ACE
Engineering Academy
(Leading institute for ESE/GATE/PSUs)

ESE – 2019 MAINS
OFFLINE TEST SERIES

ELECTRONICS & TELECOMMUNICATION ENGINEERING (E&T)

TEST – 8
SOLUTIONS

All Queries related to **ESE – 2019 MAINS Test Series** Solutions are to be sent to the following email address

testseries@aceenggacademy.com | Contact Us : 040 – 48539866 / 040 – 40136222

01. (a)

Sol: Bode plot: It has both magnitude and phase plots

Magnitude plot: $|G(s)H(s)|$ in dB Vs frequency (ω).

Phase plot: $\angle G(s)H(s)$ Vs frequency (ω)

Procedure to sketch the magnitude plot [|G(s) H(s)| in dB vs frequency (ω)]

- Arrange the TF G(s)H(s) into the standard form.
- Find the corner frequencies and gain 'K"
- Prepare the slope/magnitude change table of G(s)H(s), in the increasing order of corner frequencies with differential (or) integral terms on top of the table.
- Use the above table to draw the magnitude plot.
- $|G(s)H(s)| = |Ks^{\pm n}|\Big|_{\omega \le \text{least corner frequency(LCF)}} = 20 \log K \pm n\,(20 \log \omega)$

  Where n = no. of differential/integral terms.
- Starting point frequency is chosen in such a way that it is always less than the lowest corner frequency.
- Starting point ($\omega \to 0$) or low frequency(less than the lowest corner frequency) asymptote slope of the Bode magnitude plot is

  $\pm n(-20\text{dB/dec}) = \pm n(6\text{db/octave})$

  where n = no. of integral/differential terms, + for differential term and – for integral term.
- High frequency($\omega \to \infty$) asymptote slope of Bode magnitude plot

  $= (P - Z)(-20\text{dB/dec})$

  $= (P - Z)(-6\text{dB/octave})$.

Phase Plot:

Eg: $G(s)H(s) = \dfrac{K(s+Z)}{s(s+P_1)(s+P_2)}$

Substitute s = jω and write the phase as shown below.

$$\angle G(s)H(s) = \angle \tan^{-1}\frac{\omega}{Z} - \left(90 + \tan^{-1}\frac{\omega}{p_1} + \tan^{-1}\frac{\omega}{p_2}\right)$$

At different frequencies calculate the phase and draw the phase plot.

01. (b)

Sol: $$\text{CLTF} = \frac{\dfrac{2K}{(1+0.02s)(s)(s+2)}}{1+\dfrac{2K}{(1+0.02s)(s)(s+2)}} = \frac{2K}{s(1+0.02s)(s+2)+2K}$$

Characteristic equation

$s(1+0.02s)(s+2) + 2K = 0$

$s(50+s)(s+2)+100K = 0$

$\Rightarrow s\,(s^2 + 52s + 100) + 100\,K = 0$

$\Rightarrow s^3 + 52s^2 + 100\,s + 100\,K = 0$

$$\begin{array}{c|cc} s^3 & 1 & 100 \\ s^2 & 52 & 100k \\ s^1 & \dfrac{5200-100k}{52} & 0 \\ s^o & 100k & 0 \end{array}$$

For system to be stable,
100 K > 0 ⇒ K > 0
5200 – 100 K > 0
⇒ 52 > K
System is stable for 0 < K < 52
∴ Maximum value of K = 52

01. (c)

Sol: The convolution property states that the convolution of two signals in time domain is equivalent to the multiplication of their spectra in frequency domain. This is called the time convolution theorem.

If $x_1(t) \xleftrightarrow{FT} X_1(\omega)$ and $x_2(t) \xleftrightarrow{FT} X_2(\omega)$

Then $x_1(t) * x_2(t) \xleftrightarrow{FT} X_1(\omega) X_2(\omega)$

Proof: We know that the convolution of two signals $x_1(t)$ and $x_2(t)$ is given by

$$x_1(t) * x_2(t) = \int_{-\infty}^{\infty} x_1(\tau) x_2(t-\tau) d\tau$$

$$F[x_1(t) * x_2(t)] = \int_{-\infty}^{\infty} \left[\int_{-\infty}^{\infty} x_1(\tau) x_2(t-\tau) d\tau \right] e^{-j\omega t} dt$$

Interchanging the order of integration, we have

$$F[x_1(t) * x_2(t)] = \int_{-\infty}^{\infty} x_1(\tau) \left[\int_{-\infty}^{\infty} x_2(t-\tau) e^{-j\omega t} dt \right] d\tau$$

Substituting t – τ = P in the second integration, we have

t = p + τ and dt = dp

$$F[x_1(t) * x_2(t)] = \int_{-\infty}^{\infty} x_1(\tau) \left[\int_{-\infty}^{\infty} x_2(p) e^{-j\omega(p+\tau)} dp \right] d\tau$$

$$= \int_{-\infty}^{\infty} x_1(\tau) \left[\int_{-\infty}^{\infty} x_2(p) e^{-j\omega p} dp \right] e^{-j\omega\tau} d\tau$$

$$= \int_{-\infty}^{\infty} x_1(\tau) X_2(\omega) e^{-j\omega\tau} d\tau$$

$$= \left[\int_{-\infty}^{\infty} x_1(\tau) e^{-j\omega\tau} d\tau \right] X_2(\omega) = X_1(\omega) X_2(\omega)$$

$$x_1(t) * x_2(t) \xleftrightarrow{FT} X_1(\omega) X_2(\omega)$$

01. (d)

Sol: let, $x(t) = L^{-1}[X(s)]$

$$= L^{-1}\left[\log \frac{s(s+1)}{s^2+1} \right]$$

$$L[x(t)] = \log \left[\frac{s(s+1)}{s^2+1} \right]$$

$= \log s(s + 1) – \log (s^2 + 1)$

$= \log s + \log (s + 1) – \log (s^2 + 1)$

$$L[tx(t)] = -\frac{d}{ds}\left[\log s + \log(s+1) - \log(s^2+1)\right] = \frac{-1}{s} - \frac{1}{s+1} + \frac{2s}{s^2+1}$$

$$t\,x(t) = L^{-1}\left(\frac{-1}{s} - \frac{1}{s+1} + \frac{2s}{s^2+1}\right)$$

$$= -L^{-1}\left(\frac{1}{s}\right) - L^{-1}\left(\frac{1}{s+1}\right) + 2L^{-1}\left(\frac{s}{s^2+1}\right)$$

$$= [-1 - e^{-t} + 2\cos(t)]\,u(t)$$

$$\therefore\ x(t) = \left[\frac{2\cos t - e^{-t} - 1}{t}\right]u(t)$$

01. (e)

Sol:

```
#include <stdio.h>
void main()
{
   int number, i;
   printf("Enter a positive integer: ");
   scanf("%d",&number);
   printf("Factors of %d are: ", number);
   for (i = 1; i < = number; ++i)
   {
       if (number%i = = 0)
     {
     printf("%d ", i);
     }
   }
}
```

01. (f)

Sol: The 8051 micro controller can recognize five different interrupts during its normal execution. They are mentioned in below.

1. Timer0 overflow interrupt - TF0
2. Timer1 overflow interrupt - TF1
3. External hardware interrupt - INT0
4. External hardware interrupt - INT1
5. Serial communication interrupt - RI/TI

The timer and serial interrupts are internally generated by the controller, whereas the external interrupts (INT0, INT1) are generated by additional interfacing devices or switches that are externally connected to the microcontroller. These external interrupts can be edge triggered or level triggered.

