Ex/CSE/PC/B/T/215/2025

B.E. Computer Science and Engineering 2nd Year 1st Semester Examination 2025

Subject: Computer Organisation

Time: 3hrs **Full Marks:** 100

Answer all

Answers of all sub-parts of a question must be in ***adjacent*** locations

1	(a) (i) Give a detailed **diagram** of the IAS computer including all the important registers. (ii) Write instructions, with comments, using a (A) **1-address machine** and (B) **0-address machine** to compute the following expression: 2*(X-Y) + 2*(X-Y) + 3*(X+Y) + 3*(X+Y). You may try to optimize the number of instructions, if possible. Before doing that, replace X with Y with your name's first and second letters, respectively (if your name is ANIRUDH, then X=A, and Y=N). (iii) What are advantages of **Absolute** and **Relative** addressing modes? (iv) What is the use of the **Frame Pointer**? **OR** (b) (i) What is the role of a **parser** and **scanner** during compilation of a program? (ii) What are the **functions** of Program Counter, Memory Buffer Register, Accumulator, and Instruction Register? (iii) Give examples of **Address Modification** instructions used in the IAS computer. (iv) What is the possible reason for **failure** of Fetch and Decode of an instruction? (iv) Write a **subroutine**, with proper commands, to calculate this: **A = (2*X + 2*Y)**. Now show how we can use **stack** to pass the parameters to the subroutine and to store the return address.	4+ (4 +4) + 2+1 2+4 +1 + 2+6
2	(a) (i) Give a suitable example of the **non-restoring type division** algorithm. Let's do **A/B**, where **A≠B**, A and B are any two non-zero digits of your examination roll number. Mention all steps clearly. (ii) Give an example of a **floating point multiplication** of two binary numbers using the IEEE 754 single precision format (if required, do the re-normalization the results). Define all the steps for clear understanding of the process. (iii) Define **smallest** and **largest** magnitudes of normalized numbers in IEEE 754 single precision format. **OR** (b) (i) Give a suitable example of **multiplication** using the **Booth's** algorithm. Let's do **(-AxB)**, where **A≠B**, A and B are any two non-zero digits of your examination roll number. Mention all steps clearly. (ii) Give the block (circuit) diagram of **Floating point adder/subtractor.** (iii) How many bits are required to define the **Mantisa** and **Exponent** of floating point numbers (binary) in IEEE 754 single precision and double precision formats?	6+7 +2 6+7 +2

3	**<u>You can answer either (a) or (b), and either (d) or (e). However, (c) is mandatory.</u>** (a) (i) Consider a system with a main memory access time of **100ns** supported by a cache memory having a **30ns** access time and a hit rate of **95%**. Suppose access to cache and main memory occurs **concurrently**. What would be the effective access time (**EAT**) for this two-level memory (consider **overlapping** access)? What happen if the memory accesses **do not overlap**? (ii) With the help of a circuit diagram, discuss with working principle of a **static RAM** cell. **OR** (b) (i) Discuss the working principle of **direct memory mapping** approach. Give a hypothetical example to understand the cache miss and cache hit using this approach. (ii) What is the **limitation** of this approach? (iii) How does associative and set-associative approaches **overcome** the limitation you mentioned? (c) (i) Consider the last 6 digits of your examination roll number (X=110022), and reverse the number (Y=220011). Now from Y, form Z, where you will subtract 1 from each digit of Y (Z=119900). Now form XZY=110022119900220011. This is your reference string i.e., data reside in main memory. Now consider that in the cache memory there are **3 lines**. Apply two page replacement algorithms, least recently used (**LRU**) and least frequently used (**LFU**), separately to calculate the number of cache hits/misses. (ii) Which memory mapping technique **does not require** page replacement and why? (iii) Mention one **advantage** and one **disadvantage** of the **Write Back** cache update policy. (iv) Give the block diagram of the **Pentium-IV cache memory** organization, where all three levels of cache memory, i.e., L1, L2 and L3, have been used. (**d**) (i) How does the **read head** work in the magnetic disc? (ii) Discuss the **pros** and **cons** of two different disk layouts: Constant Angular Velocity and Multiple Zone Recording. (iii) Define the **disk performance parameters**: seek time, rotational delay, access time and transfer time. **OR** (e) (i) Describe the working principle of solid state device (SSD). Consider the discussion using **transistor level**. (ii) Mention some **typical shortcomings** of SSD. (iii) Give **solutions** to the problems you have **mentioned above**.	4+6 7+1 +2 (4+ 5) +1 +2 +3 2+4 +4 5+2 +3
4	(a) (i) Discuss three **implementation strategies** of the hardwired control unit. (ii) What is the role of **CMAR, µPC** and **µIR** registers in the micro-programmed control unit? (iii) With an example show the utility of polyphase timing of micro-instructions.	9+3 +3

	OR (b) (i) What are the **key roles** of a control unit? (ii) What are the **advantages** and **disadvantages** of the hardwired control unit in comparison with the micro-programmed control unit? (iii) Which **type of micro-instructions** provide better parallelism and **how**? (iv) How does the nano-program control unit **reduce the memory space** required to store the micro-instructions in comparison with the micro-programmed control unit? After the formulation, also give an **example.** (v) Do you find any **limitations** of this type of control unit?	1+3 + 2+7 +2
5	**<u>You can answer either (a) or (b). However, (c) is mandatory.</u>** (a). (i) Give a brief description of different **device identification strategies** used input/output organization. (ii) What are different **protocols** used in Thunderbolt? How do they **work together**? **OR** (b). Write a short note on Direct Memory Access (DMA). Your discussion should include different **configurations** of DMA and an example of a DMA **controller.** (c). (i) Let's consider that **186301** is the last 6 digits of your examination roll number. Now, design a reservation table for a **4-stage** pipeline architecture (**Fetch-Decode-Execute-Write**), and execute a minimum of **<u>6 instructions</u>** (I_1 **to** I_6). Scan your roll number from **right to left**, and add delays using the following rule: if the digit is an **odd** number, add **1** delay for that instruction, otherwise add **two** delays (you can add delays at **any** of the four stages of the pipeline). For example, for the given roll number, one and two delays will be added for I_1 and I_2, respectively. If the roll number (considering all 6 digits) is an **odd** number, I_3 would be a branch instruction, otherwise I_4 would be the branch instruction. Consider that this architecture is having an instruction pre-fetch queue, so that any cache miss for the instruction hazard can be easily solved. For each delay you will add, mention the **hazard type** due to which it got delayed (in this problem, you have to add **all three types** of hazards). (ii) How does **operand forwarding** help to avoid the data hazard? (iii) Do you find any **limitations** if we handle data hazard using Software instead of Hardware?	6+4 10 6+3 +1

Detail of COs:
CO1 and CO2: Q1 (15)
CO3: Q2 (15)
CO4: Q3 (35)
CO5: Q4 (15)
CO6: Q5 (20)