

CS470 – Computer Architecture I

Last Updated - 05/08/02

Course Manager - Virgil Bistriceanu, Instructor

3 credit hours; elective for CS, required for CPE; 100 min. lecture & 100 min. lab each week

Current Catalog Description - Introduction to the functional elements and structures of digital computers. Detailed study of specific machines at the register transfer level illustrates arithmetic, memory, I/O, and instruction processing. Prerequisites: CS 350 and ECE 218. (2-2-3)

Textbook

- David A. Patterson, John L. Hennessy, *Computer Organization and Design: the hardware/software interface*, second edition, Morgan Kaufmann, Inc. 1994, ISBN 1-55860-491-X

References - other textbooks or materials

- Harvey G. Cragon, *Memory Systems and Pipelined Processors*, Jones and Bartlett, 1995, ISBN 0-86720-474-5
- Michael J. Flynn, *Computer Architecture*, Jones and Bartlett, 1995, ISBN 0-86720-204-1
- Vincent P. Heuring, Harry F. Jordan, *Computer Systems Design and Architecture*, Addison Wesley, 1997, ISBN 0-201-89589-7

Course Goals

Students should be able to:

- Present the milestones of computer architecture history
- Fundamentals of computer design
 - Explain the difference between various measure of performance: Latency, throughput; MIPS, MPFLOS
 - Comparing performance
 - Utilize Amdahl's law to estimate the overall speedup
 - Explain the difference between a good and a bad benchmark
- Assembly level machine organization
 - Explain the basic organization of the classical von Neumann machine and its major functional units
 - Explain how an instruction is executed in a classical von Neumann machine
 - Summarize how instructions are represented at both the machine level and in the context of a symbolic assembler
 - Explain different Instruction Set formats (0 (stack), 1 (accumulator), 2, and 3-addresses per instruction; Variable length vs. fixed length formats)
 - Design the Instruction Set for a general purpose CPU
 - Explain how the basic addressing modes work: Register, Memory direct, Memory indirect, Base and displacement, Indexed
 - Explain how base and displacement addressing is used in block-based programming languages
 - Write small MIPS assembly language programs
 - Demonstrate how fundamental high-level programming constructs are implemented at the machine-language level: If-then-else, Loops (for, while, do-until), Procedure call/return
 - Explain the basic concepts of interrupts and I/O operations
- Datapath and Control

- Design a single clock-cycle datapath for a CPU
- Explain why a single clock-cycle datapath is inefficient
- Re-factor a single clock-cycle datapath into a multi clock-cycle one
- Explain the difference between a hardwired and a microprogrammed control unit
- Design the control unit for a single clock-cycle datapath
- Explain how exceptions impact the design and performance of a datapath
- Pipelining
 - Derive the formula for the throughput of an ideal pipeline with N stages
 - Explain the limiting factors in building a pipeline with too many stages
 - Explain how data and control hazards occur and how their impact can be eliminated or reduced
 - Re-factor MIPS code to reduce/eliminate data and branch hazards
 - Explain the significance of a late commit in the pipeline
 - Explain the changes in the design and implementation of a pipelined datapath to account for exceptions
 - Explain branch prediction
 - Solve problems that require finding the *real* CPI of a program running on a pipelined datapath
- The memory hierarchy
 - Identify the main types of memory technology and explain the trade-off in using them
 - Explain the effect of memory latency on running time
 - Explain the use of memory hierarchy to reduce the effective memory latency
 - Explain the differences between different cache organizations: Direct mapped, Set associative Fully associative
 - Utilize a cache simulator and access traces to compare the performance of caches with different sizes and organizations
 - Explain main memory organization alternatives to improve performance: Wide-memory, Interleaving
 - Explain the impact of access stride to performance
 - Explain the virtual memory structure and mapping
 - Explain why and how virtual memory impacts performance and how performance can be improved. TLB
 - Analyze the differences between cache organizations in systems with virtual memory: Real address caches, Pipelined real caches, Virtual address cache, Restricted virtual caches, TLB addressing
- I/O
 - Define the meaning of various I/O performance measures
 - Types and characteristics of I/O devices
 - Explain the differences between major buses (IDE, SCSI, USB, PCI): synchronous v. asynchronous, Serial v. parallel, Number of devices, Termination, Transfer rates
 - Design issues related to I/O system addressing: Memory-mapped I/O, Cache coherency, Snoopy controllers, DMA I/O configurations
 - Explain the sources of latency in a I/O subsystem

