal response as the order of the filter is increased.

21. Write the properties of Chebyshev type –1 filters. (Nov/Dec 2014) (May/June 2013)

- The magnitude response is equi-ripple in the passband and monotonic in the stopband
- The chebyshev type-1 filters are all pole designs
- The normalized magnitude function has a value of at the cutoff frequency ω_c.

- The magnitude response approaches the ideal response as the value of N increases.
- The poles of the chebyshev filter lie on an ellipse

22. Compare the Butterworth and Chebyshev Type-1 filters.

Butterworth	Chebyshev Type - 1
i. All pole design.	i. All pole design.
ii. The poles lie on a circle in s-plane.	ii. The poles lie on a ellipse in s-plane.
iii. The magnitude response is maximally flat at the origin and monotonically decreasing function of Ω.	iii. The magnitude response is equiripple in passband and monotonically decreasing in the stopband.
iv. The normalized magnitude response has a value of $1 / \sqrt{2}$ at the cutoff frequency Ω_c.	iv. The normalized magnitude response has a value of $1 / \sqrt{(1+\varepsilon^2)}$ at the cutoff frequency Ω_c.
v. Only few parameters has to be calculated to determine the transfer function.	v. A large number of parameters has to be calculated to determine the transfer function.

23. What is FIR filters?

The specifications of the desired filter will be given in terms of ideal frequency response $H_d(w)$. The impulse response $h_d(n)$ of the desired filter can be obtained by inverse fourier transform of $H_d(w)$, which consists of infinite samples. The filters designed by selecting finite number of samples of impulse response are called FIR filters.

24. What are the different types of filters based on impulse response?

Based on impulse response the filters are of two types 1. IIR filter 2. FIR filter

The IIR filters are of recursive type, whereby the present output sample depends on the present input, past input samples and output samples.

The FIR filters are of non recursive type, whereby the present output sample depends on the present input, and previous output samples.

25. What are the different types of filter based on frequency response?

The filters can be classified based on frequency response. They are I) Low pass filter ii) High pass filter iii) Band pass filter iv) Band reject filter.

26. Distinguish between FIR and IIR filters. (May June 2012)

S.No.	FIR filter	IIR filter
1.	These filters can be easily designed to have perfectly linear phase.	These filters do not have linear phase.
2.	FIR filters can be realized recursively and non-recursively.	IIR filters can be realized recursively.
3.	Greater flexibility to control the shape of their magnitude response.	Less flexibility usually limited to kind of filters.
4.	Errors due to roundoff noise are less severe in FIR filters, mainly because feedback is not used.	The roundoff noise in IIR filters are more.

27. What are the techniques of designing FIR filters?

There are three well-known methods for designing FIR filters with linear phase. These are 1) window method 2) Frequency sampling method 3) Optimal or min-max design.

28. State the condition for a digital filter to be causal and stable.

A digital filter is causal if its impulse response h(n) = 0 for n<0. A digital filter is stable if its impulse response is absolutely summable.

29. What is the reason that FIR filter is always stable?

FIR filter is always stable because it is an all zero filter which is finite and bounded. Hence it is inherently stable.

30. What are the properties of FIR filter?

FIR filter is always stable.

A realizable filter can always be obtained.

FIR filter has a linear phase response.

31. How phase distortion and delay distortions are introduced?

The phase distortion is introduced when the phase characteristics of a filter is not linear within the desired frequency band. The delay distortion is introduced when the delay is not constant within the desired frequency range.

32. Write the steps involved in FIR filter design.

Choose the desired (ideal) frequency response $H_d(w)$.

Take inverse fourier transform of $H_d(w)$ to get $h_d(n)$.

Convert the infinite duration $h_d(n)$ to finite duration h(n).

Take Z-transform of h(n) to get the transfer function H(z) of the FIR filter.

33. What are the advantages of FIR filters? (Nov/Dec 2014)

Linear phase FIR filter can be easily designed

Efficient realization of FIR filter exist as both recursive and nonrecursive structures

FIR filters realized non recursively are always stable

The round off noise can be made small in non recursive realization of FIR filters

34. What are the disadvantages of FIR filters?

The duration of impulse response should be large to realize sharp cutoff filters.

The non-integral delay can lead to problems in some signal processing applications.

35. What is the necessary and sufficient condition for the linear phase characteristic of an FIR filter?

The necessary and sufficient condition for the linear phase characteristic of an FIR filter is that the phase function should be a linear function of ω, which in turn requires constant phase and group delay.

36. What are the conditions to be satisfied for constant phase delay in linear phase FIR filters?

The conditions for constant phase delay are Phase delay, $\alpha = (N-1)/2$ (i.e., phase delay is constant) Impulse response, $h(n) = -h(N-1-n)$ (i.e., impulse response is antisymmetric)

37. How constant group delay & phase delay is achieved in linear phase FIR filters?

The following conditions have to be satisfied to achieve constant group delay & phase delay. Phase delay, $\alpha = (N-1)/2$ (i.e., phase delay is constant) Group delay, $\beta = \pi/2$ (i.e., group delay is constant) Impulse response, $h(n) = -h(N-1-n)$ (i.e., impulse response is antisymmetric)

38. What are the possible types of impulse response for linear phase FIR filters?

There are four types of impulse response for linear phase FIR filters

Symmetric impulse response when N is odd.

Symmetric impulse response when N is even.

Antisymmetric impulse response when N is odd.

Antisymmetric impulse response when N is even.

39. List the well-known design techniques of linear phase FIR filters.

There are three well-known design techniques of linear phase FIR filters. They are

Fourier series method and window method

Frequency sampling method.

Optimal filter design methods.

40. What is Gibb's phenomenon (or Gibb's Oscillation)? (May/June 2012)

In FIR filter design by Fourier series method the infinite duration impulse response is truncated to finite duration impulse response. The abrupt truncation of impulse response introduces oscillations in the passband and stopband. This effect is known as Gibb's phenomenon (or Gibb's Oscillation).

41. When cascade form realization is preferred in FIR filters?

The cascade form realization is preferred when complex zeros with absolute magnitude is less than one.

42. What are the desirable characteristics of the frequency response of window function?

The desirable characteristics of the frequency response of window function are The width of the main lobe should be small and it should contain as much of the total energy as possible. The side lobes should decrease in energy rapidly as ω tends to π.

43. Write the procedure for designing FIR filter using frequency-sampling method.

Choose the desired (ideal) frequency response $H_d(\omega)$.

Take N-samples of $H_d(\omega)$ to generate the sequence

Take inverse DFT of to get the impulse response h(n).

The transfer function H(z) of the filter is obtained by taking z-transform of impulse response.

44. What are the drawback in FIR filter design using windows and frequency sampling method? How it is overcome?

The FIR filter design using windows and frequency sampling method does not have Precise control over the critical frequencies such as w_p and w_s. This drawback can be overcome by designing FIR filter using Chebyshev approximation technique.In this technique an error function is used to approximate the ideal frequency response, in order to satisfy the desired specifications.

45. Write the characteristic features of rectangular window.

The mainlobe width is equal to 4π/N.

The maximum sidelobe magnitude is –13dB.

The sidelobe magnitude does not decrease significantly with increasing w.

46. List the features of FIR filter designed using rectangular window.

The width of the transition region is related to the width of the mainlobe of window spectrum. Gibb's oscillations are noticed in the passband and stopband. The attenuation in the stopband is constant and cannot be varied.

47. Why Gibb's oscillations are developed in rectangular window and how it can be eliminated or reduced?

The Gibb's oscillations in rectangular window are due to the sharp transitions from 1 to 0 at the edges of window sequence. These oscillations can be eliminated or reduced by replacing the sharp transition by gradual transition. This is the motivation for development of triangular and cosine windows.

48. List the characteristics of FIR filters, designed using windows.

The width of the transition band depends on the type of window.

The width of the transition band can be made narrow by increasing the value of N where N is the length of the window sequence. The attenuation in the stop band is fixed for a given window, except in case of Kaiser window where it is variable

49. Compare the rectangular window and hanning window.

Rectangular window	**Hanning Window**
i) The width of mainlobe in window spectrum is 4π/N	i)The width of mainlobe in window spectrum is 8π/N

ii) The maximum sidelobe magnitude in window spectrum is –13dB.	ii) The maximum sidelobe magnitude in window spectrum is –31dB.
iii) In window spectrum the sidelobe magnitude slightly decreases with increasing w.	iii) In window spectrum the sidelobe magnitude decreases with increasing w.
iv) In FIR filter designed using rectangular window the minimum stopband attenuation is 22dB.	iv) In FIR filter designed using hanning window the minimum stopband attenuation is 44dB.

50. Compare the rectangular window and hamming window.

Rectangular window	Hamming Window
i) The width of mainlobe in window spectrum is 4π/N	i)The width of mainlobe in window spectrum is 8π/N
ii) The maximum sidelobe magnitude in window spectrum is –13dB.	ii) The maximum sidelobe magnitude in window spectrum is –41dB.
iii) In window spectrum the sidelobe magnitude slightly decreases with increasing w.	iii) In window spectrum the sidelobe magnitude remains constant.
iv) In FIR filter designed using rectangular window the minimum stopband attenuation is 22dB.	iv) In FIR filter designed using hamming window the minimum stopband attenuation is 44dB.

51. Write the characteristic features of hanning window spectrum.

The mainlobe width is equal to 8π/N.

The maximum sidelobe magnitude is –41dB.

The sidelobe magnitude remains constant for increasing ω.

52. What is the mathematical problem involved in the design of window function?

(Nov/Dec 2012)

The mathematical problem involved in the design of window function(or sequence) is that of finding a time-limited function whose Fourier Transform best approximates a band limited function. The approximation should be such that the maximum energy is confined to mainlobe for a given peak sidelobe amplitude.

53. List the desirable features of Kaiser Window spectrum.

The width of the main lobe and the peak side lobe are variable. The parameter α in the Kaiser Window function is an independent variable that can be varied to control the side lobe levels with respect to main lobe peak. The width of the main lobe in the window spectrum can be varied by varying the length N of the window sequence

54. Compare the hamming window and Kaiser window.

Hamming Window	Kaiser Window
i)The width of mainlobe in window spectrum is 8π/N	i) The width of mainlobe in window spectrum depends on the values of α & N.
ii) The maximum sidelobe magnitude in window spectrum is –41dB. .	ii) The maximum sidelobe magnitude with respect to peak of mainlobe is variable using the parameter α.
iii) In window spectrum the sidelobe magnitude remains constant	iii) In window spectrum the sidelobe magnitude decreases with increasing w.
iv) In FIR filter designed using hamming window the minimum stopband attenuation is 44dB.	iv) In FIR filter designed using Kaiser window the minimum stopband attenuation is variable and depends on the value of α.

55. Give hamming window function (May/June 2014)

Hamming window function is given as

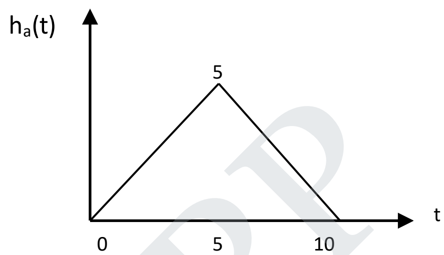
$$\omega_H(n) = 0.54 + 0.46\cos\left(\frac{2\pi n}{N-1}\right);\ \ n = -\left(\frac{N-1}{2}\right) \text{to} \left(\frac{N-1}{2}\right)$$

56. Is the given transfer function $H(z) = \frac{1+0.8z^{-1}}{1-0.9z^{-1}}$ **represents a low pass or high pass filter?**

(Nov/Dec 2013)

Given zero =0.8 and pole = -0.9. Since zero is close to (1,0) and pole is close to (-1,0), the given transfer function is a high pass filter.

57. The impulse response of an analog filter is given in below figure. Let $h(n)=h_a(nT)$ where T=1. Determine the system function. (Nov/Dec 2013)



Given h(n) = {0,1,2,3,4,5,4,3,2,1,0,…}

$$H(z) = \sum_{n=0}^{10} h(n)z^{-n}$$

$$H(z) = \begin{pmatrix} 0z^{-0} + z^{-1} + 2z^{-2} + 3z^{-2} + 4z^{-4} \\ +5z^{-5} + 4z^{-6} + 3z^{-7} + 2z^{-8} + z^{-9} \end{pmatrix}$$

58. What is the need for employing window for design of FIR filter (Nov/Dec 2012)

Unit sample response of the desired filter is obtained from frequency response $H_d(\omega)$. This unit sample response is normally infinite in length. Hence it is truncated to some finite length. This truncation creates oscillation in pass band and stop band of the filter. To avoid this problem a suitable windowing technique is used. The length of the window can be selected to desired value such that the ripples in the pass band and stop bands are reduced.

59. Comment on the pass band and stop band characteristic of Butterworth filter (Apr/May 2015)

The frequency response of the **Butterworth Filter** approximation function is also often referred to as “maximally flat” (no ripples) response because the pass band is designed to have a frequency response which is as flat as mathematically possible. However, one main disadvantage of the Butterworth filter is that it achieves this pass band flatness at the expense of a wide transition band as the filter changes from the pass band to the stop band. In practice however, Butterworth’s ideal frequency response is unattainable as it produces excessive passband ripple.

PART B

1. Design a Butterworth filter using the impulse invariance method for the following specifications:

$0.8 \le |H(e^{j\omega})| \le 1, \; 0 \le \omega \le 0.2\pi$

$|H(e^{j\omega})| \le 0.2, \; 0.6\,\pi \le \omega \le \pi$

Solution:

Given, $\dfrac{1}{\sqrt{1+\varepsilon^2}} = 0.8 \Rightarrow \varepsilon = 0.75;$

$$\frac{}{\sqrt{1+\lambda^2}} = 0.2 \Rightarrow \lambda = 4.899$$

$\omega_S = 0.6\pi\text{rad}; \; \omega_P = 0.2\pi\rho\alpha\delta; \text{ Assume } T = 1 \text{ sec}$

$$\frac{\omega_S}{\omega_P} = \frac{\Omega_s T}{\Omega_P T} = \frac{\Omega_S}{\Omega_P} = \frac{0.6\pi}{0.2\pi} = 3$$

$$N \ge \frac{\log\left(\frac{\lambda}{\varepsilon}\right)}{\log\left(\frac{\Omega_S}{\Omega_P}\right)} \ge \frac{\mathrm{LOG}\left(\frac{4.899}{0.75}\right)}{\log 3} \ge 1.71$$

$$N = 2$$

For N = 2 the transfer function of normalized Butterworth filter is,

$$H(s) = \frac{1}{s^2 + \sqrt{2s+1}}$$

$$\text{Obtauin cut – off frequency, } \Omega_c = \frac{\Omega_P}{(\varepsilon)^{1/N}} = \frac{0.2\pi}{(0.75)^{1/N}} = 0.231\pi$$

$$\begin{aligned} H_a(s) &= H(s)\big|_{s \to \frac{s}{\Omega_c}} \\ &= \frac{0.5266}{S^2 + 1.03S + 0.5266} \\ &= \frac{0.5266}{(s+0.51+j0.51)(s+0.51-j0.51)} \end{aligned}$$

Applying partial fractions,

$$H_a(s) = \frac{j0.516}{s-(-0.51-j0.51)} - \frac{j0.516}{s-(-0.51+j0.51)}$$

Using impulse invariance method.

If $H_a(s) = \sum_{k=1}^{N} \frac{c_k}{s-p_k}$, then $H(z) = \sum_{k=1}^{N} \frac{c_k}{1-e^{P_k T} z^{-1}}$

$$H(z) = \frac{j0.516}{1-e^{-0.51-j0.51}z^{-1}} - \frac{j0.516}{1-e^{-0.51-j0.51}z^{-1}}$$

$$= \frac{j0.516}{1-e^{-0.51}e^{-j0.51}z^{-1}} - \frac{j0.516}{1-e^{-0.51}e^{j0.51}z^{-1}}$$

$$= \frac{j0.516 - j0.516e^{-0.51}z^{-1} - j0.516 + j0.516e^{-0.51}e^{-j0.51}z^{-1}}{1+0.36e^{-j0.51}e^{j0.51}z^{-2} - 0.6e^{-j0.51}z^{-1} - 0.6e^{j0.51}z^{-1}}$$

$$= \frac{j0.3096[0.873+j0.488]z^{-1} - j0.3096[0.873-j0.488]z^{-1}}{1-0.6\left[e^{-j0.51}+e^{j0.51}\right]z^{-1}+0.36e^{-j0.51}e^{j0.51}z^{-2}}$$

$$\therefore H(z) = \frac{0.3022}{1-1.047z^{-1}+0.36z^{-2}}$$

2. Design a Butterworth filter using the bilinear transformation for the above problem

Solution:-

$\omega_s = 0.6\pi rad$; $\omega_p = 0.2\pi rad$. Assume T = 1 sec

Prewarping the frequencies we get,

$$\Omega_S = 2\tan\frac{\omega_s}{2} = 2.752; \Omega_p = 2\tan\frac{\omega_\infty}{2} = 0.6498$$

$$\frac{1}{\sqrt{1+\varepsilon^2}} = 0.8 \Rightarrow \varepsilon = 0.75; \quad \frac{1}{\sqrt{1+\lambda^2}} = 0.2 \Rightarrow \lambda = 4.899$$

$$N \geq \frac{\log\left(\frac{\lambda}{\varepsilon}\right)}{\log\left(\frac{1}{k}\right)} \geq \frac{\log\left(\frac{4.899}{0.75}\right)}{\log(4.235)} \geq 1.3$$

$$N = 2$$

For N = 2, the transfer function of normalized Butterworth filter is,

$$H(s) = \frac{1}{s^2+\sqrt{2s+1}}$$

Obtain cut off frequency, $\Omega_c = \frac{\Omega_p}{(\varepsilon)^{1/N}} = \frac{0.2\pi}{(0.75)^{1/2}} = 0.75 rad/sec$

$$H_a(s) = H(s)\big|_{s\to\frac{s}{\Omega_c}}$$
$$= \frac{0.5625}{s^2 + 1.06s + 0.5625}$$

For a bilinear transformatiuon,

$$H(z) = H_a(s)\big|_{S=\frac{2}{T}}\left(\frac{1-z^{-1}}{1+z^{-1}}\right); \text{Assume } T = 1$$

$$H(z) = \frac{0.5625\left(1+z^{-1}\right)^2}{4\left(1-z^{-1}\right)^2 + 2.12\left(1-z^{-2}\right) + 0.5625\left(1+z^{-1}\right)^2}$$
$$= \frac{0.084\left(1+z^{-1}\right)^2}{1-1.028z^{-1} + 0.3651z^{-2}}$$

3. Design a Chebyshev filter for the following specifications using (a) Bilinear transformation (b) impulse invariance method.

$\le|H(e^{j\omega})| \le 1,\ \ 0 \le \omega \le 0.2\pi$

$|H(e^{j\omega})| \le 0.2,\ 0.6\ \pi \le \omega \le \pi$

$$\le \left|H(e^{j\omega})\right| \le \omega \le 0.2\pi$$
$$\left|H(e^{j\omega})\right| \le 0.2, 0.6\pi \le \omega \le \pi$$

Solution;- (a). Using bilinear transformation

$$\omega_s = 0.6\pi\text{rad};\ \omega_p = 0.2\text{rad. Assume } T = 1\text{sec}$$

Prewarping the frequencies,

$$\Omega_S = 2\tan\frac{\omega_s}{2} = 2.752; \Omega_P = 2\tan\frac{\omega_p}{2} = 0.6498$$
$$\frac{1}{\sqrt{1+\varepsilon^2}} = 0.8 \Rightarrow \varepsilon = 0.75;\ \frac{1}{\sqrt{1+\lambda^2}} = 0.2 \Rightarrow \lambda = 4.899$$
$$\frac{1}{k} = \frac{\Omega_S}{\Omega_P} = 4.235$$
$$N \ge \frac{\cosh^{-1}\left(\frac{\lambda}{\varepsilon}\right)}{\cosh^{-1}\left(\frac{\Omega_S}{\Omega_P}\right)} = \frac{2.564}{2.122} = 1.208$$
$$N = 2$$

To find a & b:

$$\mu = \varepsilon^{-1} + \sqrt{\varepsilon^{-2} + 1} = 3$$

$$a = \Omega_p \frac{\left[\mu^{1/N} - \mu^{-1/N}\right]}{2} = 0.3752$$

$$b = \Omega_p \frac{\left[\mu^{1/N} + \mu^{-1/N}\right]}{2} = 0.75$$

To find the poles:

$$S_k = a\cos\phi_k + jb\sin\phi_k;\ k = 1, 2, \ldots N$$

Where, $\phi_k = \frac{\pi}{2} + \left[\frac{2k-1}{2N}\right]\pi;\ k = 1, 2 \ldots. N$

$$\phi_1 = \frac{\pi}{2} + \frac{\pi}{4} = \frac{3\pi}{4} = 132;$$

$$\phi_2 = \frac{\pi}{2} + \frac{3\pi}{4} = \frac{5\pi}{4} = 225$$

$$s_1 = -0.2653 + j0.53\ \&\ s_2 = -0.2653 - j0.53$$

Denominator of H(s):

$$(s + 0.2653 - j0.53)(s + 0.2653 + j0.53)$$

$$= (s + 0.2653)^2 + (0.53)^2$$

For even N, sub s =0 in the denominator polynomial & divide the value by $\sqrt{1+\varepsilon^2}$ which is the denominator of the transfer function.

