MADE EASY
India's Best Institute for IES, GATE & PSUs

Detailed Solutions

# ESE-2024 Mains Test Series

# E & T Engineering Test No : 5

## Section A : Computer Organization and Architecture + Materials Science

**Q.1 (a) Solution:**

**(i)** FIFO:

In FIFO, replace the cache block which is having the longest time stamp.

| 1-miss | 4-miss | 8-miss | 5-miss | 20-miss | 17-miss | 19-miss | 5-hit | 6-miss |
|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 1 | 1 | 20 | 20 | 20 | 20 | 20 |
| | 4 | 4 | 4 | 4 | 17 | 17 | 17 | 17 |
| | | 8 | 8 | 8 | 8 | 19 | 19 | 19 |
| | | | 5 | 5 | 5 | 5 | 5 | 6 |

| 9-miss | 11-miss | 4-miss | 4-hit | 3-miss | 5-miss | 6-miss | 9-miss | 17-miss |
|---|---|---|---|---|---|---|---|---|
| 9 | 9 | 9 | 9 | 9 | 5 | 5 | 5 | 5 |
| 17 | 11 | 11 | 11 | 11 | 11 | 6 | 6 | 6 |
| 19 | 19 | 4 | 4 | 4 | 4 | 4 | 9 | 9 |
| 6 | 6 | 6 | 6 | 3 | 3 | 3 | 3 | 17 |

$$\text{Hit ratio} = \frac{2}{18} = 0.11$$

**(ii) LRU:**

In LRU, replace the cache block that has been in the cache longest without no reference to it.

| 1-miss | 4-miss | 8-miss | 5-miss | 20-miss | 17-miss | 19-miss | 5-hit | 6-miss |
|---|---|---|---|---|---|---|---|---|
| | | | 5 | 20 | 17 | 19 | 5 | 6 |
| | | 8 | 8 | 5 | 20 | 17 | 19 | 5 |
| | 4 | 4 | 4 | 8 | 5 | 20 | 17 | 19 |
| 1 | 1 | 1 | 1 | 4 | 8 | 5 | 20 | 17 |

| 9-miss | 11-miss | 4-miss | 4-hit | 3-miss | 5-miss | 6-miss | 9-miss | 17-miss |
|---|---|---|---|---|---|---|---|---|
| 9 | 11 | 4 | 4 | 3 | 5 | 6 | 9 | 17 |
| 6 | 9 | 11 | 11 | 4 | 3 | 5 | 6 | 9 |
| 5 | 6 | 9 | 9 | 11 | 4 | 3 | 5 | 6 |
| 19 | 5 | 6 | 6 | 9 | 11 | 4 | 3 | 5 |

$$\text{Hit ratio} = \frac{2}{18} = \frac{1}{9} = 0.11$$

**(iii) Direct mapped cache:**

It maps each block of main memory into only one possible cache line given by $k$ mod $n$.

1-miss; 1 mod 4 = 1
4-miss; 4 mod 4 = 0
8-miss; 8 mod 4 = 0
5-miss; 5 mod 4 = 1
20-miss; 20 mod 4 = 0
17-miss; 17 mod 4 = 1
19-miss; 19 mod 4 = 3
5-miss; 5 mod 4 = 1
6-miss; 6 mod 4 = 2
9-miss; 9 mod 4 = 1
11-miss; 11 mod 4 = 3
4-miss; 4 mod 4 = 0
4-hit; 4 mod 4 = 0
3-miss; 3 mod 4 = 3
5-miss; 5 mod 4 = 1
6-hit; 6 mod 4 = 2
9-miss; 9 mod 4 = 1
17-miss; 17 mod 4 = 1

| | $k$ mod $n$ = ? |
|---|---|
| 0 | ~~4 8 20 4~~ |
| 1 | ~~1 5 17 5 9 5 9~~ 17 |
| 2 | 6 |
| 3 | ~~19 11~~ 3 |

$$\text{Hit Ratio} = \frac{2}{18} = \frac{1}{9} = 0.11$$

**(iv) 2-way set associative with LRU:**

$$s = \frac{n}{P\text{-way}} = \frac{4}{2} = 2$$

The cache lines are grouped into two sets. It maps each block of main memory into only one possible set given by $k$ mod $s$. The mapping within a set is done using LRU i.e. replacing the block that has been in the cache longest without no reference to it.

1-miss 1 mode 2 = 1
4-miss 4 mode 2 = 0
8-miss 8 mode 2 = 0
5-miss 5 mode 2 = 1
20-miss 20 mode 2 = 0
17-miss 17 mode 2 = 1
19-miss 19 mode 2 = 1
5-miss 5 mode 2 = 1
6-miss 6 mode 2 = 0
9-miss 9 mode 2 = 1
11-miss 11 mode 2 = 1
4-miss 4 mode 2 = 0
4-hit 4 mode 2 = 0
3-miss 3 mode 2 = 1
5-miss 5 mode 2 = 1
6-hit 6 mode 2 = 0
9-miss 9 mode 2 = 1
17-miss 17 mode 2 = 1

| | set | |
|---|---|---|
| Set 0 2 way | 4 20 4 | 8 6 |
| Set 1 | 1 17 5 11 5 17 | 5 19 9 3 9 |

$k$ mod $s = i$

$$\text{Hit ratio} = \frac{2}{18} = 0.11$$

**Q.1 (b) Solution:**

**(i)** Primary data types in C language along with their size in bytes.

1. **int:** Typically 2 or 4 bytes used to store integers, though it can vary depending on the compiler and system architecture.
2. **char:** Typically 1 byte, representing a single character.
3. **float:** Typically 4 bytes, used to store single precision floating-point numbers.

4. **double:** Typically 8 bytes, used to store double-precision floating-point numbers.
5. **void:** It represents an empty set of values. The size of void is complier-dependent and usually treated as 1 byte, although it's not used to declare variables.
6. **short:** Typically 2 bytes, used to store smaller integer values then int.
7. **long:** Typically 4 or 8 bytes, depending on the complier and system architecture, used to store larger integer values than int.
8. **unsigned int:** Typically 4 bytes, used to store only non-negative integer values.
9. **signed int:** Typically 4 bytes, used to store both positive and negative integer values, though signed is often implicit.
10. **bool:** Typically 1 byte, used to store Boolean values (true or false)

Typical sizes of the data types can vary depending on the compiler and system architecture.

**(ii)**

```
#include <stdio.h>
//function to check if a number is composite
int is_composite (int n) {
    if (n < = 2){
    return 0; //0, 1 and 2 are not composite numbers
    }
    for (int i = 2; i < = n – 1; i++)
    {
    if(n% i = = 0)
    {
    return 1; // if a divisor is found, the number is composite.
    }
    }
    return 0; // if no divisor is found, the number is not composite.
    }
    int main ()
    {
    int num;
    //Input from user
```

```
printf("Enter a positive integer:");
scanf("%d", &num);
//checking if the number is composite
if (is_composite (num))
{
printf ("%d is a composite number\n", num);
}
else
{
printf("%d is not a composite number.\n", num);
}
return 0;
}
```

**Q.1 (c) Solution:**

Top-down and Bottom-up methods are two approaches for synthesis of nano-materials and the fabrication of nano-structures.

**Bottom-up Approach:**

- Bottom-up approach refers to the build up of a material from the bottom: atom by atom molecule by molecule.
- Atom by atom deposition leads to formation of self-assembly of atom/molecules and clusters. These clusters come together to form assembled monolayers on the surface of substrate.
- In all the bottom-up techniques, the starting material is either gaseous state or liquid state of matter. This approach uses physical and chemical processing methods.

**Physical Techniques:**

- Physical vapour deposition (PVD): involves condensation of vapor phase species.
- Evaporation (Thermal, e-beam)
- Sputtering
- Plasma arcing

**Chemical Techniques:**

- CVD (Chemical vapor deposition): Deposition of vapor phase of reaction species.
- Self-assembled monolayer: Electrolytic deposition, sol-gel method, Microemulsion route, Pyrolysis.

**Advantages:**

- Ultra-fine nanoparticles can be prepared.
- Deposition parameters can be controlled.
- Cheaper technique
- Narrow size distribution is possible (1-20 nm)

**Disadvantages:**

- Large scale production is difficult.
- Chemical purification of nano-particles is required.

**Top-down approach**

- Top-down approach refers to slicing or successive cutting of a bulk material to get nano sized particles. It uses physical processing methods, i.e.,
  - Mechanical methods: Cutting, Etching, Grinding, Ball Milling
  - Lithographic Techniques: Photo lithography, Electron Beam lithography

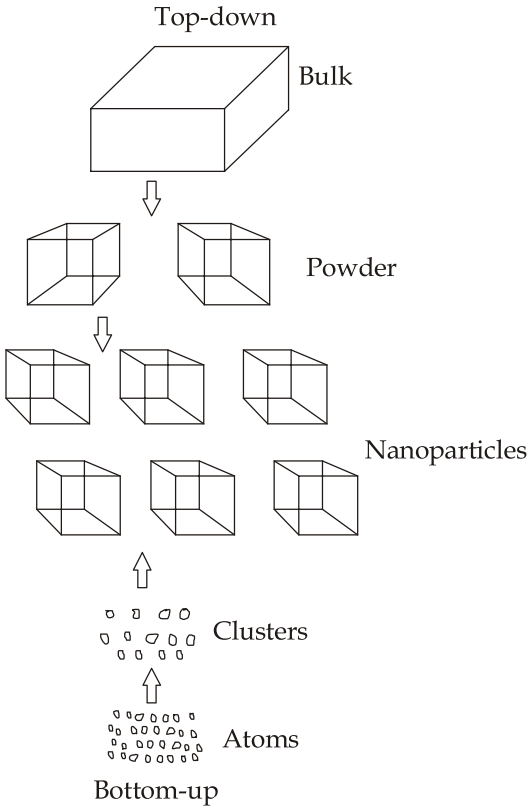
**Advantages:**

- Large scale production is possible.
- Deposition over a large substrate is possible.
- Chemical purification is not required.

**Disadvantages:**

- Expensive technique.
- Control over deposition parameters is difficult to achieve.
- Impurities like stress, defects and imperfections get introduced.

Figure shows the two approaches used for synthesis.




**Q.1 (d) Solution:**

The state of material at which resistivity reduces to zero is called superconductivity. The temperature at which there is transition from normal state to superconducting state is called transition or critical temperature. Above critical temperature ($T_C$), the materials is in familiar normal state but below $T_C$, it enters in superconducting state. The resistance of these materials in the superconducting state is at least $10^{16}$ times smaller than their room temperature values.

The two independent conditions for superconductivity are

1. Zero resistivity 2. Perfect diamagnetism

Some of the properties of superconductor are as follows:

(a) At room temperature, the resistivity 'ρ' of superconducting materials are greater than other elements.

(b) All thermoelectric effects disappear in superconducting state.

(c) When current is passed through the superconducting material, the heat loss ($I^2R$) is zero.

(d) The magnetic flux density in super conductor is zero due to a phenomenon known as the Meissner effect.

(e) When a sufficient strong magnetic field is applied to superconductor below critical temperature $T_c$, its superconducting property is destroyed.

**Q.1 (e) Solution:**

Defects are imperfections which cause disruption in what otherwise would be a perfect lattice.

**Point defects (0 - *D*) :**

- **Vacancy :** Absence of an atom in a lattice point where there is suppose to be an atom. It occurs when atoms are removed from their lattice positions (typically to the surface) as a result of thermal fluctuations. It is also known as Schottky defect.
- **Interstitial :** Atom located in a "void" (i.e., a position that is not part of the lattice or basis) within the crystal structure.
- **Frenkel defect :** Impurity atoms (an atom that does not belong to the basis) that take up the lattice positions are ordinarily occupied by the atoms that make up the crystal.
- **Interstitial defect:** An atom of the same or of a different type, occupies an interstitial site in the crystal structure.

**Line defects/ Dislocations (1 - *D*) :**

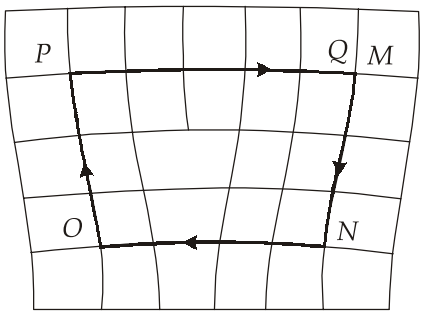
Line defecrs occur when boundary between two regions of a surface which are perfect themselves but are out of registry with each other. The resulting lattice distortion is centered along a line.

