# *CS 3443: Computer Systems*

**Required Course:** Required
**Course Number:** CS 3443
**Course Name:** Computer Systems
**Credit Hours:** 3.0
**Lecture Hours:** 3
**Lab Hours:** 0
**Instructors:** Dr. Shital Joshi

**Book Title(s):** Computer Organization and Design: The Hardware/Software Interface, 5th edition
**Book Author(s):** David Patterson and John Hennessy
**Book Year(s):** 2013

**Course Description:** Functional and register level description of computer systems, computer structures, addressing techniques, macros, linkage, input-output operations. Introduction to file processing operations and auxiliary storage devices. Programming assignments are implemented in assembly language.

**Course Prerequisites:** CS 2133 (Computer Science II) with a grade of 'C' or better.

**Course Goals:** The overall goal of the course is to enable students to analyze and design the structure and function of various components of modern computing systems. The students should learn:

- key skills of constructing cost-effective computer systems.
- to quantitatively evaluate different designs and organizations and provide quantitative arguments in evaluating different designs
- to articulate design issues in the development of processor or other components that satisfy design requirements and objective
- to analyze the tradeoffs in Instruction Set Architecture design using the MIPS assembly language as an example.
- Design and analyze the datapath and CPU control for a subset of the MIPS assembly language
- Demonstrate programming proficiency using various addressing modes and data transfer instruction of the target computers.

- Understand various conventional computational organizations and their strengths and weaknesses.
- Understand the concept of memory hierarchy.
- Understand how I/O devices interface with the processor, memory.
- Understand interrupts and how can they be handled.
- Understand how pipelining can improve CPU performance for MIPS architecture

**Student Outcomes:**

This class addresses the following student outcomes from the criteria for accrediting computing programs:

| Student Outcome | Course Outcome |
|---|---|
| 1 | • Analyze the performance of computer system in terms of commonly used metrics like CPU execution time, MIPS, MFLOPS, power/energy consumption, reliability and speedup resulting from system optimization using Amdahl's law.<br>• Determine the applicability of single cycle (MIPS), multi-cycle (MIPS), parallel, pipelined, superscalar, and RISC/CISC architectures<br>• Analyze cost performance and design trade-offs in designing and constructing a computer processor including memory. |
| 2 | • Analyze the tradeoffs in Instruction Set Architecture design using MIPS assembly language as an example.<br>• Design and analyze the datapath and CPU control for a subset of the MIPS assembly language.<br>• Translate C/C++ code into assembly language and perform simple optimizations by hand.<br>• Perform trace and debug at the assembly level. |
| 3 | • Understand the impact of instruction set architecture on cost-performance of computer design.<br>• Understand some of the design issues in terms of speed, technology, cost, performance. |
| 4 | • Understand contemporary microprocessor designs and identify various design techniques employed. |
| 6 | • Design and functional analysis of common combinational/sequential logic circuits such as adders, decoders, encoders, multiplexers and demultiplexers, flip flops.<br>• Design a pipeline for consistent execution of instructions with minimum hazards and incorporate long latency operation.<br>• Program using the capabilities of the stack, program counter and registers and understand how these are used to execute a machine code program. |

**Course Topics:**

**Knowledge Areas that contain topics and learning outcomes covered in the course:**

| **Knowledge Area (KA)** | **Total Hours of coverage** |
|---|---|
| AR/ Digital Logic and Digital Systems | 3 |
| AR/ Machine Level Representation of Data | 3 |
| AR/ Assembly Level Machine Organization | 10 |
| AR/ Memory System Organization and Architecture | 5 |
| AR/ Interfacing and Communication | 4 |
| AR/ Functional Organization | 4 |
| AR/ Multiprocessing and Alternative Architecture | 2 |
| SF/ Evaluation | 3 |

**Body of Knowledge coverage:**

| **KA** | **Knowledge Unit** | **Topic Covered** |
|---|---|---|
| AR | Digital Logic and Digital Systems | • Overview and history of computer architecture<br>• Combinational vs. sequential logic<br>• Multiple representations/layers of interpretation (hardware is just another layer)<br>• Physical constraints (gate delays, fan-in, fan-out, energy/power) |
| AR | Machine Level Representation of Data | • Bits, bytes, and words<br>• Numeric data representation and number bases<br>• Fixed- and floating-point systems<br>• Signed and twos-complement representations |
| AR | Assembly Level Machine Organization | • Basic organization of the von Neumann machine<br>• Control unit; instruction fetch, decode, and execution<br>• Instruction sets and types (data manipulation, control, I/O)<br>• Assembly/machine language programming<br>• Instruction formats<br>• Addressing modes<br>• Subroutine call and return mechanisms<br>• I/O and interrupts<br>• Heap vs. Static vs. Stack vs. Code segments<br>• Shared memory multiprocessors/multicore organization |
| AR | Memory System Organization and Architecture | • Storage systems and their technology<br>• Memory hierarchy: importance of temporal and spatial locality<br>• Main memory organization and operations |

| | | |
|---|---|---|
| | | • Latency, cycle time, bandwidth, and interleaving<br>• Cache memories (address mapping, block size, replacement and store policy)<br>• Multiprocessor cache consistency/Using the memory system for inter-core synchronization/atomic memory operations<br>• Virtual memory (page table, TLB) |
| AR | Interfacing and Communication | • I/O fundamentals: handshaking, buffering, programmed I/O, interrupt driven I/O<br>• Interrupt structures: vectored and prioritized, interrupt acknowledgment<br>• External storage, physical organization, and drives<br>• Buses: bus protocols, arbitration, direct memory access (DMA)<br>• Introduction to networks: communications networks as another layer of remote access |
| AR | Functional Organization | • Implementation of simple datapaths, including instruction pipelining, hazard detection and resolution<br>• Control unit: hardwired realization vs. microprogrammed realization<br>• Instruction pipelining |
| AR | Multiprocessing and Alternative Architecture | • Shared multiprocessor memory systems and memory consistency<br>• Multiprocessor cache coherence |
| SF | Evaluation | • Performance figures of merit<br>• Workloads and representative benchmarks, and methods of collecting and analyzing performance figures of merit<br>• CPI (Cycles per Instruction) equation as tool for understanding tradeoffs in the design of instruction sets, processor pipelines, and memory system organizations.<br>• Amdahl's Law: the part of the computation that cannot be sped up |