Interrupt vector table:

Interrupt source	Address
INT0	0003H
TF0	000BH
INT1	0013H
TF1	001BH
RI/TI	0023H

Priority table:

Interrupt source	Priority within level
INT0	Highest
TF0	↓
INT1	↓
TF1	↓
RI/TI	lowest

02. (a)

Sol:

(i) When H(S) = 1 + αs

Closed loop transfer function

$$M = \frac{G(s)}{1+G(s).H(s)}$$

$$= \frac{\frac{K}{s(s+p)}}{1+\frac{K}{s(s+p)}(1+\alpha s)}$$

$$= \frac{K}{s^2+ps+K+K\alpha s}$$

$$M = \frac{K}{s^2+s(p+K\alpha)+K}$$

Sensitivity with respect to K:

$$S_K^M = \frac{\partial M}{\partial K}\times\frac{K}{M}$$

$$= \frac{\left[s^2+s(p+K\alpha)+K\right]1-K.(s\alpha+1)}{\left[s^2+s(P+K\alpha)+K\right]^2}\times\frac{K}{\frac{K}{s^2+s(p+K\alpha)+K}}$$

$$= \frac{s^2+sp}{s^2+s(p+K\alpha)+K}$$

$$= \frac{s^2+3s}{s^2+s\left[3+(0.14\times 12)\right]+12}$$

$$S_K^M = \frac{s^2+3s}{s^2+4.68s+12}$$

Sensitivity with respect to P:

$$S_P^M = \frac{\partial M}{\partial P}.\frac{P}{M}$$

$$= \frac{-sK}{\left[s^2+s(p+K\alpha)+K\right]^2}\times\frac{p}{\frac{K}{s^2+s(p+K\alpha)+K}}$$

$$= \frac{-sp}{s^2+s(p+K\alpha)+K} = \frac{-3s}{s^2+4.68s+12}$$

Sensitivity with respect to α:

$$S_{\alpha}^{M} = \frac{\partial M}{\partial \alpha} \cdot \frac{\alpha}{M}$$

$$= \frac{-sK^2}{\left[s^2+s(p+K\alpha)+K\right]^2} \cdot \frac{\alpha}{\dfrac{K}{s^2+s(p+K\alpha)+K}}$$

$$= \frac{-sK\alpha}{s^2+s(p+K\alpha)+K}$$

$$S_{\alpha}^{M} = \frac{-1.68s}{s^2+4.68s+12}$$

(ii) O.L.T.F $G(s)H(s) = \dfrac{K}{(s+2)^2(s+3)}$

$$G.M = \frac{1}{|G(j\omega)H(j\omega)|_{\omega=\omega_{pc}}}$$

For ω_{pc}, $\angle G(j\omega)H(j\omega) = -180°$

$$-2\tan^{-1}\left(\frac{\omega}{2}\right) - \tan^{-1}\left(\frac{\omega}{3}\right) = -180°$$

$$\tan^{-1}\left(\frac{\omega}{1-\dfrac{\omega^2}{4}}\right) + \tan^{-1}\left(\frac{\omega}{3}\right) = 180°$$

$$\left(\frac{\omega}{1-\dfrac{\omega^2}{4}}\right) + \left(\frac{\omega}{3}\right) = 0$$

$$\frac{\omega^2}{4} - 1 = 3$$

$$\omega^2 = 16$$

Phase crossover frequency, $\omega_{pc} = 4$ rad/sec

Given G.M ≥ 3

$$\frac{(\omega^2+4)\sqrt{\omega^2+9}}{K} \geq 3$$

$$\frac{(16+4)(5)}{3} \geq K$$

$$K \leq \frac{100}{3}$$

02. (b)

Sol: The periodic waveform shown in Figure, with a period T = 2π can be expressed as:

$$x(t) = \begin{cases} A & ;\ 0 \le t \le \pi \\ -A & ;\ \pi \le t \le 2\pi \end{cases}$$

Let, $t_0 = 0$

$\therefore\ t_0 + T = 2\pi$ and

Fundamental frequency $\omega_0 = \frac{2\pi}{T} = \frac{2\pi}{2\pi} = 1$

Exponential Fourier series

$$C_0 = \frac{1}{T}\int_0^T x(t)dt = \frac{1}{2\pi}\left(\int_0^\pi A dt + \int_\pi^{2\pi} -A dt\right) = \frac{A}{2\pi}\left[(t)_0^\pi - (t)_\pi^{2\pi}\right] = 0$$

$$C_n = \frac{1}{T}\int_0^T x(t)e^{-jn\omega_0 t}dt \quad = \frac{1}{2\pi}\left(\int_0^\pi Ae^{-jnt}dt + \int_\pi^{2\pi} -Ae^{-jnt}dt\right) = \frac{A}{2\pi}\left[\left(\frac{e^{-jnt}}{-jn}\right)_0^\pi - \left(\frac{e^{-jnt}}{-jn}\right)_\pi^{2\pi}\right]$$

$$= -\frac{A}{j2n\pi}\left[\left(e^{-jn\pi} - e^0\right) - \left(e^{-j2n\pi} - e^{-jn\pi}\right)\right]$$

$$= -\frac{A}{j2n\pi}\left[\left[(-1)^n - 1\right] - \left[1 - (-1)^n\right]\right] = -\frac{A}{2jn\pi}\left[-2 + 2(-1)^n\right]$$

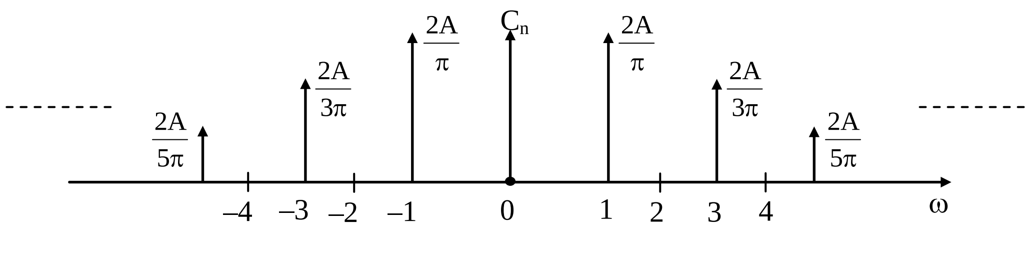
$$= \frac{A}{jn\pi}\left[1 - (-1)^n\right]$$

$$\therefore\ C_n = \begin{cases} -j\frac{2A}{n\pi} & ;\text{ for odd n} \\ 0 & ;\text{for even n} \end{cases}$$

$$\therefore\ x(t) = C_0 + \sum_{\substack{n=-\infty \\ n\neq 0}}^{\infty} C_n e^{jn\omega_0 t} = \sum_{n=-\infty}^{\infty} -j\frac{2A}{n\pi}e^{jnt} \qquad \text{for odd n}$$

$$\therefore\ C_0 = 0, C_1 = C_{-1} = \frac{2A}{\pi}, C_3 = C_{-3} = \frac{2A}{3\pi}, C_5 = C_{-5} = \frac{2A}{5\pi}$$

The frequency spectrum is shown in Figure



02. (c)

Sol:

(i) Process address space:

The process address space is the set of logical addresses that a process references in its code. For example, when 32-bit addressing is in use, addresses can range from 0 to 0x7fffffff; that is, 2^{31} possible numbers, for a total theoretical size of 2 gigabytes.

The operating system takes care of mapping the logical addresses to physical addresses at the time of memory allocation to the program. There are three types of addresses used in a program before and after memory is allocated.

Memory Addresses & Description

1. Symbolic addresses:
The addresses used in a source code. The variable names, constants and instruction labels are the basic elements of the symbolic address space.

2. Relative addresses:
At the time of compilation, a compiler converts symbolic addresses into relative addresses.

3. Physical addresses:
The loader generates these addresses at the time when a program is loaded into main memory.
Virtual and physical addresses are the same in compile-time and load-time address-binding schemes. Virtual and physical addresses differ in execution-time address-binding scheme.

