

UNIT I BASIC STRUCTURE OF A COMPUTER SYSTEM

Functional Units – Basic Operational Concepts – Performance – Instructions: Language of the Computer – Operations, Operands – Instruction representation – Logical operations – decision making – MIPS Addressing.

UNIT-I/ PART-A

1.	<p>What are the functional units present in a computer? (Apr 2017, Nov 2017)</p> <p>A computer consists of five functionally independent main parts Input, Memory, Arithmetic and Logic unit (ALU), Output and Control unit.</p>
2.	<p>Define Memory Unit.</p> <p>It is the place to store programs and data. It is basically of two types</p> <ol style="list-style-type: none"> 1. Primary memory 2. Secondary memory <p>Primary memory, also known as the main memory, is the area in a computer which stores data and information for fast access. Semiconductor chips are the principle technology used for primary memory. It's a memory which is used to store frequently used programs which can be directly accessed by the processing unit for further processing. It's a volatile memory meaning the data is stored temporarily and is liable to change or lose in case of power failure.</p> <p>Eg: internal memory such as internal storage devices.</p> <p>Secondary memory is the external memory of the computer which can be used to store data and information on a long-term basis. It's a non-volatile memory which means data stays intact even if the computer is turned off. Data cannot be directly processed by the processing unit in secondary memory; in fact, it is first transferred into the main memory and then it's transferred back to the processing unit.</p> <p>Eg: hard drives, floppy disks, magnetic tapes, USB flash drives, CDs, DVDs, etc.</p>
3.	<p>State Response time and Throughput.</p> <p>Response time – the time between the start and completion of a task also referred to as execution time.</p> <p>Datacenter managers are often interested in increasing throughput or bandwidth – the total amount of work done in a given time</p>
4.	<p>Define multiprocessing.</p> <p>Multiprocessing is the use of two or more central processing units (CPUs) within a single computer system. The term also refers to the ability of a system to support more than one processor and/or the ability to allocate tasks between them.</p>
5.	<p>Differentiate super computer and mainframe computer.</p> <p>A computer with high computational speed, very large memory and parallel structured hardware is known as a super computer. EX: CDC 6600.</p> <p>Mainframe computer is the large computer system containing thousands of IC's. It is a room-sized machine placed in special computer centers and not directly accessible to average users. It serves as a central computing facility for an organization such as university, factory or bank.</p>
6.	<p>Differentiate between minicomputer and microcomputer.</p> <p>Minicomputer is a small and low cost computer which are characterized by Short word size i.e. CPU word sizes of 8 or 16 bits. They have limited hardware and software facilities.</p> <p>Microcomputer is a smaller, slower and cheaper computer packing all the electronics of the computer in to a handful of IC's, including CPU and memory and IO chips.</p>
7.	<p>What is instruction register? (Nov 2016)</p> <p>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.</p>
8.	<p>What is program counter?</p> <p>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.</p>

9.	<p>What is processor time?</p> <p>The sum of the periods during which the processor is active is called the processor time. It doesn't count I/O or time spent running other programs. It can be broken up into system time, and user time.</p> $\text{CPU time} = N_{\text{cycles}} * t_{\text{clock}} = N_{\text{cycles}} / f_{\text{clock}}$
10.	<p>What are clock and clock cycles?</p> <p>The timing signals that control the processor circuits are called as clock. The clock cycles defines the regular time intervals.</p> $\text{Clock Cycles} = \frac{\text{seconds}}{\text{program}} = \frac{\text{cycles}}{\text{program}} \times \frac{\text{seconds}}{\text{cycle}}$
11.	<p>What is superscalar execution?</p> <p>The multiple functional units are used to create parallel paths through which different instructions can be executed in parallel. So it is possible to start the execution of several instructions in every clock cycle. This mode of operation is called superscalar execution.</p>
12.	<p>What is RISC and CISC?</p> <p>Reduced Instruction Set Computers (RISC) is a microprocessor that is designed to perform a smaller number of types of computer instructions so that it can operate at a higher speed (perform more millions of instructions per second, or MIPS).</p> <p>Complex Instruction Set Computers (CISC) is a processor design where single instructions can execute several low-level operations (such as a load from memory, an arithmetic operation, and a memory store) or are capable of multi-step operations or addressing modes within single instructions.</p>
13.	<p>List out the methods used to improve system performance?</p> <p>The methods used to improve system performance are</p> <ul style="list-style-type: none"> • Processor clock • Basic Performance Equation • Pipelining • Clock rate • Instruction set • Compiler
14.	<p>What are the addressing modes and its various types? (Nov 2017)</p> <p>The different ways in which the location of an 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.</p>
15.	<p>Define register mode addressing.</p> <p>In register mode addressing, the name of the register is used to specify the operand. Eg. Add \$s3, \$s5,\$s6. Advantage: Only a small address field is needed in the instruction and no memory is referenced. Disadvantage: Address space is very limited.</p>
16.	<p>Define immediate mode addressing.</p> <p>In immediate mode addressing, the operand is given explicitly in the instruction. Eg. Add \$s0, \$s1,20. Advantage: No memory reference other than the instruction fetch is required to obtain the operand. Disadvantage: The size of the number is restricted to the size of the address field</p>

17.	Define Base or Displacement mode addressing. In base or displacement mode addressing, the operand is in a memory location whose address is the sum of a register and a constant in the instruction. Eg. lw \$t0,32(\$s3).																																
18.	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																																
19.	State Amdahl's Law.(Nov 2014) Amdahl's Law tells us the improvement expected from specific enhancements. The performance improvement or speedup due to improvement is calculated as follows Speedup= Execution time before improvement/ Execution time after improvement																																
20.	Define Little Endian arrangement.(Nov 2014) Little-endian describes the order in which a sequence of bytes is stored in computer memory. Little-endian is an order in which the "little end" (least significant value in the sequence) is stored first. For example, in a little-endian computer, the two bytes required for the hexadecimal number 4F52 would be stored as 524F (52 at address 1000, 4F at 1001).																																
21.	Distinguish pipelining from parallelism. (May 2015) Pipelining is a method of increasing system performance and throughput. It takes advantage of the inherent parallelism in instructions. Instructions are divided into 5 stages: IF, ID, EX, EME, WB. Parallelism means using more hardware for the executing the desired task. In Parallel computing more than one processor are running in parallel. It increases performance but the area also increases.																																
22.	What is Instruction set architecture? (Nov 2015) The ISA serves as the boundary between the software and hardware. It is the structure of a computer that a machine language programmer (or a compiler) must understand to write a correct (timing independent) program for that machine. It also specifies a processor's functionality • what operations it supports • what storage mechanisms it has & how they are accessed • how the programmer/compiler communicates programs to processor.																																
23.	How to represent instruction in a Computer System? (May 2016) A form of representation of an instruction composed of fields of binary numbers. Binary representation used for communication within a computer system. <table><tr><td>Condition</td><td>F(Format)</td><td>I(Immediate)</td><td>Opcode</td><td>S(Cond branch)</td><td>Rn</td><td>Rd</td><td>Operand2</td></tr><tr><td>4bits</td><td>2bits</td><td>1bit</td><td>4bits</td><td>1bit</td><td>4bits</td><td>4bits</td><td>12bits</td></tr></table> ADD r5,r1,r2(Decimal and Binary Representation) <table><tr><td>-</td><td>-</td><td>-</td><td>4</td><td>0</td><td>1</td><td>5</td><td>2</td></tr><tr><td>-</td><td>-</td><td>-</td><td>0100</td><td>0</td><td>0001</td><td>0101</td><td>000000000010</td></tr></table>	Condition	F(Format)	I(Immediate)	Opcode	S(Cond branch)	Rn	Rd	Operand2	4bits	2bits	1bit	4bits	1bit	4bits	4bits	12bits	-	-	-	4	0	1	5	2	-	-	-	0100	0	0001	0101	000000000010
Condition	F(Format)	I(Immediate)	Opcode	S(Cond branch)	Rn	Rd	Operand2																										
4bits	2bits	1bit	4bits	1bit	4bits	4bits	12bits																										
-	-	-	4	0	1	5	2																										
-	-	-	0100	0	0001	0101	000000000010																										
24.	State the need for indirect addressing mode. Give an example. (Apr 2017) Indirect addressing is a scheme in which the address specifies which memory word or register contains not the operand but the address of the operand. For example: 1) LOAD R1, @100 Load the content of memory address stored at memory address 100 to the register R1. <div><div>R1</div><div>M[100]</div><div>M[200]</div><div>-20010</div><div>LOAD R1,@100</div><div>1020010</div></div> 2) LOAD R1, @R2 Load the content of the memory address stored at register R2 to register R1.																																

