1. **What are the five classic components of a computer? (April/Nov 2017)**
   The five classic components of a computer are input, output, memory, data path, and control, with the last two sometimes combined and called the processor.

2. **What is the function of data path and control path?**
   Data path Unit
   1. It is also called as brawn of the processor. It performs the arithmetic operations.
   2. The entire operation can be done with the help of registers. Because registers are faster than memory. Registers can hold variables and intermediate results.
   3. Memory traffic is reduced, so program runs faster.

   Control Unit
   1. It is also called as a brain of the processor. It fetches and analyses the instructions one-by-one and issue control signals to all other units to perform various operations.
   2. For a given instruction, the exact set of operations required is indicated by the control signals. The results of instructions are stored in memory.
   3. The component of the processor that commands the datapath, memory, and I/O devices according to the instructions of the program.

3. **What is instruction set architecture?**
   An abstract interface between the hardware and the lowest level software that encompasses all the information necessary to write a M/C language program that will run correctly, including instruction, registers, memory access, I/O and so on.

4. **Define Application Binary Interface (ABI).**
   The combination of the basic instruction set and the operating system interface provided for application programmers is called the application binary interface (ABI).

5. **Differentiate SRAM and DRAM.**

Static RAM (SRAM)	Dynamic RAM (DRAM)
Information will be available as long as power is available	It retains data for few ms even in the absence of a power source based on the charge of capacitor
No refreshing is needed	Refreshing is needed
Less packaging density	High packaging density
More complex Hardware	Less complex hardware
More expensive	Less expensive
No random access	Random access is possible
Access time 10 ns	Access time 50 ns

6. **Define flash memory.**
   Flash Memory is a nonvolatile semiconductor memory. It is cheaper and slower than DRAM but more expensive per bit and faster than magnetic disks. Access times are about 5 to 50 microseconds and cost per gigabyte in 2012 was $0.75 to $1.00.

7. **Compare volatile and nonvolatile memory.**

Volatile memory	Non volatile memory
Random Access Memory is an example of volatile memory	Read Only Memory is an example of Non volatile memory
Data lost when the power turns off and that is used to hold data and program while they are running.	It retains data even in the absence of a power source and that is store programs between runs.
Temporary storage medium	Permanent storage medium

8. **Differentiate throughput and response time.**

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/O activities, operating system overhead, CPU execution time, and so on.

Throughput or bandwidth

The total amount of work done in a given time is known as throughput. Decreasing response time almost always improves throughput.

9. **Write the CPU performance equation.**

The basic performance equation in terms of instruction count (the number of instructions executed by the program), CPI, and clock cycle time:

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

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

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

$$\text{Time} = \text{Seconds/Program} = \frac{\text{Instructions}}{\text{Program}} \times \frac{\text{Clock cycles}}{\text{Instruction}} \times \frac{\text{Seconds}}{\text{Clock cycle}}$$

10. **Define Amdhal's law.**

Amdahl's Law: A rule stating that the performance enhancement possible with a given improvement is limited by the amount that the improved feature is used.

$$\text{Execution time after improvement} = \frac{\text{Execution time affected by improvement}}{\text{Amount of improvement}} + \text{Execution time unaffected}$$

11. **Define CPI.**

The term clock cycles per instruction, which is the average number of clock cycles each instruction takes to execute, is often abbreviated as CPI.

12. **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.

13. **Write the formula for CPU execution time for a program.**

CPU execution time: Also called **CPU time**. The actual time the CPU spends computing for a specific task.

User CPU time: The CPU time spent in a program itself.

System CPU time: The CPU time spent in the operating system performing tasks on behalf of the program.

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

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

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

14. **Write the formula for CPU clock cycles required for a program.**

Clock cycle: Also called tick, clock tick, clock period, clock, or cycle. The time for one clock period, usually of the processor clock, this runs at a constant rate.

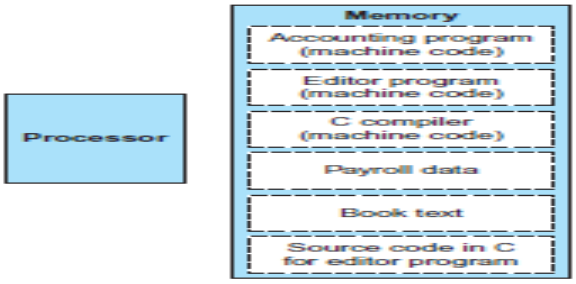
Clock period: This is the length of each clock cycle.

Clock rate: This is the inverse of the clock period.

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

15. Define – Stored Program Concepts.

- The idea that instructions and data of many types can be stored in memory as numbers, leading to the stored program computer.
- Stored programs allow a computer that performs by loading memory with programs and data and to begin executing at a given location in memory.



16. What are the fields in an MIPS instruction?

MIPS Fields

MIPS fields are given names to make them easier to discuss:

op	rs	rt	rd	shamt	funct
6 bits	5 bits	5 bits	5 bits	5 bits	6 bits

Here is the meaning of each name of the fields in MIPS instructions:

- *op:* Basic operation of the instruction, traditionally called the opcode.
- *rs:* The first register source operand.
- *rt:* The second register source operand.
- *rd:* The register destination operand. It gets the result of the operation.
- *shamt:* Shift amount. (Section 2.6 explains shift instructions and this term; it will not be used until then, and hence the field contains zero in this section.)
- *funct:* Function. This field, often called the *function code*, selects the specific variant of the operation in the op field.

17. What MIPS instruction does this represent? Choose from one of the four options below.

op	rs	rt	rd	shamt	funct
0	8	9	10	0	34

1. sub $t0, $t1, $t2
2. add $t2, $t0, $t1
3. sub $t2, $t1, $t0
4. **sub $t2, $t0, $t1**

Ans:4

18. What are the operations of an instruction?

Instruction performs one of the following operations

1. Data transfer between register and memory.
2. ALU operation.
3. Program control and sequencing.
4. I/O transfer.

19. What are the different types of operands? Give examples.

1. **Memory Operand**

   Load: The data transfer instruction that copies data from memory to a register is traditionally called load.

   Store: The instruction complementary to load is traditionally called store; it copies data from a register to memory. The format of a store is similar to that of a load.

   Ex: lw $t0,32($s3) # Temporary reg $t0 gets A[8]
   sw $t0,48($s3) # Stores h + A[8] back into A[12]

2. **Constant or Immediate Operand**

   The constants would have been placed in memory when the program was loaded. For example, to add the constant 4 to register $s3.

   Ex: addi $s3,$s3,4 # $s3 = $s3 + 4

20. What is the compiled MIPS code for this statement?

f = (g + h) – (i + j);

MIPS code:

The variables f, g, h, i, and j are assigned to the registers $s0, $s1, $s2, $s3, and $s4, respectively.

```
add $t0,$s1,$s2 # register $t0 contains g + h
add $t1,$s3,$s4 # register $t1 contains i + j
sub $s0,$t0,$t1 # f gets $t0 – $t1, which is (g + h)–(i + j)
```

21. Let's assume that A is an array of 100 words and that the compiler has associated the variables g and h with the registers $s1 and $s2 as before. Let's also assume that the starting address, or base address, of the array is in$s3. Compile this C assignment statement:

g = h + A[8];

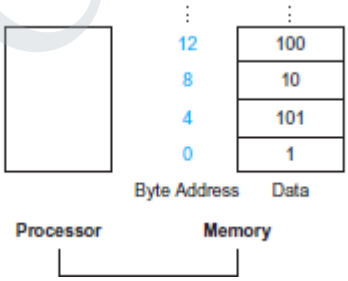
Solution:

```
lw $t0,32($s3)      # Temporary reg $t0 gets A[8]
add $s1,$s2,$t0     # g = h + A[8]
```

The constant in a data transfer instruction (8) is called the off set, and the register added to form the address ($s3) is called the base register.

