UNIT I

INSTRUCTION LEVEL PARALLELISM

1. What is ILP? What are the two major approaches for ILP?

   ILP - Instruction level parallelism – All processors use pipelining to overlap the execution of instructions and improve performance. This potential overlap among overlap is called ILP, since multiple operations (or instructions) can be executed in parallel.

   Two approaches to exploit ILP are
   - ✓ Dynamic hardware intensive approach
   - ✓ Static compiler intensive approach

2. What are the needs of ILP
   - ✓ Sufficient resources
   - ✓ Parallel scheduling
     - ➢ Hardware solution
     - ➢ Software solution
   - ✓ Application should contain ILP

3. What are the various hazards and define?

   Three types of hazards
   - ✓ Structural hazards – arise from resource conflicts when the hardware cannot support all possible combinations of instructions simultaneously in overlapped execution.
   - ✓ Data hazards – arise when an instruction depends on the results of a previous instruction in that is exposed by the overlapping of instructions in the pipeline.
   - ✓ Control hazards – arise from the pipelining of branches and other instructions that change the PC

4. What is dynamic scheduling?
   - ✓ Dynamic scheduling - hardware rearranges instruction execution to reduce stalls while maintaining data flow and exception behavior.
   - ✓ Allow instructions behind stall to proceed

5. What are the advantages of dynamic scheduling
   - ✓ Handles cases when dependences unknown at compile time
     E.g.: because they may involve a memory reference
   - ✓ It simplifies the compiler
   - ✓ It allows the processor to tolerate unpredictable delays such as cache misses, by executing other code while waiting for the miss to resolve.
   - ✓ Allows code compiled for one machine to run efficiently on a different machine, with different number of function units (FUs), and different pipelining
   - ✓ Use speculation technique to improve the performance.

6. What is meant by reservation station?

Tomasulo's algorithm implements register renaming through the use of **reservation stations.** Reservation stations are buffers which fetch and store instruction operands as soon as they are available.

7. What is common data bus?

   Results are passed directly to functional units from the reservation stations where they are buffered, rather than going through the registers. This bypassing is done with a common result bus that allows all units waiting for an operand to be loaded simultaneously (on the 360/91 this is called the common data bus, or CDB).

8. What is speculation?

   Hardware-based speculation follows the predicted flow of data values to choose when to execute instructions. This method of executing programs is essentially a data flow execution: Operations execute as soon as their operands are available.

9. What is instruction commit?

   When an instruction is no longer speculative, we allow it to update the register file or memory; we call this additional step in the instruction execution sequence instruction commit.

10. What is reorder buffer?

    Instruction execution sequence requires an additional set of hardware buffers that hold the results of instructions that have finished execution but have not committed. This hardware buffer called reorder buffer.

11. What is register renaming?

    Name dependencies is not a true dependence, instructions involved in name dependence can execute simultaneously or be reordered, if the name (register number or memory location) used in the instruction is changed so the instruction do not conflict. This renaming can be more easily done for register operands called register renaming. This register renaming can be done either statically by compiler or dynamically by hardware.

12. What are the four steps involved in instruction execution.

    - ✓ Issue
    - ✓ Execute
    - ✓ Write result
    - ✓ Commit

13. What is principle of locality

    An implication of locality is that we can predict with reasonable accuracy what instructions and data a program will use in the near future based on its accesses in the recent past. The principle of locality also applies to data accesses, though not as strongly as to code accesses.

14. What are the types of locality?

    Two different types of locality have been observed. Temporal locality states that recently accessed items are likely to be accessed in the near future. Spatial locality says that items whose addresses are near one another tend to be referenced close together in time.

15. How to calculate the value of CPI.

The value of the CPI (cycles per instruction) for a pipelined processor is the sum of the base CPI and all contributions from stalls:

Pipeline CPI = Ideal pipeline CPI + Structural stalls + Data hazard stalls + Control stalls

16. What is ideal pipeline?

The ideal pipeline CPI is a measure of the maximum performance attainable by the implementation. By reducing each of the terms of the right-hand side, we minimize the overall pipeline CPI or, alternatively, increase the IPC (instructions per clock).

17. What is loop level Parallelism

The simplest and most common way to increase the ILP is to exploit parallelism among iterations of a loop. This type of parallelism is often called loop-level Parallelism.

E.g.: for (i=1; i<1000; i++)

x[i] = x[i] + y[i];

Every iteration of loop can overlap with any other iteration.

18. What are the types of data dependencies? What are the various data hazards?

There are three different types of dependences: data dependences (also called true data dependences), name dependences, and control dependences.

- ✓ RAW - Read After Write
- ✓ WAR - Write After Read
- ✓ WAW - Write After Write

19. What is RAW

j tries to read a source before i write it, so j incorrectly gets the old value. This hazard is the most common type and corresponds to true data dependence, program order must be preserved to ensure that j receives the value from i.

20. What is WAW?

WAW (Write After Write)—j tries to write an operand before it is written by i. The writes end up being performed in the wrong order, leaving the value written by i rather than the value written by j in the destination. This hazard corresponds to output dependence. WAW hazards are present only in pipelines that write in more than one pipe stage or allow an instruction to proceed even when a previous instruction is stalled.