**Burgers vector, *b*** : A vector by which the lattice on one side of an internal surface containing the dislocation line is displaced relative to the lattice on the other side.

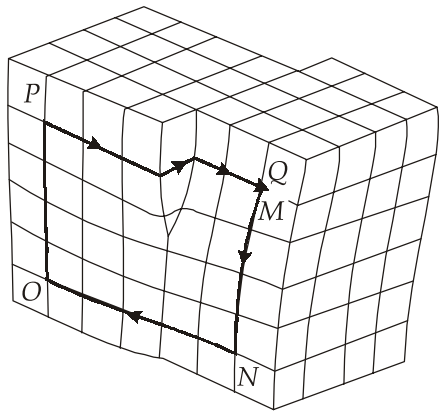
There are two special cases of dislocations:

- **Edge dislocation :** $b$ and normal vector along the dislocation line $l$ are perpendicular.
- **Screw dislocation** : $b$ and normal vector along the dislocation line $l$ are parallel.

**Edge Dislocation**



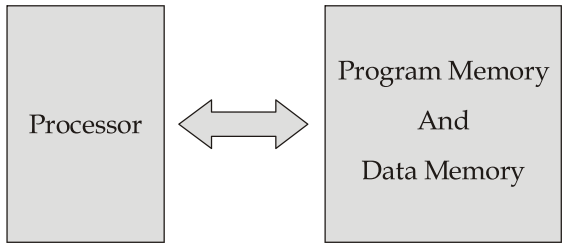

**Screw Dislocation**



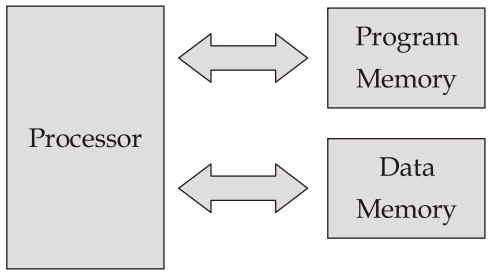
**Q.2 (a) Solution:**

**(i)** **Von Neumann Architecture:** It is a digital computer architecture whose design is based on the concept of stored program computers where program data and instruction data are stored in the same memory. This architecture was designed by the famous mathematician and physicist John Von Neumann in 1945.

**Harvard Architecture:** Harvard is the digital computer architecture whose design is based on the concept where there are separate storage and separate buses for instruction and data. It was basically developed to overcome the bottleneck of Von Neumann architecture.




Von Neumann Architecture




Hardward Architecture

Difference between Von Neumann and Harvard Architecture.

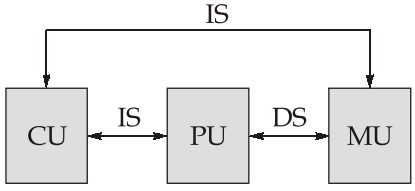
| Von Neumann Architecture | Harvard Architecture |
|---|---|
| 1. It is an ancient type of computer architecture based on stored program computer concept. | 1. It is modern computer architecture based on Harvard Mark I relay based model. |
| 2. Same physical memory address is used for instructions and data. | 2. Separate physical memory address is used for instructions and data. |
| 3. There is common bus for data and instruction transfer. | 3. Separate buses are used for tranferring data and instruction. |
| 4. Two clock cycles are required to execute single instruction. | 4. An instruction is executed in a single clock cycle. |
| 5. It is cheaper in cost. | 5. It is costly than Von Neumann Architecture. |
| 6. CPU cannot access instructions and read/write at the same time. | 6. CPU can access instructions and read/write at the same time. |
| 7. It is used in personal computers and small computers. | 7. It is used in micro-controllers and signal processing. |

**(ii)** Based on the number of instruction and data streams that can be processed simultaneously, computing systems are classified into four major categories as per Flynn's classification:

**1. Single Instruction Stream, Single Data Stream (SISD):**

A computer with a single processor is called a Single Instruction stream, Single Data stream (SISD) computer. It represents the organization of a single computer containing a control unit, a processor unit and a memory unit. Instructions are executed sequentially and the system may or may not have internal parallel processing. Parallel processing may be achieved by means of a pipeline processing.

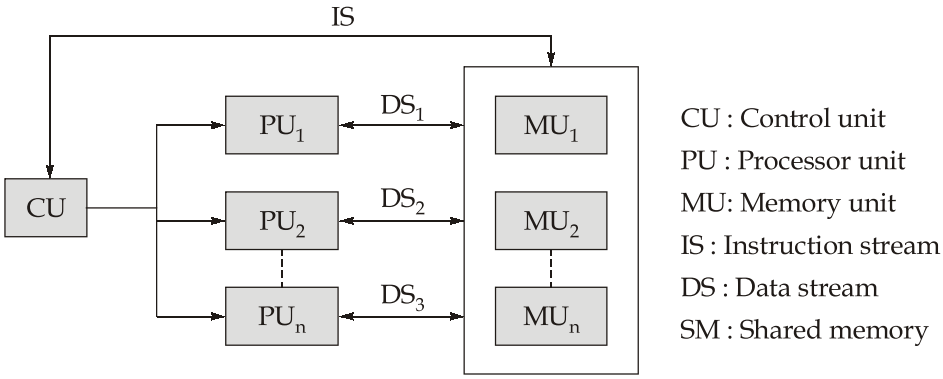
In such a computer a single stream of instructions and a single stream of data are accessed by the processing elements from the main memory, processed and the results are stored back in the main memory. SISD computer organization is shown in figure below.




CU : Control unit
PU : Processor unit
MU: Memory unit
IS : Instruction stream
DS : Data stream

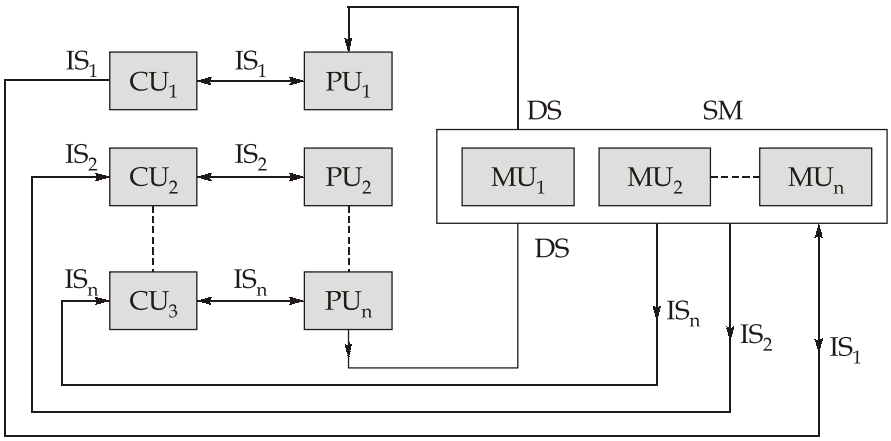
**2. Single Instruction Stream, Multiple Data Stream (SIMD):**

It represents an organization of computer which has multiple processors under the supervision of a common control unit. All processors receive the same instruction from the control unit but operate on different items of the data. SIMD computers are used to solve many problems in science which require identical operations to be applied to different data sets synchronously. Examples are adding a set of matrices simultaneously, such as $\sum_i \sum_k (a_{ik} + b_{ik})$. Such computers are known as array processors. SIMD computer organization is shown in figure below:



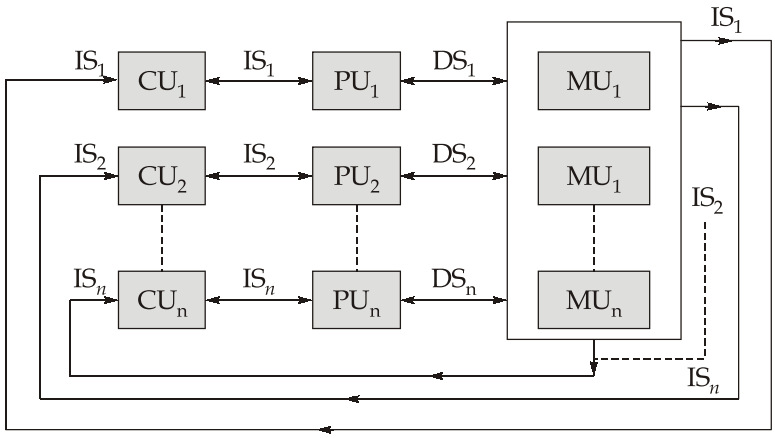

**3. Multiple Instruction Stream, Single Data Stream (MISD):**

It refers to the computer in which several instruction manipulate the same data stream concurrently. In this structure, different processing elements run different programs on the same data. This type of processor may be generalized using a 2-dimensional arrangement of processing elements. Such a structure is known as systolic processor. MISD computer organization is shown in figure below:



**4. Multiple Instruction Stream, Multiple Data Stream (MIMD):**

MIMD computers are the general purpose parallel computers. Its organization refers to a computer system capable of processing several programs at a same time i.e. it can execute several instructions, on different data sets, all at the same time. MIMD systems include all multiprocessing systems. MIMD computer organization is shown in figure below.




CU : Control unit
PU : Processor unit
MU: Memory unit
IS : Instruction stream
DS : Data stream

**Q.2 (b) Solution:**

**(i)** Nanomaterials are defined as those materials which have structured components with size less than 100 nm atleast in one dimension. Such materials are having novel properties due to its nanoscale size. The most important parameter that influences the properties of nano-structured materials is the large surface area to volume ratio of nano particles. This feature is mainly responsible for the modification of properties at the nanostructural level

Depending on the number of dimensions in the nano range, materials can be classified as follows:

**1. Zero-dimensional**

It is also called as nanoparticle, where all three dimensions of the particle are in the nano range. Note that the term 'zero-dimensional' is applied to a particle, which has all the three dimensions in the nano range and none in the larger-than nano range. In this case, electron movement is restricted in all three dimensions.

Example: Quantum dots, Nanoparticles, Carbon nanodots, Fullerene.

**2. One-dimensional**

In one-dimensional nanomaterials, two dimensions are in the nano range and the third dimension is much larger. Carbon nanotubes are typical examples.

They are cylindrical tubes of carbon atoms with diameter in the range of 1-2 nm and a much larger length reaching upto a mm.

Examples: nanorods, nanowires and nanofibres etc.

**3. Two-dimensional**

In two-dimensional nanomaterials, one dimension is in the nano range and the other two are much larger. Two-dimensional crystalline nanosheets have thickness in the nanoscale. In addition, the internal structure of the sheet can be nanosized grains.

Examples: graphene, metal nanosheets etc.

**4. Three-dimensional**

In three-dimensional nanomaterials, all three dimensions of a particle are much larger than the nano range. This category, sometimes called bulk nanoparticle, forms part of this classification, because the bulk solid itself may be composed of nanoparticles, e.g., nanosized crystals in a bulk polycrystalline material.

Examples: Nanocubes, dendrimers, etc...

**(ii)**

| Carbon dots | Quantum Dots |
|---|---|
| 1. Carbon dots are small carbon nanoparticle having some form of surface passivation. Their size is less than 10 nm. | 1. Quantum dots are small semiconductor particles on a nanoscale, having optical and electronic properties that differ from large particles according to quantum mechanics. |
| 2. Top-down and bottom-up methods are used for production of carbon dots. | 2. Colloidal synthesis, plasma synthesis, fabrication, electrochemical assembly can be used for production of Quantum dots. |
| 3. The properties of carbon dots solely depends on their structures and compositions. | 3. The properties of quantum dots are intermediate to those of bulk semiconductors and discrete molecules. |
| 4. Carbon dots are used in bioimaging, sensing, drug delivery, etc. | 4. Quantum dots are used in LEDs, single photon sources, quantum computing etc. |

**Q.2 (c) Solution:**

**(i)** "Old" computer: For execution of program,

$$\text{CPU time} = \frac{\text{CPU clock cycles}}{\text{clock rate}}$$

$$20 \text{ sec} = \frac{\text{CPU clock cycles}}{4 \times 10^9 \text{ cycles/second}}$$

$$(\text{CPU clock cycles})_{\text{old}} = 80 \times 10^9 \text{ cycles}$$

Upgraded design:

$$\text{CPU time} = \frac{1.2 \times (\text{CPU clock cycles})_{\text{old}}}{\text{clock rate}}$$

$$15 = \frac{1.2 \times 80 \times 10^9 \text{ cycles}}{\text{clock rate}}$$

$$\text{Clock rate} = \frac{1.2 \times 80 \times 10^9 \text{ cycles}}{15 \text{ sec}}$$

$$= 6.4 \times 10^9 \text{ cycles/sec}$$

$$= 6.4 \text{ GHz}$$

**(ii)** Pipeline design in computer architecture involves breaking down the execution of instructions into smaller, more manageable stages. This allows multiple instructions to be processed simultaneously, improving overall performance.