22. Define Alignment restriction.

Alignment restriction: A requirement that data be aligned in memory on natural boundaries.



23. Define Little Endian address and Big Endian Address.

Little Endian Address: A lower byte address of the word is specified in least significant byte of the word is known as Little endian address.

Big Endian Address: A lower byte address of the word is specified in most significant byte of the word is known as big endian address.

24. List the different addressing modes.

1**. Immediate addressing**, where the operand is a constant within the instruction itself

2. **Register addressing**, where the operand is a register

3. **Base or displacement addressing,** where the operand is at the memory location whose address is the sum of a register and a constant in the instruction

4. **PC-relative addressing**, where the branch address is the sum of the PC and a constant in the instruction

5. **Pseudo direct addressing**, where the jump address is the 26 bits of the instruction concatenated with the upper bits of the PC

25. What is instruction register? (Nov 2016)

The **instruction register (IR)** holds the instruction that is currently being executed. Its output is available to the control circuits which generate the timing signals that control the various processing elements involved in executing the instruction.

26. What is program counter?

The **program counter (PC)** keeps track of the execution of a program. It contains the memory address of the next instruction to be fetched and executed.

27. List out the methods used to improve system performance?

The methods used to improve system performance are

- Processor clock
- Basic Performance Equation
- Pipelining
- Clock rate
- Instruction set
- Compiler

28. What are the addressing modes and its various types? (Nov 2017)

The different ways in which the location of a operand is specified in an instruction is referred to as addressing modes. The various types are Immediate Addressing, Register Addressing, Base or Displacement Addressing, PC-Relative Addressing, Pseudo direct Addressing.

29. Define Relative mode addressing.(Nov 2014)

The **relative addressing mode** is similar to the indexed addressing mode with the exception that the PC holds the base address. This allows the storage of memory operands at a fixed offset from the current instruction and is useful for 'short' jumps. Example: jump 4

PART-B

1. Explain the various components of computer System with neat diagram (13)

2. Discuss in detail the basic operational concepts (13)

3. Discuss in detail the various measures of performance of a computer (13)

4. Explain operations and operands of computer Hardware in detail. (13)

5. i) Discuss the Logical operations and control operations of computer. (13)

6. Consider three different processors P1, P2, and P3 executing the same instruction set. P1 has a 3 GHz clock rate and a CPI of 1.5. P2 has a 2.5 GHz clock rate and a CPI of 1.0. P3 has a 4.0 GHz clock rate and has a CPI of 2.2. (8)

a. Which processor has the highest performance expressed in instructions per second?

b. If the processors each execute a program in 10 seconds, find the number of cycles and the number of instructions.

c. We are trying to reduce the execution time by 30% but this leads to an increase of 20% in the CPI. What clock rate should we have to get this time reduction?

7. Assume a program requires the execution of 50 × 106 FP instructions, 110 × 106 INT instructions, 80 × 106 L/S instructions, and 16 × 106 branch instructions. The CPI for each type of instruction is 1, 1, 4, and 2, respectively. Assume that the processor has a 2 GHz clock rate. (8)

a. By how much must we improve the CPI of FP instructions if we want the program to run two times faster?

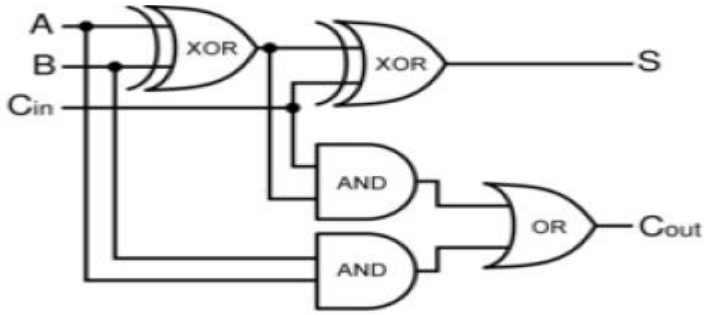
b. By how much must we improve the CPI of L/S instructions if we want the program to run two times faster?

c. By how much is the execution time of the program improved if the CPI of INT and FP instruction is reduced by 40% and the CPI of L/S and Branch is reduced by 30%?

8. Explain the following addressing modes in detail with diagram (13)

i) Immediate addressing ii) Register addressing, iii) Base or displacement addressing, iv) PC-relative addressing v) Pseudo direct addressing.

1. **Define Full Adder (FA) with logic diagram.**
   A full adder adds binary numbers and accounts for values carried in as well as out. A one-bit full adder adds three one-bit numbers, often written as A, B, and Cin; A and B are the operands, and Cin is a bit carried in (from a past addition).The full-adder is usually a component in a cascade of adders, which add 8, 16, 32, etc.



Inputs			Outputs		
a	b	CarryIn	CarryOut	Sum	Comments
0	0	0	0	0	$0 + 0 + 0 = 00_{two}$
0	0	1	0	1	$0 + 0 + 1 = 01_{two}$
0	1	0	0	1	$0 + 1 + 0 = 01_{two}$
0	1	1	1	0	$0 + 1 + 1 = 10_{two}$
1	0	0	0	1	$1 + 0 + 0 = 01_{two}$
1	0	1	1	0	$1 + 0 + 1 = 10_{two}$
1	1	0	1	0	$1 + 1 + 0 = 10_{two}$
1	1	1	1	1	$1 + 1 + 1 = 11_{two}$

2. **Write the overflow conditions for addition and subtraction.**

Operation	Operand A	Operand B	Result indicating overflow
$A + B$	≥ 0	≥ 0	< 0
$A + B$	< 0	< 0	≥ 0
$A - B$	≥ 0	< 0	< 0
$A - B$	< 0	≥ 0	≥ 0

3. **How overflow occur in subtraction? (May 2015)**
   When overflow occurs on integer addition and subtraction, contemporary machines invariably discard the high-order bit of the result and store the low-order bits that the adder naturally produces. Signed integer overflow of addition occurs if and only if the operands have the same sign and the sum has assign opposite to that of the operands.

4. **Define Exception and Interrupt.**
   Exception: An unscheduled event that disturbs the program execution is called exception. It is used to detect overflow.

   Interrupt: An exception that comes from outside of the processor is called interrupt.

5. **Define EPC.**
   MIPS include a register called the Exception Program Counter (EPC) to contain the address of the instruction that caused the exception.

