**1. Explain Data, Control and Structural hazards with suitable examples**

**i. Data Hazards:** A data hazard is any condition in which either the source or the destination operands of an instruction are not available at the time expected in the pipeline. As a result of which some operation has to be delayed and the pipeline stalls.

Whenever there are two instructions one of which depends on the data obtained from the other.

A=3+A

B=A\*4

For the above sequence, the second instruction needs the value of ‘A’ computed in the first instruction.

Thus the second instruction is said to depend on the first.

If the execution is done in a pipelined processor, it is highly likely that the interleaving of these two instructions can lead to incorrect results due to data dependency between the instructions. Thus the pipeline needs to be stalled as and when necessary to avoid errors.

For example:

i1. **R2** <- R5 + R3

i2. R4 <- **R2** + R3

**ii. Structural Hazards:**

This situation arises mainly when two instructions require a given hardware resource at the same time and hence for one of the instructions the pipeline needs to be stalled.The most common case is when memory is accessed at the same time by two instructions. One instruction may need to access the memory as part of the Execute or Write back phase while other instruction is being fetched. In this case if both the instructions and data reside in the same memory. Both the instructions can’t proceed together and one of them needs to be stalled till the other is done with the memory access part.

**iii. Control hazards:**

The instruction fetch unit of the CPU is responsible for providing a stream of instructions to the execution unit. The instructions fetched by the fetch unit are in consecutive memory locations and they are executed.

However the problem arises when one of the instructions is a branching instruction to some other memory location. Thus all the instruction fetched in the pipeline from consecutive memory locations are invalid now and need to removed(also called flushing of the pipeline).This induces a stall till new instructions are again fetched from the memory address specified in the branch instruction.

Thus the time lost as a result of this is called a branch penalty. Often dedicated hardware is incorporated in the fetch unit to identify branch instructions and compute branch addresses as soon as possible and reducing the resulting delay as a result.

**2)write a brief note on virtual memory and cache memory**

cache memory:- The cache memory (Pronounced as "cash") is the volatile computer memory which is very nearest to the CPU so also called **CPU memory**, all the Recent Instructions are Stored into the Cache Memory. It is the fastest memory that provides high-speed data access to a computer microprocessor Cache meaning is that it is used for storing the input which is given by the user and which is necessary for the computer microprocessor to Perform a Task. But the Capacity of the Cache Memory is too low in compare to Memory (random access memory (RAM)) and Hard Disk.

It acts as a high speed buffer between CPU and main memory and is used to temporary store very active data and action during processing since the cache memory is faster then main memory, the processing speed is increased by making the data and instructions needed in current processing available in cache. The cache memory is very expensive and hence is limited in capacity.

**Type of Cache memory**

Cache memory improves the speed of the CPU, but it is expensive.Type of Cache Memory is divided into different level that are L1,L2,L3:

**Level 1 (L1) cache or Primary Cache**

L1 is the primary type cache memory. The Size of the L1 cache very small comparison to others that is between 2KB to 64KB, it depent on computer processor. It is a embedded register in the computer microprocessor(CPU).The Instructions that are required by the CPU that are  firstly searched in L1 Cache. Example of registers are accumulator, address register,, Program counter etc.

**Level 2 (L2) cache or Secondary Cache**

L2 is seconday type cache memory. The Size of the L2 cache is more capacious than L1 that is between 256KB to 512KB.L2 cache is Located on computer microprocessor.After searching the Instructions in L1 Cache,if not found then it searched into L2 cache by computer microprocessor. The high-speed system bus interconnecting the cache to the microprocessor.

**Level 3 (L3) cache or Main Memory**

The L3 cache is larger in size but also slower in speed than L1 and L2,it's size is between 1MB to 8MB.In Multicore processors, each core may have seperate L1 and L2,but all core share a common L3 cache. L3 cache double speed than the RAM.

**Virtual memory:-** **Virtual memory** is the feature of an  (OS). It is responsible for memory management.In the Virtual Memory the Physical Memory (Hard Disk) will be treated as the Logical Memory (random access memory (RAM)). Means with the help of virtual Memory we can also temporarily increase the size of Logical Memory as from the Physical Memory. A user will see or feels that all the Programs are running into the Logical Memory of the Computer. With the help of virtual Memory all the Space of Hard Disk can be used as the Logical Memory So that a user can execute any Number of programs.

**Benefits of Virtual Memory**

1) **Unused Address space**: With the help of Unused Address Space a user can execute any number of programs because all the actual Addresses will be treated as the logical Addresses. All the Programs those are given by the user will be Stored into the Disk Space and all the Programs will be Stored into the Physical Address Space but they will treat as they are Stored into the Logical Address Space.

2) **Increased degree of Multiprogramming**: With the help of Virtual Memory we can Execute Many Programs at a Time because Many Programs can be fit in the Physical Memory So that More Programs can be Stored into the Memory but this will not increase the Response Time of the CPU Means this will not affect on the Execution of the Programs.

3) **Decrease Number of I/O Operations**: There will be less Operations those are to be used for performing the Swapping of the Processes. All the Programs will be automatically will be loaded into the Memory when they are needed.

**3. Write a note on a) Direct Mapping b) Fully associative mapping and c) k-way associative mapping**

**Direct Mapping –**  
The simplest technique, known as direct mapping, maps each block of main memory into only one possible cache line. or  
In Direct mapping, assigne each memory block to a specific line in the cache. If a line is previously taken up by a memory block when a new block needs to be loaded, the old block is trashed. An address space is split into two parts index field and a tag field. The cache is used to store the tag field whereas the rest is stored in the main memory. Direct mapping`s performance is directly proportional to the Hit ratio.

i = j modulo m

where

i=cache line number

j= main memory block number

m=number of lines in the cache

For purposes of cache access, each main memory address can be viewed as consisting of three fields. The least significant w bits identify a unique word or byte within a block of main memory. In most contemporary machines, the address is at the byte level. The remaining s bits specify one of the 2s blocks of main memory. The cache logic interprets these s bits as a tag of s-r bits (most significant portion) and a line field of r bits. This latter field identifies one of the m=2r lines of the cache

**Fully Associative Cache:**A type of CACHE designed to solve the problem of cache CONTENTION that plagues the DIRECT MAPPED CACHE. In a fully associative cache, a data block from any memory address may be stored into any CACHE LINE, and the whole address is used as the cache TAG: hence, when looking for a match, all the tags must be compared simultaneously with any requested address, which demands expensive extra hardware. However, contention is avoided completely, as no block need ever be flushed unless the whole cache is full, and then the least recently used may be chosen..

**k-way associative mapping:-**

 After CPU generates a memory request,

* The set number field of the address is used to access the particular set of the cache.
* The tag field of the CPU address is then compared with the tags of all k lines within that set.
* If the CPU tag matches to the tag of any cache line, a cache hit occurs.
* If the CPU tag does not match to the tag of any cache line, a cache miss occurs.
* In case of a cache miss, the required word has to be brought from the main memory.
* If the cache is full, a replacement is made in accordance with the employed replacement policy