CS8491 Computer Architecture
PUBLISHED BY STUCOR

PART-A

Q. No.	Questions	CO	Bloom's Level
	Write the basic functional units of computer? The basic functional units of a computer are input unit, output unit, memory unit, ALU unit and control unit	C204.1	BTL1
	Write the basic functional units of computer? (APR/MAY 2017,NOV/DEC 2017) The basic functional units of a computer are input unit, output unit, memory unit, ALU unit and control unit.	C204.1	BTL1
	What is a bus? What are the different buses in a CPU? [APR/MAY 2011] A group of lines that serve as a connecting path for several devices is called bus .The different buses in a CPU are 1] Data bus 2] Address bus 3] Control bus.	C204.1	BTL1
	What is meant by stored program concepts? Stored program concept is an idea of storing the program and data in the memory	C204.1	BTL1
	Define multiprogramming?(A.U.APR/MAY 2013) Multiprogramming is a technique in several jobs are in main memory at once and the processor is switched from job as needed to keep several jobs advancing while keeping the peripheral devices in use.	C204.1	BTL1
	What is meant by VLSI technology? VLSI is the abbreviation for Very Large Scale Integration. In this technology millions of transistors are put inside a single chip as tiny components. The VLSI chips do the function of millions of transistors. These are Used to implement parallel algorithms directly in hardware	C204.1	BTL6

	Define multiprocessing? Multiprocessing is the ability of an operating system to support more than one process at the same time	C204.1	BTL6
	List the eight great ideas invented by computer architecture? APR/MAY-2015 • Design for Moore's Law • Use abstraction to simplify design • Make the common case fast • Performance via Parallelism • Performance via Pipelining • Performance via Prediction • Hierarchy of Memory • Dependability via Redundancy	C204.1	BTL1
	Define power wall. • Old conventional wisdom • Power is free • Transistors are expensive • New conventional wisdom: "Power wall" • Power expensive • Transistors"free" (Can put more on chip than can afford to turn on)	C204.1	BTL1
	What are clock and clock cycles? The timing signals that control the processor circuits are called as clocks. The clock defines regular time intervals called clock cycles.	C204.1	BTL1
	What is uniprocessor? A **uniprocessor system** is defined as a computer system that has a single central processing unit that is used to execute computer tasks. As more and more modern software is able to make use of multiprocessing architectures, such as SMP and MPP, the term *uniprocessor* is therefore used to distinguish the class of computers where all processing tasks share a single CPU.	C204.1	BTL1
	What is multicore processor? A multi-core processor is a single computing component with two or more independent actual central processing units (called "cores"), which are the units that read and execute program instructions.The instructions are ordinary CPU instructions such as add, move data, and branch, but the multiple cores can run multiple instructions at the same time, increasing overall speed for programs amenable to parallel computing	C204.1	BTL1

	Differentiate super computer and mainframe computer. A computer with high computational speed, very large memory and parallel structured hardware is known as a super computer.EX: CDC 6600. 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.	C204.1	BTL1
	Differentiate between minicomputer and microcomputer. Minicomputers are small and low cost computers are characterized by Short word size i.e. CPU word sizes of 8 or 16 bits. They have limited hardware and software facilities. They are physically smaller in size.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	C204.1	BTL1
	What is instruction register?(NOV/DEC 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.	C204.1	BTL1
	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.	C204.1	BTL1
	What is processor time? The sum of the periods during which the processor is active is called the processor time	C204.1	BTL1
	Give the CPU performance equation. CPU execution time for a program =Instruction Count **X**Clock cycles per instruction**X**Clock cycle time.	C204.1	BTL1
	What is superscalar execution? In this type of execution, 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	C204.1	BTL1

	What is RISC and CISC? The processors with simple instructions are called as Reduced Instruction Set Computers (RISC). The processors with more complex instructions are called as Complex Instruction Set Computers (CISC).	C204.1	BTL1
	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	C204.1	BTL1
	Define addressing modes and its various types.(nov/dec 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, Based or Displacement Addressing, PC-Relative Addressing, Pseudodirect Addressing.	C204.1	BTL1
	Define register mode addressing. In register mode addressing, the name of the register is used to specify the operand. Eg. Add $s3, $s5,$s6.	C204.1	BTL1
	Define Based or Displacement mode addressing. In based 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).	C204.1	BTL1
	State Amdahl's Law. Amdahl's law is a formula used to find the maximum improvement improvement possible by improving a particular part of a system. In parallel computing, Amdahl's law is mainly used to predict the theoretical maximum speedup for program processing using multiple processors. $$\text{Speedup} = \frac{\text{Performance for entire task using the enhancement when possible}}{\text{Performance for entire task without using the enhancement}}$$ Alternatively, $$\text{Speedup} = \frac{\text{Execution time for entire task without using the enhancement}}{\text{Execution time for entire task using the enhancement when possible}}$$	C204.1	BTL1

	Define Relative mode addressing. (Nov 2014) In PC-relative mode addressing, the branch address is the sum of the PC and a constant in the instruction. - In the relative address mode, the effective address is determined by the index mode by using the program counter in stead of general purpose processor register. This mode is called relative address mode.	C204.1	BTL1
	Distinguish pipelining from parallelism APR/MAY 2015 parallelism means we are using more hardware for the executing the desired task. in parallel computing more than one processors are running in parallel. there may be some dedicated hardware running in parallel for doing the specific task. while the pipelining is an implementation technique in which multiple instructions are overlapped ninexecution.parallelism increases the performance but the area also increases. in case of pipelining the performance and througput increases at the cost of pipelining registers area pipelining there are different hazards like data hazards, control hazards etc.	C204.1	BTL1
	Distinguish pipelining from parallelism APR/MAY 2015 parallelism means we are using more hardware for the executing the desired task. in parallel computing more than one processors are running in parallel. there may be some dedicated hardware running in parallel for doing the specific task. while the pipelining is an implementation technique in which multiple instructions are overlapped ninexecution.parallelism increases the performance but the area also increases. in case of pipelining the performance and througput increases at the cost of pipelining registers area pipelining there are different hazards like data hazards, control hazards etc.	C204.1	BTL1
	How to represent Instruction in a computer system?MAY/JUNE 2016 Computer instructions are the basic components of a machine language program. They are also known as *macrooperations*, since each one is comprised of a sequences of microoperations. Each instruction initiates a sequence of microoperations that fetch operands from registers or memory, possibly perform arithmetic, logic, or shift operations, and store results in registers or memory. Instructions are encoded as binary *instruction codes*. Each instruction code contains of a*operation code*, or *opcode*, which designates the overall purpose of the instruction (e.g. add, subtract, move, input, etc.). The number of bits allocated for the opcode determined how many different instructions the architecture	C204.1	BTL3

	supports. In addition to the opcode, many instructions also contain one or more *operands*, which indicate where in registers or memory the data required for the operation is located. For example, add instruction requires two operands, and a not instruction requires one.		
	Brief about relative addressing mode.<u>NOV/DEC 2014</u> **Relative addressing mode** - In the relative address mode, the effective address is determined by the index mode by using the program counter in stead of general purpose processor register. This mode is called relative address mode.	C204.1	BTL1
	Distinguish between auto increment and auto decrement addressing mode? **MAY/JUNE 2016** A special case of indirect register mode. The register whose number is included in the instruction code, contains the address of the operand. Autoincrement Mode = after operand addressing , the contents of the register is incremented. Decrement Mode = before operand addressing, the contents of the register is decrement. We denote the autoincrement mode by putting the specified register in parentheses, to show that the contents of the register are used as the efficient address, followed by a plus sign to indicate that these contents are to be incremented after the operand is accessed. Thus, using register R4, the autoincrement mode is written as (R4)+. As a companion for the autoincrement mode, another mode is often available in which operands are accessed in the reverse order. *Autodecrementmode* The contents of a register specified in the instruction are decremented. These contents are then used as the effective address f the operand. We denote the autodecrement mode by putting the specified register in parentheses, preceded by a minus sign to indicate that the contents of register are to be decremented before being used as the effective address. Thus, we write (R4). This mode allows the accessing of operands in the direction of descending addresses. The action performed by the autoincrement and auto decrement addressing modes can be achieved using two instruction, one to access the operand and the other to increment or to decrement the register that contains the operand address. Combining the two operations in one instruction reduces the number if instructions needed to perform the task.	C204.1	BTL1
	If computer A runs a program in 10 seconds and computer B runs the same program in 15 seconds how much faster is A than B? 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	C204.1	BTL1

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

and A is therefore 1.5 times as fast as B.

In the above example, we could also say that computer B is 1.5 times *slower than* computer A, since

$$\frac{\text{Performance}_A}{\text{Performance}_B} = 1.5$$

means that

$$\frac{\text{Performance}_A}{1.5} = \text{Performance}_B$$

	Our favorite program runs in 10 seconds on computer A, which has a 2 GHz clock. We are trying to help a computer designer build a computer, B, which will run this program in 6 seconds. The designer has determined that a substantial increase in the clock rate is possible, but this increase will affect the rest of the CPU design, causing computer B to require 1.2 times as many clock cycles as computer A for this program. What clock rate should we tell the designer to target?	C204.1	BTL1

Let's first find the number of clock cycles required for the program on A:

$$\text{CPU time}_A = \frac{\text{CPU clock cycles}_A}{\text{Clock rate}_A}$$

$$10 \text{ seconds} = \frac{\text{CPU clock cycles}_A}{2 \times 10^9 \frac{\text{cycles}}{\text{second}}}$$

$$\text{CPU clock cycles}_A = 10 \text{ seconds} \times 2 \times 10^9 \frac{\text{cycles}}{\text{second}} = 20 \times 10^9 \text{ cycles}$$

CPU time for B can be found using this equation:

$$\text{CPU time}_B = \frac{1.2 \times \text{CPU clock cycles}_A}{\text{Clock rate}_B}$$

$$6 \text{ seconds} = \frac{1.2 \times 20 \times 10^9 \text{ cycles}}{\text{Clock rate}_B}$$

$$\text{Clock rate}_B = \frac{1.2 \times 20 \times 10^9 \text{ cycles}}{6 \text{ seconds}} = \frac{0.2 \times 20 \times 10^9 \text{ cycles}}{\text{second}} = \frac{4 \times 10^9 \text{ cycles}}{\text{second}} = 4 \text{ GHz}$$

To run the program in 6 seconds, B must have twice the clock rate of A.

