

➤ Part A (2 Marks) – HIGH-REPEAT QUESTIONS

These are the *only* 2-mark questions that reappear across multiple years.

Most Repeated (appeared 3–4 times)

1. What is pipelining?
2. Differentiate data hazards and control hazards.
3. What is cache memory / Why do we need cache?
4. Define program counter.
5. List the types of addressing modes.
6. Draw / define Von Neumann architecture.
7. Write the characteristic table of T flip-flop.
8. Compare Mealy and Moore models.
9. What is a multiplexer?
10. What is memory hierarchy?

Repeated (appeared 2 times)

11. Explain register vs register indirect addressing mode.
12. Why do we need DMA? / Which signal indicates transfer completion?
13. What is SATA?
14. Define write-through vs write-back.
15. What are data transfer instructions?
16. What is hit time?
17. What is a direct-mapped cache?
18. What is a shift register? / List its types.
19. Define half-adder sum and carry expressions.
20. Distinguish sequential logic and combinational logic.

This is the true core Part A list.

➤ Part B (13 Marks) – HIGH-REPEAT QUESTIONS

These questions appeared **almost every year in some variation**, so they form the *true exam backbone*.

Strongest Repeats (appeared 3–4 times)

1. Von Neumann Architecture – diagram + explanation
2. Addressing modes – any five with examples
3. Pipelining hazards – data vs control + mitigation
4. Full Adder design / Full Adder using two Half Adders
5. JK, D, T flip-flop explanation: diagram + characteristic/excitation tables
6. Synchronous counter design (Mod-5 or Mod-7)
7. DMA operation with neat diagram
8. K-Map simplification for given minterms
9. Decoder and Encoder design

Secondary Repeats (appeared 2 times)

10. Binary to Gray code conversion
11. 4-bit magnitude comparator ($A > B$, $A = B$, $A < B$)
12. Shift register design (4-bit)
13. Instruction cycle explanation
14. Control Unit – Hardwired vs Microprogrammed
15. Simple MIPS data path with control – ALU instruction execution
16. Virtual memory – page translation diagram
17. Interrupt-driven I/O
18. Memory hierarchy – explanation with diagram
19. Cache mapping techniques

These appear reliably but not every year.

➤ Part C (15 Marks) – HIGH-REPEAT LONG QUESTIONS

Part C has the *heaviest repetition*.

Most Common Part-C Questions (Repeated Every Year)

1. Design a Mod-n (5 or 7) synchronous counter using JK flip-flops
2. Implement a Boolean function using 8×1 MUX
3. K-Map: derive both SOP and POS expressions for given minterms

Also Repeated (2 times)

4. Design a decoder (3×8) using AND gates and inverters

Only these four appear repeatedly in Part C.

➤ Direct Mapping (Question → Years):

Question	2022	2023-May	2023-ND	2024	Repeats
Half Adder / Full Adder	✓	✓	—	—	2×
Comparator	—	✓	—	✓	2×
MUX (8×1) / Boolean via MUX	—	✓	—	✓	2×
Decoder / Encoder	✓	✓	✓	—	3×
K-map simplification	✓	✓	✓	—	3×
JK / D / T flip-flops	✓	✓	✓	✓	4×
Counters (Mod-N, BCD)	✓	✓	✓	✓	4×
Mealy vs Moore	✓	✓	✓	✓	4×
Von Neumann	✓	✓	✓	✓	4×
Addressing modes	✓	✓	✓	✓	4×
Instruction cycle	✓	✓	✓	—	3×
Pipelining definition	✓	✓	✓	✓	4×
Hazards (data/control)	✓	✓	✓	✓	4×
Hardwired vs Microprogrammed	✓	✓	—	✓	3×
DMA	✓	✓	✓	✓	4×
Memory hierarchy	✓	✓	✓	✓	4×
Virtual memory translation	✓	—	✓	—	2×

➤ Unit-Wise Question Bank (Full, Exam-Repeat Based)

This includes ALL questions that appeared across **2022–2024** and those directly derived from syllabus.