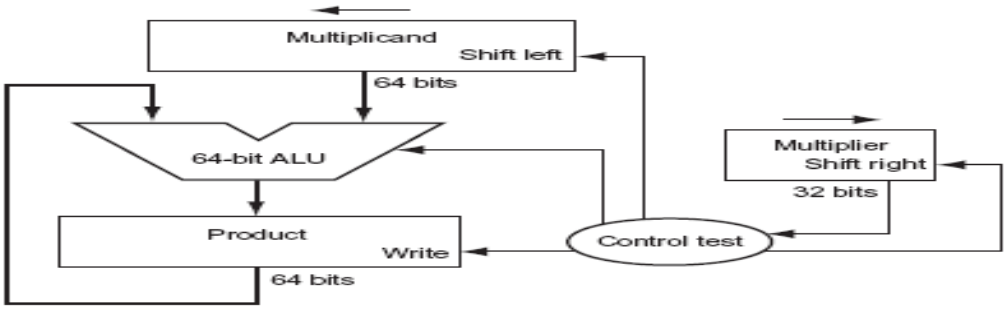
6. **Define generate and propagate function.**
   The generate function is given by Gi=xiyi and
   The propagate function is given as Pi=xi+yi.

7. **What is Carry Save addition?**
   Using carry save addition, the delay can be reduced further still. The idea is to take 3 numbers that we want to add together, x+y+z, and convert it into 2 numbers c+s such that x+y+z=c+s, and do this in O (1) time. The reason why addition cannot be performed in O (1) time is because the carry

information must be propagated. In carry save addition, we refrain from directly passing on the carry information until the very last step

8. **Explain how Boolean subtraction is performed?**
   Negate the subtrahend (i.e. in a-b, the subtrahend is b) then perform addition(2's complement)

9. **Draw the Multiplication hardware diagram.**



10. **List the steps of multiplication algorithm.**

Step: 1

- The least significant bit of the multiplier (Multiplier0) determines whether the multiplicand is added to the Product register.
- If the least significant bit of the multiplier is 1, add the multiplicand to the product.

Step: 2

- If not, go to the next step. Shift left the multiplicand register by 1 bit.

Step: 3

- Then shift right the multiplier register by 1 bit. These three steps are repeated 32 times to obtain the product.
- If each step took a clock cycle, this algorithm would require almost 100 clock cycles to multiply two 32-bit numbers.

11. **What is fast multiplication?**

- In this algorithm the multiplicand is to be added or not is known at the beginning of the multiplication by the 32 multiplier bits.
- One input is the multiplicand ANDed with a multiplier bit, and the other is the output of a prior adder.
- Rather than use a single 32-bit adder 31 times, this hardware "unrolls the loop" to use 31adders and then organizes them to minimize delay.

12. **Define Booth Algorithm.**
    Booth's multiplication algorithm is a multiplication algorithm that multiplies two signed binary numbers in two's complement notation. Booth's algorithm can be implemented by repeatedly adding (with ordinary unsigned binary addition) one of two predetermined values A and S to a product P, then performing a rightward arithmetic shift on P.

13. **List the steps of division algorithm**

Step: 1

- It must first subtract the divisor register from the Remainder register and place the result in the Remainder register.

Step: 2

- Next we performed the comparison in the set on less than instruction.
- If the result is positive, the divisor was smaller or equal to the dividend, so shift the Quotient register to the left, setting the new rightmost bit to 1.
- If the result is negative, the next step Restore the original value by adding the Divisor register to the Remainder register and placing the sum in the Remainder register.

- Also shift the Quotient register to the left, setting the new least significant bit to 0

Step: 3

- The divisor is shifted right by 1 bit and then we iterate again.
- The remainder and quotient will be found in their registers after the iterations are complete.

14. Write Restoring and Non-Restoring division algorithm?

Restoring Division Algorithm:

- Shift A and Q left one binary position.
- Subtract M from A, and place the answer back in A.
- If the sign of A is 1, set q0 to 0 and add M back to A (that is, restore A); otherwise, set q0 to 1.

Non- Restoring Division Algorithm

Step 1: Do the following n times: If the sign of A is 0, shift A and Q left one bit position and subtract M from A; otherwise, shift A and Q left and add M to A. Now, if the sign of A is 0, set q0 to 1; otherwise, set q0 to 0.

Step 2: If the Sign of A is 1, add M to A

15. What is normalization? Give an example

- A number in floating-point notation that has no leading 0^s is known as normalized number. i.e., a number start with a single nonzero digit.
- For example, $1.0_{ten} \times 10^{-9}$ is in normalized scientific notation, but $0.1_{ten} \times 10^{-8}$ and $10.0_{ten} \times 10^{-10}$ are not.

16. Give the representation of single precision floating point number

Floating point:

- Computer arithmetic that represents numbers in which the binary point is not fixed.

Fraction:

- The value, generally between 0 and 1, placed in the fraction field. The fraction is also called the mantissa.

Exponent:

- In the numerical representation system of floating-point arithmetic, the value that is placed in the exponent field.

17. Define overflow and under flow with examples

Overflow:

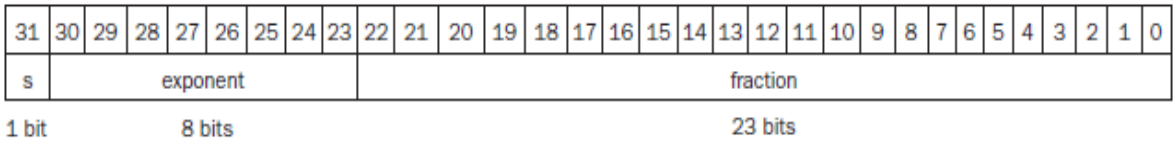
- A situation in which a positive exponent becomes too large to fit in the exponent field is known as overflow.

Underflow:

- A situation in which a negative exponent becomes too large to fit in the exponent field is known as underflow.

18. Give the representation of single precision floating point number.

A floating-point value represented in a single 32-bit word. Where s is the sign of the floating-point number (1 meaning negative), exponent is the value of the 8-bit exponent field (including the sign of the exponent), and fraction is the 23-bit number.

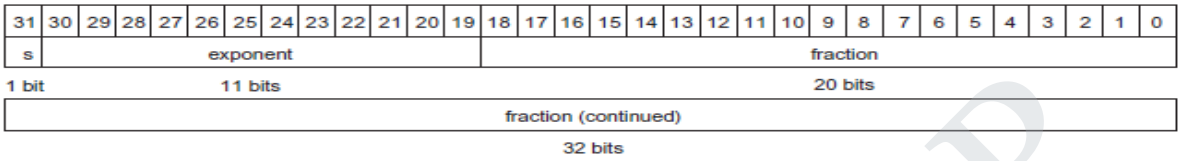


In general, floating-point numbers are of the form

$$(-1)^S \times F \times 2^E$$

19. Give the representation of double precision floating point number.

A floating-point value represented in two 32-bit words. Where s is still the sign of the number, exponent is the value of the 11-bit exponent field, and fraction is the 52-bit number in the fraction field.