21. What is WAR?

WAR (Write After Read)—j tries to write a destination before it is read by *i,* so *i* incorrectly gets the *new* value. This hazard arises from antidependence. WAR hazards cannot occur in most static issue pipelines—even deeper pipelines or floating-point pipelines—because all reads are early (in ID) and all writes are late (in WB).

22. What is control dependence?

A control dependence determines the ordering of an instruction, i, with respect to a branch instruction so that the instruction i is executed in correct program order and only when it should be. Every instruction, except for those in the first basic block of the program, is control dependent on some set of branches, and, in general, these control dependences must be preserved to preserve program order.

23. Write one of the simplest example for control dependence

```
if p1 {
S1;
```

```
};
if p2 {
S2;
}
```

24. What is loop unrolling? What are the advantages of loop unrolling?

Loop unrolling is a simple but useful method for increasing the size of straight-line code fragments that can be schedule effectively. To control the various dependencies the loop is unrolled as many times as possible.

Advantage:

- ✓ Unrolling improves the performance of the loop by eliminating overhead instructions.
- ✓ Loop unrolling can also be used to improve scheduling. Because it eliminates the branch, it allows instructions from different iterations to be scheduled together.

23. List the limitation of loop unrolling?

There are three different types of limits to the gains that can be achieved by loop unrolling.

- ✓ A decreases in the amount of overhead with each unroll
- ✓ Code size limitation
- ✓ Compiler limitation

24. What are Branch prediction buffer / Buffer history table?

The simplest dynamic branch-prediction scheme is a branch prediction buffer or branch history table. The branch prediction buffer is a small memory indexed by the lower portion of the address of the branch instruction. The memory contains a bit that says whether the branch was recently taken or not.

25. What is an imprecise exception?

An exception is imprecise if the processor state when an exception is raised does not look exactly as if the instructions were executed sequentially in strict program order.

Imprecise exceptions can occur because of two possibilities:

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

26. Differentiate between static and dynamic branch prediction approaches?

Static Branch Prediction	Dynamic Branch Prediction
Static branch predictors are sometimes used in processors where the expectation is that branch behavior is highly predictable at compile time.	Dynamic predictors are sometimes used in processors where the expectation at run branch behavior is highly predictable at run time.
Static prediction can also be used to assist dynamic predictors and able to accurately predict a branch at compile time is also helpful for scheduling data hazards. E.g.: Loop unrolling	The simplest dynamic branch prediction scheme is a branch prediction buffer or branch history table.

UNIT II

MULTIPLE ISSUE PROCESSORS

1. What is VLIW? List out the advantage of VLIW?

   VLIW (very long instruction word): Issue a fixed number of instructions formatted as: One instruction comprising independent MIPS instructions or a fixed instruction packet with the parallelism among instructions explicitly indicated by instruction, also known as EPIC - explicitly parallel instruction computers. A VLIW uses multiple, independent functional units. A VLIW packages multiple independent operations into one very long instruction.

   Advantage:

   - ✓ Simple hardware - Number of functional units can be increased without needing additional sophisticated hardware to detect parallelism like in superscalar.
   - ✓ Good compilers can detect parallelism based on global analysis of the whole program.

2. Define EPIC?

   EPIC is Explicit Parallel Instruction Computing. It is an architecture framework proposed by HP. It is based on VLIW and was designed to overcome the key limitations of VLIW while simultaneously giving more flexibility to compiler writers.

3. What is Loop Level Analysis?

   Determine what dependences exist among the operands in a loop across the iterations of that loop and Determine whether data accesses in later iterations are dependent on data values produced in earlier iterations.

4. What is loop carried dependence?

   Data dependence between different loop iterations (data produced in earlier iterations used in a later one) is called loop carried dependence.

5. What are the tasks in finding the dependence in a program?

   There are three tasks.

   - ✓ Have good scheduling of code
   - ✓ Determine which loop might contain parallelism
   - ✓ Eliminate name dependence

6. Define dependence analysis algorithm?

   Dependence analysis algorithm is algorithm used to detect the dependence by the compiler based on the assumptions that

   - ✓ Array indices are affine
   - ✓ There exists GCD of the two affine indices

7. What is copy propagation?

   Copy propagation is the algebraic simplifications of expressions and an optimization which eliminates operation that copy values.

8. What is tree-height reduction technique?

   Tree-height reduction is optimization which reduces the height of the tree structure representing a computation, making it wider but shorter.

9. What are the components of software pipeline loop?

A software pipeline loop consists of a loop body, start- up code and clean-up code.

- ✓ Start up code is to execute code left out from the first original loop iterations.
- ✓ Finish code to execute instructions from the last original iterations.

10. What is trace scheduling?

Trace scheduling is way to organize the process of global code motion it simplifies instruction scheduling by incurring the cost of possible code motion on the less critical paths.

11. List out steps used for trace scheduling?

- ✓ Trace selection
- ✓ Trace compaction

12. Define Inter-procedural analysis?

A procedure with pointer parameters and if we want to analyze the procedure across the boundaries of the particular procedure. It is called inter-procedural analysis.

13. What is software pipelining?

It is a technique for reorganizing loop such that each iteration in the code is made from instructions chosen from different iterations of original loop.

14. Define critical path?

Critical path is defined as the longest sequence of dependent instructions in a program.

15. Define IA-64 processor?

The IA-64 is a RISC-Style, register-register instruction set with the features designed to support compiler based exploitation of ILP.