Den. Polynomial: $\frac{(s+0.2653)^2 + (0.53)^2\big|_{s=0}}{\sqrt{1+\varepsilon^2}}$

Numerator of $H(s) = \frac{0.28}{s^2 + 0.5306s + 0.3516}$

Using bilinear transformation

$$H(z) = H(s)\Big|_{s=\frac{2}{T}\left(\frac{1-z^{-1}}{1+z^{-1}}\right)}$$

$$H(z) = \frac{0.28\left(1+z^{-1}\right)^2}{5.4128 - 7.298z^{-1} + 3.29z^{-2}} = \frac{0.052\left(1+z^{-1}\right)^2}{1 - 1.3480z^{-1} + 0.608z^{-2}}$$

(b) Impulse invariant method

$$\omega = \Omega t$$

$$\omega_p = \Omega_p T \quad \& \omega_s = \Omega_S T$$

For T = 1 sec

$$\frac{\omega_s}{\omega_p} = \frac{\Omega_S}{\Omega_P} = \frac{0.6\pi}{0.2\pi} = 3$$

$$N = \frac{\cos^{-1}(\lambda/\varepsilon)}{\cos^{-1}(1/k)} = \frac{\cos^{-1}\frac{4.899}{0.75}}{\cos^{-1}(3)} = 1.45 = 2$$

$$\mu = \varepsilon^{-1} + \sqrt{\varepsilon^2 + 1} = 3$$

$$a = \Omega_P\left[\frac{\mu^{1/N} - \mu^{-1/N}}{2}\right] = 0.3627$$

$$a = \Omega_P\left[\frac{\mu^{1/N} + \mu^{-1/N}}{2}\right] = 0.7255$$

$$\phi_1 = 135°,\ \phi_2 = 225$$

$$s_1 = -0.2564 + j0.513$$

$$s_2 = -0.2564 - j0.513$$

Denominator of $H(s) = s^2 + 0.513s + 0.33$

Numerator of $H(s) = 0.264$

$$H(s) = \frac{0.264}{s^2 + 0.513s + 0.33}$$

$$H(s) = \frac{0.263}{(s + 0.257 + j0.515)(s + 0.257 - j0.515)}$$

Applying partial fractions,

$$H(s) = \frac{j0.257}{(s + 0.257 + j0.515)} - \frac{j0.257}{(s + 0.257 - j0.515)}$$

If $H_a(s) = \sum_{k=1}^{N} \frac{c_k}{s - p_k}$ then $H(z) = \sum_{k=1}^{N} \frac{c_k}{1 - e^{p_k T} z^{-1}}$

$$P_1 = -0.257 - j0.515$$

$$P_2 = -0.257 + j0.515$$

$$H(z) = \frac{j0.257}{1 - e^{-0.257} e^{-j0.515} z^{-1}} - \frac{j0.257}{1 - e^{-0.257} e^{j0.515} z^{-1}}$$

$$= \frac{j0.257\left[1 - e^{-0.257} e^{j0.515} z^{-1}\right] - j0.257\left[1 - e^{-0.257} e^{-j0.515} z^{-1}\right]}{\left[1 - 0.77e^{-j0.52} z^{-1}\right]\left[1 - 0.77e^{j0.52} z^{-1}\right]}$$

$$= \frac{-j0.198z^{-1}\left[e^{j0.515} - e^{-j0.515}\right]}{1 + 0.5929z^{-2} - 1.34z^{-1}} = \frac{-j0.198z^{-1}\left[2j\sin 0.542\right]}{1 + 0.5929z^{-2} - 1.34z^{-1}}$$

$$H(z) = \frac{0.1967z^{-1}}{1 + 0.5929z^{-2} - 1.34z^{-1}}$$

4. Explain the design procedure of low pass Butterworth IIR filter. **(16)**

Solution: Designing IIR digital filter involves the design of equivalent analog filter & then converting analog filter to digital filter.

• First analog Butterworth IIR filter transfer function is determined using the specifications.

• Then, analog transfer function is converted to a digital filter transfer function using,

Impulse invariance transformation

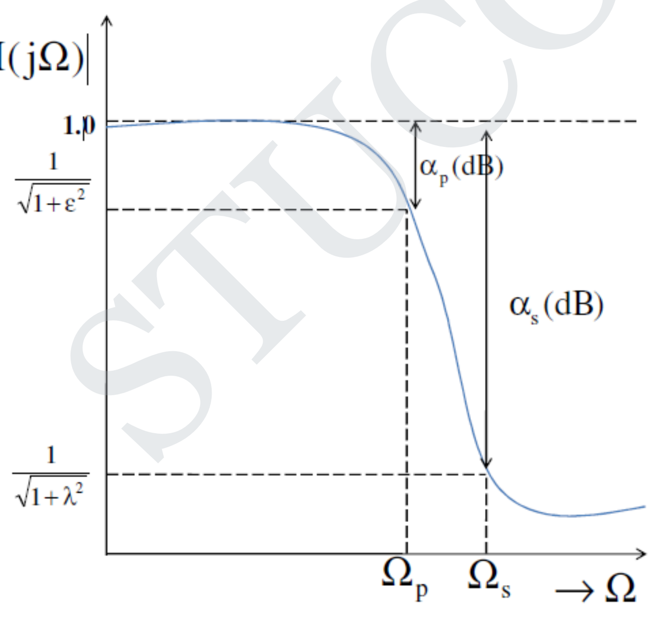
Bilinear transformation

Analog Butterworth filter:

The magnitude response of LPF is given by,

$$|H_a(\Omega)|^2 = \frac{1}{1+\left(\frac{\Omega}{\Omega_C}\right)^{2N}}$$

The magnitude response of the Butterworth filter is said to be maximally flat.



Find order of the filter,

$$N \geq \frac{\log\sqrt{\dfrac{10^{0.1\alpha_s}-1}{10^{0.1\alpha}-1}}}{\log\left(\dfrac{\Omega_S}{\Omega_p}\right)} \geq \frac{\log\left(\dfrac{\lambda}{\varepsilon}\right)}{\log\left(\dfrac{1}{k}\right)}$$

Where ε – parameter specifying allowable passband

λ - parameter specifying allowable stopband

$$\frac{1}{\sqrt{1+\varepsilon^2}} = \alpha_p \qquad \frac{1}{\sqrt{1+\lambda^2}} = \alpha_s$$

Obtain cut – off frequency,

$$\Omega_c = \frac{\Omega_p}{\left(10^{0.1\alpha_p}-1\right)^{1/2N}} = \frac{\Omega_p}{(\varepsilon)^{1/2N}}$$

With the order of the filter, obtain the transfer function H(s), by substituting s by $\frac{s}{\Omega_c}$

Convert analog transfer function $H_a(s)$ to digital function H(z), using either impulse invariance method or bilinear transformation method.

To obtain the poles which h lies on LHS of the s – plane

$$s_k = e^{j\phi k};\ \text{where } \phi_k = \frac{\pi}{2} + \frac{(2k-1)\pi}{2N}$$

When N = 1; k =1

$$s_1 = e^{j\phi_1}:$$
$$s_1 = e^{j\pi} = -1$$

Denominator: $\Rightarrow [s-(-1)] = (s+1)$

List of Butter worth polynomials:

N	Denominator of H(s)
1	(s+1)
2	$s^2+\sqrt{2}s+1$
3	$(s+1)(s^2+s+1)$
4	$(s^2+0.76537s+1)(s^2+1.8477s+1)$

5. Convert the analog transfer function $H_a(s)=\dfrac{2}{(s+1)(s+2)}$ to digital IIR filter using impulse invariance method and bilinear transformation method. Assume T = 1 sec. (Nov/Dec 2013) (16)

Solution:-

Impulse invariance method

$$H_a(s)=\frac{2}{(s+1)(s+2)}$$

Applying partial fractions,

$$H(s)=\frac{2}{(s+1)(s+2)}$$
$$=\frac{2}{\left(s-(-1)\right)}-\frac{2}{\left(s-(-2)\right)}$$

$$H_s(s)=\sum_{k=1}^{N}\frac{c_k}{P_k}\Rightarrow H(z)=\sum_{k=1}^{N}\frac{c_k}{1-e^{p_kT}z^{-1}}$$

$$H(s)=\frac{2}{\left(s-(-1)\right)}-\frac{2}{\left(s-(-2)\right)}$$

$$H(z)=\frac{2}{1-e^{-1}z^{-1}}-\frac{2}{1-e^{-2}z^{-1}}$$
$$=\frac{2}{1-0.3678z^{-1}+0.1353z^{-1}}$$

Simplifying

$$H(z)=\frac{0.465z^{-1}}{1-0.503z^{-1}+0.0497z^{-2}}$$

Bilinear transformation method

Substitute,

$$s=\frac{2}{T}\left[\frac{1-z^{-1}}{1+z^{-1}}\right]$$

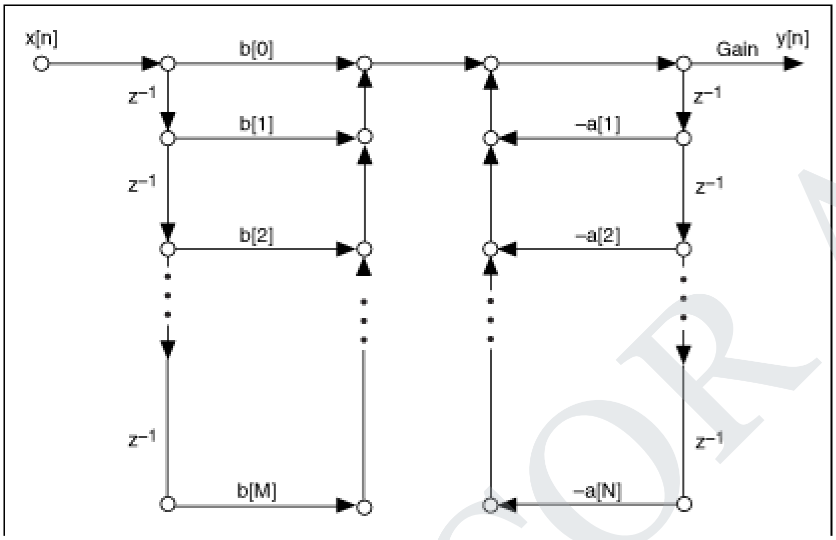
$$H(z)=\left.\frac{2}{(s+1)(s+2)}\right|_{s=\frac{2}{T}\left[\frac{1-z^{-1}}{1+z^{-1}}\right]}$$

$$H(z)=\frac{2}{\left(2\left[\frac{1-z^{-1}}{1+z^{-1}}\right]+1\right)\left(2\left[\frac{1-z^{-1}}{1+z^{-1}}\right]+2\right)}$$

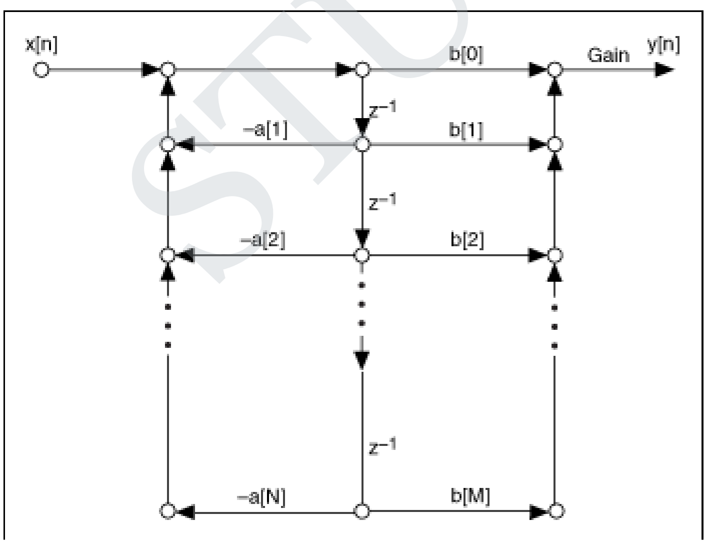
$$=\frac{\left(1+z^{-1}\right)^2}{6\left(1-\frac{1}{3}z^{-1}\right)}$$

6. Draw the structures of IIR filters

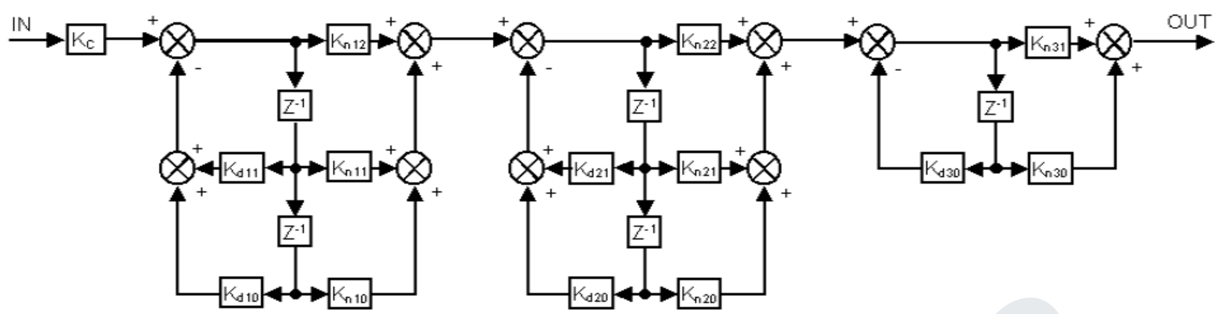
Direct form I



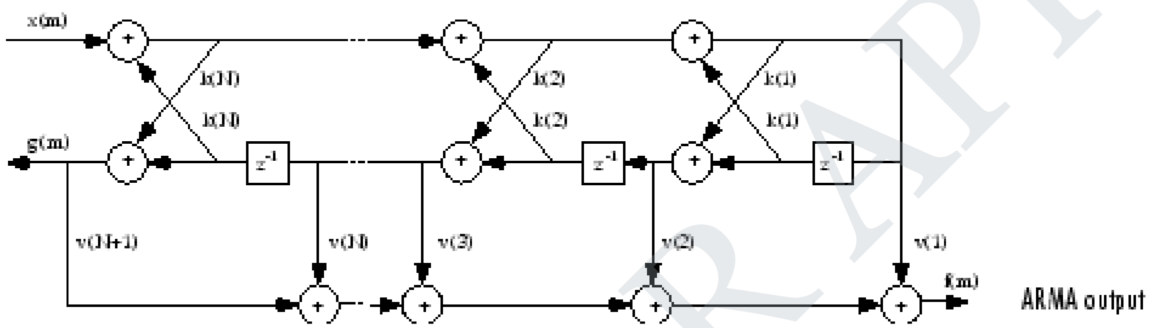
Direct form II



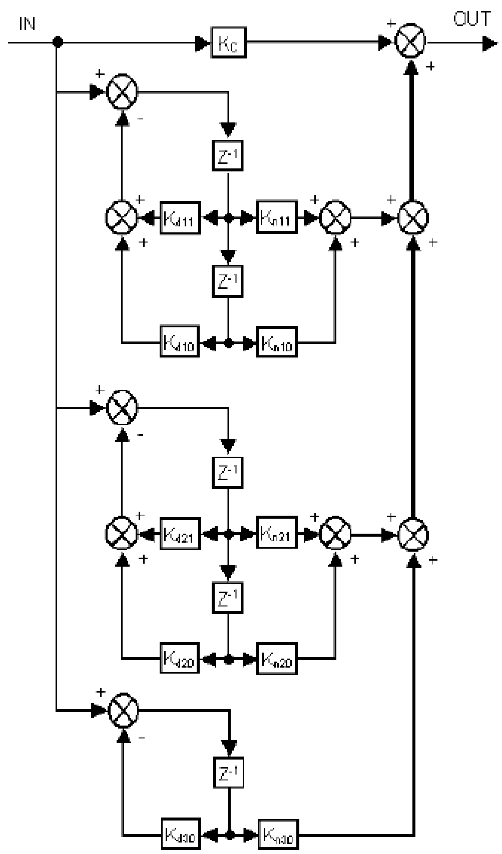
Cascade structure



Lattice Ladder structure



Parallel structure



7. Convert the analog filter with system function $H_a(s)$ into digital filter using impulse invariance method

$$H_a(s) = \frac{s+0.1}{(s+0.1)^2+9} \qquad \textbf{(16)}$$

Solution:- We note that the anolog filter has a zero at s = -0.1 and a point of complex conjugate poles at

$$p_k = -0.1 \pm j3$$

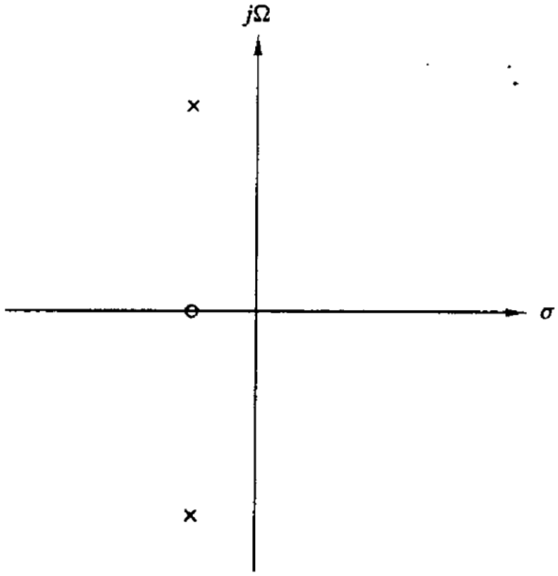
We do not have to determine the impulse response $h_a(t)$ in order to design the digital IIR filter based on the method of impulse invariant. Instead, we directly determine H(z), as given by, from the partial – fraction expansion of Ha(s).

Thus we have

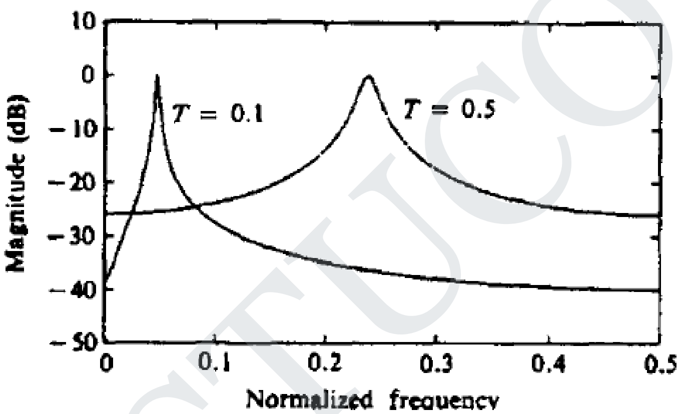
$$`H(s) = \frac{\frac{1}{2}}{s+0.1-j3} + \frac{\frac{1}{2}}{s+0.1+j3}$$

Then

$$H(z) = \frac{\frac{1}{2}}{1-e^{-0.1}e^{j3T}} + \frac{\frac{1}{2}}{1-e^{-0.1T}e^{-j3T}z^{-1}}$$



Pole-zero locations for analog filter

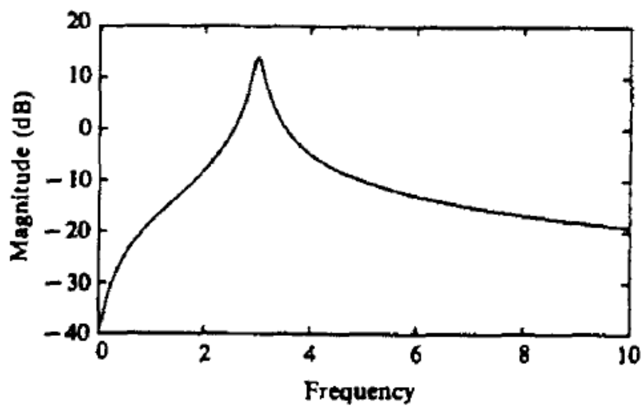


Frequency response of digital filter

Since the two poles are complex conjugates, we cab combine them to form a single two – pole filter with system function

$$H(z) = \frac{1\left(e^{-0.1T}\cos 3T\right)z^{-1}}{1-\left(2e^{-0.1T}\cos 3T\right)z^{-1}+e^{-0.2T}z^{-1}}$$

The magnitude of the frequency response characteristics of this filter is plotted in fig. 8.34 for T = 0.1 and T = 0.5. For purpose of comparison, we have also plotted that aliasing is significantly more prevalent when T = 0.5 than when T = 0.1. also note the shift of the resonant frequency as T changes.