20. What are the floating point instructions in MIPS?

- Floating-point addition, single (add.s) and addition, double (add.d)
- Floating-point subtraction, single (sub.s) and subtraction, double (sub.d)
- Floating-point multiplication, single (mul.s) and multiplication, double (mul.d)
- Floating-point division, single (div.s) and division, double (div.d)
- Floating-point comparison, single (c.x.s) and comparison, double (c.x.d),where x may be equal (eq), not equal (neq), less than (lt), less than or equal(le), greater than (gt), or greater than or equal (ge)
- Floating-point branch, true (bc1t) and branch, false (bc1f)

21. Define Guard and Round bit

Guard Bit:

- Extra bits kept on the right during intermediate calculations of floating point numbers is called guard bit and it used to improve rounding accuracy.

Round:

- Method to make the intermediate floating-point result fit the floating-point format.
- The goal is typically to find the nearest number that can be represented in the format.

22. Define Sticky bit

Sticky Bit:

- A bit used in rounding in addition to guard and round that is set whenever there are nonzero bits to the right of the round bit.

23. Write the IEEE 754 floating point format.

- IEEE 754 makes the leading 1-bit of normalized binary numbers implicit.
- Hence, the number is actually 24 bits long in single precision (implied 1 and a 23-bit fraction), and 53 bits long in double precision (1+52).

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

Single precision		Double precision		Object represented
Exponent	Fraction	Exponent	Fraction	
0	0	0	0	0
0	Nonzero	0	Nonzero	± denormalized number
1–254	Anything	1–2046	Anything	± floating-point number
255	0	2047	0	± infinity
255	Nonzero	2047	Nonzero	NaN (Not a Number)

24. State the rule for floating point addition.

- Choose the number with the smaller exponent and shift its mantissa right a number of steps equal to the difference in exponents.
- Set the exponent of the result equal to the larger exponent.
- Perform the addition on the mantissa and determine the sign of the result.
- Normalize the resulting value if necessary.

25. State the representation of double precision floating point number. (Nov 2015)

Double precision representation contains 11 bits, excess -1023 exponent E' which has the range $1 \leq E' \leq 2046$ for normal values. This means that the actual exponent E is in range $-1022 \leq E \leq 1023$. The 53 bit mantissa provides a precision equivalent to about 16 decimal digits.

26. What is guard bit? What are the ways to truncate the guard bits? (Nov 2016)

Although the mantissa of initial operands is limited to 24 bits, it is important to retain extra bits, called as guard bits. There are several ways to truncate the guard bits: Chopping, VonNeumann rounding, Rounding.

27. What is overflow and underflow case in single precision?

Underflow-The normalized representation requires an exponent less than -126
Overflow-The normalized representation requires an exponent greater than -126

28. Why floating point number is more difficult to represent and process than integer?

In floating point numbers we have to represent any number in three fields sign, exponent and mantissa. The IEEE 754 standard gives the format for these fields and according to format the numbers are to be represented. In case of any process the mantissa and exponent are considered separately.

29. When can you say that a number is normalized?

When the decimal point is placed to the right of the first (nonzero) significant digit the number is said to be normalized.

30. Write the rules for add/sub operation on floating point numbers? (May 2017)

- Choose the number with the smaller exponent and shift its mantissa right a number of steps equal to the difference in exponents.
- Set the exponent of the result equal to the larger exponent
- Perform addition / subtraction on the mantissa and determine the sign of the result
- Normalize the resulting value, if necessary

31. Define Truncation.

To retain maximum accuracy, all extra bits during operation (called guard bits) are kept (e.g., multiplication). If we assume n=3 bits are used in final representation of a number, n=3extra guard bits are kept during operation. By the end of the operation, the resulting 2n=6 bits need to be truncated to n=3 bits by one of the three methods.

32. Define Chopping.

There are several ways to truncate. The simplest way is to remove the guard bits and make no changes in the retained bits. This is called Chopping. Chopping discards the least significant bits

and retains the 24 most significant digits. This is easy to implement, and biased, since all values are rounded to-wards a lower mantissa value. The maximum rounding error is 0≤e<+1 LSB.

33. What is excess-127 format?

Instead of the signed exponent E, the value actually stored in the exponent field is and unsigned integer E¢=E+127.This format is called excess-127.

34. What do mean by Subword Parallelism?(May 2015, May 2016,)

Subword parallelism is a technique that enables the full use of word-oriented data paths when dealing with lower precision data. It is a form of low-cost, small-scale SIMD parallelism.

PART-B

1. Briefly Explain Carry Look-ahead adder. (Nov 2014)
2. Explain the Multiplication algorithm in detail with diagram and multiply the following pair of signed 2's complements numbers: A = 010111, B = 101100.
3. Explain the Booth's multiplication algorithm with suitable example (May 2016) (Nov 2016)
4. Multiply the following pair of signed nos. using Booth's bit-pair recoding of the multiplier. A=+13 (Multiplicand) and B= -6 (Multiplier). (Nov 2014)
5. i) Perform X + Y and Y-X using 2's complements for given two binary numbers X = 0000 1011 1110 1111 and Y = 1111 0010 1001 1101.

   ii)Multiply the following signed 2's complement numbers using the Booth's algorithm. A= 001110 and B=111001 where A is multiplicand and B is multiplier. (May 2018)
6. Demonstrate multiplication of two binary numbers with an example. Design an arithmetic element to perform the multiplication.(May 2017)
7. Discuss in detail about division algorithm in detail with diagram and examples. (Nov 2015, Nov 2016,Nov 2017) (May 2018)
8. Divide (12)10 by (3)10 using the Restoring and Non- restoring division algorithm with step by step intermediate results and explain. (Nov 2014)(May 2017)
9. Discuss in detail about division algorithm in detail with diagram and calculate the division of A and B. A = 3.264 X 103 B= 6.52 X 102
10. Design an arithmetic element to perform the basic floating point operations.(May 2017)

    What is meant by subword parallelism? Explain.
11. Add the numbers (0.75)10 and (-0.275)10 in binary using the Floating point addition algorithm. (May 2018)
12. Add the numbers (0.5)10 and (0.4375)10 using the floating point addition.(Nov 2017)
13. Draw and explain the block diagram of floating point adder – subtractor unit with an example. (May 2016)
14. Explain in detail about floating point addition and add the numbers 0.510 and -0.437510 using binary floating point addition algorithm.
15. Explain in detail about floating point multiplication and multiply 1.1010X 1010and 9.200X10-5 using binary Floating point multiplication.
16. Show the IEEE 754 binary representation of the number -0.7510 in single and double precision.

