

## COMP 2280 - Introduction to Computer Systems

**Calendar Description:** Data representation and manipulation, machine-level representation of programs, assembly language programming, and basic computer architecture.

**Prerequisites:** COMP 2130 and COMP 2140 and COMP 2160.

Lab Required

**This course is a prerequisite for:** COMP 3090, COMP 3290, COMP 3370, and COMP 3720

### Outline

1. Introduction to course (1/2 week)
2. Data representation (1 week)

Signed and unsigned numbers, One and two's complement representation, Addition and subtraction of two's complement numbers, Conversion from binary to decimal and from decimal to binary, Hexadecimal representation, Logical operations (AND, OR, NOT, XOR), Sign extension and overflow, Fixed point representation i.e. IEEE floating point representation j. ASCII
3. Introduction to ISA and assembly language (1 week)

Introduce instruction set architecture, Discuss high level design of ISA, Introduce the ISA of machine that we will be using, Introduce assembly language and machine code, The assembly process-Two passes, Symbol tables, Linking and loading, Examples
4. Introduction to assembly language instructions of machine that we will be using (1 week)

Introduce the instructions of the machine, examples
5. Introduction to circuits (2 weeks)

Discuss basic logic gates, Truth tables, Discuss useful combinational circuits and their implementations – Multiplexers, Decoders, Adders, Sequential circuits - Latches and flip-flops (RS and D), Registers, Memory (address space and organization, building a memory, examples, f) introduction to timing diagrams
6. Addressing modes and Control Statements (1/2 week)

Introduce conditional branching, Addressing modes, Examples for machine we are using
7. The CPU (2 weeks)

Components of the CPU, Memory transfer - MAR and MDR registers, Program execution in CPU, The CPU state machine for executing instructions (fetch, decode, etc.), Micro-operations, Structure of CPU - Internal buses, Control points, Tri-state buffers, Important operations and illustrate how to set the appropriate control points - Read from memory, Write to memory, Register transfer, ALU operations, Control unit implementation methods- Hardwire logic, Micro-coding
8. Sub-programs (1 week)

Introduce subprograms, Run-time stacks- Passing arguments on the stack, allocating local variables on the stack, Returning results, examples
9. Synchronous Finite State Machines (FSMs) (1.5 week)

Introduce state machines - Implementation of storage of state using flip-flops, Implementation of combinatorial logic, Work through a detailed example from problem to final solution (circuit), Briefly discuss the state machine of control unit of machine we are studying
10. Input/output (1 week)

Introduce polling and interrupt driven I/O, Examples using the machine we are studying
11. Buses (1 week)

System bus, Masters and slaves, Bus design issues- Timing, Arbitration, Do an example of bus arbitration
12. Demo (1 class)

An in-class demo of a digital system. (E.g. an FPGA that generated video output to a VGA monitor), briefly discuss current tools available to developers, Talk about some exciting types of digital systems that can be built.

**Text:** Course Notes