16. What is CFM and what is its use?

- ✓ CFM stands for Current Frame Pointer
- ✓ CFM pointer points to the set of registers to be used by a given procedure.

17. What are the parts of CFM pointer?

There are two parts. They are

- ✓ Local area – Used for local storage
- ✓ Output area - Used to pass values to any called procedure.

18. What is Itanium processor?

Itanium processor is an implementation of Intel IA-64 processor. It is capable of having six issues per clock cycle. The six issues include three branches and two memory reference.

19. What are the parts of 10 stage pipeline in Itanium processor?

- ✓ Front end
- ✓ Instruction delivery(EXP, REN)
- ✓ Operand delivery(WLD, REG)
- ✓ Execution(EXE, DEG, WRB)

20. What are the limitations of ILP?

- ✓ Limitations on hardware model
- ✓ Limitations on window size and maximum issue count
- ✓ Effect of finite register
- ✓ Effects of imperfect alias analysis

21. List the two techniques for eliminating dependent computations?

- ✓ Software pipelining
- ✓ Trace scheduling

22. Define Trace selection and Trace compaction?

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

Trace Compaction - Trace compaction tries to squeeze the trace into a small number of wide instructions. Trace compaction is code scheduling hence it attempts to move operations as early as it can in a sequence packing the operations into as few wide instructions as possible.

23. Define Superblocks.

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

24. Use of conditional or predicted instructions?

Conditional or predicted instructions are used to eliminate braches, converting control dependencies and potentially improving performance.

25. Define Instruction Group?

Instruction group is a sequence of consecutive instructions with no register data dependencies among them. All the instructions in a group could be executed in parallel if sufficient hardware resources existed and if any dependence through memory were preserved.

26. Use of template field in bundle?

The 5 bit template field within each bundle describes both the presence of any stops associated with the bundle and the execution unit type required by each instruction within the bundle.

27. List the two types of speculation supported by IA 64 processor?

- ✓ Control Speculation
- ✓ Memory reference speculation

28. Define Advance loads?

Memory reference support in the IA 64 uses a concept called advanced loads. Advance load is a load that has been speculatively moved above store instructions on which it is potentially dependent. To speculatively perform a load the ld.a instruction is used.

29. Define ALAT?

Executing advance load instructions created an entry in a special table called ALAT. It stores both the register destination of the load and the address of the accessed memory location. When a store is executed, an associative look up against the active ALAT entries is performed. If there is an ALAT entry with the same memory address as the store, mark the ALAT entry as invalid.

30. What are the functional units in Itanium Processor?

There are nine functional units in the Itanium processor.

- ✓ Two I units
- ✓ Two M units
- ✓ Three B units
- ✓ Two F units

All the functional units are pipelined.

31. Define Scoreboard?

In Itanium processor 10 stage pipelines divided into 4 parts. In operand delivery part scoreboard is used to detect when individual instruction can proceed so that a start of one instruction in a bundle need not cause the entire bundle to stall.

32. Define Book Keeping Code?

Basic block consists of one entry and one exit code. This code is known as Book Keeping Code.

33. What is the major difference between superscalar and VLIW processors?

Superscalar processors issue more than one instruction at a time; they can be statically scheduled by the compiler or dynamically, in hardware, based on techniques like Tomasulo and score boarding;

VLIW processor, in contrast, issue a fixed number of instructions formatted either as one large instruction or as a fixed instruction packet; they are inherently statically scheduled by the compiler.

34. Define sentinel and what is the use of it?

The original location of the speculative instruction is marked by a sentinel, which tells hardware that the earlier speculative instruction is no longer speculative and vales may be committed.

35. What are multiple issue processors? What are the types of multiple issues processor?

Multiple-Issue Processors

- ✓ Superscalar: varying no. instructions/cycle (0 to 8), scheduled by HW (dynamic issue capability)
- ✓ IBM PowerPC, Sun UltraSparc, DEC Alpha, Pentium III/4, etc. VLIW (very long instr. word): fixed number of instructions (4-16) scheduled by the compiler (static issue capability)
- ✓ Intel Architecture-64 (IA-64, Itanium), TriMedia, TI C6x

The goal of the multiple-issue processors is to allow multiple instructions to issue in a clock cycle.

- ✓ Statically scheduled superscalar processors,
- ✓ VLIW (very long instruction word) processors,
- ✓ Dynamically scheduled superscalar processors.

36. How does superscalar processor vary from VLIW processor?

The two types of superscalar processors issue varying numbers of instructions per clock and use in-order execution if they are statically scheduled or out-of order execution if they are dynamically scheduled. VLIW processors, in contrast, issue a fixed number of instructions formatted either as one large instruction or as a fixed instruction packet with the parallelism among instructions explicitly indicated by the instruction. VLIW processors are inherently statically scheduled by the compiler.

37. What is the limitation of VLIW processors?

- ✓ Very smart compiler needed (but largely solved!)
- ✓ Loop unrolling increases code size
- ✓ Unfilled slots waste bits

- ✓ Cache miss stalls whole pipeline

38. How is EPIC a better alternative?
    - ✓ Superscalar: expensive but binary compatible
    - ✓ VLIW: simple, but not compatible

39. Write down the capabilities of a compiler to speculate?
    - ✓ The ability of the compiler to find instructions that, with the possible use of register renaming, can be speculatively moved and not affect the program data flow
    - ✓ The ability to ignore exceptions in speculated instructions, until we know that such exceptions should really occur, and
    - ✓ The ability to speculatively interchange loads and stores, or stores and stores, which may have address conflicts.