(ii) Static vs Dynamic Loading

The choice between Static or Dynamic Loading is to be made at the time of computer program being developed. If you have to load your program statically, then at the time of compilation, the complete programs will be compiled and linked without leaving any external program or module dependency. The linker combines the object program with other necessary object modules into an absolute program, which also includes logical addresses.
If you are writing a dynamically loaded program, then your compiler will compile the program and for all the modules which you want to include dynamically, only references will be provided and rest of the work will be done at the time of execution.
At the time of loading, with static loading, the absolute program (and data) is loaded into memory in order for execution to start.
If you are using dynamic loading, dynamic routines of the library are stored on a disk in relocatable form and are loaded into memory only when they are needed by the program.

(iii) Static vs Dynamic linking

As explained above, when static linking is used, the linker combines all other modules needed by a program into a single executable program to avoid any runtime dependency.
When dynamic linking is used, it is not required to link the actual module or library with the program, rather a reference to the dynamic module is provided at the time of compilation and linking. Dynamic Link Libraries (DLL) in Windows and Shared Objects in Unix are good examples of dynamic libraries.

(iv) Fragmentation in OS

As processes are loaded and removed from memory, the free memory space is broken into little pieces. It happens after sometimes that processes cannot be allocated to memory blocks considering their small size and memory blocks remains unused. This problem is known as Fragmentation.

Fragmentation is of two types:

1. **External fragmentation:**
   Total memory space is enough to satisfy a request or to reside a process in it, but it is not contiguous, so it cannot be used.

2. **Internal fragmentation:**
   Memory block assigned to process is bigger. Some portion of memory is left unused, as it cannot be used by another process.

03. (a)

Sol: (i) Principle of Argument states that let F(s) be an analytic function and if an arbitrary closed clockwise contour is chosen in s-plane, so that F(s) is analytic at every point on the closed contour in s-plane then the corresponding F(s) plane contour mapped in the F(s) plane will encircle the origin, N times in anticlockwise direction, where N is the difference between the number of poles and number of zeros of F(s) that are encircled by the chosen closed contour in s-plane mathematically, it is expressed as, N = P – Z

(ii) The loop transfer function, G(s) H(s) is expressed as

$G(s)\,H(s) = [1 + G(s)\,H(s)] - 1 = F(s) - 1$

From above equation, it can be concluded that contour of F(s) drawn with respect to origin of F(s) plane is same as contour of F(s) – 1 drawn with respect to –1 + j0 of F(s) plane. We know that F(s) = 1 + G(s) H(s) is the characteristic equation, origin (0, 0) is the critical point for F(s).

$\therefore$ (–1, 0) is the critical point for F(s) –1 = G(s)H(s)

$\therefore$ The critical point in using the Nyquist criterion is (–1, j0) in G(s)H(s) plane and not the origin (0, j0).

(iii) For a minimum phase transfer function, no right hand poles for G(s)H(s) should be present.

For stability, the polar plot of a minimum phase system should not enclose (–1, j0) critical point

$\therefore$ Polar plot is sufficient to determine the stability of a system.

03. (b)

Sol: The energy of a signal x(t) is given by

$$E = \int_{-\infty}^{\infty} |x(t)|^2 dt = \int_{-\infty}^{\infty} |e^{-4t}u(t)|^2 dt$$

$$= \int_{0}^{\infty} |e^{-4t}|^2 dt = \int_{0}^{\infty} e^{-8t} dt = \left[\frac{e^{-8t}}{-8}\right]_0^{\infty} = \frac{1}{8}\,\text{joule}$$

Now, according to Parseval's theorem, we have

$$E = \frac{1}{2\pi}\int_{-\infty}^{\infty} |X(\omega)|^2 d\omega$$

$$X(\omega) = F[x(t)] = \int_{-\infty}^{\infty} e^{-4t}u(t)e^{-j\omega t} dt$$

$$= \int_{0}^{\infty} e^{-(4+j\omega)t} dt = \left[\frac{e^{-(4+j\omega)t}}{-(4+j\omega)}\right]_0^{\infty} = \frac{1}{4+j\omega}$$

$$|X(\omega)| = \frac{1}{\sqrt{4^2+\omega^2}}$$

$$\therefore\quad E = \frac{1}{2\pi}\int_{-\infty}^{\infty} \frac{1}{4^2+\omega^2} d\omega = \frac{1}{8\pi}\tan^{-1}\left[\frac{\omega}{4}\right]_{-\infty}^{\infty}$$

$$= \frac{1}{8\pi}\left[\frac{\pi}{2} - \left(\frac{-\pi}{2}\right)\right] = \frac{1}{8}\,\text{joule}$$

Thus, from the above equations, we see that energy is same in both the cases.
Hence Parseval's theorem is verified.

03. (c)

Sol: 1. First Approach (Using Logical-AND operator)

```
#include <stdio.h>
void  main()
{
   int A, B, C;
  printf("Enter the numbers A, B and C: ");
  scanf("%d %d %d", &A, &B, &C);
  if (A >= B && A >= C)
       printf("%d is the largest number.", A);
  if (B >= A && B >= C)
       printf("%d is the largest number.", B);
  if (C >= A && C >= B)
       printf("%d is the largest number.", C);
}
```

2. Second Approach (Using Nested if-else)

```
#include <stdio.h>
void main()
{
    int A, B, C;
      printf("Enter three numbers: ");
      scanf("%d %d %d", &A, &B, &C);
      if (A >= B) {
            if (A >= C)
                  printf("%d is the largest number.", A);
            else
             printf("%d is the largest number.", C);
                 }
    else{
             if (B >= C)
                  printf("%d is the largest number.", B);
              else
                  printf("%d is the largest number.", C);
            }
}
```

03. (d)

Sol: From Deal Grove growth law,

$x^2 + Ax = B(t+t_o)$

Where x - thickness of oxide layer

t - time required

t_o = initial time

The above expression can be rearranged as

$$t = \frac{x^2}{B} + \frac{x}{B/A} - t_o \quad \text{\_\_\_\_(1)}$$

(i) Dry oxidation:

Given that initial oxide thickness $x_o = 0.020\mu m$

$$t_o = \frac{x_o^2}{B} + \frac{x_o}{B/A}$$

$$= \frac{(0.020\mu m)^2}{1.11\times10^{-6}\mu m^2/sec} + \frac{0.020\mu m}{2.63\times10^{-6}\mu m/sec}$$

$= 360.36 + 7604.56$

$t_o = 7965$ sec

now from equation (1)

$$t = \frac{x^2}{B} + \frac{x}{B/A} - t_o$$

$$= \frac{(0.120)^2}{1.11\times10^{-6}} + \frac{0.120}{2.63\times10^{-6}} - 7965$$

$= 50630$ sec

T = 14.06hr

(ii) Wet oxidation:

Since no, initial oxidation is given in wet oxidation,

$$t = \frac{x^2}{B} + \frac{x}{B/A}$$

$$= \frac{(0.120)^2}{4.31\times10^{-5}} + \frac{0.120}{4.20\times10^{-5}}$$

$= 334.1 + 2857.1$

t = 3189 sec

t = 0.886hr

Comment:

Clearly, wet oxidation is much faster than dry oxidation in general, a processing time of 14.06hrs is unacceptably long in a modern manufacturing environment. This result clearly illustrates the "Throughput" advantage of wet oxidation over dry oxidation. However, dry oxidation is advantageous for fabrication of very thin oxides since the interface state density for dry oxides is generally lower.