	Suppose we have two implementations of the same instruction set architecture. Computer A has a clock cycle time of 250 ps and a CPI of 2.0 for some program, and computer B has a clock cycle time of 500 ps and a CPI of 1.2 for the same program. Which computer is faster for this program and by how much? We know that each computer executes the same number of instructions for the program; let's call this number *I*. First, find the number of processor clock cycles for each computer: $$\text{CPU clock cycles}_A = I \times 2.0$$ $$\text{CPU clock cycles}_B = I \times 1.2$$ Now we can compute the CPU time for each computer: $$\text{CPU time}_A = \text{CPU clock cycles}_A \times \text{Clock cycle time}$$ $$= I \times 2.0 \times 250 \text{ ps} = 500 \times I \text{ ps}$$ Likewise, for B: $$\text{CPU time}_B = I \times 1.2 \times 500 \text{ ps} = 600 \times I \text{ ps}$$ Clearly, computer A is faster. The amount faster is given by the ratio of the execution times: $$\frac{\text{CPU performance}_A}{\text{CPU performance}_B} = \frac{\text{Execution time}_B}{\text{Execution time}_A} = \frac{600 \times I \text{ ps}}{500 \times I \text{ ps}} = 1.2$$ We can conclude that computer A is 1.2 times as fast as computer B for this program.	C204.1	BTL1
	Define CPU execution time and list the types. **CPU execution time** Also called **CPU time**. The actual time the CPU spends computing for a specific task. **Types:** **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	C204.1	BTL1
	Define response time **Response time:** 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.	C204.1	BTL1

	What is Throughput? Also called **bandwidth**. Another measure of performance, it is the number of tasks completed per unit time.	C204.1	BTL1
	Define Clock cycles: All computers are constructed using a **clock that determines when events take place in the hardware.** These discrete time intervals are called **clock cycles** (or **ticks**, **clock ticks**, **clock periods**, **clocks**, **cycles**).	C204.1	BTL1
	Write 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, the clock rate is the inverse of clock cycle time: $$\text{CPU time} = \frac{\text{Instruction count} \times \text{CPI}}{\text{Clock rate}}$$	C204.1	BTL1

	Compile given Two C Assignment Statements into MIPS
	a = b + c;
	d = a – e;
	Answer
	add a, b, c
	sub d, a, e
		C204.1	BTL1

	Compile givenC Assignment Statement into MIPS
	f = (g + h) – (i + j);
	add t0,g,h # temporary variable t0 contains g + h
	add t1,i,j # temporary variable t1 contains i + j
	sub f,t0,t1 # f gets t0 –t1, which is
	(g + h) – (i + j)
		C204.1	BTL1

	Compile givenC Assignment Statement into MIPS g = h + A[8]; **Answer** The first compiled instruction is lw$t0,8($s3) # Temporary reg $t0 gets A[8] add$s1,$s2,$t0 # g = h + A[8]	C204.1	BTL1

	What are the three types of operands in MIPS **1**.word 2.Memory Operands 3.Constant or Immediate Operands	C204.1	BTL1
	Compile givenC Assignment Statement into MIPS A[12] = h + A[8]; **Answer** $t0: lw$t0,32($s3) # Temporary reg $t0 gets A[8] add$t0,$s2,$t0 # Temporary reg $t0 gets h + A[8] sw$t0,48($s3) # Stores h + A[8] back into A[12]	C204.1	BTL1
	Write MIPS To add 4 to register $s3. addi$s3,$s3,4# $s3 = $s3 + 4	C204.1	BTL1
	Define Instruction format A form of representation of an instruction composed of fields of binary numbers.The numeric version of instructions **machine language** and a sequence of such instructions *machine code*.	C204.1	BTL1
	What are the types of instruction format in MIPS 1. *R-type* (for register) or *R-format*. *2.I-type* (for immediate) or *I-format* *3.J-type or Jump*	C204.1	BTL1

	What are the types of instruction in MIPS.(APR/MAY2018) **1. Arithmetic instruction** **2. Data transfer Instruction** **3. Logical Instruction** **4. Conditional Branch Instruction** **5. Unconditional jump Instruction**	C204.1	BTL1
	Compile givenC Statement into MIPS **if (i == j) f = g + h; else f = g – h;** bne $s3,$s4,Else# go to Else if i ≠ j add $s0,$s1,$s2# f = g + h (skipped if i ≠ j)	C204.1	BTL1

	Compile givenC Statement into MIPS
	
	while (save[i] == k)
	i += 1;
	Ans:
	Loop: sll$t1,$s3,2# Temp reg $t1 = i * 4
	add $t1,$t1,$s6# $t1 = address of save[i]
	lw $t0,0($t1)
	# Temp reg $t0 = save[i]
	bne $t0,$s5, Exit
	# go to Exit if save[i] ≠ k
	addi $s3,$s3,1# i = i + 1
	jLoop# go to Loop
	Exit:
		C204.1	BTL1

	State indirect addressing mode give example.(**APR/May 2017**) Indirect Mode. The effective address of the operand is the contents of a register or main memory location, location whose address appears in the instruction. ... Once it's there, instead of finding an operand, it finds an address where the operand is located. LOAD R1, @R2 Load the content of the memory address stored atregister R2 to register R1.	C204.1	BTL1

PART-B

Q. No.	Questions	CO	Bloom's Level

1.	**i)**Discuss in detail about Eight great ideas of computer Architecture.**(8)** ***Page.No:11-13)*** **ii**) Explain in detail about Technologies for Building Processors and Memory **(8))*(Page.No:24-28)***	C204.1	BTL5
2.	Explain the various components of computer System with neat diagram **(16) .(NOV/DEC2014,NOV/DEC2015,APR/MAY 2016,NOV/DEC 2016,APR/MAY2018))** ***(Page.No:16-17)***	C204.1	BTL5
3.	Discuss in detail the various measures of performance of a computer(16) ***(Page.No:28-40)***	C204.1	BTL6
4.	Define Addressing mode and explain the different types of basic addressing modes with an example **(APRIL/MAY2015 ,NOV/DEC2015,APR/MAY 2016,NOV/DEC 2016,APR/MAY2018)** ***(Page.No:116-117)***	C204.1	BTL5
5.	**i)Discuss the Logical operations and control operations of computer (12)** ***(Page.No:87-89)*** **ii)Write short notes on Power wall(6)** ***(Page.No:40-42)***	C204.1	BTL6
6.	Consider three diff erent 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. **(APR/MAY 2018)** 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? **(Refer Notes)**	C204.1	BTL5
7.	Explain various instruction format illustrate the same with an example **NOV/DEC2017** ***(Page.No:80-86)***	C204.1	BTL5
8.	Explain direct ,immediate ,relative and indexed addressing modes with example **APR/MAY2018** ***(Page.No:116-117)***	C204.1	BTL5
9.	State the CPU performance equation and the factors that affect performance (8) **(NOV/DEC2014) (Refer Notes)**	C204.1	BTL5

10.	Discuss about the various techniques to represent instructions in a computer system. **(APRIL/MAY2015,NOV/DEC 2017)** ***(Page.No:80-86)***	C204.1	BTL6
11.	What is the need for addressing in a computer system?Explain the different addressing modes with suitable examples.**(APRIL/MAY2015)** ***(Page.No:116-117)***	C204.1	BTL5
12.	**Explain types of operations and operands with examples.(NOV/DEC 2017)** ***(Page.No:63-70)***	C204.1	BTL5
13.	Consider two diff erent implementations of the same instruction set architecture. Th e instructions can be divided into four classes according to their CPI (class A, B, C, and D). P1 with a clock rate of 2.5 GHz and CPIs of 1, 2, 3, and 3, and P2 with a clock rate of 3 GHz and CPIs of 2, 2, 2, and 2. Given a program with a dynamic instruction count of 1.0E6 instructions divided into classes as follows: 10% class A, 20% class B, 50% class C, and 20% class D, which implementation is faster? a. What is the global CPI for each implementation? b. Find the clock cycles required in both cases. **(Refer Notes)**	C204.1	BTL5
14.	To what should the CPI of load/store instructions be reduced in order for a single processor to match the performance of four processors using the original CPI values? **(Refer Notes)**	C204.1	BTL5
15.	Describe the steps that transform a program written in a high-level language such as C into a representation that is directly executed by a computer processor. **(Refer Notes)**	C204.1	BTL4

stored first. For example, in a little-endian computer, the two bytes required for thehexadecimal number 4F52 would be stored as 524F (52 at address 1000, 4F at 1001).

PART -A

Q. No.	Questions	CO	Bloom's Level
	What is half adder and full adder? A half adder is a logic circuit with two inputs and two outputs, which adds two bits at a time, producing a sum and a carry. A full adder is logic circuit with three inputs and two outputs, which adds three bits at a time giving a sum and a carry.	C204.2	BTL1
	What are the overflow conditions for addition and subtraction.(NOV/DEC 2015) peration 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	C204.2	BTL1
	State the rule for floating point addition. 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.	C204.2	BTL1
	What is signed binary? A system in which the leading bit represents the sign and the remaining bits the magnitude of the number is called signed binary. This is also known as sign magnitude.	C204.2	BTL1
	What is a carry look-ahead adder? The input carry needed by a stage is directly computed from carry signals obtained from all the preceding stages i-1,i-2,…..0, rather than waiting for normal carries to supply slowly from stage to stage. An adder that uses this principle is a called carry look-ahead adder	C204.2	BTL1
	Define Booth Algorithm. 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.	C204.2	BTL6

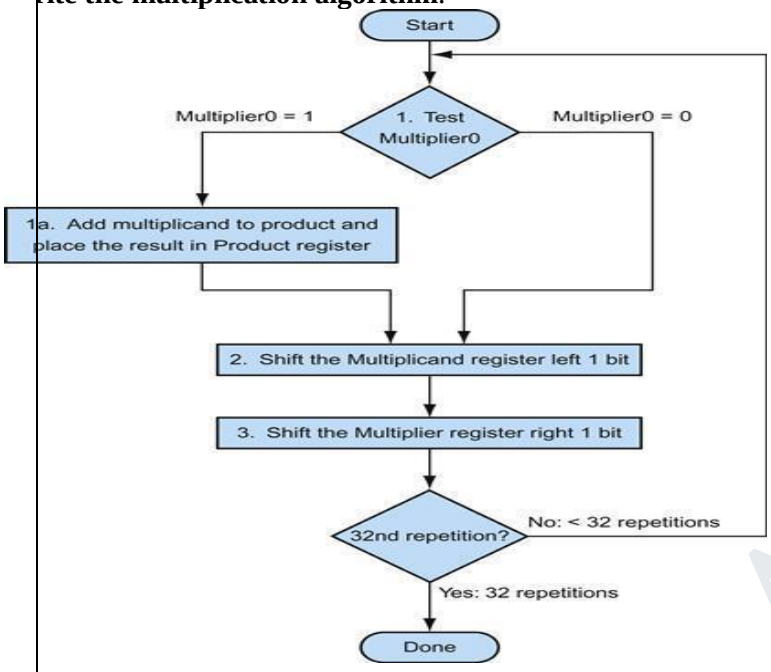
	What are the main features of Booth's algorithm? • It handles both positive and negative multipliers uniformly. • It achieves some efficiency in the number of addition required when the multiplier has a few large blocks of 1s.	C204.2	BTL6
	Define Integer Division and give its rule. s are the set of whole 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 {... -4, -3, -2, -1, 0, 1, 2, 3, 4...}Negative integers: {. . . -4, -3, -2, -1} Positive integers: {1, 2, 3, 4 ...}{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.	C204.2	BTL1
	Define Truncation. To retain maximum accuracy, all extra bits during operation (called *guard bts*) 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	C204.2	BTL1
	Explain how Boolean subtraction is performed? the subtrahend (i.e. in a-b, the subtrahend is b) then perform addition(2's complement).	C204.2	BTL1
	What do you mean by Subword Parallelism? APR/MAY 2015,MAY/JUNE 2016 Given that the parallelism occurs within a wide word, the extensions are classified as sub-word parallelism. It is also classified under the more general name of data level parallelism. They have been also called vector or SIMD, for single instruction, multiple data . The rising popularity of multimedia applications led to arithmetic instructions that support narrower operations that can easily operate in parallel.	C204.2	BTL1
	How can we speed up the multiplication process? There are two techniques to speed up the multiplication process: 1) The first technique guarantees that the maximum number of summands that must be added is n/2 for n-bit operands.	C204.2	BTL1