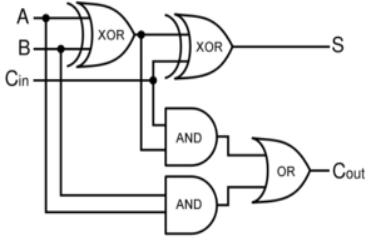
25.	How CPU execution time for a program is calculated? (Nov 2015) (Nov 2016) CPU execution time for a program is given by the formula CPU Execution time=Instruction Count * Clock cycles per instruction * Clock cycle time.																
26.	Distinguish between auto increment and auto decrement addressing mode? (May 2016)																
	Auto Increment	Auto Decrement															
	Add R1, (R2)+	Add R1,-(R2)															
	R1<-R1+Mem[R2] R2 <- R2 + d	R2 <-R2-d R1 <- R1 + Mem[R2]															
	Useful for stepping through arrays in a loop. R2-start of array d - size of an element	Same as autoincrement.															
	Used to implement a stack as push and pop	Used to implement a stack as push and pop															
27.	Write the equation for the dynamic power required per transistor. (May 2018) Transient power consumption can be calculated using equation 4. $P_T = C_{pd} \times V_{CC}^2 \times f_i \times N_{SW}$ Where: P_T = transient power consumption V_{CC}^2 = supply voltage f_i = input signal frequency N_{SW} = number of bits switching C_{pd} = dynamic power-dissipation capacitance																
28.	Classify the instructions based on the operations they perform and give one example to each category. (May 2018) The instruction sets can be differentiated by 1. Operand storage in the CPU 2. Number of explicit operands per instruction 3. Operand location 4. Operations 5. Type and size of operands A stack (the operands are implicitly on top of the stack) An accumulator (one operand is implicitly the accumulator) A set of registers (all operands are explicit either registers or memory locations) Example: <table><tr><td>STACK</td><td>ACCUMULATOR</td><td>REGISTER</td></tr><tr><td>PUSH A</td><td>Load A</td><td>Load R1,A</td></tr><tr><td>PUSH B</td><td>ADD B</td><td>ADD R1,B</td></tr><tr><td>ADD</td><td>Store C</td><td>Store C,R1</td></tr><tr><td>POP C</td><td></td><td></td></tr></table>		STACK	ACCUMULATOR	REGISTER	PUSH A	Load A	Load R1,A	PUSH B	ADD B	ADD R1,B	ADD	Store C	Store C,R1	POP C		
STACK	ACCUMULATOR	REGISTER															
PUSH A	Load A	Load R1,A															
PUSH B	ADD B	ADD R1,B															
ADD	Store C	Store C,R1															
POP C																	
UNIT-I/ PART-B																	
1.	Explain the important measure of the performance of a computer and derive the basic performance equation. (May 2017)																
2.	Explain various instruction formats and illustrate the same with an example. (Nov 2017)																
3.	What is an addressing mode? What is the need for addressing in a computer system? Explain the various addressing modes with suitable examples. (May 2015, Nov 2015 , May 2016, Nov 2016)																
4.	Discuss about the various techniques to represent instructions in a computer system(May 2015)																
5.	What are the various logical operations and explain the instructions supporting the logical operations.																

6.	What are the various control operations and explain the instructions supporting the control operations.																												
7.	Explain with an example about the operations and operands of the computer Hardware?(Nov 2017)																												
8.	<p>(i) Assume a two address format specified as source, destination. Examine the following sequence of instructions and explain the addressing modes used and the operation done in every instruction.</p> <p>Move(R5)+, R0 Add (R5)+, R0 Move R0, (R5) Move 16(R5), R3 Add #40, R5</p> <p>(ii) Consider the computer with three instruction classes and CPI measurements as given below and Instruction counts for each instruction class for the same program from two different compilers are given. Assume that the computer's clock rate is 4GHZ. Which Code sequence will execute faster according to execution time?</p> <table><tr><td>Code from</td><td colspan="3">CPI for this Instruction Class</td></tr><tr><td></td><td>A</td><td>B</td><td>C</td></tr><tr><td>CPI</td><td>1</td><td>2</td><td>3</td></tr></table> <table><tr><td>Code from</td><td colspan="3">Instruction Count for each Class</td></tr><tr><td></td><td>A</td><td>B</td><td>C</td></tr><tr><td>Compiler 1</td><td>2</td><td>1</td><td>2</td></tr><tr><td>Compiler 2</td><td>4</td><td>1</td><td>1</td></tr></table> <p style="text-align: right;">(Nov 2014)</p>	Code from	CPI for this Instruction Class				A	B	C	CPI	1	2	3	Code from	Instruction Count for each Class				A	B	C	Compiler 1	2	1	2	Compiler 2	4	1	1
Code from	CPI for this Instruction Class																												
	A	B	C																										
CPI	1	2	3																										
Code from	Instruction Count for each Class																												
	A	B	C																										
Compiler 1	2	1	2																										
Compiler 2	4	1	1																										
9.	<p>(i) Explain in detail the various components of computer system with neat diagram. (Nov 2014, Nov 2015, May 2016, Nov 2016)</p> <p>(ii) State the CPU performance equation and discuss the factors that affect performance(Nov 2014)</p>																												
10.	Explain direct, immediate, relative and indexed addressing modes with example. (May 2017)																												
11.	<p>(i) Suppose you want to achieve a speed-up of 90 times faster with 100 processors. What percentage of the original computation can be sequential?(Nov 2017)</p> <p>(ii) 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 dimensions 10 by 10.For now let's assume only the matrix sum is parallelizable. What speed-up do you get with 10 versus 40 processors? Next, calculate the speed-ups assuming the matrices grow to 20 by 20. (Nov 2017)</p>																												
12.	<p>a) (i) Consider three different processors P1, P2 and P3 executing the same instruction set. P1 has 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.</p> <p>a) Which processor has the highest performance expresses in instructions per second?</p> <p>b) If the processors each execute a program in 10 seconds. Find the number of cycles and the number of instructions in each processor.</p> <p>(ii) Explain in detail the components of a computer system.</p> <p>b) (i) Translate the following C code to MIPS assemble code. Use a minimum number of instructions. Assume that i and k correspond to registers \$s3 and \$s5 and the base of the array save in \$s6. What is the MIPS assembly code corresponding to this C segment?</p> <p>While (save[i] ==k) i+=1;</p> <p>(ii)What is an addressing mode in a computer? Classify MIPS addressing modes and give one example instruction to each category. (May 2018)</p>																												

