Carleton UNIVERSITY | Department of Systems and Computer Engineering

SYSC 2320
Introduction to Computer Organization and Architecture

Calendar description

Computer organization: processor, memory, input/output, system bus. Microarchitecture. Instruction set architecture. Assembly language programming: addressing modes, instruction encoding, execution. Assembler. Simple digital I/O, programmable timer. Input/output methods: polling, hardware interrupts.

Includes: Experiential Learning Activity.

Lectures three hours a week, laboratory three hours a week.

http://calendar.carleton.ca/undergrad/courses/SYSC/

Prerequisites

SYSC 2310 and second-year status in Engineering.

Precludes additional credit for SYSC 2001 and SYSC 3006.

Prior knowledge

Students should have knowledge of:

- Numeral systems (e.g., binary, decimal, hexadecimal)
- Basic C programming (or equivalent)
- Logic gates (e.g., AND, XOR)
- Combinational circuits (e.g., MUXs, Decoders, etc.)
- Basic sequential circuits (e.g., flip-flops, counters, etc.)
- Finite state machines.

Course objectives

This is a first course on the organization of computer systems at the hardware/software interface. Understanding this interface involves the processor's Instruction Set Architecture (ISA), the system's memory organization, and the system's I/O peripherals. Engineers working with microcontrollers (including mobile devices) must understand computer systems at this level, and these concepts form the foundation on which more powerful computer systems are based (such as desktop systems, servers, multiprocessors, and supercomputers). The processor's native assembly language is a central software topic in the course. The increasing use of high-level languages such as C is reducing the need for assembly language programming proficiency. As a result, the course will focus more on understanding how an assembly language manipulates information at the hardware/software interface and can support a high-level language, and will not emphasize developing complete applications in assembly language.

List of topics

- Computer system components: processor, memory, I/O, interconnection bus:
  - Information encoding, data representation in binary, hexadecimal
  - Number systems, unsigned integers, signed integers, 2's complement
  - Computer hardware organization: datapath and control
  - Registers, instruction cycle
- Hardware/software interface (Instruction Set Architecture):
  - Instructions: data manipulation, data transfer, control flow, instruction encoding
  - Computer arithmetic, flags
- Microcontroller example:
  - Microcontroller concept
  - System on Chip, memory model, ISA
- Assembly language programming:
  - Code snippets, examples
  - Assembly process, linker, loader
- High-level Language Support:
  - Variables, arrays, structures, assignment, looping, conditional statements
  - Procedures and functions, parameter passing
- Peripheral I/O and Interrupts:
  - Register model of peripheral devices: parallel I/O, serial I/O, timers
  - Polling
  - Hardware interrupts: vectored and prioritized, Nested Vector Interrupt Controller(NVIC)
  - Examples: timer, serial
  - Software interrupts, o/s calls

Learning outcomes

By the end of this course, students should be able to:

- Develop hardware circuits at the core of microprocessors and microcontrollers.
- Develop hardware alternate designs to achieve similar objectives.
- Design finite state machines necessary for the implementation of various instructions.
- Develop machine code and corresponding assembly language programs to implement specific programming tasks.
- Use FPGA modules to build hardware circuits using the Verilog hardware description language.
- Build hardware circuits necessary to interface input/output modules using polling and interrupt-based techniques.

Graduate Attributes (GAs)

The Canadian Engineering Accreditation Board requires graduates of engineering programs to possess 12 attributes at the time of graduation. Activities related to the learning outcomes listed above are measured throughout the course and are part of the department's continual improvement process. Graduate attribute measurements will not be taken into consideration in determining a student's grade in the course. For more information, please visit: https://engineerscanada.ca/.

Graduate Attribute	Learning outcome(s)
1.5.S: Knowledge Base: Developed: Computer Systems	1-6
2.3: Problem Analysis: Introduced: Use of assumptions	2, 3
4.2: Design: Introduced: Detailed design specifications and requirements	4, 5
4.5: Design: Introduced: Design implementation / task(s) definition	3, 6
4.6: Design: Developed: Alternate solution(s) definition	2

Accreditation Units (AUs)

For more information about Accreditation Units, please visit: https://engineerscanada.ca/.

The course has a total of 55 AUs, divided into:

- Engineering Science: 100%

Instructor and TA contact

Specific to course offering (tbd)

Textbook (or other resources)

Specific to course offering (tbd)

Evaluation and grading scheme

Specific to course offering (tbd)

Breakdown of course requirements

Specific to course offering (tbd)

Tentative week-by-week breakdown

Specific to course offering (tbd)

General regulations

Specific to course offering (tbd)