Prerequisites by Topic

- Basic understanding of a von-Neumann computer organization
- The ability to explain the differences between a high level instruction and a compiled instruction
- Knowledge of the steps involved in the execution of an instruction
- Solid understanding of basic building blocks for a datapath: ALU, register, counter, multiplexer, decoder, glue logic
- Working knowledge of Boolean logic

Major Topics Covered in Course

1. Overview and history of computer architecture	1 hour
2. Fundamentals of computer design	3 hours
3. Basic organization of a von Neumann computer	1 hours
4. Instruction Set design	3 hours
5. Datapath and Control	4 hours
6. Pipelining	5 hours
7. The memory hierarchy	4 hours
8. I/O	4 hours
Introduction: discuss class structure, objectives, and requirements, Midterm	3 hours
Project presentation	2 hours
Laboratory	30 hours
Final Exam	-
	Total 60 hours

Laboratory projects (specify number of weeks on each)

The purpose of the lab is to give the students first hand experience with the instruction set of a selected (RISC) reference architecture. The lab uses freely available simulators that run under both UNIX and MS-Windows, thus allowing students to do work at home without any of the hassles of being remotely connected to the Department's network.

Each lab has three sections, pre-lab (to do at home in preparation for the in-lab section), in-lab, and post-lab (to do at home).

Introduction: tools of the trade	2 hours
Introduction to MIPS	2 hours
The Virtual Machine	2 hours
Control Structures	4 hours
Procedures	6 hours
Memory	4 hours
Arithmetic	4 hours
Exceptions	6 hours
	Total 30 hours

Estimate CSAB Category Content in Credit Hours

	CORE	ADVANCED		CORE	ADVANCED
Data Structures			Computer Organization and Architecture		3
Algorithms			Concepts of Programming Languages		
Software Design					

Oral and Written Communications - Every student is required to submit at least one written report (the project report) of typically 10 pages and to make 1 oral presentation of typically 20 minutes duration. The best three projects are presented in class by their respective teams.

Social and Ethical Issues - Please list the topics that address the social and ethical implications of computing covered in all course sections. Estimate the class time spent on each topic. In what ways are the students in this course graded on their understanding of these topics (e.g., test questions, essays, oral presentations, and so forth)?

- None

Theoretical Foundations - Please list the types of theoretical material covered, and estimate the time devoted to such coverage in contact (lecture and lab) hours.

- 40% theory
- 10% case studies and project
- 50% laboratory

Problem Analysis - Please describe the problem analysis experiences common to all course sections.

- Case studies:
 - Analyze the impact (hardware and performance) of adding a new instruction to the MIPS instruction set
 - Pipeline structure for two major RISC architectures/implementations (e.g. MIPS, Sparc, Pentium)
 - Cache organization for two major CPU implementations (e.g. Pentium, Sparc)

Solution Design - Please describe the design experiences common to all course sections

- Students work in groups of two to design the instruction set and the datapath for a 4-bit, general purpose, CPU. General issues related to design (limited encoding space, small memory, memory mapped I/O, etc.) are addressed during lecture. Extensive design support provided during office hours.

Other Course Information

- Additional Suggested Course Assignments
 - 4 homework assignments
- Planned Course Enhancements
 - Possible new course description - Understand the fundamentals of computing by studying the interaction between hardware and software at various levels. Discuss the design trade-offs that drive the performance of computer systems. Topics covered include Performance Definition, Instruction Set Design, Datapath and Control, Pipelining, the Memory Hierarchy, Input/Output systems. Prerequisite: CS 350 and ECE 218. (2-2-3)
 - Departmental coordination is required to address existing some overlaps between cs350 and cs470 and to better define the objectives of cs350, cs470, and cs471. Interdepartmental coordination may be needed to address similar issues between cs350, cs470, cs470 and classes taught by the EE Department, e.g. ece218, ece242, etc.
 - Planned enhancement on topics to be covered by CS470 (Fall 2002)

1. Overview and history of computer architecture	1 hour
2. Fundamentals of computer design	2 hours
3. Basic organization of a von Neumann computer	1 hours
4. Instruction Set design	2 hours
5. Datapath and Control	4 hours
6. Pipelining	5 hours
7. The memory hierarchy: Memory latency, cache organizations, virtual memory	4 hours
8. I/O: I/O performance measures, different I/O devices, major buses, I/O system addressing, latency in a I/O subsystem	4 hours
9. Multiprocessors	2 hours
Introduction: discuss class structure, objectives, and requirements, Midterm	3 hours
Project presentation	2 hours
Laboratory	30 hours
Final Exam	-

Total 60 hours