Syllabus – CET2123C

Course Title: Computer Architecture	**Term:** Fall 2025
Number: CET3126C **Reference:** XXXXXX	**Campus:** North **Room:** 3231-00
Credits: **4** Credits	**Time:**
Professor: Lester D. Suarez	**Telephone:**
Email: lsuarez9@mdc.edu	**Office Hours:** Mo 7:40PM - 8:30PM Mo 8:30PM - 9:50PM
Department: School of Engineering + Technology	
Office location: North-Bldg 3, Room 3231-00	

Course Description

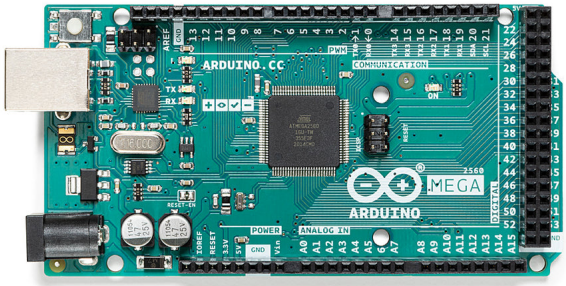
This course is intended for upper division students majoring in Electronics Engineering Technology. This course introduces the study of advanced microprocessor design. Students will learn the basic organization of computer systems including instruction-set architecture, execution pipeline, memory hierarchy, virtual memory, and Input/Output (I/O) subsystems. Students also learn advanced processor microarchitecture issues such as dynamic instruction scheduling, branch prediction, lock-up free caches, instruction-level parallelism, multiple instruction fetch/issuing, speculative execution, etc. to improve computer processor performance. Students will experimentally verify microarchitecture designs using industry standard microarchitecture simulators. Prerequisite: CET 2123C. Laboratory fee. (2hr lecture, 4 hr lab)

Textbooks/Resources/Supplies

Required Text: Computer Organization and Design MIPS Edition : The Hardware/Software Interface. **ISBN**: 9780128226742

Required: Arduino kit to be used in laboratory experiments and project.

Arduino Mega 2560 Rev3



Competencies:

Upon successful completion of the course, the student will

1. **The student will demonstrate an understanding of the history, key terminology, and current industry trends in computer architecture by:**
    a. Describing the history and development of computational devices.
    b. Identifying the different classes of computing applications and their characteristics (e.g., desktop computers, servers, and embedded computers).
    c. Listing the classic components of a computational system (e.g., Input, Output, Memory, Datapath, Control) and understanding their functionality.
    d. Discussing the implications of the evolution of transistor sizes and the impending Power Wall.
    e. Discussing the significance of switching from uniprocessors to multiprocessors.
    f. Identifying the steps of the manufacturing process of integrated circuits.
    g. Calculating performance measures of a system using standard industry metrics such as Cycles Per Instruction (CPI), Instructions Per Cycle (IPC), and Speedup.

2. **The student will demonstrate an understanding of how data is represented within a computer system and instructions set fundamentals by:**
    a. Identifying and performing operations as implemented in computer hardware.
    b. Converting between decimal, binary, and hexadecimal representations.
    c. Converting between signed and unsigned numbers.
    d. Converting between decimal and floating-point notation.
    e. Identifying and performing logical operations as implemented in hardware.
    f. Distinguishing between Reduced Instruction Set Computer (RISC) and Complex Instruction Set Computer (CISC) instruction sets and identifying advantages/disadvantages of each.
    g. Identifying the different ISA addressing modes of instruction set architecture (e.g., MIPS).
    h. Converting between binary, assembly code, and higher-level instructions.
    i. Comparing various programming languages including compiled and interpreted languages.
    j. Comparing the implications of 32-bit instruction set architectures (ISAs) versus 64-bit ISAs.
    k. Enumerating the major ISAs prevalent in industry.
3. **The student will demonstrate an understanding of the processor (CPU) subsystem by:**
    a. Identifying the major components of a CPU (e.g., instruction/data cache, registers, ALU).
    b. Describing the stages of the common five-stage pipeline (Instruction Fetch, Instruction Decode, Execute, Memory, Write Back).
    c. Identifying datapaths for various operations including arithmetic operations of a microprocessor.
    d. Identifying control paths for various operations of a microprocessor.
    e. Describing the implementation of pipelining datapaths and controllers.
    f. Identifying data hazards including Read after Write (RAW), Write after Write (WAW), Write after Read (WAR) and deciding between forwarding or stalling.
    g. Identifying and managing control hazards.
    h. Calculating the theoretical speed-up from pipelining.

i. Identifying common forms of branch prediction (e.g., static, two-bit, bimodal, etc.).
j. Determining the success rate of a given branch predictor for a sequence of code.
k. Discussing advanced instruction-level parallelism concepts such as static multiple issue, dynamic multiple issue (superscalar), speculation, out-of-order execution, and register renaming.

