St.Joseph College of Engineering

Dept. of ECE

CS6303 - COMPUTER ARCHITECTURE

UNIT-I

OVERVIEW & INSTRUCTIONS

1. What are the eight great ideas in computer architecture?

The eight great ideas in computer architecture are:

1. Design for Moore's Law
2. Use Abstraction to Simplify Design
3. Make the Common Case Fast
4. Performance via Parallelism
5. Performance via Pipelining
6. Performance via Prediction
7. Hierarchy of Memories
8. Dependability via Redundancy

2. What are the five classic components of a computer?

The five classic components of a computer are input, output, memory, datapath, and control, with the last two sometimes combined and called the processor.

3. Define ISA

The instruction set architecture, or simply architecture of a computer is the interface between the hardware and the lowest-level software. It includes anything programmers need to know how to make a binary machine language program work correctly, including instructions, I/devices, and son.

4. Define ABI

Typically, the operating system will encapsulate the details of doing I/O, allocating memory, and other low-level system functions so that application programmers do not need worry about such details. The combination of the basic instruction set and the operating system interface provided for application programmers is called the application binary interface (ABI).

5. What are the advantages of network computers?

Networked computers have several major advantages:

Communication: Information is exchanged between computers at high speeds.

Resource sharing: Rather than each computer having its own I/O dev ices, computers on the network can share I/O devices.

Nonlocal access: By connecting computers over long distances, users need

not be near the computer they are using.

6. Define Response Time

Response time is also called execution time. The total time required for the computer to complete a task, including disk accesses, memory accesses, I/activities, operating system overhead, CPU execution time, and son is called response time.

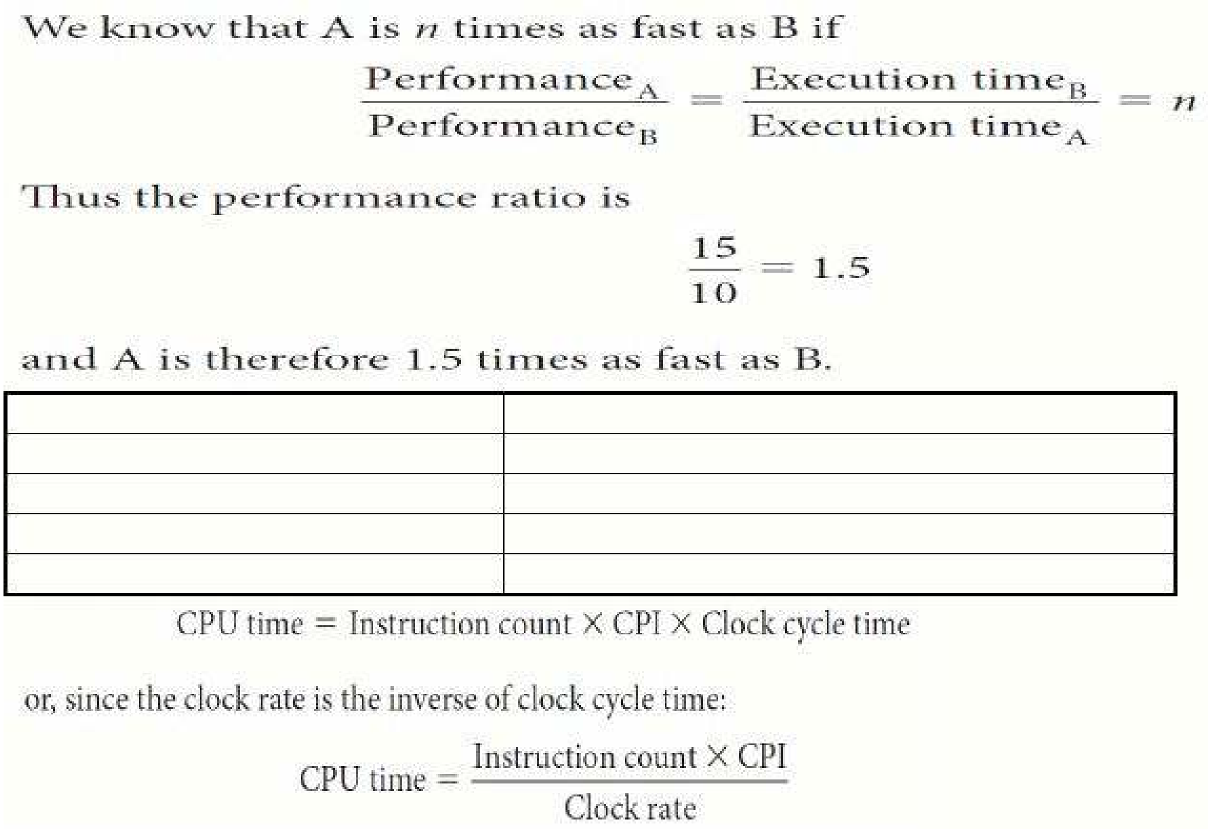
7. Define Throughput

Throughput or bandwidth is the total amount of wrk done in a given time.

8. Write the CPU performance equation.

The Classic CPU Performance Equation in terms of instruction count (the number of instructions executed by the program), CPI, and clock cycle time:

9. If computer A runs a program in 10 seconds, and computer B runs the same program in 15 seconds, how much faster is A over B.



We know that A is n times as fast as B if

$$\frac{\text{Performance}_A}{\text{Performance}_B} = \frac{\text{Execution time}_B}{\text{Execution time}_A} = n$$

Thus the performance ratio is

$$\frac{15}{10} = 1.5$$

and A is therefore 1.5 times as fast as B.

$$\text{CPU time} = \text{Instruction count} \times \text{CPI} \times \text{Clock cycle time}$$

or, since the clock rate is the inverse of clock cycle time:

$$\text{CPU time} = \frac{\text{Instruction count} \times \text{CPI}}{\text{Clock rate}}$$