Frequency response of analog filter

The preceding example illustrates the importance of selecting a small value for T to minimize the effect of aliasing. Due to the presence of aliasing, the impulse invariant method is appropriate for the design of low pass and bandpass filters only.

8. The system function of the analog filter is given as, (16)

$H_a(s) = \dfrac{s+0.1}{(s+0.1)^2+16}$ obtain the system function of the digital filter using bilinear transformation which is resonant at $\omega_r = \dfrac{\pi}{2}$.

Solution:- First, we note that the anolog filter has a resonant frequency $\Omega_R = 4$. This frequency is to be, mapped into $\omega_r = \dfrac{\pi}{2}$ by selecting the value of the have parameter T. from the relationship in, we must select T = ½ in order to have $\omega_r = \dfrac{\pi}{2}$. Thus the desired mapping is

$$s = 4\left(\frac{1-z^{-1}}{1+z^{-1}}\right)$$

The resulting digital filter has the system function

$$H(z) = \frac{0.128+0.006z^{-1}-0.122z^{-2}}{1+0.0006z^{-1}+0.975z^{-2}}$$

We note that the coefficient of the z^{-1} term in the denominator of H(z) is extremely small and can be approximated by zero. Thus we have the system function.

$$H(z) = \frac{0.128+0.006z^{-1}-0.122z^{-2}}{1+0.975z^{-2}}$$

This filter has poles at

$$p_{1.2} = 0.987e^{\pm\pi/2}$$

And zeros at

$$Z_{1.2} = -1.095$$

Therefore, we have succeeded in designing a two- pole filter that resonates near $\omega = \pi / 2$

In this example the parameter T was selected to map the resonant frequency of the analog filter into the desired resonant frequency of the digital filter. Usually the design of the digital filter begins with specification in the digital domain, which involve the frequency variable ω. These specifications in frequency are converted to the analog domain by means of the relation in (8.3.43). The analog filter is then designed that meets these specifications and converted to a digital filter by means of the bilinear transformation in (8.3.401). In this procedure, the parameter T is transparent and may be set to any arbitrary value (e.g, T = 1). The following example illustrates this point.

9. The system function of the first order low pass butterworth filter is given as (16)

$$H_a(s) = \frac{\Omega_C}{S + \Omega_C}$$

Here Ω_c is the 3dB cutoff frequency of the analog filter. Apply bilinear transformation to this filter such that the digital filter will have 3dB frequency of 0.2π.

Solution:- The digital filter is specified to have its -3dB gain at $\omega_c = 0.2\pi$. In the frequency domain of the analog filter $\omega_c = 0.2\pi$ corresponds to

$$\begin{aligned}\Omega_C &= \frac{2}{T}\tan 0.1\pi \\ &= \frac{0.65}{T}\end{aligned}$$

Thus the analog filter has the system function

$$H(s) = \frac{0.65/T}{s + 0.65/T}$$

This represents our filter design in the analog domain.

Now, we apply the bilinear transformation given by (8.3.40) to convert the analog filter into the desired digital filter. Thus we obtain

$$H(z) = \frac{0.245\left(1 + z^{-1}\right)}{1 - 0.509z^{-1}}$$

Where the parameter T has been divided out.

The frequency response of the digital filter is

$$H(\omega)=\frac{0.245\left(1+e^{-j\omega}\right)}{1-0.509e^{-j\omega}}$$

At $\omega=0 H(0)=1, \& \omega=0.2\pi, |H(0.2\pi)|=0.707,$ Which is the desired response.

10. The desired frequency response of a low pass filter is **(16)**

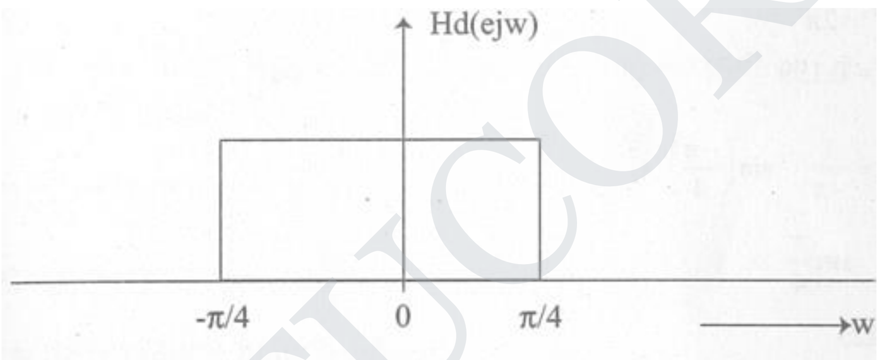
$$H_a(e^{j\omega})=e^{-j2\omega}, -\pi/4\omega \le \pi/4$$
$$0 \quad ,\pi/4<|\omega|\le\pi$$

Determine $H_d(n)$. Also determine h(n) using rectangular window with N= 5. Determine the frequency response $H(e^{j\omega})$ of the designed filter.

Solution:

Given

$$H_a(e^{j\omega})=e^{-j2\omega}, -\pi/4\omega \le \pi/4$$
$$0 \quad ,\pi/4<|\omega|\le\pi$$



Step 1:

To find hd(n)

$$h_0(n) = \frac{1}{2\pi}\int_{-\pi}^{\pi} Hd(e^{jw})e^{jwn}dw$$

$$= \frac{1}{2\pi}\int_{-\pi/4}^{\pi/4} e^{-2jw}dw$$

$$= \frac{1}{2\pi}\int_{-\pi/4}^{\pi/4} e^{jw(n-2)}dw$$

$$= \frac{1}{2\pi}\frac{1}{(n-2)j}\left[\frac{e^{j(n-2)\frac{\pi}{4}} - e^{j(n-2)\frac{-\pi}{4}}}{2j}\right]$$

$$= \frac{1}{\pi(n-2)}\left[\frac{e^{j(n-2)\frac{\pi}{4}} - e^{-j(n-2)\frac{\pi}{4}}}{2j}\right]$$

$$h_0(n) = \frac{1}{\pi(n-2)}\sin(n-2)\frac{\pi}{4}$$

When n = 0

$$h_D(0) = \frac{1}{-2\pi}\sin\left(\frac{-2\pi}{4}\right)$$

$$= \frac{-1}{-2\pi}\sin\left(\frac{\pi}{2}\right)$$

$$h_D(0) = \frac{1}{2\pi} = 0.159$$

$$h_D(0) = 0.159$$

When n= 1

$$h_D(1) = \frac{1}{-\pi}\sin\left(\frac{-\pi}{4}\right)$$

$$= \frac{1}{\pi}\sin\frac{\pi}{4}$$

$$h_D(1) = 0.225$$

When n= 2

$$h_D(2) = \frac{1}{\pi(n-2)}\sin(n-2)\frac{\pi}{4}$$

Using ‘L’ hospital rule

$$\frac{dh_D(n)}{dn} = \frac{\cos\left[(n-2)\frac{\pi}{4}\right]\frac{\pi}{4}}{\pi}$$

$$= \frac{\cos(0)\frac{\pi}{4}}{\pi} = 0.25$$

$$h_D(2) = 0.25$$

Since FIR filter impulse response is response is symmetric in nature

$$h(n) = h(N-1-n)$$
$$h_D(3) = h_D(5-1-3) = h_D(1)$$
$$h_D(4) = h_D(5-1-4) = h_D(0)$$
$$h_D(3) = 0.225$$
$$h_D(4) = 0.159$$

Given window is rectangular

$$w(n) = 1 \qquad 0 \le n \le 4$$
$$0 \qquad \text{otherwise}$$

Step 2: Top find the coefficients:

$$h(n) = hd(n)w(n)$$
$$h(0) = hd(0)w(0) = 0.159(1) = 0.159$$
$$h(1) = hd(1)w(1) = 0.225$$
$$h(2) = hd(2)w(2) = 0.25$$
$$h(3) = hd(3)w(3) = 0.225$$
$$h(4) = hd(4)w(4) = 0.159$$

Filter coefficients:

$$h(0) = 0.159,\ h(1) = 0.225,\ h(2) = 0.25,\ \ h(3) = 0.225,\ \ h(4) = 0.159$$

To find the system function:

$$H(z) = \sum_{n=0}^{N-1} h(n)z^{-n}$$

$$H(z) = \sum_{n=0}^{4} h(n)z^{-n}$$

$$= h(0) + h(1)z^{-1} + h(2)z^{-2} + h(3)z^{-3} + h(4)z^{-4}$$
$$= h(0)\left[1+z^{-4}\right] + h(1)\left[z^{-1}+z^{-3}\right] + h(2)z^{-2}$$
$$H(z) = 0.159\left[1+z^{-4}\right] + 0.225\left[z^{-1}+z^{-3}\right] + 0.25z^{-2}$$

To find the frequency response:

$$H(e^{jw}) = H(Z = e^{jw})$$

$$H(e^{jw}) = 0.159\left[1+e^{-j4w}\right]+0.225\left[e^{-jw}+e^{-j3w}\right]+0.25e^{-2jw}$$

$$= e^{-2jw}\left[0.25+0.159\left(e^{2jw}+e^{-2jw}\right)+0.225\left(e^{jw}+e^{-jw}\right)\right]$$

$$H(e^{jw}) = e^{-2jw}\left[0.25+0.318\cos 2w+0.45\cos w\right]$$

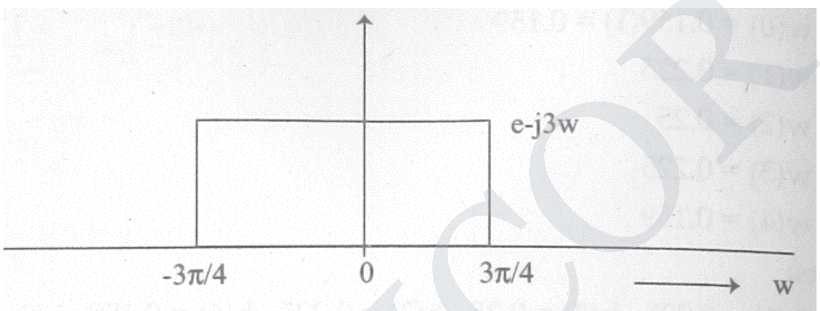
11. Design a low pass filter with desired frequency response **(16)**

$$H_d(e^{j\omega}) = e^{-j3\omega} \text{ for } -3\pi/4 \le \omega \le \pi/4$$
$$= 0 \text{ for } 3\pi/4 \le |\omega| \le \pi$$

Use Hamming window for N = 7.

Solution:-

$$H_d(e^{j\omega}) = e^{-j3\omega} \text{ for } -3\pi/4 \le \omega \le \pi/4$$
$$= 0 \text{ for } 3\pi/4 \le |\omega| \le \pi$$



Given N = 7

$$Hd(e^{jw}) = e^{-j3w} \frac{-3\pi}{4} \le w \le \frac{3\pi}{4}$$

Step 1:

To find hd(n)

$$hd(n) = \frac{1}{2\pi}\int_{-\pi}^{\pi} Hd\left(e^{jw}\right)e^{jwn}dw$$

$$= \frac{1}{2\pi}\int_{-\pi}^{\frac{3\pi}{4}} e^{-j3w}e^{jwn}dwe$$

$$= \frac{1}{2\pi}\int_{\frac{-3\pi}{4}}^{\frac{3\pi}{4}} e^{jw}\left(n-3\right)dw$$

$$= \frac{1}{\pi 2j\left(n-3\right)}\left[\frac{e^{j(n-3)}}{j\left(n-3\right)}\right]_{\frac{-3\pi}{4}}^{\frac{3\pi}{4}}$$

$$= \frac{1}{\pi 2j(n-3)}\left[e^{j(n-3)\frac{3\pi}{4}} - e^{-j(n-3)\frac{3\pi}{4}}\right]$$

$$hd(n) = \frac{1}{\pi(n-3)}\sin\left[(n-3)\frac{3\pi}{4}\right]$$

N = 7 n = 0, 1, 2, 3, 4, 5, 6, 7 …

$$hd(0) = \frac{1}{-3\pi}\sin\left[\frac{-9\pi}{4}\right]$$

$$= \frac{1}{3\pi}\sin\left[\frac{9\pi}{4}\right]$$

$$hd(0) = 0.0750$$

$$hd(1) = \frac{1}{2\pi}\sin\left[\frac{3\pi}{2}\right]$$

$$hd(1) = -0.159$$

$$hd(2) = \frac{1}{\pi}\sin\left[\frac{3\pi}{4}\right]$$

$$hd(2) = 0.225$$

When n = 3

$$\frac{dhd(n)}{d(n)} = \frac{\cos\left[(n-3)\frac{3\pi}{4}\right]\frac{3\pi}{4}}{\pi}$$

$$= \frac{\cos 0\frac{3\pi}{4}}{\pi}$$

$$hd(3) = 0.75$$

$$hd(4) = 0.225$$

$$hd(5) = hd(1) = -0.159$$

$$hd(6) = hd(0) = 0.0750$$

Step 2: To find window coefficients Hamming

$$w(n) = 0.54 - 0.46\cos\frac{2\pi n}{N-1}$$
$$w(n) = 0.54 - 0.46\cos\left(\frac{2\pi n}{6}\right)$$
$$w(0) = 0.54 - 0.46 = 0.08$$
$$w(1) = 0.54 - 0.46\cos\left(\frac{2\pi}{6}\right) = 0.31$$
$$w(2) = 0.54 - 0.46\cos\left(\frac{4\pi}{6}\right) = 0.77$$
$$w(3) = 0.54 - 0.46\,\mathrm{cso}\left(\frac{6\pi}{6}\right) = 1$$
$$w(4) = w(2) = 0.77$$
$$w(5) = w(1) = 0.31$$
$$w(6) = w(0) = 0.08$$

Step 3: To find the filter coefficients

$$h(n) = hd(n)\ w(n)$$
$$h(0) = hd(0)\ \ w(0) = 0.0750 \times 0.08 = 0.006$$
$$h(1) = hd(1)\ w(1) = -0.159 \times 0.31 = -0.0494$$
$$h(2) = hd(2)\ \ w(2) = 0.225 \times 0.77 = 0.1733$$
$$h(3) = hd(3)\ \ w(3) = 0.75 \times 1 = 0.75$$
$$h(4) = hd(4)\ \ w(4) = h(2) = 0.1733$$
$$h(5) = hd(1) = -0.0494$$
$$h(6) = hd(0) = 0.006$$

Filter coefficients

$$h(n) = 0.006, -0.0494,\ 0.1733, 0.75, 0.1733, -0.0494, 0.006$$

Step 4: To find the system response

$$H(z) = \sum_{n=0}^{N-1} h(n)$$
$$= \sum_{n=0}^{6} h(n)z^{-n}$$

$$= h(0) + h(1)z^{-1} + h(2)z^{-2} + h(3)z^{-3} + h(4)z^{-4} + h(5)z^{-5} + h(6)z^{-6}$$
$$= h(0)(1+z^{-6}) + h(1)\left(z^{-1}+z^{-5}\right) + h(2)\left(z^{-2}+z^{-4}\right) + h(3)\left(z^{-3}+z^{-4}\right)$$
$$H(z) = 0.006\left(1+z^{-6}\right) - 0.0494\left(z^{-1}+z^{-5}\right) + 0.1733\left(z^{-2}+z^{-4}\right) + 0.75z$$

12. Design an ideal HPF with $H_d\left(e^{jw}\right) = 1 \text{ for } \pi/4 \le \omega \le \pi$, $= 0 \text{ for } |\omega| \le \pi/4$ **(May/June 2012)** **(16)**

Using Hamming window for N = 11

Solution;-

Given the desired frequency response

$$Hd(e^{jw}) = 1 \qquad -\frac{\pi}{4} \le |w| \le \frac{\pi}{4}$$

$$hd(n) = \frac{1}{2\pi}\int_{-\pi}^{\pi} Hd\left(e^{jw}\right)e^{jwn}dw$$

$$= \frac{1}{2\pi}\left[\int_{\frac{-\pi}{4}}^{\frac{-\pi}{4}} 1e^{jwn}dw + \int_{\frac{\pi}{4}}^{\frac{\pi}{4}} e^{jwn}dw\right]$$

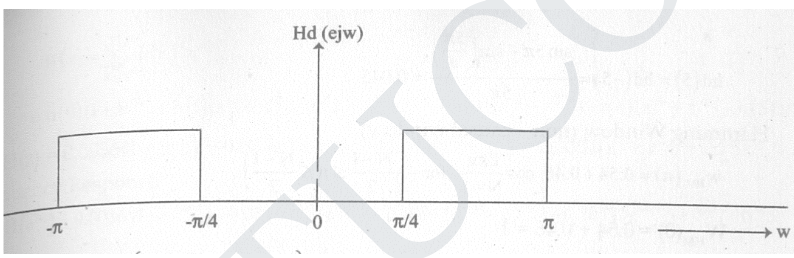
Solution:

Given the desired frequency response

$$Hd\left(e^{jw}\right) = 1 \quad -\frac{\pi}{4} \le |w| \le \frac{\pi}{4}$$

$$hd(n) = \frac{1}{2\pi}\int_{-\pi}^{\pi} Hd\left(e^{jw}\right) e^{jwn}\, dw$$

$$= \frac{1}{2\pi}\left[\int_{\frac{-\pi}{4}}^{\frac{-\pi}{4}} 1\ \ e^{jwn}dw + \int_{\frac{\pi}{4}}^{\pi} e^{jwn}\ dw\right]$$



$$hd(n) = \frac{1}{2\pi}\left\{\left[\frac{e^{jwn}}{j^{n}}\right]_{\frac{\pi}{4}}^{\frac{-\pi}{4}} + \left[\frac{e^{jwn}}{j^{n}}\right]_{\frac{\pi}{4}}^{\pi}\right\}$$

$$= \frac{1}{2\pi jn}\left[\left(e^{j\pi n} - e^{-j\pi n}\right) - \left(e^{j\frac{n\pi}{4}} - e^{-j\frac{n\pi}{4}}\right)\right]$$

$$hd(n) = \frac{1}{n\pi}\left[\sin\ n\pi - \sin\frac{n\pi}{4}\right]$$

From the given frequency response $\tau = 0$ hence the filter coefficients are symmetrica about n = 0 satisfying the condition h(n) h(-n)

To find hd(0)

$$\frac{dhd(n)}{dn}=\frac{\cos n\pi(\pi)-\cos\left(\frac{n\pi}{4}\right)\frac{p}{4}}{\pi}$$
$$=\cos n\pi-\cos\left(\frac{n\pi}{4}\right)\frac{1}{4}$$

Hd(0) = 0.75

$$hd(1)=hd(-1)=\frac{\sin\pi-\sin\frac{\pi}{4}}{\pi}=-0.225$$
$$hd(2)=hd(-2)=\frac{\sin 2\pi-\sin\frac{\pi}{2}}{2\pi}=-0.159$$
$$hd(3)=hd(-3)=\frac{\sin 3\pi-\sin\frac{3\pi}{2}}{3\pi}=-0.075$$
$$hd(4)=hd(-4)=\frac{\sin 4\pi-\sin\pi}{4\pi}=0$$
$$hd(5)=hd(-5)=\frac{\sin 5\pi-\sin\frac{5\pi}{2}}{5\pi}=0.045$$