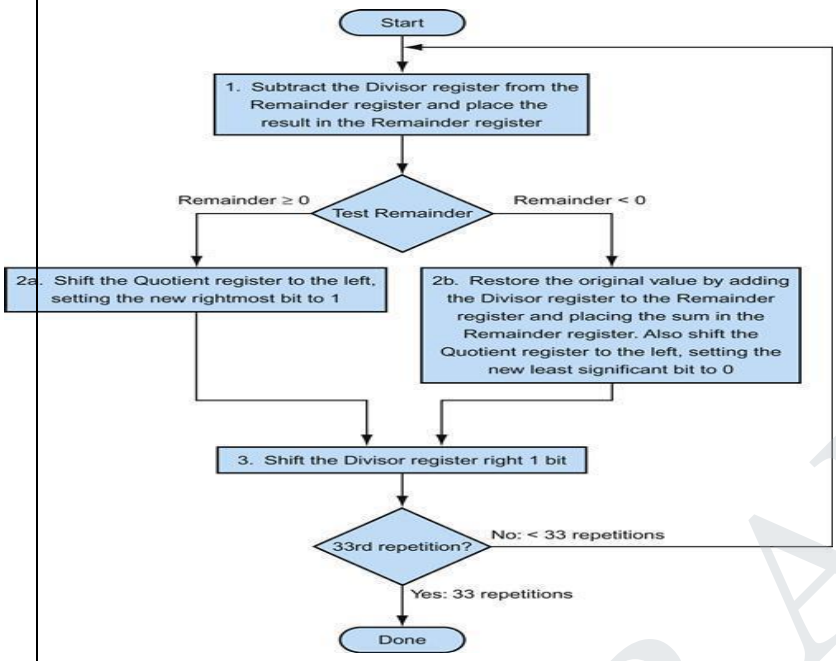
	What is bit pair recoding? Give an example. Bit pair recoding halves the maximum number of summands. Group the Booth-recoded multiplier bits in pairs and observe the following: The pair (+1 -1) is equivalent to to the pair (0 +1). That is instead of adding -1 times the multiplicand m at shift position i to +1 M at position i+1, the same result is obtained by adding +1 M at position i. Eg: 11010 – Bit Pair recoding value is 0 -1 -2	C204.2	BTL1
	What are the two methods of achieving the 2's complement? a. Take the 1's complement of the number and add 1. b. Leave all least significant 0's and the first unchanged and then complement the remaining bits	C204.2	BTL1
	What is the advantage of using Booth algorithm? 1) It handles both positive and negative multiplier uniformly. 2) It achieves efficiency in the number of additions required when the multiplier has a few large blocks of 1's. 3) The speed gained by skipping 1's depends on the data	C204.2	BTL1
	Write the algorithm for restoring division. Do the following for n times: 1) Shift A and Q left one binary position. 2) Subtract M and A and place the answer back in A. 3) If the sign of A is 1, set q0 to 0 and add M back to A. Where A- Accumulator, M- Divisor, Q- Dividend. ***Step 1:*** Do the following for n times: 1) 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. 2) Now, if the sign of A is 0, set q0 to 1; otherwise, set q0 to0. ***Step 2:*** if the sign of A is 1, add M to A.	C204.2	BTL1
	What is Carry Save addition? 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.	C204.2	BTL1

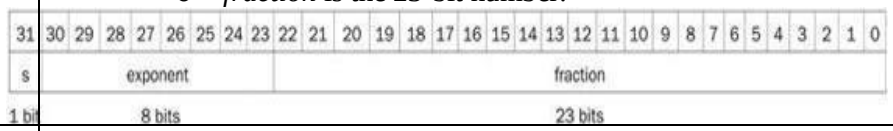
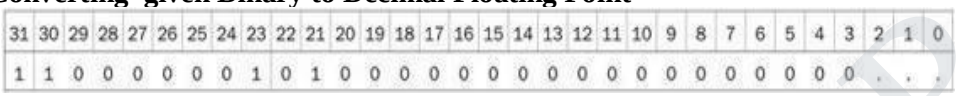
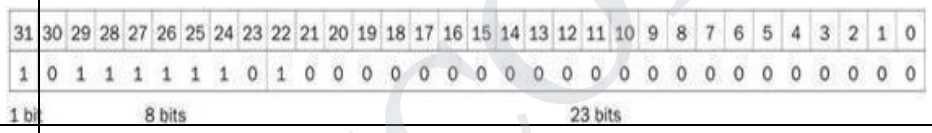
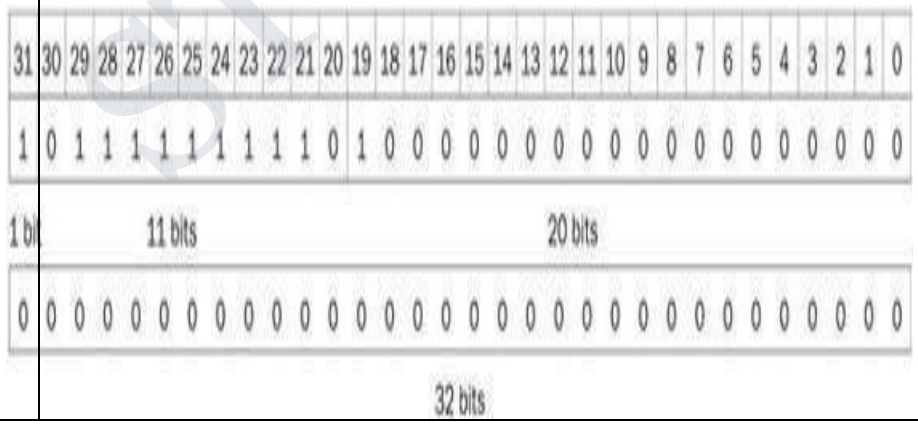
	When can you say that a number is normalized? When the decimal point is placed to the right of the first (nonzero) significant digit, the number is said to be normalized. The end values 0 to 255 of the excess-127 exponent E are used to represent special values such as: a) When E = 0 and the mantissa fraction M is zero the value exact 0 is represented. 1. When E = 255 and M=0, the value is represented. 2. When E = 0 and M 0, denormal values are represented. 3. When E = 2555 and M 0, the value represented is called Not a number.	C204.2	BTL1
	How overflow occur in subtraction?APRIL/MAY2015 If 2 Two's Complement numbers are subtracted, and their signs are different, then overflow occurs if and only if the result has the same sign as the subtrahend. Overflow occurs if • (+A) − (−B) = −C • (−A) − (+B) = +C	C204.2	BTL1
	Write the Add/subtract rule for floating point numbers. 1) Choose the number with the smaller exponent and shift its mantissa right a number of steps equal to the difference in exponents. 2) Set the exponent of the result equal to the larger exponent. 3) Perform addition/subtraction on the mantissa and determine the sign of the result 4) Normalize the resulting value, if necessary.	C204.2	BTL1
	Define ALU. MAY/JUNE 2016 The **arithmetic and logic unit (ALU)** of a computer system is the place where the actual execution of the instructions take place during the processing operations. All calculations are performed and all comparisons (decisions) are made in the **ALU**. The data and instructions, stored in the primary storage prior to processing are transferred as and when needed to the ALU where processing takes place	C204.2	BTL1
	rite the multiply rule for floating point numbers. 1) Add the exponent and subtract 127. 2) Multiply the mantissa and determine the sign of the result 3) Normalize the resulting value, if necessary	C204.2	BTL1

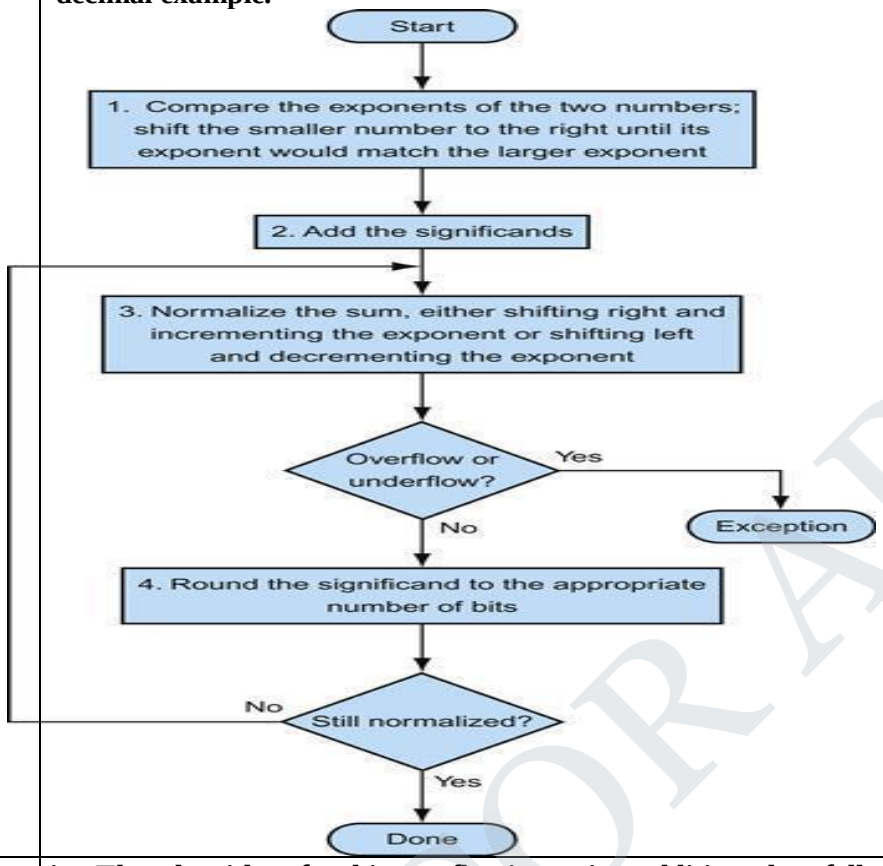
	State double precision floating point number?NOV/DEC 2015 **Double-precision floating-point** format is a computer **number** format that occupies 8 bytes (64 bits) in computer computer memory and represents a wide, dynamic range of values by using a **floating point**	C204.2	BTL1
	hat is excess-127 format? Instead of the signed exponent E, the value actually stored in the exponent field is and unsigned integer E. 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.	C204.2	BTL1
	What is guard bit? Although the mantissa of initial operands are limited to 24 bits, it is important to retain extra bits, called as guard bits.	C204.2	BTL1
	hat are the ways to truncate the guard bits? There are several ways to truncate the guard its: 1) Chopping 2) Von Neumann rounding 3) Rounding	C204.2	BTL1
	hat are generate and propagate function? The generate function is given by Gi=xiyi and The propagate function is given as Pi=xi+yi.	C204.2	BTL1
	In floating point numbers when so you say that an underflow or overflow has occurred? In single precision numbers when an exponent is less than -126 then we say that an underflow has occurred. In single precision numbers when an exponent is less than +127 then we say that an overflow has occurred.	C204.2	BTL3