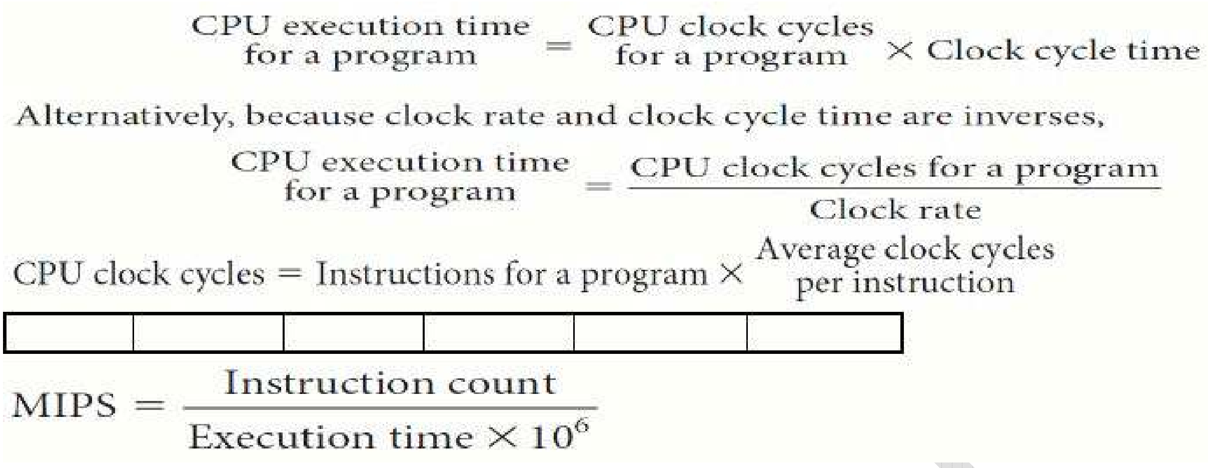
10. What are the basic components of performance?

The basic components of performance and how each is measured are:

Components of Performance	Units of measure
CPU execution time for a program	Seconds for the program
Instruction count	Instruction executed for the program
Clock cycles per instruction(CPI)	Average number of clock cycles per
instruction Clock cycle time `	Seconds per clock cycle

11. Define MIPS

Million Instructions Per Second (MIPS) is a measurement of program execution speed based on the number of millions of instructions. MIPS is computed as:



$$\text{CPU execution time for a program} = \text{CPU clock cycles for a program} \times \text{Clock cycle time}$$

Alternatively, because clock rate and clock cycle time are inverses,

$$\text{CPU execution time for a program} = \frac{\text{CPU clock cycles for a program}}{\text{Clock rate}}$$

$$\text{CPU clock cycles} = \text{Instructions for a program} \times \text{Average clock cycles per instruction}$$

$$\text{MIPS} = \frac{\text{Instruction count}}{\text{Execution time} \times 10^6}$$

12. Define Addressing Modes

The different ways in which the operands f an instruction are specified are called as addressing modes. The MIPS addressing modes are the following:

1. Immediate addressing 2. Register addressing 3. Base or displacement addressing 4. PC-relative addressing 5. Pseudo direct addressing

13. What is the difference between Machine Language and Assembly Language?

Machine language is the actual bits used to control the processor in the computer, usually viewed as a sequence of hexadecimal numbers (typically bytes). The processor reads these bits in from program memory, and the bits represent "instructions" as to what to do next. Thus machine language provides a way of entering instructions into a computer (whether through switches, punched tape, or a binary file). Assembly language is a more human readable view of machine language. Instead of representing the machine language as numbers, the instructions and registers are given names (typically abbreviated words, or mnemonics, e.g ld means "load"). Unlike a high level language, assembler is very close to the machine language. The main abstractions (apart from the mnemonics) are the use of labels instead of fixed memory addresses, and comments. An assembly language program (i.e. a text file) is translated to machine language by an assembler. A dis-assembler performs the reverse function (although the comments and the names of labels will have been discarded in the assembler process).

14. Why 2's complement method is used for representation of negative numbers?

The **two's complement** of a binary number is defined as the value obtained by subtracting the number from a large power of two (specifically, from 2 for an *N* -bit two's complement). A **two's-complement system** or **two's-complement arithmetic** is a system in which negative numbers are represented by the two's complement of the absolute value; this system is the most common method of representing signed integers on computers. In such a system, a number is negated (converted from positive to negative or vice versa) by computing its two's complement. An N-bit two's-complement numeral system can represent every integer in the range -2 t+2 -1. N-1 N-1The two's-complement system has the advantage of not requiring that the addition and subtraction circuitry examine the signs of the operands to determine whether to add or subtract. This property makes the system both simpler to implement and capable of easily handling higher precision arithmetic. Also, zero has only a single representation, obviating the subtleties associated with negative zero, which exists in ones'-complement systems.

15. Differentiate between Programmed I/O and I/O mapped I/O

Memory mapped I/is mapped in to the same address space as program memory and/or user memory, and is accessed in the same way. I/O mapped I /O (also known as port mapped I/O) uses a separate, dedicated address space and is accessed via a dedicated set of microprocessor instructions. If you're using a microprocessor or microcontroller that doesn't support port mapped I/O, then you have to use memory mapped I/O. Microprocessors that support port mapped I/include Intel x86 and compatible processors, and also the Zilog Z80 and Intel 8080. Microprocessors that DON'T support port mapped I/(and hence require the use of memory mapped I/O) include the Motorola 6800 and the MOS Technology 6502.

The advantage of port mapped I/is that it makes for neater code and requires fewer external components to implement I/O. However, it adds to the complexity and pin count of the microprocessor itself.

16. Differentiate between Assembler and Compiler.

Assembler - A computer program that takes computer instructions and converts them inta pattern of bits that the computer can understand and perform by it certain operations.

Compiler - This is a special program that processes statements written in a programming language and turns them in to machine language that a computer's processor uses.

17. What do you understand by Interleaved DMA?

A Stack is a type of data container/ data structure that implements the LAST-IN-FIRST-OUT (LIFO) strategy for inserting and recovering data. This is a very useful strategy, related t many types of natural programming tasks. For instance: Keeping track of nested invocation calls in a procedural programming language, Evaluating arithmetic expressions and eliminate the need for direct implementation of recursion.

18. How pipeline helps in a faster execution of an Instruction?

An **instruction pipeline** is a technique used in the design of computers and other digital electronic devices to increase their instruction throughput (the number of instructions that can be executed in a unit of time). The fundamental idea is to split the processing of a computer instruction inta series of independent steps, with storage at the end of each step. This allows the computer's control circuitry tissue instructions at the processing rate of the slowest step, which is much faster than the time needed to perform all steps at once. The term pipeline refers to the fact that each step is carrying data at once (like water), and each step is connected to the next (like the links of a pipe.)

19. List various methods of data transfer.

IDE and SCSI controllers can use any of three data transfer methods to move data to and from system memory. The first method, Programmed I/(PIO), relies entirely on the host PC's CPU to conduct data back and forth between the controller and memory.

Although PIO is cheap and easily implemented because it requires no special hardware, PIO-based disk I/heavily taxes the host CPU and makes it unsuitable for multitasking environments such as Windows NT, UNIX, and NetWare. All implementations of the

IDE/ATA specification can use PIO, whereas very few SCSI controllers (Adaptec is an exception) ever employ this method.