UNIT I — Combinational Logic

1. Construct a Half Adder and Full Adder with circuit diagrams.
 2. Design a Full Adder using two Half Adders.
 3. Derive $A > B$, $A < B$, $A = B$ expressions for a 2-bit/4-bit comparator.
 4. Draw the 4-bit magnitude comparator and explain its outputs.
 5. Design a 3×8 decoder using AND gates and inverters.
 6. Construct an octal-to-binary encoder with logic diagram.
 7. Design a priority encoder using logic gates.
 8. Explain binary-to-octal decoder and octal-to-binary encoder.
 9. Implement the given Boolean function using 8×1 MUX.
 10. Design 8×1 MUX and show function implementation.
 11. Simplify the function using K-map for given minterms.
 12. Obtain both SOP and POS from a K-map.
 13. Hazard-free realization of Boolean function (AND-OR network).
 14. Explain binary-to-gray code conversion with example.
-

UNIT II — Sequential Logic

1. Explain SR latch using NOR gates with truth/function table.
2. Describe JK flip-flop with diagram, characteristic & excitation table.
3. Describe D flip-flop—circuit, characteristic & excitation tables.
4. Describe T flip-flop—operation & tables.
5. Realize D flip-flop using SR flip-flop.
6. Design D flip-flop using two D latches and inverter.
7. Explain Mealy and Moore models with diagrams.

8. Compare Mealy vs Moore.
 9. Explain 4-bit shift register with diagram.
 10. Design 4-bit down counter.
 11. Design a synchronous Mod-5 counter using JK flip-flops.
 12. Design a synchronous Mod-7 counter using JK flip-flops.
 13. Design a BCD ripple counter using JK flip-flops.
 14. Explain design procedure of clocked sequential circuits.
-

UNIT III — Computer Fundamentals

1. Draw and explain Von Neumann architecture.
 2. List functional units of a digital computer.
 3. Describe 5 addressing modes with examples.
 4. Explain addressing modes with examples.
 5. Explain instruction cycle with diagram.
 6. Discuss instruction sequencing with examples.
 7. Discuss branching and instruction sequencing.
 8. Explain instruction encoding and instruction types.
 9. Explain interaction between assembly and high-level languages.
 10. What is ISA? Discuss its components.
-

UNIT IV — Processor

1. What is pipelining? Explain the pipeline stages.
2. Differentiate data hazard and control hazard with examples.
3. Describe methods to avoid control hazards.
4. Explain techniques used to mitigate pipeline hazards.
5. Outline control unit and its functions.
6. Hardwired vs Microprogrammed control—compare.
7. Draw simple MIPS datapath and explain ALU instruction execution.

8. What are pipeline hazards? Explain types.
 9. Solve pipeline performance (branch instructions not overlapped).
 10. Define essential hazards and explain significance.
-

 **UNIT V — Memory & I/O**

1. Explain memory hierarchy with diagram.
 2. Explain direct, associative, set-associative cache mapping.
 3. Explain virtual-to-physical address translation.
 4. Explain virtual memory organization with page translation.
 5. Describe DMA operation with diagram.
 6. Explain DMA transfer modes.
 7. How memory mapping techniques locate cache blocks?
 8. Explain address translation, segmentation, page table, swap space.
 9. Explain interrupt-driven I/O with diagram.
 10. Describe USB and SATA standards.
 11. What is SATA? State its purpose.
-

 **PART C – HIGH-CHANCE LONG QUESTIONS (Repeated every year)**

1. Design Mod-5 or Mod-7 synchronous counter using JK FFs.
2. Implement a Boolean function using 8×1 multiplexer.
3. K-map: derive SOP & POS for a given minterm list.
4. Design a 3×8 decoder using AND + NOT gates.
5. Design a BCD ripple counter using JK flip-flops.