Sheridan

# SYST27198

## CPU Architecture And Assembly

Course Introduction

Fall 2024

# Your Instructor

- Professor Rachel Jiang
  - Email: rachel.jiang@sheridancollege.ca
    - Important emails from me go to your main Sheridan email, not Slate!
    - The Maximum of 2 business days for getting the replies to your emails.
  - My office: E202
  - My office hour: Mon. 4.00–5.00pm; Weds. 7.00- 8.00pm

    or other times by appointment, email me if you're coming

# Your Instructor

- My background
  - Software design and development
  - Medical image processing
  - Computer 3D graphics
  - Artificial Intelligence
  - Gaming

# Our Goals for Today

- Get to know each other
- Understand the course
  - Structure
  - Requirement
  - What's expected of *you*
  - What you can expect from *me*
- Start learning about CPU architecture and assembly

# Our Goals for the Entire Course

- Learning Outcomes
  - This course is important for your rest of courses in the BAISc (Info Systems Security) program
  - I assume that you have no previous experience in CPU architecture and assembly language programming
    - If you do, that is fine
  - The main tool that you'll need: Your brain

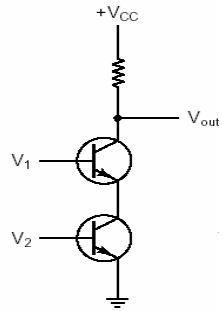
# Our Goals for the Entire Course

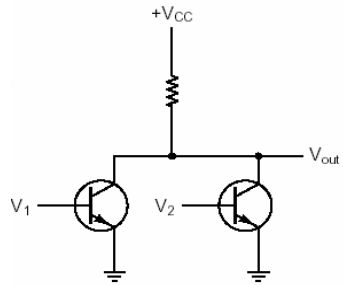
## Learning Outcomes

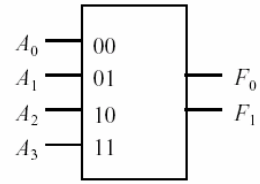
- In this course, you will learn:
  - Select digital components
    - MUX, DEMUX, decoder, priority decoder, PLA, shifters, full adder) using Boolean algebra methodology, and logical gates.
  - The basic architecture of the von Neumann machine and the fetch-decode-execute-write model.
  - The micro-architectures, instruction sets, and assembly language levels for various CPU designs.

# Learning Outcomes (cont'd)

- modern computer architecture design concepts, parallel processing cache, pipeline depth and superscalar instruction issues.
- the architectural features of modern computer systems
- the fundamental limitations and bottlenecks in modern processors.
- Programming in assembly programming language.

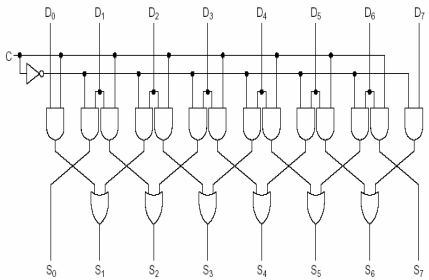



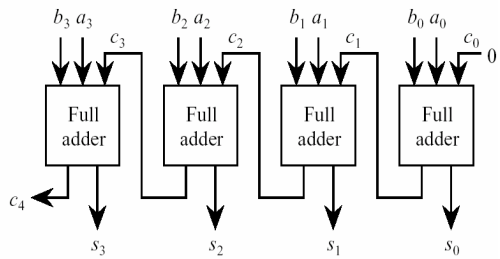



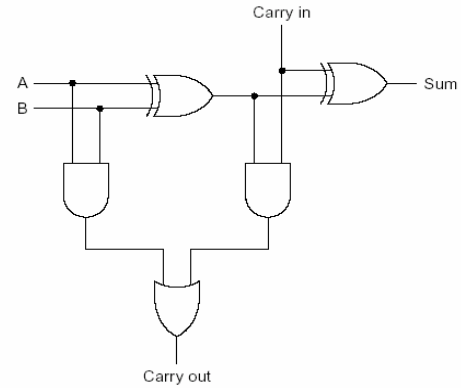

$$F_0 = \overline{A_0}\,\overline{A_1}\,A_3 + \overline{A_0}\,\overline{A_1}\,A_2$$