The other two methods of data transfer, which are more sophisticated than PIO, are known as *third-party DMA* and *first-party DMA*. Direct memory access (DMA) uses special hardware, either on the host system's motherboard or on a controller card, to facilitate the transfer of data to and from system memory without involving the CPU.

20. Differentiate between Push operation and POP operation in a stack.

A stack is a type of data structure -- a means of storing information in a computer. When a new object is entered in a stack, it is placed on top of all the previously entered objects.

The push instruction pushes a value on to the stack. The value is put ON TOP of the stack. The stack's size will increase by one. The pop instruction takes the TOP VALUE from the stack and assigns it to the specified variable. The stack's size will decrease by one. If there are n values on the stack (e.g. the stack's size is equal to zero), then the error flag will be set.

21. What is Multiprogramming?

Multiprogramming is a technique used to utilize maximum CPU time by running multiple programs simultaneously. The execution begins with the first program and continues till an instruction waiting for a peripheral is reached, the context of this program is stored, and the second program in memory is given a chance to run. The process continued until all programs finished running. Multiprogramming has no guarantee that a program will run in a timely manner. Usually on a mainframe - the computer has a number of programs loaded into memory and the operating system switches quickly between them, processing a little bit of each one in turn. The high speed of the processor makes it seem like more than one program is being run at the same time. On a PC it is usually called multitasking.

UNIT-II

ARITHMETIC OPERATIONS

1. Define Moore's Law

Moore's Law has provided so much more in resources that hard ware designers can now build much faster multiplication and division hardware. Whether the multiplicand is to be added or not is known at the beginning of the multiplication by looking at each of the 32 multiplier b its.

2. What are the floating point instructions in MIPS?

MIPS supports the IEEE 754 single precision and double precision formats with these instructions:

Floating-point addition

Floating-point subtraction

Floating-point multiplication

Floating-point division

Floating-point comparison

Floating-point branch

3. Define Guard and Round

Guard is the first of two extra bits kept on the right during intermediate calculations of floating point numbers. It is used to improve rounding accuracy. Round is a method to make the intermediate floating-point result fit the floating-pint format; t he goal is typically to find the nearest number that can be represented in the format. IEEE 754, therefore, always keeps two extra bits on the right during intermediate additions, called guard and round, respectively.

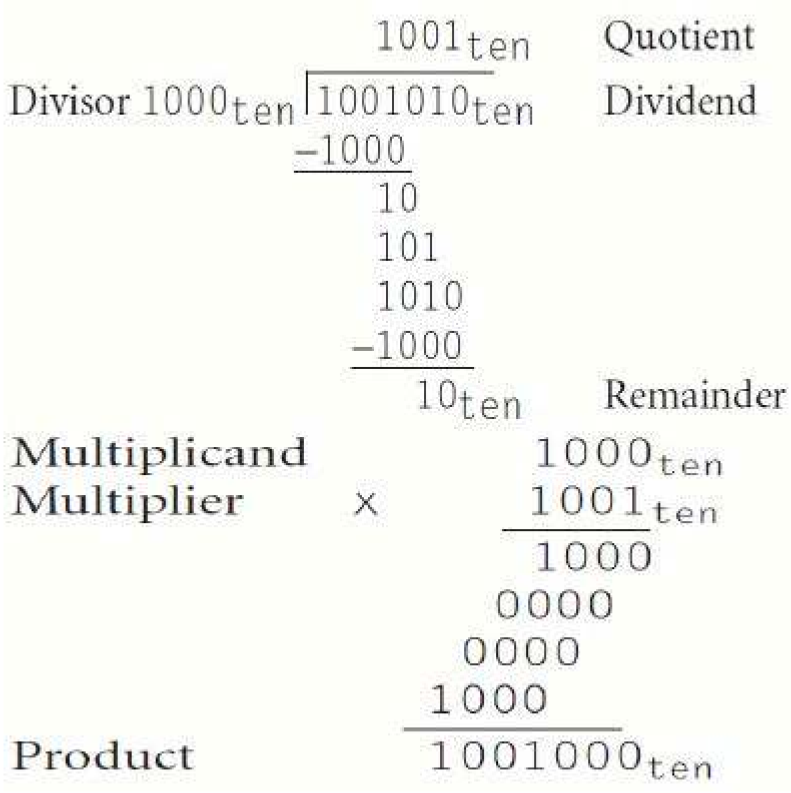
4. Define ULP

Units in the Last Place is defined as the number of bits in error in the least significant bits of the actual number and the number that can be represented.

5. What is meant by sub-word parallelism?

Given that the parallelism occurs within a wide word, the extensions are classified as sub-word parallelism. It is also classified under the more general name of data level parallelism. They have been also called vector or SIMD, for single instruction, multiple data. The rising popularity of multimedia applications led to arithmetic instructions that support narrower operations that can easily operate in parallel.