1. **Define datapath. [May-14, 18, Dec-16]**
   Datapath is a unit used to operate on or hold data within a processor. Its elements include the instruction and data memories, the register file, the ALU, and adders.
2. **What is the use of PC register?**
   The program counter (PC) that contains the code and fetches the instruction from that memory. The register containing the address of the instruction in the program being executed is called program counter.
3. **Define register file.**
   All general purpose registers are combined into a single block called the register file.
4. **What are the two types of datapath element?**
   1. **Combinational Elements**:
   The ALU is an example of a combinational element. Given a set of inputs, it always produces the same output because it has no internal storage.
   2. **State Elements:**
   An element contains state if it has some internal storage. A state element has at least two inputs and one output.
5. **What are the two state elements needed to store and access an instruction?**
   1. Clocking Methodology: It defines when signals can be read and when they can be written.
   2. Edge Triggered Clocking Methodology.
6. **Define asserted and deasserted signals.**
   Asserted: The signal is logically high or true.
   Deasserted: The signal is logically low or false.
7. **Define Sign Extend.**
   To increase the size of a data item by replicating the high-order sign bit of the original data item in the high order bits of the larger, destination data item is called sign extend.

b. Sign extension unit

8. **What is mean by branch?**
   A type of branch where the instruction immediately following the branch is always executed, independent of whether the branch condition is true or false.
9. **Differentiate branch taken from branch not taken.**
   Branch taken
   A branch where the branch condition is satisfied and the program counter (PC) becomes the branch target. All unconditional jumps are taken branches.
   Branch not taken or (untaken branch)
   A branch where the branch condition is false and the program counter (PC) becomes the address of the instruction that sequentially follows the branch.
10. **What do you meant by delayed branch?**
    An instruction that always executes after the branch is called delayed branch.
11. **Write the instruction format for the jump instruction.**

Field	000010	address
Bit positions	31:26	25:0

12. What is meant by branch prediction? Why it is needed? [May'12]

A method of 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. It is needed to reduce the branch penalty.

13. What is mean by branch history table / branch prediction / branch target buffer? [May'15]

A branch prediction buffer or **branch history table** 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.

14. What is mean by dynamic branch prediction?

Prediction of branches at runtime using runtime information is called dynamic branch prediction. If a branch was taken the last time this instruction was executed and, if so, to begin fetching new instructions from the same place as the last time. This technique is called **dynamic branch prediction.**

15. What is called ALUOP?

To generate the 4-bit ALU control input using a small control unit that has as inputs the function field of the instruction and a 2-bit control field, which we call ALUOp. ALUOp indicates whether the operation to be performed should be add (00) for loads and stores, subtract (01) for beq, or determined by the operation encoded in the funct field (10). The output of the ALU control unit is a 4-bit signal that directly controls the ALU by generating one of the 4-bit combinations.

16. Define TruthTable.

It is a representation of a logical operation by listing all the values of the inputs and then in each case showing what the resulting outputs should be.

17. Define don't care term.

An element of a logical function in which the output does not depend on the values of all the inputs. Don't-care terms may be specified in different ways.

18. What is opcode?

The field that denotes the operation and format of an instruction.

19. Name the control signals required to perform arithmetic operation. [May'17]

Control signals RegDst, RegWrite and ALUop1 are required to perform arithmetic operation.

20. What is meant by single clock cycle?

An implementation in which an instruction is executed in one clock cycle is called single clock cycle.

21. Why a single-cycle implementation is not used today?

1. Inefficient
2. Overall performance of a single cycle implementation is poor since the clock cycle is too long
3. It is not suitable for floating point operation.
4. It is not suitable for the instruction set with more complex instructions.
5. It also violate the common case fast idea.

22. What is pipeline stall? [Dec'16]

Pipeline stall also called bubble. It is a delay in execution of an instruction in an instruction pipeline in order to resolve a hazard.

23. Define pipelining.

Pipelining is a technique of decomposing a sequential process into sub operations with each sub process being executed in a special dedicated segment that operates concurrently with all other

segments. An implementation technique in which multiple instructions are overlapped in execution is called pipeline.

24. Mention the various types of pipelining.

1. Instruction pipelining 2. Arithmetic pipelining

25. What is instruction pipelining?

Performing fetch, decode and execute cycles for several instruction simultaneously to reduce overall processing time is called instruction pipelining.

26. List the five stages in the instruction pipelining. [Dec'17]

The different pipelining stages are,

1. Fetch - Fetch instruction from memory.
2. Decode - Read registers while decoding the instruction. The regular format of MIPS instructions allow reading and decoding to occur simultaneously.
3. Execute - Execute the operation or calculate an address.
4. Access - Access an operand in data memory.
5. Write - Write the result into a register.

27. How addressing modes affect the instruction pipelining?

Due to address dependency where operand addresses cannot be calculated without available information needed by addressing mode. Hence operand access is delayed degrading the performance of pipeline.

28. What would be the effect, if we increase the number of pipelining stages?

As the number of pipeline stages increase, the probability of the pipeline being stalled, also increases because more instructions are being executed concurrently and the branch penalties may become more significant.

29. What are the advantages of pipelining? [May'16]

1. Increasing Instruction throughput and the instruction cycle time of the processor is reduced.
2. Increasing in pipeline stages increase number of instructions that can be processed at once which reduces delay between completed instructions.

30. Write the formula for calculating time between instructions in a pipelined processor.

$$\text{Time between instructions}_{\text{pipelined}} = \frac{\text{Time between instruction}_{\text{nonpipelined}}}{\text{Number of pipe stages}}$$

31. What is the ideal speed-up expected in a pipelined architecture with 'n' stages? Justify your answer. [Mar'07]

$$S(m) = \frac{T(1)}{T(m)}$$

Where T(m) is the execution time for some program on a m-stage pipeline and T(1) is the execution time for the same program on a similar nonpipelined processor.

32. What are hazards? Write its types. [May'13,17]

The condition that makes the pipeline to stall is called Hazards. The idle period in the pipeline execution is called Stall or Bubble.

Types of hazards:

1. **Structural Hazard:** The hazard due to insufficient resources and not possible to overlap the operation is called structural hazard. [May'14]
2. **Data Hazard:** When either the source or destination operands of an instruction are not available at the time expected in the pipeline or as a result pipeline is stalled called data hazard. [Dec'07, May'12,13]

3. **Control Hazard:** The hazard due to pipelining branch and other instructions that change the content of program counter is called Control hazard. [May'12,13]

33. What are the schemes to resolve data hazard? [May'12,13]

1. Operand forwarding(Hardware)
2. Reordering Code (software)
3. By using stall

34. What is meant by forwarding?

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 is called forwarding or bypassing.

35. What is meant by load-use data hazard?

A specific form of data hazard in which the data being loaded by a load instruction has not yet become available when it is needed by another instruction is called load-use data hazard.

36. What are the classification of data hazards? [May'17]

1. RAW- Read After Write Hazard
2. WAW- Write After Write Hazard
3. WAR- Write After Read Hazard.

37. What are the two schemes to resolve control hazard?

1. Branch prediction
2. Delayed branching

38. What are exceptions and interrupts?

Exceptions and **interrupts** events other than branches or jumps that change the normal flow of instruction execution.

Type of event	From where?	MIPS terminology
I/O device request	External	Interrupt
Invoke the operating system from user program	Internal	Exception
Arithmetic overflow	Internal	Exception
Using an undefined instruction	Internal	Exception
Hardware malfunctions	Either	Exception or interrupt

39. What are imprecise and precise interrupts? [Dec'14, May'16]

Imprecise interrupt

Imprecise interrupt also called imprecise exception. Interrupts or exceptions in pipelined computers that is not associated with the exact instruction that was the cause of the interrupt or exception.

Precise interrupt

Precise interrupt also called precise exception. An interrupt or exception that is always associated with the correct instruction in pipelined computers.

40. What are the two main methods used to communicate the reason for an exception?

- The first method used in the MIPS architecture is to include a **status register** (called the Cause register), which holds a field that indicates the reason for the exception.
- A second method is to use **vectored interrupts**. In a vectored interrupt, the address to which control is transferred is determined by the cause of the exception.