UNIT II ARITHMETIC FOR COMPUTERS

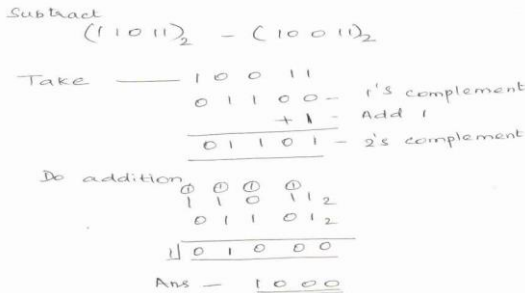
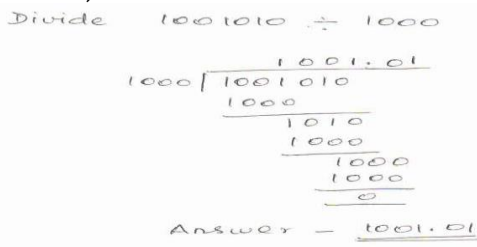
Addition and Subtraction – Multiplication – Division – Floating Point Representation – Floating Point Operations – Subword Parallelism

UNIT-II/ PART-A

1.	<p>Define Full Adder (FA) with logic diagram.</p> <p>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 C_{in}; A and B are the operands, and C_{in} 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.</p> 
2.	<p>State the rule for floating point addition.</p> <p>Choose the number with the smaller exponent and shift its mantissa right a number of steps equal to the difference in exponents.</p> <p>Set the exponent of the result equal to the larger exponent.</p> <p>Perform the addition on the mantissa and determine the sign of the result.</p> <p>Normalize the resulting value if necessary.</p>
3.	<p>State the representation of double precision floating point number. (Nov 2015)</p> <p>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.</p>
4.	<p>What is guard bit? What are the ways to truncate the guard bits? (Nov 2016)</p> <p>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.</p>
5.	<p>What is overflow and underflow case in single precision?</p> <p>Underflow-The normalized representation requires an exponent less than -126</p> <p>Overflow-The normalized representation requires an exponent greater than -126</p>
6.	<p>Why floating point number is more difficult to represent and process than integer?</p> <p>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.</p>
7.	<p>Define Booth Algorithm.</p> <p>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.</p>
8.	<p>What is arithmetic overflow? (Nov 2016)</p> <p>In a computer, the condition that occurs when a calculation produces a result that is greater in magnitude than which a given register or storage location can store or represent. In a computer, the amount by which a calculated value is greater in magnitude than that which a given register or storage location can store or represent.</p>

9.	<p>When can you say that a number is normalized?</p> <p>When the decimal point is placed to the right of the first (nonzero) significant digit the number is said to be normalized.</p>
10.	<p>What is Carry Save addition?</p> <p>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</p>
11.	<p>Define Integer Division and give its rule.</p> <p>Integers are the set of whole numbers and their opposites. The sign of an integer is positive if the number is greater than zero, and the sign is negative if the number is less than zero. The set of all integers represented by the set $\{\dots -4, -3, -2, -1, 0, 1, 2, 3, 4, \dots\}$ Negative integers: $\{\dots -4, -3, -2, -1\}$ Positive integers: $\{1, 2, 3, 4, \dots\}$ $\{0\}$ is neither positive nor negative, neutral. DIVISION RULE: The quotient of two integers with same sign is positive. The quotient of two integers with opposite signs is negative.</p>
12.	<p>Write Restoring and Non-Restoring division algorithm?</p> <p>Restoring Division Algorithm:</p> <ul style="list-style-type: none"> 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 q_0 to 0 and add M back to A (that is, restore A); otherwise, set q_0 to 1. <p>Non-Restoring Division Algorithm</p> <p>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 q_0 to 1; otherwise, set q_0 to 0.</p> <p>Step 2: If the Sign of A is 1, add M to A</p>
13.	<p>Write the rules for add/sub operation on floating point numbers? (May 2017)</p> <ul style="list-style-type: none"> 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
14.	<p>Write the rules for multiply operation on floating point numbers?</p> <ul style="list-style-type: none"> Add the exponents and subtract 127. Multiply the mantissa and determine the sign of the result. Normalize the resulting value, if necessary. Write the rules for divide operation on floating point numbers Subtract the exponents and subtract 127. Divide the mantissa and determine the sign of the result. Normalize the resulting value, if necessary.
15.	<p>Define Truncation.</p> <p>To retain maximum accuracy, all extra bits during operation (called <i>guard bits</i>) are kept (e.g., multiplication). If we assume $n=3$ bits are used in final representation of a number, $n=3$ extra 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.</p>

16.	Explain how Boolean subtraction is performed? Negate the subtrahend (i.e. in $a-b$, the subtrahend is b) then perform addition (2's complement)
17.	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 $0 \leq e < +1$ LSB.
18.	Define Von Neumann Rounding. If at least one of the guard bits is 1, the least significant bit of the retained bits is set to 1 otherwise nothing is changed in retained bits and simply guard bits are dropped.
19.	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.
20.	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 sign opposite to that of the operands.
21.	Define generate and propagate function. The generate function is given by $G_i = x_i y_i$ and The propagate function is given as $P_i = x_i + y_i$.
22.	What is excess-127 format? Instead of the signed exponent E , the value actually stored in the exponent field is an unsigned integer $E' = E + 127$. This format is called excess-127.
23.	What is floating point numbers? In some cases, the binary point is variable and is automatically adjusted as computation proceeds. In such case, the binary point is said to float and the numbers are called floating point numbers
24.	Define ALU. (May 2016) An Arithmetic Logic Unit (ALU) is a digital circuit used to perform arithmetic and logic operations. It represents the fundamental building block of the central processing unit (CPU) of a computer. Modern CPUs contain very powerful and complex ALUs. In addition to ALUs, modern CPUs contain a control unit (CU).
25.	What are the overflow/underflow conditions for addition and subtraction? (Nov 2015) When result cannot be represented in the allocated number of bits. Overflow occurs if the Result $>$ Max value. Underflow occurs if the Result $<$ Min value. Overflow can occur when two positive numbers are added and result is out of range. After addition, the result will become negative. Underflow can occur when two negative numbers are added and result is out of range. After addition, the result will become positive. While adding a positive number with a negative number. No overflow or underflow can occur. Unsigned number representation using n-bits <ul style="list-style-type: none">• Overflow when result $> 2^n - 1$.• Underflow when result < 0. Signed number representation using n-bits <ul style="list-style-type: none">• Overflow when result $> 2^{n-1} - 1$.• Underflow when result $< -2^{n-1}$.
26.	Write the IEEE 754 floating point format? The IEEE 754 standard floating point representation is almost always an approximation of the