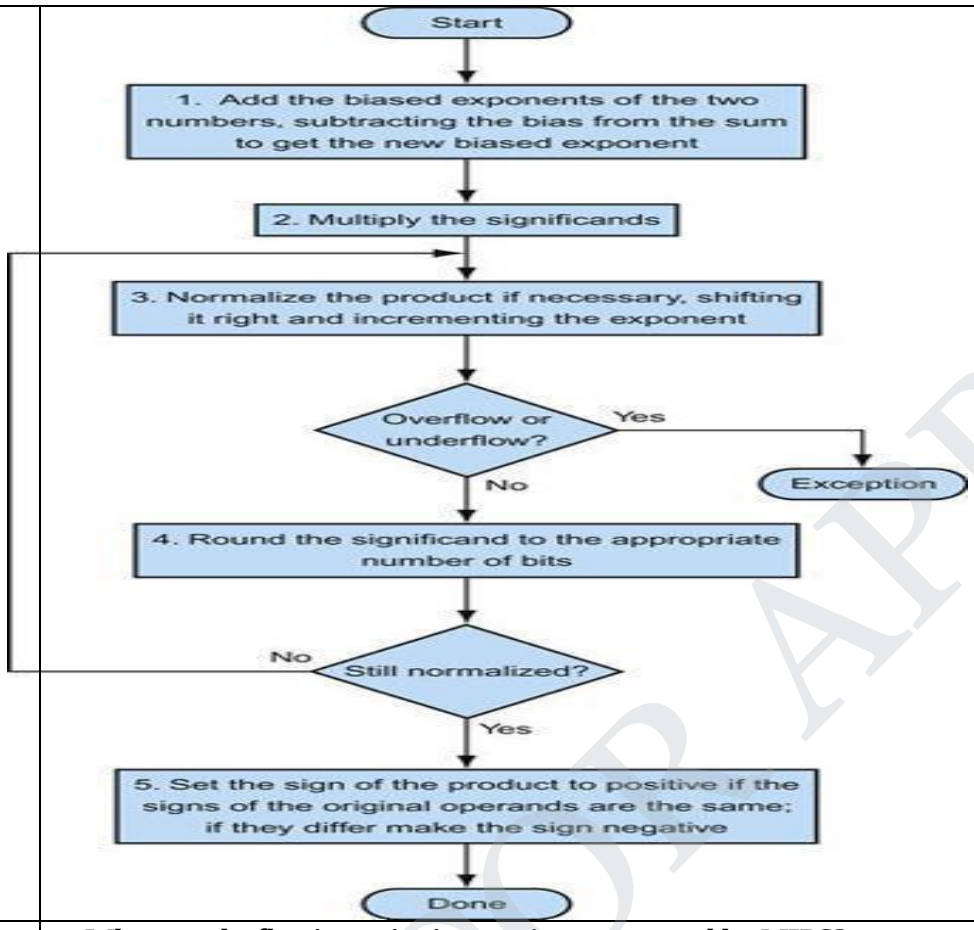
	Define Von Neumann Rounding. ast one of the guard bits is 1, the least significant bit of the retained bits is set to 1 otherwise nothing is changed in retained bits and simply guard bits are dropped. example, ARM added more than 100 instructions in the NEON multimedia instruction extension to support sub-word parallelism, which can be used either with ARMv7 or ARMv8. .Multiply 100010 * 100110.	C204.2	BTL1
	Write the algorithm for restoring division. **n- Restoring Division Algorithm** **p 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. **p 2:** If the Sign of A is 1, add M to A.	C204.2	BTL1
	Define Exception Also called **interrupt**. An unscheduled event that disrupts program execution; used to detect overflow.	C204.2	BTL1
	Define Interrupt An exception that comes from outside of the processor. (Some architectures use the term *interrupt* for all exceptions.)	C204.2	BTL1
	Multiplying 1000_{ten} by 1001_{ten}: Multiplicand 1000_{ten} Multiplier x 1001_{ten} 1000 0000 0000 1000 Product 1001000_{ten}	C204.2	BTL1

	rite the multiplication algorithm.  	C204.2	BTL1
	Divide 1,001,010$_{ten}$ by 1000$_{ten}$: 1001_{ten} Quotient Divisor 1000_{ten} $\overline{)1001010_{ten}}$ Dividend -1000 10 101 1010 -1000 10_{ten} Remainder	C204.2	BTL1

	rite the division algorithm.  	C204.2	BTL1
	DefineScientific notation A notation that renders numbers with a single digit to the left of the decimal point. $0.1_{ten} \times 10^{-8}$	C204.2	BTL1
	Define Normalized notation A number in floating-point notation that has no leading 0s.	C204.2	BTL1
	Define Overflow in floating-point. A situation in which a positive exponent becomes too large to fit in the exponent field.	C204.2	BTL1
	Define Underflow infloating-point A situation in which a negative exponent becomes too large to fit in the exponent field	C204.2	BTL1

	Define Single precision A floating-point value represented in a single 32-bit word. ○ *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. 	C204.2	BTL1
	Converting given Binary to Decimal Floating Point  **Answer** The sign bit is 1, the exponent field contains 129, and the fraction field contains 1×2^{-2}=1/4, or 0.25. Using the basic equation, $(-1)^S \times (1 + \text{Fraction}) \times 2^{(\text{Exponent}-\text{Bias})} = (-1)^1 \times (1 + 0.25) \times 2^{(129-127)}$ $= -1 \times 1.25 \times 2^2$ $= -1.25 \times 4$ $= -5.0$	C204.2	BTL1
	Write The single precision binary representation of -0.75_{ten}**(APR/MAY2018)** 	C204.2	BTL1
	Write double precision binary representation of -0.75_{ten} $(-1)^1 \times (1 + .1000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000_{two}) \times 2^{(1022-1023)}$ 	C204.2	BTL1

	Write The algorithm for binary floating-point addition that follows this decimal example. 	C204.2	BTL1
	rite The algorithm for binary floating-point addition that follows this decimal example	C204.2	BTL1

			
	What are the floating point instructions supported by 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)	C204.2	BTL1
	Define Biased Notation: With the bias being the number subtracted from the normal, unsigned representation to determine the real value. Bias of 127 for single precision, so an exponent of −1 is represented by the bit pattern of the value $-1+127_{ten}$, or $126_{ten}=0111\ 1110_{two}$, and +1 is represented by 1+127, or $128_{ten}=1000\ 0000_{two}$. The exponent bias for double precision is 1023. Biased exponent is $(-1)^S \times (1 + \text{Fraction}) \times 2^{(\text{Exponent} - \text{Bias})}$	C204.2	BTL1

	What are the advantages to represent number in IEEE format: 1. It simplifies exchange of data that includes floating-point numbers; 2. it simplifies the floating-point arithmetic algorithms to know that numbers will always be in this form; and ncreases the accuracy of the numbers that can be stored in a word, since the unnecessary leading 0s are replaced by real digits to the right of the binary point	C204.2	BTL1
	State Rules for floating point addition.(**APR/MAY 2017**) Assume that only four decimal digits of the significand and two decimal digits of the exponent. ***Step 1:*** Align the decimal point of the number that has the smaller exponent ***Step 2:*** addition of the significands: ***Step 3:*** This sum is not in normalized scientific notation, so adjust it: ***Step 4:*** Since the significand can be only four digits long (excluding the sign), we round the number. truncate the number if the digit to the right of the desired point .	C204.2	BTL1

PART-B

Q. No.	Questions	CO	Bloom's Level
1. .	Explain the sequential version of Multiplication algorithm in detail with diagram hardware and examples **(APRIL/MAY2015)** ***(Page.No:183-188)***	C204.2	BTL5
2. .	Discuss in detail about division algorithm in detail with diagram and examples(16)**NOV/DEC15,NOV/DEC 2016,nov/dec 2017,APR/MAY2018 (*Page.No:189-196)***	C204.2	BTL6
3.	Explain how floating point addition is carried out in a computer system.Give example for a binary floating point addition(**APRIL/MAY2015) (*Page.No:203-206)***	C204.2	BTL5
4.	Explain in detail about floating point multiplication**(*Page.No:206-211)***	C204.2	BTL5
5.	Multiply the following pair of signed 2's complement numbers : A = 010111, B = 101100 **(Refer notes.)**	C204.2	BTL2
6.	Add the numbers 0.5 and -0.4375 using binary Floating point Addition algorithm**(NOV/DEC 2017) (Refer notes.)**	C204.2	BTL2
7.	. Multiply 1.10 10 X 101010 and $9.200X10^{-5}$ using binary Floating point multiplication **(Refer notes.)**	C204.2	BTL2
8.	Calculate the division of A and B A : $3.264 X 10^3$ B: $6.52 X 10^2$ **(Refer notes.)**	C204.2	BTL2
9.	Show the IEEE 754 binary representation of the number -0.75 in single and double precision **(Refer notes.)**	C204.2	BTL2

10.	Briefly explain Carry lookahead adder**(NOV/DEC2014)** (6) ***(Page.No:718-719)***	C204.2	BTL2
11.	Multiply the following pair of signed nos.using Booth's bit –pair recoding of the multiplier A=+13 (multiplicand) and b=-6(multiplier) **(NOV/DEC2014)** **(Refer notes.)**	C204.2	BTL2
12.	Discuss in detail about multiplication algorithm with suitable examples and diagram(16)**NOV/DEC 15** ***(Page.No:183-188)***	C204.2	BTL6
13.	Explain briefly about floating point addition and subtraction algorithms.(16)**MAY/JUNE 16** ***(Page.No:203-206)***	C204.2	BTL5
14.	Explain Booth Multiplication algorithm with suitable example(16) **MAY/JUNE2016,NOV/DEC 2016** **(Refer notes)**	C204.2	BTL5
15.	What is the disadvantage of ripple carry addition and how it is overcome in carry look ahead adder and draw the logic circuit CLA.**NOV/DEC 2016** ***(Page.No:708-710)***	C204.2	BTL1

.

PART-A

Q. No.	Questions	CO	Bloom's Level
	What is pipelining? The technique of overlapping the execution of successive instruction for substantial improvement in performance is called pipelining. .	C204.3	BTL1
	What is and precise exception? A precise exception is one in which all instructions prior to the faulting instruction are complete and instruction following the faulting instruction,	C204.3	BTL1

	including the faulty instruction; do not change the state of the machine.		
	Define processor cycle in pipelining. The time required between moving an instruction one step down the pipeline is a processor cycle.	C204.3	BTL1
	What is meant by pipeline bubble?(NOV/DEC 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.	C204.3	BTL1
	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.	C204.3	BTL1
	What are the major characteristics of a pipeline? The major characteristics of a pipeline are: 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. 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	C204.3	BTL6
	What is data path?(NOV/DEC 2016,APR/MAY2018) 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.	C204.3	BTL6
	What is a pipeline hazard and what are its types? Any condition that causes the pipeline to stall is called hazard. They are also called as stalls or bubbles. The various pipeline hazards are: Hazard Control Hazard	C204.3	BTL1
	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.	C204.3	BTL1

	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	C204.3	BTL1
	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	C204.3	BTL1
	What do you mean by branch penalty? The time lost as a result of a branch instruction is often referred to as branch penalty	C204.3	BTL1
	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.	C204.3	BTL1
	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	C204.3	BTL1
	Define exception and interrupt. **Dec 2012,NOV/DEC 14,MAY/JUNE 2016,APR/MAY2018))** **Exception:** The term exception is used to refer to any event that causes an interruption. **Interrupt:** An exception that comes from outside of the processor. There are two types of interrupt. 1. Imprecise interrupt and 2.Precise interrupt	C204.3	BTL1

	Why is branch prediction algorithm needed? Differentiate between the static and dynamic techniques. (May 2013,APR/MAY 2015,NOV/DEC 15) 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. **Static Branch prediction** The static branch prediction, assumes that the branch will not take place and to continue to fetch instructions in sequential address order. **Dynamic Branch prediction** 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.	C204.3	BTL1
	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	C204.3	BTL1
	How do control instructions like branch, cause problems in a pipelined processor? Pipelined processor gives the best throughput for sequenced line instruction. In branch instruction, as it has to calculate the target address, whether the instruction jump from one memory location to other. In the meantime, before calculating the larger, the next sequence instructions are got into the pipelines, which are rolled back, when target is calculated.	C204.3	BTL1
	What is meant by super scalar processor? Super scalar processors are designed to exploit more instruction level parallelism in user programs. This means that multiple functional units are used. With such an arrangement it is possible to start the execution of several instructions in every clock cycle. This mode of operation is called super scalar execution.	C204.3	BTL1
	Define pipeline speedup. [APR/MAY 2012] (A.U.NOV/DEC 2012) Speed up is the ratio of the average instruction time without pipelining to the average instruction time with pipelining. Average instruction time without pipelining Speedup= Average instruction time with pipelining	C204.3	BTL1

