EECE 2322 – Fundamentals of Digital Design and Computer Organization

Prerequisites: CS 3500 or EECE 2160 or instructor approval

Overview

This course offers a global perspective on the design of a full processor taking logic gates as the simplest building blocks. The course begins with the design of simple combinational and sequential circuits, continues with an introduction to the RISC-V instruction set architecture (ISA), and wraps up with the construction of a simple processor with support for a subset of the RISC-V ISA, using a single-cycle and a multi-cycle implementation approach.

The course lectures are accompanied by lab sessions that guide the student through an incremental design of a fully functional single-cycle processor data path. All design steps are synthesized into the FPGA of a TUL PYNQ-Z2 board. We will only be using the FPGA fabric for this class, and not the ARM processor on-board.

Textbook

The core material presented in this course is extracted from the following textbook, though the order of concepts will differ. Even though this book is a good reference material for you, all of the material given during lectures and on Canvas will be self-contained. Important: We are **not** using the ARM version of the textbook.

- D. M. Harris and S. L. Harris, "*Digital Design and Computer Architecture, RISC-V Edition*", 1st edition, October 2021, ISBN 978-0128200643

Software for both EECE 2322 and EECE 2323

This course is tightly integrated with ECE2323. Course assignments and the ECE 2323 laboratory experiments use the Xilinx Vivado tools version 2022.1 for logic design, simulation, and synthesis. These tools are available through the COE VLAB. To access the VLAB, you must have a COE account. Students enrolled in this class who are not COE students can get an account. More information on using the VLAB is provided in a separate handout.

Grading

Your final grade is calculated as a numeric grade between 0 and 100 based on the percentages below:

Weights

- Homework 25%
- Midterm 1 20%
- Midterm 2 25%
- Final exam 30%

Your numerical semester grade is converted to a letter grade based on the following scale (your numerical grade is rounded up to the nearest percent):
93%-100% = A 90%-92% = A- 87%-89% = B+ 83%-86% = B 80%-82% = B- 77%-79% = C+
73%-76% = C 70%-72% = C- 67%-69% = D+ 63%-66% = D 60%-62% = D- range < 59% = F

Homework assignments

There will be a total of 10 weekly homework assignments. Assignments will be posted on Canvas at least 7 days before their due date and must be submitted on Canvas as well. Each homework assignment will require you to upload a PDF file with your answers, which you can produce from most common word editors (Microsoft Word, LibreOffice, LaTeX, …).

Homework due dates are strict deadlines with no exceptions. The exact due dates are specified at the end of this document. Please make sure that you submit your assignments in advance to avoid unexpected submission problems due to Internet connectivity issues, trouble with PDF document generation, problems with the submission link, etc.

To add some flexibility to this policy, the average grade for homework assignments will be calculated by discarding either that which received the lowest grade or which was not submitted on time at all. This exception is aimed at covering any inevitable situation that prevented you from submitting a homework assignment on time, while it also benefits those students with no missing assignment.

Midterm and final exams

The course has two **midterm exams** and a **final exam**. The exams will be administered in class and will be closed book and closed notes. Tentative exam dates are listed below. Final exam date: Week of Dec 12th, 2022.

Course Policies

Attendance and Punctuality

Attendance to lecture sessions in-person is highly recommended to fully participate in class activities and exercises. Please use the course textbooks also to read through the topics presented in class.

Punctuality in the classroom is indispensable and constitutes a basic rule of respect toward your classmates and your instructor.

Computing and Course Resources

This course will use the Northeastern University Canvas Course Management System accessible through MyNortheastern or by going to https://canvas.northeastern.edu/, Check the course site regularly for class materials and additional resources.

Important Dates

Week	**Assignment**
Week 1 Sept 7th	
Week 2 Sept 12th	
Week 3 Sept 19th	Wednesday Sept 21st - Hwk-1 Due
Week 4 Sept 26th	Wednesday Sept 28th - Hwk-2 Due
Week 5 Oct 3rd	Wednesday Oct 5th - Hwk-3 Due Thursday Oct 6th – **Midterm 1**
Week 6 Oct 10th	*Monday Oct 10th – Columbus Day*
Week 7 Oct 17th	Wednesday Oct 19th - Hwk-4 Due
Week 8 Oct 24th	Wednesday Oct 26th - Hwk-5 Due
Week 9 Oct 31st	Wednesday Nov 2nd - Hwk-6 Due Thursday Nov 3rd – **Midterm 2**
Week 10 Nov 7th	Wednesday Nov 9th - Hwk-7 Due
Week 11 Nov 14th	Wednesday Nov 16th - Hwk-8 Due
Week 12 Nov 21st	*Wed Nov 23th & Thu Nov 24th - Thanksgiving*
Week 13 Nov 28th	Wednesday Nov 30th - Hwk-9 Due
Week 14 Dec 5th	Wednesday Nov 30th - Hwk-10 Due Review
Week 15 **TBD on week of Dec 12th**	**Final Exam** TDB

