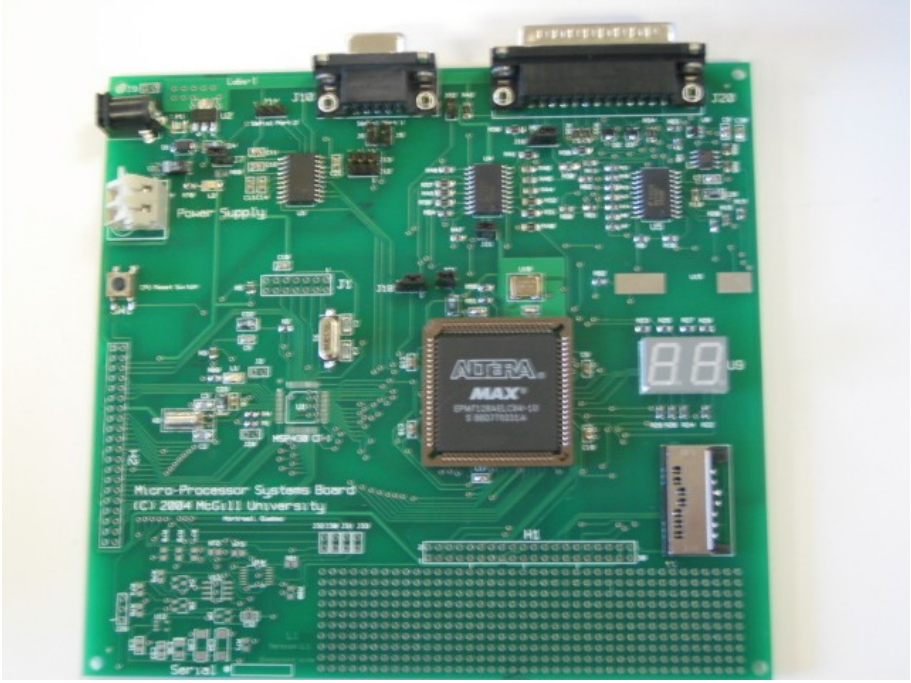
ECSE-426

Microprocessor Systems



Academic Integrity

McGill University values academic integrity.

Therefore all students must understand the meaning and consequences of cheating, plagiarism and other academic offences under the Code of Student Conduct and Disciplinary Procedures (see www.mcgill.ca/students/srr/honest/) for more information).

Course Material

I extend my thanks to Prof. Mark Coates for providing the bulk of the material to teach this course

Course Outline

- **Prerequisites:** ECSE-323 and EDEC 206
- **Instructor:** Jean-Samuel Chenard
  Room 544 (VLSI Lab) McConnell
  (Ph) 514-800-1266
  e-mail: jsamch@macs.ece.mcgill.ca
- **Please put ECSE-426 as part of e-mail subject**
- **Office Hours**: after lectures; by appointment.
- **Teaching Assistants: George Ciobanu, Vlad Gabriel Feyer, Xuan Liu**

My Background

- B.Eng. '99 McGill, M.Eng '05
  - Low-power wireless sensor networks
  - Planar antenna design methods
- Worked in Industry
  - 1 year firmware design + board-level
  - 4 years in ASIC/FPGA in optical networking
- Working on my PhD
  - Debug methodologies for Network-on-Chip
  - Flexible (re-configurable) data processing
- Part time FPGA hardware and Embedded Systems consultant

Reference Books

- Reference
  - A. Tanenbaum, *Structured Computer Organization*, 5th edition, Prentice-Hall, 2005.
  - J. H. (John H.) Davies, *MSP430 microcontroller basics*, Oxford: Newnes, 2008.
    - *Note: Available as an e-book from http://muse.mcgill.ca*
  - C. Hamacher, Z. Vranesic and S. Zaky, *Computer Organization*, 5th edition, McGraw-Hill, 2002.

Course Objectives

- Primary objective: necessary understanding and skills to design and build microprocessor systems.
- By the end of the course, you should:
  - Understand computer organisation and basic design principles;
  - Have developed proficiency in assembly programming for embedded systems and awareness of its benefits and disadvantages;
  - Be experienced in developing embedded-C solutions;
  - Know how to connect peripheral devices and how to program a microprocessor to interface with them;
  - Understand the principles of instruction set architectures and microarchitectures and their impact on microprocessor design and implementation;

Assessment

- 16% Participation and Understanding
  - 4 quizzes (see course outline)
- 44% Lab Experiments
- 40% Project.

