COA Important Questions PDF (Unit-wise)

The most important COA questions are bus organisation and register transfer, addressing modes, Booth's multiplication, restoring and non-restoring division, IEEE 754 floating point, hardwired versus microprogrammed control, instruction pipelining and hazards, cache mapping (direct, associative, set-associative) with numericals, virtual memory, DMA and interrupt-driven I/O.

Key takeaways

- Booth's algorithm, IEEE 754 conversion, cache mapping and pipeline speed-up are the most common numericals.
- Draw block diagrams for the control unit, DMA controller and memory hierarchy.
- Learn the formulas for speed-up, hit ratio and average access time.
- These are practice questions, not past papers; check your own syllabus and official previous papers.

About these important questions

This page lists the most important Computer Organization and Architecture questions for B.Tech CSE and IT exams, arranged unit by unit. The units follow the standard syllabus that most Indian technical universities share, so the order may differ slightly from yours.

Please note: these are practice questions based on common syllabus topics. They are not verified past papers, and we do not claim that any question will appear in your exam. Check your own university syllabus first, then solve the official previous year papers. AKTU students can download them from the AKTU old question paper archive.

Unit 1: Introduction and Register Transfer

Covers functional units, bus structures, register transfer language and micro-operations.

Short questions (2 marks)

- List the functional units of a digital computer.
- What is a bus? Differentiate between a single bus and a multiple bus structure.
- What is register transfer language (RTL)?

Long questions

1. Explain the basic functional units of a computer and how they interact, with a diagram.

2. Explain bus arbitration: daisy chaining, polling and independent requests.
3. Explain arithmetic, logic and shift micro-operations with examples.
4. Design a 4-bit arithmetic circuit or a common bus system using multiplexers.
5. Explain the stack organisation of a CPU, and register stack versus memory stack.

Unit 2: Arithmetic and Logic Unit

Covers adders, multiplication, division and floating-point representation. This unit is heavy on numericals.

Short questions (2 marks)

- Why is a carry look-ahead adder faster than a ripple carry adder?
- Represent -13 in 1's and 2's complement form.
- What is a guard bit?

Long questions

1. Multiply (-7) × (3) using Booth's algorithm and show each step in a table. *Hint:* write the multiplier with an extra Q(-1) bit and apply the 10, 01, 00/11 rules before each arithmetic right shift.
2. Divide 11 by 3 using the restoring and non-restoring division algorithms.
3. Represent a decimal number such as -12.625 in IEEE 754 single and double precision.
4. Explain the design of a carry look-ahead adder with equations.
5. Explain floating-point addition and subtraction with a flowchart.

Unit 3: Control Unit and Instruction Set

Covers instruction formats, addressing modes, instruction cycle and control unit design.

Short questions (2 marks)

- Differentiate between RISC and CISC.
- What is a micro-instruction?
- Name four addressing modes.

Long questions

1. Explain the addressing modes with an example of the effective address for each.
2. Explain zero-, one-, two- and three-address instructions, and evaluate X = (A + B) × (C + D) using each.
3. Explain the instruction cycle (fetch, decode, execute) with a flowchart.
4. Compare hardwired and microprogrammed control units with diagrams.
5. Explain horizontal and vertical micro-programming.
6. Explain the types of interrupts and how the CPU handles them.

Unit 4: Memory Organisation

Covers memory hierarchy, main memory, cache and virtual memory.

Short questions (2 marks)

- Differentiate between SRAM and DRAM.
- Define hit ratio.
- What is locality of reference?

Long questions

1. Explain the memory hierarchy with a diagram and the reason it works.
2. Explain direct, associative and set-associative cache mapping. For a given cache and main memory size, find the tag, index (set) and offset bits.
3. A system has a cache access time of 10 ns, main memory of 100 ns and a hit ratio of 0.9. Find the average access time.
4. Explain cache write policies (write-through and write-back) and replacement policies.
5. Explain virtual memory, paging and the role of the TLB.
6. Explain memory interleaving and associative memory.

Unit 5: Input-Output Organisation and Pipelining

Covers I/O interfaces, data transfer modes and pipelining. Some universities put pipelining in Unit 3.

Short questions (2 marks)

- What is DMA?
- What is an I/O processor?
- Define pipeline speed-up.