41. What are the correlating predictor and tournament predictor

Correlating predictor:

- A branch predictor that combines local behavior of a particular branch and global information about the behavior of some recent number of executed branches.

Tournament branch predictor

- A branch predictor with multiple predictions for each branch and a selection mechanism that chooses which predictor to enable for a given branch.

42. Define flush.

To discard instructions in a pipeline, usually due to an unexpected event is called flush.

43. Define nop.

An instruction that does no operation to change state.

44. State the advantages of using multiple levels of decoding.

1. It reduces the size of the main control unit
2. To increase the speed of the control unit.

45. What is control store?

The micro routines for all instructions in the instruction set of a computer are stored in a special memory called control store.

PART B

1. Draw and explain the function block diagram for the basic MIPS implementation with necessary multiplexers and control lines. [Dec'15]
2. Show how to build a datapath and explain the simple combine datapath for the MIPS architecture.
3. Design a simple datapath with control implementation scheme and explain in detail. [May'18]
4. Explain the basic concepts of pipelining. [May'12,16]
5. Explain basic operation of a four stage pipelining with a neat diagram. [May'09,13]
6. Explain the function of a six segment pipeline and draw a space diagram for a six segment pipeline showing the time it takes to process eight tasks. [May'07]
7. What is pipelining? Discuss about pipelined datapath and control. [May'16]
8. Discuss limitations of pipelining a processor's datapath. Suggest the methods to overcome them. [May'18]
9. Explain the different types of pipeline hazards with suitable examples. [Dec'17, May'15,16]
10. Describe the operand forwarding in a pipeline processor with a diagram. [May'17]
11. Explain the hazards caused by unconditional branching statements. [May'17]
12. Discuss the modified datapath to accommodate pipelined executions with a diagram. [May'17]
13. Explain how the instruction pipeline works. What are the various situations where an instruction pipeline can stall? Illustrate with an example. [Dec'15,16]
14. Why is branch prediction algorithm needed? Differentiate between the static and dynamic techniques. [Dec'16]
15. What is data hazard? How do you overcome it? What are its side effects? [May'14]
16. What is a control hazard? Explain the methods for dealing with the control hazards. [May'14]
17. Explain dynamic branch prediction in detail. [May'13]
18. Explain in detail how exceptions are handled in MIPS architecture? [May'15]

CS8491 Computer Architecture 2 Marks

1. **What is ILP? Why it is needed? [Dec-15,16, May-16,17]**

   Pipelining exploits the potential parallelism among instructions. This parallelism is called instruction-level parallelism (ILP). It is needed to achieve high performance.

2. **Define superscalar. [Dec-15]**

   An advanced pipelining technique that enables the processor to execute more than one instruction per clock cycle by selecting them during execution.

3. **Define parallel processing program.**

   A single program that runs on multiple processors simultaneously is called parallel processing program.

4. **Define data level parallelism.**

   Parallelism achieved by performing the same operation on independent data.

5. **Define reorder buffer.**

   The buffer that holds results in a dynamically scheduled processor until it is safe to store the results to memory or a register.

6. **Define out-of-order execution.**

   A situation in pipelined execution when an instruction blocked from executing does not cause the following instructions to wait.

7. **Define in-order commit.**

   A commit in which the results of pipelined execution are written to the programmer visible state in the same order that instructions are fetched.

8. **Define multiprocessor.**

   A computer system with at least two processors is multiprocessor.

9. **Define task level parallelism (TLP).**

   Task-level parallelism or process-level parallelism is utilizing multiple processors by running independent programs simultaneously.

10. **Define multicore microprocessor.**

    A microprocessor containing multiple processors (“cores”) in a single integrated circuit. Virtually all microprocessors today in desktops and servers are multicore.

11. **Define shared memory microprocessor (SMP).**

    A parallel processor with a single physical address space is called shared memory microprocessor.

12. **What are the difficulties to write parallel processing programs?**

    1. Scheduling.
    2. Partitioning the work into parallel pieces.
    3. Balancing the load evenly between tasks.
    4. Time to synchronize.
    5. Overhead for communication.

13. **State the Amdahl’s law? [Dec-14]**

    It states that the performance improvement to be gained from using some faster mode of execution is limited by the fraction of the time the faster mode can be used.

14. **Define strong scaling and weak scaling. [May-15, Dec-17]**

    Speedup achieved on a multiprocessor without increasing the size of the problem is called strong scaling.

    Speedup achieved on a multiprocessor while increasing the size of the problem proportionally to the increase in the number of processors is called weak scaling.

15. Define vector lane.

One or more vector functional units and a portion of the vector register file. Inspired by lanes on highways that increase traffic speed, multiple lanes execute vector operations simultaneously.

16. What is multithreading? [Dec-16]

A mechanism by which the instruction streams is divided into several smaller streams (threads) and can be executed in parallel is called multithreading.

17. Define hardware multithreading.

Increasing utilization of a processor by switching to another thread when one thread is stalled is called hardware multi threading.

18. Define thread.

A thread includes the program counter, the register state, and the stack. It is a lightweight process, whereas threads commonly share a single address space.

19. Define process.

A process includes one or more threads, the address space, and the operating system state. Hence, a process switch usually invokes the operating system, but not a thread switch.

20. Define coarse-grained multithreading.

A version of hardware multithreading that implies switching between threads only after significant events such as last-level cache miss.

21. Define fine- grained multithreading. [May-16]

A version of hardware multithreading that implies switching between threads after every instruction.

22. Define simultaneous multithreading (SMT).

A version of multithreading that lowers the cost of multithreading by utilizing the resources needed for multiple issue, dynamically scheduled micro architecture.

23. What do you mean by implicit and explicit multithreading? [May-17]

Implicit multithreading refers to the concurrent execution of multiple threads extracted from a single sequential program.

Explicit multithreading refers to the concurrent execution of instruction from different explicit threads, either by interleaving instructions from different threads on shared pipelines or by parallel execution on parallel pipelines.

24. Define uniform memory access (UMA). [April/May-15]

In this model, main memory is uniformly shared by all processors in multiprocessor systems and each processor has equal access time to shared memory.

25. Define non uniform memory access (NUMA). [April/May-15]

It depends on the location of the memory. Thus, all memory words are not accessed uniformly. All local memories form a global address space accessible by all processors.

26. Define synchronization.

The process of coordinating the behavior of two or more processes, which may be running on different processors, is called synchronization.

27. Define Symmetric & Asymmetric.

Symmetric:

All processors have equal access to all peripheral devices. All processors are identical.

Asymmetric:

One processor (master) executes the operating system other processors may be of different types and may be dedicated to special tasks.

28. Define lock.

A synchronization device that allows access to data to only one processor at a time is called lock.

29. What is message passing?

Communicating between multiple processors by explicitly sending and receiving information is called message passing.

30. Define cluster.

A set of computers connected over a local area network that functions as a single large multiprocessor.

31. What is a Warehouse – scale computer?

A warehouse-Scale computer is a cluster comprised of tens of thousands of servers.