**Pipeline Design:** The design of pipeline generally involve the following stages:

1. **Instruction Fetch (IF):** Fetching the instruction from memory.
2. **Instruction Decode (ID):** Decoding the instruction and determining the necessary operations.
3. **Execute (EX):** Performing the arithmetic or logical operation specified by the instruction.
4. **Memory Access (MEM):** Accessing memory, if needed (e.g., for load/store operations.)
5. **Write Back (WB):** Writing the result of the operation, back to registers or memory.

**Performance Analysis with pipeline:**

The cycle time $\tau$ of an instruction pipeline is the time needed to advance a set of instructions one stage through the pipeline. We have,

$$\text{Cycle time } \tau = (\tau_i)_{\max} + d = \tau_m + d;\ 1 \le i \le k$$

where $\tau_m$ = maximum stage delay (delay through stage which experience the largest delay)

$k$ = Number of stages in the instruction pipeline.

$d$ = Time delay of a latch needed to advance signals and data from one stage to the next.

In general, the time delay $d$ is equivalent to a clock pulse and $\tau_m >> d$. The total time required $T_k$ to execute all n instructions is

$$T_k = [k + (n - 1)]\tau$$

A total of $k$ cycles are required to complete the first instruction and the remaining $(n - 1)$ instructions require $(n - 1)$ cycles.

The time required to execute n instructions without pipeline is $T_1 = nk\tau$ because to execute one instruction it will take $k\tau$ cycles.

The speed up factor for the instruction pipeline compared to execution without the pipeline is defined as:

$$s_k = \frac{T_1}{T_k} = \frac{nk\tau}{[k + (n-1)]\tau} = \frac{nk}{k + (n-1)} = \frac{nk}{(k-1) + n}$$

If the number of instruction executed is much more higher than the number of stages in the pipeline then as, $n >> k - 1$. Thus $s_k = \frac{nk}{n} = k$.

The speed up factor is a function of the number of stages ($k$) in the instruction pipeline.

**Common issues in pipeline design:**

1. **Data Hazards:** These occur when an instruction depends on the result of a previous instruction that has not yet completed. Data hazards can lead to stalls or pipeline bubbles.
2. **Control Hazards:** These occur when the pipeline must wait for a conditional branch instruction to determine the next instruction's address.

   Branch prediction and speculative execution are techniques used to mitigate control hazards.
3. **Structural Hazards:** These occur when hardware resources required by multiple instructions overlap in time. For example, two instructions may need to access the same resource simultaneously.
4. **Pipeline flushes:** These occur when the pipeline must be cleared and restarted due to an exception or mispredicted branch. Pipeline flushes result in wasted cycles and reduced performance.
5. **Instruction Dependencies:** Dependencies between instructions can limit the degree of parallelism achievable in a pipeline. Techniques like out-of-order execution and register renaming are used to mitigate these dependencies and improve performance.

Effective pipeline design involves balancing trade-offs between through put, latency and complexity.

Techniques such as pipelining, superscalar execution and out-of-order execution are commonly employed to maximize performance while minimizing issues.

**Q.3 (a) Solution:**

**(i)** The program under execution is called process. The process from its creation to completion passes through various states which are described in process state transition diagram as discussed below.

**Process states:**

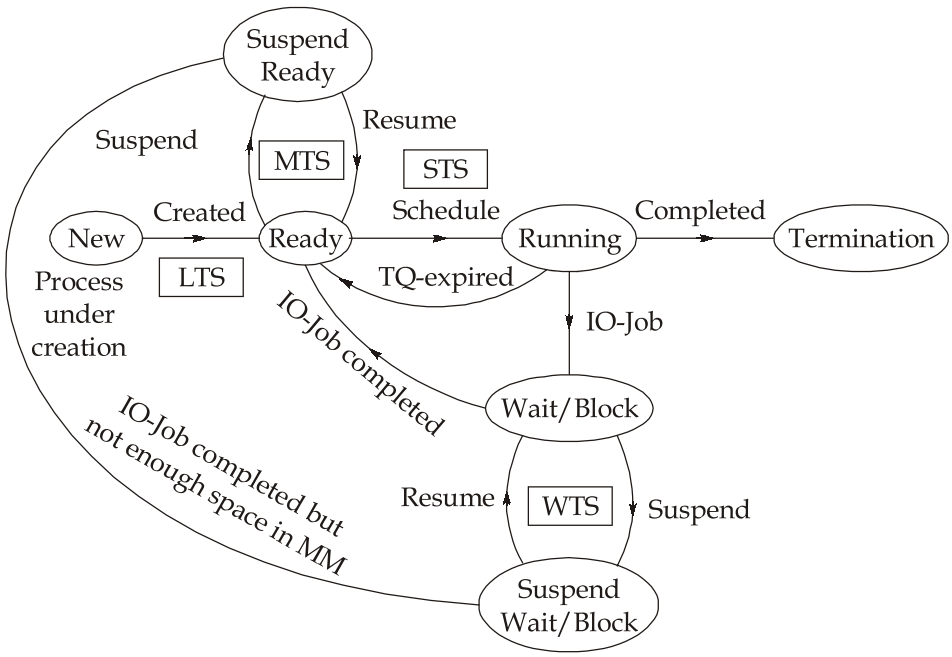
1. **New:** A program which is going to be picked up by the OS into the main memory is called a new process.
2. **Ready:** Whenever a process is created, it directly enters in the ready state, in which it waits for the CPU to be assigned. The OS picks the new processes from the secondary memory and put all of them in the main memory. The processes which are ready for the execution and reside in the main memory are called ready state processes. There can be many processes present in the ready state.
3. **Running:** One of the processes from the ready state will be chosen by the OS depending upon the scheduling algorithm. Hence, if we have only one CPU in our system, the number of running process at a particular time will always be one. If we have '*n*' processors in the system, then we can have '*n*' process running simultaneously.
4. **Wait/Block:** From the running state, process can make the transition to the block or wait state depending upon the scheduling algorithm or the intrinsic behaviour of the process.

   When a process waits for a certain resource to be assigned or for the input from the user, then the OS move this process to the block or wait state and assigns the CPU to the other processes.
5. **Suspend Ready state:** A process in the ready state, which is moved to secondary memory from the main memory due to lack of resources (mainly primary memory) is called in the suspend ready state.

   If the main memory is full and a higher priority process comes for the execution then the OS have to make the room for the process in the main memory by throwing the lower priority process out into the secondary memory.

   The suspend ready process remain in the secondary memory until the main memory gets available.

6. **Suspend Wait state:** Instead of removing the process from the ready queue, it's better to remove the blocked process which is waiting for some resources in the main memory. Since it is already waiting for some resources to get available, hence it is better if it waits in the secondary memory and make room for the higher priority process. These processes complete their execution once the main memory gets available and their wait is finished.
7. **Completion or Termination:** When a process finishes its execution, it comes in the termination state. All the context of the process (process control block) will also be deleted and process will be terminated by the operating system.




**(ii) CPU Scheduling:** CPU scheduling is a process which allows one process to use the CPU while the execution of another process is on hold (in waiting state) due to unavailability of any resource like I/O etc, thereby making full use of CPU. The aim of CPU scheduling is to make the system efficient, fast and fair. Whenever the CPU becomes idle, the operating system must select one of the processes in the ready queue to be executed. The selection process is carried out by the short-term scheduler (or CPU scheduler). The scheduler selects from among the processes in memory that are ready to execute, and allocates the CPU to one of them.

**First Come First Serve Scheduling:**

In the "First Come First Serve" scheduling algorithm as the name suggest, the process which arrives first gets executed first, or we can say that the process which requests the CPU first, gets the CPU allocated first.

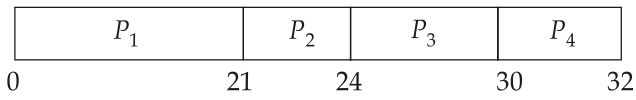
- First come first serve is just like FIFO (First in First out) Queue data structure, where the data element which is added to the queue first, is the one who leaves the queue first.
- This is used in batch systems.
- It is easy to understand and implement programmatically, using a queue data structure, where a new process enters through the tail of the queue, and the scheduler selects process from the head of the queue.

A perfect real life example of FCFS scheduling is buying tickets at ticket counter.

Example: Consider the processes $P_1$, $P_2$, $P_3$, $P_4$ given in the below table, arrives for execution at $T$ = 0, 1, 3 and 4 ms with burst time of 21, 3, 6 and 2 ms respectively.

Let's find the average waiting time using the FCFS scheduling algorithm.

| Process ID | Arrival time (ms) (AT) | Burst time (ms) (BT) | Completion time (CT) | Turn Around Time (TAT) = CT – AT | Waiting Time (WT) = TAT – BT |
|---|---|---|---|---|---|
| $P_1$ | 0 | 21 | 21 | 21 | 0 |
| $P_2$ | 1 | 3 | 24 | 23 | 20 |
| $P_3$ | 3 | 6 | 30 | 27 | 21 |
| $P_4$ | 4 | 2 | 32 | 28 | 26 |



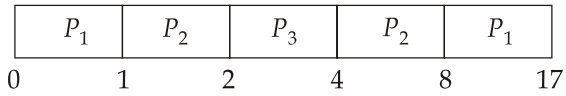

The average waiting time will be $\frac{0 + 20 + 21 + 26}{4} = 16.75 \text{ ms}$.

**Shortest remaining time first (SRTF) scheduling:**

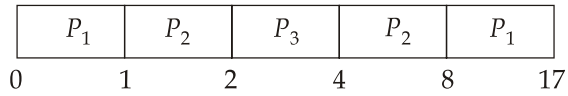
Shortest remaining time scheduling is the preemptive counter part of Shortest Job First (SJF) scheduling and is useful in time sharing system. In SRTF, process with smallest estimated run time to completion is run next, whereas in SJF once a job begin executing, it runs to completion. In SRTF, a running process may be preempted by a user process with a shorter estimated run time.

Consider, an example, where three processes arrived in the order $P_1$, $P_2$, $P_3$ at the time mentioned below, and then the average waiting time using SJF scheduling algorithm will be calculated as below:

| Process | CPU Brust Time (ms) | Time of Arrival (ms) |
|---|---|---|
| $P_1$ | 10 | 0 |
| $P_2$ | 5 | 1 |
| $P_3$ | 2 | 2 |




In this, the CPU will be taken away from the currently executing process whenever a process will less CPU Burst time arrives. Then, we get




From GANTT Chart we have,

| Process ID | Arrival time (ms) (AT) | Burst time (ms) (BT) | Completion time (CT) | Turn Around Time (TAT) = CT – AT | Waiting Time (WT) = TAT – BT |
|---|---|---|---|---|---|
| $P_1$ | 0 | 10 | 17 | 17 | 7 |
| $P_2$ | 1 | 5 | 8 | 8 | 2 |
| $P_3$ | 2 | 2 | 4 | 4 | 0 |

$$\text{Average waiting time} = \frac{(7+2+0)}{3} = 3 \text{ millisecond}$$

**Q.3 (b) Solution:**

**(i)** The modulus of elasticity of the composite is calculated using Equation

$$E_{cl} = (3.4\text{ GPa})(0.6) + (69\text{ GPa})(0.4)$$
$$= 29.64\text{ GPa}$$

**(ii)** To solve this portion of the problem, first find the ratio of fiber load to matrix load,

$$\frac{F_f}{F_m} = \frac{E_f V_f}{E_m V_m} = \frac{(69\text{ GPa})(0.4)}{(3.4\text{ GPa})(0.6)} = 13.53$$

or $$F_f = 13.53\, F_m$$

In addition, the total force sustained by the composite $F_c$ may be computed from the applied stress σ and total composite cross-sectional area $A_c$ according to

$$F_c = A_c\sigma = (250\text{ mm}^2)(50\text{ MPa}) = 12{,}500\text{ N}$$

However, this total load is the sum of the loads carried by fiber and matrix phase, that is

$$F_c = F_f + F_m = 12500 \text{ N}$$

Since, $F_f = 13.53\, F_m$

$\therefore$ $F_c = (13.53 + 1)F_m = 12500$ N

$F_m = 860.29$ N

We have, $F_f = (13.53)(860.29)$

$F_f = 11639.71$ N

Thus, the fiber phase supports the vast majority of load.