Hamming window (non – casual window)

$$w_{HM}(n)=0.54+0.46\cos\frac{2\pi n}{N-1}\text{for}\left(\frac{N-1}{2}\le n\le\frac{N-1}{2}\right)$$
$$W_{HM}(0)=0.54+0.46=1$$
$$W_{HM}(-1)=W_{HM}(1)=0.54+0.46\cos\frac{\pi}{5}=0.912$$
$$W_{HM}(-2)=W_{HM}(2)=0.54+0.46\cos\frac{2\pi}{5}=0.682$$
$$W_{HM}(-3)=W_{HM}(3)=0.54+0.46\cos\frac{3\pi}{5}=0.398$$
$$W_{HM}(-4)=W_{HM}(4)=0.54+0.46\cos\frac{4\pi}{5}=0.1678$$
$$W_{HM}(-5)=W_{HM}(5)=0.54+0.46\cos\pi=0.08$$

Filter coefficients

$$h(n)=hd(n)W_M(m)-5\le n\le 5$$
$$h(0)=hd(0)W_{HM}(0)=1.075-0.75$$
$$h(-1)=hd(1)W_{HM}(1)=(-0.225)(0.912)=-0.205$$
$$h(-2)=hd(2)W_{HM}(2)=(-0.159)(0.682)=-0.1084$$
$$h(-3)=hd(3)W_{HM}(3)=(-0.075)(0.398)=-0.03$$
$$h(-4)=hd(4)W_{HM}(4)=0(0.1678)=0$$
$$h(-5)=hd(5)W_{HM}(5)=(-0.045)(0.08)=0.0036$$

N	-5	-4	-3	-2	-1	0	1	2	3	4	5
$H_d(e^{jw})$db	0.0036	0	-0.03	-0.108	-0.205	0.75	-0.205	0.108	-0.03	0	0.0036

Table: Filter coefficient

$$h(0) = h(10) = 0.0036$$
$$h(1) = h(9) = 0$$
$$h(2) = h(8) = -0.03$$
$$h(3) = h(7) = -0.1084$$
$$h(5) = 0.75$$

To find the system function

$$H(z) = \sum_{n=0}^{10} h(n)z^{-n}$$

$$= h(0)\left(1+z^{-10}\right)+h(1)\left(z^{-1}+z^{-9}\right)+h(2)\left(z^{-2}+z^{-8}\right)+h(3)\left(z^{-3}+z^{-7}\right)$$
$$+h(4)\left(z^{-4}+z^{-6}\right)+h(5)z^{-5}$$

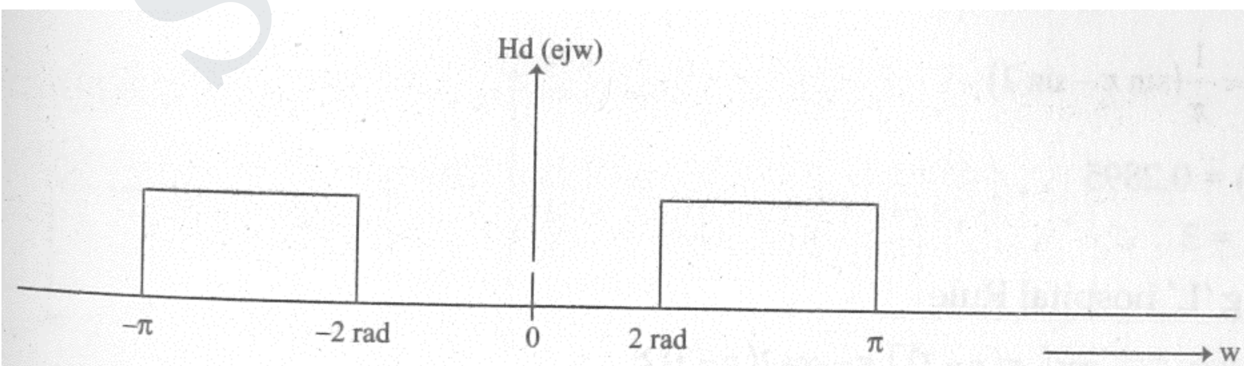
$$H(z) = 0.0036\left(1+z^{-10}\right)-0.03\left(z^{-2}+z^{-8}\right)-0.1084\left(z^{-3}+z^{-7}\right)-0.2502\left(z^{-4}+z^{-6}\right)+0.75z^{-5}$$

Frequency response:

$$H(e^{jw}) = 0.0036\left(1+e^{-j10w}\right)-0.03\left[e^{-2jw}+e^{-8jw}\right]-0.1084\left[e^{-3jw}+e^{-7jw}\right]-0.2052\left[e^{-4jw}+e^{-6jw}\right]+0.75e^{-5jw}$$
$$= e^{-5jw}\left[0.75+0.0036\left(e^{5jw}+e^{5jw}\right)-0.03\left(e^{+3jw}+e^{-3jw}\right)-0.1084\left(e^{-2jw}+e^{-2jw}\right)-0.2052\left(e^{jw}+e^{jw}\right)\right]$$
$$H(e^{jw}) = e^{-5jw}\left[0.75+0.0072\cos 5w-0.06\cos 3w-0.21684\cos 2w-0.4104\cos w\right]$$

13. Design a high pass filter of length 7 with cutoff frequency of 2 rad/sec using Hamming window. Determine its magnitude and phase response. (Nov/Dec 2007) (16)

Solution:



Given Wc = 2 rad/sec

N =7

$$Hd(n)=\frac{1}{2\pi}\left[\int_{\pi}^{-2} e^{-3jw}e^{jwn}dw+\int_{\frac{\pi}{4}}^{\pi} e^{-3jw}e^{jwn}dw\right]$$

$$hd(n)=\frac{1}{2\pi}\left\{\left[\frac{e^{jw}(n-3)}{j(n-3)}\right]_{-\pi}^{-2}+\left[\frac{e^{jw}(n-3)}{j(n-3)}\right]_{-2}^{\pi}\right\}dw$$

$$=\frac{1}{2\pi j(n-3)}\left\{\left[e^{j-2(n-3)}-e^{-j\pi(n-3)}\right]+\left[e^{j\pi(n-3)}-e^{j2(n-3)}\right]\right\}$$

$$=\frac{1}{\pi(n-3)}\left\{\frac{e^{j\pi(n-3)}-e^{-j\pi(n-3)}}{2j}-\frac{e^{2j(n-3)}-e^{-2j(n-3)}}{2j}\right\}$$

$$hd(n)=\frac{1}{\pi(n-3)}\left[\sin\pi(n-3)-\sin 2(n-3)\right]$$

$$hd(0)=\frac{1}{-3\pi}\left[\sin(-3\pi)-\sin(-6)\right]$$

$$=\frac{1}{3\pi}(\sin 3\pi-\sin 6)$$

$$=\frac{1}{3\pi}(0-(-0.279))$$

$$hd(0)=0.0296$$

$$hd(1)=\frac{1}{-2\pi}\left[\sin(-2\pi)-\sin 2(-2)\right]$$

$$=\frac{1}{2\pi}(\sin 2\pi-\sin 4)$$

$$=\frac{1}{2\pi}(0.7568)$$

$$=0.1205$$

$$hd(2)=\frac{1}{-\pi}\left[\sin(-\pi)-\sin(-2)\right]$$

$$=\frac{1}{\pi}(\sin\pi-\sin 2)$$

$$=0.2895$$

For n = 3

Using 'L' hospital Rule

$$\lim_{n\to 3}\frac{\cos\left[\pi(n-3)\right]\pi-\cos 2(n-3)2}{\pi}$$

$$\frac{\pi-2}{\pi}$$

$$hd(3) = 0.363$$
$$hd(4) = hd(2) = 0.2895$$
$$hd(5) = hd(1) = 0.1205$$
$$hd(6) = hd(0) = 0.0296$$

Find the window coefficients

$$W_{HM}(n) = 0.54 - 0.46\cos\frac{2\pi n}{N-1}$$
$$W_{HM}(0) = W_{HM}(6) = 0.54 - 0.46 = 0.08$$
$$W_{HM}(1) = W_{HM}(5) = 0.54 - 0.46\cos\left(\frac{2\pi}{6}\right) = 0.31$$
$$W_{HM}(2) = W_{HM}(4) = 0.54 - 0.46\cos\left(\frac{4\pi}{6}\right) = 0.77$$
$$W_{HM}(3) = W_{HM}(3) = 0.54 - 0.46\cos\left(\frac{6\pi}{6}\right) = 1$$

To find the filter coefficients

$$h(n) = hd(n)w(n)$$
$$h(0) = h(6) = 0.0296 \times 0.08 = 0.002368$$
$$h(1) = h(5) = 0.1205 \times 0.31 = 0.0373$$
$$h(2) = h(4) = 0.2895 \times 0.77 = 0.2229$$
$$h(3) = h(3) = 0.363 \times 1 = 0.363$$

System function

$$H(z) = \sum_{n=0}^{6} h(n)z^{-n}$$

UNIT V - DIGITAL SIGNAL PROCESSOR

PART A

1. Write short notes on general purpose DSP processors (Apr/May 2015)

General-purpose digital signal processors are basically high speed microprocessors with hard ware architecture and instruction set optimized for DSP operations. These processors make extensive use of parallelism, Harvard architecture, pipelining and dedicated hardware whenever possible to perform time consuming operations

2. Write notes on special purpose. (Apr/May 2015)

There are two types of special purpose DSP processors based on hardware. (i) Hardware designed for efficient execution of specific DSP algorithms such as digital filter, FFT. (ii) Hardware designed for specific applications, for example telecommunication, digital audio.

3. Briefly explain about Harvard architecture. (May/June 2013)

The principal feature of Harvard architecture is that the program and the data memories lie in two separate spaces, permitting full overlap of instruction fetch and execution.

Typically these types of instructions would involve their distinct type.

- Instruction fetch
- Instruction decode
- Instruction execute

4. Briefly explain about multiplier accumulator.

The way to implement the correlation and convolution is array multiplication Method. For getting down these operations we need the help of adders and multipliers. The combination of these accumulator and multiplier is called as multiplier accumulator.

5. What are the types of MAC is available?

There are two types MAC'S available

- Dedicated & integrated
- Separate multiplier and integrated unit

6. What is meant by pipeline technique? What are the advantages?

(May/June 2013) (Nov/Dec 2012)

The pipeline technique is used to allow overall instruction executions to overlap. That is where all four phases operate in parallel. By adapting this technique, execution speed is increased.

7. What are four phases available in pipeline technique? (Nov/Dec 2014)

The four phases are

- Fetch
- Decode
- Read
- Execution

In a non-pipeline machine, the instruction fetch, decode and execute take 30ns, 45 ns and 25 ns respectively. Determine the increase in throughput if the instruction were pipelined.

Assume a 5ns pipeline overhead in each stage and ignore other delays. The average instruction time is = 30 ns+45 ns + 25 ns = 100 ns

Each instruction has been completed in three cycles = 45 ns * 3 = 135ns

Throughput of the machine = The average instruction time/Number of M/C per instruction

= 100/135 = 0.7407.

But in the case of pipeline machine, the clock speed is determined by the speed of the slowest stage plus overheads. In our case is = 45 ns + 5 ns =50 ns,

The respective throughput is = 100/50 = 2.00,

The amount of speed up the operation is = 135/50 = 2.7 times

Assume a memory access time of 150 ns, multiplication time of 100 ns, addition time of 100 ns and overhead of 10 ns at each pipe stage. Determine the throughput of MAC

After getting successive addition and multiplications

The total time delay is 150 + 100 + 100 + 5 = 355 ns

System throughput is = 1/355 ns.

8. Write down the name of the addressing modes.

- Direct addressing.
- Indirect addressing.
- Bit-reversed addressing.
- Immediate addressing.
- Short immediate addressing

- Long immediateaddressing
- Circular addressing

9. What are the instructions used for block transfer in C5X Processors?

The BLDD, BLDP and BLPD instructions use the BMAR to point at the source or destination space of a block move. The MADD and MADS also use the BMAR to address an operand in program memory for a multiply accumulator operation

10. Briefly explain about the dedicated register addressing modes.

The dedicated-registered addressing mode operates like the long immediate addressing modes, except that the address comes from one of two special-purpose memory-mapped registers in the CPU: the block move address register (BMAR) and the dynamic bit manipulation register (DBMR).

The advantage of this addressing mode is that the address of the block of memory to be acted upon can be changed during execution of the program.

11. Briefly explain about bit-reversed addressing mode?

In the bit-reversed addressing mode, INDX specifies one-half the size of the FFT. The value contained in the current AR must be equal to 2n-1, where n is an integer, and the FFT size is 2n. An auxiliary register points to the physical location of a data value. When we add INDX t the current AR using bit reversed addressing, addresses are generated in a bit-reversed fashion. Assume that the auxiliary registers are eight bits long, that AR2 represents the base address of the data in memory (0110 00002), and that INDX contains the value 0000 10002.

12. Briefly explain about circular addressing mode.

Many algorithms such as convolution, correlation, and finite impulse response (FIR) filters can use circular buffers in memory to implement a sliding window; which contains the most recent data to be processed. The 'C5x supports two concurrent circular buffer operating via the ARs. The following five memory-mapped registers control the circular buffer operation.

CBSR1- Circular buffer 1 start register.

CBSR2- Circular buffer 2 start Register,

CBER1- Circular buffer 1 end register

CBER2- Circular buffer 2 end register

CBCR - Circular buffer control register.

13. Write the name of various part of C5X hardware.

Central arithmetic logic unit (CALU)

Parallel logic unit (PLU)

Auxiliary register arithmetic unit (ARAU)

Memory-mapped registers.

Program controller

14. List the various registers used with ARAU of DSP processor

(Nov/Dec 2014) (May/June 2014)

- Eight auxiliary registers
- Auxiliary register pointer
- Unsigned 16 bit ALU

15. Write short notes about arithmetic logic unit and accumulator.

The 32-bit general-purpose ALU and ACC implement a wide range of arithmetic and logical functions, the majority of which execute in a single clock cycle. Once an operation is performed in the ALU, the result is transferred to the ACC, where additional operations, such as shifting, can occur. Data that is input to the ALU can be scaled by the prescaler.

The following steps occur in the implementation of a typical ALU instruction:

- Data is fetched from memory on the data bus,
- Data is passed through the prescaler and the ALU, where the arithmetic is performed, and
- The result is moved into the ACC.

The ALU operates on 16-bit words taken from data memory or derived from immediate instructions. In addition to the usual arithmetic instructions, the ALU can perform Boolean operations, thereby facilitating the bit manipulation ability required of high-speed controller. One input to the ALU is always supplied by the ACC. The other input can be transferred from the PREG of the multiplier, the ACCB, or the output of the prescaler. After the ALU has performed the arithmetic or logical operation, the result is stored in the ACC.

16. Write short notes about parallel logic unit. (Nov/Dec 2012)

The parallel logic unit (PLU) can directly set, clear, test, or toggle multiple bits in control/status register pr and data memory location. The PLU provides a direct logic operation path to data memory values without affecting the contents of the ACC or the PREG.

17. What is meant by auxiliary register file?

The auxiliary register file contains eight memory-mapped auxiliary registers (AR0-AR7), which can be used for indirect addressing of the data memory or for temporary data storage. Indirect auxiliary register addressing allows placement of the data memory address of an instruction operand into one of the AR. The ARs are pointed to by a 3-bit auxiliary register pointer (ARP) that is loaded with a value from 0-7, designating AR0-AR7, respectively.

18. Write short notes about circular registers in C5X.

The 'C5x devices support two concurrent circular buffers operating in conjunction with user-specified auxiliary register. Two 16-bit circular buffer start registers (CBSR1 and CBSR2) indicate the address where the circular buffer starts. Two 16-bit circular buffer end registers (CBER1 and CBER2) indicate the address where the circular buffer ends. The 16-bit circular buffer control register (CBCR) controls the operation of these circular buffers and identifies the auxiliary registers to be used.

19. What are the different buses of TMS 320C54X processor and list their functions?

(May/June 2014)

The TMS 320C54X processor has the following 4 pairs/8 buses

PB	: Program Bus	Program memory bus to read opcode
PAB	: Program Address Bus	immediate operand
CB	: C Bus	Two independent data memory buses to read two
CAB	: C Address Bus	data simultaneously from memory
DB	: D Bus	
DAB	: D Address Bus	
EB	: E Bus	Data memory bus to write data in data memory
EAB	: E Address Bus	

PART B

1. **Discuss about the addressing modes used in DSP processor**

(Apr’15)(May’12) (Nov’12) (May’13) (Nov’13) (8)

- Direct addressing
- Indirect addressing
- Immediate addressing
- Dedicated-register addressing
- Memory-mapped register addressing
- Circular addressing

Direct Addressing

In the direct memory addressing mode, the instruction contains the lower 7 bits of the data memory address (dma). The 7-bit dma is concatenated with the 9 bits of the data memory page pointer (DP) in status register 0 to form the full 16-bit data memory address. This 16-bit data memory address is placed on an internal direct data memory address bus (DAB). The DP points to one of 512 possible data

memory pages and the 7-bit address in the instruction points to one of 128 words within that data memory page. You can load the DP bits by using the LDP or the LST #0 instruction.

Figure illustrates how the 16-bit data memory address is formed.