04. (a)

Sol: $x(n) \leftrightarrow X(k)$

$$x((n-m))_N \leftrightarrow e^{-j\frac{2\pi}{N}mk} X(k)$$

$$\text{DFT}\left[x((n-m))_N\right] = \sum_{n=0}^{N-1} x((n-m))_N e^{-j\frac{2\pi}{N}nk}$$

$$= \sum_{n=0}^{m-1} x((n-m))_N e^{-j\frac{2\pi}{N}nk} + \sum_{n=m}^{N-1} x((n-m))_N e^{-j\frac{2\pi}{N}nk}$$

Since, $x((n-m))_N = x(N-m+n)$

$$\sum_{n=0}^{m-1} x((n-m))_N e^{-j\frac{2\pi}{N}nk} = \sum_{n=0}^{m-1} x(N-m+n) e^{-j\frac{2\pi}{N}nk}$$

Let, N – m + n = ℓ

$$\sum_{n=0}^{m-1} x((n-m))_N e^{-j\frac{2\pi}{N}nk} = \sum_{\ell=N-m}^{N-1} x(\ell) e^{-j\frac{2\pi}{N}(N-m+\ell)k}$$

$$= \sum_{\ell=N-m}^{N-1} x(\ell) e^{-j\frac{2\pi}{N}(\ell+m)k}$$

Similarly, $\sum_{n=m}^{N-1} x((n-m))_N e^{-j\frac{2\pi}{N}nk} = \sum_{\ell=0}^{N-1-m} x(\ell) e^{-j\frac{2\pi}{N}(m+\ell)k}$

So, DFT $[x((n-m))_N] = \sum_{\ell=0}^{N-m-1} x(\ell) e^{-j\frac{2\pi}{N}(m+\ell)k} + \sum_{\ell=N-m}^{N-1} x(\ell) e^{-j\frac{2\pi}{N}(m+\ell)k}$

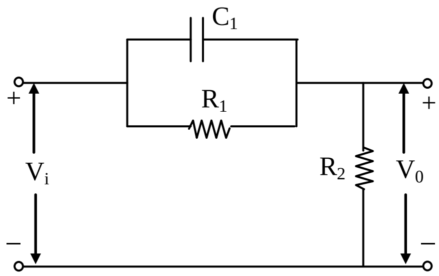
$$= e^{-j\frac{2\pi}{N}mk} \sum_{\ell=0}^{N-1} x(\ell) e^{-j\frac{2\pi}{N}\ell k}$$

$$= e^{-j\frac{2\pi}{N}mk} X(k)$$

04. (b)

Sol: Lead Compensator: The output of the lead compensator always leads with respect to the input. In other words, lead compensator always produces the positive phase.

The lead compensator can be realized by the following RC network.



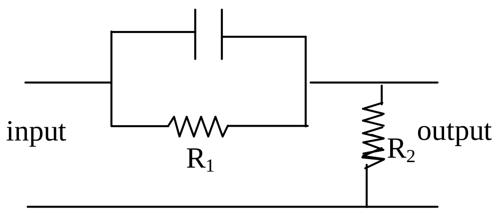
Effects of lead Compensator:

1. The lead compensator adds a zero to the right of the pole, which causes increased damping.
2. The increase in damping means less overshoot, less rise time and less delay time. Due to this, the transient performance is increased
3. It improves the gain margin and phase margin of system hence, the relative stability is improved.
4. It increases the bandwidth of the system more and gives quick response. Steady state error is not affected.

Limitations:

1. This lead compensator requires additional increase in gain (a) to offset (nullify) the attenuator.
2. To obtain large gain, more number of amplifiers are to be used, which increases the space, weight and cost.
3. Sometimes, more bandwidth may not be required, which makes the system more noisy.
4. From a single lead compensator, the maximum phase lead obtained is nearly 40^0 to 60^0. If required phase is more than 60^0, multiple stages are used.

Lead controller:

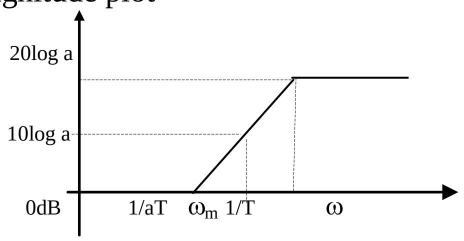


$$TF = \left(\frac{1+aTs}{1+Ts}\right)$$

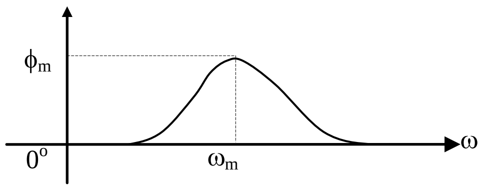
Where $T = \frac{R_1R_2}{R_1+R_2}C$

$$a = \frac{R_1+R_2}{R_2}$$

Magnitude plot



Phase plot



04. (c)

Sol:

(i) **Enum:** Enumeration (or enum) is a user defined data type in C. It is mainly used to assign names to integral constants, the names make a program easy to read and maintain.
Following is an example of enum declaration:
enum flag{constant1, constant2, constant3, ... };
The name of enumeration is "flag" and the constant are the values of the flag. By default, the values of the constants are as follows:
constant1 = 0, constant2 = 1, constant3 = 2 and so on.
Variables of type enum can also be defined. They can be defined in two ways:
enum week{Mon, Tue, Wed};
enum week day;
Or
enum week{Mon, Tue, Wed}day;
In both of the above cases, "day" is defined as the variable of type week.

(ii) **Typedef:**
The C programming language provides a keyword called typedef, which you can use to give a type a new name. Following is an example to define a term BYTE for one-byte numbers

typedef unsigned char BYTE;

After this type definition, the identifier BYTE can be used as an abbreviation for the type unsigned char, for example..

BYTE b1, b2;

By convention, uppercase letters are used for these definitions to remind the user that the type name is really a symbolic abbreviation, but you can use lowercase, as follows

typedef unsigned char byte;

You can use typedef to give a name to your user defined data types as well. For example, you can use typedef with structure to define a new data type and then use that data type to define structure variables directly as follows

```
# include <stdio.h>
# include <string.h>
typedef struct Books {
  char title[50];
  char author[50];
  char subject[100];
  int book_id;
} Book;

int main( ) {
Book book;
strcpy( book.title, "C Programming");
strcpy( book.author, "Nuha Ali");
strcpy( book.subject, "C Programming Tutorial");
book.book_id = 6495407;
printf( "Book title : %s\n", book.title);
printf( "Book author : %s\n", book.author);
printf( "Book subject : %s\n", book.subject);
printf( "Book book_id : %d\n", book.book_id);
 return 0;
}
```

When the above code is compiled and executed, it produces the following result

```
Book  title : C Programming
Book  author : Nuha Ali
Book  subject : C Programming Tutorial
Book  book_id : 6495407
```

04. (d)

Sol: The given second order system

$$a\frac{d^2\theta}{dt^2} + b\frac{d\theta}{dt} + c\theta = F(t)$$

Taking Laplace transform on both sides

$$as^2\theta(s) + bs\theta(s) + c\theta(s) = F(s)$$

$$\Rightarrow \left(as^2 + bs + c\right)\theta(s) = F(s)$$

Transfer function $= \dfrac{\theta(s)}{F(s)} = \dfrac{1}{as^2 + bs + c}$

$\therefore$ Step response $\theta(S) = \left(\dfrac{1}{as^2+bs+c}\right)\dfrac{1}{s} = \left(\dfrac{1/a}{s^2+\frac{b}{a}s+\frac{c}{a}}\right)\left(\dfrac{1}{s}\right)$

$$= \frac{1}{c}\left(\frac{\frac{c}{a}}{s^2+\frac{b}{a}s+\frac{c}{a}}\right)\frac{1}{s}$$

Compare with standard second system $\dfrac{\omega_n^2}{s\left(s^2+2\xi s\omega_n+\omega_n^2\right)}$, we get response as

$$\theta(t) = 1-\frac{e^{-\xi\omega_n t}}{\sqrt{1-\xi^2}}(\sin\omega_d t+\phi)$$

$$\omega_n = \sqrt{\frac{c}{a}}$$

$$2\xi\omega_n = \frac{b}{a}$$

$$\xi = \frac{1}{2}\frac{b}{\sqrt{ac}}$$

$$\omega_d = \omega_n\sqrt{1-\xi^2} = \sqrt{\frac{c}{a}}\sqrt{1-\frac{1}{4}\frac{b^2}{ac}} = \sqrt{\frac{c}{a}}\sqrt{\frac{4ac-b^2}{4ac}}$$

$$\omega_d = \frac{1}{2a}\sqrt{4ac-b^2}$$

Let assume $\xi = \dfrac{1}{2}\dfrac{b}{\sqrt{ac}} < 1$

Then

$$\theta(t) = \frac{1}{c}\left[1-\frac{e^{\frac{-bt}{2a}}}{\sqrt{\frac{4ac-b^2}{4ac}}}\left(\sin\left[\frac{1}{2a}\sqrt{4ac-b^2}\right]t+\cos^{-1}\left(\frac{b}{2\sqrt{ac}}\right)\right)\right]$$

