

ECSE324 : Computer Organization

Introduction

Christophe Dubach

Fall 2020

Original slides from Prof. Warren Gross – 2017.

Updated by Christophe Dubach – 2020.

Timestamp: 2020/09/02 08:57:22

Disclaimer

Lectures are recorded live and will be posted **unedited** on *mycourses* on the same day.

It is possible (and even likely) that I will (sometimes) make mistakes and give incorrect information during the live lectures. If you have any doubts, please check the book, the course webpage or ask on Piazza for clarifications.

Christophe Dubach — brief CV



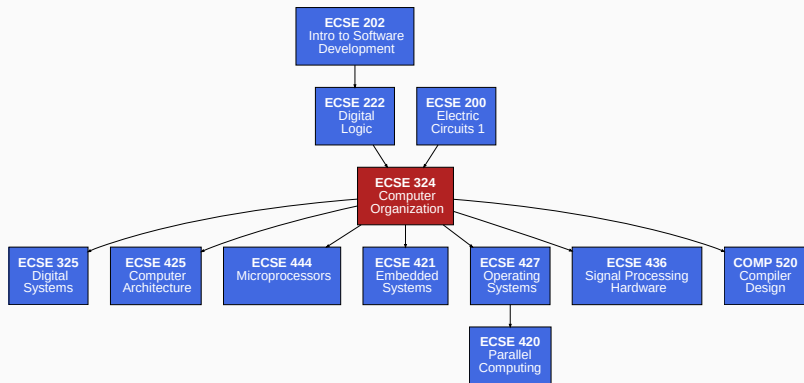
- 2005: MSc
- 2009: PhD, *Using machine-learning to efficiently explore the architecture/compiler co-design space*
- 2012: Lecturer (Assistant Professor)
- 2017: Reader (Associate Professor)
- 2010: Visiting Scientist
LiquidMetal: a language, compiler, and runtime for high level synthesis of reconfigurable hardware
- 2020: Associate Professor (ECE/CS)
 - ECSE 324 : Computer Organization
 - COMP 520 : Compiler Design

What is Computer Organization?

- We will learn about the *design of computers* both hardware and software
- In previous classes you learned about:
 - building blocks of digital hardware (ECSE 222)
 - how to write programs in Java (ECSE 202)

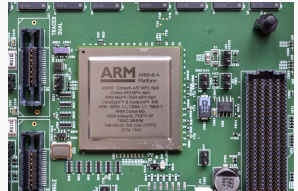
This class will fill in the missing details to tell the story of how a digital machine called a computer runs programs.

This class in the curriculum



What will you take away from this class?

- you will learn about the main computer structures and how the processor hardware executes a program
- you will learn how to write programs at the machine level in “assembly language” for one of the most popular modern processors (ARM). ARM processors are used in almost every smartphone (95%)
- you will learn how high-level languages are translated into assembly language.



source: CC0 1.0 Universal Public Domain Dedication

Why learn assembly language?

Learning assembly is the best way to understand hardware.

Before we can describe the hardware, we first need to know what it does!

Why take this class?

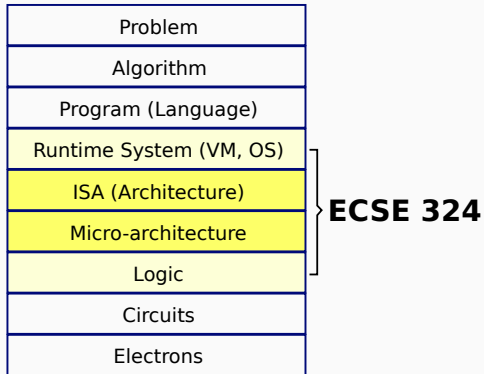
- Computer engineers:
This is the main event – understand how computers work.
- Electrical engineers:
Computers are at the heart of almost all EE systems.
- Software engineers:
Learn how to write efficient code by understanding the machine it runs on

Is this a hardware or software class?

Both!

The concepts behind computer organization we will cover have not changed for **several decades** and are unlikely to change in the near future.

Computing system stack: Big picture



What knowledge do you need from ECSE 202 & ECSE 222?

- You should be able to *write programs* in a “C-like” *high-level language*, such as C or Java. This course will use C as the example language.
- You should be able to *think algorithmically*
- You should know all about the basics of *digital logic*: binary numbers, logic gates and flip-flops, binary addition, and finite-state machines.
⇒ These are the building blocks of computers.

