CIS*2030: Structure and Application of Microcomputers
Computer Science, Fall 2019

Instructor: G. Grewal
Office: Reynolds 2208
Phone: x52630
Email: ggrewal@uoguelph.ca
Course Website: https://courselink.uoguelph.ca

General Description

This course is intended to provide an introduction to the topic of *Computer Architecture*. Computer architecture refers to those attributes of a system that are visible to a programmer, or put another way, those attributes that have a direct impact on the logical execution of a program. As a result of this course, students will gain insight into the operation of the major components of a computer system, assembly-language programming, and computer interfacing. The Motorola 68000 has been selected as the example architecture to study, due to the simplicity and clarity of its architecture.

Course Outline

Each bullet corresponds to *roughly* 1-3 weeks of the semester.

- Binary encoding and data representation: number systems, binary arithmetic, complement arithmetic, ranges, arithmetic overflow, and other code systems
- Basic hardware components, overall architecture and data flow of a typical computer, Instruction-Set Architecture for M68000
- An assembler language view of the M68000
- Hardware support for high-level language (C) concepts: data structures, loops, conditional statements, hardware stack, functions, parameter passing, return values, stack frames, local variables, recursion and re-entrant code
- Hardware support for Operating Systems: privileged states; exception processing
- Interaction among components: device operation, device control, bus signals, input/output protocols: polling, interrupts, direct-memory access
- Memory Hierarchy and Caching
- Fixed- and Floating-Point represenations

Textbook

- MacKenzie S., (1995). *The 68000 Microprocessor*, Prentice-Hall. (Required)
- 68KMB Lab Exercises (Provided to you in lab)

Lectures

There are two sections, each with three *three* lectures per week: MWF, 1:30pm to 2:20pm, ROZH 102, and MWF, 3:30pm to 4:20pm, JTP 2266.

Course Evaluation

Your final grade will be determined as follows:

Weight	Description	Notes
18%	6 Homework assignments	**Due at beginning of class** – Late assignments will not be accepted for any reason; however, the lowest weight assignment will be dropped when calculating your final grade. See course website for dates.
32%	8 Laboratory Assignments	Each lab, save for the first, contains a **pre-lab** and **in-lab** component. Pre-labs are due at the beginning of the lab; in-labs are due before the end of the lab. Further details will be provided during the first lab.
20%	Midterm	**Date:** October 25, Time and Place TBA. Weight of a missed midterm will be transferred to the final exam.
30%	Exam	**Date:** December 9, 2:30pm to 4:30pm, Room: TBA
Total Grade = 18% (Homework) + 32% (Labs) + 20% (Midterm) + 30% (Exam)		

You must achieve a passing grade on the examination portion of the course (i.e., at least 25 out of 50) and a passing grade (i.e., at least 16 out of 32) on the labs. Failure to satisfy either of the previous cases results in the following calculation for your final grade: Final Grade = MIN(45, Total Grade).

Teaching Assistants

Teaching Assistants allocated to this course:

- Marshall Asch masch@uoguelph.ca
- Bazyli Debowksi bdebowsk@uoguelph.ca
- Jeremy Foxcroft foxcroft@uoguelph.ca
- Abbas Abir abbasa@uoguelph.ca
- Evan Kish ekish@uoguelph.ca
- Zina Ramirez ramirezr@uoguelph.ca
- Stewart Thompson sthoms06@uoguelph.ca

All requests for re-grades must be made by email to *your* marker within one week of an assignment/exam being returned.

Advising hours

- TA advising hours: See course website for dates and times.
- Gary: Open-door policy; otherwise, email me to request a specific time. Please do not send questions by email.

A Word of Caution

Needless to say, plagiarism in any form must be dealt with severely. Discussion with fellow students about problems is healthy. However, when answering questions do it yourself. Be original. All cases of academic misconduct are handled by the Dean, in conjunction with the Department Chair. Successive infractions of misconduct affirmed by this process could have consequences as serious as expulsion from the University. *(It is your responsibility to acquaint yourself with the definitions and ramifications of academic misconduct as described in the university's undergraduate Calendar.)* The risks are sufficiently great that they are not worth taking. If you are having trouble, please see the teaching assistant or the instructor for help.

Lab Schedule for Fall 2019

	MON	TUE	WED	THR	FRI
SEP					6 *first class*
	9	10	11	12	13
	16 LAB 1	17 LAB 1	18 LAB 1	19 LAB 1	20 LAB 1
	23 LAB 2	24 LAB 2	25 LAB 2	26 LAB 2	27 LAB 2
OCT	30 LAB 3	1 LAB 3	2 LAB 3	3 LAB 3	4 LAB 3
	7 LAB 4	8 LAB 4	9 LAB 4	10 LAB 4	11 LAB 4
	14 Holiday ☺	15 Holiday ☺	16	17	18
	21 LAB 5	22 LAB 5	23 LAB 5	24 LAB 5	25 LAB 5
	28 LAB 6	29 LAB 6	30 LAB 6	31 LAB 6	1 LAB 6
NOV	4 LAB 7	5 LAB 7	6 LAB 7	7 LAB 7	8 LAB 7
	11 LAB 8	12 LAB 8	13 LAB 8	14 LAB 8	15 LAB 8
	18	19	20	21	22
	25	26	27	28	29

Lab Sections 0105 and 0205: Monday, 8:30am to 11:20am, THRN 3401
Lab Sections 0101 and 0201: Tuesday, 8:30am to 11:20am, THRN 3401
Lab Sections 0104 and 0204: Wednesday, 8:30am to 11:20am, THRN 3401
Lab Sections 0102 and 0202: Thursday, 8:30am to 11:20am, THRN 3401

Lab Sections 0103 and 0203: Friday, 8:30am to 11:20am, THRN 3401
Lab Sections 0107 and 0207: Tuesday, 2:30pm to 5:20pm, THRN 3401
Lab Sections 0110 and 0210: Thursday, 2:30pm to 5:20pm, THRN 3401
Lab Sections 0108 and 0208: Tuesday, 5:30pm to 8:20pm, THRN 3401
Lab Sections 0106 and 0206: Wednesday, 5:30pm to 8:20pm, THRN 3401
Lab Sections 0109 and 0209: Thursday, 5:30pm to 8:20pm, THRN 3401

Learning Outcomes

Successfully completing the homework and labarotory assignments in this course will contribute to the following learning outcomes:

1. Explain the digital representation of numeric and non-numeric data.
2. Identify the main abstractions that exist between programs and the actual hardware they run on, explain why these abstractions exist, and describe how they build upon each other.
3. Create assembly-language programs to implement some of the main abstractins and evaluate your program through testing and debugging.
4. Implement interfaces with external devices using common input-output strategies.
5. Describe the memory hierarchy and list trade-offs.