Course Topics

Unit 1 – Boolean Algebra and Logic Gates

- The Von Neumann architecture
- Logic gates (review)
- Boolean simplification (review)
- Circuit design (review)
- Completeness of NAND gates
- The SystemVerilog hardware description language
- Structural, dataflow, and behavioral models
- Test benches

Unit 2 – Binary Arithmetic in SystemVerilog

- Binary adder
- Binary subtractor
- Binary multiplier
- The magnitude comparator

Unit 3 – Combinational Logic in SystemVerilog

- The decoder
- The encoder
- The multiplexer
- The k-bit multiplexer
- The demultiplexer
- Discussion on sensitivity lists and if-else/case statements in behavioral models

Unit 4 – Sequential Logic in SystemVerilog

- The D latch
- Asynchronous inputs for flip-flops
- The D flip-flop with *clear*
- Registers
- Register files
- The RISC-V register file
- Non-blocking assignments on SystemVerilog

Unit 5 – The RISC-V ISA

- RISC-V operands
- Arithmetic-logic instructions
- Constant operands

- 32-bit immediate values
- Memory operands
- The program memory layout
- Assembler directives
- System calls
- Flow control I: if-then-else
- Flow control II: while loop
- Flow control III: for loop
- Integer multiplication/division
- Instruction encoding
- The program stack
- Functions

Unit 6 – The Single-Cycle Datapath

- The state of the processor: PC, register file, instruction/data memory
- Datapath design methodology
- Datapath element I: instruction fetch (new hardware + connections)
- Datapath element II: support for R-type instructions (new hardware + connections)
- Datapath element III: support for sw/lw (new hardware + connections)
- Datapath element IV: support for beq (new hardware + connections)
- The control unit
- SystemVerilog for the single-cycle datapath
- Performance analysis
- Limitations of the single-cycle datapath

Unit 7 – Finite State Machines

- Definition
  - Mealy and Moore FSMs, notation
- Analysis of FSMs
  - Gate diagram
  - State and output equations
  - State table
  - State diagram
- Design of FSMs
  - State diagram
  - State table

    - State and output equations
    - Gate diagram
- SystemVerilog for FSMs
- State diagram simplification

Unit 8 – The Multi-Cycle Datapath

- Design methodology: state diagram
- State 0: Fetch – all instructions (new hardware, connections)
- State 1: Decode – all instructions (new hardware, connections)
- State 2: Execute – R-type instructions (new hardware, connections)
- State 3: AluWriteback – R-type instructions (connections)
- State 4: MemAddress – lw/sw instructions (new hardware, connections)
- State 5: MemRead – lw instruction (new hardware, connections)
- State 6: MemWriteback – lw instruction (connections)
- State 7: MemWrite – sw instruction (connections)
- Branch target address computation – beq instruction (new hardware, connections)
- State 8: Branch – the beq instruction (connections)
- The control unit
- SystemVerilog for the multi-cycle datapath
- Performance analysis

Unit 9 – RISC-V Floating Point Operations

- Floating point number representation
- RISC-V Floating point assembly language operations

Accommodations for Disabilities:
Northeastern University and the Disability Resource Center (DRC) are committed to providing disability services that enable students who qualify to participate fully in the activities of the university. Students with documented disabilities who may need accommodations, or any student considering obtaining documentation should visit the DRC website at www.northeastern.edu/drc or contact their staff at 617.373.2675.

Statement on Academic Integrity:
A commitment to the principles of academic integrity is essential to the mission of Northeastern University. The Academic Integrity Policy can be found in the undergraduate student handbook (pages 38-41), or from the Office of Student Conduct & Conflict Resolution (OSCCR) http://www.northeastern.edu/osccr/academic-integrity-policy/. I encourage you to familiarize yourself with it. If a student violates this policy in any way, I reserve the right to impose a sanction of failure on the assignment/assessment or failure in the course. If you have questions about appropriate citations, please ask."