

Architecture and Organization (AR)

(adapted from the Final Report of the Computing Curricula 2001 project,
a joint undertaking of the IEEE-CS and the ACM)

The computer lies at the heart of computing. Without it most of the computing disciplines today would be a branch of theoretical mathematics. To be a professional in any field of computing today, one should not regard the computer as just a black box that executes programs by magic. All students of computing should acquire some understanding and appreciation of a computer system's functional components, their characteristics, their performance, and their interactions. There are practical implications as well. Students need to understand computer architecture in order to structure a program so that it runs more efficiently on a real machine. In selecting a system to use, they should be able to understand the tradeoff among various components, such as CPU clock speed vs. memory size.

Topics:

1. Bits, bytes, and words
2. Numeric data representation and number bases
3. Fixed- and floating-point systems
4. Signed and twos-complement representations
5. Representation of nonnumeric data (for example, character codes)
6. Representation of arrays
7. Basic organization of the von Neumann machine
8. Control unit; instruction fetch, decode, and execution
9. Instruction sets and types (data manipulation, control, I/O)
10. Assembly/machine language programming
11. Instruction formats
12. Addressing modes
13. Subroutine call and return mechanisms
14. I/O and interrupts
15. Storage systems and their technology
16. Memory hierarchy
17. Main memory organization and operations
18. Latency, cycle time, bandwidth, and interleaving
19. Cache memories (address mapping, block size, replacement and store policy)
20. Virtual memory (page table, TLB)

Learning objectives:

1. Explain the reasons for using different formats to represent numerical data.
2. Explain how negative integers are stored in sign-magnitude and twos-complement representation.
3. Convert numerical data from one format to another.
4. Discuss how fixed-length number representations affect accuracy and precision.
5. Describe the internal representation of nonnumeric data.
6. Describe the internal representation of characters, strings, records, and arrays.

7. Explain the organization of the classical von Neumann machine and its major functional units.
8. Explain how an instruction is executed in a classical von Neumann machine.
9. Summarize how instructions are represented at both the machine level and in the context of a symbolic assembler.
10. Explain different instruction formats, such as addresses per instruction and variable length vs. fixed length formats.
11. Write simple assembly language program segments.
12. Demonstrate how fundamental high-level programming constructs are implemented at the machine-language level.
13. Explain how subroutine calls are handled at the assembly level.
14. Explain the basic concepts of interrupts and I/O operations.
15. Identify the main types of memory technology.
16. Explain the effect of memory latency on running time.
17. Explain the use of memory hierarchy to reduce the effective memory latency.
18. Describe the principles of memory management.
19. Describe the role of cache and virtual memory.