**(iii)** The stress for both fiber and matrix phases must first be calculated. Then, by using the elastic modulus for each (from part (i)), the strain values may be determined.

$$A_m = V_m A_c = (0.6)(250 \text{ mm}^2) = 150 \text{ mm}^2$$

$$A_f = V_f A_c = (0.4)(250 \text{ mm}^2) = 100 \text{ mm}^2$$

Hence, $$\sigma_m = \frac{F_m}{A_m} = \frac{860.29}{150} = 5.74 \text{ MPa}$$

$$\sigma_f = \frac{F_f}{A_f} = \frac{11639.71}{100} = 116.39 \text{ MPa}$$

Finally, strains are computed as,

$$\in_m = \frac{\sigma_m}{E_m} = \frac{5.74 \text{ MPa}}{3.4 \times 10^3 \text{ MPa}} = 1.69 \times 10^{-3}$$

$$\in_f = \frac{\sigma_f}{E_f} = \frac{116.39 \text{ MPa}}{69 \times 10^3 \text{ MPa}} = 1.69 \times 10^{-3}$$

Therefore, strains for both matrix and fiber phase are identical.

**Q.3 (c) Solution:**

**(i)** **1.** Simple cubic lattice, $a = 2r$

Unit cell volume $= a^3 = (2r)^3 = 8r^3$

Simple cubic lattice has 1 atom per cell, so volume of the atoms $= (1)\left[\frac{4\pi r^3}{3}\right]$

Then, $$\text{Ratio} = \frac{\left(\frac{4\pi r^3}{3}\right)}{8r^3} \times 100\% \Rightarrow \text{Ratio} = 52.4\%$$

**2.** Face-centred cubic lattice:

$$d = 4r = a\sqrt{2}$$

$$\Rightarrow \quad a = \frac{d}{\sqrt{2}} = 2\sqrt{2}\,r$$

Unit cell volume = $a^3 = (2\sqrt{2}\,r)^3 = 16\sqrt{2}\,r^3$

FCC lattice has 4 atoms per cell, so volume of the atoms = $4\left(\frac{4\pi r^3}{3}\right)$

$$\text{Ratio} = \frac{4\left(\frac{4\pi r^3}{3}\right)}{16\sqrt{2}\,r^3} \times 100\% \quad \Rightarrow \text{Ratio} = 74\%$$

**3.** Body centred cubic lattice,

$$d = 4r = a\sqrt{3}$$

$$\Rightarrow \quad a = \frac{4}{\sqrt{3}}r$$

$$\text{Unit cell volume} = a^3 = \left(\frac{4}{\sqrt{3}}r\right)^3$$

BCC lattice has 2 atoms per cell, so volume of the atoms = $2\left(\frac{4\pi r^3}{3}\right)$

$$\text{Ratio} = \frac{2\left(\frac{4\pi r^3}{3}\right)}{\left(\frac{4r}{\sqrt{3}}\right)^3} \times 100\% = 68\%$$

**4.** Diamond lattice,

$$\text{Body diagonal} = d = 8r = a\sqrt{3}$$

$$\Rightarrow \quad a = \frac{8}{\sqrt{3}}r$$

$$\text{Unit cell volume} = a^3 = \left(\frac{8r}{\sqrt{3}}\right)^3$$

Diamond lattice has 8 atoms per cell, so volume of atoms = $8\left(\frac{4\pi r^3}{3}\right)$

$$\text{Ratio} = \frac{8\left(\frac{4\pi r^3}{3}\right)}{\left(\frac{8r}{\sqrt{3}}\right)^3} \times 100\% = 34\%$$

**(ii)** **Given:** $V = 50\ \text{kV} = 50 \times 10^3\ \text{V}$ ; $M = 74.6$ ; $\rho = 1.99 \times 10^3\ \text{kg/m}^3$

Number of atoms/unit cell, $n = 4$ (FCC)

$$\lambda = 0.248\ \text{Å}$$

We know that, $$a^3\rho = \frac{nM}{N}$$

$$\therefore \quad a = \sqrt[3]{\frac{nM}{N} \cdot \frac{1}{\rho}} = \sqrt[3]{\frac{4 \times 74.6}{6.023 \times 10^{23}} \times \frac{1}{1.99 \times 10^6}}$$

$$\therefore \quad a = 6.29 \times 10^{-10}\ \text{m}$$

Since, KCl is ionic crystal,

$$\therefore \quad d = \frac{a}{2} = \frac{6.29}{2} \times 10^{-10} = 3.145 \times 10^{-10}\ \text{m}$$

Using Bragg's law, $n\lambda = 2d \sin\theta$

Consider $n = 1$ for first order reflection,

$$\therefore \quad \sin\theta = \frac{1 \times \lambda}{2d} = \frac{1 \times 0.248 \times 10^{-10}}{2 \times 3.145 \times 10^{-10}} = 0.0394$$

$$\therefore \quad \theta = \sin^{-1}(0.0394) = 2.26°$$

**Q.4 (a) Solution:**

**(i)** **Transaction:** A transaction in a single logical unit of work which accesses and possibly modifies the contents of a database. Transactions access data using read and write operations.

In order to maintain consistency in a data base, before and after transaction, certain properties are followed.

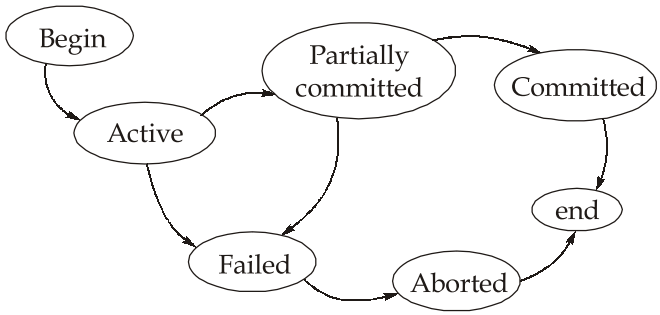
These are called ACID properties:

- **Atomicity:** By this, we mean that either the entire transaction takes place at once or doesn't happen at all. There is no midway i.e., transactions do not occur partially. Each transaction is considered as one unit and either runs to completion or is not executed at all. It involves following two operations:

  **Abort:** If a transaction aborts, changes made to database are not visible.

  **Commit:** If a transaction commits, changes made are visible.

  Atomicity is also known as the 'All or nothing rule':




Consider the following transaction $T$ consisting of $T_1$ and $T_2$: Transfer of 100 from account $X$ to account $Y$.

| Before : $X$ : 500 | $Y$ : 200 |
|---|---|
| Transaction $T$ | |
| $T_1$ | $T_2$ |
| Read ($X$)<br>$X$ : = $X$ – 100<br>Write ($X$) | Read ($Y$)<br>$Y$ : = $Y$ – 100<br>Write (Y) |
| After : $X$ : 400 | $Y$ : 300 |

If the transactions fails after completion of $T_1$ but before completion of $T_2$ (say, after write ($X$) but before write ($Y$)), then amount has been deducted from $X$ but not added to $Y$. This results in an inconsistent database state. Therefore, the transaction must be executed in entirety in order to ensure correctness of database state.

- **Consistency:** This means that integrity constraints must be maintained so that the database is consistent before and after the transaction. It refers to correctness of a database. Referring to the example above, the total amount before and after the transaction must be maintained.

  Total before T occurs = 500 + 200 = 700.

  Total after T occurs = 400 + 300 = 700

  Therefore, database is consistent. Inconsistency occurs in case $T_1$ completes but $T_2$ fails. As a result, $T$ is incomplete.

- **Isolation:** This property ensures that multiple transactions can occurs concurrently without leading to inconsistency of database state. Transactions occur independently without interference. Changes occurring in a particular transaction will not be visible to any other transaction until that particular change in that transaction in written to memory or has been committed. This property ensures that the execution of transactions concurrently will result in a state that is equivalent to a state achieved these were executed serially in some order.

Let $X = 500$, $Y = 500$

Consider two transactions $T$ and $T''$.

| $T$ | $T''$ |
|---|---|
| Read ($X$)<br>$X := X * 100$<br>Write ($X$)<br>Read ($Y$)<br>$Y := Y - 50$<br>Write | Read ($X$)<br>Read ($Y$)<br>$Z := X + Y$<br>Write ($Z$) |

Suppose $T$ has been executed till Read($Y$) and then $T''$ starts. As a result, interleaving of operations takes place due to which $T''$ read correct value of $X$ but incorrect value of $Y$ and sum computed by

$T''$: ($X + Y = 50{,}000 + 500 = 50500$)

is thus not consistent with the sum at end of transaction:

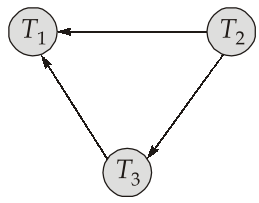
$T$: ($X + Y = 50000 + 450 = 50450$)

This results in database inconsistency, due to a loss of 50 units. Hence, transactions must take place in isolation and changes should be visible only after they have been made to the main memory

- **Durability:** This property ensures that once the transaction has completed execution, the updates and modifications to the database are stored in and written to disk and they persist even if system failure occurs. These updates now become permanent and are stored in a non-volatile memory. The effects of the transaction, thus are never lost.

**(ii)** Checking conflict serializability:

1. Draw precedence graph.
2. Check cycle exist in precedence graph.
   - if yes, then not conflict serializable
   - else conflict serializable




Since precedence graph does not contain cycle, the given schedule is conflict serializable.

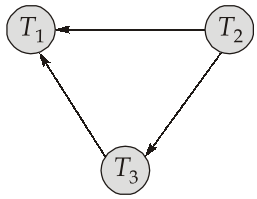
**Checking view serializability:**

1. **Initial Read:** If transaction $T_i$ read data item A from initial database in some serial schedule $S$, then in the given schedule also, $T_i$ should read item $A$ from initial database only.

2. **Write Read:** If transaction $T_i$ reads data item $A$ which is updated by $T_j$ in some serial schedule $S$, then in the given schedule also $T_i$ should read data item A which is modified by $T_j$ only.
3. **Final write:** If transaction $T_j$, update database item $A$, finally in some serial schedule $S$, then in the given schedule also $T_i$ should update data item finally. As

| $T_1$ | $T_2$ | $T_3$ |
|---|---|---|
| $r_1(X)$ | | |
| | | $r_3(Y)$ |
| | | $r_3(X)$ |
| | $r_2(Y)$ | |
| | $r_2(Z)$ | |
| | | $w_3(Y)$ |
| | $w_2(Z)$ | |
| $r_1(Z)$ | | |
| $w_1(X)$ | | |
| $w_1(Z)$ | | |

Now, draw precedence graph from the table.




As, there is no cycle exists in the precedence graph. Hence, it is view serializable with serial scheduling of $T_2 \rightarrow T_3 \rightarrow T_1$.

**Q.4 (b) Solution:**

**(i)** Given,Dielectric constant, $\in_r = 3.4$.

Density, $D = 2.07 \text{ gm/cc} = 2.07 \times 10^3 \text{ kg/m}^3$

Atomic weight = 32.07

Crystal structure of sulphur is cubic.

From Clausius-Mossotti equation,

$$\frac{(\in_r - 1)}{(\in_r + 2)} = \frac{N\alpha_e}{3\in_o}$$

$$\alpha_e = \frac{3\epsilon_o}{N} \times \left[\frac{\epsilon_r - 1}{\epsilon_r + 2}\right]$$

where, $N$ = Number of atoms/unit volume can be written as

$$N = \frac{N_A D}{\text{Atomic weight}}$$

$$N = \frac{6.023 \times 10^{23} \times 2.07 \times 10^6}{32.07} = 3.89 \times 10^{28}/\text{m}^3$$

By substituting values for various parameters,

$$\alpha_e = \frac{3 \times 8.854 \times 10^{-12}}{3.89 \times 10^{28}} \left[\frac{3.4 - 1}{3.4 + 2}\right]$$

$$\alpha_e = 3.035 \times 10^{-40}\ \text{Fm}^2$$

**(ii)** For FCC structure, $a = \frac{4r}{\sqrt{2}}$

$$a = \frac{4 \times 1.44}{\sqrt{2}} = 4.073\ \text{Å}$$

The interplanar distance between two adjacent and parallel planes having Miller indices ($h$, $k$, $l$) is given by

$$d_{hkl} = \frac{a}{\sqrt{h^2 + k^2 + l^2}}$$

For (220) plane, $d_{220} = \frac{4.073}{\sqrt{2^2 + 2^2 + 0^2}} = 1.44\ \text{Å}$

For (200) plane, $d_{200} = \frac{4.073}{\sqrt{2^2 + 0^2 + 0^2}} = 2.036\ \text{Å}$

For (111) plane, $d_{111} = \frac{4.073}{\sqrt{1^2 + 1^2 + 1^2}} = 2.35\ \text{Å}$

**Q.4 (c) Solution:**

**(i)** Virtual address = 47 bits

Page size = 16 kB

Page table entry size (PTES) = 8 Byte

We have to perform multi-level paging until page table size ≤ Page size.