05. (a)

Sol: In direct mapped cache the main memory address is divided into 3 parts (fields) as follows:

$\longleftarrow$ Main memory address $\longrightarrow$

Tag	Cache block number	Byte offset

Block size = 64B = 2^6 B $\Rightarrow$ Byte offset = 6-bits

Number of blocks in cache memory $= \dfrac{\text{cache size}}{\text{block size}}$

$$= \frac{256\text{MB}}{64\text{B}}$$

$$= \frac{2^8.2^{20}}{2^6}$$

$$= 2^{22}$$

Hence cache block no. = 22-bits
Tag size = 32–(22+6) = 4-bits

32		
Tag	Cache block number	Byte offset
4	22	6

Tag directory (Meta-data) size = no. of blocks in cache * 1 entry size

$$= 2^{22} * (\text{Tag} + \text{extra bits})$$
$$= 2^{22} * (4+2) \text{ bits}$$
$$= 2^{22} * 6 \text{ bits}$$
$$= 24 * 2^{20} \text{ bits}$$
$$= 24 \text{ Mbits}$$
$$= 0.3 \text{ Mbytes}$$

05. (b)

Sol: The time required is

(i) 50 nano seconds to get the page number from associative memory and 750 nano-seconds to read the desired word from memory.
Time = 50+750= 800 nano seconds.

(ii) Now the time when not in associative memory is
Time = 50+750+750= 1550 nano seconds
One memory access extra is required to read the page table from memory.

(iii) Effective access time
= Page number in associative memory + Page number not in associated memory.
Page number in associative memory = 0.8 * 800.
Page number not in associated memory = 0.2 * 1550.
Effective access time = 0.8 * 800 + 0.2 * 1550 = 950 nano seconds

05. (c)

Sol: Given, Input $x(t) = te^{-3t}\,u(t)$

$$X(s) = \frac{1}{(s+3)^2}$$

Impulse response $h(t) = 2e^{-4t}\,u(t)$

$$\therefore \quad H(s) = \frac{2}{s+4}$$

We know that, Output $y(t) = x(t) * h(t)$

$$\therefore \; Y(s) = X(s)\,H(s)$$

So, Output $y(t) = L^{-1}[X(s)\,H(s)]$

$$Y(s) = X(s)H(s) = \left[\frac{1}{(s+3)^2}\right]\left(\frac{2}{s+4}\right) = \frac{2}{(s+3)^2(s+4)}$$

Taking partial fractions, we have

$$Y(s) = \frac{2}{(s+3)^2(s+4)} = \frac{A}{(s+3)^2} + \frac{B}{s+3} + \frac{C}{s+4}$$

$$Y(s) = \frac{2}{(s+3)^2} - \frac{2}{s+3} + \frac{2}{s+4}$$

Taking inverse Laplace transform on both sides, we have the output
$y(t) = 2te^{-3t}\,u(t) - 2e^{-3t}\,u(t) + 2e^{-4t}\,u(t)$

05. (d)

Sol: Given open loop transfer function

$$G(s) = \frac{K(s+4)}{s(s+1}$$

No. of root locus branches = 2(P > Z)

No. of Asymptotes N = |P – Z| = 1

$$\text{Angle of Asymptotes} = \frac{(2\ell+1)180^0}{P-Z} \quad l = 0$$

$$= \frac{(2(0)+1)180^0}{1} = 180^0$$

Here, only one asymptote is present, therefore centroid is not required.

Break Point CE is $1 + KG_1(s)H_1(s) = 0$

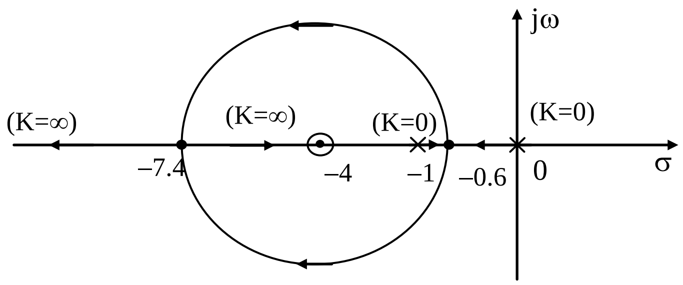


Fig. Root Locus diagram

$$K = \frac{-1}{G_1(s)H_1(s)}$$

$$G_1(s)H_1(s) = \frac{s+4}{s(s+1)}$$

$$\frac{dK}{ds} = \frac{d}{ds}\left(\frac{-1}{G_1(s)H_1(s)}\right) = 0$$

$$\frac{d}{ds}\left(-\frac{s(s+1)}{s+4}\right) = 0$$

$$\Rightarrow \frac{(s+4)(2s+1)-(s^2+s)(1)}{(s+4)^2} = 0$$

$\Rightarrow s^2 + 8s + 4 = 0$

$\therefore$ s = –0.6 and s = –7.4

The system is critically damped when s= –0.6 and s = –7.4 (roots are real and equal)

$$K = \frac{\text{Product of distances from poles}}{\text{Product of distances from zeros}}$$

$$K = \frac{(0.6)(0.4)}{3.4} = 0.07 \text{ (at } s = -0.6)$$

$$K = \frac{(7.4)(6.4)}{3.4} = 13.92 \text{(at } s = -7.4)$$

05. (e)

Sol: Fermi potential for p-substrate and n-type poly silicon are given by

$$\phi_F(\text{substrate}) = \frac{KT}{q}\ln\left(\frac{n_i}{N_A}\right) = 0.026\ln\left(\frac{1.45\times10^{10}}{2\times10^{16}}\right) = -0.367\text{V}$$

$$\phi_F(\text{gate}) = \frac{KT}{q}\ln\left(\frac{N_D}{n_i}\right) = 0.026\ln\left(\frac{2\times10^{19}}{1.45\times10^{10}}\right) = 0.547\text{V}$$

The work function difference,

$$\phi_{ms} = \phi(\text{substrate}) = \phi(\text{gate})$$

$$= -0.367 - 0.547$$

$\phi_{ms} = -0.914\text{V}$

Oxide capacitance per unit area

$$C_{ox} = \frac{\varepsilon_{ox}}{t_{ox}} = \frac{3.97\times8.85\times10^{-14}}{300\times10^{-8}} = 1.17\times10^{-7}\,\text{F/cm}^2$$

The depletion layer charge,

$$Q_{bo} = -\sqrt{2qN_A\varepsilon_{si}\,|2\phi_F(\text{substrate})|}$$

$$= -\sqrt{2\times1.6\times10^{-19}\times2\times10^{16}\times11.7\times8.85\times10^{-14}\,|-2\times-0.367|}$$

$Q_{bo} = -6.97\times10^{-8}\text{c/cm}^2$

The fixed oxide charge, $Q_{ox} = qN_{ox} = 1.6\times10^{-19}\times10^{10} = 1.6\times10^{-9}\text{c/cm}^2$

Now, threshold voltage $V_t = \phi_{ms} - 2\phi_F - \frac{Q_{bo}}{C_{ox}} = \frac{Q_{ox}}{C_{ox}}$

$$= -0.914 - 2\times(-0.367) - \left(\frac{-6.97\times10^{-8}}{1.17\times10^{-7}}\right) - \left(\frac{1.6\times10^{-9}}{1.17\times10^{-7}}\right)$$

$$= -0.914 + 0.734 + 0.594 - 0.0138$$

$V_t = 0.4002\text{V}$

06. (a)