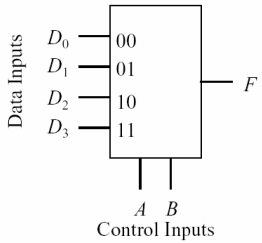
$$F_1 = \overline{A_0}\,A_2 A_3 + \overline{A_0}\,A_1$$



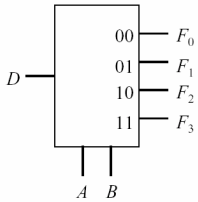

| A | B | Carry in | Sum | Carry out |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 0 |
| 0 | 1 | 0 | 1 | 0 |
| 0 | 1 | 1 | 0 | 1 |
| 1 | 0 | 0 | 1 | 0 |
| 1 | 0 | 1 | 0 | 1 |
| 1 | 1 | 0 | 0 | 1 |
| 1 | 1 | 1 | 1 | 1 |




| $A$ | $B$ | $F$ |
|---|---|---|
| 0 | 0 | $D_0$ |
| 0 | 1 | $D_1$ |
| 1 | 0 | $D_2$ |
| 1 | 1 | $D_3$ |

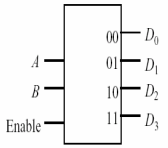
$$F = \overline{A}\,\overline{B}\,D_0 + \overline{A}\,B\,D_1 + A\,\overline{B}\,D_2 + A\,B\,D_3$$




$$F_0 = D\,\overline{A}\,\overline{B} \qquad F_2 = D\,A\,\overline{B}$$

$$F_1 = D\,\overline{A}\,B \qquad F_3 = D\,A\,B$$

| $D$ | $A$ | $B$ | $F_0$ | $F_1$ | $F_2$ | $F_3$ |
|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 | 1 | 0 | 0 |
| 1 | 1 | 0 | 0 | 0 | 1 | 0 |
| 1 | 1 | 1 | 0 | 0 | 0 | 1 |




| Enable = 1 | |
|---|---|
| $A$ $B$ | $D_0$ $D_1$ $D_2$ $D_3$ |
| 0 0 | 1 0 0 0 |
| 0 1 | 0 1 0 0 |
| 1 0 | 0 0 1 0 |
| 1 1 | 0 0 0 1 |

| Enable = 0 | |
|---|---|
| $A$ $B$ | $D_0$ $D_1$ $D_2$ $D_3$ |
| 0 0 | 0 0 0 0 |
| 0 1 | 0 0 0 0 |
| 1 0 | 0 0 0 0 |
| 1 1 | 0 0 0 0 |

$D_0 = \overline{A}\,\overline{B}$ $\quad D_1 = \overline{A}\,B$ $\quad D_2 = A\,\overline{B}$ $\quad D_3 = A\,B$

# Course Evaluation

| | |
|---|---|
| Midterm Test | 30% (week 7) |
| Final Exam | 30% (week 13) |
| Project and Presentations | 20% |
| Quizzes and In-class Exercises (20%) | |

- To pass the course students must:
  - *Average 50% or more on the midterm test plus final exam*
  - *Average 50% or more overall*
    - There may be additional in-class exercises or quizzes which don't count toward your final grade

# Course Info

- For the official course outline see the link in my class plan which is available on SLATE under week0 folder.

- If you need extra accommodation, e.g. extra time for tests, please introduce yourself to me

  – Register at Accessible Learning Services

# Our SLATE Site

- Let’s have a look at the SLATE site for our section of SYST27198 …

# Important Course Materials

- Course textbook:
  - Computer Architecture: A Quantitative Approach, Hennessy, J.l., & Patterson, D., A., Morgan Kaufmann, 5th. ed., ISBN 978-0123838728, 2011

- My course slides & notes, available on SLATE usually is posted just before each class
- Notes you take in class
- Exercises, labs, quizzes you've completed
- **You will need to use all of these to succeed!**
  - Not everything will be in my slides

# Additional Course Materials

- Recommended Reading:
  - Stallings, W. (2015). "Computer Organization and Architecture: Designing for Performance 10th edition", Prentice Hall.