I$^{st}$ level paging:

$$\text{Number of pages} = \frac{\text{Virtual address}}{\text{Page size}} = \frac{2^{47}}{2^{14}} = 2^{33}$$

$$\text{Page table size} = \text{Number of pages} \times \text{PTES}$$
$$= 2^{33} \times 8 = 64 \text{ GB}$$

Since page table size > page size, so

II$^{nd}$ level paging:

$$\text{page table size} = \frac{64 \text{ GB}}{2^{14}} \times 8\text{B}$$
$$= \frac{2^{36}}{2^{14}} \times 2^3\text{B} = 32 \text{ MB}$$

Since page table > page size, so

III$^{rd}$ level paging

$$\text{page table size} = \frac{32 \text{ MB}}{2^{14}} \times 8\text{B} = \frac{2^{25}}{2^{14}} \times 2^3\text{B}$$
$$= 2^{14}\text{B} = 16 \text{ kB}$$

Now, page table size = page size

So, 3 levels are required to map logical address space if every page table is required to fit in single page.

**(ii)** (a) Disk capacity = #Surface in disk × #Tracks per surface × #Sectors per track × #Bytes per sector

$$= 32 \times 2 \times 256 \times 512 \times 1024 \text{ Bytes}$$
$$= 2^{33} = 8 \text{ GB}$$

(b) Seek time = 12.5 msec

- Rotational latency depends on rotational speed.

$$8800 \text{ revolution} = 60 \text{ sec}$$

$$1 \text{ revolution} = \frac{60}{8800} = 6.81 \text{ msec}$$

$$\text{Average rotational latency} = \frac{1}{2} \times 6.81 = 3.40 \text{ msec}$$

- Transfer time depends on rotational speed

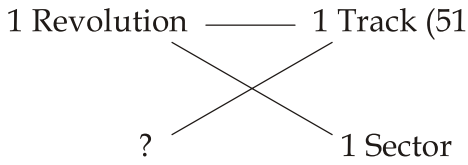
$$1 \text{ revolution access} = 1 \text{ track (i.e., 512 sequential sectors)}$$

$$\text{Transfer time for 72 sequential sectors} = \frac{72 \times 6.81}{512} = 0.957 \text{ msec}$$

So, Time required to access 72 sequential sectors,

$$T_{avg} = 12.5 + 3.4 + 0.957 + 0$$
$$= 16.857 \text{ ms}$$

(c) Random sector accessing requires adjustment for every sector, so




Transfer time for 1 sector = $\frac{6.81 \times 1}{512} = 0.0133$ msec

So,

Time required to access one sector = (12.5 + 3.4 + 0.0133 + 0) ms

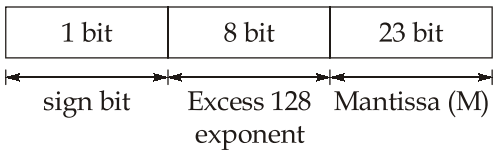
Time required to access 120 random sectors = (12.5 + 3.4 + 0.0133) × 120

$$= 1909.6 \text{ ms}$$
$$= 1.91 \text{ sec}$$

(d) In one revolution, 512 x 1024 bytes of data can be accessed. Since one revolution time is 6.81 msec, hence Data Transfer Rate

$$= \frac{512 \times 1024}{6.81} \times 10^3 = 76.987 \times 10^6 \text{ Bytes/sec}$$
$$= 76.987 \text{ MBps}$$

**(iii)** The given 32-bit floating point format with 1 sign bit (S), 8 -bit biased exponents (BE) and 23 -bit mantissa(M) represents IEEE singe precision format.




For Minimum number

$$\text{Mantissa} = \underbrace{0\,0\,0\,0\,\ldots\,0}_{23\ 0's}$$

$$\text{Biased Exponent} = 00000000$$
$$\text{Actual exponent} = 0 - 128 = -128$$
$$\Rightarrow \quad \text{Minimum number} = 1.0000\ldots\ldots 0 \times 2^{-128} = 2^{-128}$$

Maximum number

$$\text{Mantissa} = \underbrace{1111\ldots.1}_{23\ 1's}$$

$$\text{Biased Exponent} = 11111111 = 255$$

$$\text{Actual exponent} = 255 - 128 = 127$$

$$\Rightarrow \quad \text{Maximum number} = 1.1111\ldots.1 \times 2^{127}$$

$$= (1 - 2^{-23}) \times 2^{127}$$

## Section B : Electronic Devices & Circuits-1 + Advanced Communications Topics-1 + Analog & Digital Communication Systems-2

**Q.5 (a) Solution:**

In steady state, the recombination rate at the surface and in the bulk is equal i.e.

$$\frac{\Delta P_{n,\text{bulk}}}{\tau_{p,\text{bulk}}} = \frac{\Delta P_{n,\text{surface}}}{\tau_{p,\text{surface}}}$$

So, the excess minority carrier concentration at the surface,

$$\Delta P_{n,\text{surface}} = 10^{14} \cdot \frac{10^{-7}}{10^{-6}} = 10^{13}\ \text{cm}^{-3}$$

The generation rate can be determined from the steady-state conditions in the bulk,

$$G = \frac{10^{14}}{10^{-6}} = 10^{20}\ \text{cm}^{-3}\ \text{s}^{-1}$$

The minority carrier diffusion equation for n-type semiconductor is

$$\frac{\partial P_n}{\partial t} = D_p \frac{\partial^2 P_n}{\partial x^2} - \frac{P_x - P_{xo}}{\tau_p} + G$$

At steady state, $\partial P_n / \partial t = 0$.

We can write, $D_p \dfrac{\partial^2 P}{\partial x^2} + G - \dfrac{\Delta P}{\tau_p} = 0$

The boundary conditions are $\Delta P(x = \infty) = \Delta P_{n,\text{bulk}} = 10^{14}\ \text{cm}^{-3}$

and $\Delta P(x = 0) = \Delta P_{n,\text{surface}} = 10^{13}\ \text{cm}^{-3}$

Hence, $\Delta P(x) = \Delta P(\infty) + [\Delta P(0) - \Delta P(\infty)]\exp(-x/L_p)$

$$\Delta P(x) = 10^{14}\,(1 - 0.9e^{-x/L_p})$$

where, $L_p = \sqrt{D_p \cdot \tau_{p.\text{bulk}}} = \sqrt{10 \times 10^{-6}}\ \text{cm} = 31.6\ \mu\text{m}$

**Q.5 (b) Solution:**

**(i)** Received signal strength by mobile, $P_{rm} = -90$ dBm

As a first approximation, the transmitted power of the mobile, $P_{tm}$(dBm) is set as

$$P_{tm}(\text{dBm}) = -76 \text{ dB} - P_{rm}(\text{dBm})$$
$$= -76 \text{ dB} - (-90 \text{ dBm}) = 14 \text{ dBm}$$

**(ii)** Required transmitter power of the mobile = +9 dBm.

Distance in mobile transmitter levels = +14 dBm – (+9 dBm) = +5 dBm

Since, the mobile transmitter power level is adjusted by 1-dB step after every 1.25 ms, i.e. time taken to adjust 1-dB level = 1.25 ms.

Number of steps required to adjust mobile transmitter level to +9 dBm = 5.

Hence, Time needed to adjust mobile transmitter level to +9 dBm

$$= 5 \times 1.25 \text{ ms} = 6.25 \text{ ms}$$

**Q.5 (c) Solution:**

For binary BPSK, the two signals can be represented as

$$s_1(t) = A \cos \omega_c t \text{ and}$$
$$s_2(t) = -A \cos \omega_c t \text{ in the symbol interval } T.$$

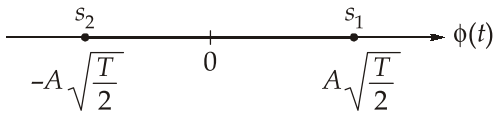
Expressing them in terms of the basis function,

$$\phi(t) = \sqrt{\frac{2}{T}} \cos \omega_c t$$

We get,

$$s_1(t) = A\sqrt{\frac{T}{2}}\phi(t)$$

$$s_2(t) = -A\sqrt{\frac{T}{2}}\phi(t)$$




$$d_{\min}^2 = \left(2A\sqrt{\frac{T}{2}}\right)^2 = 2A^2T$$

Energy per bit,

$$E_b = \frac{A^2T}{2}$$

$$P_{e(\text{BPSK})} = Q\left(\sqrt{\frac{d_{\min}^2}{2N_0}}\right) = Q\left(\sqrt{\frac{2A^2T}{2N_0}}\right)$$

$$P_{e(\text{BPSK})} = Q\left(\sqrt{\frac{A^2T}{N_0}}\right) = Q\left(\sqrt{\frac{2E_b}{N_0}}\right)$$

For binary BFSK orthogonal scheme, the two signals have different frequency and are orthogonal to each other.

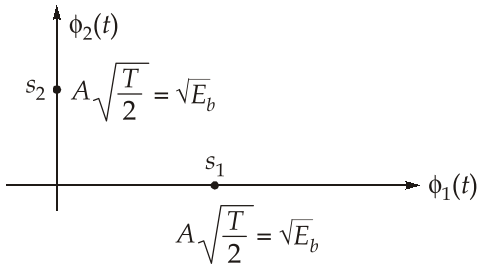
$$s_1(t) = A\cos\omega_1 t$$
$$s_2(t) = A\cos\omega_2 t$$

We need therefore two basis function to represent this system.

$$\phi_1(t) = \sqrt{\frac{2}{T}}\cos\omega_1 t$$

and $$\phi_2(t) = \sqrt{\frac{2}{T}}\cos\omega_2 t$$

we have, $$s_1(t) = A\sqrt{\frac{T}{2}}\phi_1(t)$$




$$s_2(t) = A\sqrt{\frac{T}{2}}\phi_2(t)$$

$$s_1(t) = \sqrt{E_b}\phi_1(t)$$

$$s_2(t) = \sqrt{E_b}\phi_2(t)$$

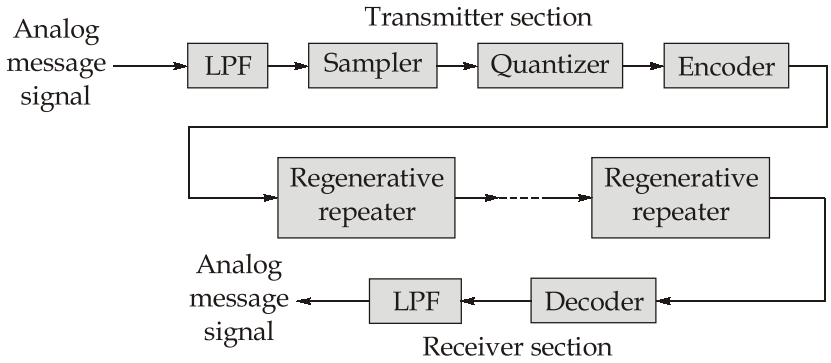
Here, $$E_b = \frac{A^2T}{2}$$ (energy of $s_1$ and $s_2$) and

$$d^2_{\min} = E_b + E_b = 2E_b = A^2T$$

$\therefore$ $$P_{e(\text{BFSK})} = Q\left(\sqrt{\frac{d^2_{\min}}{2N_0}}\right) = Q\left(\sqrt{\frac{A^2T}{2N_0}}\right) = Q\left(\sqrt{\frac{E_b}{N_0}}\right)$$

**Q.5 (d) Solution:**

**(i) Block Diagram:**




1. **Analog Input:** This is the starting point where the continuous time analog signal is introduced. It could be a voice signal from a microphone, music from a CD player, or any other analog data source.
2. **Sampling:**
   - Sampling involves taking regular interval samples of the analog signal to produce discrete samples.
   - The sampling rate is crucial; it determines how often samples are taken and is measured in samples per second or Hertz (Hz).
   - The Nyquist Shannon sampling theorem states that the sampling rate should be atleast twice the highest frequency component of the analog signal to avoid aliasing.
3. **Quantization:**
   - After sampling, each sample's amplitude is quantized to a specific level.
   - This process involves mapping the continuous amplitude values of the samples to a set of discrete levels or quantization levels.
   - The number of quantization levels is determined by the number of bits per sample, which is usually 8, 16, 24 bits etc. More bits mean more quantization levels and better fidelity.
4. **Encoding:**
   - The quantized samples are then encoded into digital binary code.
   - Each quantization level is represented by a unique binary code.
   - The encoder converts the quantized levels into binary numbers, typically using binary codes like binary decimal or gray code.

