Carleton UNIVERSITY | Department of Systems and Computer Engineering

SYSC 3006
Computer Organization

Calendar description

Computer organization: processor, memory, input/output, system bus. Number systems: binary, decimal, hexadecimal. Assembly language programming: representation of data, instruction encoding, execution. Devices: keyboard, programmable timer, parallel interface. Input/output methods: polling, hardware/software interrupts.

Includes: Experiential Learning Activity.

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

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

Prerequisites

SYSC 2006 and ELEC 2607.

Precludes additional credit for SYSC 2001, SYSC 2003, SYSC 2320 and SYSC 3310. May not be taken for credit by students in Computer Systems Engineering, Communications Engineering, or Software Engineering.

Prior knowledge

Students should have knowledge of:

- Logic gates, switching circuits and their logical behavior.
- Boolean algebra, truth tables of binary values and Karnaugh maps.
- Basic knowledge of combinational circuits.
- Basic knowledge of sequential circuits and state machine.
- Binary arithmetic operations e.g. addition/subtraction and 2’s complement notation.

Course objectives

Engineers working with microprocessors and 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). Therefore, this course introduces students to basic computer organization and functional design down to the gate-level abstraction, and to hardware/software interfacing.

List of topics

- Computer system architecture:
  - 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, floating point
  - Computer hardware organization: datapath and control
  - Registers, instruction cycle
- Hardware/software interface:
  - 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:

- Understand how a computer works at hardware/software interface through the Instruction Set Architecture (ISA).
- Understand microarchitecture (computer organization) to fetch an instruction from the memory and execute using the data path.
- Understand hardware circuits necessary to interface input/output modules using polling and interrupt-based techniques.
- Design finite state machines necessary for the implementation of various instructions.
- Design hardware circuits at the core of microprocessors and microcontrollers.
- Develop machine code and corresponding assembly language programs to implement specific programming tasks.

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	
4.2: Design: Introduced: Detailed design specifications and requirements	
4.5: Design: Introduced: Design implementation / task(s) definition	
5.2: Use of engineering tools: Document-processing and graphics packages	

Accreditation Units (AUs)

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

The course has a total of 49 AUs, divided into:

- Engineering Science: 60%
- Engineering Design: 40%

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)