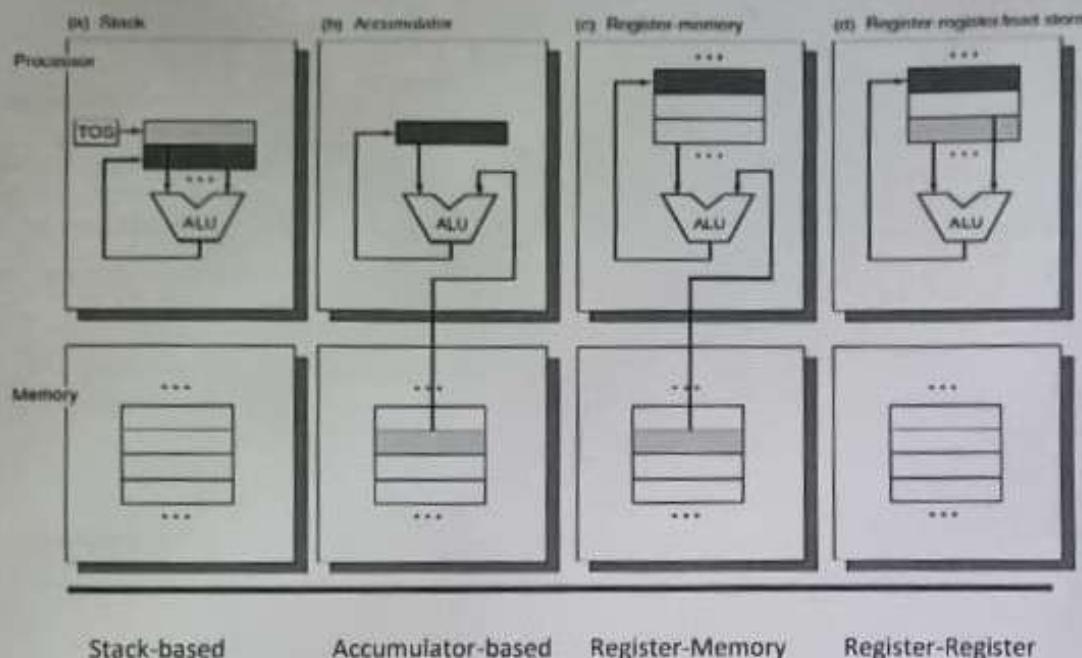


Name:

Reh

Department:



- ✓ 5. A software application is executed on a single-core processor, taking a total execution time of 100 seconds. Within this execution, a segment that lasts 40 seconds is completely sequential and cannot be parallelized, meaning it must always run on a single core, regardless of the number of processors available. The remaining 60 seconds of the execution, however, can be parallelized and executed simultaneously on multiple processing cores. Now, consider a scenario where this parallelizable portion runs on a 4-core system, where each core operates with perfect efficiency, meaning there is no overhead due to synchronization, communication, or scheduling issues. Given these conditions, determine the speedup achieved by using the 4-core system compared to the single-core execution time. [5]

8. True or False? [5]

 - i. Changing the microarchitecture of a processor always require rewriting software that runs on it.
 - ii. Different processors with the same ISA can have different microarchitectures.
 - iii. Immediate addressing mode fetches data from memory at runtime.
 - iv. Clock cycle time does not depend on hardware technology nodes such as 90nm, 180nm, etc.
 - v. Instruction count of a process is independent of ISA and Compiler Technology.

7. Convert the following arithmetic expression into a dataflow graph: [5]
 $X = (A + B) * (C - D) + (E/F) \cdot G$

- c. A hypothetical processor, designed for optimal computational efficiency, is tasked with executing a staggering 1 billion instructions over a precise time span of 2 seconds while operating at a clock speed of 2 GHz. Given these parameters, determine the IPC, a crucial metric that evaluates the processor's ability to execute instruction. [3]

Name:

Roll:

Department:

CSL7070: Computer Architecture

Minor, IIT Jodhpur, Instructor: Palash Das, Date: 22.02.2025

Time: 2-hours

Full Marks: 40

Attempt all questions. Keep your answers brief and to the point. Avoid including unnecessary information, as it may lead to negative marking. Use the back side of the answer sheet for rough work, and ensure your final answers are written clearly and legibly. If any assumptions are made, state them explicitly. Read each question carefully before answering. Good luck!

1. (a) Explain the key differences between open-row and closed-row DRAM scheduling policies.
(b) For each of the following scenarios, determine which scheduling policy is more suitable and justify your answer:

- AI-driven neural network inference performing large matrix multiplications.
- High-frequency trading systems that process financial transactions with unpredictable memory access patterns.

[2+3]

2. Consider the execution of the following MIPS instruction:

SW R1, 20(R190)

Explain, with a diagram, the components of the MIPS datapath involved in executing this instruction.

[5]

3. A MIPS-based processor, designed to follow a reduced instruction set computing (RISC) architecture, operates with a clock rate of 2.5 GHz, meaning it completes 2.5 billion clock cycles per second, which plays a crucial role in determining execution speed and overall performance. The processor exhibits a Cycles Per Instruction (CPI) of 1.5 for memory operations, implying that any instruction involving memory access, such as load (LW) and store (SW) operations, requires an average of 1.5 clock cycles to complete. Additionally, for arithmetic and logical operations, which include basic computations like addition, subtraction, multiplication, and bitwise operations, the CPI is 2 cycles per instruction, suggesting that these operations take slightly longer to execute compared to load-store instructions, possibly due to data dependencies, ALU usage, or pipeline stalls. Furthermore, the processor is equipped with only one core, meaning it lacks parallel execution capabilities, thereby restricting its ability to handle multiple threads or execute instructions concurrently, which could be a potential bottleneck for performance-intensive applications. Unlike modern multi-core processors that can distribute workloads across several execution units, this single-core design requires all instructions to be executed sequentially, with no possibility of parallel processing, making optimization techniques such as instruction pipelining and branch prediction even more critical to enhancing efficiency.

return_X() is a function that computes the following:

$$X = (A + B) * (C - D) + E$$

If the function return_X() is executed 10 million times on the given system, calculate the total CPU time required, assuming no cache optimizations. Show the step-by-step computation.

[5]

4. List down the key features of each of the following machines.

[5]