5. **Digital Output:** The final output is a stream of digital binary codes.

   The digital data can be transmitted over digital communication channels, stored in digital storage media.

   The digital output retains the essential information of the original analog signal but in a format suitable for digital transmission, storage or processing.

6. **Regenerative Repeater:** They are used to compensate the signal loss and reconstruct the signal, and thereby, increase its strength of signal for long distance transmission.

7. **Decoder:** The decoder circuit decodes the pulse coded waveform to reproduce the original signal. This circuit acts as the demodulator.

8. **Low Pass/Reconstruction Filter:** The reconstruction filter, decoder, and quantizer work together as a D/A (Digital to Analog Converter). A reconstruction filter helps in the smooth conversion of the digital signal back to the original analog signal.

**(ii)** Given: $f_m = 3$ kHz, $N = 24$

Nyquist sampling rate; $2f_m = 6$ kHz

$$\text{Actual sampling rate, } f_s = \frac{4}{3} \times 6000$$

$$f_s = 8 \text{ kHz}$$

For a step-size $\Delta$, the maximum quantization error is $\pm\frac{\Delta}{2}$.

$\therefore$ According to the question

$$\frac{\Delta}{2} \leq \frac{0.5}{100}(A_m)$$

where, $$\Delta = \frac{2A_m}{2^n}$$

$\Rightarrow$ $$\frac{2A_m}{2(2^n)} \leq \frac{0.5}{100}(A_m)$$

$\therefore$ $$2^n \geq 200$$

$$n = 8 \text{ bits/sample}$$

The minimum transmission bandwidth required to transmit the multiplexed signal is given as

$$B = \frac{nNf_s}{2} = \frac{8 \times 24 \times 8 \times 10^3}{2} = 768 \text{ kHz}$$

**Q.5 (e) Solution:**

**(i)** Since given parity check matrix $[H]$ is $3 \times 7$ matrix, therefore, we have $n = 7$ and $k = 4$.

We have, $H = [P_T \mid I_{n-k}]$

therefore, $$P^T = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

We can obtain the generator matrix as under

$$G = [I_k : P^T] = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}$$

**(ii)** Codeword can be found by the following expression:

$$X = MG$$

$$X = \begin{bmatrix} 1 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}$$

or $$X = \begin{bmatrix} 1 & 0 & 1 & 0 & 0 & 1 & 1 \end{bmatrix}$$

**(iii)** Given, $$Y = \begin{bmatrix} 0 & 1 & 1 & 1 & 1 & 0 & 0 \end{bmatrix}$$

then, $$S = YH^T = \begin{bmatrix} 0 & 1 & 1 & 1 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$S = \begin{bmatrix} 1 & 0 & 1 \end{bmatrix} = eH^T$$

Hence, we get the error code as 0010000. Therefore, there is an error in 3rd bit; and the correct codeword is 0101100.

**Q.6 (a) Solution:**

**(i) Given:** $E_1 - E_F = 0.30$ eV ; $T = 300$ K

The Fermi-dirac probability function with Boltzmann's approximation,

$$f(E) = \frac{1}{1+\exp\left(\frac{E-E_F}{kT}\right)} \cong \exp\left[\frac{-(E-E_F)}{kT}\right]$$

At $E = E_1$, $\quad f(E_1) = \exp\left[-\frac{(E_1-E_F)}{kT}\right]$

$E_1 - E_F = 0.3$ eV

$kT = 0.0259$ eV

$$\therefore \quad f(E_1) = \exp\left[\frac{-(0.30)}{0.0259}\right] = 9.3 \times 10^{-6}$$

which is the probability of existence of an electron in the energy state $E_1$.

The probability that an energy state at $E = E_2$ is empty is $1 - f(E)$.

From the energy level diagram,

$E_F - E_2 = 1.12 - 0.30 = 0.82$ eV

$$1 - f(E) = 1 - \frac{1}{1+\exp\left(\frac{-(E_F-E_2)}{kT}\right)} = 1 - \frac{1}{1+\exp\left(-\frac{(0.82)}{0.0259}\right)}$$

$$1 - f(E) = \frac{1+\exp\left(-\frac{0.82}{0.0259}\right)-1}{1+\exp\left(\frac{-0.82}{0.0259}\right)}$$

$$1 - f(E) = \frac{1}{\frac{1}{\exp\left(\frac{-0.82}{0.0259}\right)} + \frac{\exp\left(\frac{-0.82}{0.0259}\right)}{\exp\left(\frac{-0.82}{0.0259}\right)}} = \frac{1}{1+\exp\left(\frac{0.82}{0.0259}\right)}$$

From Boltzmann's approximation,

$$1+\exp\left(\frac{0.82}{0.0259}\right) \simeq \exp\left(\frac{0.82}{0.0259}\right)$$

$$\therefore \quad 1 - f(E) = \frac{1}{\exp\left(\frac{0.82}{0.0259}\right)} = \exp\left(\frac{-0.82}{0.0259}\right)$$

$\therefore\ 1 - f(E) = 1.78 \times 10^{-14}$, which is the probability that an energy state at $E = E_2$ is empty.

**(ii)** **Given:** Resistivity, $\rho_1 = 50\ \Omega$-cm, $\rho_2 = 5\ \Omega$-cm; Temperature, $T_1 = 300$ K, $T_2 = 330$ K

We know that, resistivity of intrinsic semiconductor,

$$\rho_1 = \frac{1}{n_i q \mu}, \text{ where } \mu = \mu_n + \mu_p$$

Considering intrinsic concentration, $n_{i1}$ at temperature $T_1$ and $n_{i2}$ at temperature $T_2$. Thus,

$$\frac{\rho_1}{\rho_2} = \left(\frac{\frac{1}{n_{i_1} q \mu}}{\frac{1}{n_{i_2} q \mu}}\right) = \frac{n_{i_2}}{n_{i_1}} \qquad \text{...(i)}$$

But intrinsic carrier concentration,

$$n_i = A \cdot T^{3/2} e^{\frac{-E_g}{2kT}}$$

$\therefore$ $$n_i \propto T^{3/2}\, e^{-\frac{E_g}{2kT}}$$

Thus, $$n_{i_1} = T_1^{3/2}\, e^{-E_g/2kT_1}$$

$$n_{i_2} = T_2^{3/2}\, e^{-E_g/2kT_2}$$

From equation (i), $$\frac{50}{5} = \left(\frac{T_2}{T_1}\right)^{3/2} \frac{e^{-E_g/2kT_2}}{e^{-E_g/2kT_1}}$$

$$10 = \left(\frac{330}{300}\right)^{3/2} \exp\left[-E_g\left(\frac{1}{2kT_2} - \frac{1}{2kT_1}\right)\right]$$

$$8.66 = \exp\left[-E_g\left[\frac{1}{2kT_2} - \frac{1}{2kT_1}\right]\right]$$

where, $$kT_1 = 0.0259 \text{ eV}$$

$$kT_2 = \frac{T_2}{11600} = \frac{330}{11600} = 0.0284 \text{ eV}$$

$$8.66 = \exp\left[-E_g\left[\frac{1}{2 \times 0.0284} - \frac{1}{2 \times 0.0259}\right]\right]$$

$$8.66 = \exp\left[-E_g[17.605 - 19.305]\right]$$

$$8.66 = \exp[1.7\, E_g]$$

$$\ln(8.66) = 1.7\, E_g$$

$$\therefore \quad E_g = 1.27 \text{ eV}$$

**Q.6 (b) Solution:**

We have, total number of allocated traffic channels = 395

Number of cells = 7

Probability of blocking (GOS) < 1 %

**(i)** Average number of calls, $\lambda = A_{avg}/H$

where, $H$ = Call holding time = $\frac{3}{60}$ hour = 0.05 hour

and $A_{avg}$ = Offered traffic load

The number of channels per cell ($N$)

$$N = \frac{\text{Total channels}}{\text{Cells per cluster}} = \frac{395}{7} = 56.43 \approx 57 \text{ traffic channels}$$

Corresponding to blocking probability of 0.01 and N = 57, the offered load ($A_{avg}$) is given to be 44.2 Erlangs.

$$\lambda = \frac{44.2 \text{ Erlangs}}{0.05 \text{ hour}} = 884 \text{ calls per hour}$$

**(ii)** 3-sector configuration cellular system:

Number of sectors per cell = 3

Thus, number of channels per sector = $\frac{57}{3} = 19$ traffic channels

Corresponding to blocking probability of 0.01 and $N$ = 19, the offered load, $A_{avg}$ per sector is given to be 11.2 Erlangs.

Average number of calls in a sector = 11.2 Erlangs/0.05 hours = 224 calls per hour

Average number of calls in a cell, $\lambda = 3 \times 224 = 672$ calls per hour

$\Rightarrow$ Decrease in trunking efficiency = $\frac{(884 - 672)}{884} = 0.24$ or 24%

**(iii)** 6-system configuration cellular system:

Number of sectors per cell = 6

Thus, number of channels per sector = $\frac{57}{6} = 9.5$ traffic channels.

Corresponding to blocking probability of 0.01 and $N$ = 9.5, the offered load, $A_{avg}$ per sector is given to be 4.1 Erlangs.

Average number of calls in a sector = 4.1 Erlangs/0.05 hour = 82 calls per sector.

Average number of calls in a cell, $\lambda = 6 \times 82 = 492$ calls per hour

Decrease in turking efficiency = $\frac{(884 - 492)}{884} = 0.44$ or 44%

**Q.6 (c) Solution:**

**(i)** Given:

Bandwidth (B) = 4 kHz

$\mu = 100$

For $L = 64$, $n = \log_2(L) = 6$

Transmission bandwidth = $nB = 6 \times 4 = 24$ kHz

$$\text{SNR}, \left(\frac{S_0}{N_0}\right)_{dB} = (\alpha + 6n)\ \text{dB}$$

where, $$\alpha = 10\log_{10}\left(\frac{3}{[\ln(1+\mu)]^2}\right)$$

Here, $$\alpha = 10\log_{10}\left[\frac{3}{[\ln(101)]^2}\right] = -8.51$$

Thus, $$\left(\frac{S_0}{N_0}\right) = -8.51 + 6 \times 6$$

$$= 27.49\ \text{dB}$$

Now, for the other case,

$L$ = 256, thus n = 8

Transmission bandwidth = $8 \times 4 = 32$ kHz

$$\left(\frac{S_0}{N_0}\right) = (\alpha + 6n)\ \text{dB}$$

$$= -8.51 + 6 \times 8$$

$$= 39.49\ \text{dB}$$

Comparison:

The difference between two SNR is of 12 dB i.e. a ratio of approximately 16. Hence, the SNR for $L$ = 256 is 16 times the SNR for $L$ = 64. Also, the bandwidth required is about 33% more compare to the later.

**(ii)** **1.** Non-coherent binary modulation is a type of modulation scheme where the receiver does not require knowledge of the phase of the carrier signal. In non-coherent modulation, only the amplitude or frequency of the signal is detected.

**2.** Differential Phase Shift Keying (DPSK) is a specific type of non-coherent binary modulation technique where the phase difference between consecutive symbols is used to encode data. Unlike standard Phase Shift Keying (PSK), which requires coherent detection, DPSK allows for detection without a phase reference at the receiver.

**Merits of DPSK:**

**(a) Simplicity:** DPSK does not require phase synchronization between the transmitter and receiver, making it simpler to implement.

**(b) Robustness:** DPSK is less sensitive to phase variations caused by channel impairments, making it more robust in noisy environments.