	real number. The format is: $(-1)^s(1+\text{Fraction}) \times 2^{(\text{Exponent}-\text{Bias})}$
27.	Subtract $(11011)_2 - (10011)_2$ using 2's complement. (May 2017, Nov 2017) 
28.	Divide $(1001010)_2 / 1000$ (Nov 2017) 
UNIT-II/ PART-B	
1.	Explain the Booth's multiplication algorithm with suitable example (May 2016) (Nov 2016)
2.	Explain the various methods of performing multiplication of n-bit numbers with suitable examples.
3.	Discuss in detail about division algorithm in detail with diagram and examples. (Nov 2015, Nov 2016, Nov 2017) (May 2018)
4.	Explain how floating point addition is carried out in a computer system. Give an example for a binary floating point addition. (May 2015)
5.	Describe subword parallelism in detail.
6.	Explain in detail about the multiplication algorithm with suitable example and diagram. (Nov 2015, May 2015)
7.	Draw and explain the block diagram of floating point adder - subtractor unit with an example.
8.	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)
9.	Briefly Explain Carry Look-ahead adder. (Nov 2014)
10.	Explain briefly about floating point addition and subtraction algorithms (May 2016)
11.	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)
12.	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)
13.	Demonstrate multiplication of two binary numbers with an example. Design an arithmetic element to perform the multiplication.(May 2017) Describe non-restoring division with an example.
14.	Design an arithmetic element to perform the basic floating point operations.(May 2017) What is meant by subword parallelism? Explain.
15.	Add the numbers $(0.75)_{10}$ and $(-0.275)_{10}$ in binary using the Floating point addition algorithm. (May 2018)
16.	Add the numbers $(0.5)_{10}$ and $(0.4375)_{10}$ using the floating point addition.(Nov 2017)

UNIT III PROCESSOR AND CONTROL UNIT

A Basic MIPS implementation – Building a Datapath – Control Implementation Scheme – Pipelining – Pipelined datapath and control – Handling Data Hazards & Control Hazards – Exceptions.

UNIT-III/ PART-A

1.	What is pipelining? The technique of overlapping the execution of successive instruction for substantial improvement in performance is called pipelining.
2.	What is precise exception? A precise exception is one in which all instructions prior to the faulting instruction are complete and instruction following the faulting instruction, including the faulty instruction; do not change the state of the machine.
3.	Define processor cycle in pipelining. The time required between moving an instruction one step down the pipeline is a processor cycle.
4.	What is meant by pipeline bubble? (Nov 2016) To resolve the hazard the pipeline is stall for 1 clock cycle. A stall is commonly called a pipeline bubble, since it floats through the pipeline taking space but carrying no useful work.
5.	What is pipeline register delay? Adding registers between pipeline stages me adding logic between stages and setup and hold times for proper operations. This delay is known as pipeline register delay.
6.	What are the major characteristics of a pipeline? The major characteristics of a pipeline are: <ol style="list-style-type: none"> 1. Pipelining cannot be implemented on a single task, as it works by splitting multiple tasks into a number of subtasks and operating on them simultaneously. 2. The speedup or efficiency achieved by suing a pipeline depends on the number of pipe stages and the number of available tasks that can be subdivided.
7.	What is data path? (Nov 2016) (May 2018) As instruction execution progress data are transferred from one instruction to another, often passing through the ALU to perform some arithmetic or logical operations. The registers, ALU, and the interconnecting bus are collectively referred as the data path.
8.	What do you mean by branch penalty? The time lost as a result of a branch instruction is often referred to as branch penalty.
9.	Define structural hazards. This is the situation when two instruction require the use of a given hardware resource at the same time. The most common case in which this hazard may arise is in access to memory.
10.	What is side effect? When a location other than one explicitly named in an instruction as a destination operand is affected, the instruction is said to have a side effect.
11.	What is Instruction or control hazard? The pipeline may be stalled because of a delay in the availability of an instruction. For example, this may be a result of a miss in the cache, requiring the instruction to be fetched from the main memory. Such hazards are often called control hazards or instruction hazard.
12.	What is branch folding? When the instruction fetch unit executes the branch instruction concurrently with the execution of the other instruction, then this technique is called branch folding.
13.	What do you mean by delayed branching? Delayed branching is used to minimize the penalty incurred as a result of conditional branch instruction. The location following the branch instruction is called delay slot. The instructions in

	the delay slots are always fetched and they are arranged such that they are fully executed whether or not branch is taken. That is branching takes place one instruction later than where the branch instruction appears in the instruction sequence in the memory hence the name delayed branching
14.	What are the two types of branch prediction techniques available? The two types of branch prediction techniques are static branch prediction and dynamic branch prediction.
15.	What is a hazard? What are its types? (Nov 2015) Any condition that causes the pipeline to stall is called hazard. They are also called as stalls or bubbles. The various pipeline hazards are: <ul style="list-style-type: none"> • Data hazard • Structural Hazard • Control Hazard
16.	Why is branch prediction algorithm needed? The branch instruction will introduce branch penalty which would reduce the gain in performance expected from pipelining. Branch instructions can be handled in several ways to reduce their negative impact on the rate of execution of instructions. Thus the branch prediction algorithm is needed.
17.	What is branch Target Address? The address specified in a branch, which becomes the new program counter, if the branch is taken. In MIPS the branch target address is given by the sum of the offset field of the instruction and the address of the instruction following the branch.
18.	What is an interrupt? An exception is the one that comes from outside of the processor. There are two types of interrupt. They are imprecise interrupt and precise interrupt.
19.	Define Pipeline speedup. The ideal speedup from a pipeline is equal to the number of stages in the pipeline. $\text{Speedup} = \frac{\text{Time per instruction on unpipelined machine}}{\text{Number of pipe stages}}$
20.	What is meant by vectored interrupt? An interrupt for which the address to which control is transferred is determined by the cause of the exception.
21.	Define exception. (Nov 2014, May 2016) The term exception is used to refer to any event that causes an interruption otherwise an unexpected change in the control flow. When an exception or interrupt occurs, the hardware begins executing code that performs an action in response to the exception. This action may involve killing a process, outputting a error message, communicating with an external device.
22.	What are R-type instructions? (May 2015) R instructions are used when all the data values used by the instruction are located in registers. All R-type instructions have the following format: OP rd, rs,rt. Where "OP" is the mnemonic for the particular instruction. rs, and rt are the source registers, and rd is the destination register.
23.	What is a branch prediction buffer? (May 2015) The simplest thing to do with a branch is to predict whether or not it is taken. This helps in where the branch delay is longer than the time it takes to compute the possible target PC _s .
24.	What is meant by branch prediction? (Nov 2015) Branch Instructions may introduce branch penalty. To avoid it, branch prediction is done by two ways. Static Branch prediction The static branch prediction, assumes that the branch will not take place and to continue to fetch