	What is Vectorizer? The process to replace a block of sequential code by vector instructions is called vectorization. The system software, which generates parallelism, is called as vectorizing compiler.	C204.3	BTL1
	What is pipelined computer? When hardware is divided in to a number of sub units so as to perform the sub operations in an overlapped fashion is called as a pipelined computer.	C204.3	BTL1
	List the various pipelined processors. 8086, 8088, 80286, 80386. STAR 100, CRAY 1 and CYBER 205 etc	C204.3	BTL1
	Classify the pipeline computers. Based on level of processing → processor pipeline, instruction pipeline, arithmetic pipelines Based on number of functions→ Uni-functional and multi functional pipelines. Based on the configuration → Static and Dynamic pipelines and linear and non linear pipelines Based on type of input→ Scalar and vector pipelines.**Asf**	C204.3	BTL1
	Define Pipeline speedup. (Nov/Dec 2013) The ideal speedup from a pipeline is equal to the number of stages in the pipeline. $$\frac{\text{Time per instruction on unpipelined machine}}{\text{Number of pipe stages}}$$	C204.3	BTL1
	Write down the expression for speedup factor in a pipelined architecture. [MAY/JUNE '11] The speedup for a pipeline computer is S = (k + n -1) tp Where,K → number of segments in a pipeline,N → number of instructions to be executed. Tp → cycle time	C204.3	BTL1

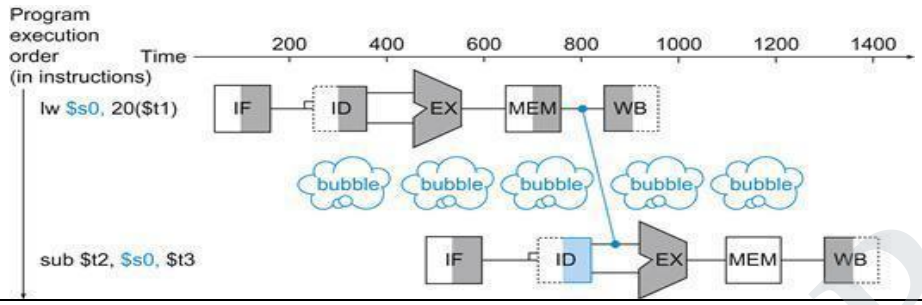
	What are the problems faced in instruction pipeline. Resource conflicts → Caused by access to the memory by two at the same time. Most of the conflicts can be resolved by using separate instruction and data memories. Data dependency → Arises when an instruction depends on the results of the previous instruction but this result is not yet available. Branch difficulties → Arises from branch and other instruction that change the value of PC (Program Counter).	C204.3	BTL1
	What is meant by vectored interrupt? (Nov/Dec 2013) An interrupt for which the address to which control is transferred is determined by the cause of the exception.	C204.3	BTL1
	What is the need for speculation?NOV/DEC 2014 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 it as dependence in executing other instructions. For example, we might speculate on the outcome of a branch, so that instructions after the branch could be executed earlier. Speculation (also known as *speculative loading*), is a process implemented in Explicitly Parallel Instruction Computing (EPIC) processors and their compiler s to reduce processor-memory exchanging bottlenecks or latency by putting all the data into memory in advance of an actual load instruction	C204.3	BTL3
	Define Imprecise , Precise interrupt Imprecise interrupt Also called imprecise exception. Interrupts or exceptions in pipelined computers that are not associated with the exact instruction that was the cause of the interrupt or exception. Precise interrupt Also called precise exception. An interrupt or exception that is always associated with the correct instruction in pipelined computers	C204.3	BTL1
	What are the advantages of pipelining?MAY/JUNE 2016 The cycle time of the processor is reduced; increasing the instruction throughput.Some combinational circuits such as adders or multipliers can be made faster by adding more circuitry. If pipelining is used instead, it can save circuitry vs. a more complex combinational circuit.	C204.3	BTL1
	What is Program counter (PC)(Fetching) The register containing the address of the instruction in the program being executed	C204.3	BTL1

	What is Adder: An adder is needed to compute the next instruction address. The adder is an ALU wired to always add its two 32-bit inputs and place the sum on its output.	C204.3	BTL1
	What is Register file(decoding): A state element that consists of a set of registers that can be read and written by supplying a register number to be accessed.	C204.3	BTL1
	Define Sign-extend in data path. 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. a unit to **sign-extend** the 16-bit offset field in the instruction to a 32-bit signed value	C204.3	BTL1
	Define Shifter: ■ The jump instruction operates by replacing the lower 28 bits of the PC with the lower 26 bits of the instruction shifted left by 2 bits. Simply concatenating 00 to the jump offset accomplishes this shift	C204.3	BTL1
	What is Delayed 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.	C204.3	BTL1
	What are the control lines of MIPS functions. **ALU control lines** — **Function** 0000 — AND 0001 — OR 0010 — add 0110 — sub	C204.3	BTL1

		0111 Set lessthan 1100 NOR		
	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		C204.3	BTL1
	What are the Function of seven control lines?		C204.3	BTL1
	What are the Disadvantages of single cycle implementation? ● Although the single-cycle design will work correctly, it would not be used in modern designs because it is inefficient. ● Although the CPI is 1 the overall performance of a single-cycle implementation is likely to be poor, since the clock cycle is too long. ● The penalty for using the single-cycle design with a fixed clock cycle is significant,. ● To implement the floating-point unit or an instruction set with more complex instructions, this single-cycle design wouldn't work well . ● A single-cycle implementation thus violates the great idea of making the		C204.3	BTL1

Signal name	Effect when deasserted	Effect when asserted
RegDst	The register destination number for the Write register comes from the rt field (bits 20:16).	The register destination number for the Write register comes from the rd field (bits 15:11).
RegWrite	None.	The register on the Write register input is written with the value on the Write data input.
ALUSrc	The second ALU operand comes from the second register file output (Read data 2).	The second ALU operand is the sign-extended, lower 16 bits of the instruction.
PCSrc	The PC is replaced by the output of the adder that computes the value of PC + 4.	The PC is replaced by the output of the adder that computes the branch target.
MemRead	None.	Data memory contents designated by the address input are put on the Read data output.
MemWrite	None.	Data memory contents designated by the address input are replaced by the value on the Write data input.
MemtoReg	The value fed to the register Write data input comes from the ALU.	The value fed to the register Write data input comes from the data memory.

	common case fast.		
	What is Structural hazard? When a planned instruction cannot execute in the proper clock cycle because the hardware does not support the combination of instructions that are set to execute. If there is a single memory instead of two memories. If the pipeline had a fourth instruction, that in the same clock cycle the first instruction is accessing data from memory while the fourth instruction is fetching an instruction from that same memory. Without two memories, pipeline could have a structural hazard. **To avoid structural hazards** ● When designing a pipeline designer can change the design By providing sufficient resources	C204.3	BTL1
	Define Data Hazards. .(**APR/MAY 2017**) Data hazard is also called a **pipeline data hazard**. When a planned instruction cannot execute in the proper clock cycle because data that is needed to execute the instruction is not yet available. ● In a computer pipeline, data hazards arise from the dependence of one instruction on an earlier one that is still in the pipeline ● ***Example:*** add instruction followed immediately by a subtract instruction that uses the sum ($s0): add$s0, $t0, $t1 sub$t2, $s0, $t3	C204.3	BTL1
	Define data Forwarding Forwarding is also called as **bypassing**. 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.	C204.3	BTL1
	Define 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	C204.3	BTL1

	Define Pipeline stall Pipeline stall is also called as **bubble**. A stall initiated in order to resolve a hazard. 	C204.3	BTL1
	What is Control Hazard? Control hazard is also called as **branch hazard**. When the proper instruction cannot execute in the proper pipeline clock cycle because the instruction that was fetched is not the one that is needed; that is, the flow of instruction addresses is not what the pipeline expected.	C204.3	BTL1
	What are theSchemes for resolving control hazards ? 1. Assume Branch Not Taken: 2. Reducing the Delay of Branches: 3. Dynamic Branch Prediction:	C204.3	BTL1
	Define Branch delay slot The slot directly after a delayed branch instruction, which in the MIPS architecture is filled by an instruction that does not affect the branch.	C204.3	BTL1
	Define Correlating , Tournament branch 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	C204.3	BTL1

	Name control signal to perform arithmetic operation.(APR/MAY 2017) 1.Regdst 2.Regwrite 3.ALU Src	C204.3	BTL1
52	what is ideal cycle per instruction in pipelining?(APR/MAY 2018) With pipelining, a new instruction is fetched every clock cycle by exploiting instruction-level parallelism, therefore, since one could theoretically have five instructions in the five pipeline stages at once (one instruction per stage), a different instruction would complete stage 5 in every clock cycle	C204.3	BTL1

PART-B

Q. No.	Questions	CO	Bloom's Level
1. .	Explain the basic MIPS implementation with binary multiplexers and control lines(16) **NOV/DEC 15 (*Page.No:*244-251**)	C204.3	BTL5
2.	What is hazards ?Explain the different types of pipeline hazards with suitable examples**.(NOV/DEC2014,APRIL/MAY2015,MAY/JUNE 2016,NOV/DEC2017) (*Page.No:303*-324**)	C204.3	BTL5
3.	Explain how the instruction pipeline works. What are the various situations where an instruction pipeline can stall? Illustration with an example? **NOV/DEC 2015,NOV/DEC 2016.(*Page.No:301*-302**)	C204.3	BTL5
4.	Explain data path in detail**(NOV/DEC 14,NOV/DEC2017) (*Page.No:251*-259**)	C204.3	BTL5
5.	Explain dynamic branch prediction **.(*Page.No:321*-323**)	C204.3	BTL5
6.	Explain in detail How exceptions are handled in MIPS architecture.**(APRIL/MAY2015) .(*Page.No:*325-332**)	C204.3	BTL5
7.	Explain in detail about building a datapath(**NOV/DEC2014 (*Page.No:251*-259**)	C204.3	BTL5
8.	Explain in detail about control implementation scheme(**APR/MAY 2018) (*Page.No:*259-271**)	C204.3	BTL5

9.	What is pipelining?Discuss about pipelined datapath and control(16)**MAY/JUNE2016 (*Page.No* :286-303)**	C204.3	BTL6
10.	Why is branch prediction algorithm needed?Differentiate between static and dynamic techniques?**NOV/DEC 2016 .(*Page.No:321*-323**)	C204.3	BTL3
11.	Design a simple path with control implementation and explain in detail(**MAY/JUN 2018) (*Page.No:251*-271**)	C204.3	BTL6
12.	Discuss the limitation in implementing the processor path. Suggest the methods to overcome them**(NOV/DEC 2018) (Refer notes)**	C204.3	BTL6
13.	When processor designers consider a possible improvement to the processor datapath, the decision usually depends on the cost/performance trade-off . In the following three problems, assume that we are starting with a datapath where I-Mem, Add, Mux, ALU, Regs, D-Mem, and Control blocks have latencies of 400 ps, 100 ps, 30 ps, 120 ps, 200 ps, 350 ps, and 100 ps, respectively,and costs of 1000, 30, 10, 100, 200, 2000, and 500, respectively.Consider the addition of a multiplier to the ALU. Th is addition will add 300 ps to the latency of the ALU and will add a cost of 600 to the ALU. Th e result will be 5% fewer instructions executed since we will no longer need to emulate the MUL instruction. 1 What is the clock cycle time with and without this improvement? 2 What is the speedup achieved by adding this improvement? 3 Compare the cost/performance ratio with and without this improvement. **(Refer notes)**	C204.3	BTL5
14.	For the problems in this exercise, assume that there are no pipeline stalls and that the breakdown of executed instructions is as follows: add addi not beq lw sw 20% 20% 0% 25% 25% 10% 14.1 In what fraction of all cycles is the data memory used? 14.2 In what fraction of all cycles is the input of the sign-extend circuit needed? What is this circuit doing in cycles in which its input is not needed? **(Refer notes)**	C204.3	BTL3
15.	Consider the following loop. loop:lw r1,0(r1) and r1,r1,r2 lw r1,0(r1) lw r1,0(r1) beq r1,r0,loop Assume that perfect branch prediction is used (no stalls due to control hazards),that there are no delay slots, and that the pipeline has full forwarding support. Also assume that many iterations of this loop are executed before the loop exits. **(Refer notes)**	C204.3	BTL3

.