6. Multiply 1000 * 1001 .



1001_{ten} Quotient

Divisor 1000_{ten} $\overline{)1001010_{ten}}$ Dividend

$\underline{-1000}$

10

101

1010

$\underline{-1000}$

10_{ten} Remainder

Multiplicand 1000_{ten}

Multiplier × $\underline{1001_{ten}}$

1000

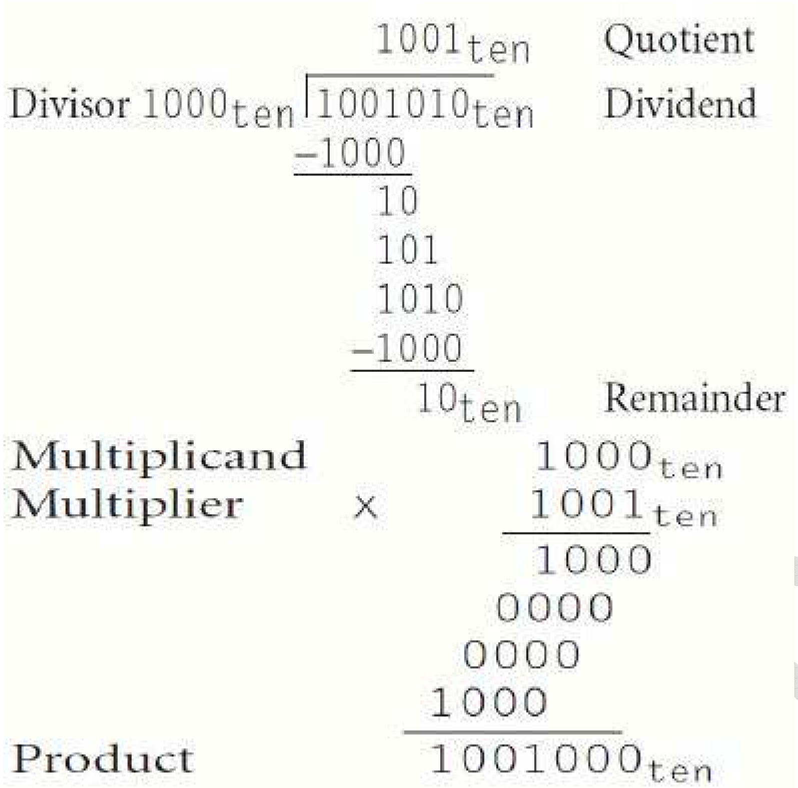
0000

0000

$\underline{1000}$

Product 1001000_{ten}

7. Divide 1,001,010 by 1000.



```
                        1001ten        Quotient
Divisor 1000ten | 1001010ten           Dividend
                 -1000
                     10
                     101
                     1010
                    -1000
                       10ten           Remainder
Multiplicand                1000ten
Multiplier       x          1001ten
                            1000
                           0000
                          0000
                         1000
Product                  1001000ten
```

8. What are the steps in the floating-point addition?

The steps in t he floating-point addition are

1. Align the decimal point f the number that has the smaller exponent.

2. Addition of the significands.

3. Normalize the sum.

4. Round the result.

9. Write the IEEE 754 floating point format.

The IEEE 754 standard floating point representation is almost always an approximation of the real number.

$$(-1)^S \times (1 + \text{Fraction}) \times 2^{(\text{Exponent} - \text{Bias})}$$

10. Explain the Multiplication algorithm in detail with diagram and examples

Signed multiplication

Booth algorithm

Booth algorithm for Signed multiplication

Faster Multiplication

Multiply in MIPS

11. Discuss in detail about division algorithm in detail with diagram and examples.

Signed division

Faster division

Division in MIPS

Step 1: Test divisor<dividend.

Step 2: if divisor<dividend.

Step 1: shift the divisor right by 1 bit

12. Explain in detail about floating point representation

IEEE 754 standard

Scientific notation in binary

Single precision floating point IEEE 754 standard

Double precision floating point IEEE 754 standard

Normalization representation

Sizes

Sign bit

Exponent

Special values

De-normalized representation

Floating point under flow and over flow

Guard and rounding.

13. Explain in detail about floating point arithmetic operation.

Floating point addition and subtraction

Floating point addition Procedure Example

Floating point Multiplication

Procedure

Example

Floating point in MIPS

14. Explain in detail about basic concepts of ALU design

1-bit ALU design

Full adder

32-bit ALU design

MIPS ALU design

Arithmetic for multimedia

15. Explain in detail about arithmetic operation

Boolean addition

Boolean subtraction

Overflow

MIPS Overflow handling

Ripple carry adder

Fast adder circuit

Carry look ahead adder

16. What are the 2 IEEE standards for floating point numbers?

1. single
2. double

17. What is overflow, underflow case in single precision (sp)?

Underflow-In SP it means that the normalized representation requires an exponent less than -126. Overflow-In SP it means that greater than +127.

18.What are the exceptions encounted for FP operation?

The exceptions encountered for FP operation are overflow, underflow, I/0, inexact and invalid values.

19. What is guard bits?

Guard bits are extra bits which are produced during the intermediate steps to yield maximum accuracy in the final results.

20. What are the ways to truncate guard bits?

1. Chopping
2. Von Neumann rounding
3. Rounding procedure

UNIT

UNIT III

PROCESSOR AND CONTROL UNIT

1. What is meant by data path element?

A data path element is a unit used to operate on or hold data within a processor. In the MIPS implementation, t he data path elements include the instruction and data memories, the register file, the ALU, and adders.

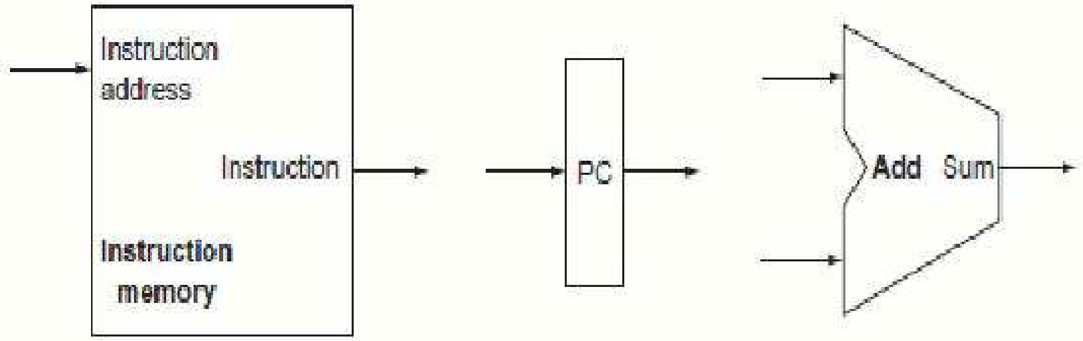
2. What is the use of PC register?

Program Counter (PC) is the register containing the address of the instruction in the program being executed.

3. What is meant by register file?

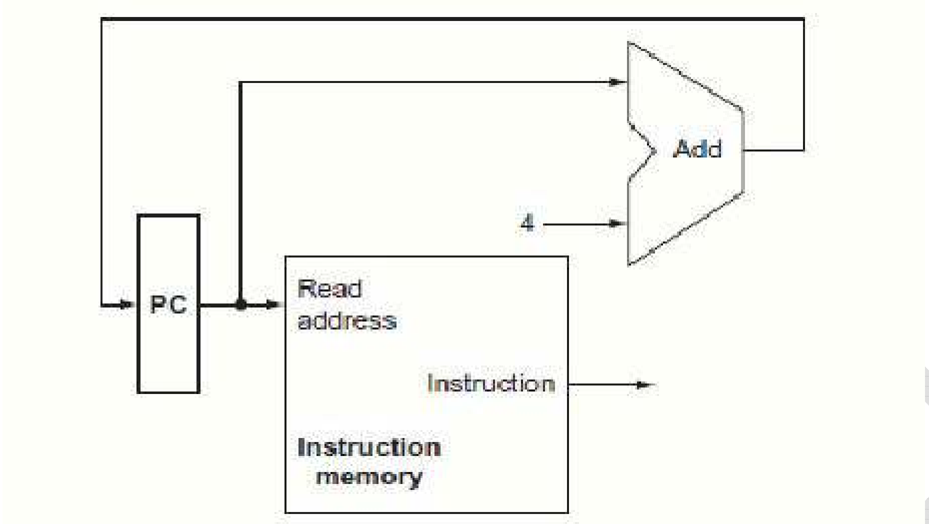
The processor's 32 general purpose registers are stored in a structure called a register file. A register file is a collection of registers in which any register can be read or written by specifying the number of the register in the file. The register file contains the register state of the computer.

