Boston University Metropolitan College

Computer Architecture

MET CS472 A1
Biology Research Building, Room 122
5 Cummington Mall
Mondays, 6:00 – 8:45

Dave Hendrickson
hendrick@bu.edu
Office hours: In person before/after class; Always available by email
Teaching Assistant: Yeryoung Kim (yeryoung@bu.edu)

Course Description
Computer organization with emphasis on processors, memory, and input/output. Includes pipelining, ALUs, caches, virtual memory, parallelism, measuring performance, and basic operating systems concepts. Assembly language instruction sets and programming as well as internal representation of instructions. (In short, enable you to understand the internal workings of computers, what makes them fast, and what the primary design challenges are.)

Books
Computer Organization & Design MIPS Edition: The Hardware/Software Interface
David A. Patterson and John L. Hennessy
Sixth Edition (ISBN 978-0128201091) Print or electronic.
Available at the BU bookstore or online. *No other edition is acceptable*!

Courseware
Blackboard: CS472 A1 Computer Architecture (Spring 26) 26sprgmetcs472_a1

Prerequisites
MET CS231 or CS232. **Do not ignore this or consider it a CO-requisite!** You must be proficient in some language, comfortable with algorithms and handling arrays and structures or classes of data. Assembly language is helpful but not mandatory.

Learning Outcomes

By successfully completing this course you will be able to:

- Use binary and hexadecimal numbers, and two's complement signed numbers
- Understand Instruction Set design and the internal representation of instructions
- Apply the internal representation of instructions to their behavior in processors
- Understand key concepts in processor design (datapaths, control, pipelines)
- Build on the basic concepts to more advanced (multicores, superpipelines, superscalar)
- Articulate how caches bridge the chasm between processor and DRAM performance
- Understand Virtual Addresses and how to translate them to Physical Addresses
- Apply basic principles of logic design to create a simple Arithmetic Logic Unit (ALU)
- Understand how ALUs perform arithmetic from addition/subtraction to multiplication
- Apply Computer Architecture principles to other disciplines (OS and compiler design, some elements of Software design, hardware design)

Grades

There will be a midterm (30%) and a final (40%) which will together account for 70% of the grade. There will be also three programming projects (in the language of your choice) for the other 30%. Late submissions will be accepted with a 10% pro-rated penalty for each week late *except for the final project*.

Attendance is not a part of your grade. However, you are expected to take the exams at the scheduled time. If that is impossible, you must take the exam before the rest of the class. If you are a "no show" you get a zero.

In general, an "A" will be awarded for work that totals 92-100% of the possible points, "A-" for 90-92%, "B+" for 88-90%, "B" for 82-88% and so on down to F for below 60%. Grades may be scaled upwards based on class scores (though rarely to get an A), but not downwards.

Grades are YOUR responsibility. If you need a particular grade to get into the M.S. program, receive tuition reimbursement or stay academically eligible, then it is YOUR responsibility to perform at that level. "A" work will get you an "A" and "F" work (or cheating) will get you an "F". I must distinguish between exceptional work and that which falls short of that level. Do not expect the typical grade to be an A**. Grades of D or F are almost always the result of cheating or not doing projects.**

I try to be friendly and to inject humor into the lectures, but don't mistake that for anything less than a zero-tolerance policy toward cheating. Cheating and plagiarism will not be tolerated at all. They will result in an F for the course. Not an F for the assignment or the exam. An F for the course. YOU CHEAT, YOU FAIL, subject to procedural review.

Please take the time to review the Student Academic Conduct Code: http://www.bu.edu/met/metropolitan_college_people/student/resources/conduct/code.html.

All projects in this course are **INDIVIDUAL**. Feel free to discuss ideas with your classmates, but **any** shared code will mean an F for the course. **No exceptions!** It is not acceptable for someone else to do your work, whether that be a classmate, a spouse or anyone **OR CHATGPT OR ANY OTHER AI TOOL**. Do not share even one line of code.

Backups

Buy a USB drive (they are cheap) and **back up your work on projects daily**. (Or just email the code to yourself.) There are no project extensions or waived late penalties for a disk crash.

Boston University Metropolitan College

Class Meetings, Lectures & Assignments

Lectures, Readings, and Assignments subject to change, and will be announced in class as applicable within a reasonable time frame.

Date	Topic	Readings	Projects
Jan 26	Course Overview, Number systems, Signed numbers	2.4, 2.9, skim Chap 1	n/a
Feb 2	MIPS Instructions – Assembly Language & Internal Representation	2.1-2.7, 2.10 (skip asm programming, focus - internal rep)	Project 1 Assigned (Due Feb 23)
Feb 9	Measuring Computer Performance, CISC/RISC, Memory Hierarchy (intro)	1.6, 5.1, 5.3, 5.4 (ignore math), 5.8	
Feb 11, Wed 8 PM	OPTIONAL Project 1 Review Session (Zoom), Run by Grader Yeryoung Kim (will be recorded)		
Feb 17, **Tues**	**HOLIDAY RESCHEDULE** -> TUES Memory Hierarchy - Caches (Part 1)		Project 2 Assigned (Due Mar 23)
Feb 23	Memory Hierarchy - Caches (Part 2)		Project 1 Due
Feb 25, Wed 8 PM	OPTIONAL Project 2 and Midterm Review Session (Zoom), Run by Grader Yeryoung Kim (recorded)		
March 2	The Processor: Datapath and Control	4.1 - 4.4	
Mar 9	NO CLASS – Spring Recess		
Mar 16	MIDTERM EXAM		
Mar 23	Midterm Results, Pipelines (Part 1)	Rest of Chap. 4	Project 2 Due
Mar 30	Pipelines (Part 2)		Project 3 Assigned (Due Apr 22)
Apr 6	Advanced Architectures - Superscalar, Superpipelining		
Apr 8, Wed 8 PM	OPTIONAL Project 3 Review Session (Zoom), Run by Grader Yeryoung Kim (will be recorded)		
Apr 13	Parallelism, Multicores, OS Concepts, Virtual Memory, Advanced Caches	5.7, Chapter 6	
Apr 22, **Wed**	**HOLIDAY RESCHEDULE** Logic Design, Computer Arithmetic and ALUs	App. B-2, B-3 (sum prod), B-5, 3.2, 3.3	Project 3 Due
Apr 27	Final Exam Review		
May 4	FINAL EXAM		