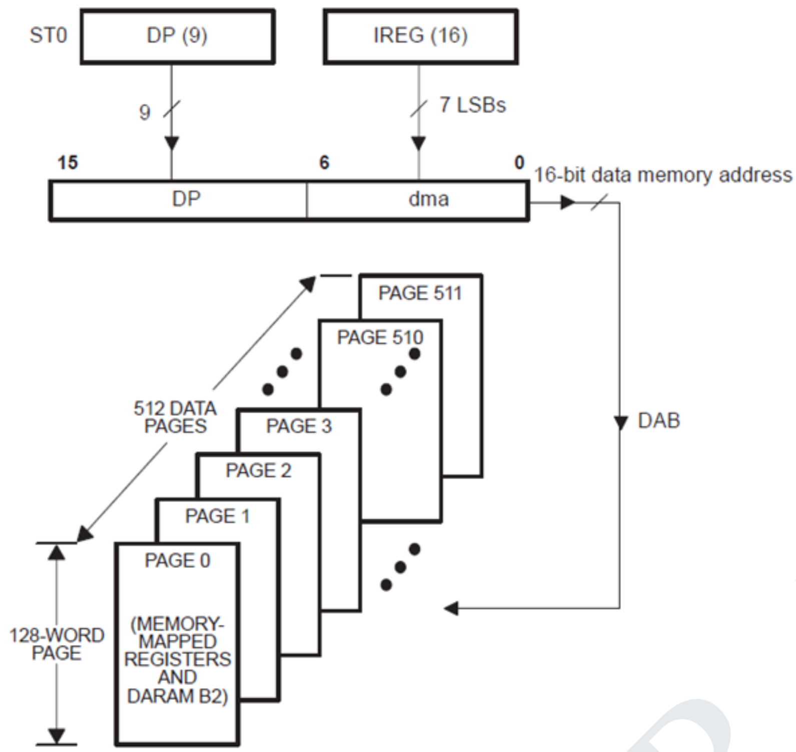
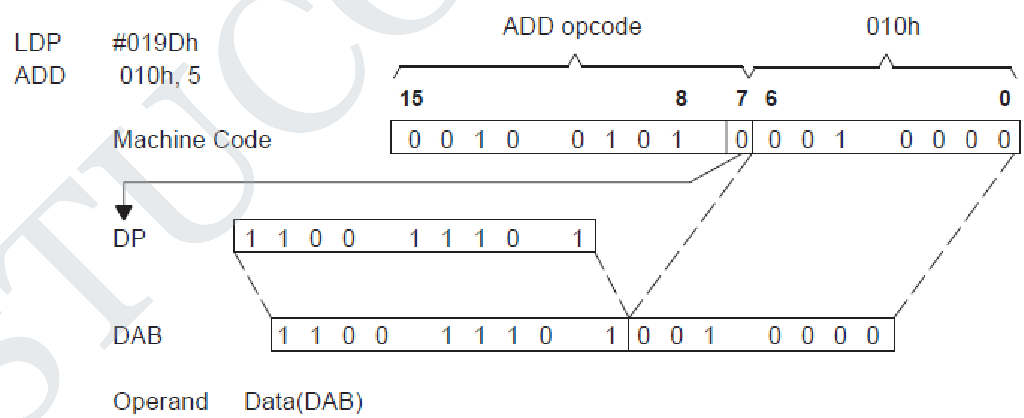


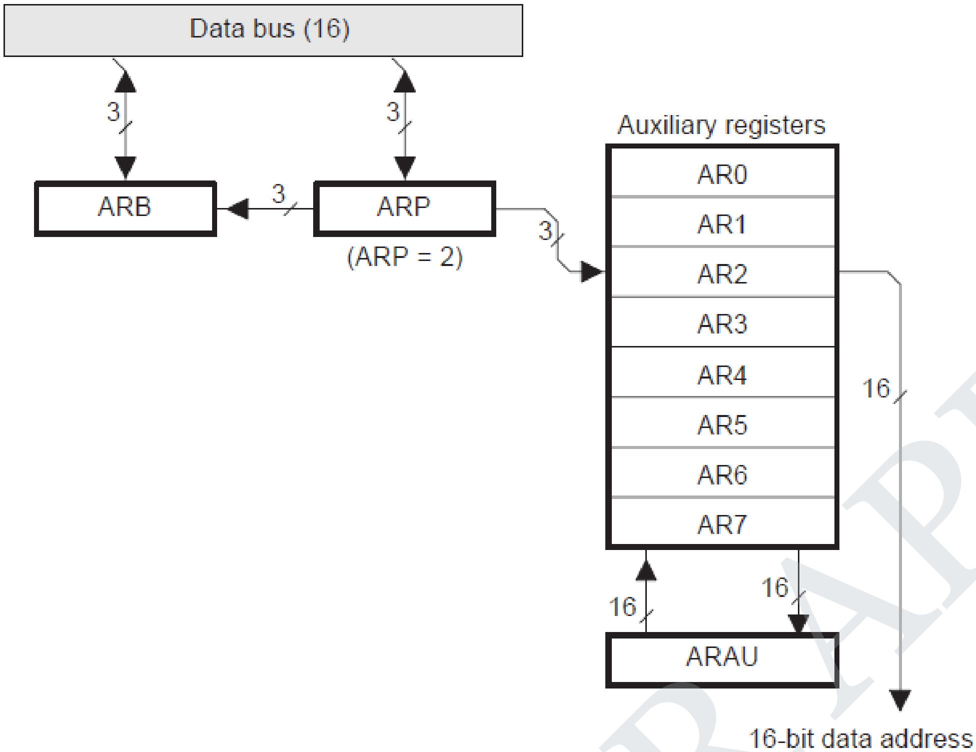
Figure shown below illustrates the direct addressing mode. Bits 15 through 8 contain the opcode. Bit 7, with a value of 0, defines the addressing mode as direct, and bits 6 through 0 contain the dma.



Note: DAB is the 16-bit internal data memory address bus.

Indirect Addressing

Eight 16-bit auxiliary registers (AR0–AR7) provide flexible and powerful indirect addressing. In indirect addressing, any location in the 64K-word data memory space can be accessed using a 16-bit address contained in an AR. Figure shows the hardware for indirect addressing.



To select a specific AR, load the auxiliary register pointer (ARP) with a value from 0 through 7, designating AR0 through AR7, respectively. The register pointed to by the ARP is referred to as the current auxiliary register (current AR). You can load the address into the AR using the LAR instruction and you can change the content of the AR by the:

ADRK instruction

MAR instruction

SBRK instruction

Indirect addressing field of any instruction supporting indirect addressing.

The content of the current AR is used as the address of the data memory operand. After the instruction uses the data value, the content of the current AR can be incremented or decremented by the auxiliary register arithmetic unit (ARAU), which implements unsigned 16-bit arithmetic.

The ARAU performs auxiliary register arithmetic operations in the decode phase of the pipeline (when the instruction specifying the operation is being decoded). This allows the address to be generated before the decode phase of the next instruction. The content of the current AR is incremented or decremented

after it is used in the current instruction. You can load the ARs via the data bus by using memory-mapped writes to the ARs. The following instructions can write to the memory-mapped ARs:

APL	OPL	SAMM	XPL
BLDD	SACH	SMMR	
LMMR	SACL	SPLK	

Be careful when using these memory-mapped loads of the ARs because, in this case, the memory-mapped ARs are modified in the execute phase of the pipeline. This causes a pipeline conflict if one of the next two instruction words modifies that AR. For further information on the pipeline and possible pipeline conflicts.

There are two ways to use the ARs for purposes other than referencing data memory addresses:

Use the ARs to support conditional branches, calls, and returns by using the CMPR instruction. This instruction compares the content of the current AR with the content of the auxiliary register compare register (ARCR) and puts the result in the test/control (TC) flag bit of status register ST1.

Use the ARs for temporary storage by using the LAR instruction to load a value into the AR and the SAR instruction to store the AR value to a data memory location.

Indirect Addressing Options

The 'C5x provides four indirect addressing options:

No increment or decrement: The instruction uses the content of the current AR as the data memory address, but neither increments nor decrements the content of the current AR.

Increment or decrement by one: The instruction uses the content of the current AR as the data memory address and then increments or decrements the content of the current AR by one.

Increment or decrement by an index amount: The value in INDX is the index amount. The instruction uses the content of the current AR as the data memory address and then increments or decrements the content of the current AR by the index amount.

Increment or decrement by an index amount using reverse carry: The value in INDX is the index amount. The instruction uses the content of the current AR as the data memory address and then increments or decrements the content of the current AR by the index amount. The addition or subtraction is done using reverse carry propagation.

Indirect Addressing Opcode Format:

Indirect addressing can be used with all instructions except those with immediate operands or with no operands.

Indirect Addressing Opcode Format Diagram

15–8	7	6	5	4	3	2–0
Opcode	I	IDV	INC	DEC	N	NAR

Immediate Addressing:

In immediate addressing, the instruction word(s) contains the value of the immediate operand. The ’C5x has both 1-word (8-bit, 9-bit, and 13-bit constant) short immediate instructions and 2-word (16-bit constant) long immediate instructions. Table 5–5 lists the instructions that support immediate addressing.

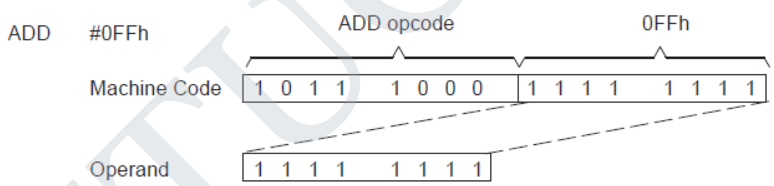
Instructions That Support Immediate Addressing

Short Immediate (1-Word)			Long Immediate (2-Word)	
8-Bit Constant	**9-Bit Constant**	**13-Bit Constant**	**16-Bit Constant**	
ADD	LDP	MPY	ADD	OR
ADRK			AND	RPT
LACL			APL	RPTZ
LAR			CPL	SPLK
RPT			LACC	SUB
SBRK			LAR	XOR
SUB			MPY	XPL
			OPL	

Short Immediate Addressing

In short immediate instructions, the operand is contained within the instruction machine code. Figure 5–5 shows an example of the short immediate mode. Note that in this example, the lower 8 bits are the operand and will be added to the ACC by the CALU.

Short Immediate Addressing Mode



Long Immediate Addressing

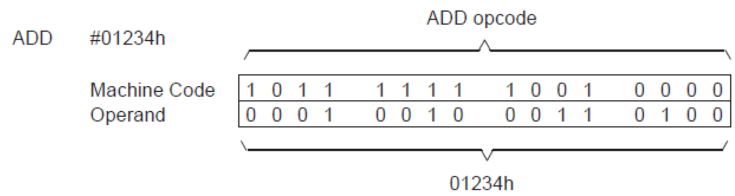
In long immediate instructions, the operand is contained in the second word of a two-word instruction. There are two long immediate addressing modes:

- One-operand instructions
- Two-operand instructions

Long Immediate Addressing with Single/No Data Memory Access

Figure shows an example of long immediate addressing with no data memory access. In Figure 5–6, the second word of the 2-word instruction is added to the ACC by the CALU.

Long Immediate Addressing Mode — No Data Memory Access



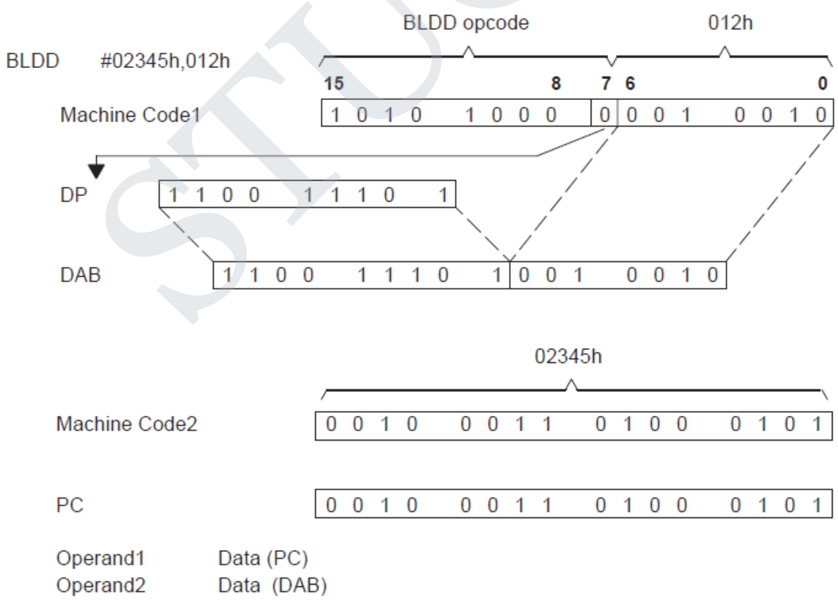
Long Immediate Addressing with Dual Data Memory Access

The long immediate addressing also could apply for a second data memory access for the execution of the instruction. The prefetch counter (PFC) is pushed onto the microcall stack (MCS), and the long immediate value is loaded into the PFC. The program address/data bus is then used for the operand fetch or write. At the completion of the instruction, the MCS is popped back to the PFC, the program counter (PC) is incremented by two, and execution continues. The

PFC is used so that when the instruction is repeated, the address generated can be autoincremented.

Figure shows an example of long immediate addressing with two operands. In Figure, the source address (OPERAND1) is fetched via PAB, and the destination address (OPERAND2) uses the direct addressing mode. Bits 15 through 8 of machine code1 contain the opcode. Bit 7, with a value of 0, defines the addressing mode as direct, and bits 6 through 0 contain the dma.

Long Immediate Addressing Mode — Two Operands



DAB is the 16-bit internal data memory address bus.

Dedicated-Register Addressing

The dedicated-registered addressing mode operates like the long immediate addressing mode, except that the address comes from one of two special-purpose memory-mapped registers in the CPU: the block move address register (BMAR) and the dynamic bit manipulation register (DBMR). The advantage of this addressing mode is that the address of the block of memory to be acted upon can be changed during execution of the program. The syntax for dedicated-register addressing can be stated in one of two ways:

Specify BMAR by its predefined symbol:

BLDD BMAR,DAT100 ;DP = 0. BMAR contains the value 200h.

The content of data memory location 200h is copied to data memory location

100 on the current data page.

Exclude the immediate value from a parallel logic unit (PLU) instruction:

OPL DAT10 ;DP = 6. DBMR contains the value FFF0h.

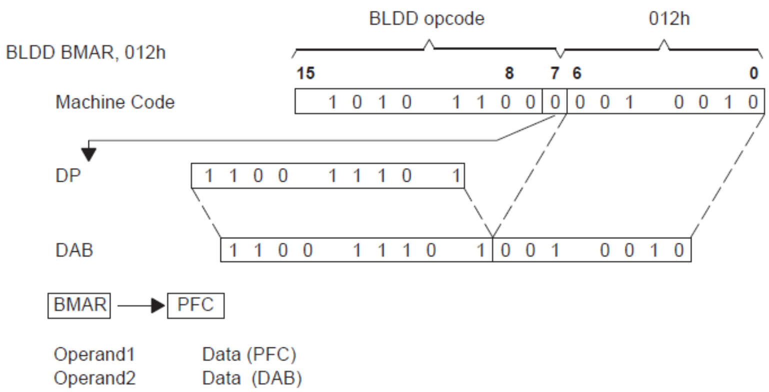
;Address 030Ah contains the value 01h

The content of data memory location 030Ah is ORed with the content of the DBMR. The resulting value FFF1h is stored back in memory location 030Ah.

Using the Contents of the BMAR

The BLDD, BLDP, and BLPD instructions use the BMAR to point at the source or destination space of a block move. The MADD and MADS instructions also use the BMAR to address an operand in program memory for a multiply accumulate operation. Figure shows how the BMAR is used in the dedicated-register addressing mode. Bits 15 through 8 of the machine code contain the opcode. Bit 7, with a value of 0, defines the addressing mode as direct, and bits 6 through 0 contain the dma.

Dedicated-Register Addressing Using the BMAR

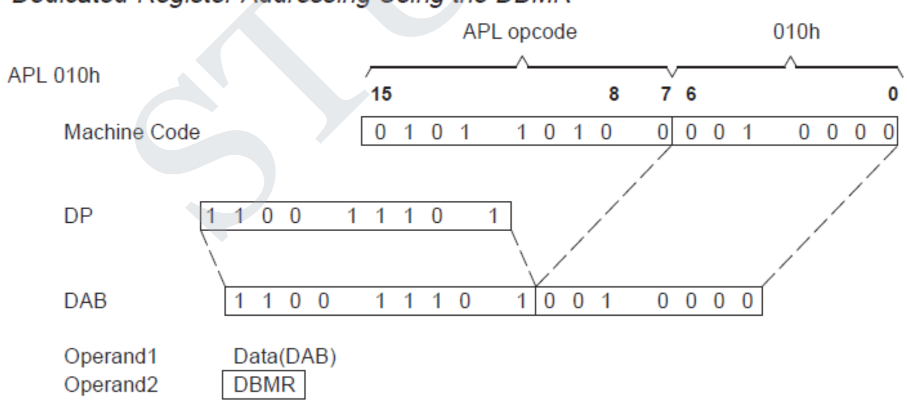


DAB is the 16-bit internal data memory address bus.

Using the Contents of the DBMR

The APL, CPL, OPL, and XPL instructions use the PLU and the contents of the DBMR when an immediate value is not specified as one of the operands. Figure 5–9 illustrates how the DBMR is used as an AND mask in the APL instruction. Bits 15 through 8 of the machine code contain the opcode. Bit 7, with a value of 0, defines the addressing mode as direct, and bits 6 through 0 contain the dma.

Dedicated-Register Addressing Using the DBMR



DAB is the 16-bit internal data memory address bus.

Memory-Mapped Register Addressing

With memory-mapped register addressing, you can modify the memory mapped registers without affecting the current data page pointer value. In addition, you can modify any scratch pad RAM (DARAM B2) location or data page 0. The memory-mapped register addressing mode operates like the direct addressing mode, except that the 9 MSBs of the address are forced to

0 instead of being loaded with the contents of the DP. This allows you to address the memory-mapped registers of data page 0 directly without the overhead of changing the DP or auxiliary register.

The following instructions operate in the memory-mapped register addressing mode. Using these instructions does not affect the contents of the DP:

LAMM — Load accumulator with memory-mapped register

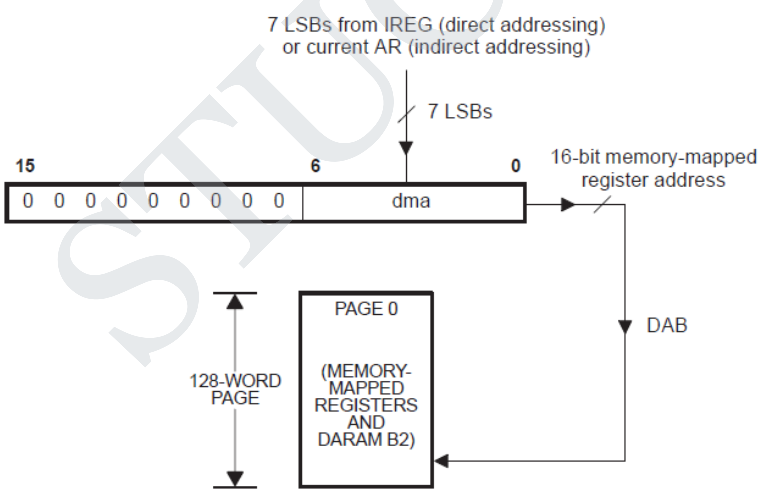
LMMR — Load memory-mapped register

SAMM — Store accumulator in memory-mapped register

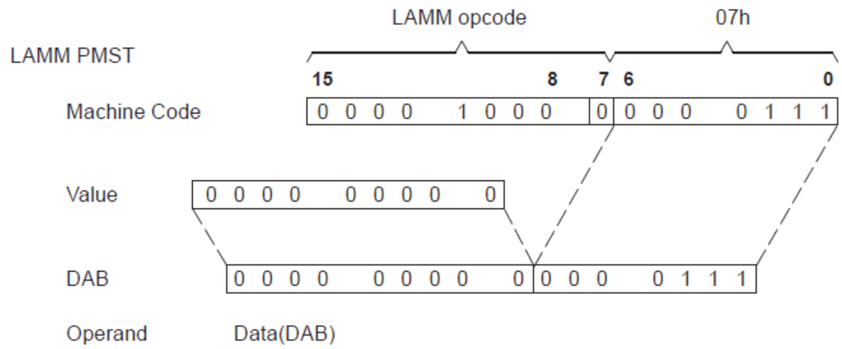
SMMR — Store memory-mapped register

Figure illustrates how this is done by forcing the 9 MSBs of the data memory address to 0, regardless of the current value of the DP when direct addressing is used or of the current AR value when indirect addressing is used.

Memory-Mapped Register Addressing



Memory-Mapped Addressing in the Direct Addressing Mode



DAB is the 16-bit internal data memory address bus.

Circular Addressing

Many algorithms such as convolution, correlation, and finite impulse response (FIR) filters can use circular buffers in memory to implement a sliding window, which contains the most recent data to be processed. The ’C5x supports two concurrent circular buffers operating via the ARs. The following five memory-mapped registers control the circular buffer operation:

CBSR1 — Circular buffer 1 start register

CBSR2 — Circular buffer 2 start register

CBER1 — Circular buffer 1 end register

CBER2 — Circular buffer 2 end register

CBCR — Circular buffer control register

The 8-bit CBCR enables and disables the circular buffer operation. To define circular buffers, you first load the start and end addresses into the corresponding buffer registers; next, load a value between the start and end registers for the circular buffer into an AR. Load the proper AR value, and set the corresponding circular buffer enable bit in the CBCR. Note that you must

not enable the same AR for both circular buffers; if you do, unexpected results occur. The algorithm for circular buffer addressing below shows that the test of the AR value is performed before any modifications:

If (ARn = CBER) and (any AR modification),

Then: ARn = CBSR.

Else: ARn = ARn + step.

If ARn = CBER and no AR modification occurs, the current AR is not modified and is still equal to CBER. When the current AR = CBER, any AR modification (increment or decrement) will set the current AR = CBSR.