4. What are the two state elements needed to store and access an instruction?



a. Instruction memory b. Program counter c. Adder

5. Draw the diagram of portion of datapath used for fetching instruction.



6. Define Sign Extend

Sign-extend is used to increase the size f a data item b y replicating the high-order sign bit f the original data item in the high order bits f the larger, destination data item.

7. What is meant by branch target address?

Branch target address is the address specified in a branch, which becomes the new program counter (PC) if the branch is taken. In the MIPS architecture the branch target is given by the sum of the offset field of the instruction and the address f the instruction following the branch.

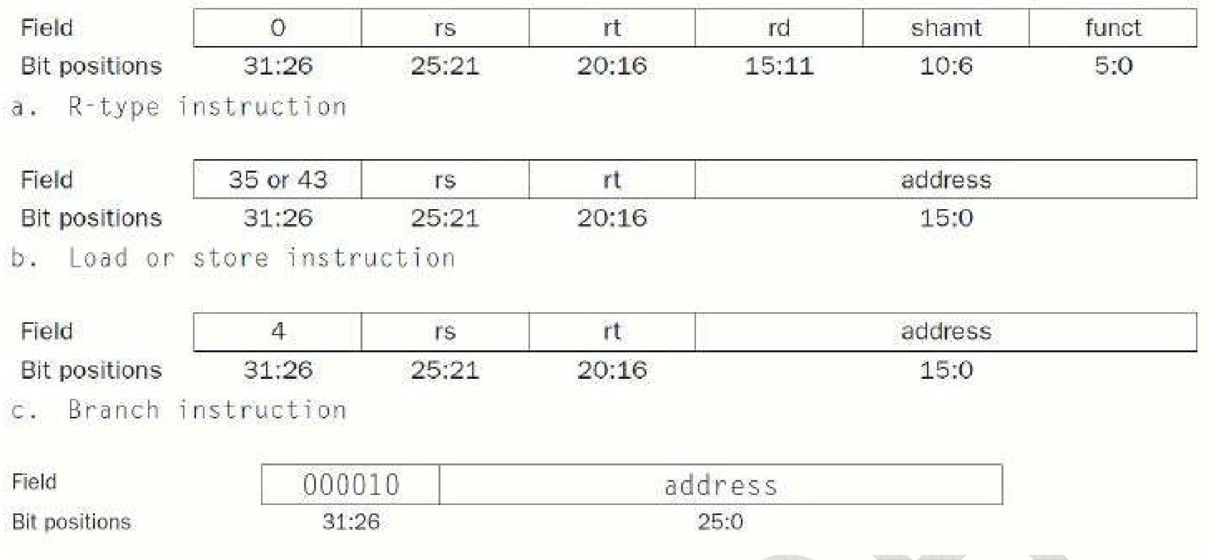
8. Differentiate branch taken from branch not taken.

Branch taken is a branch where the branch condition is satisfied and the program counter (PC) becomes the branch target. All unconditional ju mps are taken branches. Branch not taken or (untaken branch) is a branch where the branch condition is false and the program counter (PC) becomes the address of the instruction that sequentially follows the branch.

9. What is meant by delayed branch?

Delayed branch is a type of branch where the instruction immediately following the branch is always executed, independent of whether the branch condition is true or false

10. What are the three instruction classes and their instruction formats?



Field	0	rs	rt	rd	shamt	funct
Bit positions	31:26	25:21	20:16	15:11	10:6	5:0

a. R-type instruction

Field	35 or 43	rs	rt	address
Bit positions	31:26	25:21	20:16	15:0

b. Load or store instruction

Field	4	rs	rt	address
Bit positions	31:26	25:21	20:16	15:0

c. Branch instruction

Field	000010	address
Bit positions	31:26	25:0

The three instruction classes (R-type, load and store, and branch) use two different instruction formats.

11. Write the instruction format for the jump instruction.

The destination address for a jump instruction is formed by concatenating the upper 4 bits of the current PC + 4 to the 26-bit address field in the ju mp instruction and add ing 00 as the 2 low-order bits.

12. What is meant by pipelining?

Pipelining is an implementation technique in which multiple instructions are overlapped in execution. Pipelining improves performance b y increasing instruction throughput, as opposed to decreasing the execution time of an individual instruction.

13. What is meant by forwarding?

Forwarding, also called bypassing, is a method of resolving a data hazard by retrieving the missing data element from internal buffers rather than waiting for it to arrive from programmer visible registers or memory.

14. What is pipeline stall?

Pipeline stall, also called bubble, is a stall initiated in order to resolve a hazard. They can be seen elsewhere in the pipeline.

15. What is meant by branch prediction?

Branch prediction is a method f resolving a branch hazard that assumes a given outcome for the branch and proceeds from that assumption rather than waiting to ascertain the actual outcome.

16. Explain the basic MIPS implementation of instruction set

The memory –reference instruction load word(lw) and store word(sw)

The arithmetic –logical instruction add, sub and or and slt

The instruction branch equal (beq) and jump (j)

Clocking methodology

17. Explain in detail about building a data path

Data path elements

Data path for branch instruction

Creating a single Data path

18. Explain in detail about control implementation scheme.

The ALU control

Designing the control unit

Operation of the Data path for an R type instruction

Finalizing the control

A multi cycle implementation

19. What are control hazards? Explain the methods for dealing with the control hazards.

Reducing the delay of branch

Pipeline branch

dynamic branch prediction

1-bit prediction scheme

2-bit prediction scheme

20. Discuss the data hazards and forwarding in pipelining

1a) EX/MEM. Register Rd =ID/EX. register RS

1b) EX/MEM. Register Rd =ID/EX. register RT

2a) MEM/WB. Register Rd =ID/EX. register RS

2b) MEM/WB. Register Rd =ID/EX. register RT

Dependence detection

ß Sub $1,$2,$3 ß add $1,$2,$3 ß or $1,$2,$3

EX Hazards

MEM Hazards

21. How exceptions are handled in MIPS

Instruction fetch and memory stages

Memory protection violation

Instruction decode stages

Undefined illegal opcode

Execution stage

Arithmetic exception

Write back stages

UNIT-IV

PARALLELISM

1. What is meant by ILP?

Pipelining exploits the potential parallelism among instructions. This parallelism is called instruction-level parallelism (ILP). There are two primary methods for increasing the potential amount of instruction-level parallelism. 1. Increasing the depth of the pipeline to overlap more instruct ions. 2. Multiple issue.