40. Define Poisson bits and write its use?

    A set of status bits, called poison bits, are attached to the result registers written by speculated instructions when the instructions cause exceptions.

    The poison bits cause a fault when a normal instruction attempts to use the register.

41. Write about Itanium Instruction format

    Instructions grouped in 128-bit bundles
    - ✓ 3 * 41-bit instruction
    - ✓ 5 template bits, indicate type and stop location

42. What is superscalar processor?

    Superscalar: multiple instructions issued per cycle
    - ✓ Statically scheduled
    - ✓ Dynamically scheduled

43. How to construct a superblock?

    To construct a superblock use tail duplication to create a separate block that corresponds to the portion of the trace after the entry. Each unrolling of the loop would create an exit from the superblock to a residual loop that handles the remaining iterations.

44. What are super pipelined processors

    Anticipated success of multiple instructions led to Instructions Per Cycle (IPC) metric instead of CPI

45. What are the advanced complier support techniques?
    - ✓ Loop-level parallelism
    - ✓ Software pipelining
    - ✓ Global scheduling (across basic blocks)

46. What is software pipelining?

    Software pipelining is a related technique that that consumes less code space. It interleaves instructions from different iterations.

47. What is global code scheduling?

Loop unrolling and software pipelining work well when there are no control statements (if statements) in the loop body i.e., if the loop is a single basic block. So if there are control statements then Global code scheduling is implemented scheduling/moving code across branches: larger scheduling scope.

UNIT III

MULTIPROCESSORS AND THREAD LEVEL PARALLELISM

1. Define cache coherence problem.
   Cache coherence problem describes how two different processors can have two different values for the memory location.
2. What is write serialization?
   Serializing the writes ensures that every processor will see the write done the case that some processor could see the write of P2 first and then see the write of P1, maintaining the value written by P1 indefinitely. The simplest way to avoid such difficulties is to ensure that all writes to the same location are seen in the same order; this property is called *write serialization.*
3. What is snoop cache and write through cache?
   Every cache that has a copy of the data from a block of physical memory also has a copy of the sharing status of the block, but no centralized state is kept. The caches are all accessible via some broadcast medium (a bus or switch), and all cache controllers monitor or *Snoop* on the medium to determine whether or not they have a copy of a block that is requested on a bus or switch access. We focus on this approach in this section.
4. What is symmetric shared memory?
   Symmetric shared-memory machines usually support the caching of both shared and private data.
5. What is private data and shared data?
   Private data are used by a single processor, while *shared data* are used by multiple processors; essentially providing communication among the processors through reads and writes of the shared data.
6. What happens when a private and shared item is cached?
   When a private item is cached, its location is migrated to the cache, reducing the average access time as well as the memory bandwidth required. Since no other processor uses the data, the program behavior is identical to that in a uniprocessors.
7. What are the two aspects of cache coherence problem?
   - ✓ Coherence- It determines what value can be returned by the particular read operation.
   - ✓ Consistency- It determines when the value may be returned by the read operation.
8. What is true sharing miss?
   The first source is the so-called true sharing misses that arise from the communication of data through the cache coherence mechanism. In an invalidation based protocol, the first write by a processor to a shared cache block causes an invalidation to establish ownership of that block. Additionally, when another processor attempts to read a modified word in that cache block, a miss occurs and the resultant block is transferred. Both these misses are classified as true sharing misses since they directly arise from the sharing of data among processors.
9. What is false sharing miss?
   False sharing arises from the use of invalidation based coherence algorithm with a single valid bit per cache block. False sharing occurs when a block is invalidated (and a subsequent

reference causes a miss) because some word in the block, other than the one being read, is written into. If, however, the word being written and the word read are different and the invalidation does not cause a new value to be communicated, but only causes an extra cache misses, then it is a false sharing miss. In a false sharing miss, the block is shared, but no word in the cache is actually shared, and the miss would not occur if the block size were a single word.

10. What are the advantages of having a distributed memory?

    Distributing the memory among the nodes has two major benefits. First, it is a cost-effective way to scale the memory bandwidth if most of the accesses are to the local memory in the node. Second, it reduces the latency for accesses to the local memory.

11. What is the disadvantage of having a distributed memory?

    The key disadvantages for distributed memory architecture are that communicating data between processors becomes somewhat more complex, and that it requires more effort in the software to take advantage of the increased memory bandwidth afforded by distributed memories.

12. What is TLP?

    This higher-level parallelism is called thread-level parallelism (TLP) because it is logically structured as separate threads of execution.

13. What are the two types of cache coherence protocol?
    - ✓ Directory based protocol.
    - ✓ Snooping protocol.

14. Define Directory based protocol.

    The shared portion of the main memory may be kept in one common place called directory. From this directory we can retrieve the data.

15. Name the different types of snooping protocol?
    - ✓ Invalidate protocol
    - ✓ Update / write broadcast protocol.

16. Difference between write Update and invalidate protocol?

    Write update:
    - ✓ Multiple write broadcast is present
    - ✓ Here they consider separate word for each cache block
    - ✓ Access time is less

    Invalidate:
    - ✓ Only one invalidation is present
    - ✓ Invalidation is performed for entire cache block
    - ✓ Access time is high

17. What are the different types of access in distributed shared memory architecture?

    Local - If the processor refers the local memory then it is called local access.

