UNIT I

(Two Marks Questions & Answers)

- **Discuss the different ways how instruction set architecture can be classified?**
  Stack Architecture ,Accumulator Architecture, Register-Memory Architecture ,Register-Register Architecture.

- **Explain the concept behind pipelining**.
  Pipelining is an implementation technique whereby multiple instructions are overlapped in execution. It takes advantage of parallelism that exists among actions needed to execute an instruction.

- **Explain pipeline hazard and mention the different hazards in pipeline**.

  Hazards are situations that prevent the next instruction in the instruction stream from executing during its designated clock cycle. Hazards reduce the overall performance from the ideal speedup gained by pipelining. The three classes of hazards are,

  Structural hazards,

  Data hazards,

  Control hazards.

- **What are the different types of data hazards?**

  RAW Hazard(Read-After-Write)

  WAR Hazard(Write-After-Read)

  WAW Hazard(Write-After-Write)

- **List various data dependence**.

  True Data dependence ,Name dependence, Control Dependence

- **What is name dependence ? When it oocurs?**

Name dependence occurs when two instructions **use** the same register or memory location, called a *name*, but there is no flow of data between the instructions associated with that name. Antidependence and output dependence are two kinds of name dependences.

- **What are antidependence & output dependence ?**

  There are two types of name dependences between an instruction *i* that *precedes* instruction *j* in program order:

  (1) An *antidependence* between instruction *i* and instruction *j* occurs when instruction *j* writes a register or memory location that instruction *i* reads. The original ordering must be preserved to ensure that *i* reads the correct value.

  (2)An *output dependence* occurs when instruction *i* and instruction *j* write the same register or memory location. The ordering between the instructions must be preserved to ensure that the value finally written corresponds to instruction *j*.

  The antidependences and output dependences can be overcome by register renaming.

- **Explain the concept of forwarding**.

  Forwarding can be generalized to include passing a result directly to the functional unit that fetches it. The result is forwarded from the pipeline register corresponding to the output of one unit to the input of the same unit.

- **Mention the different schemes to reduce pipeline branch penalties**.

  a. Freeze or flush the pipeline

  b. Treat every branch as not taken

  c. Treat every branch as taken

  d. Delayed branch

- **Consider an unpipelined processor. Assume that it has a 1ns clock cycle and that it uses 4 cycles for ALU operations and branches and 5 cycles for memory operations. Assume that the relative frequencies of these operations are 40%, 20% and 40% respectively. Suppose that due to clock skew and setup, pipelining the processor adds 0.2 ns of overhead to the clock. Ignoring any latency impact, how much speedup in the instruction execution rate will we gain from a pipeline?**

The average instruction execution time on an unpipelined processor is = clock cycle x Average CPI = 1 ns x ((40% x 4)+(20 x 4)+(40 x 5)) = 4.4 ns.

The average instruction execution time on an pipelined processor is = 1+0.2ns = 1.2ns Speedup = Avg. instruction time unpipelined/ Avg. instruction time pipelined = 4.4/1.2 = 3.7 times

- **When do data hazards arise?**

  Data hazards arise when an instruction depends on the results of a previous instruction in a way that is expressed by the overlapping of instructions in the pipeline.

- **What is Instruction Level Parallelism?**

  Pipelining is used to overlap the execution of instructions and improve performance. This potential overlap among instructions is called instruction level parallelism (ILP) since the instruction can be evaluated in parallel.

- **Give an example of control dependence?**

  ```
   if p1
    {s1;}
  if p2
   {s2;}
  ```

  S1 is control dependent on p1, and s2 is control dependent on p2.

- **What is the limitation of the simple pipelining technique?**

  These technique uses in-order instruction issue and execution. Instructions are issued in program order, and if an instruction is stalled in the pipeline, no later instructions can proceed.

- **Briefly explain the idea behind using reservation station?**

  Reservation station fetches and buffers an operand as soon as available, eliminating the need to get the operand from a register.