2. What is multiple issue? Write any two approaches.

Multiple issue is a scheme whereby multiple instructions are launched in one clock cycle. It is a method for increasing the potential amount f instruction-level parallelism. It is done by replicating the internal components of the cmputer so that it can launch multiple instructions in every pipeline stage. The two approaches are: 1. Static multiple issue (at compile time) 2. Dynamic multiple issue (at run time)

3. What is meant by speculation?

One of the most important methods for finding and exploiting more ILP is speculation. It is an approach where by the compiler or processor guesses the outcome of an instruction to remove it as dependence in executing other instructions. For example, we might speculate on the outcome of a branch, so that instructions after the branch could be executed earlier.

4. Define Static Multiple Issue

Static multiple issue is an approach to implement a multiple-issue processor where many decisions are made by the compiler before execution.

5. Define Issue Slots and Issue Packet

Issue slots are the positions from which instructions could be issued in a given clock cycle. By analogy, these correspond to positions at the starting blocks for a sprint. Issue packet is the set of instruct ions that issues together in one clock cycle; the packet may be determined statically by the compiler or dynamically by the processor.

6. Define VLIW

Very Long Instruction Word (VLIW) is a style of instruct ion set architecture that launches many operations that are defined to be independent in a single wide instruction, typically with many separate opcode fields.

7. Define Superscalar Processor

Superscalar is an advanced pipelining technique that enables the processor to execute more than one instruct ion per clock cycle by selecting them during execution. Dynamic multiple-issue processors are also known as superscalar processors, or simply superscalars.

8. What is meant by loop unrolling?

An important compiler technique to get more performance from loops is loop unrolling, where multiple cop ies of the loop body are made. After unrolling, there is more ILP available b y overlapping instructions from different iterations.

9. What is meant by anti-dependence? How is it removed?

Anti-dependence is an ordering forced b y the reuse of a name, typically a register, rather than by a true dependence that carries a value between two instructions. It is also called as name dependence. Register renaming is t he technique used to remove anti-dependence in which the registers are renamed by the compiler or hardware.

10. What is the use of reservation station and reorder buffer?

Reservation station is a buffer within a functional unit that holds the operands and the operation. Reorder buffer is the buffer that holds results in a dynamically scheduled processor until it is safe to store the results to memory or a register.

11. Differentiate in-order execution from out-of-order execution.

Out-f-order execution is a situation in pipelined execution when an instruction is b locked from executing does not cause the following instructions to wait. It preserves the data flow order of the program. In-order execution requires the instruction fetch and decodes unit tissue instructions in order, which allows dependences tbe tracked, and requires t he commit unit to write results to registers and memory in program fetch order. This conservative mode is called in-order commit.

12. What is meant by hardware multithreading?

Hardware multithreading allows multiple threads to share the functional units of a single processor in an overlapping fashion to try to utilize the hardware resources efficiently. To permit this sharing, the processor must duplicate the independent state of each thread. It Increases the utilization of a processor.

13. What are the two main approaches to hardware multithreading?

There are two main approaches to hardware multithreading. Fine-grained multithreading switches between threads on each instruction, resulting in interleaved execution of multiple threads. This interleaving is often done in a round-robin fashion, skipping any threads that are stalled at that clock cycle. Coarse-grained multithreading is an alternative to fine-grained multithreading. It switches threads only on costly stalls, such as last-level cache misses.

14. Explain Instruction level parallelism

Dynamic, hardware intensive approach

Static, complier intensive approach

Loop level parallelism

Data dependences and hazards

Data dependences

Control dependences

Structure dependences

ILP architecture

15. Explain the difficulties faced by parallel processing programs

First major challenges –goods speedup

Second major challenges –remote access in parallel processing

16. Explain in detail Flynn’s classification of parallel hardware

Introduction

Flynn’s taxonomy

Single instruction stream, single data stream (SISD)

Single instruction stream, multiple data stream (SIMD)

Multiple instruction streams, single data stream (MISD)

Multiple instruction stream, multiple data stream (MIMD)

17. Explain in detail hardware Multithreading

Fine grained multithreading

Advantages

Disadvantages

Coarse grained multithreading

Advantages

Disadvantages

SMT

A super scalar without multithreading

A super scalar Fine grained multithreading

A super scalar Coarse grained multithreading

A super scalar SMT multithreading

18. Explain Multicore processor

Centralized shared memory architecture

Classification based on communication models

ß Distributed shared memory

ß Message passing multiprocessor

19. Explain briefly about queues.

A **queue** is a First-In-First-Out (FIFO) data structure. In a FIFO data structure, the first element added to the queue will be the first one to be removed. A queue is an example of a linear data structure. Queue **overflow** results from trying to add an element onto a full queue and queue **underflow** happens when trying to remove an element from an empty queue. A **bounded queue** is a queue limited to a fixed number of items.

20. What are the steps of fetching instruction?

Let us assume that we are trying to fetch the instruction at memory location 2005. That means that the program counter is now set to that value.

The following is the sequence of operations:

- The program counter places the address value on the address bus and the controller issues a RD signal.
- The memory's address decoder gets the value and determines which memory location is being accessed.
- The value in the memory location is placed on the data bus.
- The value on the data bus is read into the instruction decoder inside the microprocessor.
- After decoding the instruction, the control unit issues the proper control signals to perform the operation.