PART-A

Q. No.	Questions	CO	Bloom's Level
	What is Instruction level parallelism?NOV/DEC 2015,NOV/DEC 2016, .(APR/MAY 2017) ILP is a measure of how many of the operationsin a computer programcan be performed simultaneously. The potential overlap among instructions is called instruction level parallelism	C204.4	BTL1
	What are the various types of Dependences in ILP. ● Data Dependences ● Name Dependences ● Control Dependences	C204.4	BTL1
	What is multiprocessors? Mention the categories of multiprocessors? Multiprocessor is the use of two or more central processing units (CPUs) within a single computer system. It is used to increase performance and improve availability. The different categories are SISD, SIMD, MIMD	C204.4	BTL1
	Define Static multiple issue and Dynamic multiple issue. **Static multiple issue** -An approach to implementing a multiple-issue processor where manydecisions are made by the compiler before execution. **Dynamic multiple issue** -An approach to implementing a multiple-issue processor wheremany decisions are made during execution by the processor.	C204.4	BTL1
	What is Speculation? An approach whereby the compiler or processor guesses the outcome of an instruction to remove it as dependence in executing other instructions	C204.4	BTL1
	Define Use latency. Number of clock cycles between a load instruction and an instruction that can use the result of the load with-out stalling the pipeline	C204.4	BTL6

	What is Loop unrolling? A technique to get more performance from loops that access arrays, in which multiple copies of the loop body are made and instructions from different iterations are scheduled together	C204.4	BTL6
	Define Register renaming. The renaming of registers by the compiler or hardware to remove anti-dependences	C204.4	BTL1
	What is Superscalar and Dynamic pipeline schedule? **Superscalar**-An advanced pipelining technique that enables the processor to execute morethan one instruction per clock cycle by selecting them during execution. **Dynamic pipeline schedule**-Hardware support for reordering the order of instructionexecution so as to avoid stalls.	C204.4	BTL1
	Define Commit unit. The 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	C204.4	BTL1
	What is Reservation station? A buffer within a functional unit that holds the operands and the operation.	C204.4	BTL1
	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	C204.4	BTL1
	Define Out of order execution. A situation in pipelined execution when an instruction blocked from executing does not cause the following instructions to wait	C204.4	BTL1
	What is 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	C204.4	BTL1
	. **Define Strong scaling and weak scaling.** **APRIL/MAY 2015,NOV/DEC2017** **Strong scaling** Speed-up achieved on a multi-processor without increasing the size of the problem. **Weak scaling.** Speed-up achieved on a multi-processor while increasing the size of the problem proportionally to the increase in the number of processors.	C204.4	BTL1

	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 uniprocessormachines like a PC	C204.4	BTL1
	Define Single Instruction, Multiple Data streams(SIMD) and Multiple Instruction,Single Data stream (MISD). **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 processoror GPU.	C204.4	BTL1
	Define Multiple Instruction, Multiple Data streams(MIMD) and Single program multiple data streams . **Multiple Instruction, Multiple Data streams (MIMD)** Multiple autonomous processors simultaneously executing different instructions on different data. Distributed systemsare generally recognized to be MIMD architectures; either exploiting a single shared memory space or a distributed memory space. A multi-coresuperscalar processor is an MIMD processor. **Single program multiple data streams :** Multiple autonomous processors simultaneously executing the same program on different data.	C204.4	BTL1
	Define multithreading.(NOV/DEC2014,NOV/DEC 2016) **Multithreading** is the ability of a program or an operating system to serve more than one user at a time and to manage multiple simultaneous requests without the need to have multiple copies of the programs running within the computer. To support this, central processing units have hardware support to efficiently execute multiple threads	C204.4	BTL1
	What are Fine grained multithreading and Coarse grained multithreading?MAY/JUNE 2016,NOV/DEC2017 **Fine grained multithreading** Switches between threads on each instruction, causing the execution of multiples threads to be interleaved, - Usually done in a round-robin fashion, skipping any stalled	C204.4	BTL1

	threads - CPU must be able to switch threads every clock **Coarse grained multithreading** Switches threads only on costly stalls, such as L2 cache misses		
	What is multiple issue? Write any two approaches. Multiple issue is a scheme whereby multiple instructions are launched in one clock cycle. It is a method for increasing the potential amount of instruction-level parallelism. It is done by replicating the internal components of the computer so that it can launch multiple instructions in every pipeline stage. The two approaches are: 1. Static multiple issue (at compile time) 2. Dynamic multiple issue (at run time)	C204.4	BTL1
	What is meant by speculation? **what is the need for speculation(NOV/DEC2014)** 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 it as dependence in executing other instructions. For example, we might speculate on the outcome of a branch, so that instructions after the branch could be executed earlier. Speculation (also known as *speculative loading*), is a process implemented in Explicitly Parallel Instruction Computing (EPIC) processors and their compiler s to reduce processor-memory exchanging bottlenecks or latency by putting all the data into memory in advance of an actual load instruction	C204.4	BTL1
	Define – Static Multiple Issue Static multiple issue is an approach to implement a multiple-issue processor where many decisions are made by the compiler before execution.	C204.4	BTL1
	What is meant by anti-dependence? How is it removed? Anti-dependence is an ordering forced by the reuse of a name, typically a register, rather than by a true dependence that carries a value between two instructions. It is also called as name dependence. Register renaming is the technique used to remove anti-dependence in which the registers are renamed by the compiler or hardware	C204.4	BTL1
	What is meant by loop unrolling? An important compiler technique to get more performance from loops is loop unrolling, where multiple copies of the loop body are made. After unrolling, there is more ILP available by overlapping instructions from different iterations	C204.4	BTL1

	Differentiate UMA from NUMA. (APRIL/MAY2015) Uniform memory access (UMA)is a multiprocessor in which latency to any word in main memory is about the same no matter which processor requests the access. Non uniform memory access (NUMA) is a type of single address space multiprocessor in which some memory accesses are much faster than others depending on which processor asks for which word.	C204.4	BTL1
	What is flynn'sclassification?NOV/DEC 2014,NOV/DEC2017,APR/MAY 2018) Michael Flynn proposed a classification for computer architectures based on the number of instruction steams and data streams Single Instruction, Single Data stream (SISD) Single instruction, multiple data (SIMD) Multiple instruction, single data(MISD) Multiple instruction, multiple data(MIMD)	C204.4	BTL1
	Define A super scalar processor? NOV/DEC 2015 Super scalar processors are designed to exploit more instruction level parallelism in user programs. This means that multiple functional units are used. With such an arrangement it is possible to start the execution of several instructions in every clock cycle. This mode of operation is called super scalar execution.	C204.4	BTL1
	state the need for Instrucion Level Parallelism?MAY/JUNE 2016 **Instruction-level parallelism** (**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 approaches to instruction level parallelism:Hardware ,Software	C204.4	BTL3
	What are symmetric multi-core processor and asymmetric multi-core processor? A symmetric multi-core processor is one that has multiple cores on a single chip, and all of those cores are identical. Example: Intel Core In an asymmetric multi-core processor, the chip has multiple cores onboard, but the cores might be different designs. Each core will have different capabilities	C204.4	BTL1
	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.	C204.4	BTL1

	Define cluster. Group of independent servers (usually in close proximity to one another) interconnected through a dedicated network to work as one centralized data processing resource. Clusters are capable of performing multiple complex instructions by distributing workload across all connected servers. Clustering improves the system's availability to users, its aggregate performance, and overall tolerance to faults and component failures. A failed server is automatically shut down and its users are switched instantly to the other servers	C204.4	BTL1
	What is ware scale computer? A cluster is a collection of desktop computers or servers connected together by a local area network to act as a single larger computer. A warehouse-scale computer (WSC) is a cluster comprised of tens of thousands of servers	C204.4	BTL1
	What is message passing multiprocessor? Message passing systems provide alternative methods for communication and movement of data among multiprocessors (compared to shared memorymultiprocessor systems). A message passing system typically combines local memory and processor at each node of the interconnection network.	C204.4	BTL1
	What are thePros and Cons of Message Passing Message sending and receiving is much slower than addition, for example • But message passing multiprocessors are much easier for hardware designers to design – Don't have to worry about cache coherency for example • The advantage for programmers is that communication is explicit, so there are fewer "performance surprises" than with the implicit communication in cache-coherent SMPs. – Message passing standard MPI (www.mpi-forum.org) • However, its harder to port a sequential program to a message passing multiprocessor since every communication must be identified in advance. – With cache-coherent shared memory the hardware figures out what data needs to be communicated	C204.4	BTL1
	Differentiate Explicit threads Implicit Multithreading(apr/may2017) **Explicit threads** User-level threads which are visible to the application program and kernel-level threads which are visible only to operating system, both are referred to as explicit threads. **Implicit Multithreading** Implicit Multithreading refers to the concurrent execution of multiple threads extracted from a single sequential program. Explicit Multithreading refers to the concurrent execution of instructions from different explicit threads, either by interleaving instructions from different threads on shared pipelines or by parallel execution on parallel pipelines	C204.4	BTL1

	What are the advantages of Speculation? • Speculating on certain instructions may introduce exceptions that were formerly not present. • Example a load instruction is moved in a speculative manner, but the address it uses is not legal when the speculation is incorrect. • Compiler-based speculation, such problems are avoided by adding special speculation support that allows such exceptions to be ignored until it is clear that they really should occur. • In hardware-based speculation, exceptions are simply buffered until it is clear that the instruction causing them is no longer speculative and is ready to complete; at that point the exception is raised, and nor-mal exception handling proceeds. • Speculation can improve performance when done properly and decrease performance when done carelessly.	C204.4	BTL1
	Define issue packet The set of instructions that issues together in one clock cycle; the packet may be determined statically by the compiler or dynamically by the processor	C204.4	BTL1
	State Very Long Instruction Word (VLIW) A style of instruction set architecture that launches many operations that are defined to be independent in a single wide instruction, typically with many separate opcode fields	C204.4	BTL1
	Define use latency. Number of clock cycles between a load instruction and an instruction that can use the result of the load without stalling the pipeline.	C204.4	BTL1
	Define Name Dependence /Antidependence It is an ordering forced by the reuse of a name, typically a register, rather than by a true dependence that carries a value between two instructions.	C204.4	BTL1
	What are the Advantages of register renaming? Renaming the registers during the unrolling process allows the compiler to move these independent instructions subsequently so as to better schedule the code. The renaming process eliminates the name dependences, while preserving the true dependences. .	C204.4	BTL1

