EEL-4713: Computer Architecture

Spring 2021

Instructor:	C. Harrys Konstantinou	**Time:**	24/7
Email:	ckonstantinou@fsu.edu	**Place:**	YouTube
Course webpage:	canvas.fsu.edu or famu.instructure.com	**Office:**	Zoom

Course Description: Modern computer architectures are presented by studying how the relationships between hardware and software impact performance, machine language definition, processor data path and control designs, interfacing, and advanced topics, such as caching and pipelining.

Objectives: At the completion of this course the student should be able to:

1. Calculate and interpret different performance and cost metrics of computer systems. (HW1)
2. Derive machine code from assembly instructions. (HW2)
3. Derive assembler code from an equivalent C code representation. (HW3)
4. Calculate and interpret IEEE standard binary floating-point number representations. (HW4) (The above four CIOs will be checked by Exam 1).
5. Design and implement multiplication and division algorithms (Project 1).
6. Analyze the control and data flow within a single-cycle CPU and/or arithmetic logic unit when executing specific instructions (Project 2).
7. Analyze a multicycle datapath of a microprocessor (Project 3) Note: Pipeline is an implementation for the multi-cycle datapath and control).
8. Analyze and compare different cache architectures, and/or identify the most suitable cache design for a given need (Project 4).
9. Explain or analyze contemporary issues in computer architecture.
10. Recognize the need for and engage in life-long learning. (The above six CIOs will be checked by Exam 2).

    Note: Only exams (e.g., Midterm, and Final Exams) and projects are included into ABET materials as evidence for PO (program outcome).

Prerequisites: EEL-4746, C Programming (e.g., COP 3014 (FSU) or CGS 3408 (FAMU)).

Instructor's Office Hours:

- Tuesdays, 11:00am - 12:30pm, or
- Other times by appointment/email only.

Main References:

- Computer Organization and Design ARM Edition: The Hardware and Software Interface, by David Patterson and John Hennessy, 1st Edition, Morgan Kaufman Publishers, 2017.

Grading Policy:

- Exam 1 (25%).
- Exam 2 (25%).
- Homework (25%).
- Project (4 individual projects) (25%).

Grading Scale:

- A if $\geq 80\%$ & $\leq 100\%$
- B if $\geq 70\%$ & $\leq 79\%$

- C if $\geq 60\%$ & $\leq 69\%$
- D if $\geq 50\%$ & $\leq 59\%$
- F otherwise

Course Policy:

Grading:

- Questions, problems and errors involving the grading of any test must be brought to the attention of the instructor within 1 week of the graded work's return to the class.
- A student's absence from class does not extend the time limit.
- After 1 week the grade is final and will not be reviewed at the student's request.

Exams:

- Test dates announced at least 2 week in advance
- The Exam 1 and 2 will be 24hr take-home exams.
- No make-ups will be granted unless prior approval has been obtained from the instructor or for university-recognized excused absence.

Homework and Project:

- All assignments are collected (submitted online) on the course website. Don't submit your work to the instructor via email. Don't put into the drop box on the door of the instructor's office.
- Projects are due at midnight on the due date. The due date is the exact two weeks after the project is given, usually a Thursday.
- Projects are required to be submitted online through course webpage with a single zip file of a name such as lastname_project1.zip. Do not submit several files for one project. Drop the single zip file into the digital drop.
- Late submission will NOT be accepted.
- Assignments are considered independent unless specifically stated otherwise.
- Group assignments or projects may be assigned.

Ethics/Honor Code:

- All students are bound by the Academic Honor Code of their university.
- Students are encouraged to discuss topics and homework, but the work itself is to be performed on an individual basis.
- Violations of the honor code will be reported. Penalties include but are not limited to
  - failing grade on the assignment (project or exam); and
  - failing grade on the entire course.

Students with Disabilities: Students with disabilities needing academic accommodations should:

- Register with and provide documentation to the Student Disability Resource Center (SDRC); and
- Bring a letter to the instructor from the SDRC indicating you need accommodations. This should be done within the first week of class.

Computer Usage:
This course will make use of compilers and simulators. Course material such as assignments, schedule, notes and solutions will also be posted on the course web page (Blackboard). Your grades will also be posted (password protected) on the web page.

Academic Honesty:
The full text of the Academic Honesty Policy is in the *Student Handbook*.

Resources:

- Undergraduate students: https://www.eng.famu.fsu.edu/undergraduate/student-handbook
- Graduate students: https://www.eng.famu.fsu.edu/ece/graduate/resources

Registring for this course means your agreement to this class policy and syllabus.

IMPORTANT UNIVERSITY & COLLEGE POLICIES provided as an APPENDIX

Tentative Course Outline:

1. Week 1: Compute Abstraction and Technology (Chapter 1)
    - Understanding and Evaluating Performance
2. Week 2: C, Assembly Language, Machine Code, and ARM Instructions (Chapter 2)
3. Week 3: instruction formats and addressing modes
4. Week 4: programming examples
    - Required: a platform to run assembly programs:
    - either Raspberry Pi 3 hardware platform, or the Fast Models of the Virtual Platform in the DS-5 Toolchain (the free community edition) for ARMv8, see community.arm.com for details.
    - Other emulator such as Visual: `https://salmanarif.bitbucket.io/visual/downloads.html`
    - The built-in Core Simulator in Keil MDK may be good enough for our learning purpose.
    - There are some online emulator that can compile and run your asm programs in ARM.
    - How to get started with the Pi?
5. Week 5: Arithmetic for Computers: Multiplication, Division, and Floating Point (Chapter 3)
6. Week 6: programming multiplication and division in ARM.
    - Review and Q/A Session for Exam 1
7. Week 7: The Processors: Datapath and Control (Chapter 4)
8. Week 8: Single-cycle and multiple-cycle implementation
9. Week 9: Pipeline implementation
10. Week 10: Hazards and exception handling
    - Enhancing Performance with Pipelining
    - Required: Using Logisim for designing and simulating digital logic circuits.
    - How to use Logisim?
    - Optional: Using a HDL: Verilog (or VHDL), for logic design
    - Optional: Using a Verilog simulation and synthesis tool: Icarus Verilog
11. Week 11: Memory Hierarchy and Cache Architectures (Chapter 5)
12. Week 12: Common framework for memory organization
13. Week 13: Virtual memory organization
14. Week 14: Contemporary R&D Issues in Computer Architectures:
15. Week 15: Multicores, shared memory processors, and clusters.
    - Parallel Processors from Client to Cloud (Chapter 6)
16. Week 16:
    - Review and Q/A Session for Exam 2

Relationship to ABET Program Outcomes: A, C, E, I, J and O (CpE).