Assessment

Evaluations	Contribution to Final Grade		
Experiment 1	14	Demo	8
		Report	6
Experiment 2	15	Demo	10
		Lab Notes	5
Experiment 3	15	Demo	10
		Lab Notes	5
Project	40	Demo 1	13
		Final Demo	15
		Report	12
Quizzes	16		16

Assessment

- Demos – marks for performance, robustness, code quality, performance testing.
- Reports or Lab notes (concise but comprehensive, 1 per group) – detailed report guidelines will be posted on WebCT. Follow these closely.
- Students receive an individual grade for the demonstration component
- Generally the grades of the members in a group will be very similar
- Differentiation: response to questions or quality of components for which each member is responsible.
- Final project includes a peer evaluation sheet

Course Content

- Top-down approach to microprocessor programming and design.
- Lectures focus on structured computer organization, and progress through “layers” :
  - problem-oriented language level (embedded C) + assembly language level
  - Instruction set architecture
  - Microarchitecture
  - Hardware

Course Content

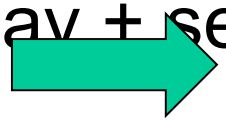
- Discuss design principles and impact on real-world architectures
  - Texas Instruments MSP430
  - ARM and commercial derivatives
  - Pentium
- The course focuses primarily on **experimental** work.
  - Each lecture: 45 minutes on theory and 30 minutes on how it applies to your hardware device and experiments.

Course Component	Lectures
Introduction Computer systems organization, Basic Design Principles	1
Problem-Oriented Language Level and Assembler Embedded C and assembly language	1
Instruction Set Architecture Properties, registers, instructions, data types, instruction formats, addressing, instruction types, control flow	2
I/O, Processor Interfacing Interrupts, clocks/timers, interacting with peripheral devices (UART/USART)	2
Microarchitecture Microinstructions, pipelining, branch prediction, prefetching,	2
Design Principles Embedded Systems, RISC vs. CISC, dual-core, hyperthreading, Pentium + picoJava machines	2

Course Experiment Structure

- 3 experiments + final project.

- Experiment 1: Assembly and C.
- Experiment 2: Intro. to hardware, timing, UART.
- Experiment 3: I/O, Interrupts.
- Project: Real-time clock, LCD display + serial interface between boards, sensors sensor network mote.



Course Timeline

Week (Mon.)	Lecture	Tutorial (*)	Experiment
1.Aug. 31	Intro		
2: Sept. 7	C + Assembly	MSP430	1: Assembly
3: Sept. 14	Instruction Set Arch.	Timers	1: Timers + C
4: Sept. 21	ISA ctd, Quiz 1	UART	2: UART I/O
5: Sept. 28	I/O, Interfacing		2: CPLD
6: Oct. 5	I/O, Interfacing, Quiz 2	USART	3: Wireless
7: Oct. 12	Microarchitecture	Wireless	3: Wireless
8: Oct. 19	Microarchitecture, Q 3	Peripherals	Project Part A
9: Oct. 26	Design Principles		Project Part A
10: Nov. 2	Design Principles, Q 4		Partial Demo
11. Nov. 9			Project Part B
12:Nov.16			Project Part B
13:Nov 23		* May be re-ordered	Final Demo

Course Info

- Labs conducted in pairs – write these on the circulated paper (otherwise random assignment).
- Both members must belong to same section!
- Final project in group of 4.
- Each team of 2 students need only submit one report for each experiment
- Four 15 minute quizzes will be conducted at the start of four lectures during the term.
  - lecture material since the previous quiz + most recently completed experiment.
- Penalties for Late Assignments
  - 5 percent per day (Fri.-Mon. = 1 day)
- Missed demo
  - reschedule for 65 percent of grade

Demonstration Time Slots

- Thursdays and Fridays
  - OK with everyone ?
  - Report conflicts with other courses ASAP
- Demos are scheduled during the time slot by the TAs
  - Don't miss your demo slot !
  - Both partners need to be present
  - Demo will last about 10 minutes per team
- Usually not the best time to add a feature...

Your Jobs This Week

- Preparation: review assembly/C programming for MSP430
- Read the assignment and think about it (construct a plan for a solution)
  - Check WebCT Postings
- Attend tutorial on Monday or Tuesday next week (according to your course section)

Next Week's Tutorial

- Overview of the CrossStudio for MSP430
  - Simple examples and use of the MSP430 core simulator.
- Tutorials are held at the start of the lab session in the MicroP lab.
  - 13:35 for Monday's tutorial
  - 14:35 for Tuesday's tutorial