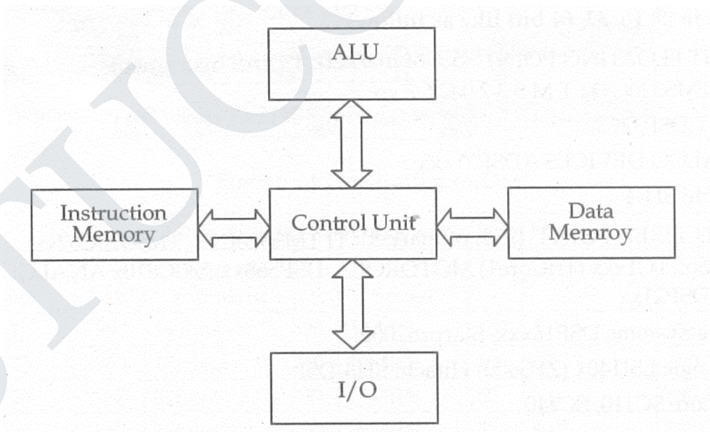
In circular addressing, the step is the quantity that is being added to or subtracted from the specified AR. Take care when using a step of greater than 1 to modify the AR pointing to an element of the circular buffer. If an update to an AR generates an address outside the range of the circular buffer, the ARAU does not detect this situation, and the buffer does not wrap around. AR updates are performed, Indirect Addressing. Because of the pipeline, there is a two-cycle latency between configuring the CBCR and performing AR modifications. Circular buffers can be used in increment- or decrement-type updates. For incrementing the value in the AR, the value in CBER must be greater than the value in CBSR. For decrementing the value in the AR, the value in CBSR must be greater than the value in CBER.

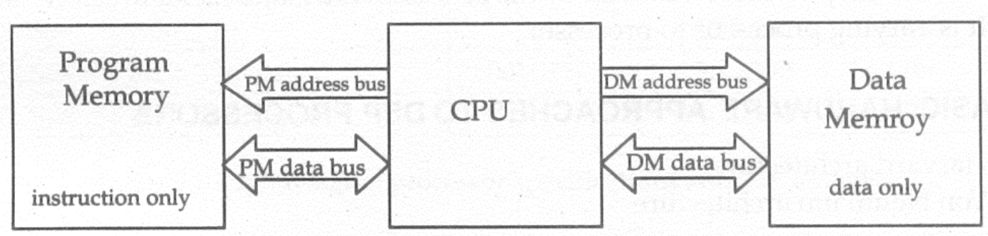
2. Draw the block diagram of Von Neumann, Harvard architecture and modified Harvard architecture and explain. (Nov/Dec 2012) (Nov/Dec 2013) (16)

Solution:

HARVARD ARCHITECTURE

The **Harvard architecture** is a computer architecture with physically separate storage and signal pathways (buses) for instructions and data. The term originated from the **Harvard Mark I** relay-based computer, which stored instructions on punched tape (24 bits wide) and data in electro-mechanical counters. These early machines had data storage entirely contained within the central processing unit, and provided no access to the instruction storage as data. Programs needed to be loaded by an operator; the processor could not initialize itself. Today, most processors implement such separate signal pathways for performance reasons, but actually implement a modified Harvard architecture, so they can support tasks like loading a program from disk storage as data and then executing it.





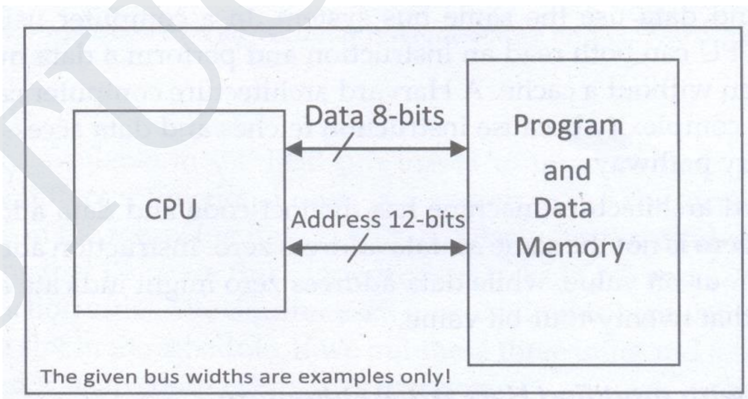
Harvard Architecture

VON NEUMANN ARCHITECTURE

The **von Neumann architecture**, also known as the **von Neumann model** and **Princeton architecture**, is a computer architecture based on that described in 1945 by the mathematician and physicist **John von Neumann** and others .This describes a design architecture for an electronic digital computer with parts consisting of a processing unit containing an arithmetic logic unit and processor registers, a control unit containing an instruction register and program counter, a memory to store both data and instructions, external mass storage, and input and output mechanisms. The meaning has evolved to be any stored-program computer in which an instruction fetch and a data operation cannot occur at the same time because they share a common bus. This is referred to as the von Neumann bottleneck and often limits the performance of the system.

The design of a von-Neumann architecture is simpler than the more modern Harvard architecture which is also a stored-program system but has one dedicated set of address and data buses for reading data from and writing data to memory, and another set of address and data buses for fetching instructions.

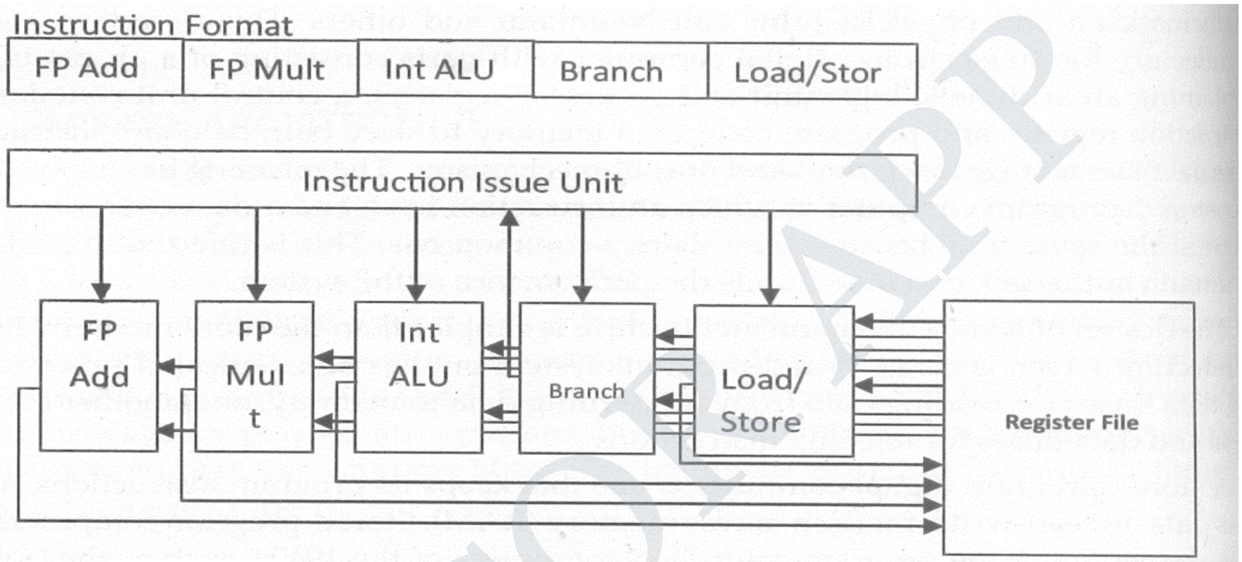
A stored-program digital computer is one that keeps its program instructions, as well as its data, in read-write, random-access memory (RAM). Stored-program computers were advancement over the program-controlled computers of the 1940s, such as the Colossus and the ENIAC, which were programmed by setting switches and inserting patch leads to route data and to control signals between various functional units. In the vast majority of modern computers, the same memory is used for both data and program instructions, and the von Neumann vs. Harvard distinction applies to the cache architecture, not the main memory.



VON NEUMANN ARCHITECTURE

VERY LONG INSTRUCTION WORD (VLIW) OR MODIFIED VON NEUMANN ARCHITECTURE

Very **Long Instruction Word** (VLIW) or **modified von Neumann architecture** refers to processor architectures designed to take advantage of Instruction Level Parallelism (ILP). Whereas conventional processors mostly allow programs only to specify instructions that will be executed in sequence, a VLIW processor allows programs to explicitly specify instructions that will be executed at the same time (that is in parallel). This type of processor architecture is intended to allow higher performance without the inherent complexity of some other approaches.



Very Long Instruction Word (VLIW) architecture

Contrast with von Neumann architectures

Under pure von Neumann architecture the CPU can be either reading an instruction or reading/writing data from/to the memory. Both cannot occur at the same time since the instructions and data use the same bus system. In a computer using the Harvard architecture, the CPU can both read an instruction and perform a data memory access at the same time, even without a cache. A Harvard architecture computer can thus be faster for a given circuit complexity because instruction fetches and data access do not contend for a single memory pathway.

Also, a Harvard architecture machine has distinct code and data address spaces: instruction address zero is not the same as data address zero. Instruction address zero might identify a twenty-four bit value, while data address zero might indicate an eight-bit byte that is not part of that twenty-four bit value.

Contrast with modified Harvard architecture

A modified Harvard architecture machine is very much like a Harvard architecture machine, but it relaxes the strict separation between instruction and data while still letting the CPU concurrently access two (or more) memory buses. The most common modification includes separate instruction and data caches backed by a common address space. While the CPU executes from cache, it acts as a pure Harvard machine. When accessing backing memory, it acts like a von Neumann machine (where code can be moved around like data which is a powerful technique). This modification is widespread in modern processors, such as the ARM architecture and x86 processors. It is sometimes loosely called Harvard architecture, overlooking the fact that it is actually "modified".

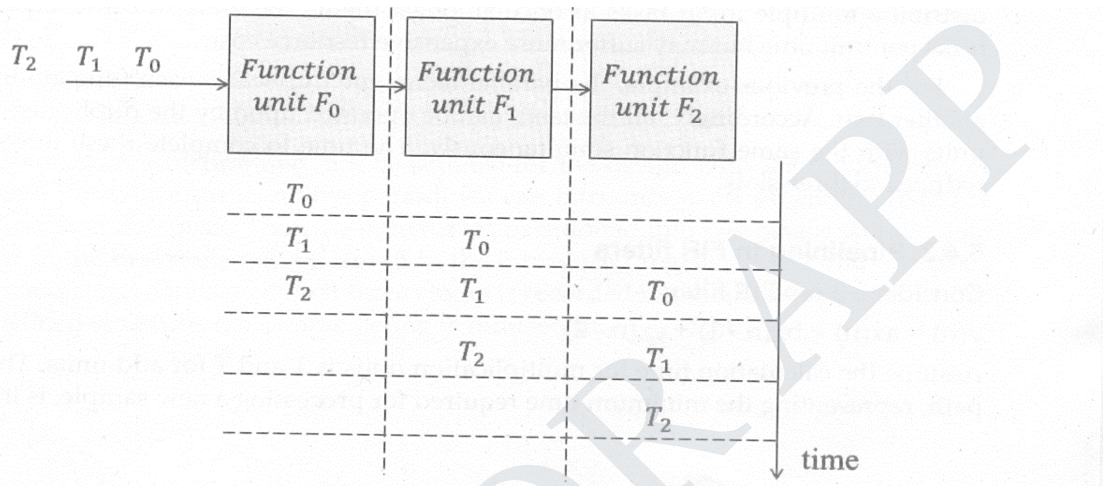
Another modification provides a pathway between the instruction memory (such as ROM or flash memory) and the CPU to allow words from the instruction memory to be treated as read-only data. This technique is used in some microcontrollers, including the Atmel AVR. This allows constant data, such as text strings or function tables, to be accessed without first having to be copied into data memory, preserving scarce (and power-ungry) data memory for read/write variables. Special machine language instructions are provided to read data from the instruction memory. (This is distinct from instructions which themselves embed constant data, although for individual constants the two mechanisms can substitute for each other.)

VAN-NEUMANN Architecture	Harvard Architecture
Used in conventional processors found in PCs and Servers, and embedded systems with only control functions.	Used in DSPs and other processors found in latest embedded systems and Mobile communication systems, audio, speech, image processing systems
The data and program are stored in the same memory	The data and program memories are separate
The code is executed serially and takes more clock cycles	The code is executed in parallel
There is no exclusive Multiplier	It has MAC (Multiply Accumulate)
Absence of Barrel Shifter	Barrel Shifter help in shifting and rotating operations of the data
The programs can be optimized in lesser size	The program tend to grow large in size

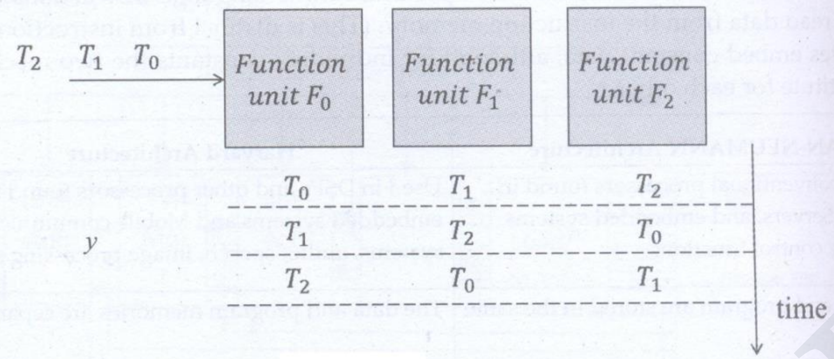
Comparison of Van-Neumann Architecture and Harvard Architecture

3. Explain in detail about Pipelining (May/June 2014) (16)

It is a key facility available in the DSP processors to improve their performance. The Pipelining allows different functional units of a system to run concurrently. Consider an informal example in the following figure. A system includes three sub-function units (F_0, F_1 and F_2). Assume that there are three independent tasks (T_0, T_1 and T_2) being performed by these three function units. The time for each function unit to complete a task is the same and will occupy a slot in the schedule. If we put these three units and tasks in a sequential order, the required time to complete them is five slots.



However, if we pipeline T_0 to T_2 concurrently, the aggregate time is reduced to three slots.



Pipelining

Therefore, it is possible for an adequate pipelined design to achieve significant enhancement on speed.

Disadvantages

Pipelining cannot decrease the processing time required for a single task. The advantage of pipelining is that it increases the throughput of the system when processing a stream of tasks.

Applying too many pipelined functions can lead to increased latency - that is, the time required for a single task to propagate through the full pipe is prolonged. A pipelined system may also require more resources (buffers, circuits, processing units, memory etc.), if the reuse of resources across different stages is restricted.

Comparison with parallel approaches

Another technique to enhance the efficiency through concurrency is parallel processing. The core difference is that parallel techniques usually duplicate function units and distribute multiple input tasks at once amongst them. Therefore, it can complete more tasks per unit time but may suffer more expensive resource costs.

For the previous example, the parallel technique duplicates each function units into another two. Accordingly, all the tasks can be operated upon by the duplicated function units with the same function simultaneously. The time to complete these three tasks is reduced to three slots.

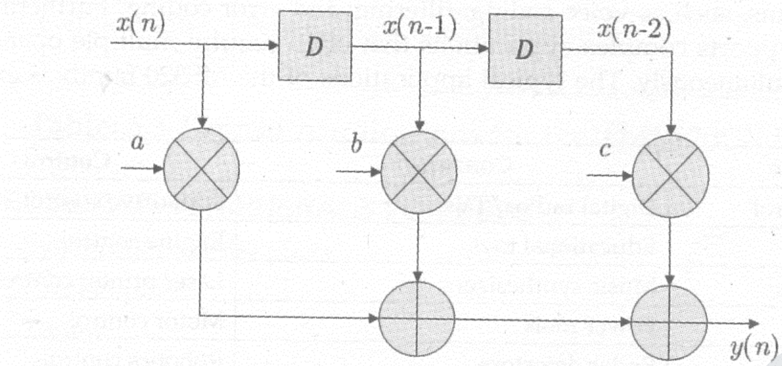
Pipelining in FIR filters

Consider a 3-tap FIR filter:

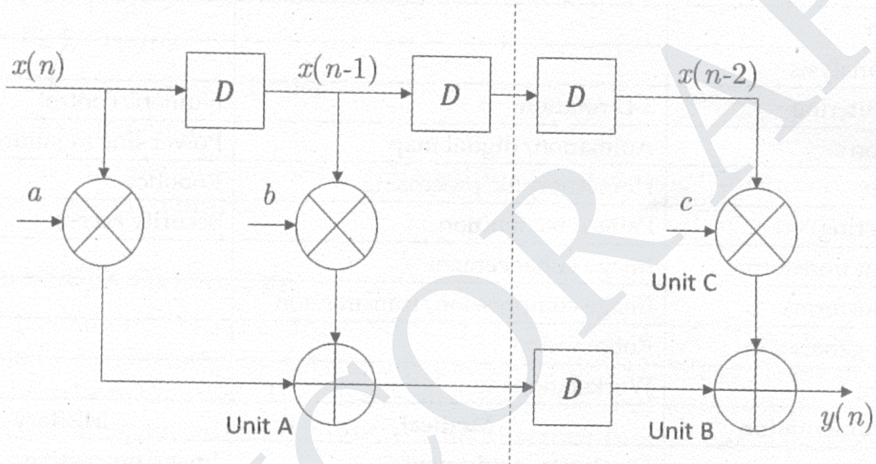
$$y(n) = ax(n) + bx(n-1) + cx(n-2)$$

Assume the calculation time for multiplication units is T and T for add units. The critical path, representing the minimum time required for processing a new sample, is limited by

1 multiplication and 2 add function units. Therefore, the sample period is given by T_{sample} $T_m + 2T_a$



Which is as shown in the following figure.



Pipelining in FIR filters

Clock	Input	Unit A	Unit B	Unit C	Output
0	x(0)	ax(0) + bx(-1)	-	-	-
1	x(1)	ax(1) + bx(0)	ax(0) + bx(-1)	cx(-2)	y(1)
2	x(2)	ax(2) + bx(1)	ax(0) + bx(0)	cx(-1)	y(2)
3	x(3)	ax(3) + bx(2)	ax(2) + bx(1)	cx(0)	y(3)

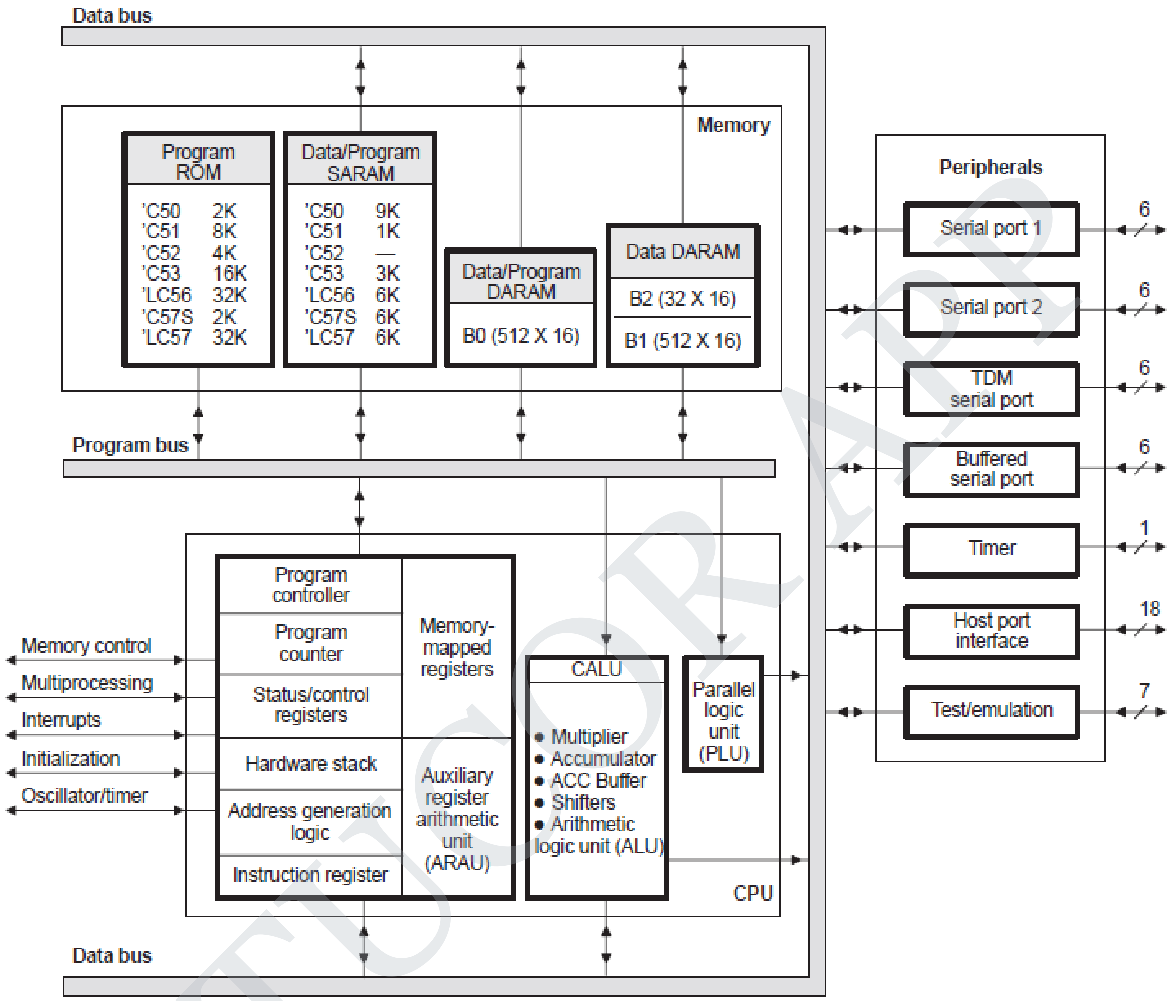
However, such structure may not be suitable for the design with the requirement of high speed. To reduce the sampling period, we can introduce extra pipelining registers along the critical data path. Then the structure is partitioned into two stages and the data produced in the first stage will be stored in the introduced registers, delaying one clock to the second stage. The data in first three clocks is recorded in the following table. Under such pipelined structure, the sample period is reduced to T_{sample} $T_m + T_a$.