	<p>instructions in sequential address order.</p> <p>Dynamic Branch prediction</p> <p>The idea is that the processor hardware assesses the likelihood of a given branch being taken by keeping track of branch decisions every time that instruction is executed. The execution history used in predicting the outcome of a given branch instruction is the result of the most recent execution of that instruction.</p>												
25.	<p>What are the advantages of pipelining? (May 2016)</p> <p>The cycle time of the processor is reduced and increases the instruction throughput. The more pipeline stages a processor has, the more instructions it can process "at once" and the less of a delay there is between completed instructions.</p>												
26.	<p>Name the control signals required to perform arithmetic operations. (May 2017)</p> <p>ALU control: specifies what operation ALU performs - I.e., ALU operation control signals</p> <ul style="list-style-type: none"> - Eight input combinations (3 input control signals) - Five combinations used to select operation <table> <thead> <tr> <th>ALU control input</th><th>Function</th></tr> </thead> <tbody> <tr> <td>000</td><td>AND</td></tr> <tr> <td>001</td><td>OR</td></tr> <tr> <td>010</td><td>add</td></tr> <tr> <td>110</td><td>subtract</td></tr> <tr> <td>111</td><td>set on less than</td></tr> </tbody> </table> <ul style="list-style-type: none"> • Based on instruction class, one of these will be done 	ALU control input	Function	000	AND	001	OR	010	add	110	subtract	111	set on less than
ALU control input	Function												
000	AND												
001	OR												
010	add												
110	subtract												
111	set on less than												
27.	<p>Mention the various types of pipelining. (Nov 2017)</p> <p>Linear pipelines - A linear pipeline processor is a series of processing stages and memory access.</p> <p>Non-linear pipelines - A non-linear pipelining (also called dynamic pipeline) can be configured to perform various functions at different times. In a dynamic pipeline, there is also feed-forward or feed-back connection. A non-linear pipeline also allows very long instruction words. Eg:</p>												
28.	<p>Define Hazard. Give an example for data hazard.(May 2017)</p> <p>Any condition that causes the pipeline to stall is called hazard. Any condition in which either the source or the destination operands of an instruction are not available at the time expected in the pipeline is called data hazard. Eg:</p> <pre> Loop: L.D F0,0(R1) ;F0=array element ADD.D F4,F0,F2 ;add scalar in F2 S.D F4,0(R1) ;store result DADDUI R1,R1,#-8 ;decrement pointer 8 bytes BNE R1,R2,LOOP ;branch R1!=R2 </pre> <p>The data dependences in this code sequence involve both floating-point data:</p> <pre> Loop: L.D F0,0(R1) ;F0=array element ADD.D F4,F0,F2 ;add scalar in F2 S.D F4,0(R1) ;store result </pre> <p>and integer data:</p> <pre> DADDUI R1,R1,#-8 ;decrement pointer ;8 bytes (per DW) BNE R1,R2,Loop ;branch R1!=R2 </pre>												
29.	<p>Mention the various phase in executing an instruction. (Nov 2017)</p> <p>Various phase in executing an instruction are:</p> <p>IF: Instruction Fetch</p> <p>ID: Instruction decode and register file read</p> <p>EX: Execution or Address Calculation</p> <p>MEM: Data Memory Access</p> <p>WB: Write Back</p>												

30.	<p>What is the ideal CPI of a pipelined processor? (May 2018)</p> <p>In computer architecture, CPI is cycles per instruction (clock cycles per instruction, clocks per instruction, or CPI) is one aspect of a processor's performance: the average number of clock cycles per instruction for a program or program fragment.[1] It is the multiplicative inverse of instructions per cycle. The average of Cycles Per Instruction in a given process is defined by the following:</p> $CPI = \sum_i (IC_i)(CC_i)/(IC)$ <p>Where IC_i is the number of instructions for a given instruction type i, CC_i is the clock-cycles for that instruction type and $IC = \sum_i (IC_i)$ is the total instruction count. The summation sums over all instruction types for a given benchmarking process.</p>
31.	<p>What is meant by exception? Give one example of MIPS exception. (May 2018)</p> <p>MIPS terminology</p> <p>Exception: any unexpected change in the internal control flow</p> <ul style="list-style-type: none"> – Invoking an operating system service from user program – Integer arithmetic overflow – Using an undefined or unimplemented instruction – Hardware malfunctions
UNIT-III / PART-B	
1.	Explain in detail about the basic MIPS implementation with suitable diagram.
2.	What are the major components required to execute MIPS instruction while building a datapath.
3.	Explain how the instruction pipeline works? What are the various situations where an instruction pipeline can stall? Illustrate with an example. (Nov 2015, Nov 2016)
4.	Explain the basic MIPS implementation with necessary multiplexers and control lines. (Nov 2015)
5.	What is pipelining? Discuss about pipelined datapath and control.(May 2016) (May 2018)
6.	Discuss the limitations of pipelining a processor's datapath. Suggest the methods to overcome them. (May 2018)
7.	Explain the pipeline hazard in detail (Nov 2017)
8.	Explain in detail how exceptions are handled in MIPS architecture.(May 2015)
9.	<p>(i) Why is branch prediction algorithm needed? Differentiate between the static and dynamic techniques. (May 2016)</p> <p>(ii) Describe the techniques for handling control hazards in pipelined datapath.</p>
10.	Explain Data path and its control in detail. (Nov 2014, Nov 2017)
11.	What is Hazard? Explain its types with suitable examples. (Nov 2014, May 2015, May 2016)
12.	Discuss the modified datapath to accommodate pipelined executions with a diagram. (May 2017)
13.	Explain the hazards caused by the unconditional branching statements. (May 2017). Describe operand forwarding in a pipeline processor with a diagram.
14.	A pipelined processor uses the delayed branch technique. You are asked to recommend one of two possibilities for the design of the processor. In the first possibility, the processor has a 4-stage pipeline and one delay slot, and in the second possibility it has a 6-stage pipeline with two delay slots. Assume that 20% of the instructions are branch instructions and that an optimizing compiler succeeds in filling 80% of the single delay slot. For the second alternative, the compiler is able to fill the second slot 25% of the time. (May 2017)

UNIT IV PARALLELISM

Parallel processing challenges - Flynn's classification - SISD, MIMD, SIMD, SPMD, and Vector Architectures - Hardware multithreading - Multi-core processors and other Shared Memory Multiprocessors - Introduction to Graphics Processing Units, Clusters, Warehouse Scale Computers and other Message-Passing Multiprocessors.

