CMP1203 Computer Architecture and Organization

Period per Week			Contact Hour per Semester	Weighted Total Mark	Weighted Exam Mark	Weighted Continuous Assessment Mark	Credit Units
LH	PH	TH	CH	WTM	WEM	WCM	CU
45	00	30	60	100	40	100	4

Rationale

Computer architecture is a key component of computer engineering and the practicing computer engineer should have a practical understanding of this topic. It is concerned with all aspects of the design and organization of the central processing unit and the integration of the CPU into the computer system itself.

Objectives

- To provide an overview of computer architecture and teach students the operation of a typical computing machine.
- To cover basic principles, while acknowledging the complexity of existing commercial systems.
- To reinforce topics that are common to other areas of computer engineering; for example, teaching register indirect addressing reinforces the concept of pointers in C.
- To expose the student to how various peripheral devices interact with, and how they are interfaced to a CPU.

Course Content

1. History and Overview

- Reasons for study
- Influences on Computer Architecture Organization
- System organization and architecture, memory, interfacing, microprocessors, and performance; Differences between computer organization and architecture; binary arithmetic
- Components of computer design; CPU organization; Caching; Multiprocessing strategies

2. ***Fundamentals of computer architecture***
   - Organization of the von Neumann machine, Instruction formats, The fetch/execute cycle; instruction decoding and execution , Registers and register files, Instruction types and addressing modes, Subroutine call and return mechanisms, Programming in assembly language, I/O techniques and interrupts
3. ***Computer Arithmetic***
   - Representation of integers (positive and negative numbers), Algorithms for common arithmetic operations (addition, subtraction, multiplication, division)
   - Significance of range, precision, and accuracy in computer arithmetic, Representation of real numbers (standards for floating-point arithmetic)
   - Algorithms for carrying out common floating-point operations, Converting between integer and real numbers , Multi-precision arithmetic
   - Hardware and software implementation of arithmetic unit
   - The generation of higher order functions from square roots to transcendental functions
4. ***Memory System Organization and Architecture***
   - Memory systems hierarchy, Coding, data compression, and data integrity , Electronic, magnetic and optical technologies
   - Main memory organization and its characteristics and performance, Latency, cycle time, bandwidth, and interleaving, Cache memories (address mapping, line size, replacement and write-back policies), Virtual memory systems
   - Memory technologies such as DRAM, EPROM, and FLASH, Reliability of memory systems; error detecting and error correcting systems
5. ***Interfacing and Communication***
   - I/O fundamentals: handshaking, buffering
   - I/O techniques: programmed I/O, interrupt-driven I/O, DMA
   - Interrupt structures: vectored and prioritized, interrupt overhead, interrupts and reentrant code
   - Memory system design and interfacing
   - Buses: bus protocols, local and geographic arbitration
6. ***Device Subsystems***
   - External storage systems
   - Organization and structure of disk drives and optical memory
   - Basic I/O controllers such as a keyboard and a mouse, RAID architectures, Video control, I/O Performance, SMART technology and fault detection
   - Processor to network interfaces
7. ***Processor System Design***
   - The CPU interface: clock, control, data and address buses
   - Address decoding and memory interfacing
   - Basic parallel and serial interfaces

- Timers
- System firmware

8. ***Organization of the CPU***
   - Implementation of the von Neumann machine
   - Single vs. multiple bus data paths
   - Instruction set architecture
   - Machine architecture as a framework for encapsulating design decisions
   - Relationship between the architecture and the compiler
   - Implementing instructions
   - Control unit: hardwired realization vs. microprogrammed realization
   - Arithmetic units, for multiplication and division
   - Instruction pipelining
   - Trends in computer architecture: CISC, RISC, VLIW
   - Introduction to instruction-level parallelism (ILP)
   - Pipeline hazards: structural, data and control, reducing the effects of hazards.
9. ***Performance***
   - Metrics for computer performance; clock rate, MIPS, Cycles per instruction, benchmarks
   - Strengths and weaknesses of performance metrics
   - Averaging metrics: arithmetic, geometric and harmonic
   - The role of Amdahl's law in computer performance
10. ***Distributed System Models***
    - Classification of models: parallel machine models (SIMD, MIMD, SISD, MISD)
    - Flynn's taxonomy, Handler's classification, message passing , Granularity, levels of parallelism,
    - Multiprocessors and multi-computers
    - Topology, tightly coupled and loosely coupled architectures
    - Processes: threads, clients, servers, code migration, software agents,
    - Physical and logical clocks: clock synchronizing algorithms, Lamport timestamps, vector timestamps,
    - Election Algorithms,
    - Mutual Exclusion algorithms,
    - Distributed transactions: models, classification, concurrency control