Sol: Arrange the given transfer function as,

$$\frac{Q(s)}{I(s)} = \frac{1}{J\left[s^2 + \frac{f}{J}s + \frac{K}{J}\right]} = \frac{\left[\frac{1}{J}\right]}{s^2 + \frac{f}{J}s + \frac{K}{J}}$$

Comparing denominator with $s^2 + 2\zeta\omega_n s + \omega_n^2$,

$\omega_n^2 = \frac{K}{J}$ i.e. $\omega_n = \sqrt{\frac{K}{J}}$ ….. (1)

and $2\zeta\omega_n = \frac{f}{J}$ i.e., $\zeta = \frac{f}{2\sqrt{KJ}}$ ….. (2)

Now M_p = 6% i.e., 0.06

$\therefore\ 0.06 = e^{-\pi\zeta/\sqrt{1-\zeta^2}}$

$ln\,(0.06) = \frac{-\pi\zeta}{\sqrt{1-\zeta^2}}$

Solving for ζ, $\zeta = 0.667$ ……. (3)

$$T_p = \frac{\pi}{\omega d} = \frac{\pi}{\omega_n\sqrt{1-\zeta^2}} = 1 \text{ sec}$$

$$\omega_n = \frac{\pi}{\sqrt{1-(0.667)^2}} = 4.2165 \text{ rad/sec. … (4)}$$

The Laplace transform of output is Q(s).

Now input is step of 10 Nm hence corresponding Laplace transform is,

$$I(s) = \frac{10}{s}$$

$$\therefore \quad \frac{Q(s)}{\left(\frac{10}{s}\right)} = \frac{1}{Js^2+fs+K} \Rightarrow Q(s) = \frac{1}{s\left(Js^2+fs+K\right)}$$

The steady state of output can be obtained by final value theorem.

Steady state output = $\underset{s\to 0}{\text{Lim}}\, s\,Q(s)$

$$0.5 = \underset{s\to 0}{\text{Lim}} \frac{s.10}{s\left(Js^2+fs+K\right)} = \frac{10}{k}$$

$\therefore \quad K = 20$

Equating (1) and (4), $4.2165 = \sqrt{\frac{K}{J}}$

$$4.2165 = \sqrt{\frac{20}{J}}$$

J = 1.1249

From equation (2), $0.667 = \frac{f}{2\sqrt{KJ}}$

$$0.067 = \frac{f}{2\sqrt{20\times 1.1249}}$$

f = 6.3274

06. (b)

Sol: Given $\quad x(t) = C_0 + \sum_{n=1}^{\infty} C_n \cos(n\omega_0 t + \theta_n)$

(i) Autocorrelation:

Autocorrelation of function x(t) is given by

$$R(\tau) = \underset{T\to\infty}{\text{Lt}} \frac{1}{T}\int_{-T/2}^{T/2} x(t)x(t+\tau)dt$$

$$R(\tau) = \underset{T\to\infty}{\text{Lt}} \frac{1}{T}\int_{-T/2}^{T/2}\left[C_0 + \sum_{n=1}^{\infty} C_n\cos(n\omega_0 t+\theta_n)\right]\left[C_0 + \sum_{n=1}^{\infty} C_n\cos(n\omega_0 t + n\omega_0\tau + \theta_n)\right]$$

$$= \underset{T\to\infty}{\text{Lt}} \frac{1}{T}\int_{-T/2}^{T/2}\left[C_0^2 + C_0C_n\sum_{n=1}^{\infty}\cos(n\omega_0 t + n\omega_0\tau+\theta_n) + \sum_{n=1}^{\infty} C_0C_n\cos(n\omega_0 t+\theta_n)\right.$$

$$\left. + C_n^2\sum_{n=1}^{\infty}\cos(n\omega_0 t+\theta_n)\cos(n\omega_0 t + n\omega_0\tau+\theta_n)dt\right]$$

$$= \underset{T\to\infty}{Lt}\frac{1}{T}\int_{-T/2}^{T/2}C_0^2dt + \underset{T\to\infty}{Lt}\frac{1}{T}\int_{-T/2}^{T/2}C_0C_n\sum_{n=1}^{\infty}\cos(n\omega_0t+n\omega_0\tau+\theta_n)dt$$

$$+ \underset{T\to\infty}{Lt}\frac{1}{T}\int_{-T/2}^{T/2}\sum_{n=1}^{\infty}C_0C_n\cos(n\omega_0t+\theta_n)dt$$

$$+ \underset{T\to\infty}{Lt}\frac{1}{T}\int_{-T/2}^{T/2}\sum_{n=1}^{\infty}C_n^2\cos(n\omega_0t+\theta_n)\cos(n\omega_0t+n\omega_0\tau+\theta_n)dt$$

$$= C_0^2+0+0+\sum_{n=1}^{\infty}\underset{T\to\infty}{Lt}\frac{C_n^2}{2T}\int_{-T/2}^{T/2}2\cos(n\omega_0t+\theta_n)\cos(n\omega_0t+n\omega_0\tau+\theta_n)dt$$

$$= C_0^2+\sum_{n=1}^{\infty}\underset{T\to\infty}{Lt}\frac{C_n^2}{2T}\int_{-T/2}^{T/2}[\cos(2n\omega_0t+n\omega_0\tau+2\theta_n)+\cos(n\omega_0\tau)]dt$$

$$= C_0^2+\sum_{n=1}^{\infty}\underset{T\to\infty}{Lt}\frac{C_n^2}{2T}\int_{-T/2}^{T/2}\cos n\omega_0\tau dt = C_0^2+\sum_{n=1}^{\infty}\underset{T\to\infty}{Lt}\frac{C_n^2}{2T}\cos(n\omega_0\tau)\left(\frac{T}{2}+\frac{T}{2}\right)$$

$$= C_0^2+\sum_{n=1}^{\infty}\underset{T\to\infty}{Lt}\frac{C_n^2}{2T}\cos n\omega_0\tau(T)$$

$$\therefore\ R(\tau)=C_0^2+\frac{1}{2}\sum_{n=1}^{\infty}C_n^2\cos n\omega_0\tau$$

Power spectral density (PSD) and autocorrelation form a Fourier transform pair

$$\therefore\ PSD = F[R(\tau)]$$

$$PSD = F\left[C_0^2+\frac{1}{2}\sum_{n=1}^{\infty}C_n^2\cos n\omega_0\tau\right]$$

$$= C_0^2[2\pi\delta(\omega)]+\frac{1}{2}\sum_{n=1}^{\infty}C_n^2\pi[\delta(\omega-n\omega_0)+\delta(\omega+n\omega_0)]$$

06. (c)

Sol:

(i) Memory size = 4GB

word size = 64-bits = 8 Bytes

Memory size (in words) = $\frac{4GB}{8B}$

= 0.5G words

= 512 M words

= 2^{29} words

Address size = $\log_2(2^{29})$ = 29.bits

(ii) Drawbacks in each Scheduling Policy

A) FIFO replacement policy:

1. A page which is being accessed quite often may also get replaced because it arrived earlier than those present
2. Ignores locality of reference. A page which was referenced last may also get replaced, although there is high probability that the same page may be needed again.

B) LRU replacement policy:

1. A good scheme because focuses on replacing dormant pages and takes care of locality of reference, but
2. A page which is accessed quite often may get replaced because of not being accessed for some time, although being a significant page may be needed in the very next access.