UNIT-IV / PART-A

1.	What is Instruction level parallelism? (Nov 2015, Nov 2016, May 2017) ILP is a measure of how many of the operations in a computer program can be performed simultaneously. The potential overlap among instructions is called instruction level parallelism. There are two primary methods for increasing the potential amount of instruction-level parallelism. 1. Increasing the depth of the pipeline to overlap more instructions. 2. Multiple issue.
2.	Define Static multiple issue. It is an approach to implement multiple-issue processor where many decisions are made by the compiler before execution.
3.	Define Dynamic multiple issue. It is an approach to implement multiple-issue processor where many decisions are made during execution by the processor.
4.	What is Speculation? One of the most important methods for finding and exploiting more ILP is speculation. It is an approach whereby the compiler or processor guesses the outcome of an instruction to remove its 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.
5.	What is Loop unrolling? It is a technique to get more performance from loops that helps in accessing the arrays, in which multiple copies of the loop body are made and instructions from different iterations are scheduled together.
6.	Define Register renaming. The renaming of registers is done by the compiler or hardware to remove anti-dependences, renaming removes WAW/WAR hazard.
7.	Define a super scalar processor. (Nov 2015) Superscalar is an advanced pipelining technique that enables the processor to execute more than one instruction per clock cycle by selecting them during execution. Dynamic multiple-issue processors are also known as superscalar processors, or simply superscalars.
8.	Define Commit unit. It is a unit in a dynamic or out-of-order execution pipeline that decides when it is safe to release the result of an operation to programmer visible registers and memory.
9.	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.
10.	Define Out of order execution. A situation in pipelined execution when an instruction blocked from executing does not cause the following instructions to wait.
11.	What is In order commit? It is a commit in which the results of pipelined execution are written to the programmer visible state in the same order that instructions are fetched.
12.	Define Single Instruction Multiple Data streams (SIMD). A computer which exploits multiple data streams against a single instruction stream to perform operations which may be naturally parallelized. For example, an array processor.

13.	Define Single Instruction, Single Data stream (SISD). A sequential computer which exploits no parallelism in either the instruction or data streams. Single control unit (CU) fetches single Instruction Stream (IS) from memory. The CU then generates appropriate control signals to direct single processing element (PE) to operate on single Data Stream (DS) i.e. one operation at a time. Examples of SISD architecture are the traditional uniprocessor machines like a PC.
14.	Differentiate between Strong scaling and weak scaling.(May 2015,Nov 2017) Strong scaling: Speed-up is achieved on a multi-processor without increasing the size of the problem. Weak scaling: Speed-up is achieved on a multi-processor while increasing the size of the problem proportionally to the increase in the number of processors.
15.	Define Multiple Instruction, Single Data stream (MISD). Multiple instructions operate on a single data stream. It is uncommon architecture which is generally used for fault tolerance. Heterogeneous systems operate on the same data stream and must agree on the result. Examples include the Space Shuttle flight control computer.
16.	Define Multiple Instruction, Multiple Data streams (MIMD). Multiple autonomous processors are simultaneously executing different instructions on different data. Distributed systems are generally recognized to be MIMD architectures; either exploiting a single shared memory space or a distributed memory space. A core superscalar processor is an MIMD processor.
17.	What is Fine grained multithreading? (May 2016) Switches between threads on each instruction, causing the execution of multiples threads to be interleaved, <ol style="list-style-type: none"> Usually done in a round-robin fashion, skipping any stalled threads CPU must be able to switch threads every clock
18.	What is Coarse grained multithreading? Switches threads only on costly stalls, such as L2 cache misses.
19.	Define Multicore processors. A multi-core processor is a processing system composed of two or more independent cores. The cores are typically integrated onto a single integrated circuit die or they may be integrated onto multiple dies in a single chip package.
20.	What is symmetric multi-core processor? Symmetric multi-core processors are one that has multiple cores on a single chip, and all of those cores are identical. Example: Intel Core 2.
21.	What is asymmetric multi-core processor? In an asymmetric multi-core processor, the chip has multiple cores onboard, but the cores might have different designs. Each core will have different capabilities.
22.	Define multithreading. (Nov 2014,Nov 2016) Multiple threads to share the functional units of 1 processor via overlapping processor must duplicate independent state of each thread e.g., a separate copy of register file, a separate PC, and for running independent programs, a separate page table memory shared through the virtual memory mechanisms, which already support multiple processes.
23.	What is the need for Speculation? (Nov 2014) It is one of the most important methods for finding and exploiting more ILP. It is an approach that allows the compiler or the process to guess about the properties of an instruction, so as to enable execution to begin for other instructions that may depend on the speculated instruction.

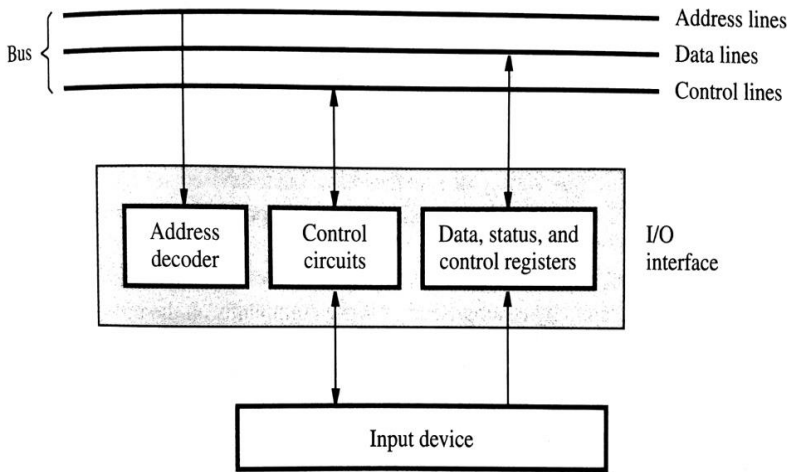
24.	What is Flynn's Classification? (Nov 2014) Michael Flynn uses the stream concept for describing a machine's structure. A stream is nothing but a sequence of items (data or instruction). The parallelism in the instruction and data stream called for by the instruction at the most constrained of the multiprocessor and placed all computers into one to four categories. <ul style="list-style-type: none"> • Single Instruction, Single Data stream (SISD) • Single Instruction, Multiple Data streams (SIMD) • Multiple Instruction, Single Data stream (MISD) • Multiple Instruction, Multiple Data streams (MIMD) 	
25.	Compare UMA and NUMA multiprocessors.(May 2015) The main difference between the NUMA and UMA memory architecture is the location of the Memory. The UMA architecture nodes have first and second cache memory levels joint with the processor, next levels of the memory hierarchy are "in the other side" of the interconnection network.	
26.	State the need for Instruction Level Parallelism (May 2016) A new way to improve uniprocessor performance. The proposals such as VLIW, superscalar, and even relatively old ideas such as vector processing try to improve computer performance by exploiting instruction-level parallelism. They take advantage of this parallelism by issuing more than one instruction per cycle explicitly (as in VLIW or superscalar machines) or implicitly (as in vector machines).	
27.	Define implicit multithreading and explicit multithreading. (May 2017) All commercial processors and most experimental ones use explicit multithreading <ul style="list-style-type: none"> - Concurrently execute instructions from different explicit threads - Interleave instructions from different threads on shared pipelines or parallel execution on parallel pipelines Implicit multithreading is concurrent execution of multiple threads extracted from single sequential program <ul style="list-style-type: none"> - Implicit threads defined statically by compiler or dynamically by hardware. 	
28.	Define GPUs. A programmable logic chip (processor) specialized for display functions. The GPU renders images, animations and video for the computer's screen. GPUs are located on plug-in cards, in a chipset on the motherboard or in the same chip as the CPU.	
29.	Differentiate Fine-grained multithreading and Coarse-grained multithreading.(Nov 2017)	
	Fine-grained Multithreading	Coarse-grained multithreading
	It switches between threads on each instruction, resulting in interleaved execution of multiple threads. It can hide the throughput losses that arise from both short and long stalls, since instructions	It switches threads only on costly stalls, such as second-level cache misses. Coarse-grained multithreading suffers, however, from a major drawback: it is limited in its ability to overcome throughput losses, especially from shorter stalls
30.	Define Message passing. Message passing is a technique for invoking behavior (i.e., running a program) on a computer. In contrast to the traditional technique of calling a program by name, message passing uses an object model to distinguish the general function from the specific implementations. The invoking program sends a message and relies on the object to select and execute the appropriate code.	