- Supplementary online ebook (free!)
  - Randall Hyde (March 2010) “Art of Assembly Language”, 2nd Edition (Free Ebook) ISBN: 978-1-59327-207-4
  - https://www.nostarch.com/assembly2.htm

# What Sheridan Expects of You

- All students are expected to follow the Sheridan Student Code of Conduct:

http://www.sheridancollege.ca/~/media/Files/Sheridan%20College/Life%20At%20Sheridan/Student%20Services/Student%20Rights/Student%20Rights%20and%20Responsibilities/student%20code%20of%20conduct%20approvedfinal.pdf

- Strictly avoid plagiarism, copying, cheating
  - Everyone must view the library **Academic Integrity Tutorial**:

    http://sheridancollege.libguides.com/academic_integrity

  Watch the intro video, then click the orange "Get Started" arrow for the rest of the tutorial
  - Never email or otherwise copy your work, or in any way allow it to be copied

# Academic Integrity (cheating/plagiarism)

- Sheridan has a formal Academic Integrity process and I will use it.
  - First offence: Mark of zero and a letter in your file
  - Second offence: Termination from the course (F/TM)
  - Third offence: Severe penalty, expulsion

- Major types of plagiarism
  - Copying or emailing assignments: DON'T DO THIS
    - Discussing ideas, concepts, and methods is OK
  - Copying from the internet: REWRITE text (sentences) in your own words, write code yourself

# Who Copied From Who?

- It doesn't matter if plagiarism was on purpose or by accident
- It doesn't matter who did the work and who didn't
  - When plagiarism happens, all students involved are equally guilty!
- Never email or otherwise copy your work, or in any way allow it to be copied
- Don't be the "helpful" student who gives your work to others to "help" them
    - You will get a zero, maybe an F in the course, maybe get expelled
- Don't share passwords or mirror your hard drive using the cloud!

# Online Resources

- Don’t copy sentences/paragraphs from online sources, for example Wikipedia, ChatGPT, and etc…
  - Don’t copy answers directly from any references, not even your textbook
  - Rewrite in your own words (and use proper English sentences)
  - Use more than one source of information
- Don’t copy code from online sources, even open source sites like sourceforge
  - The programming part of this course, you must write the programs yourself
  - *Copying example code from our textbooks or my slides is OK*

# Working Together and Helping Each Other

- I encourage students to help each other with assignments.
  - When one student helps another in a productive way, they both end up understanding better.
- Some ways of helping are more productive than others, and some ways of “helping” are actually academic dishonesty –cheating.
- Good ways to help…
  - Talking things over with someone to help them understand a concept
  - Helping someone find the information they need
  - Testing another student’s program to look for mistakes
  - Sitting with someone to advise them while they debug a program they are having trouble with

# Working Together and Helping Each Other

- Bad ways to help… (cheating)
  - Writing a part of somebody's code for them
  - Showing someone your code so they can write it down
  - Mailing somebody your program so they can use it as a template, cut and paste parts of it, or change it slightly and hand it in as their own

# Artificial Intelligence tools

- Appropriate use of generative Artificial Intelligence tools
  - In alignment with Sheridan's Academic Integrity Policy
    - students should consult with their professors and/or refer to evaluation instructions regarding the appropriate use, or prohibition, of generative Artificial Intelligence (AI) tools for coursework.

    - **Turnitin** AI detection software may be used by faculty members to screen assignment submissions or exams for unauthorized use of artificial intelligence. (ref link is pending upon website is active)

# What I Expect of you in Class

- While in Class You Should NOT
  - Use your laptop to play games, watch YouTube, check facebook etc.
  - Work on tasks outside the scope of this class (for example assignments in other classes that are due)
- While I'm speaking or presenting material
  - LISTEN, take notes
  - Don't use your laptop for anything not related to today's topic (close the lid)
  - Cell phones off, or in Airplane mode

# What You Can Expect of Me

- I'm here to help you learn and succeed
  - Lend my experience and knowledge
  - Work together to overcome issues
  - This course should hopefully be interesting and fun!
- Feel free to email or arrange to meet me for any reason or concern you may have
  - Trouble with an exercise or assignment
  - Help with studying for quizzes & tests
  - Decisions related to the course
  - When in doubt, come to my office hour!

# What if you have trouble on an assignment?

- *Don't* copy from your friends
- Come to my office hour (email first)
  - Time & location near the start of these slides
- Email me to arrange to see me another time
- Ask questions in class
  - Maybe 10 other people have the same question!

# NOW I'D LIKE TO HEAR A LITTLE ABOUT YOU!


# Let’s Have Fun!

- There are lots of interesting things to learn…
- Rest of today:
  - Start learning about CPU architecture
  - Introduction to Von Neumann’s Computer Model