4. **The student will demonstrate an understanding of the memory subsystem and hierarchy by:**
    a. Discussing the need for multilevel hierarchical memory design including registers, cache, main memory, and secondary storage.
    b. Identifying the basics of caches including direct-mapped, set-associative, fully associative, and common replacement policies.
    c. Designing caches of various sizes and associativity.
    d. Tracing the hits and misses given a cache scheme and a pattern of memory addresses.
    e. Calculating the hit and miss rate of a cache given a cache scheme and a pattern of memory accesses.
    f. Describing the various methods of cache coherence.
    g. Discussing the performance and security implications of virtual memory.
    h. Distinguishing between virtual and physical memory addresses for given physical memory configurations.
    i. Discussing the Translation Lookaside Buffer (TLB) and the process by which hardware translates from a virtual memory address to a physical memory address.
    j. Calculating slowdown due to memory speeds.

5. **The student will demonstrate an understanding of the various types of storage and the Input/Output (I/O) subsystem by:**
    a. Identifying the main aspects of storage (e.g., Dependability, Reliability, and Availability).
    b. Distinguishing between persistent versus non-persistent storage (memory vs. I/O).
    c. Calculating disk capacity given sectors, heads, etc.
    d. Identifying the various RAID configurations.
    e. Describing the process by which the processor, memory, and I/O devices are connected.
    f. Determining I/O performance measures.
6. **The student will demonstrate an understanding of the issues associated with multiprocessors and benchmarking by:**
    a. Discussing the difficulties faced in creating parallel processing programs.
    b. Describing the implication of shared-memory multiprocessors and multicore devices.
    c. Distinguishing clusters and other message-passing multiprocessors.
    d. Describing the benefit of hardware multithreading.
    e. Calculating the theoretical speedup from N processing cores versus 1 processing core.
    f. Identifying specialized multi-processing architectures (e.g., SISD, MIMD, SIMD, SPMD, and Vectors).
    g. Articulating the need for benchmarking and benchmark suites.
    h. Describing the relevance, benefits, and limitations of benchmarking.
    i. Identifying popular benchmarking suites available for use.

7. **The student will demonstrate practical skills related to computer architecture research by:**
    a. Determining the "instruction mix" of industry-standard benchmarks through the use of profiling tools.
    b. Experimentally verifying the performance of cache modifications such as replacement policy, size, and associativity using industry-standard benchmarks (e.g., SPEC) and simulation software (e.g., SimpleScalar, M5, etc.).
    c. Experimentally verifying the performance of branch prediction modifications such as different schemes (e.g., perfect, always taken, always not taken, bi-modal, etc.) and sizes using industry-standard benchmarks (e.g., SPEC) and simulation software (e.g., SimpleScalar, M5, etc.).
    d. Presenting a research paper from a recent industry conference (e.g., ASPLOS, ISCA) or industry journal (e.g., IEEE Transactions, SIGARCH, SIGGRAPH).

Important Note:

Students are required to read the assigned sections and complete homework as assigned by the instructor. Additionally, the above outlines and objectives are to be used as a guideline for the materials intended to be covered during the term. It may be necessary, due to extenuating circumstances, to change the order or content of the material. It is ultimately the responsibility of the student to keep up with class activities / changes and attend class every lecture.

Software Applications/Tools

Arduino IDE 2.3.2 - **https://www.arduino.cc/en/software**
CodeBlocks Arduino IDE - **http://arduinodev.com/codeblocks/**
GitHub - **https://github.com/lsuarez9/cet3126c**

Course Evaluation/Grading Policy/Assessment Methods/Schedule

Assessment	Graded Points	Percent of Final Grades
1 Exam	30	30%
Independent Study/Homework	10	10%
6 Labs (10 Points/lab)	60	60%
Total	100	100%

Grading Scale	
Grade = A	90 - 100%
Grade = B	80 - 89%
Grade = C	70 - 79%
Grade = D	60 - 69%
Grade = F	0 - 59%

Course/Departmental Policies

- Attendance: You are allowed up to three absences, after which you will be purged from the class roster.
- After 15 minutes of the beginning of the class you will be marked absent.
- Assignments and projects are due as scheduled and will be collected at the beginning of the class. Projects may be handed in late, upon approval from the instructor.
- Extra credit will be assigned at the instructor's discretion.
- No Makeup work is accepted. Extenuating circumstances are subject to verification and professor discretion to accept assignments on a one-time exception.
- Academic honesty: The school policy will be adhered to. Evidence of cheating directly or indirectly will result in an "F" for the given assignment, project, etc.
- Students with Special Needs should contact the instructor to make the necessary provisions.
- Student feedback of instructor will be conducted throughout the semester.

Student ACCESS Statement:

By providing a variety of services that address a spectrum of disabilities, the ACCESS department works to ensure equal access and opportunity throughout the college experience. Students with a documented disability are encouraged to contact the campus ACCESS (Disability Services) Department **in advance** for information on appropriate policies and procedures for obtaining assistance. Retroactive auxiliary aids and services cannot be provided. The ACCESS department is in Building 1, Room 1113, and can be reached at (305) 237-4027. Please note, it is the **student's responsibility to self-identify** at each campus where they are taking courses and seeking services.