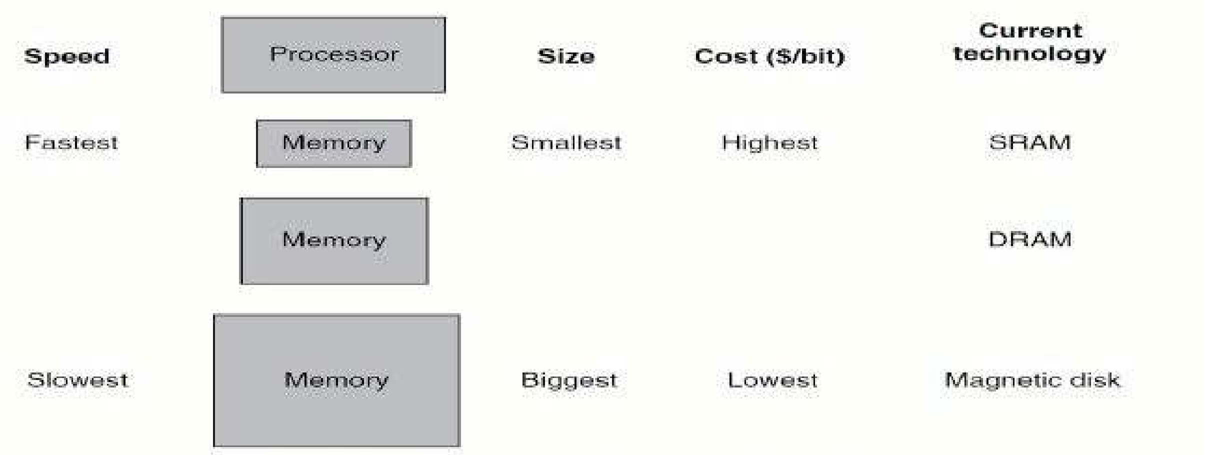
UN IT-V

MEMORY AND I/O SYSTEMS

1. What are the temporal and spatial localities of references?

Temporal locality (locality in time): if an item is referenced, it will tend to be referenced again soon. Spatial locality (locality in space): if an item is referenced, items whose addresses are close by will tend to be referenced son.

2. Write the structure of memory hierarchy.



The basic structure of a memory hierarchy.

3. What are the various memory technologies?

The various memory technologies are: 1. SRAM semiconductor memory 2. DRAM semiconductor memory 3. Flash semiconductor memory 4. Magnetic disk

4. Differentiate SRAM from DRAM.

SRAMs are simply integrated circuits that are memory arrays with a single access port that can provide either a read or a write. SRAMs have a fixed access time to any datum.

SRAMs don't need to refresh and so the access time is very close to the cycle time. SRAMs typically use six to eight transistors per bit to prevent the information from being disturbed when read. SRAM needs only minimal power to retain the charge in standby mode. In a dynamic RAM (DRAM), the value kept in a cell is stored as a charge in a capacitor. A single transistor is then used taccess this stored charge, either tread the value or toverwrite the charge stored there. Because DRAMs use only a single transistor per bit of storage, they are much denser and cheaper per bit than SRAM.

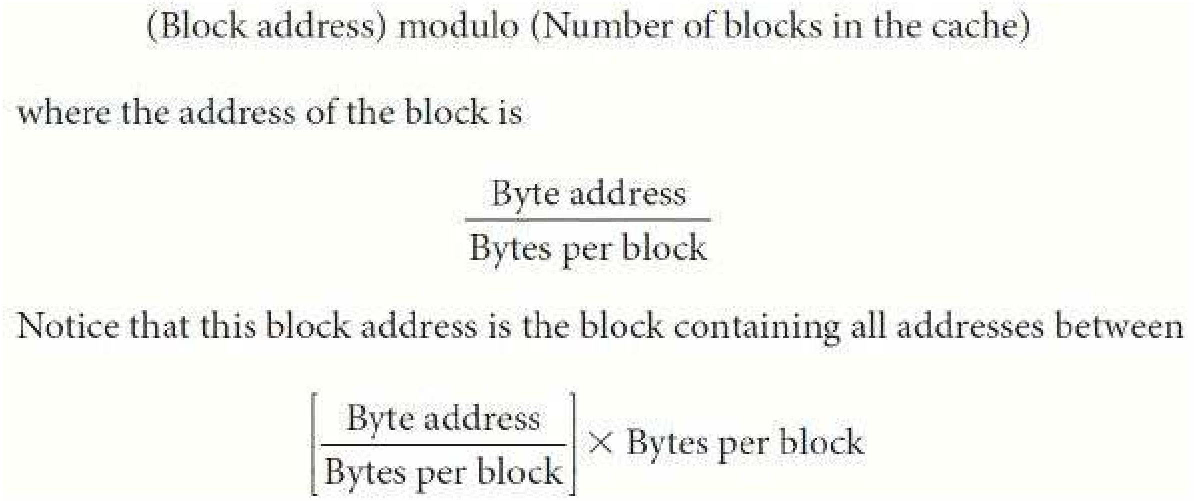
6. Define - Rotational Latency

Rotational latency, also called rotational delay, is the time required for the desired sector of a disk to rotate under the read/write head, usually assumed to be half the rotating time.

7. What is direct-mapped cache?

Direct-mapped cache is a cache structure in which each memory location is mapped to exactly one location in the cache. For example, almost all direct-mapped caches use this mapping to find a block, (Block address) module (Number f blocks in the cache)

8. Consider a cache with 64 blocks and a block size of 16 bytes. To what block number does byte address 1200 map?



(Block address) modulo (Number of blocks in the cache)

where the address of the block is

$$\frac{\text{Byte address}}{\text{Bytes per block}}$$

Notice that this block address is the block containing all addresses between

$$\left\lfloor \frac{\text{Byte address}}{\text{Bytes per block}} \right\rfloor \times \text{Bytes per block}$$

10. What are the writing strategies in cache memory?

Write-through is a scheme in which writes always update both the cache and the next lower level of the memory hierarchy, ensuring that data is always consistent between the two. Write-back is a scheme that handles writes by updating values only to the block in the cache, then writing the modified block to the lower level of the hierarchy when the block is replaced.

11. What are the steps tbe taken in an instruction cache miss?

The steps to be taken on an instruction cache miss are

1. Send the original PC value to the memory.
2. Instruct main memory to perform a read and wait for the memory to complete its access.

12. What is meant by virtual memory?

Virtual memory is a technique that uses main memory as a "cache" for secondary storage. Two major motivations for virtual memory: to allow efficient and safe sharing of memory among multiple programs, and to remove the programming burdens f a small, limited amount of main memory.

13. Differentiate physical address from logical address.

Physical address is an address in main memory. Logical address (or) virtual address is the CPU generated addresses that corresponds to a location in virtual space and is translated by address mapping to a physical address when memory is accessed.

14. Define Page Fault

Page fault is an event that occurs when an accessed page is not present in main memory.