(iii) In direct mapped cache main memory address is divided into 3 fields as follows:

← Main memory address →

Tag	Cache block no.	Byte offset

Main memory address = 32-bits given
Tag = 16-bits given
Cache block no. = 12-bits (because no. of blocks in cache = 2^{12})
Byte offset = 32-(16+12) = 4-bits
Block size = 2^4 = 16 Bytes

07. (a)

Sol: Hanning window: $w_{Hn}(n) = 0.5 + 0.5\cos\left(\frac{2\pi n}{N-1}\right)$ for $-\left(\frac{N-1}{2}\right) \le n \le \left(\frac{N-1}{2}\right)$

= 0 otherwise

N = 11
$w_{Hn}(0) = 1$
$w_{Hn}(1) = w_{Hn}(-1) = 0.9045$
$w_{Hn}(2) = w_{Hn}(-2) = 0.655$
$w_{Hn}(3) = w_{Hn}(-3) = 0.345$
$w_{Hn}(4) = w_{Hn}(-4) = 0.0945$
$w_{Hn}(5) = w_{Hn}(-5) = 0$

The filter coefficients are obtained as

$$h_d(n) = \frac{1}{2\pi}\int_{-\pi}^{\pi} H_d(e^{j\omega}).e^{j\omega n}d\omega = \frac{\sin \pi n - \sin\left(\frac{\pi}{4}n\right)}{\pi n}$$

$$h_d(0) = \underset{n\to 0}{\text{Lt}}\ \frac{\sin(n\pi) - \sin\left(\frac{n\pi}{4}\right)}{n\pi} = 1 - \frac{1}{4} = 0.75$$

$h_d(-1) = h_d(1) = -0.225$
$h_d(-2) = h_d(2) = -0.159$
$h_d(-3) = h_d(3) = -0.075$
$h_d(-4) = h_d(4) = 0$
$h_d(-5) = h_d(5) = 0.045$

The filter coefficients using hanning window are

$h(n) = h_d(n).w_{Hn}(n)$ for $-5 \le n \le 5$

$= 0$ otherwise

$h(0) = h_d(0)\ w_{Hn}(0) = 0.75$

$h(1) = h(-1) = h_d(1)\ w_{Hn}(1) = -0.204$

$h(2) = h(-2) = h_d(2)\ w_{Hn}(2) = -0.104$

$h(3)= h(-3) = h_d(3)\ w_{Hn}(3) = -0.026$

$h(4) = h(-4) = h_d(4)\ w_{Hn}(4) = 0$

$h(5) = h(-5) = h_d(5)\ w_{Hn}(5) = 0$

The transfer function of the filter is $H(z) = h(0) + \sum_{n=1}^{5} h(n)\left[z^n + z^{-n}\right]$

$$H(z) = 0.75 - 0.204\left(z + z^{-1}\right) - 0.104\left[z^2 + z^{-2}\right] - 0.026\left[z^3 + z^{-3}\right]$$

The transfer function of realizable filter is

$$H^1(z) = z^{-5}.H(z) = -0.026z^{-2} - 0.104z^{-3} - 0.204z^{-4} + 0.75z^{-5} - 0.204z^{-6} - 0.104z^{-7} - 0.026z^{-8}$$

The causal filter coefficients are h(0) = h(1) = h(9) = h(10) = 0

h(2) = h(8) = – 0.026

h(3) = h(7) = – 0.104

h(4) = h(6) = – 0.204

h(5) = 0.75

07. (b)

Sol:

(i) (A) Logical address will have

3 bits to specify the page number (for 8 pages) .

10 bits to specify the offset into each page (210 =1024 words) = 13 bits.

(B) For 32 (2^5) frames of 1024 words each (Page size = Frame size)

We have 5 + 10 = 15 bits.

(ii) Factors for a good page replacement policy:

1. Context
2. Access count
3. Time of last access
4. Time of arrival

1. **Context:** To avoid replacing most significant pages
2. **Access count along with time of arrival -** Frequently accessed page with respect to the time when the page was brought into memory.
3. **Time of last access –** Since access count along with time of arrival takes care of locality of reference (as a page which has come some time back is accessed quite frequently in proportion, it is high priority that the same page may be called again), thus last time of access is now not that important, although maintain its significance as such and thus should be considered.

07. (c)

Sol: The given transfer function

$$G(s)H(s) = \frac{K}{s(s+2)(s+4)}$$

Characteristic equation

$$1 + G(s)H(s) = 0$$

$s(s^2 + 6s + 8) + K = 0$

$s^3 + 6s^2 + 8s + K = 0$

Apply Routh Hurwitz criteria,

$$\begin{array}{c|cc} s^3 & 1 & 8 \\ s^2 & 6 & K \\ s^1 & \dfrac{48-K}{6} & 0 \\ s^0 & K & \end{array}$$

If there are any sign changes in the first column, the system is unstable. So to be stable $\dfrac{48-K}{6} > 0$,

$K > 0 \Rightarrow K < 48, K > 0$

$\therefore\ 0 < K < 48$

For the system to be stable.

For remaining values of K system is unstable. i.e., $K < 0$, $K > 48$.

At the point of intersection of root loci with imaginary axis system is critically stable.

$\therefore\ 48 - K = 0$

$K = 48$

Substitute K value in second row.

We will get auxiliary equation as

$6s^2 + K = 0$

$6s^2 + 48 = 0$

$s^2 + 8 = 0$

$s = \pm j2\sqrt{2}$

At $s = \pm j2\sqrt{2}$, root loci intersects with imaginary axis.

07. (d)

Sol:

(i)

CPLD	FPGA
1. Predictable performance independent of internal placement and routing	1. Performance depends on the routing implemented for a particular application.
2. Functionality is implemented by PAL like structures	2. Functionality is implemented by look up tables.
3. Suitable for low to medium density designs	3. Suitable for medium to high density designs
4. Regular PAL like architecture	4. More complex and register rich architecture
5. Cross bar type interconnection type fabric	5. Channel based interconnection fabric
6. Can be reprogrammed a limited number of times	6. Can be reprogrammed as many times as possible.

(ii)

CVD	PVD
1. It is preferred for poly silicon layer and silicon nitride	1. It is preferred to produce a metal vapour that can be deposited on electrically conductive materials.
2. It produces hazardous by products	2. Generally no hazardous by products are released or produced.
3. Requires high temperature	3. Requires lower temperature
4. Requires a high vacuum (LPCVD) is most common)	4. Requires high vacuum.
5. Poor directionality	5. directional

07. (e)

Sol: The given characteristic equation

$s^3 + 4s^2 + 8s + 11 = 0$

$$\begin{array}{c|cc} s^3 & 1 & 8 \\ s^2 & 4 & 11 \\ s^1 & 21/4 & 0 \\ s^0 & 11 & \end{array}$$

No. of sign changes in the first column = 0

$\therefore$ Number of right half of s-plane poles = 0

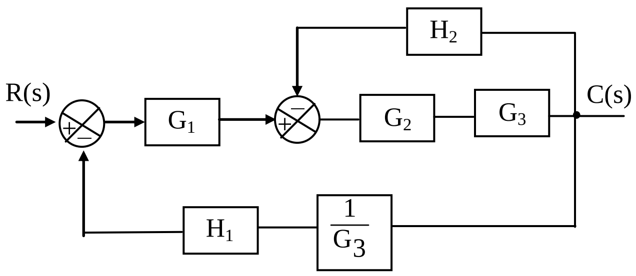
Number of $j\omega$ poles = 0

Number of left half of s-plane poles = 3

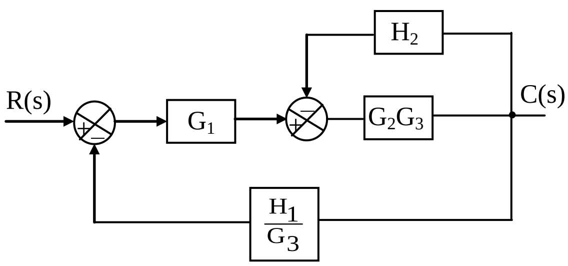
$\therefore$ System is stable.