32. What are the characteristics of WSC that are not shared with servers?

1. Ample Parallelism
2. Operational costs count
3. Scale and its opportunities and problems.

PART- B

1. What is Instruction level parallelism? Explain static issue and multiple issue processors. (13)
2. Discuss the challenges in parallel processing with necessary examples.(15) [Dec-14,17, May-17,18]
3. What percentage of the original computation can be sequential to achieve a speedup of 90 times faster with 100 processors? (8) [Dec-17]
4. We have to perform two sums. One is a sum of 20 scalar variables. And one is a matrix sum of a pair of two dimensional arrays, with dimension 20 by 20. Let us assume only the matrix sum is parallelizable. What speedup do we get with 10 versus 50 processors? Also calculate the speedups assuming the matrices grow to 40 by 40. (8)
5. Suppose you want to perform two sums. One is a sum of 10 scalar variables and one is a matrix sum of a pair of two dimensional arrays, with dimension 10 by 10. Let us assume only the matrix sum is parallelizable. What speedup do we get with 10 versus 40 processors? Also calculate the speedups assuming the matrices grow to 20 by 20. (7) [Dec-17]
6. Protein String Matching Code has 4 days execution time on current machine doing integer instruction in 20% of time, doing I/O in 35% of time and other operations in the remaining time. Which is better tradeoff among the following two proposals? First compiler optimization that reduces number of integer instructions by 25%. (Assume each integer instruction takes the same amount of time) Second Hardware optimization that reduces the latency of each I/O operation from 6µs to 5 µs. (2) [May-18]
7. To achieve the speed-up of 20.5 on the previous larger problem with 40 processors, we assumed the load was perfectly balanced. That is, each of the 40 processors had 2.5% of the work to do. Instead, show the impact on speed-up if one processor's load is higher than all the rest. Calculate at twice the load (5%) and five times the load (12.5%) for that hardest working processor. How well utilized are the rest of the processors?
8. To achieve the speed-up of 31.15 on the previous larger problem with 50 processors, we assumed the load was perfectly balanced. That is, each of the 50 processors had 2% of the work to do. In this

problem we have to calculate the impact on speedup if one processor's load is higher than all the rest. Calculate the impact on speedup if the hardest working processor's load is 4% and 10%. Also calculate the utilization of the rest of the processors?

9. Discuss about SISD, MIMD, SIMD, SPMD and VECTOR systems.(13) [May'15]
10. Explain in detail Flynn's classification of parallel hardware. (13) [Dec-15,16, May-16,18]
11. Explain Vector processor and Vector Lane with neat diagram. (13)
12. What is hardware multithreading? Compare and contrast Fine grained Multi-Threading and Coarse grained Multi-Threading. (13) [May-15]
13. Explain in detail about hardware multithreading. (13) [Dec-15, May-16]
14. Explain the four principle approaches to multithreading with necessary diagrams. (13) [May-17]
15. Discuss shared memory multiprocessor with a neat diagram. (13) [Dec-16]
16. Write a short note on multiprocessors. (8) [May-16, Dec-14]
17. Explain Multicore processors. (8)
18. Draw and explain the block diagram of GPU architecture and compare CPU and GPU. (13)
19. Draw and discuss about the Cluster Architecture and its types. (13)
20. Write an essay on Message Passing Multiprocessors. (8)
21. Explain cluster and other Message passing Multiprocessor. (8)

1. **Define hit and miss.**
   Hit: If the data requested by the processor appears in some block in the upper level, this is called a **hit**.
   Miss: If the data is not found in the upper level, the request is called a **miss.**

2. **Define hit rate and miss rate. [Dec-15]**
   Hit rate or Hit ratio: It is the fraction of memory accesses found in the upper level; it is often used as a measure of the performance of the memory hierarchy.
   Miss rate: Miss rate (1−hit rate) is the fraction of memory accesses not found in the upper level.

3. **Define block.**
   The minimum unit of information that can be either present or not present in the two-level hierarchy is called a **block or a line.**

4. **Define hit time & miss penalty.**
   Hit time: It is the time to access the upper level of the memory hierarchy, which includes the time needed to determine whether the access is a hit or a miss.
   Miss penalty: It is the time to replace a block in the upper level with the corresponding block from the lower level, plus the time to deliver this block to the processor.

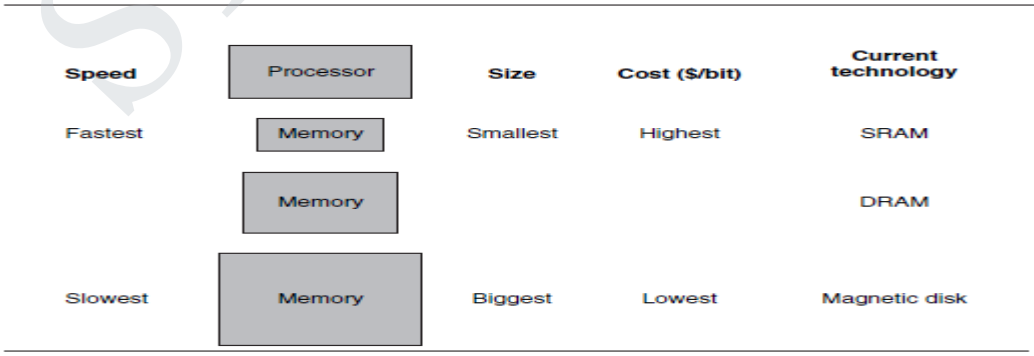
5. **What is called principles of locality? [May-08]**
   It states that programs access a relatively small portion of their address space at any instant of time.

6. **What are the temporal and spatial localities of references? [May-14]**
   Temporal locality: The principle stating that if a data location is referenced then it will be likely to be referenced again soon.
   Spatial locality: The locality principle stating that if a data location is referenced, data locations with nearby addresses will be likely to be referenced soon.

7. **Define DDRSDRAM. [Dec-11]**
   The fastest version is called **Double Data Rate (DDR) SDRAM.** The name means data transfers on both the rising and falling edge of the clock, thereby getting twice as much bandwidth as you might expect based on the clock rate and the data width.

8. **Define address leaving.**
   Sending an address to several banks permits them all to read or write simultaneously. For example, with four banks, there is just one access time and then accesses rotate between the four banks to supply four times the bandwidth. This rotating access scheme is called **address interleaving.**

9. **Write the structure of memory hierarchy. [May 16]**



10. **What are the various memory technologies?**
    1. SRAM Technology
    2. DRAM Technology
    3. Flash Memory
    4. Disk Memory

11. Differentiate SRAM from DRAM. [June-08]

Static RAM (SRAM)	Dynamic RAM (DRAM)
Information will be available as long as power is available	It retains data for few ms even in the absence of a power source based on the charge of capacitor
No refreshing is needed	Refreshing is needed
Less packaging density	High packaging density
More complex Hardware	Less complex hardware
More expensive	Less expensive
No random access	Random access is possible
Access time 10 ns	Access time 50 ns

12. What is flash memory?

Flash memory is a type of electrically erasable programmable read-only memory (EEPROM). EEPROM technologies use flash memory bits for writing purpose.

13. Define − Rotational Latency.

Once the head has reached the correct track, we must wait for the desired sector to rotate under the read/write head. This time is called the **rotational latency or rotational delay.**

14. Define track and sector.

Tracks: Each disk surface is divided into concentric circles, called **tracks.** There are typically tens of thousands of tracks per surface.