31.	Give example for each class in Flynn's classification.(May 2018) <ul style="list-style-type: none"> • Single Instruction, Single Data stream (SISD) - traditional uniprocessor machines • Single Instruction, Multiple Data streams (SIMD) - pipelining or multiple functional units • Multiple Instruction, Single Data stream (MISD) - the Space Shuttle flight control computer • Multiple Instruction, Multiple Data streams (MIMD) - multi-core superscalar processors, and distributed systems
UNIT-IV / PART-B	
1.	Explain in detail Flynn's classification of parallel hardware. (Nov 2015,May 2016,Nov 2016) (or)Explain with diagrammatic illustration Flynn's classification. (Nov 2017)
2.	Explain SISD and SIMD with suitable example. (May 2015)
3.	Explain MISD and MIMD with suitable example. (May 2015)
4.	Discuss the centralized and distributed shared memory multiprocessors with suitable diagrams (or) Discuss Shared memory multiprocessor with a neat diagram.(Nov 2016) (May 2018)
5.	Discuss briefly about the motivation of Multi-core computing
6.	Describe dependency and the various types of dependencies in detail.
7.	What is hardware multithreading? Compare and contrast Fine-grained multithreading and Coarse-grained multithreading.(May 2015) (May 2018)
8.	Explain the Dynamic & static multiple issue processor and their scheduling with block diagram.
9.	Explain Instruction Level Parallel Processing. State the challenges of parallel processing. (Nov 2014) (May 2018)
10.	Explain the term: Multicore Processor (ii) Hardware Multithreading (Nov 2014, Nov 2015,May 2016)
11.	Discuss the challenges in parallel processing with necessary examples. May 2017) Explain Flynn's classification of parallel processing with necessary diagrams.
12.	Explain the four principal approaches to multithreading with necessary diagrams(May 2017)
13.	Describe Simultaneous Multithreading (SMT) with an example. (Nov 2017)
14.	Discuss in detail about Graphics Processing Units.
15.	Explain about the Computer Architecture Of Warehouse-Scale Computers in detail with its diagram.
UNIT V MEMORY AND I/O SYSTEMS	
Memory Hierarchy - memory technologies - cache memory - measuring and improving cache performance - virtual memory, TLB's - Accessing I/O Devices - Interrupts - Direct Memory Access - Bus structure - Bus operation - Arbitration - Interface circuits - USB.	
UNIT-V / PART-A	
1.	What is principle of locality? The principle of locality states that programs access a relatively small portion of their address space at any instant of time. Two different types of locality have been observed: Temporal locality: states that recently accessed items are likely to be accessed in the near future. Spatial locality: says that items whose addresses are near one another tend to be referenced close together in time.
2.	Define temporal locality. The principle stating that a data location is referenced then it will tend to be referenced again soon. Temporal locality is found in instruction loops, data stacks and variable accesses.

3.	Define spatial locality. The locality principle states that if a data location is referenced, data locations with nearby addresses will tend to be referenced soon.
4.	What is the need to implement Memory as Hierarchy?(May 2015) <ul style="list-style-type: none"> • It is a structure that uses multiple levels of memory with different speeds and sizes. • The memory unit is an essential component in a digital computer since it is needed for storing program and data. • They are used for storing system programs, large data files, and other backup information. • Only programs and data currently needed by the processor reside in main memory.
5.	Define Hit and Miss. The performance of cache memory is frequently measured in terms of a quantity called hit ratio. When the CPU refers to memory and finds the word in cache, it is said to produce a hit. If the word is not found in cache, then it is in main memory and it counts as a miss.
6.	What is cache memory? (Nov 2016) It is a fast memory that is inserted between the larger slower main memory and the processor. It holds the currently active segments of a program and their data.
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) modulo (Number of blocks in the cache)
8.	Define write through. It is a scheme in which writes always update both the cache and the next lower level of the memory hierarchy, ensuring the data is always consistent between the two.
9.	Define write buffer. It is a queue that holds data while the data is waiting to be written to memory.
10.	What is write-back? It 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 is Virtual memory?(Nov 2017) The data is to be stored in physical memory locations that have addresses different from those specified by the program. The memory control circuitry translates the address specified by the program into an address that can be used to access the physical memory.
12.	Distinguish between memory mapped I/O and I/O mapped I/O. Memory mapped I/O: When I/O devices and the memory share the same address space, the arrangement is called memory mapped I/O. The machine instructions that can access memory are used to transfer data to or from an I/O device. I/O mapped I/O: Here the I/O devices have different address space. It has special I/O instructions. The advantage of a separate I/O address space is that I/O devices deal with fewer address lines.
13.	How does a processor handle an interrupt? Assume that an interrupt request arises during execution of instruction i . Steps to handle interrupt by the processor are as follows: <ol style="list-style-type: none"> 1. Processor completes execution of instruction i 2. Processor saves the PC value, program status on to stack. 3. It loads the PC with starting address of ISR 4. After ISR is executed, the processor resumes the main program execution by reloading PC with $(i+1)$th instruction address.

14.	What is SCSI? Small Computer System Interface, a interface standard. SCSI interfaces provide for faster data transmission rates (up to 80 megabytes per second) than standard serial and parallel ports. In addition, you can attach many devices to a single SCSI port, so that SCSI is really an I/O,bus rather than simply an interface.
15.	Define USB. Universal Serial Bus, an external bus standard that supports data transfer rates of 12 Mbps. A single USB port can be used to connect up to 127 peripheral devices, such as mice, modems, and keyboards. USB also supports Plug-and-Play installation and hot plugging.
16.	Distinguish between isolated and memory mapped I/O. The isolated I/O method isolates memory and I/O addresses so that memory address values are not affected by interface address assignment since each has its own address space. In memory mapped I/O , there are no specific input or output instructions. The CPU can manipulate I/O data residing in interface registers with the same instructions that are used to manipulate memory words.
17.	What is meant by vectored interrupt? Vectored Interrupts are type of I/O interrupts in which the device that generates the interrupt request (also called IRQ in some text books) identifies itself directly to the processor.
18.	Compare Static RAM and Dynamic RAM. (May 2018) Static RAM is more expensive, requires four times the amount of space for a given amount of data than dynamic RAM, but, unlike dynamic RAM, does not need to be power-refreshed and is therefore faster to access. One source gives a typical access time as 25 nanoseconds in contrast to a typical access time of 60 nanoseconds for dynamic RAM. (More recent advances in dynamic RAM have improved access time.) Static RAM is used mainly for the level-1 and level-2 caches that the microprocessor looks in first before looking in dynamic RAM. Dynamic RAM uses a kind of capacitor that needs frequent power refreshing to retain its charge. Because reading a DRAM discharges its contents, a power refresh is required after each read. Apart from reading, just to maintain the charge that holds its content in place, DRAM must be refreshed about every 15 microseconds. DRAM is the least expensive kind of RAM.
19.	What are the various memory technologies? (Nov 2015) SRAM, DRAM, Magnetic Disks
20.	Differentiate Programmed I/O and Interrupt I/O. (Nov 2014) In programmed I/O all data transfers between the computer system and external devices are completely controlled by the computer program. Part of the program will check to see if any external devices require attention and act accordingly. Interrupt I/O is a way of controlling input/output activity in which a peripheral or terminal that needs to make or receive a data transfer sends a signal that causes a program interrupt to be set.
21.	What is the purpose of Dirty/Modified bit in Cache memory? (Nov 2014) During Write back the information is written only to the block in the cache. The modified cache block is written to main memory only when it is replaced. To reduce the frequency of writing back blocks on replacement, a dirty bit is commonly used. This status bit indicates whether the block is dirty (modified while in the cache) or clean (not modified). If it is clean the block is not written on a miss.
22.	Point out how DMA can improve I/O speed?(May 2015) Direct memory access (DMA) is a feature of computer systems that allows certain hardware subsystems to access main system memory (RAM) independently of the central processing unit (CPU).