4. Draw and explain the architecture of TMS 320C54x processor (Apr/May 2015) (Nov/Dec 2014) (May/June 2014) (May/June 2013) (May/June 2012) (16)

The architectural structure of the C5x which consists of the buses, on-chip memory, central processing unit (CPU) and on-chip peripherals.

The ’C5x uses an advanced, modified Harvard-type architecture based on the ’C25 architecture and maximizes processing power with separate buses for program memory and data memory. The instruction set supports data transfers between the two memory spaces. The Figure shown below is a functional block diagram of the ’C5x. All ’C5x DSPs have the same CPU structure; however, they have different on-chip memory configurations and on-chip peripherals.

Functional Block Diagram



Bus Structure

Separate program and data buses allow simultaneous access to program instructions and data, providing a high degree of parallelism. For example, while data is multiplied, a previous product can be loaded into, added to, or subtracted from the accumulator and, at the same time, a new address can be generated. Such parallelism supports a powerful set of arithmetic, logic, and bit-manipulation operations that can all be performed in a single machine cycle. In addition, the 'C5x includes the control mechanisms to manage interrupts, repeated operations, and function calling.

The 'C5x architecture is built around four major buses:

- Program bus (PB)

- Program address bus (PAB)
- Data read bus (DB)
- Data read address bus (DAB)

The PAB provides addresses to program memory space for both reads and writes. The PB also carries the instruction code and immediate operands from program memory space to the CPU. The DB interconnects various elements of the CPU to data memory space. The program and data buses can work together to transfer data from on-chip data memory and internal or external program memory to the multiplier for single-cycle multiply/accumulate operations

Central Processing Unit (CPU)

The ’C5x CPU consists of these elements:

- Central arithmetic logic unit (CALU)
- Parallel logic unit (PLU)
- Auxiliary register arithmetic unit (ARAU)
- Memory-mapped registers
- Program controller

The ’C5x CPU maintains source-code compatibility with the ’C1x and ’C2x generations while achieving high performance and greater versatility. Improvements include a 32-bit accumulator buffer, additional scaling capabilities, and a host of new instructions. The instruction set exploits the additional hardware features and is flexible in a wide range of applications. Data management has been improved through the use of new block move instructions and memory-mapped register instructions.

Central Arithmetic Logic Unit (CALU)

The CPU uses the CALU to perform 2s-complement arithmetic. The CALU consists of these elements:

- 16-bit x 16-bit multiplier
- 32-bit arithmetic logic unit (ALU)
- 32-bit accumulator (ACC)
- 32-bit accumulator buffer (ACCB)
- Additional shifters at the outputs of both the accumulator and the product register (PREG)

Parallel Logic Unit (PLU)

The CPU includes an independent PLU, which operates separately from, but in parallel with, the ALU. The PLU performs Boolean operations or the bit manipulations required of high-speed controllers. The PLU can set, clear, test, or toggle bits in a status register, control register, or any data memory location.

The PLU provides a direct logic operation path to data memory values without affecting the contents of the ACC or PREG. Results of a PLU function are written back to the original data memory location.

Auxiliary Register Arithmetic Unit (ARAU)

The CPU includes an unsigned 16-bit arithmetic logic unit that calculates indirect addresses by using inputs from the auxiliary registers (ARs), index register (INDX), and auxiliary register compare register (ARCR). The ARAU can autoindex the current AR while the data memory location is being addressed and can index either by 1 or by the contents of the INDX. As a result, accessing data does not require the CALU for address manipulation; therefore, the CALU is free for other operations in parallel. For information on the ARAU.

Memory-Mapped Registers

The ’C5x has 96 registers mapped into page 0 of the data memory space. All ’C5x DSPs have 28 CPU registers and 16 input/output (I/O) port registers but have different numbers of peripheral and reserved registers. Since the memory-mapped registers are a component of the data memory space, they can be written to and read from in the same way as any other data memory location. The memory-mapped registers are used for indirect data address pointers, temporary storage, CPU status and control, or integer arithmetic processing through the ARAU.

Program Controller

The program controller contains logic circuitry that decodes the operational instructions, manages the CPU pipeline, stores the status of CPU operations, and decodes the conditional operations. Parallelism of architecture lets the ’C5x perform three concurrent memory operations in any given machine cycle: fetch an instruction, read an operand, and write an operand. The program controller consists of these elements:

- Program counter
- Status and control registers
- Hardware stack
- Address generation logic

- Instruction register

On-Chip Memory

The ’C5x architecture contains a considerable amount of on-chip memory to aid in system performance and integration:

- Program read-only memory (ROM)
- Data/program dual-access RAM (DARAM)
- Data/program single-access RAM (SARAM)

The ’C5x has a total address range of 224K words x 16 bits. The memory space is divided into four individually selectable memory segments: 64K-word program memory space, 64K-word local data memory space, 64K-word input/ output ports, and 32K-word global data memory space.

Program ROM

All ’C5x DSPs carry a 16-bit on-chip maskable programmable ROM. The ’C50 and ’C57S DSPs have boot loader code resident in the on-chip ROM, all other ’C5x DSPs offer the boot loader code as an option. This memory is used for booting program code from slower external ROM or EPROM to fast on-chip or external RAM. Once the custom program has been booted into RAM, the boot ROM space can be removed from program memory space by setting the MP/MC bit in the processor mode status register (PMST). The on-chip ROM is selected at reset by driving the MP/MC pin low. If the on-chip ROM is not selected, the ’C5x devices start execution from off-chip memory. The on-chip ROM may be configured with or without boot loader code. However, the on-chip ROM is intended for your specific program. Once the program is in its final form, you can submit the ROM code to Texas Instruments for implementation into your device.

Data/Program Dual-Access RAM

All ’C5x DSPs carry a 1056-word x 16-bit on-chip dual-access RAM (DARAM). The DARAM is divided into three individually selectable memory blocks:

512-word data or program DARAM block B0, 512-word data DARAM block B1, and 32-word data DARAM block B2. The DARAM is primarily intended to store data values but, when needed, can be used to store programs as well. DARAM blocks B1 and B2 are always configured as data memory; however, DARAM block B0 can be configured by software as data or program memory. The DARAM can be configured in one of two ways:

- All 1056 words × 16 bits configured as data memory
- 544 words × 16 bits configured as data memory and 512 words × 16 bits configured as program memory

DARAM improves the operational speed of the ’C5x CPU. The CPU operates with a 4-deep pipeline. In this pipeline, the CPU reads data on the third stage and writes data on the fourth stage. Hence, for a given instruction sequence, the second instruction could be reading data at the same time the first instruction is writing data. The dual data buses (DB and DAB) allow the CPU to read from and write to DARAM in the same machine cycle.

Data/Program Single-Access RAM

All ’C5x DSPs except the ’C52 carry a 16-bit on-chip single-access RAM (SARAM) of various sizes (see Table 1–1). Code can be booted from an offchip ROM and then executed at full speed, once it is loaded into the on-chip SARAM. The SARAM can be configured by software in one of three ways:

- All SARAM configured as data memory
- All SARAM configured as program memory
- SARAM configured as both data memory and program memory

The SARAM is divided into 1K- and/or 2K-word blocks contiguous in address memory space. All ’C5x CPUs support parallel accesses to these SARAM blocks. However, one SARAM block can be accessed only once per machine cycle. In other words, the CPU can read from or write to one SARAM block while accessing another SARAM block. When the CPU requests multipleaccesses, the SARAM schedules the accesses by providing a not-ready condition to the CPU and executing the multiple accesses one cycle at a time. SARAM supports more flexible address mapping than DARAM because SARAM can be mapped to both program and data memory space simultaneously.However, because of simultaneous program and data mapping, an instruction fetch and data fetch that could be performed in one machine cycle with DARAM may take two machine cycles with SARAM.

On-Chip Memory Protection

The ’C5x DSPs have a maskable option that protects the contents of on-chip memories. When the related bit is set, no externally originating instruction can access the on-chip memory spaces.

On-Chip Peripherals

All ’C5x DSPs have the same CPU structure; however, they have different onchip peripherals connected to their CPUs. The ’C5x DSP on-chip peripherals available are:

- Clock generator
- Hardware timer
- Software-programmable wait-state generators
- Parallel I/O ports
- Host port interface (HPI)
- Serial port
- Buffered serial port (BSP)
- Time-division multiplexed (TDM) serial port
- User-maskable interrupts

Clock Generator

The clock generator consists of an internal oscillator and a phase-locked loop (PLL) circuit. The clock generator can be driven internally by a crystal resonator circuit or driven externally by a clock source. The PLL circuit can generate an internal CPU clock by multiplying the clock source by a specific factor, so

you can use a clock source with a lower frequency than that of the CPU.

Hardware Timer

A 16-bit hardware timer with a 4-bit prescaler is available. This programmable timer clocks at a rate that is between 1/2 and 1/32 of the machine cycle rate (CLKOUT1), depending upon the timer’s divide-down ratio. The timer can be stopped, restarted, reset, or disabled by specific status bits.

Software-Programmable Wait-State Generators

Software-programmable wait-state logic is incorporated in 'C5x DSPs allowing wait-state generation without any external hardware for interfacing with slower off-chip memory and I/O devices. This feature consists of multiple waitstate generating circuits. Each circuit is user-programmable to operate in different wait states for off-chip memory accesses.

Parallel I/O Ports

A total of 64K I/O ports are available, sixteen of these ports are memory-mapped in data memory space. Each of the I/O ports can be addressed by the IN or the OUT instruction. The memory-mapped I/O ports can be accessed with any instruction that reads from or writes to data memory. The IS signal indicates a read or write operation through an I/O port. The 'C5x can easily interface with external I/O devices through the I/O ports while requiring minimal off-chip address decoding circuits.

Host Port Interface (HPI)

The HPI available on the 'C57S and 'LC57 is an 8-bit parallel I/O port that provides an interface to a host processor. Information is exchanged between the DSP and the host processor through on-chip memory that is accessible to both the host processor and the 'C57.

Number of Serial/Parallel Ports Available in Different 'C5x Package Types

TMS320 Device	Package ID†	High-Speed Serial Port	TDM Serial Port	Buffered Serial Port	Host Port (Parallel)
'C50/'LC50	PQ	1	1	–	–
'C51/'LC51	PQ/PZ	1	1	–	–
'C52/'LC52	PJ/PZ	1	–	–	–
'C53/'LC53	PQ	1	1	–	–
'C53S/'LC53S	PZ	2	–	–	–
'LC56	PZ	1	–	1	–
'C57S/'LC57S	PGE	1	–	1	1
'LC57	PBK	1	–	1	1

† PGE is a 20 × 20 × 1.4 mm thin quad flat-pack (TQFP) package
PJ is a 14 × 20 × 2.7 mm quad flat-pack (QFP) package
PQ is a 20 × 20 × 3.8 mm bumpered quad flat-pack (BQFP) package
PZ and PBK are a 14 × 14 × 1.4 mm thin quad flat-pack (TQFP) package

Serial Port

Three different kinds of serial ports are available: a general-purpose serial port, a time-division multiplexed (TDM) serial port, and a buffered serial port (BSP). Each ’C5x contains at least one general-purpose, high-speed synchronous, full-duplexed serial port interface that provides direct communication with serial devices such as codecs, serial analog-to-digital (A/D) converters, and other serial systems.

The serial port is capable of operating at up to onefourth the machine cycle rate (CLKOUT1). The serial port transmitter and receiver are double-buffered and individually controlled by maskable external interrupt signals. Data is framed either as bytes or as words.

Buffered Serial Port (BSP)

The BSP available on the ’C56 and ’C57 devices is a full-duplexed, doublebuffered serial port and an autobuffering unit (ABU). The BSP provides flexibility on the data stream length. The ABU supports high-speed data transfer and reduces interrupt latencies.

TDM Serial Port

The TDM serial port available on the ’C50, ’C51, and ’C53 devices is a fullduplexed serial port that can be configured by software either for synchronous operations or for time-division multiplexed operations. The TDM serial port is commonly used in multiprocessor applications.

User-Maskable Interrupts

Four external interrupt lines (INT1–INT4) and five internal interrupts, a timer interrupt and four serial port interrupts, are user maskable. When an interrupt service routine (ISR) is executed, the contents of the program counter are saved on an 8-level hardware stack, and the contents of eleven specific CPU registers are automatically saved (shadowed) on a 1-level-deep stack. When a return from interrupt instruction is executed, the CPU registers’ contents are restored.

Test/Emulation

On the ’C50, ’LC50, ’C51, ’LC51, ’C53, ’LC53, ’C57S and ’LC57S, an IEEE standard 1149.1 (JTAG) interface with boundary scan capability is used for emulation and test. This logic provides the boundary scan to and from the interfacing devices. It can be used to test pin-to-pin continuity and to perform operational tests on devices that are peripheral to the ’C5x.

On the ’C52, ’LC52, ’C53S, ’LC53S, ’LC56, and ’LC57, an IEEE standard 1149.1 (JTAG) interface without boundary scan capability is used for emulation purposes only and is interfaced to other internal scanning logic circuitry that has access to all of the on-chip resources. Thus, the ’C5x can perform on-board emulation by means of the IEEE standard 1149.1 serial scan pins and the emulation-dedicated pins.

The on-chip analysis block in conjunction with the ’C5x debugger software provides the capability to perform debugging and performance evaluation functions in a target system. The full analysis block provides the following capabilities:

Flexible breakpoint setup. Breakpoints can be triggered based on the following events:

- Program fetches/reads/writes
- EMU0/1 pin activity
- Data reads/writes
- CPU events (calls, returns, interrupts/traps, branches, pipeline clock)
- Event counter overflow

Counting of the following events for performance analysis:

- CPU clocks
- Pipeline advances
- Instruction fetches
- Calls, returns, interrupts/traps, branches
- Program fetches/reads/writes
- Data reads/writes

Program counter discontinuity trace buffer to monitor program counter flow.

The reduced analysis block on the ’C53S and ’LC53S provides the capability for breakpoint triggering based on program fetches/reads/writes and EMU0/1 pin activity.

DOWNLOADED FROM STUCOR APP

IEEE Std.1149.1 (JTAG)/Boundary-Scan Interface Configurations for the 'C5x

TMS320 Device	IEEE Std.1149.1 Interface	Boundary Scan Capability	On-Chip Analysis Block
'C50/'LC50	Yes	Yes	Full
'C51/'LC51	Yes	Yes	Full
'C52/'LC52	Yes	No	Full
'C53/'LC53	Yes	Yes	Full
'C53S/'LC53S	Yes	No	Reduced
'LC56	Yes	No	Full
'C57S/'LC57S	Yes	Yes	Full
'LC57	Yes	No	Full

CENTRAL PROCESSING UNIT (CPU)

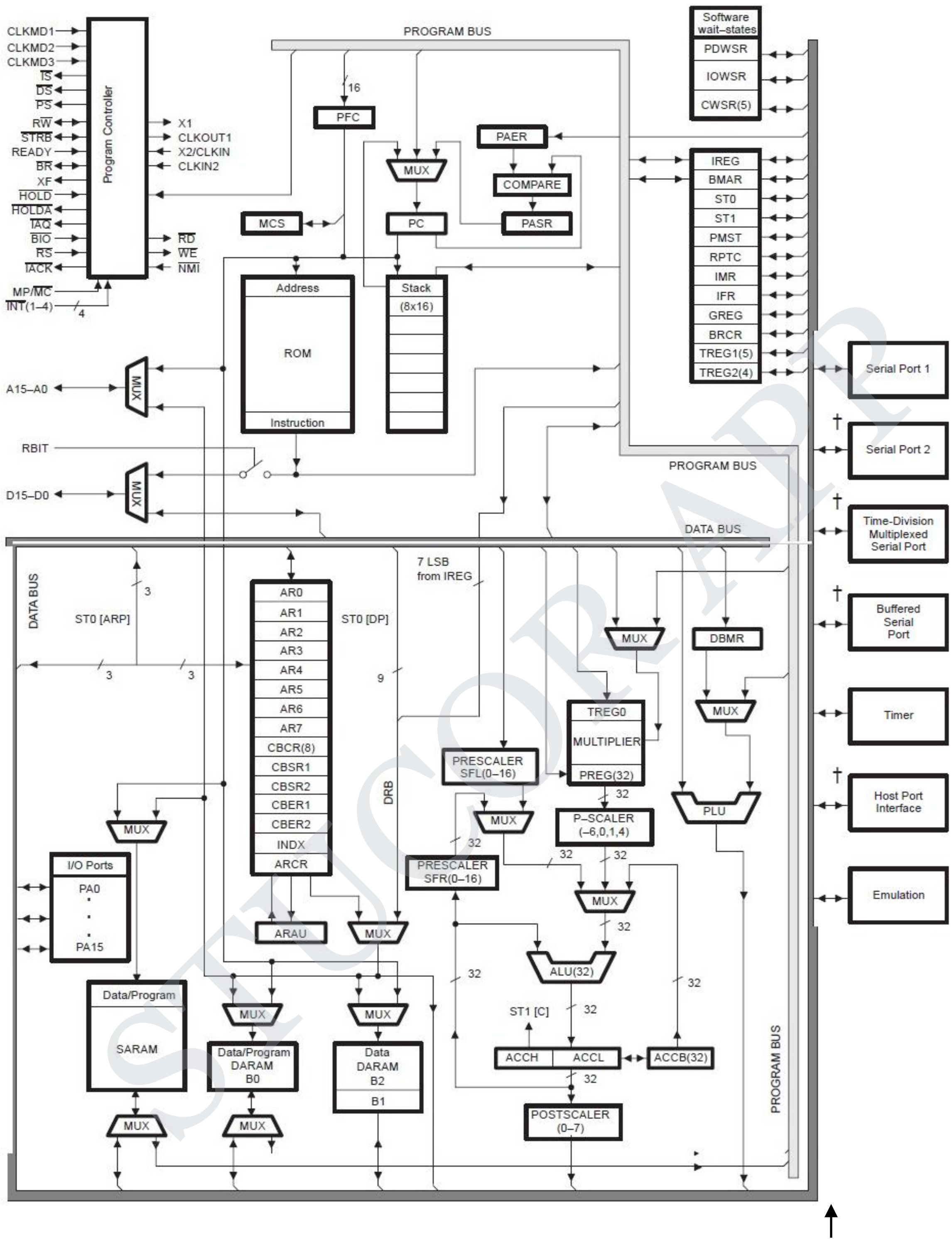
The TMS320C5x DSP central processing unit (CPU) can perform high-speed arithmetic within a short instruction cycle by means of its highly parallel architecture, which consists of the following elements:

- Program controller
- Central arithmetic logic unit (CALU)
- Parallel logic unit (PLU)
- Auxiliary register arithmetic unit (ARAU)
- Memory-mapped registers

Functional Overview

The block diagram shown in Figure outlines the principal blocks and data paths within the 'C5x. The succeeding sections provide further details of the functional blocks of the CPU. The internal hardware of the 'C5x executes functions that other processors typically implement in software or microcode. For example, the 'C5x contains hardware for single-cycle 16 x 16-bit multiplication, data shifting, and address manipulation.