    Remote - If the processor refers the other process memory then it is called remote access

18. What are the disadvantages of remote access?
    - ✓ Compiler mechanism for cache coherence is very limited
    - ✓ Without the cache coherence property the multiprocessor system loss the advantage of fetch and use multiple words
    - ✓ Prefetch is very useful only when the multiprocessor fetch multiple word

19. What are the states available in directory based protocol?
    Shared - One or more processor can have the copies of same dat.
    Uncached - No processor has the copy of data block.
    Exclusive - Exactly one processor has the copy of data block.

20. What are the nodes available in distributed system?
    - ✓ Local Node
    - ✓ Home Node
    - ✓ Remote Node

21. Define Synchronization.
    Synchronization is the mechanism that is build with user level software routine, which depends on hardware supplied synchronization instruction.

22. Name the basic hardware primitives?
    - ✓ Atomic Exchange
    - ✓ Test and set
    - ✓ Fetch and Increment

23. Define spinlock.
    It is a lock that a processor continuously tries to acquire spinning around a loop until it succeeds. It is mainly used when the programmer wants to use the lock for a small period of time.

24. What is the mechanism to implement locks?
    There are two methods to implement the locks.
    - ✓ Implementing lock without using cache coherence
    - ✓ Implementing lock using cache coherence.

25. What are the advantages of using spin lock?
    There are two advantages of using spin lock
    - ✓ They have low overhead
    - ✓ Performance is high

26. What are the two primitives used for implementing synchronization?
    - ✓ Lock Based Implementation
    - ✓ Barrier based Implementation

27. Define sequential consistency.
    It requires that the result of any execution be the same as, if the memory access executed by each processor where kept in order and accesses among different processor are interleaved. It reduces the amount of incorrect execution

28. Define multithreading.
    The process of executing the multiple thread by common memory or common processor in which the execution is done is overlapping fashion.

29. What are the types of multi threading?
    - ✓ Fine grained multithreading - It has the ability to switch threads for each instruction.
    - ✓ Coarse grained multithreading - It has the ability to switch the threads only for costly stalls.

UNIT IV

MEMORY AND I/O

1. Define cache.
   Cache is the name given to the first level of the memory hierarchy encountered once the address leaves the CPU. E.g.: file caches, name caches.
2. Define the term cache miss and cache hit.
   When the CPU finds a request data item in the cache, it is called cache hit. When the CPU does not finds the data item it needs in the cache, it is called cache miss
3. How to evaluate Cache Performance?
   The average memory access time is calculated as follows
   Average memory access time = hit time + Miss rate x Miss Penalty.
   Where Hit Time is the time to deliver a block in the cache to the processor (includes time to determine whether the block is in the cache).
4. Explain Miss Rate and Miss Penalty?
   Miss Rate is the fraction of memory references not found in cache (misses/references) and Miss Penalty is the additional time required because of a miss.
5. What are the factors on which the cache miss depends on?
   The time required for the cache miss depends on both
   - ✓ Latency
   - ✓ Bandwidth
6. What is the principle of locality? Define its types?
   Program access a relatively small portion of the address space at any instant of time is called principle of locality. There are two types of locality. They are
   - ✓ Temporal locality (Locality in time)
   - ✓ Spatial locality (Locality in space)

   Temporal locality - It's an accessed item has a high probability being accessed in the near future
   Spatial locality - These are items close in space to a recently accessed item have a high probability of being accessed next
7. What is called pages and segments?
   The address space is usually broken into fixed-size blocks, called pages. Each page resides either in main memory or on disk. The variable-size blocks are called segments.
8. What is called memory stall cycles?
   The number of cycles during which the CPU is stalled waiting for a memory access is called memory stall cycles.
9. Write down the formula for calculating average memory access time?
   Average memory access time=Hit time Miss Rate*Miss penalty. When hit time is the time to hit in the cache, the formula can help us decide between split caches and a unified cache.
10. What is sequence recorded?
    The sequence recorded on the magnetic medics is a sector number, a gap, the information for that sector including error correction code, a gap, and the sector number of the next sector and so on.

11. What is termed as cylinder?

The term cylinder is used to refer to all the tracks under the arms at a given point on all surfaces.

12. List the components to a disk access.

There are three mechanical components to a disk access:

- ✓ Rotation latency
- ✓ Transfer time
- ✓ Seek time

13. What is average seek time?

Average seek time is the sum of the time for all possible seeks divided by the number of possible seek. Average seek times are advertised to be 5 ms to 12 ms.

14. What is transfer time?

Transfer time is the time it takes to transfer a block of bits, typically a sector, under the read-write head. This time is a function of the block size, disk size, rotation speed, recording density of the track, and speed of the electronics connecting the disk to computer.

15. Write the formula to calculate the CPU execution time.

CPU execution time = (CPU clock cycles+ memory stall cycles)*clock cycle time.

16. Write the formula to calculate the CPU time.

CPU time= (CPU execution clock cycles+ memory stall clock cycles)* clock cycle time.

17. Define miss penalty for an out of order execution processor.

For an out of order execution processor, miss penalty is defined as follows.

(Memory stalls cycles/Instruction) *(misses/instruction) *(total Miss Latency overlapped miss latency).