Sector: Each track is in turn divided into **sectors** that contain the information; each track may have thousands of sectors. Sectors are typically 512 to 4096 bytes in size.

15. Define seek time.

To access data, the operating system must direct the disk through a three-stage process. The first step is to position the head over the proper track. This operation is called a seek, and the time to move the head to the desired track is called the **seek time.**

16. What is direct-mapped cache?

A cache structure in which each memory location is mapped to exactly one location in the cache.

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

17. 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

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

and

$$\left\lfloor \frac{\text{Byte address}}{\text{Bytes per block}} \right\rfloor \times \text{Bytes per block} + (\text{Bytes per block} - 1)$$

Thus, with 16 bytes per block, byte address 1200 is block address

$$\left\lfloor \frac{1200}{6} \right\rfloor = 75$$

which maps to cache block number (75 modulo 64) = 11. In fact, this block maps all addresses between 1200 and 1215.

18. How many total bits are required for a direct-mapped cache with 16 KB of data and 4-word blocks, assuming a 32-bit address?

Solution:

We know that 16 KiB is 4096 (2^{12}) words. With a block size of 4 words (2^2), there are 1024 (2^{10}) blocks. Each block has 4 * 32 or 128 bits of data plus a tag, which is (32-10-2-2) bits, plus a valid bit. Thus, the total cache size is

$$2^{10} \times (4 \times 32 + (32 - 10 - 2 - 2) + 1) = 2^{10} \times 147 = 147 \text{ Kibibits}$$

or 18.4 KiB for a 16 KiB cache. For this cache, the total number of bits in the cache is about 1.15 times as many as needed just for the storage of the data.

19. What are the techniques to improve the cache performance?

1. By reducing the miss rate
2. By reducing the miss penalty.

20. What are the writing strategies in cache memory?

1. Inconsistent
2. Write through
3. Write buffer
4. Write back
5. Split cache

21. Define inconsistent.

After the write into the cache, memory would have a different value from that in the cache. In such a case, the cache and memory are said to be **inconsistent.**

22. Define write through.

The simplest way to keep the main memory and the cache consistent is always to write the data into both the memory and the cache. This scheme is called **write-through.**

23. Define write buffer.

A write buffer stores the data while it is waiting to be written to memory. After writing the data into the cache and into the write buffer, the processor can continue execution.

24. Define write back.

In a write-back scheme, when a write occurs, the new value is written only to the block in the cache. The modified block is written to the lower level of the hierarchy when it is replaced.

25. Define split cache.

A scheme in which a level of the memory hierarchy is composed of two independent caches that operate in parallel with each other, with one handling instructions and one handling data.

26. What are the steps to be taken in an instruction cache miss?

1. Send the original PC value (current PC – 4) to the memory.
2. Instruct main memory to perform a read and wait for the memory to complete its access.
3. Write the cache entry, putting the data from memory in the data portion of the entry, writing the upper bits of the address (from the ALU) into the tag field, and turning the valid bit on.
4. Restart the instruction execution at the first step, which will refetch the instruction, this time finding it in the cache.

27. Define – AMAT

Average memory access time.

AMAT=Time for a hit + Miss rate × Miss penalty

28. What are the various block placement schemes in cache memory?

1. FIFO 2. LRU 3.MFU

29. Define page fault.

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

30. Define page table and page table register.

It is the table containing the virtual to physical address translations in a virtual memory system. The table, which is stored in memory, is typically indexed by the virtual page number; each entry in the table contains the physical page number for that virtual page if the page is currently in memory.

31. Define page table register.

To indicate the location of the page table in memory, the hardware includes a register that points to the start of the page table; we call this the **page table register**.

32. Define reference and dirty bit.

Reference Bit: If we get a hit, the physical page number is used to form the address, and the corresponding reference bit is turned on.

Dirty Bit: If the processor is performing a write, the dirty bit is also turned on. If a miss in the TLB occurs, we must determine whether it is a page fault or purely a TLB miss.

33. Define true page fault.

If the page is not present in memory, then the TLB miss indicates a true page fault. In this case, the processor invokes the operating system using an exception. Because the TLB has many fewer entries than the number of pages in main memory, TLB misses will be much more frequent than true page faults.

34. Define cycle stealing and burst mode.

Cycle Stealing: DMA Controller "steals" memory cycles from processor, though processor originates most memory access.

Block or Burst mode: The DMA controller may give exclusive access to the main memory to transfer a block of data without interruption.

35. Define bus arbitration and write their types.

Bus Arbitration: Process by which the next device to become master is selected.

Types of bus arbitration:

1. Centralized and Distributed Arbitration
2. Distributed Arbitration

36. What are the instructions executed by IOP?

1. Data transfer instructions
2. Branch instructions
3. I/O device control instructions

37. Define polling.

Polling is the simplest way for an I/O device to communicate with the processor.

38. Define – TLB.

Translation-lookaside buffer (TLB) a cache that keeps track of recently used address mappings to try to avoid an access to the page table.

39. What is meant by virtual memory?

It is a technique that uses main memory as a "cache" for secondary storage.

40. What is meant by address mapping? [June-08]

It is also called address translation. The process by which a virtual address is mapped to an address used to access memory.

41. What will be the width of address and data buses for a 512 K * 8 memory chip? [Dec-07]

The widths of address and data buses are 19 and 8 respectively.

42. What is the need to implement memory as a hierarchy? [May-15]

Computer memory should be fast, large and inexpensive. Unfortunately it is impossible to meet all the three of these requirements using one type of memory. Hence it is necessary to implement memory as a hierarchy.

43. What are the different types of buses?

1. Synchronous Bus
2. Asynchronous Bus

44. List any four features of USB.

1. Simple connectivity
2. Plug-and Play
3. Low cost
4. Flexibility

45. What is called a Hub and Root Hub?

Each node of the tree has a device called a Hub. Which act as a an intermediate control point between the host and the I/O device.

At the root of a tree, a root hub connects the entire tree to the host computer.

PART- B

1. Explain in detail about memory Technologies. [May-15] (8)
2. Explain in detail about memory Hierarchy with neat diagram. [Dec-14] (8)
3. Describe the basic operations of cache in detail with diagram and discuss the various mapping schemes used in cache design with example. [Dec-15,16,18] (16)
4. 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. (8)
5. Discuss the methods used to measure and improve the performance of the cache. (8)
6. Explain the virtual memory address translation and TLB with necessary diagram. [May-15,16,17] (16)
7. Draw the typical block diagram of a DMA controller and explain how it is used for direct data transfer between memory and peripherals. [Dec-16,18] (16)
8. Explain in detail about interrupts with diagram. [May-13,18] (16)
9. Describe in detail about programmed Input/Output with neat diagram. (10)
10. Draw and explain the timing diagram for Synchronous and Asynchronous Bus data transfer. [May-08] (8)
11. Explain in detail about the bus arbitration techniques. [May-17] (8)
12. Explain the function of a typical interface circuits. Also compare between serial and parallel interface. [May-11] (13)
13. Write short notes on USB. [May-13] (8)