	Write down the difference between this **simple superscalar** and a **VLIW processor**: ● The code, whether scheduled or not, is guaranteed by the hardware to execute correctly. ● The compiled code always run correctly independent of the issue rate or pipeline structure of the processor. ● In some VLIW designs, recompilation was required when moving across different processor models. ● In other static issue processors, code would run correctly across different implementations, but often so poorly.	C204.4	BTL1
	Give examples of each dependence in ILP **Data Dependence** • ReadAfterWrite(RAW) Instruction j triesto read operand beforeiInstrwrites it **I: add r1,r2,r3** **J: sub r4,r1,r3** **anti-dependence** **Instr$_J$ writes operand *before*Instr$_I$reads it** **I: sub r4,r1,r3** **J: add r1,r2,r3** **K: mul r6,r1,r7** **output dependence** Instr$_J$ writes operand *before*Instr$_I$writes it. **I: sub r1,r4,r3** **J: add r1,r2,r3** **K: mul r6,r1,r7**	C204.4	BTL1
	List the Advantages of Dynamic Scheduling 1. It uses hardware-based speculation, especially for branch outcomes. By predicting the direction of a branch, a dynamically scheduled processor can continue to fetch and execute instructions along the predicted path. Because the instructions are committed in order.A speculative, dynamically scheduled pipeline can also support speculation on load addresses, allowing load-store reordering, and using the commit unit to avoid incorrect speculation. 2. Not all stalls are predictable; in particular, cache misses can cause unpredictable stalls. Dynamic scheduling allows the processor to hide some of those stalls by continuing to execute instructions while waiting for the stall to end. 3. If the processor speculates on branch outcomes using dynamic branch prediction, it cannot know the exact order of instructions at compile time, since it depends on the predicted and actual behavior of branches. 4. As the pipeline latency and issue width change from one implementation to another, the best way to compile a code sequence also changes. 5. Old code will get much of the benefit of a new implementation without the need for recompilation.	C204.4	BTL1

	Write down the Speed-up (Performance Improvement) equation. It tells us how much faster a task can be executed using the machine with the enhancement as compare to the original machine. It is defined as Speedup = $\frac{Performance\ for\ entire\ task\ using\ improved\ machine}{Performance\ for\ entire\ task\ using\ old\ machine}$ or Speedup = $Fraction_{enhanced}(F_e)$	C204.4	BTL1
	What are the advantages and disadvantages of SIMD. Advantages of SIMD ● Reduces the cost of control unit over dozens of execution units. ● It has reduced instruction bandwidth and program memory. ● It needs only one copy of the code that is being executed simultaneously. ● SIMD works best when dealing with arrays in 'for' loops. Hence, for parallelism to work in SIMD, there must be a great deal of identically structured data, which is called data-level parallelism. Disadvantages of SIMD ● SIMD is at its weakest in case or switch statements, where each execution unit must perform a different operation on its data, depending on what data it has. ● Execution units with the wrong data are disabled, so that units with proper data may continue. Such situation essentially run at 1/nth performance, where 'n' is the number of cases.	C204.4	BTL1
	Write down the advantages and dis advantages of fine grained multithreading. Advantages: fine-grained multithreading is that it can hide the throughput losses that arise from both short and long stalls, since instructions from other threads can be executed when one thread stalls. Disadvantages: Fine-grained multithreading is that it slows down the execution of the individual threads, since a thread that is ready to execute without stalls will be delayed by instructions from other threads.	C204.4	BTL1
	Write down the advantages and dis advantages of course grained multithreading. Advantages: Advantages: ● coarse-grained multithreading is much more useful for reducing the penalty of high-cost stalls Disadvantages: ● Coarse-grained multithreading is limited in its ability to overcome throughput losses, especially from shorter stalls. ● This limitation arises from the **pipeline** start-up costs of coarse-grained multithreading. Because a processor with coarse-grained multithreading issues instructions from a single thread, when a stall occurs, the pipeline must be emptied or frozen. The new thread that begins executing after the stall must fill the pipeline before instructions will be able to complete	C204.4	BTL1

	What are the Advantages SMT. • Simultaneous Multithreaded Architecture is superior in performance to a multiple-issue multiprocessor (multiple-issue CMP). • SMP boosts utilization by dynamically scheduling functional units among multiple threads. • SMT also increases hardware design flexibility. • SMT increases the complexity of instruction scheduling. • With register renaming and dynamic scheduling, multiple instructions from independent threads can be issued without regard to the dependences among them; the resolution of the dependences can be handled by the dynamic scheduling capability. • Since you are relying on the existing dynamic mechanisms, SMT does not switch resources every cycle. Instead, SMT is always executing instructions from multiple threads, leaving it up to the hardware to associate instruction slots and renamed registers with their proper threads.	C204.4	BTL1
	What are the advantages and disadvantages of multicore processor? advantages The proximity of multiple CPU cores on the same die allows the cache coherency circuitry to operate at a much higher clock rate than is possible if the signals have to travel off-chip. Combining equivalent CPUs on a single die significantly improves the performance of cache snoop (alternative: Bus snooping) operations. Put simply, this means that signals between different CPUs travel shorter distances, and therefore those signals degrade less. These higher-quality signals allow more data to be sent in a given time period, since individual signals can be shorter and do not need to be repeated as often. Disadvantages Maximizing the usage of the computing resources provided by multi-core processors requires adjustments both to the operating system (OS) support and to existing application software. Also, the ability of multi-core processors to increase application performance depends on the use of multiple threads within applications	C204.4	BTL1

PART-B

Q. No.	Questions	CO	Bloom's Level
1.	Explain Instruction level parallel processing state the challenges of parallel processing.**(NOV/DEC2014,APR/MAY2018)** **-(*Page.No:620*-625**)	C204.4	BTL5
2.	Explain the difficulties faced by parallel processing programs**(APR/MAY 2018) (*Page.No:*-625-631**)	C204.4	BTL2
3.	Explain shared memory multiprocessor with a neat diagram?**NOV/DEC 2016 (*Page.No:*-517-521**)	C204.4	BTL5
4.	Explain in detail Flynn's classification of parallel hardware **(NOV/DEC 2015,MAY/JUNE 2016,NOV/DEC**	C204.4	BTL5

	2016,NOV/DEC2017) (*Page.No:*-634-640)		
5.	Explain cluster and other Message passing Multiprocessor (**Refer notes.)**	C204.4	BTL5
6.	Explain in detail about hardware Multithreading**(NOV/DEC2015,MAY/JUNE2016) (*Page.No:*-631-640**)	C204.4	BTL5
7.	Explain Multicore processors**(NOV/DEC2014,MAY/JUNE2016) (*Page.No:*-517-521**)	C204.4	BTL5
8.	Explain the different types of multithreading **(*Page.No:*-631-640**)	C204.4	BTL5
9.	What is hardware Multithreading?compare and contrast Fine grained Multi-Threading and coarse grained multithreading**(APRIL/MAY2015,APR/MAY 2018) (Refer notes.)**	C204.4	BTL1
10.	Discuss about SISD,MIMD,SIMD,SPMD and VECTOR SYSTEM(16) **APRIL/MAY2015 (*Page.No:*-509-517**)	C204.4	BTL6
11.	Brief about cluster and its application. **(Refer notes.)**	C204.4	BTL4
12.	Explain in detail about data warehouse and its application. **(Refer notes.)**	C204.4	BTL5
13.	Explain about different types of message passing multiprocessor. **(Refer notes.)**	C204.4	BTL5
14.	Explain in detail about SMT**(NOV/DEC 2017) (*Page.No:*-515-517**)	C204.4	BTL5
15.	Classify shared memory multiprocessor based on memory latency**(MAY/JUN 2018) (*Page.No:*-624-630**)	C204.4	BTL2

.

PART -A

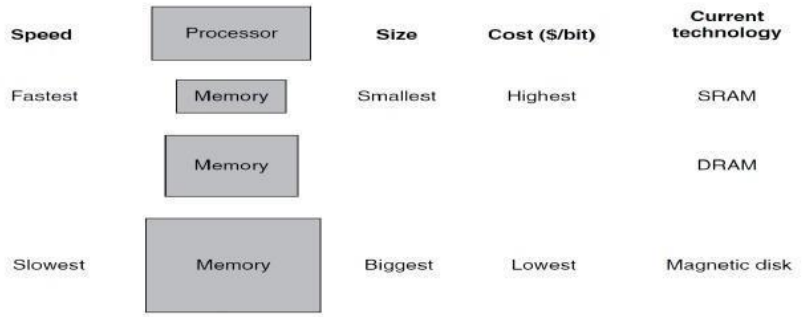
Q. No.	Questions	CO	Bloom's Level
	What is principle of locality? The principle of locality states that programs access a relatively small portion of their address space at any instant of time	C204.5	BTL1

	Define spatial locality. The locality principle stating that if a data location is referenced, data locations with nearby addresses will tend to be referenced soon.	C204.5	BTL1
	Define Memory Hierarchy.(MAY/JUNE 2016) A structure that uses multiple levels of memory with different speeds and sizes. The faster memories are more expensive per bit than the slower memories.	C204.5	BTL1
	Define hit ratio. (A.U.APR/MAY 2013,NOV/DEC 2015) When a processor refers a data item from a cache, if the referenced item is in the cache, then such a reference is called Hit. If the referenced data is not in the cache, then it is called Miss, Hit ratio is defined as the ratio of number of Hits to number of references. Hit ratio =Total Number of references	C204.5	BTL1
	What is TLB? What is its significance? Translation look aside buffer is a small cache incorporated in memory management unit. It consists of page table entries that correspond to most recently accessed pages. Significance The TLB enables faster address computing. It contains 64 to 256 entries	C204.5	BTL1
	Define temporal locality. The principle stating that a data location is referenced then it will tend to be referenced again soon. .	C204.5	BTL6
	How cache memory is used to reduce the execution time. (APR/MAY'10) If active portions of the program and data are placed in a fast small memory, the average memory access time can be reduced, thus reducing the total execution time of the program. Such a fast small memory is called as cache memory.	C204.5	BTL6
	Define memory interleaving. (A.U.MAY/JUNE '11) (apr/may2017) In order to carry out two or more simultaneous access to memory, the memory must be partitioned in to separate modules. The advantage of a modular memory is that it allows the interleaving i.e. consecutive addresses are assigned to different memory module	C204.5	BTL1

	Define Hit and Miss? (DEC 2013) 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	C204.5	BTL1
	What is cache memory?NOV/DEC 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	C204.5	BTL1
	What is memory system? [MAY/JUNE '11] [APR/MAY 2012] Every computer contains several types of devices to store the instructions and data required for its operation. These storage devices plus the algorithm-implemented by hardware and/or software-needed to manage the stored information from the memory system of computer	C204.5	BTL1
	What is Read Access Time? [APR/MAY 2012] A basic performance measure is the average time to read a fixed amount of information, for instance, one word, from the memory. This parameter is called the read access time	C204.5	BTL1
	What is the necessary of virtual memory? State the advantages of virtual memory? MAY/JUNE 2016 Virtual memory is an important concept related to memory management. It is used to increase the apparent size of main memory at a very low cost. Data are addressed in a virtual address space that can be as large as the addressing capability of CPU. Virtual memory is a technique that uses main memory as a "cache" for secondary storage. Two major motivations for virtual memory: to allow efficient and safe sharing of memory among multiple programs, and to remove the programming burdens of a small, limited amount of main memory	C204.5	BTL1
	What are the units of an interface? (Dec 2012) DATAIN, DATAOUT, SIN, SOUT	C204.5	BTL1
	Distinguish between isolated and memory mapped I/O? (May 2013) 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	C204.5	BTL1