18. What are the techniques available to reduce cache penalty or miss rate via parallelism?

The three techniques that overlap the execution of instructions are

- ✓ Non blocking caches to reduce stalls on cache miss- to match the out of order processors
- ✓ Hardware Prefetching of instructions and data
- ✓ Compiler- controlled Prefetching.

19. List the advantage of memory hierarchy?

Memory hierarchy takes advantage of

- ✓ Locality
- ✓ Cost/performance of memory technologies

20. What is the goal of memory hierarchy?

The goal is to provide a memory system with

- ✓ Cost almost as low as the cheapest level of memory
- ✓ Speed almost as fast as the faster level

21. Define hit rate and hit time?

When the CPU finds a requests data item in the cache, it is called a cache hit.

- ✓ Hit Rate: the fraction of cache access found in the cache
- ✓ Hit Time: time to access the upper level which consists of RAM access time + Time to determine hit\miss

22. Define miss rate and miss penalty?

When the CPU does not find a data item it needs in the cache, a cache miss occurs

- ✓ Miss Rate - the fraction of cache access not found in the cache
- ✓ Miss penalty-Time to replace a block in cache + time to deliver the block to the processor

23. What does Latency and Bandwidth determine?

Latency - Determine the time to retrieve the first word of the block.
Bandwidth - Determine the time to retrieve the rest of this block.

24. How does page fault occur?

When the CPU references an item within a page that is not present in the cache or main memory, a page fault occurs, and the entire page is moved from the disk to main memory.

25. What is called the miss penalty?

The number of memory stall cycles depends on both the number of misses and the cost per miss, which is called the miss penalty.

26. What is Average memory access time?

The average memory access time for processors is the better measure of memory hierarchy performance with in-order execution

27. What are the categories of cache miss(3cs of cache miss)

- ✓ Compulsory
- ✓ Capacity
- ✓ Conflict

28. What is Local Miss Rate?

This rate is simply the number of misses in a cache divided by the total number of memory accesses to this cache. As you would expect, for the first-level cache it is equal to Miss Rate L1 and for the second-level cache it is Miss Rate L2.

29. What is Global Miss Rate?

The number of misses in the cache is divided by the total number of memory accesses generated by the CPU. Using the terms above, the global miss rate for the first-level cache is still just Miss Rate L1 but for the second-level cache it is Miss Rate L1 x Miss RateL2.

30. What is Write through?

Write through caches rely on write buffers, as all stores must be sent to the next lower level the hierarchy.

31. What is Write Back?

Write back caches use a simple buffer when a block is replaced. If the write buffer is empty, the data and the full address are written in the buffer, and the write is finished from the CPU's perspective.

32. What is Critical Word First and early restart?

Request the missed word first from memory and send it to the CPU as soon as it arrives; let the CPU continue execution while filling the rest of the words in the block. Critical-word-first fetch is also called wrapped fetch and requested word first.

Fetch the words in normal order, but as soon as the requested word of the block arrives, send it to the CPU and let the CPU continue execution.

33. What is Write Merging?

The buffer contains other modified blocks; the addresses can be checked to see if the address of this new data matches the address of the valid write buffer entry. If so, the new data are combined with that entry is called write merging.

34. What is virtual memory?

Virtual memory divides physical memory into blocks (called page or segment) and allocates them to different processes. With virtual memory, the CPU produces virtual addresses that are translated by a combination of HW and SW to physical addresses, which accesses main memory.

35. What is meant by Split Transactions?

With multiple masters, a bus can offer higher bandwidth by using packets, as opposed to holding the bus for the full transaction. This technique is called split transaction.

36. What is a bus master?

The devices can initiate a read or write transaction; the CPU, for instance, is always a bus master. A bus has multiple masters when there are multiple CPUs or when 110 devices can initiate a bus transaction.

37. What are Transient faults and hard faults?

Transient faults are faults that come and go, at least temporarily fixing themselves. Hard faults stop the device from working properly, and will continue to misbehave until repaired.

38. What is meant by MTTF and MTTR?

If a single disk fails, the lost information can be reconstructed from redundant information. Mean time to failure (MTTF). Another disk fail between the time the first disk fails and the time it is replaced (termed mean time to repair, or MTTR).

39. Define Reliability?

Module reliability is a measure of the continuous service accomplishment (or equivalently. of the time to failure) from a reference initial instant. Hence, the mean time to failure (MTTF) is a reliability measure. The reciprocal of MTTF is a rate of failures, generally reported as failures per billion hours of operation, or FIT (for failures in time). Mean 'lime between failures (MTBF) is simply the sum of MTTF + MTTR.

Although MTBF is widely used, MTTF is often the more appropriate term. If a collection of modules have exponentially distributed lifetimes-meaning that the age of a module is not important in probability of failure-the overall failure rate of the collection is the sum of the failure rates.

40. Define Module Availability

Module availability is a measure of the service accomplishment with respect to the alternation between the two states of accomplishment and interruption. For non-redundant systems with repair, module availability is

Module availability = MTTF / (MTTF + MTTR).

41. List the two states of Service level Agreement

Two states of service with respect to an SLA.

Service Accomplishment - Where the service is delivered as specified.

Service Interruption - Where the delivered service is different from the SLA.