Course information

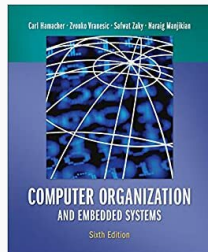
- Main *website* is: <http://ecse324.ece.mcgill.ca/>
 - you should check it regularly
 - contains the link to the live lectures and office hours (among other things)
- Complemented by *Piazza* forum:
 - Important announcements about the course will be posted there
 - *I assume you read them*
 - Make sure to register:
<http://piazza.com/mcgill.ca/fall2020/ecse324>

1. Computer Technology and Abstractions
2. Instruction Set Architecture
3. Software
4. Input/Output
5. Memory
6. Processor Implementation
7. Arithmetic

Computer Organization and Embedded Systems

Hamacher, Vranesic, Zaky, and Manjikian 6th Edition, McGraw-Hill, 2012.

Should be available at McGill Bookstore
(online format)



Course online meetings

You will have the following zoom meetings:

- 3 hours of *lecture* per week
- 2 hours of *tutorial* per week
- 2 hours of *lab* per week

All lectures and tutorials will be recorded and a link to the recording will be posted on the day of the meeting.

You must register for a 2-hour weekly laboratory session and a 2-hour weekly tutorial session.

For registration, timetable issues, conflicts, etc... please do not contact the course staff. Instead, please contact the McGill ECE Undergraduate office.

Labs

The laboratory experiments are meant to reinforce the lecture material, and form an **integral part** of the learning experience in this course.

There will be a total of 4 labs:

- Lab 0 is not graded
- labs 1–3 are graded
 - For each lab, you will have to produce a short report
 - and present a demo.



Due to remote teaching, the labs will be performed using an **online emulator**.

Labs are done individually!

Check course website for more details about what constitutes cheating.

Tutorials are here to help you understand how to solve problems.

- We will release assignments a few days before each tutorial session to give you a chance to try to solve the assignments on your own.
- During the tutorial session, the TA will explain how to solve the assignments and answer any questions you may have.

This should **prepare you for the midterm and final exam.**

Tutorials and Labs Schedule

There will be no labs/tutorials this week or next week.

- Tutorials and labs start the week of September 14

Sessions:

- To switch lab (or tutorial) sessions, do it **through the University**.
- Only attend the session you are registered for. Any graded activities **will not be graded** if done in the session you are not registered for.

Evaluation

Final exam	55%
Lab experiments	25%
Midterm exam	20%
Piazza Bonus	10%

- Midterm will be held online on October 26.
- Piazza bonus is a function of the number of *your* endorsed answers

Labs and exams will be conducted online *individually*.

The schedule is on the course website.

Zero cheating tolerance

Any case of suspected cheating will be reported to the faculty.

Exam format

Both the midterm and final exam will be in the form of an **online open book exam**. You will be given 48 hours to complete each exam.

The exam is designed so that it can be completed in:

- 1 hour for the midterm
- 3 hours for the final exam

assuming you know the material and do not have to look up the information.

The exam will have several multiple-choice questions and a few problems to solve. The assignments and tutorials should prepare you for the exam problems you will encounter.

Getting help

Remote learning is challenging for students and **TAs alike** (they may have their own difficult circumstances).

Nonetheless, we will do our best to help you succeed in this course.

There are four main mechanisms for getting help:

- Live recorded **tutorial sessions** where we will explain how to solve assignments and you will have the chance to ask questions.
- Live **lab sessions**, where you have the chance to ask questions about the labs (individually).
- **Piazza** forum where you can ask questions at any time and get answers from your peers or the TAs/myself.
- **Online office hours** on Wednesdays:
 - 9am - 9.30am
 - 12.30pm - 1pm
 - 4.30pm - 5pm

What you need to do to succeed at this course

- Attend the live lecture - even if you cannot follow along with everything (or are bored), this will force you to follow the course on a **regular schedule**.
- This course is different: little math, no equations, but lots of concepts that build on one another and take time to really understand.
- Don't cram – there is a fair amount of technical material that cannot be absorbed quickly.
- Attend live tutorial/lab sessions and **ask questions**.
- Try to do all the assignments before the tutorial sessions.

One last word

All questions through Piazza, no email!

A typical week for any Professor: hundreds of emails to deal with.

Although not encouraged, if you are unsure, you can post privately on Piazza (only the TAs and myself will see it).



Only exception to this rule: if you are having a personal issue you need to discuss with me only (you can also do it via the office hours by the way).