

COP 3402 Fundamentals of Computer Systems

Credits: 3

Syllabus: Overview of computer systems organization. Data representation. Boolean and digital logic. Instruction set architecture. Assembly language.

Prerequisite: COP 2210 or equivalent.

Corequisite: MAD 2104 (this course must have been either taken before or enrolled in the current term)

Course outline: (number of lectures shown in parentheses)

- Virtual machine concept (0.5)
- Data representation (1.5)
 - signed and unsigned binary integers
 - binary and hexadecimal arithmetic
 - character representation
 - floating-point binary representation
- Boolean and digital logic (3)
 - simple boolean operations
 - boolean functions and truth tables
 - transistors and digital gates
 - multiplexers and decoders
 - D-latches
 - registers
- Instruction set architecture (5)
 - machine language
 - instruction format
 - operands, opcodes
 - addressing modes
 - data path, instruction execution cycle
 - memory allocation and management
 - memory read cycle
 - registers and flags
 - microcomputer components
 - memory types
 - pipelining and superscalar concepts
 - input-output system
 - interrupts and traps
- Basic elements of assembly language (4)
 - assembling, linking, and running programs
 - defining data and constants
 - data transfer instructions
 - addition and subtraction
 - status flag usage
 - counting loops
 - addressing modes
 - indirect addressing
 - declaring and calling procedures

- Stack frames (3)
 - basic stack operations
 - local variables
 - stack parameters
 - recursive functions
- Conditional processing (2)
 - boolean and comparison instructions
 - conditional jumps
 - IF and WHILE programming patterns
- Advanced arithmetic (2)
 - bit shifting and rotation
 - multiplication and division algorithms
 - extended precision arithmetic
 - multiplication and division instructions
 - generating code from arithmetic expressions
- Assembling and linking (3)
 - introduction to macros
 - two-pass assembly details
 - linking and loading

Optional Topics (3)

- constructing and verifying error-correcting codes
- macro programming
- array manipulation
- parsing text with finite-state machines
- keyboard input
- video programming