Learning Outcomes

On completion of this course the student should be able to:

- Identify some contributors to computer architecture and organization and relate their achievements to the knowledge area; explain the reasons and strategies for different architectures; articulate differences between computer organization and computer architecture; identify some of the components of a computer; indicate some strengths and weaknesses inherent in different architectures; and describe how computer engineering uses or benefits from computer architecture and organization.
- Explain the organization of a von Neumann machine and its major functional units; explain how a computer fetches from memory and executes an instruction; articulate the strengths and weaknesses of the von Neumann architecture; explain the relationship between the representation of machine level operation at the binary level and their representation by a symbolic assembler; explain why a

designer adopted a given different instruction formats, such as the number of addresses per instruction and variable length vs. fixed length formats; write small programs and fragments of assembly language code to demonstrate an understanding of machine level operations; implement some fundamental high-level programming constructs at the machine-language level; and use computer simulation packages to investigate assembly language programming

- Appreciate how numerical values are represented in digital computers; understand the limitations of computer arithmetic and the effects of errors on calculations; and appreciate the effect of a processor's arithmetic unit on its overall performance
- Identify the main types of memory technology; explain the effect of memory latency and bandwidth on performance; explain the use of memory hierarchy to reduce the effective memory latency; describe the principles of memory management; and understand how errors in memory systems arise and how to resolve them.
- Explain how to use interrupts to implement I/O control and data transfers; write small interrupt service routines and I/O drivers using assembly language; identify various types of buses in a computer system; describe data access from a magnetic disk drive; and analyze and implement interfaces.
- Compute the various parameters of performance for standard I/O types; explain the basic nature human computer interaction devices; and describe data access from magnetic and optical disk drives
- Understand how a CPU chip becomes a complete system; design an interface to memory; understand how to interface and use peripheral chips; write sufficient EPROM-based system software to create a basic stand-alone system; and specify and design simple computer interfaces.
- Understand the factors that contribute to computer performance; understand the limitations of performance metrics; select the most appropriate performance metric when evaluating a computer; and discuss the impact on control and datapath design for performance enhancements.
- Explain the differences between different paradigms and their usefulness and applicability; understand how client server model works in a decentralized fashion; understand how agents work and how they solve simple tasks; understand the concept of logical clocks vs. physical clocks and how they affect implementation of distributed systems and be familiar with simple election and mutual exclusion algorithms and their applicability.
- Discuss how various architectural enhancements affect system performance; discuss how to apply parallel processing approaches to design scalar and superscalar processors; discuss how to apply vector-processing techniques to enhance instruction sets for multimedia and signal processing; understand how each of the functional parts of a computer system affects its overall performance; and estimate the effect on system performance of changes to functional units.

Recommended and Reference Books

[1] Linda Null and Julia Lubor. *The Essentials of Computer Organization and Architecture*, Second Edition, Jones & Bartlett. ISBN-10: 0763737690, ISBN-13: 978-0763737696

[2] Harold Lorin, 1982. *Introduction to Computer Architecture and Organization*, Wiley.
ISBN 0471866792, 9780471866794

[3] John L. Hennessy, 2002. *Computer Architecture: A Quantitative Approach*, 3rd Edition, Morgan Kaufmann, ISBN-10 / ASIN: 1558605967, ISBN-13 / EAN: 9781558605961

[4] William Stallings, 2002. *Computer Organization and Architecture,* 6th edition, Prentice Hall. ISBN-10 / ASIN: 0130351199, ISBN-13 / EAN: 9780130351197

[5] Computer Design and Architecture, 3rd Edition Revised and Expanded CRC. ISBN: 0824703685

[6] Mostafa Abd-El-Barr, 2004. *Fundamentals of Computer Organization and Architecture (Wiley Series on Parallel and Distributed Computing)*, Wiley- Interscience, ISBN-10 / ASIN: 0471467413, ISBN-13 / EAN: 9780471467410

[7] Douglas E. Comer, 2004. *Essentials of Computer Architecture,* US edition, Prentice Hall
ISBN-10: 0131491792, ISBN-13: 978-0131491793