BLOCK DIAGRAM OF 'C5X DSP – CENTRAL PROCESSING UNIT (CPU)



Data Bus

Note: All registers and data lines are 16-bits wide unless otherwise specified.

†Not available on all devices.

Central Arithmetic Logic Unit (CALU)

The CALU components, shown in Figure below, consists of the following:

- 16-bit x 16-bit parallel multiplier
- 32-bit 2s-complement arithmetic logic unit (ALU)
- 32-bit accumulator (ACC)
- 32-bit accumulator buffer (ACCB)
- 0-, 1-, or 4-bit left or 6-bit right shifter
- to 16-bit left barrel shifter
- to 16-bit right barrel shifter
- to 7-bit left barrel shifter

Multiplier, Product Register (PREG), and Temporary Register 0 (TREG0)

The 16-bit x 16-bit hardware multiplier can compute a signed or an unsigned 32-bit product in a single machine cycle. All multiply instructions except the multiply unsigned (MPYU) instruction perform a signed multiply operation in the multiplier. That is, two numbers being multiplied are treated as 2s-complement

numbers, and the result is a 32-bit 2s-complement number. One input to the multiplier is from memory-mapped temporary register 0 (TREG0), and the other input is from the data bus or the program bus. The 32-bit result from the multiplier is stored in the PREG and is available to the ALU. The ALU uses the 16-bit words taken from data memory or derived from an immediate instruction, or the ALU uses the 32-bit result stored in the PREG to perform arithmetic operations. The ALU can also perform Boolean operations.

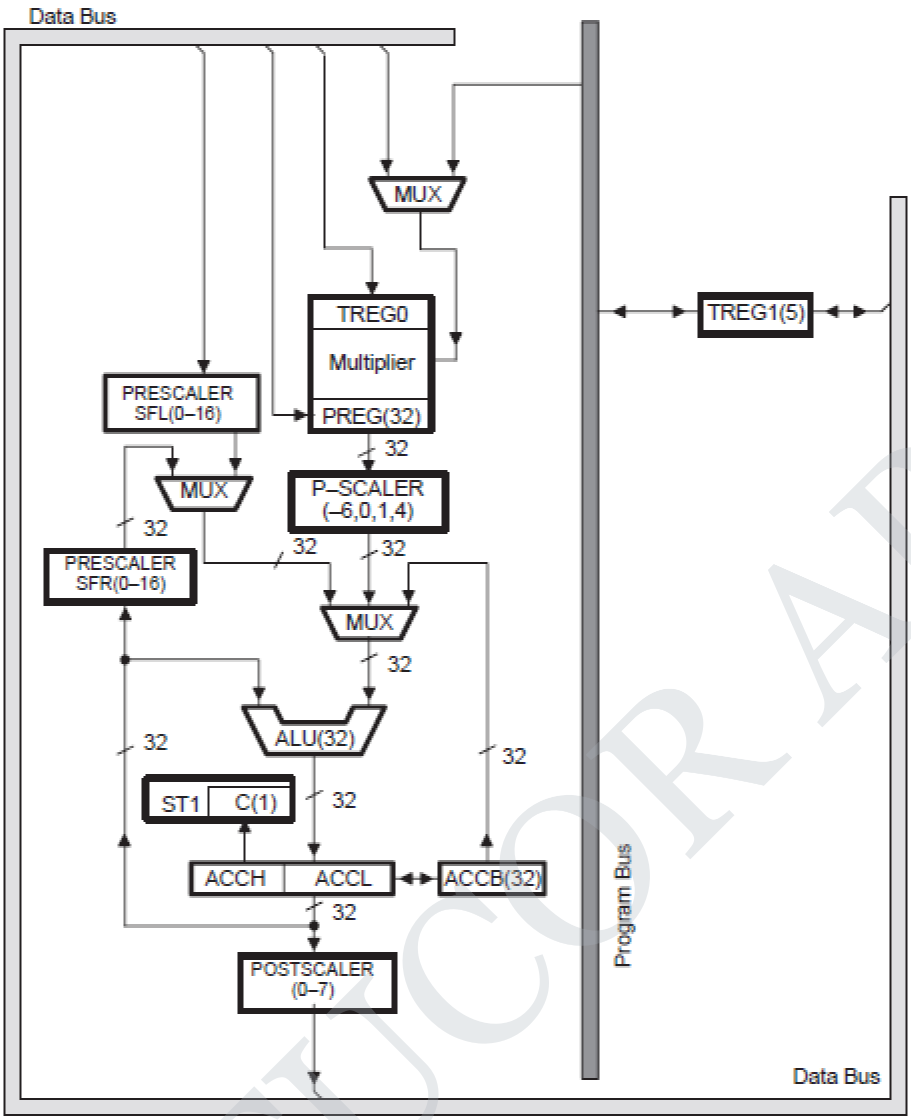
The 32-bit result from the ALU is stored in the ACC; the ACC also supplies the second input to the ALU. Instructions are provided for storing the high and low-order accumulator words in memory. The shifters (p-scaler, prescaler, and postscaler) make it possible for the CALU to perform numerical scaling, bit extraction, extended-precision arithmetic, and overflow prevention. These shifters are connected to the output of the PREG and the ACC.

The four product shift modes (PM) at the PREG output are useful for performing multiply/accumulate operations and fractional arithmetic and for justifying fractional products. The PM field of status register ST1 specifies the PM shift mode of the p-scaler:

If PM = 002, the PREG 32-bit output is not shifted when transferred into the ALU or stored.

If PM = 012, the PREG output is left-shifted 1 bit when transferred into the ALU or stored, and the LSB is zero filled. This shift mode compensates for the extra sign bit gained when multiplying two 16-bit 2s-complement numbers.

Central Arithmetic Logic Unit



Notes: All registers and data lines are 16-bits wide unless otherwise specified.

If PM = 102, the PREG output is left-shifted 4 bits when transferred into the ALU or stored, and the 4 LSBs are zero filled. This shift mode is used in conjunction with the MPY instruction with a short immediate value (13 bits or less) to eliminate the four extra sign bits gained when multiplying a16-bit number times a 13-bit number.

If PM = 112, the PREG output is right-shifted 6 bits, sign extended, when transferred into the ALU or stored, and the 6 LSBs are lost. This shift mode enables the execution of up to 128 consecutive multiply/accumulates without the possibility of overflow. Note that the product is always sign extended, regardless of the value of the sign extension mode (SXM) bit in ST1.

The PM shifts also occur when the PREG contents are stored to data memory.The PREG contents remain unchanged during the shifts.

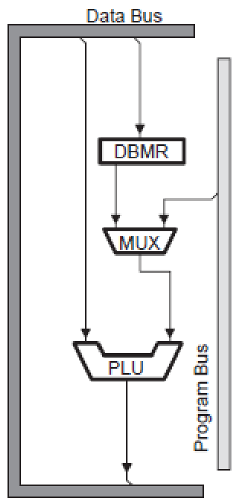
The LT (load TREG0) instruction loads TREG0, from the data bus, with the first operand; the MPY instruction provides the second operand for multiplication operations. To perfrom a multiplication with a short or long immediate operand, use the MPY instruction with an immediate operand. A product can be obtained every two cycles except when a long immediate operand is used.

Four multiply/accumulate instructions (MAC, MACD, MADD, and MADS) fully utilize the computational bandwidth of the multiplier, which allows both operands to be processed simultaneously. The data for these operations can be transferred to the multiplier each cycle via the program and data buses. When any of the four multiply/accumulate instructions are used with the RPT or RPTZ instruction, the instruction becomes a single-cycle multiply/accumulate function. In these repeated instructions, the coefficient addresses are generated by the PC while the data addresses are generated by the ARAU. This allows the RPT instruction to sequentially access the values from the coefficient table and step through the data in any of the indirect addressing modes. The RPTZ instruction also clears the ACC and the PREG to initialize the multiply/accumulate operation.

Parallel Logic Unit (PLU)

The parallel logic unit (PLU) can directly set, clear, test, or toggle multiple bits in a control/status register or any data memory location. The PLU provides a direct logic operation path to data memory values without affecting the contents of the ACC or the PREG The PLU executes a read-modify-write operation on data stored in data space. First, one operand is fetched from data memory space, and the second is fetched from a long immediate on the program bus or from the dynamic bit manipulation register (DBMR). Then, the PLU executes a logical operation on the two operands as defined by the instruction. The result is written to the same data memory location from which the first operand was fetched.

Parallel Logic Unit Block Diagram



Note: All registers and data lines are 16-bits wide unless otherwise specified.

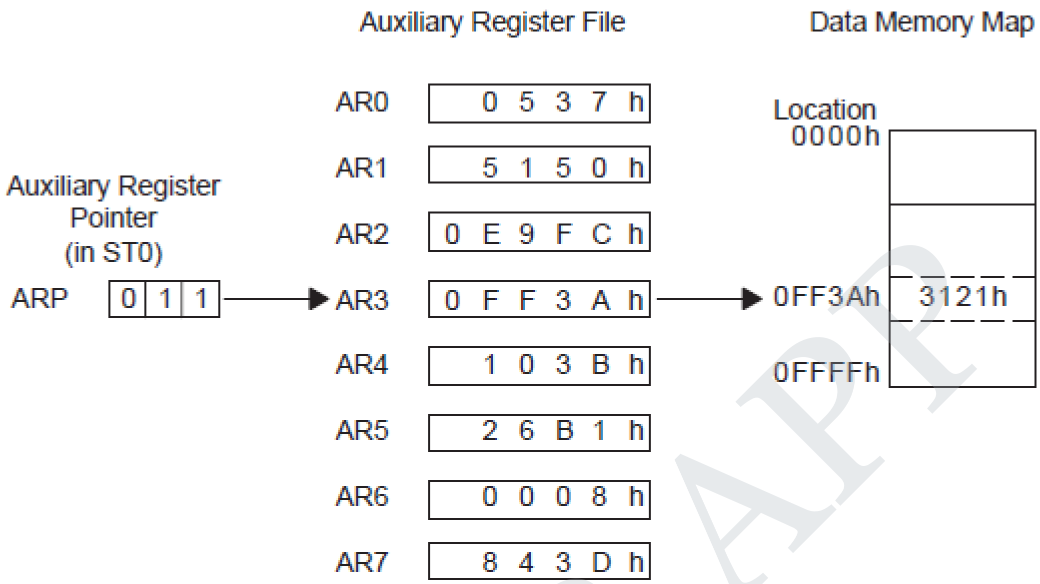
Auxiliary Register Arithmetic Unit (ARAU)

The auxiliary register file contains eight memory-mapped auxiliary registers (AR0–AR7), which can be used for indirect addressing of the data memory or for temporary data storage. Indirect auxiliary register addressing allows placement of the data memory address of an instruction operand into one of the AR. The ARs are pointed to by a 3-bit auxiliary register pointer (ARP) that is loaded with a value from 0–7, designating AR0–AR7, respectively.

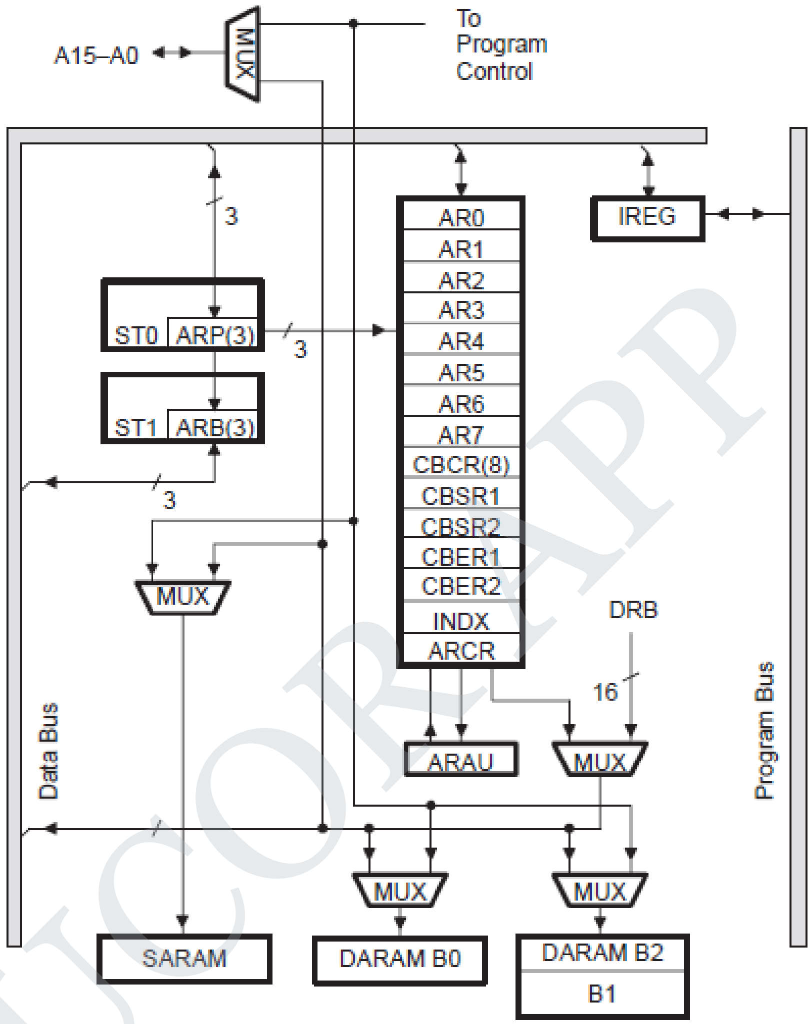
The ARs and the ARP can be loaded from data memory, the ACC or the PREG or by an immediate operand defined in the instruction. The contents of the ARs can be stored in data memory or used as inputs to the CALU. The memory-mapped ARs reside in data page 0, Local Data Memory Address Map.

The auxiliary register file (AR0–AR7) is connected to the auxiliary register arithmetic unit (ARAU), shown in Figure 3–6. The ARAU can autoindex the current AR while the data memory location is being addressed; it indexes either by ±1 or by the contents of the index register (INDX). As a result, the CALU is not needed for address manipulation when tables of information are accessed; it is free for other operations in parallel. For more advanced address manipulation, such as multidimensional array addressing, the CALU can directly read from or write to the ARs.

Indirect Auxiliary Register Addressing Example



Auxiliary Register Arithmetic Unit

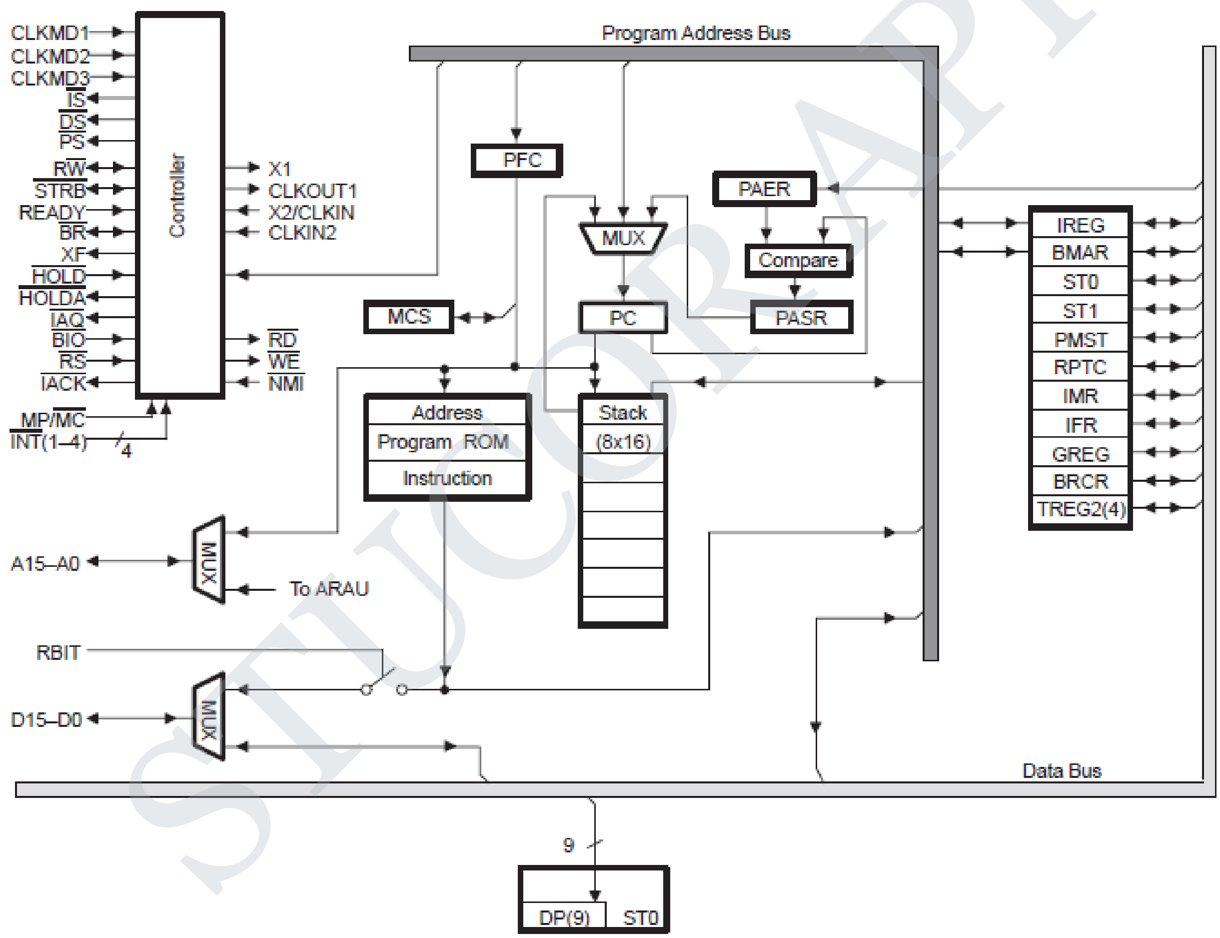


Notes: All registers and data lines are 16-bits wide unless otherwise specified.

Program Counter (PC)

The 'C5x has a 16-bit program counter (PC) which contains the address of internal or external program memory used to fetch instructions. The PC addresses program memory, either on-chip or off-chip, via the program address bus (PAB). Through the PAB, an instruction is loaded into the instruction register (IREG). Then the PC is ready to start the next instruction fetch cycle. Refer to Figure 4–1 for a functional block diagram of the program control elements.The PC is loaded in a number of ways. Table shows what address is loaded into the PC, depending on the code operation performed.

Program Control Functional Block Diagram



Notes: All registers and data lines are 16 bits wide unless otherwise specified.

5. Explain what is meant by bit reversed addressing mode. (Nov/Dec 2014) (8)

In the bit-reversed addressing mode, INDX specifies one-half the size of the FFT. The value contained in the current AR must be equal to 2n-1, where n is an integer, and the FFT size is 2n. An auxiliary register points to the physical location of a data value. When we add INDX t the current AR using bit reversed addressing, addresses are generated in a bit-reversed fashion. Assume that the auxiliary registers are eight bits long, that AR2 represents the base address of the data in memory (0110 00002), and that INDX contains the value 0000 10002.

Example shown below is a sequence of modifications to AR2 and the resulting values of AR2. Table 5–4 shows the relationship of the bit pattern of the index steps and the four LSBs of AR2, which contain the bit reversed address.

Example: Sequence of Auxiliary Register Modifications in Bit-Reversed Addressing

```
*BR0+ ;AR2  =  0110 0000    (0th value)
*BR0+ ,AR2  =  0110 1000    (1st value)
*BR0+ ;AR2  =  0110 0100    (2nd value)
*BR0+ ;AR2  =  0110 1100    (3rd value)
*BR0+ ;AR2  =  0110 0010    (4th value)
*BR0+ ;AR2  =  0110 1010    (5th value)
*BR0+ ;AR2  =  0110 0110    (6th value)
*BR0+ ;AR2  =  0110 1110    (7th value)
```

Bit-Reversed Addresses

Step	Bit Pattern	Bit-Reversed Pattern	Bit-Reversed Step
0	0000	0000	0
1	0001	1000	8
2	0010	0100	4
3	0011	1100	12
4	0100	0010	2
5	0101	1010	10
6	0110	0110	6
7	0111	1110	14
8	1000	0001	1
9	1001	1001	9
10	1010	0101	5
11	1011	1101	13
12	1100	0011	3
13	1101	1011	11
14	1110	0111	7
15	1111	1111	15