**(c) Bandwidth Efficiency:** DPSK can achieve similar data rates as coherent PSK with less bandwidth.

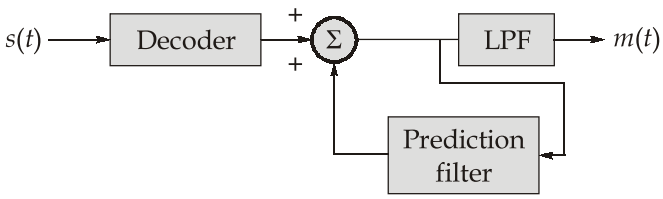
**Demerits of DPSK:**

**(a) Lower Performance:** Due to its non coherent nature, DPSK generally has a higher error rate compared to coherent PSK schemes.

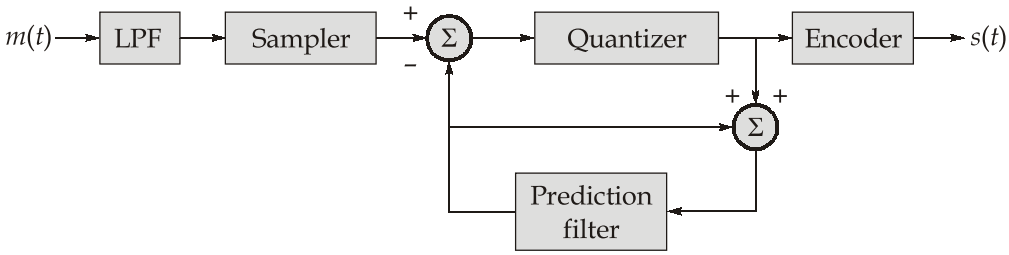
**(b) Limited Data Rates:** DPSK may not support as high data rates as some coherent modulation techniques.

**(c) Complexity in synchronization:** Although DPSK does not require phase synchronization, it may still require synchronization for timing and frequency, which can add complexity in the design.

**3.** **Decoding of DPCM signals:**




Generation of DPCM signals



**Q.7 (a) Solution:**

**(i)** **First group:** Each customer needs 256 $= 2^8$ address. Therefore, 8-bits are needed to define each host. The prefix length is 32 – 8 = 24 bits.

The addresses are:

$1^{st}$ customer = 190.100.0.0/24 – 19.100.0.255/24
$2^{nd}$ customer = 190.100.1.0/24 – 190.100.1.255/24
⋮
$64^{th}$ customer = 190.100.63/0/24 – 190.100.63.255/24
Total addresses allocated = 64 × 256 = 16384

**(ii)** **Second Group:** For this group, each customer needs 128 addresses. Therefore, 7-bits are needed to define each host.

The suffix length is 7 ($\because 2^7 = 128$). The prefix length is then 32 – 7 = 25.

The addresses are:

$1^{st}$ customer = 190.100.64.0/25 – 190.100.64.127/25
$2^{nd}$ customer = 190.100.64.128/25 – 190.100.64.255/25
⋮
$128^{th}$ customer = 190.100.127.128/25 – 190.100.127.255/25
Total addresses allocated = 128 × 128 = 16384

**(iii)** **Third Group:** For this group, each customer needs 64 address. Therefore, 6-bits are needed to define each host.

This means suffix length is 6 ($\because 2^6 = 64$). The prefix length is then 32 – 6 = 26.

The addresses are:

$1^{st}$ customer = 190.100.128.0/26 – 190.100.128.63/26
$2^{nd}$ customer = 190.100.128.64/26 – 190.100.128.127/26
⋮
$128^{th}$ customer = 190.100.159.192/26 – 190.100.159.255/26
Total addresses allocated = 128 × 64 = 8192

Hence, number of addresses granted to the ISP = 65536

Number of allocated addresses by the ISP = 16384 + 16384 + 8192 = 40960.

Number of available addresses by the ISP = 65536 – 40960 = 24576.

Hence, 24576 addresses are available, with the available addresses are from 190.100.160.0 to 190.100.255.255.

**Q.7 (b) Solution:**

**(i) 1.** Given that

$$f_{m1} = 3.6 \text{ kHz}$$
$$f_{m2} = 1.2 \text{ kHz}$$
$$f_{m3} = 1.2 \text{ kHz}$$
$$f_{m4} = 1.2 \text{ kHz}$$

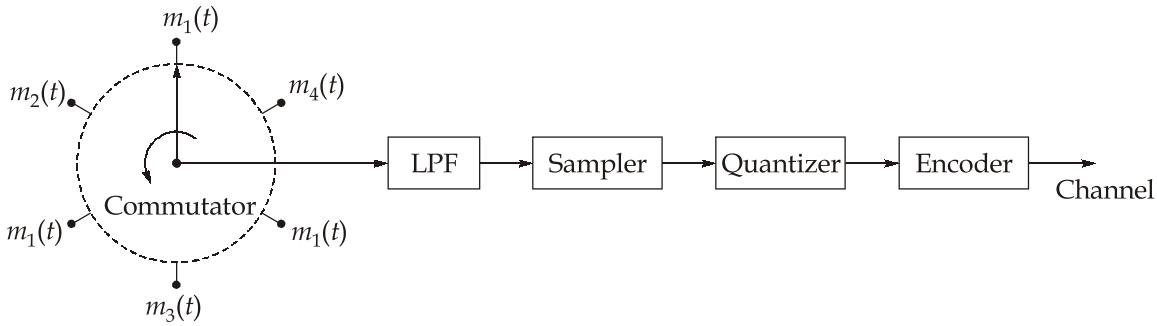
The Nyquist rate for each of the signal can be computed as:

$$f_{s1} = 7.2 \text{ kHz}$$
$$f_{s2} = 2.4 \text{ kHz}$$
$$f_{s3} = 2.4 \text{ kHz}$$
$$f_{s4} = 2.4 \text{ kHz}$$

So, overall sampling rate,

$$f_s = f_{s1} + f_{s2} + f_{s3} + f_{s4} = 14.4 \text{ kHz}$$

If the commutator is rotated at the rate of 2400 revolutions/sec, then in one revolution we obtain 6 samples, one sample each of $m_2(t)$, $m_3(t)$, and $m_4(t)$ and three samples from $m_1(t)$. Thus, the commutator must have atleast six poles connected to the signals as shown below:




**2.** Speed in samples/second = 14400 samples/second

Number of samples in a minute = 14400 × 60

= 864000 samples

1 rotation ..... 6 samples

So, to obtain 864000 samples/minute, commutator should rotate at

$$\frac{864000}{6} = 144000 \text{ rpm}$$

Hence, speed of commutator = 144000 rpm

**3.** Given $M = 1024 = 2^n$

where $M$ = Number of quantization levels and $n$ is the number of bits per sample

Hence, $2^n = 1024 \Rightarrow n = 10$ bits/sample

Bit rate, $R_b = nf_s$

$R_b = 10 \times 14.4 = 144$ kbps

**4.** The minimum channel bandwidth,

$$\text{B.W.} = \frac{nf_s}{2} = \frac{R_b}{2} = 72 \text{ kHz}$$

**(ii)** For $(n, k)$ block code, there are $2^{q-1}$ distinct non-zero syndromes. There are ${}^nC_1 = n$ single error patterns, ${}^nC_2$ double error patterns, ${}^nC_3$ triple error patterns and so on. Therefore, to correct '$t$' errors per word, the following equation should be satisfied:

$$2^{q-1} \geq {}^nC_1 + {}^nC_2 + .... + {}^nC_t$$

This equation describes condition for correction of '$t$' errors per word. For correction of single error per code vector (i.e., $t = 1$), the above equation will have only first term of RHS i.e.,

$$2^{q-1} \geq {}^nC_1$$

Since, $q = n - k$ and ${}^nC_1 = n$

therefore, $2^{(n-k)} - 1 \geq n$

or $(n - k) \geq \log_2(n + 1)$

$$n \geq k + \log_2(n + 1) \quad \text{...(i)}$$

which is proved.

Now, we have to design a linear block code with $d_{min} = 3$ and $k = 8$. We know that the code with $d_{min} = 3$ is a single error correcting code. Hence, we can use the relation given by equation (i) above.

Thus, for single error correction ($d_{min} = 3$)

$$n \geq k + \log_2(n + 1)$$

Substituting $k = 8$ as given, we get

$$n \geq 8 + \log_2(n + 1)$$

On solving, we obtain $n = 12$

Hence, the code will be (12, 8)

Therefore, $q = 12 - 8 = 4$. The parity check matrix will be of size $q \times n$ i.e., parity check matrix size will be $4 \times 12$ i.e.,

$$[H]_{4 \times 12} = [P^T : I_4]_{4 \times 12}$$

The matrix $P^T$ would be of size 4 × 8 and $I_4$ will be an identity matrix of size 4 × 4.

Select the $P^T$ matrix in a manner that

1. Its size should be 4 × 8. Thus, $P$ submatrix which will be of size 8 × 4.
2. No row must be zero, and
3. All rows must be distinct

Such matrix is given below:

$$P^T = \begin{bmatrix} 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 \end{bmatrix}_{4\times 8}$$

Here, it may be noted that we have the freedom of which combination must form a particular column. Therefore, parity check matrix will be

$$H = \left[\begin{array}{cccccccc:cccc} 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 \end{array}\right]_{4\times 12}$$

From the above matrix, the generator matrix and code vectors may be obtained.

**Q.7 (c) Solution:**

**(i)** $$Y = aX + b$$

We have, $$x = \frac{y-b}{a}$$

$$\frac{dy}{dx} = a$$

So, $$f_Y(y) = \frac{1}{a} f_X\left(\frac{y-b}{a}\right)$$

Differential entropy of $Y$,

$$H(Y) = -\int_{-\infty}^{\infty} f_Y(y) \log_2 f_Y(y) dy$$

$$H(Y) = -\int_{-\infty}^{\infty} \frac{1}{a} f_X\left(\frac{y-b}{a}\right) \log_2 \left[\frac{1}{a} f_X\left(\frac{y-b}{a}\right) dy\right]$$

let $$\frac{y-b}{a} = u \quad \Rightarrow \quad dy = a\, du$$

so, $$H(Y) = -\int_{-\infty}^{\infty} f_X(u)\log_2\left[\frac{1}{a}f_X(u)\right]du$$

$$= -\int_{-\infty}^{\infty} f_X(u)\log_2 f_X(u)du + \int_{-\infty}^{\infty} f_X(u)\log_2(a)du$$

$$= H(X) + \log_2 a\int_{-\infty}^{\infty} f_X(\mu)d\mu$$

$$H(Y) = H(X) + \log_2(a)$$

**(ii)** For both the functions to be orthgonal

$$\int_0^{T_b} S_1(t)\cdot S_2(t) = 0$$

We have, $$\int_0^{T_b} S_1(t)\cdot S_2(t)dt = \int_0^{T_b/2} S_1(t)\cdot S_2(t)dt + \int_{\frac{T_b}{2}}^{T_b} S_1(t)S_2(t)dt$$

$$= \int_0^{\frac{T_b}{2}} \frac{2V^2}{T_b}tdt - \int_{\frac{T_b}{2}}^{T_b} 2V^2\left(1-\frac{t}{T_b}\right)dt$$

$$= \frac{2V^2}{T_b}\times\left(\frac{t^2}{2}\right)_0^{T_b/2} - 2V^2\left[(t)_{T_b/2}^{T_b} - \frac{1}{T_b}\left(\frac{t^2}{2}\right)_{T_b/2}^{T_b}\right]$$

$$= \frac{V^2T_b}{4} - V^2T_b + \frac{3V^2T_b}{4} = 0$$

Hence, $S_1(t)$ and $S_2(t)$ are orthogonal.