Long questions

1. Compare programmed I/O, interrupt-driven I/O and DMA.
2. Explain the DMA controller and its data transfer modes (burst, cycle stealing).
3. Explain asynchronous data transfer: strobe control and handshaking.
4. Explain instruction pipelining and its hazards (structural, data and control), with ways to reduce each.
5. A non-pipelined processor takes 50 ns per task; a 5-stage pipeline has 10 ns stages. Find the speed-up for 100 tasks. *Hint:* pipelined time is $(k + n - 1) \times t$.
6. Explain Flynn's classification (SISD, SIMD, MISD, MIMD).

Sample answer outline: Cache mapping numerical

A typical question: main memory is 64 KB, the cache is 2 KB and the block size is 16 bytes. Find the address format for direct and 2-way set-associative mapping.

1. **Find the address size.** 64 KB = 2^16 bytes, so the address is 16 bits.
2. **Find the offset bits.** A 16-byte block needs 4 bits.
3. **Direct mapping.** The cache has 2 KB / 16 = 128 lines, so 7 index bits. Tag = 16 - 7 - 4 = 5 bits. Format: tag 5, line 7, word 4.

4. **2-way set-associative.** 128 lines / 2 = 64 sets, so 6 set bits. Tag = 16 - 6 - 4 = 6 bits. Format: tag 6, set 6, word 4.
5. **Fully associative.** No index: tag = 16 - 4 = 12 bits.
6. **Explain the trade-off.** Direct mapping is fastest and cheapest but suffers conflict misses. Associative mapping reduces misses but needs more comparators.

Draw the address split as a box divided into fields. Examiners give marks for this diagram.

Marks weightage and how to write answers

A common pattern, used by AKTU and many other universities, has a compulsory short-answer section that covers every unit, followed by long-answer questions with a choice inside each unit. Marks per question change between universities and years, so confirm the scheme on a recent official paper.

- **Short answers (about 2 marks):** a crisp definition, one example or diagram, and nothing more. Three to five lines is enough.
- **Long answers (7 to 10 marks):** start with a definition, add a labelled diagram or table, explain the steps, and finish with an example, advantages or a comparison.
- For Booth's and division questions, always show a step table with the registers A, Q, Q(-1) and M and the operation in each row.
- Underline the final answer in IEEE 754 and write the sign, exponent and mantissa fields separately.

A 7-day revision plan for COA

1. **Days 1 to 5:** one unit a day. Read your notes, then write the long answers for that unit from memory and check them against your notes.
2. **Day 6:** redo one numerical each of Booth's algorithm, restoring division, IEEE 754, cache mapping and pipeline speed-up.
3. **Day 7:** solve one full official previous year paper in three hours, then check which units cost you time.

Keep a single sheet of formulas, definitions and diagrams for the last hour before the exam.

Frequently asked questions

Which COA topics are most important?

Booth's multiplication, division algorithms, IEEE 754, addressing modes, hardwired versus microprogrammed control, cache mapping, DMA and pipelining are asked most often.

How do I remember Booth's algorithm rules?

Look at the last bit of Q and the extra bit Q(-1). If they are 10, subtract M from A; if 01, add M to A; if 00 or 11, do nothing. Then do an arithmetic right shift. Repeat once for every bit of the multiplier.

Is COA useful beyond the exam?

Yes. Cache, memory and pipelining concepts help with performance work, systems programming and GATE preparation.

Are these COA questions from previous year papers?

No. They are practice questions built from topics that appear in most B.Tech Computer Organization and Architecture syllabi. Use them with your own university's official previous papers, for example the AKTU archive at https://aktu.ac.in/old-question-paper.html.

Is this PDF free?

Yes. Fill in the short form on this page and the PDF downloads right away.

Sources and further reading

- AKTU: old question papers (official archive)
- NPTEL: online courses from IITs (search for Computer Organization and Architecture)
- MIT OpenCourseWare: Computation Structures (6.004)

About this PDF

These are practice questions for revision, prepared by the Uncodemy trainer team from common B.Tech syllabus topics. They are not past papers, a certificate or a promise of marks. Spot an error? Write to info@uncodemy.com and the team will review it.

More free resources and courses: uncodemy.com | info@uncodemy.com | +91 98183 66550