- **Give an example for data dependence.**

  ```
  Loop:   L.D F0,0(R1)
   ADD.D F4,F0,F2
  S.D F4,0(R1)
  DADDUI R1,R1,#-8
  BNE R1,R2, loop
  ```

- **Explain the idea behind dynamic scheduling?**

  In dynamic scheduling the hardware rearranges the instruction execution to reduce the stalls while maintaining data flow and exception behavior.

- **Mention the advantages of using dynamic scheduling?**

  It enables handling some cases when dependences are unknown at compile time and it simplifies the compiler. It allows code that was compiled with one pipeline in mind run efficiently on a different pipeline.

- **What are the possibilities for imprecise exception?**

  The pipeline may have already completed instructions that are later in program order than instruction causing exception. The pipeline may have not yet completed some instructions that are earlier in program order than the instructions causing exception.

- **What are multilevel branch predictors?**

  These predictors use several levels of branch-prediction tables together with an algorithm for choosing among the multiple predictors.

- **What are branch-target buffers?**

  To reduce the branch penalty we need to know from what address to fetch by end of IF (instruction fetch). A branch prediction cache that stores the predicted address for the next instruction after a branch is called a branch-target buffer or branch target cache.

UNIT II

(Two Marks Questions & Answers)

- **What is loop unrolling?**
  A simple scheme for increasing the number of instructions relative to the branch and overhead instructions is loop unrolling. Unrolling simply replicates the loop body multiple times, adjusting the loop termination code.

- **When static branch predictors are used?**
  They are used in processors where the expectation is that the branch behavior is highly predictable at compile time. Static predictors are also used to assists dynamic predictors.

- **Mention the different methods to predict branch behavior?**
  Predict the branch as taken Predict on basis of branch direction (either forward or backward) Predict using profile information collected from earlier runs.

- **Explain the VLIW approach?**
  They uses multiple, independent functional units. Rather than attempting to issue multiple, independent instructions to the units, a VLIW packages the multiple operations into one very long instruction.

- **Mention the techniques to compact the code size in instructions?**
  Using encoding techniques Compress the instruction in main memory and expand them when they are read into the cache or are decoded.

- **Mention the advantage of using multiple issue processor?**
  They are less expensive.
  They have cache based memory system.
  More parallelism.

- **What are loop carried dependence?**
  They focuses on determining whether data accesses in later iterations are dependent on data values produced in earlier iterations; Such a dependence is called loop carried dependence.

  Example

  ```
  for(i=0;i<100;i++)
  x[i]=x[i]+s;                    No loop carried dependence

  for(i=0;i<100;i++)
  x[i+1]=x[i]+s;                Loop carried dependence
  ```

- **Mention the tasks involved in finding dependences in instructions?**

  Good scheduling of code. Determining which loops might contain parallelism Eliminating name dependence .

- **Use the G.C.D test to determine whether dependence exists in the following loop:**

  for(i=1;i<=100;i=i+1)

  X[2*i+3]=X[2*i]*5.0;

  Solution:

  a=2,b=3,c=2,d=0

  GCD(a,c)=2 and d-b=-3

  Since 2 does not divide -3, no dependence is possible.

- **What is software pipelining?**

  Software pipelining is a technique for reorganizing loops such that each iteration in the software pipelined code is made from instruction chosen from different iterations of the original loop.

- **What is global code scheduling?**

  Global code scheduling aims o compact code fragment with internal control structure into the shortest possible sequence that preserves the data and control dependence. Finding a shortest possible sequence is finding the shortest sequence for the critical path.

- **What is trace?**

  Trace selection tries to find a likely sequence of basic blocks whose operations will be put together into a smaller number of instructions; this sequence is called trace.

- **Mention the steps followed in trace scheduling?**
  - Trace selection
  - Trace compaction

- **What is superblock?**

  Superblocks are formed by a process similar to that used for traces, but are a form of extended basic block, which are restricted to a single entry point but allow multiple exits.