16. Explain Memory Technologies

Main memory is the next level hierarchy

It satisfies the demand of caches and serves as I/interface

Types

SRAM

CMOS

DRAM

SYCHRONOUS DRAM

DOUBLE DATA RATE DRAM

RAMBUS MEMORY

ROM

PROM

EPROM

EEPROM

FLASH MEMORY

FLASH CARD

FLASH DRIVE

17. Explain about cache memory in detail

• Cache memory is a small amount of fast memory Placed between two levels of memory hierarchy To bridge the gap in access times

– Between processor and main memory (our focus)

– Between main memory and disk (disk cache)

How Cache Memory Works

• Pre-fetch data into cache before the processor needs it

∗ Need to predict processor future access requirements

» Not difficult owing to locality of reference

• Important terms

∗ Miss penalty

∗ Hit ratio

∗ Miss ratio = (1 – hit ratio)

∗ Hit time

18. Explain about DMA Controller

DMA channel: DMA channel is issued to transfer data between main memory and peripheral device in order to perform the transfer of data. The DMA controller access rs address and data buses. DMA with help of schematic diagram of controller on tile needs the dual circuits of and e to communicate with -CPU and I/O device. In addition, it nee s an address register; a word count register, and a set of es. The address register and address lines are used for communication with memory to word count register specifies the no. of word that - must be transfer may be done directly between the device and memory.

19. Explain about I/O processor

Input/output processor/information processor: It is designed to handle input/ output processes of a device or the computer. This processor is separate from the main processor (CPU). I/processor is similar to CPU but it controls input output operations only. The computer having I/O processor relieves CPU from Input/output operations only. CPU is the master processor of the computer and it instructs the I/processor to handle the input output tasks. I/processor cannot work independently and is controlled by the CPU.

The I/O processor is composed of commercially available TTL logic circuits that generate the micro instructions necessary to implement the I/instructions. The I/O processor is fully synchronous with the system clock and main processor. it receives starting control from the main processor (CPU) whenever an input output instruction is read from memory. The I/O processor makes use of system buses after taking the permission from the CPU. It can instruction the I/processor 1/0 processor responds to CPU by placing a status word at prescribed location to be checked out by the CPU later on CPU informs the 1/0 processor to find out the 1/0 program and ask 1/0 processor to transfer the data. I/O

20. What are the advantages you got with virtual memory?

Permit the user to construct program as though a large memory space were available, equal to totality auxiliary memory. Each address that is referenced by CPU goes through an address mapping from so called virtual address to physical address main memory.

There are following advantages we got with virtual memory:

1. Virtual memory helps in improving the processor utilization.

2. Memory allocation is also an important consideration in computer programming due thigh cost of main memory.

3. The function of the memory management unit is therefore to translate virtual address to the physical address.

4. Virtual memory enables a program to execute on a computer with less main memory when it needs.

5. Virtual memory is generally implemented by demand paging concept In demand paging, pages are only loaded to main memory when they are required

6. Virtual memory that gives illusion to user that they have main memory equal to capacity of secondary stages media.

The virtual memory is concept of implementation which is transferring the data from secondary stage media to main memory as and when necessary. The data replaced from main memory is written back to secondary storage according to predetermined replacement algorithm.

PART B

UN IT-I

1. Explain the Eight ideas of the Computer architects in detail.(8)

2. Explain the components of a computer with the block diagram in detail.(16)

3. Explain the technologies for building computer over time with a neat graph.(6)

4. Explain the chip manufacturing process with a neat diagram in detail.(10)

4. Explain the techniques used to measure the performance of a computer.(8)

5. (i)Prove that how performance and execution are inverse to each other.(2)

(ii) If computer A runs a program in 10 seconds and computer B runs the same program in 15 seconds, how much faster is A than B?(2)

(iii)Write the formula to calculate the CPU execution time for a program.(2)

(iv) Write the formula to calculate the CPU clock cycles.(2)

(v) Write the formula to calculate the classic CPU Performance equation.(2)

6. Explain how clock rate and power are related to each other in microprocessor over years with a neat graph.(6)

7. Explain the need to switch from uni-processors to multiprocessors and draw the performance chart for processors over years. (6)

8. Explain the basic instruction types with examples.(6)

9. (i)Explain the different types of instruction set architecture in detail(6)

(ii)Explain MIPS assembly language notation for arithmetic, Data transfer, logical and conditional branch and unconditional branch operations.

10. What do you mean by addressing modes? Explain various addressing modes with the help of examples.(16)

UN IT-II

1. Explain the design of ALU in detail.(16)

2. Explain with an example how to multiply two unsigned binary numbers.(8)

3. Explain the Working of a Carry-Look Ahead adder. (16)

4. Derive and explain an algorithm for adding and subtracting two floating point binary numbers.(8)

5. Describe the algorithm for integer division with suitable examples.(16)

6. Perform the multiplication using Carry save addition of summands. (6) 45 X 45

7. Perform the integer division using non-restoring and restoring division. (10) 9 / 4

UN IT-III

1. Discuss the basic concepts of pipelining.

2. State and explain the different types of hazards that can occur in a pipeline.

3. Draw and explain the modified three-bus structure of the processor suitable for four stage pipelined execution. How this structure is suitable to provide four-stage pipelined execution?

4. What is data hazard? Explain the methods for dealing with the data hazards

5. Describe the data and control path techniques in pipelining.(10)

6. What is instruction hazard? Explain in detail how to handle the instruction hazards in pipelining with relevant examples.(10)

7. Describe the techniques for handling control hazards in pipelining.(10)

8. Write note on exception handling.(6)

UN IT-IV

1. Explain instruction level parallelism in detail?

2. Explain parallel processing challenges in detail?

4. What is hardware multithreading? Explain the various approaches in detail?

5. Explain multicore processors in detail?

6. Compare SISD, SIMD, MISD, and MIMD in detail?

7. Explain the following:

i) Implicit and Explicit multithreading.

ii) Interleaved, Blocked and Simultaneous multithreading.

8. What are multicore processors? Explain the common configurations that support multiprocessing?

UN IT-V

1. Explain in detail about memory Technologies

2. Explain in detail about memory Hierarchy with neat diagram

3. Describe the basic operations of cache in detail with diagram

4. Discuss the various mapping schemes used in cache design(10)

A byte addressable computer has a small data cache capable of holding eight 32-bit words. Each cache block contains 132-bit word. When a given program is executed, the processor reads data from the following sequence of hex addresses – 200, 204, 208, 20C, 2F4, 2F0, 200, 204,218, 21C, 24C, 2F4. The pattern is repeated four times. Assuming that the cache is initially empty, show the contents of the cache at the end of each pass, and compute the hit rate for a direct mapped cache. (6)

5. Discuss the methods used to measure and improve the performance of the cache.

6. Explain the virtual memory address translation and TLB with necessary diagram.

7. Draw the typical block diagram of a DMA controller and explain how it is used for direct data transfer between memory and peripherals.

8. Explain in detail about interrupts with diagram

9. Describe in detail about programmed Input/ Output with neat diagram

10. Explain in detail about I/O processor.

11. Describe in detail about IOP organization.

12. Write short notes on the following
   (a) Magnetic disk drive (8) (b) Optical drives (8)