The energy of the signals $S_1(t)$ and $S_2(t)$ and be calculated as below:

$$E_1 = \text{Energy of } S_1(t) = 2\int_0^{\frac{T_b}{2}}\left(\frac{2V}{T_b}\right)^2 t^2dt$$

$$= \frac{8V^2}{T_b^2}\times\frac{T_b^3}{8\times 3} = \frac{V^2T_b}{3}$$

$$E_2 = \text{Energy of } S_2(t) = \int_0^{T_b} V^2\cdot dt = V^2T_b$$

**Q.8 (a) Solution:**

**(i)** The given system is an ASK system. Here,

$$S_1(t) = A\sin\left(\frac{\pi t}{T}\right) \text{ and } S_2(t) = 0$$

Writing in terms of the orthonormal basis function $\phi(t)$,

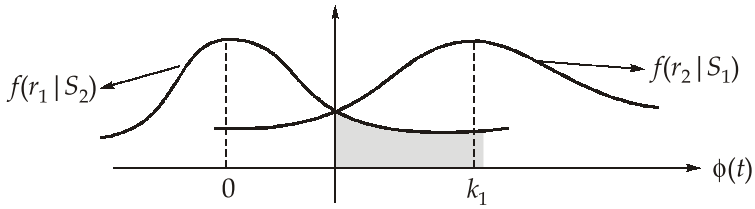
$$S_1(t) = \sqrt{\frac{A^2T}{2}}\cdot\sqrt{\frac{2}{T}}\sin\left(\frac{\pi t}{T}\right)$$

$$S_2(t) = 0$$

Thus, $S_1(t) = k_1\phi(t)$

$S_2(t) = k_2\phi(t)$ where $k_2 = 0$

The conditional probability density function of the received signal $r_1$ and $r_2$ corresponding to the transmitted signals $S_1$ and $S_2$ is given below:




$$P_e = \frac{1}{2}P_e\left(\frac{r_2}{S_1}\right) + \frac{1}{2}P_e\left(\frac{r_1}{S_2}\right)$$

($\because$ the signals are equiprobable signals)

$$\therefore \quad P_e = P_e\left(\frac{r_2}{S_1}\right)$$ ($\because$ both are same)

Thus, in terms of $Q$ function, it can be written as

$$P_e = Q\left(\frac{d}{\sqrt{2N_0}}\right)$$

$$k_1 = \sqrt{\frac{A^2T}{2}} \quad \text{and } k_2 = 0$$

$$\therefore \quad d = k_1 - k_2 = \sqrt{\frac{A^2T}{2}}$$

$$\therefore \quad P_e = Q\left(\sqrt{\frac{A^2T}{4N_0}}\right) = Q\left(\sqrt{\frac{0.04\times10^{-6}\times2\times10^{-6}}{4\times2\times10^{-15}}}\right)$$

$$P_e = Q\left(\sqrt{10}\right)$$

**(ii)** Mean of random variable '$X$':

$$E[X] = \int_{-\infty}^{\infty} x \cdot f_X(x)dx$$

$$= \int_{0}^{\infty} x \cdot \frac{x}{\sigma^2} e^{-\frac{x^2}{2\sigma^2}} dx = \frac{1}{\sigma^2}\int_{0}^{\infty} x^2 \cdot e^{\frac{-x^2}{2\sigma^2}} dx$$

We know, for a Gaussian random variable for zero mean and variance $\sigma^2$,

$$\frac{1}{\sqrt{2\pi\sigma^2}}\int_{-\infty}^{\infty} x^2 \cdot e^{\frac{-x^2}{2\sigma^2}} dx = \sigma^2$$

Since, the given function under integration is an even function, so,

$$\frac{1}{\sqrt{2\pi\sigma^2}}\int_{0}^{\infty} x^2 \cdot e^{\frac{-x^2}{2\sigma^2}} dx = \frac{1}{2}\sigma^2$$

$$\frac{1}{\sigma^2}\int_{0}^{\infty} x^2 \cdot e^{\frac{-x^2}{2\sigma^2}} dx = \frac{\sqrt{2\pi\sigma^2}}{2} = \sigma\sqrt{\frac{\pi}{2}}$$

Hence, $$E[X] = \frac{1}{\sigma^2}\int_{0}^{\infty} x^2 e^{\frac{-x^2}{2\sigma^2}} dx = \sigma\sqrt{\frac{\pi}{2}}$$

$$E[X] = \sigma\sqrt{\frac{\pi}{2}}$$

To find VAR[$X$] we need to calculate $E[X^2]$,

$$E[X^2] = \int_{-\infty}^{\infty} x^2 f_X(x)dx = \int_{0}^{\infty} x^2 \cdot \frac{x}{\sigma^2} e^{\frac{-x^2}{2\sigma^2}} dx$$

Let $\frac{x^2}{2\sigma^2} = t \quad \Rightarrow \quad \frac{2x}{2\sigma^2}dx = dt \quad \Rightarrow \quad \frac{x}{\sigma^2}dx = dt$

$$= \int_{0}^{\infty} 2\sigma^2 t \cdot e^{-t} dt = 2\sigma^2 \int_{0}^{\infty} t \cdot e^{-t} dt$$

$$= 2\sigma^2\left[\left\{-t(e^{-t})\right\}_0^{\infty} - \left\{e^{-t}\right\}_0^{\infty}\right]$$

$$= 2\sigma^2[0 - e^{-\infty} + e^{0}] = 2\sigma^2$$

$$E[X^2] = 2\sigma^2$$

$$\text{VAR}[X] = E[X^2] - \{E[X]\}^2$$

$$= 2\sigma^2 - \left(\sigma\sqrt{\frac{\pi}{2}}\right)^2 = 2\sigma^2 - \frac{\pi}{2}\sigma^2$$

$$\text{VAR}[X] = \left(2 - \frac{\pi}{2}\right)\sigma^2$$

**Q.8 (b) Solution:**

**(i)** The average power of the message signal is

$$P_m = \int_{-W}^{W} S_M(f)df = 2S_0 f_0 \int_0^W \frac{(1/f_0)}{1+\left(\frac{f}{f_0}\right)^2} df$$

$$= 2S_0 f_0 \left[\tan^{-1}\left(\frac{f}{f_0}\right)\right]_0^W$$

$$= 2S_0 f_0 \tan^{-1}\left(\frac{W}{f_0}\right)$$

The average power of the emphasized signal is

$$P_y = \int_{-W}^{W} S_M(f)\left|H(f)\right|^2 df$$

$$P_y = \int_{-W}^{W} S_0 K^2 df = 2S_0 K^2 W$$

To have $P_y = P_m$,

$$2S_0K^2W = 2S_0 f_0 \tan^{-1}\left(\frac{W}{f_0}\right)$$

$$K = \sqrt{\left(\frac{f_0}{W}\right)\tan^{-1}\left(\frac{W}{f_0}\right)}$$

$$K = \sqrt{\frac{1}{2}\tan^{-1}(2)}$$

$$K = 0.744$$

**(ii) 1.** Given data:

$$B = 4\text{ kHz},\ \frac{S}{N} = 28$$

From the Shannon's channel capacity theorem,

$$\text{Channel capacity, } C = B\log_2\left(1+\frac{S}{N}\right)$$

$$C = 4\times10^3\log_2(1+28)$$

$$C = 19.432\text{ kbps}$$

Now, when BW is doubled i.e., $B' = 8$ kHz and signal power is constant, then

$$\text{Capacity, } C = B'\log_2\left(1+\frac{S}{N}\right)$$

$$C = 8\times10^3\log_2\left(1+\frac{S}{N_0B'}\right)$$

$$C = 8\times10^3\log_2\left(1+\frac{S}{N_0\times 2B}\right)$$

$$C = 8\times10^3\log_2\left[1+\left(\frac{S}{N_0B}\right)\times\frac{1}{2}\right]$$

But, $\quad \dfrac{S}{N_0B} = 28$

$\therefore$ $\quad C = 8\times10^3\log_2\left(1+\dfrac{28}{2}\right)$

$$C = 8\times10^3\log_2(15)$$

$$C = 31.255\text{ kbps}$$

**2.** From the Shannon's channel capacity theorem,

$$C = B\log_2\left(1+\frac{S}{N}\right)$$

where,

$C$ = Capacity of the channel

$B$ = Bandwidth of channel

$S$ = Average signal power

$N$ = Average noise power in the channel

$N = N_0B$

When the channel is ideal with infinite bandwidth,

$$C_\infty = \lim_{B\to\infty} B\log_2\left[1+\frac{S}{N_0B}\right]$$

$$= \underset{B\to\infty}{lt}\frac{\log_2\left[1+\frac{S}{N_0B}\right]}{\left(\frac{1}{B}\right)} = \underset{B\to\infty}{lt}\frac{\ln\left(1+\frac{S}{N_0B}\right)}{\ln(2)\times\left(\frac{1}{B}\right)}$$

Using L-Hospital's Rule,

$$C_\infty = \frac{1}{\ln 2}\underset{B\to\infty}{lt}\left[\frac{1\times\left(\frac{-1}{B^2}\right)\times\frac{S}{N_0}}{\left(1+\frac{S}{N_0B}\right)\times\left(\frac{-1}{B^2}\right)}\right]$$

$$= \frac{1}{\ln 2}\underset{B\to\infty}{lt}\left[\frac{\left(\frac{S}{N_0}\right)}{\left(1+\frac{S}{N_0B}\right)}\right]$$

$$C_\infty = \frac{1}{\ln 2}\times\left(\frac{S}{N_0}\right)\frac{\text{bits}}{\text{symbol}}$$

**Q.8 (c) Solution:**

Given, $R = \dfrac{np-n_i^2}{\tau_{p0}(n+n')+\tau_{n0}(p+p')}$

also given, $\tau_{p0} = \tau_{n0} = \tau_0$ and $n' = p' = n_i$

We know that, $n = n_i\exp\left(\dfrac{E_{Fn}-E_{Fi}}{kT}\right)$

$$p = n_i\exp\left(\frac{E_{Fi}-E_{Fp}}{kT}\right)$$

We have, $(E_{Fn}-E_{Fi})+(E_{Fi}-E_{Fp}) = qV_a$

$$(E_{Fi}-E_{Fp}) = qV_a-(E_{Fn}-E_{Fi})$$

then, $p = n_i\exp\left[\dfrac{qV_a-(E_{Fn}-E_{Fi})}{kT}\right]$

$$= n_i \exp\left(\frac{qV_a}{kT}\right)\exp\left[\frac{-(E_{Fn}-E_{Fi})}{kT}\right]$$

Let $V = \dfrac{qV_a}{kT}$ and $V' = \left(\dfrac{E_{Fn}-E_{Fi}}{kT}\right)$

Then the recombination rate can be written as

$$R = \frac{\left(n_i e^{V'}\right)\left(n_i e^{V}\cdot e^{-V'}\right) - n_i^2}{\tau_0\left[n_i e^{V'} + n_i + n_i e^{V}\cdot e^{-V'} + n_i\right]}$$

(or) $$R = \frac{n_i(e^V - 1)}{\tau_0(2 + e^{V'} + e^V \cdot e^{-V'})}$$

For maximum recombination rate, set

$$\frac{dR}{dV'} = 0$$

$$0 = \frac{n_i(e^V-1)}{\tau_0}\frac{d}{dV'}\left[2 + e^{V'} + e^V\cdot e^{-V'}\right]^{-1}$$

$$0 = \frac{n_i(e^V-1)}{\tau_0}(-1)\left[2 + e^{V'} + e^V\cdot e^{-V'}\right]^{-2}\times\left[e^{V'} - e^V\cdot e^{-V'}\right]$$

$$0 = \frac{-n_i(e^V-1)}{\tau_0}\cdot\frac{\left[e^{V'} - e^V\cdot e^{-V'}\right]}{\left[2 + e^{V'} + e^V\cdot e^{-V'}\right]^2}$$

$\Rightarrow \quad e^{V'} - e^V\cdot e^{-V'} = 0$

$$e^{2V'} = e^V \Rightarrow V' = \frac{1}{2}V$$

Then the maximum recombination rate becomes

$$R_{max} = \frac{n_i\left(e^V - 1\right)}{\tau_0\left[2 + e^{V/2} + e^V\cdot e^{-V/2}\right]}$$

$$= \frac{n_i\left(e^V - 1\right)}{\tau_0\left[2 + e^{V/2} + e^{V/2}\right]}$$

$$= \frac{n_i\left(e^V - 1\right)}{2\tau_0\left[e^{V/2} + 1\right]} = \frac{n_i\left[\exp\left(\dfrac{qV_a}{kT}\right) - 1\right]}{2\tau_0\left[\exp\left(\dfrac{qV_a}{2kT}\right) + 1\right]}$$

If $V_a >> \frac{kT}{q}$, then neglect '–1' in numerator and +1 in denominator

$$R_{max} = \frac{n_i\left[\exp\left(\frac{qV_a}{kT}\right)\right]}{2\tau_0\left[\exp\left(\frac{qV_a}{2kT}\right)\right]}$$

$$= \frac{n_i}{2\tau_0}\exp\left[\frac{qV_a}{kT} - \frac{qV_a}{2kT}\right]$$

$$R_{max} = \frac{n_i}{2\tau_0}\exp\left[\frac{qV_a}{2kT}\right]$$

❍❍❍❍