- **Mention the advantages of predicated instructions?**
  - Remove control dependence .
  - Maintain data flow enforced by branch
  - Reduce overhead of global code scheduling

- **Mention the limitations of predicated instructions?**

They are useful only when the predicate can be evaluated early. Predicated instructions may have speed penalty.

- **What is poison bit?**

  Poison bits are a set of status bits that are attached to the result registers written by the speculated instruction when the instruction causes exceptions. The poison bits cause a fault hen a normal instruction attempts to use the register.

- **What are the disadvantages of supporting speculation in hardware?**

  Complexity Additional hardware resources required

- **Mention the methods for preserving exception behavior?**
  - Ignore Exception
  - Instructions that never raise exceptions are used
  - Using poison bits
  - Using hardware buffers

- **What is an instruction group?**

  It is a sequence of consecutive instructions with no register data dependence among them. All the instructions in the group could be executed in parallel. An instruction group can be arbitrarily long.

UNIT III

(Two Marks Questions & Answers)

- **What are multiprocessors? Mention the categories of multiprocessors?**

  Multiprocessors are used to increase performance and improve availability. The different categories are SISD, SIMD, MISD, MIMD

- **What are threads?**

  These are multiple processors executing a single program and sharing the code and most of their address space. When multiple processors share code and data in the way, they are often called threads.

- **What is cache coherence problem?**

  Two different processors have two different values for the same location.

- **What are the protocols to maintain coherence?**

  Directory based protocol and Snooping Protocol

- **What are the ways to maintain coherence using snooping protocol?**

  Write Invalidate protocol

  Write update or write broadcast protocol

- **What is write invalidate and write update?**

  Write invalidate provide exclusive access to caches. This exclusive caches ensure that no other readable or writeable copies of an item exists when the write occurs. Write update updates all cached copies of a data item when that item is written.

- **What are the disadvantages of using symmetric shared memory?**

  Compiler mechanisms are very limited. Larger latency for remote memory access Fetching multiple words in a single cache block will increase the cost.

- **Mention the information in the directory?**

  It keeps the state of each block that are cached. It keeps track of which caches have copies of the block.

- **What the operations that a directory based protocol handle?**

  Handling read miss and Handling a write to a shares clean cache block

- **What are the states of cache block in the directory of the directory based cache coherence implementation?**

  Uncached

Shared,

Exclusive

- **What are the states of a cache block in a snoopy based scheme**.

  Invalid,

  Shared,

  Exclusive

- **What are the uses of having a bit vector?**

  When a block is shared, the bit vector indicates whether the processor has the copy of the block. When block is in exclusive state, bit vector keep track of the owner of the block.

- **When do we say that a cache block is exclusive?**

  When exactly one processor has the copy of the cached block, and it has written the block. The processor is called the owner of the block.

- **Explain the types of messages that can be send between the processors and directories?**

  Local node – Node where the requests originates

  Home Node – Node where memory location and directory entry of the address resides.

  Remote Node - The copy of the block in the third node called remote node

- **What is consistency?**

  Consistency says in what order must a processor observe the data writes of another processor.

- **Mention the models that are used for consistency**?

  Sequential consistency

  Relaxed consistency model

- **What is sequential consistency?**

  It requires that the result of any execution be the same, as if the memory accesses executed by each processor were kept in order and the accesses among different processors were interleaved.

- **What is relaxed consistency model?**

  Relaxed consistency model allows reads and writes to be executed out of order. The three sets of ordering are: W-> R ordering W->W ordering R->W and R-> R ordering.

- **What is multi threading?**

  Multithreading allows multiple threads to share the functional units of the single processor in an overlapping fashion.

- **What is fine grained multithreading?**

  It switches between threads on each instruction, causing the execution of multiple threads to be interleaved.

- **What is coarse grained multithreading?**

  It switches threads only on costly stalls. Thus it is much less likely to slow down the execution of an individual thread.