42. What is a victim cache?

The write victim buffer or victim buffer contains the dirty blocks that are discarded from a cache because of a miss. Rather than stall on a subsequent cache miss, the contents of the buffer are checked on a miss to see if they have the desired data before going to the next lower-level memory called victim cache.

UNIT V

MULTI-CORE ARCHITECTURES

1. What are multiprocessors? Mention the categories of multiprocessors?
   Multiprocessor is used to increase performance and improve availability. The different categories are SISD, SIMD, and MIMD.

2. What is multitasking? Explain about Multithreading?
   Multitasking is the execution of two or more tasks at one time by using content switching (functionality). Multithreading is a process wherein multiple threads to share the functional units of one processor via overlapping Multithreading computers have hardware support to efficiently execute multiple threads. These are distinguished from multiprocessing systems (such as multi-core systems) in that the threads have to share the resources of a single core: the computing units, the CPU caches and the translation lookaside buffer (TLB). Where multiprocessing systems include multiple complete processing units, multithreading aims to increase utilization of a single core by leveraging thread- level as well as instruction- level parallelism.

3. Write the advantages of Multithreading.
   If a thread gets a lot of cache misses, the other thread(s) can continue, taking advantage of the unused computing resources, which thus can lead to faster overall execution, as these resources would have been idle if only a single thread was executed. If a thread cannot use all the computing resources of the CPU (because instructions depend on each other's result), running another thread permits to not leave these idle. If several threads work on the same set of data, they can actually share their cache, leading to better cache usage or synchronization on its values.

4. Write the disadvantages of Multithreading.
   Multiple threads can interfere with each other when sharing hardware resources such as caches or translation lookaside buffers (TLBs). Execution times of a single-thread are not improved but can be degraded, even when only one thread is executing. This is due to slower frequencies and/or additional pipeline stages that arc necessary to accommodate thread-switching hardware. Hardware support for Multithreading is more visible to software, thus requiring more changes to both application programs and operating systems than Multi processing.

5. Define Software Multithreading
   Software multithreading is a piece of software that is aware of more than one core / processor and can use these to be able to simultaneously complete multiple tasks.

6. Define Hardware Multithreading
   Hardware multithreading is a multithreading that allows multiple to share the functional units of a single processor in an overlapping fashion.

7. List some advantages of Software Multithreading.
   - ✓ Increased responsiveness and worker productivity.
     - Increased application responsiveness when different tasks run in parallel.
   - ✓ Improved performance in parallel environments.
     -When running computations on multiple processors.
   - ✓ More computations per cubic foot of data center.
     -Web based applications are often multi-threaded in nature.

8. What is CMT?

Chip multiprocessors - also called multi-core microprocessors or CMPs for short - are now the only way to build high-performance microprocessors, for a variety of reasons. Large uniprocessors are no longer scaling in performance, because it is only possible to extract a limited amount of parallelism from a typical instruction stream using conventional superscalar instruction issue techniques. In addition, one cannot simply ratchet up the clock speed on today's processors, or the power dissipation will become prohibitive in all but water- cooled systems.

9. What is SMT?

Simultaneous multithreading, often abbreviated as SMT, is a technique for improving the overall efficiency of superscalar CPUs with hardware multithreading. SMT permits multiple independent threads of execution to better utilize the resources provided by modern processor architectures.

10. What are the Disadvantages of SMT?

Simultaneous multithreading cannot improve performance if any of the shared resources are limiting bottlenecks for the performance. In fact, some applications run slower when simultaneous multithreading are enabled. Critics argue that it is a considerable burden to put on software developers that they have to test whether simultaneous multithreading is good or bad for their application in various situations and insert extra logic to turn it off if it decreases performance.

11. What is a Heterogeneous Multi-core processor?

Heterogeneous Multi-core processor is a processor in which multiple cores of different types are implemented in one CPU.

12. Write the advantages of heterogeneous multi core architectures?

- ✓ Efficient adaptation to application diversity
- ✓ Efficient use of die area for a given thread parallelism

13. Explain about IBM Cell Processor architecture.

The Cell chip can have a number of different configurations; the basic configuration is a multi-core chip composed of one "Power Processor Element" ("PPE") (sometimes called "Processing Element", or "PE") and multiple “Synergistic Processing Elements” (SPE).

14. List the components of IBM cell architecture

- ✓ Power Processing Elements (PPE).
- ✓ Synergistic Processor Elements (SPE).
- ✓ I/O controller.
- ✓ Element Interconnect Bus (EIB).

15. What are the components of PPE?

The PPE is made out of two main units.

- ✓ Power Processor Unit(PPU)
- ✓ Power Processor Storage Subsystem(PPSS)

16. What is Memory Flow Controller (MFC)?

The Memory Flow Controller is actually the interface between the Synergistic Processor (SPU) and the rest of the cell chip. Actually, the MFC interfaces the SPU with the EIB.

17. What is multicore'?

At its simplest, multi-core is a design in which a single physical processor contains the core logic of more than one processor. It's as if an Intel Xeon processor were opened up and inside were packaged all the circuitry and logic for two (or more) Intel Xeon processors. The multi-core design takes several such processor "cores" and packages them as a single physical processor. The goal of this design is to enable a system to run more tasks simultaneously and thereby achieve greater overall system performance.