	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 is used to trfer data to or from an I/O device. **I/O mapped I/O:** Here the I/O devices the memories have different address space. It has special I/O instructions. The advantage of a separate I/O address space is that I/O devices deals with fewer address lines.	C204.5	BTL1
	Define virtual memory.(nov/dec 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	C204.5	BTL1
	What is Semi Random Access? Memory devices such as magnetic hard disks and CD-ROMs contain many rotating storage tracks. If each track has its own read write head, the tracks can be accessed randomly, but access within each track is serial. In such cases the access mode is semi random.	C204.5	BTL1
	What is the use of DMA? (Dec 2012)(Dec 2013,APR/MAY2018) DMA (Direct Memory Access) provides I/O transfer of data directly to and from the memory unit and the peripheral.	C204.5	BTL1
	Mention the advantages of USB. (May 2013) The Universal Serial Bus (USB) is an industry standard developed to provide two speed of operation called low-speed and full-speed. They provide simple, low cost and easy to use interconnect system.	C204.5	BTL1
	What is meant by vectored interrupt?(Dec 2013) Vectored Interrupts are type of I/O interrupts in which the device that generates the interruptrequest (also called IRQ in some text books) identifies itself directly to the processor	C204.5	BTL1

<table>
<tr>
<td></td>
<td>Compare Static RAM and Dynamic RAM.(Dec 2013,APR/MAY2018)
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. 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.
SRAMs are simply integrated circuits that are memory arrays with a single access port that can provide either a read or a write. SRAMs have a fixed access time to any datum.
SRAMs don’t need to refresh and so the access time is very close to the cycle time. SRAMs typically use six to eight transistors per bit to prevent the information from being disturbed when read. SRAM needs only minimal power to retain the charge in standby mode.
In a dynamic RAM (DRAM), the value kept in a cell is stored as a charge in a capacitor. A single transistor is then used to access this stored charge, either to read the value or to overwrite the charge stored there. Because DRAMs use only a single transistor per bit of storage, they are much denser and cheaper per bit than SRAM
DRAMs store the charge on a capacitor, it cannot be kept indefinitely and must periodically be refreshed.</td>
<td>C204.5</td>
<td>BTL1</td>
</tr>
<tr>
<td></td>
<td>what is DMA <u>?(NOV/DEC 2014)</u>
Direct memory access (DMA) is a method that allows an input/output (I/O) device to send or receive data directly to or from the main memory, bypassing the CPU to speed up memory operations. The process is managed by a chip known as a DMA controller (DMAC).</td>
<td>C204.5</td>
<td>BTL1</td>
</tr>
<tr>
<td></td>
<td>Differentiate programmed I/O and interrupt i/O..(NOV/DEC2014)
<table>
<tr><th>programmed I/O</th><th>interrupt i/O</th></tr>
<tr><td>Programmed IO is the process of IO instruction written in computer program</td><td>Interrupt Initiated IO is done by using interrupt and some special command.</td></tr>
<tr><td>In Programmed IO technique to transfer data,required constant motoring on peripheral by CPU,once data transfer is initiated, CPU have to wait for next transfer.</td><td>In Interrupt Initiated IO once data transfer initiated ,CPU execute next program without wasting time and the interface keep monitoring the device.</td></tr>
</table></td>
<td>C204.5</td>
<td>BTL1</td>
</tr>
<tr>
<td></td>
<td>what is the purpose of dirty /modified bit in cache memory.(NOV/DEC2014)
A dirty bit or modified bit is a bit that is associated with a block of computer memory and indicates whether or not the corresponding block of memory has been modified.[1] The dirty bit is set when the processor writes to (modifies) this memory. The bit indicates that its associated block of memory has been modified and has not yet been saved to storage.</td>
<td>C204.5</td>
<td>BTL1</td>
</tr>
</table>

	What is the need to implement memory as a hierarchy? (APRIL/MAY2015)  **The basic structure of a memory hierarchy.**	C204.5	BTL1
	Point out how DMA can improve I/O speed? APRIL/MAY 2015 CPU speeds continue to increase, and new CPUs have multiple processing elements on the same chip.A large amount of data can be processed very quickly Problem in the transfer of data to CPU or even memory in a reasonable amount of time so that CPU has some work to do at all time . Without DMA, when the CPU is using programmed input/output, it is typically fully occupied for the entire duration of the read or write operation, and is thus unavailable to perform other work. With DMA, the CPU first initiates the transfer, then it does other operations while the transfer is in progress, and it finally receives an interrupt from the DMA controller when the operation is done.	C204.5	BTL1
	What are the various memory Technologies?NOV/DEC 2015 **Memory Technologies** Main memory is implemented from DRAM (dynamic random access memory), while levels closer to the processor (caches) use SRAM (static random access memory). DRAM is less costly per bit than SRAM, although it is substantially slower. The price difference arises because DRAM uses significantly less area per bit of memory, and DRAMs thus have larger capacity for the same amount of silicon; **Memory technology** — **Typical access time** — **$ per GiB in 2012** SRAM semiconductor memory — ›–2.5 ns — 00–$1000 DRAM semiconductor memory — –70 ns — 0–$20 Flash semiconductor memory —)00–50,000 ns — .75–$1.00 Magnetic disk —)00,000–20,000,000 ns — .05–$0	C204.5	BTL1
	What is flash memory? Flash memory is a type of electrically erasable programmable read-only memory (EEPROM). Unlike disks and DRAM, EEPROM technologies can wear out flash memory bits. To cope with such limits, most flash products include a controller to spread the writes by remapping blocks that have been written many times to less trodden blocks. This technique is called wear leveling.	C204.5	BTL3

	In many computers the cache block size is in the range 32 to 128 bytes. What would be the main Advantages and disadvantages of making the size of the cache blocks larger or smaller? Larger the size of the cache fewer be the cache misses if most of the data in the block are actually used. It will be wasteful if much of the data are not used before the cache block is moved from cache. Smaller size means more misses	C204.5	BTL1
	Define USB. Universal Serial Bus, an external busstandard that supports data transfer ratesof 12 Mbps. A single USB portcan be used to connect up to 127 peripheral devices, such as mice, modems, and keyboards. USB also supportsPlug-and-Play installationandhot plugging. .	C204.5	BTL1
	Define Memory latency **The amount of time it takes to transfer a word of data to or from the memory.**	C204.5	BTL1
	Define Memory bandwidth **Tthe number of bits or bytes that can be transferred in one second. It is used to measure how much time is needed to transfer an entire block of data.**	C204.5	BTL1
	Define miss Rate. The miss rate (1−hit rate) is the fraction of memory accesses not found in the upper level.	C204.5	BTL1
	Define Hit rate. Hit rate⊇ The fraction of memory accesses found in a level of the memory hierarchy. •	C204.5	BTL1
	Define miss rate. Miss rate⊇ The fraction of memory accesses not found in a level of the memory hierarchy.	C204.5	BTL1
	Define Hit time. Hit time 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	C204.5	BTL1

	Define miss penalty The miss penalty 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	C204.5	BTL1
	Define tag in TLB Tag⊇ A field in a table used for a memory hierarchy that contains the address information required to identify whether the associated block in the hierarchy corresponds to a requested word.	C204.5	BTL1
	What are the steps to be taken on 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	C204.5	BTL1
	What is write through cache 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.	C204.5	BTL1
	What is write back cache In a write back scheme, when a write occurs, the new value is written only to the block in the cache.	C204.5	BTL1
	What are the techniques to improve cache performance? Two different techniques for improving cache performance. • One focuses on reducing the miss rate by reducing the probability that two different memory blocks will participate for the same cache location. • The second technique reduces the miss penalty by adding an additional level to the hierarchy. This technique, called multilevel caching	C204.5	BTL1
	Define dirty bit dirty bit is commonly used. This status bit indicates whether the block is dirty (modified while in the cache) or clean (not modified).	C204.5	BTL1
	What is 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.	C204.5	BTL1

	What are the messages transferred in DMA? To initiate the transfer of a block of words , the processor sends, i) Starting address ii) Number of words in the block iii)Direction of transfer.	C204.5	BTL1
	Define Burst mode. Burst Mode: The DMA controller may be given exclusive(limited) access to the main memory to transfer a block of data without interruption. This is known as Burst/Block Mode. •	C204.5	BTL1
	Define bus master Bus Master: The device that is allowed to initiate data transfers on the bus at any given time is called the bus master	C204.5	BTL1
	Define bus arbitration. Bus Arbitration: It is the process by which the next device to become the bus master is selected and the bus mastership is transferred to it.	C204.5	BTL1
	What are the approaches for bus arbitration? There are 2 approaches to bus arbitration. They are i)Centralized arbitration (A single bus arbiter performs arbitration) ii)Distributed arbitration (all devices participate in the selection of next bus master).	C204.5	BTL1

PART -B

Q. No.	Questions	CO	Bloom's Level
1.	Explain in detail about memory Technologies**(APRIL/MAY2015,NOV/DEC2017) (*Page.No:*-378-383**)	C204.5	BTL5
2.	Expain in detail about memory Hierarchy with neat diagram **(*Page.No:*-374-378**)	C204.5	BTL5
3.	Discuss the various mapping schemes used in cache memory(**NOV/DEC2014) (*Page.No:*-383-397**)	C204.5	BTL6
4.	Discuss the methods used to measure and improve the performance of the cache**.(NOV/DEC 2017) (*Page.No:*-398-417**)	C204.5	BTL6
5.	Explain the virtual memory address translation and TLB with necessary diagram**.(APRIL/MAY2015,NOV/DEC 2015,NOV/DEC 2016,APR/MAY2018) (*Page.No:*-427-452**)	C204.5	BTL5

6.	Draw the typical block diagram of a DMA controller and explain how it is used for direct data transfer between memory and peripherals. **(NOV/DEC 2015,MAY/JUNE 2016,NOV/DEC 2016,MAY/JUN 2018)** ***Page.No:*****-399-402**)	C204.5	BTL5
7.	Explain in detail about interrupts with diagram ***(Page.No:*****-436-242**)	C204.5	BTL5
8.	Describe in detail about programmed Input/Output with neat diagram **(MAY/JUN 2018) (Refer notes)**	C204.5	BTL5
9.	Explain in detail about the bus arbitration techniques**.(NOV/DEC2014)**(8) ***(Page.No:*****-237-242**)	C204.5	BTL5
10.	Draw different memory address layouts and brief about the technique used to increase the average rate of fetching words from the main memory (8)(**NOV/DEC2014)** **(Refer notes)**	C204.5	BTL5
11.	Explain in detail about any two standard input and output interfaces required to connect the I/O devices to the bus.(**NOV/DEC2014)** ***(Page.No:*****-438-452**)	C204.5	BTL5
12.	Explain mapping functions in cache memory in cache memory to determine how memory blocks are placed in cache **(Nov/Dec 2014) (Refer notes)**	C204.5	BTL5
13.	Explain the various mapping techniques associated with cache memories **(MAY/JUNE 2016,MAY/JUN 2018)** **(Refer notes)**	C204.5	BTL5
14.	Explain sequence of operations carried on by a processor when interrupted by a peripheral device connected to it**(MAY/JUN 2018)** ***(Page.No:*****-436-242**)	C204.5	BTL5
15.	Explain virtual memory and the advantages of using virtual memory ***(Page.No:*****-427-252**)	C204.5	BTL5