UNIT IV

Memory & I/O

(Two Marks Questions & Answers)

- **What is cache miss and cache hit?**

  When the CPU finds a requested data item in the cache, it is called cache hit. When the CPU does not find that data item it needs in the cache, a cache miss occurs.

- **What is write through and write back cache?**

  Write through- the information is written to both the block in the cache and to the block in the lower level memory.

  write back- The information is written only to the block in the cahce. The modified cache block is written to main memory only when it is replaced.

- **What is miss rate and miss penalty?**

  Miss rate is the fraction of cache access that result in a miss. Miss penalty depends on the number of misses and clock per miss.

- **Give the equation for average memory access time?**

  Average memory access time= Hit time + Miss rate x Miss penalty

- **What is striping?**

  Spreading multiple data over multiple disks is called striping, which automatically forces accesses to several disks.

- **Mention the problems with disk arrays?**

  When devices increases, dependability increases Disk arrays become unusable after a single failure .

- **What is hot spare?**

  Hot spares are extra disks that are not used in normal operation. When failure occurs, an idle hot spare is pressed into service. Thus, hot spares reduce the MTTR.

- **What is mirroring?**

  Disks in the configuration are mirrored or copied to another disk. With this arrangement data on the failed disks can be replaced by reading it from the other mirrored disks.

- **Mention the drawbacks with mirroring?**

  - Writing onto the disk is slower
  - Since the disks are not synchronized seek time will be different
  - Imposes 50% space penalty hence expensive.

- **Mention the factors that measure I/O performance measures?**

  Diversity Capacity Response time Throughput Interference of I/o with CPU execution.

- **What is transaction time?**

  The sum of entry time, response time and think time is called transaction time.

- **State little's law?**

  Littles law relates the average number of tasks in the system. Average arrival rate of new asks. Average time to perform a task.

- **Give the equation for mean number of tasks in the system?**

  Mean number of arrival in the system = Arrival rate x Mean response time.

- **What is server utilization?**

  Mean number of tasks being serviced divided by service rate

Server utilization = Arrival Rate/Server Rate

The value should be between 0 and 1 otherwise there would be more tasks arriving than could be serviced.

- **What are the steps to design an I/O system?**

  - cost-performance design and evaluation
  - Availability of design
  - Response time
  - Realistic cost-performance, design and evaluation
  - Realistic design for availability and its evaluation.

- **Briefly discuss about classification of buses?**

  I/O buses - These buses are lengthy and have any types of devices connected to it.

  CPU memory buses – They are short and generally of high speed. Connects memory and CPU.

- **Explain about bus transactions?**

  Read transaction – Transfer data from memory

  Write transaction – Writes data to memory

- **What is the bus master?**

  Bus masters are devices that can initiate the read or write transaction.

  E.g CPU is always a bus master.

  The bus can have many masters when there are multiple CPU’s and when the Input devices can initiate bus transaction.

- **Mention the advantage of using bus master?**

  It offers higher bandwidth by using packets, as opposed to holding the bus for full transaction.

- **What is spilt transaction?**

  The idea behind this is to split the bus into request and replies, so that the bus can be used in the time between request and the reply.

- **Explain the different technique to reduce cache miss penalty?**

  Multiple Level caches

  Critical word first and early restart

  Giving priority to read misses over writes

  Merging write buffers
  Victim caches

- **Explain the different technique to reduce miss rate?**

  Larger block size

  Larger caches

  Higher associativity

  Way prediction and pseudoassociative caches

  Compiler optimization

- **Discuss how main memory is organized to improve performance?**

  Wider main memory

  Simple interleaved memory

  Independent memory banks

- **Explain the various levels of RAID?**

No redundancy

Mirroring

Bit-interleaved parity

Block- interleaved parity

P+Q redundancy

- **Explain the various ways to measure I/O performance?**

Throughput versus response time

Little queuing theory