18. Write the software implications of a multicore processor?

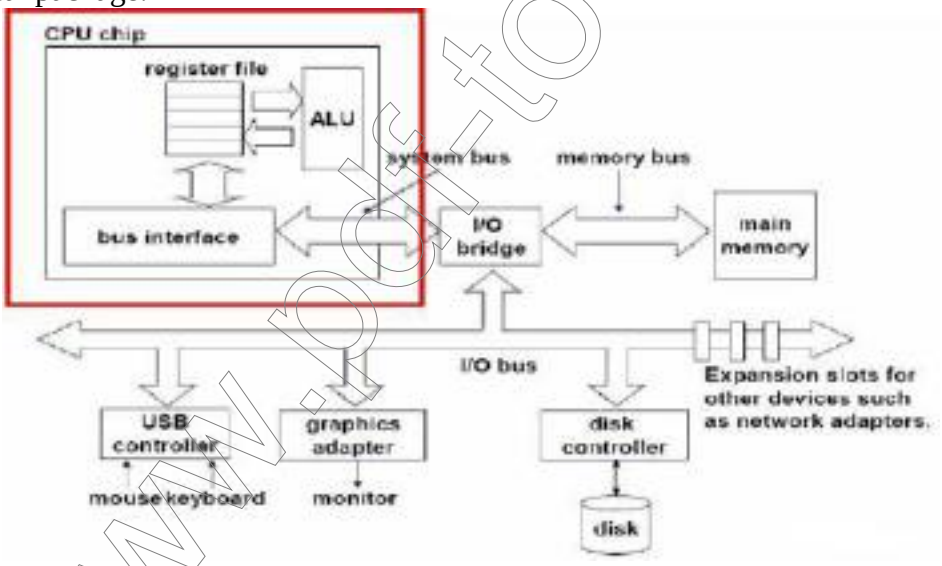
Multi-core systems will deliver benefits to all software, but especially multi-threaded programs. All code that supports HT Technology or multiple processors, for example, will benefit automatically from multi-core processors, without need for modification. Most server-side enterprise packages and many desktop productivity tools fall into this category

19. What is fine grained multithreading?

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

20. Draw a diagram for multicore processors.

Multi-core processors, as the name implies, contain two or more distinct cores in the same physical package.



21. What is coarse grained multithreading?

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

22. What is a cell processor?

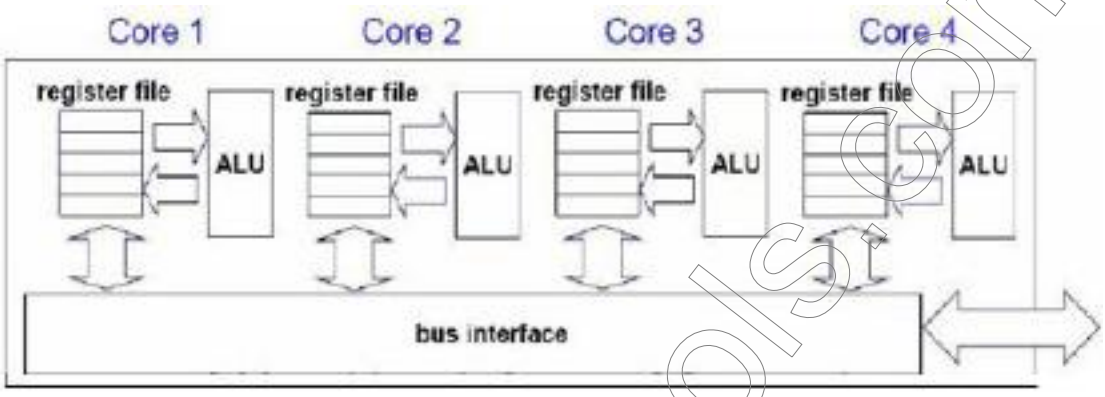
Cell is a heterogeneous chip multiprocessor that consists of an IBM 64-bit Power core, augmented with eight specialized co-processors based on a novel single-instruction multiple-data (SIMD) architecture called Synergistic Processor Unit (SPU), which is for data-intensive processing, like that found in cryptography, media and scientific applications. The system is integrated by a coherent on-chip bus.

23. What is the function of Power Processing Unit?

- ✓ A full set of 64-bit power pc register.
- ✓ 32-168 bit vector multimedia register.
- ✓ 32 KB LI data cache.
- ✓ 32 KB LI instruction cache.

24. List out the disadvantages of Heterogeneous multi-core processors?
    - ✓ Developer productivity.
    - ✓ Portability.
    - ✓ Manage ability.

25. Draw Chip Multiprocessor Architecture diagram?



26. What is hyper threading and HT Technology?

    Intel version of simultaneous multithreading (SMT).It makes single physical processor appear as multiple logical processors. Operating system can schedule to logical processors. Two single threads execute simultaneously on the same processor core - HT technology.

27. What are the various processor configurations?
    - ✓ A superscalar with no multithreading support
    - ✓ A superscalar with coarse-grained multithreading
    - ✓ A superscalar with fine-grained multithreading
    - ✓ A superscalar with simultaneous multithreading

28. What happens to a superscalar without multithreading support?

    In the superscalar without multithreading support, the use of issue slots is limited by a lack of ILP In addition, a major stall, such as an instruction cache miss, can leave the entire processor idle.

29. What are the design challenges in SMT?
    - ✓ Larger register file needed to hold multiple contexts.
    - ✓ Not affecting clock cycle time, especially in Instruction issue - more candidate instructions need to be considered.