08. (a)

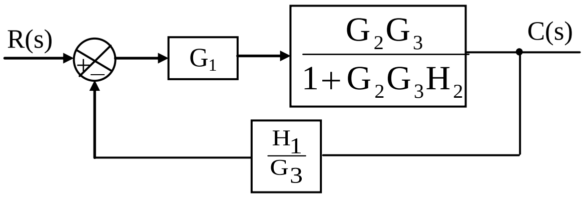
Sol: **Step (i):** Shifting the takeoff point before block G_3 to after block G_3.



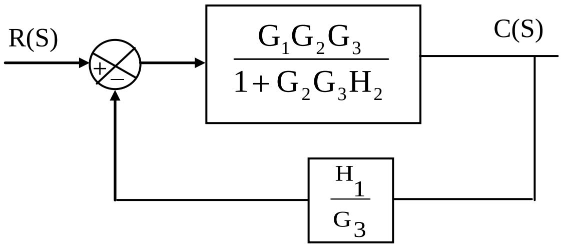
Step (ii): Cascading the blocks G_2 and G_3, H_1 and $1/G_3$.



Step (iii): Eliminate the feedback loop with H_2.



Step (iv): Cascade the forward path blocks.



Step (v): Eliminate the feedback loop with H_1/G_3.

$$\therefore \text{Transfer function, } \frac{C(s)}{R(s)} = \frac{\dfrac{G_1G_2G_3}{1+G_2G_3H_2}}{1+\dfrac{G_1G_2G_3\left(\dfrac{H_1}{G_3}\right)}{1+G_2G_3H_2}}$$

$$\frac{C(s)}{R(s)} = \frac{G_1G_2G_3}{1+G_2G_3H_2+G_1G_2H_1}$$

08. (b)

Sol:

(i) There are three types of misses in cache memory:

1. Compulsory or Cold miss
2. Capacity miss
3. Conflict miss

1. **Compulsory or Cold Miss:** The program has never requested this data before. This is mostly unavoidable. To reduce the number of cold misses, increase the block size.

2. **Capacity Miss:** Occur because blocks are being discarded from cache because cache cannot contain all blocks needed for program execution (program working set is much larger than cache capacity). To reduce the number of capacity misses, increase the cache size.

3. **Conflict Miss:** In the case of set associative or direct mapped block placement strategies, conflict misses occur when several blocks are mapped to the same set or block frame; also called collision misses or interference misses. To reduce the number of cold misses, increase the associativity of the cache.

(ii) Only one time address is sent to main memory. But memory is accessed to access entire cache block. Cache block size is given as 4 words and one cell capacity in main memory is 1 word. Then main memory is accessed 4/1 = 4 times. One content can be transferred in 1 cycle: which is one cell capacity to transfer total 4 words, 4 cycles are needed.

Miss penalty time = Address sending time = 1

+

Memory access time = (4*10)

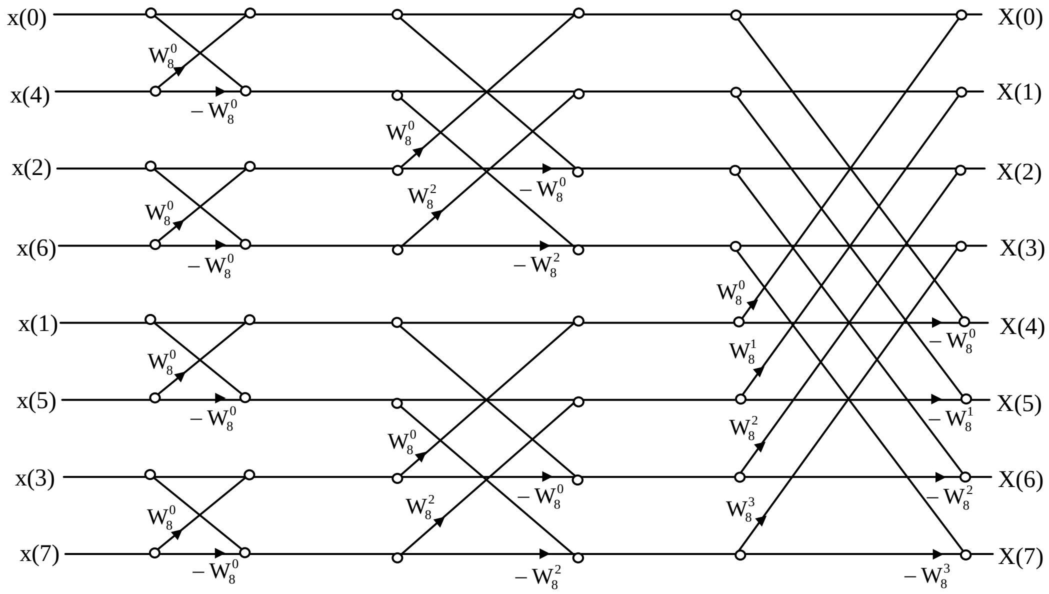
+

Block Transfer time = 4 * 1

45 cycles

08. (c)

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



Inputs	Stage 1 outputs	Stage 2 outputs	Stage 3 outputs
1	1 + 4 = 5	5 + 5 = 10	10+10 = 20
4	1– 4 = –3	–3 + (–j) 1 = –3–j	–3 –j + (0.707–j0.707)(–1–3j) = –5.828 – j2.414
3	3 + 2 = 5	5 – 5 = 0	0
2	3 – 2 = 1	–3 – (–j)1 = –3+j	–3 +j + (–0.707–j0.707)(–1+3j) = –0.172–j0.414
2	2 + 3 = 5	5 + 5 = 10	10–10=0
3	2 – 3 = –1	–1 +(–j) 3 = – 1–j3	–3 –j – (0.707–0.707)(–1–3j) = –0.172 + j0.414
4	4 + 1 = 5	5 – 5 = 0	0
1	4 – 1 = 3	–1 –(–j) 3 = – 1+j3	(–3 +j) – (–0.707–0.707)(–1+3j) = –5.828+j2.414

X(k) = {20, –5.828–j2.414, 0, –0.172–j0.414, 0, –0.172+j0.414, 0, –5.828+j2.414}

08. (d)

Sol:

(i) $(a)^n u(n) \leftrightarrow \dfrac{1}{1-az^{-1}}$

$-(a)^n u(-n-1) \leftrightarrow \dfrac{1}{1-az^{-1}}$

Given, $x(n) = \left(-\dfrac{1}{5}\right)^n u(n) + 5\left(\dfrac{1}{2}\right)^{-n} u(-n-1)$

Apply z-transform

$$X(Z) = \frac{1}{1+\frac{1}{5}z^{-1}} - \frac{5}{1-2z^{-1}}$$

ROC = $(|z|>0.2)\cap(|z|<2) = 0.2 < |z| < 2$

(ii) $\delta(n) \leftrightarrow 1$

$\delta(n-n_0) \leftrightarrow z^{-n_0}$

So, $X(z) = \dfrac{1}{2} + z^{-1} - \dfrac{1}{3}z^{-2}$

ROC is entire z-plane except at z = 0.

(iii) Given $x(n) = n\left(\dfrac{1}{3}\right)^n u(n) + 0.5\left(\dfrac{1}{3}\right)^n u(n)$.

$$\left(\frac{1}{3}\right)^n u(n) \leftrightarrow \frac{1}{1-\frac{1}{3}z^{-1}}$$

$$n\left(\frac{1}{3}\right)^n u(n) \leftrightarrow -z\frac{d}{dz}\left[\frac{1}{1-\frac{1}{3}z^{-1}}\right] = \frac{\frac{1}{3}z^{-1}}{\left(1-\frac{1}{3}z^{-1}\right)^2}$$

$$\text{So, } X(Z) = \frac{\frac{1}{3}z^{-1}}{\left(1-\frac{1}{3}z^{-1}\right)^2} + \frac{0.5}{1-\frac{1}{3}z^{-1}}$$

ROC is $|z| > 1/3$.