23.	<p>Differentiate physical address from logical address.</p> <p>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.</p>
24.	<p>What is DMA? (Nov 2014)</p> <p>DMA (Direct Memory Access) provides I/O transfer of data directly to and from the memory unit and the peripheral. The following DMA transfer combinations are possible:</p> <ul style="list-style-type: none"> • Memory to memory • Memory to peripheral • Peripheral to memory <p>Peripheral to peripheral</p>
25.	<p>Define Hit ratio. (Nov 2015)</p> <p>The hit rate, or hit ratio, 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.</p>
26.	<p>Define memory hierarchy. (May 2016)</p> <p>In computer architecture the memory hierarchy is a concept used to discuss performance issues in computer architectural design, algorithm predictions, and lower level programming constructs involving locality of reference. The memory hierarchy in computer storage separates each of its levels based on response time.</p>
27.	<p>State the advantages of virtual memory. (May 2016)</p> <p>The primary benefits of virtual memory include</p> <ul style="list-style-type: none"> • freeing applications from having to manage a shared memory space • increased security due to memory isolation and • being able to conceptually use more memory than might be physically available, using the technique of paging.
28.	<p>What is meant by address mapping? (Nov 2016)</p> <p>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) modulo (Number of blocks in the cache)</p>
29.	<p>Summarize the sequence of events involved in handling an interrupt request from a single device. (May 2017)</p> <ol style="list-style-type: none"> 1. The device raises an interrupt request. 2. The processor interrupts the program currently being executed at the time. 3. Interrupts are disabled by changing the control bits in the PS. 4. The device is informed that its request has been recognized, and in response, it deactivates the interrupt request signal. 5. The action requested by the interrupt is performed by the interrupt - service routine. 6. Interrupts are enabled and 7. Execution of the interrupted program is resumed
30.	<p>Define memory interleaving. (May 2017)</p> <p>In computing, interleaved memory is a design made to compensate for the relatively slow speed of dynamic random-access memory (DRAM) or core memory, by spreading memory addresses evenly across memory banks. That way, contiguous memory reads and writes are using each memory bank in turn, resulting in higher memory throughputs due to reduced waiting for memory banks to become ready for desired operations.</p>

31.	<p>How many total bits are required for a direct-mapped cache with 16KB of data and 4-word blocks, assuming a 32-bit address?</p> <p>We know that 16 KB is 4K (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</p> $2^{10} \times (4 \times 32 + (32 - 10 - 2 - 2) + 1) = 2^{10} \times 147 = 147 \text{ Kbits}$ <p>or 18.4 KB for a 16 KB 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.</p>
32.	<p>Define Bus.</p> <p>The bus consists of three sets of lines used to carry address, data, and control signals. I/O device interfaces are connected to these lines</p> 
33.	<p>What is Bus Arbitration?</p> <p>It is the process by which the next device to become the bus master is selected and the bus mastership is transferred to it.</p> <p>Types:</p> <p>There are 2 approaches to bus arbitration. They are</p> <ol style="list-style-type: none"> Centralized arbitration (A single bus arbiter performs arbitration) Distributed arbitration (all devices participate in the selection of next bus master).
34.	<p>What is USB?</p> <p>The Universal Serial Bus (USB) is the most widely used interconnection standard. A large variety of devices are available with a USB connector, including mice, memory keys, disk drives, printers, and cameras. The commercial success of the USB is due to its simplicity and low cost. The original USB specification supports two speeds of operation, called low-speed (1.5 Megabits/s) and full-speed (12 Megabits/s). It supports data transfer rates up to 5 Gigabits/s.</p>
35.	<p>What is the use of DMA controller? (May 2018)</p> <p>DMA can also be used for "memory to memory" copying or moving of data within memory.</p>
UNIT-V / PART-B	
1.	Explain in detail about the basic structure of a memory level hierarchy with suitable diagram.
2.	Elaborate on the various memory technologies and its relevance. (May 2015, Nov 2017)
3.	Explain in detail about the different ways of measuring and improving the performance of cache memory.

4.	Discuss the steps involved in the address translation of Virtual Memory with necessary block diagram. (Nov 2016) (May 2018)
5.	Draw the typical block diagram of a DMA controller and explain how it is used for direct data transfer between memory and peripherals? (Nov 2015, May 2016, Nov 2016) (May 2018) What is meant by Direct Memory Access? Explain the use of DMA controllers in a computer system.(May 2017)
6.	What is an interrupt? Explain the different types of interrupts and the different ways of handling the interrupts. (May 2018)
7.	Write short note on I/O processor. What is the need for an I/O interface? Describe the functions of SCSI interface with a neat diagram.
8.	(i) Define Cache Memory? Explain various mapping techniques associated with cache memory. (May 2016)(May 2017) (ii) Explain in detail about the Bus Arbitration techniques in DMA. (Nov 2014)(May 2017)
9.	(i) Draw different memory address layouts and brief about the technique used to increase the average rate of fetching words from the main memory. (ii) Explain in detail about any two standard input and output interfaces required to connect the I/O device to the Bus.(Nov 2014)
10.	What is virtual memory? Explain in detail about how virtual memory is implemented with neat diagram? (May 2015, Nov 2015) (May 2017)(May 2018)
11.	Explain mapping functions in cache memory to determine how memory blocks are placed in cache.(May 2017)
12.	What is cache memory? How to improve cache performance? Discuss. (Nov 2017)
13.	Assume the miss rate of an instruction cache is 2% and the miss rate of the data cache is 4%.If a processor has a CPI of 2 without any memory stalls and the miss penalty is 100 cycles for all misses, determine how much faster a processor would run with a perfect cache that never missed. Assume the frequency of all loads and stores is 36%. (Nov 2017)
14.	Explain the following in detail: i. Bus structure ii. Bus arbitration iii. Bus operation
15.	Explain about two types of interface circuits in detail.
16.	Explain USB in detail.