KAU CPCS 214 • 2011–2012

Computer Organization and Architecture

Dr. Muhammad Al-Hashimi

Credits 3

Prerequisite CPCS 211 • C language basics

Homepage www.hashimi.ws/cs214

Email cs214@hashimi.ws

Phone 640-0000 ✆ Ext. 52436

Office Room 202 • Building 31

Office Hours Visit homepage for current times

This first course on computer organization and architecture for computer science major is intended to explain how computers are designed and how they work. Students are introduced to modern computer principles using a typical processor. They learn how efficient memory systems are designed to work closely with the processor, and how input/output (I/O) systems bring the processor and memory together with a wide range of devices. The course emphasizes system-level issues and understanding program performance, and the use of abstraction as a tool to manage complexity.

Topics Material designed for 13 teaching weeks. Check the homepage for current teaching schedule.

- ✎ **Instruction Sets** Registers, ops and operands, assembly (symbolic) and machine instructions, addressing modes, instruction families, machine instruction design, instruction set architecture.

- ✎ **Computer Arithmetic** Signed and unsigned arithmetic, floating point arithmetic and accuracy, arithmetic instructions.

- ✎ **Performance Basics** Performance measurement and reporting, Amdahl's law.

- ✎ **Processors** ALU, datapath, control, exceptions and interrupts, processor performance, instruction overlapping, pipelined execution, pipeline hazards, introduction to static multiple-issue, superscalar and dynamically-scheduled pipelines.

- ✎ **Memory** Locality of reference, memory hierarchies, cache organization, cache design and performance, interfacing with main memory, virtual memory, virtual address translation, page management, page faults, fast traslation and TLB, integration with cache and TLB.

- ✎ **I/O Systems** Requirements, I/O device characteristics, interconnect systems (shared bus, point-to-point), addressing I/O devices (dedicated, memory-mapped), interfacing I/O (programmed, interrupt- driven, DMA).

- ✎ **Parallel Processing** An introduction to multiprocessing, multicores, clusters and GPU.

Textbook David A. Patterson and John L. Hennessy, *Computer Organization & Design: The Hardware/Software Interface*, Morgan Kaufmann Publishers, 4th Edition, 2008 ISBN: 978-0123744937

Assessment Details on the homepage.

15%	Test 1
15%	Test 2
20%	Lab and homework assignments
10%	Reading discussion assignments
40%	Final exam

Learning Resources Check the homepage for the latest lecture schedule, summary and slides, discussion forums, software tools and online supporting resources.

Learning Outcomes Broadly (check lecture summaries for detailed learning outcomes):

1. Determine how typical high level data types are organized in memory.
2. Compile generic C-based assignments, conditionals, loops, and procedures into MIPS assembly using proper MIPS calling conventions.
3. Describe the principles and decisions used to design MIPS instructions.
4. Write the binary representation for selected MIPS assembly instructions given opcode and register code tables.
5. Trace the execution of a small MIPS program and its supporting data structures.
6. Write simple MIPS assembly program using MIPS software conventions.
7. Examine how hardware interprets and handles a sequence of bits, and the consequences of finite store on representation and operations on integer and real numbers.
8. Convert between numbers and their internal bit representation.
9. Identify the design principles, choices and limitations of IEEE 754 representation of real numbers.
10. Perform step-by-step straight cases of floating point addition and multiplication.
11. Explain the function of a clocking signal in computers.

12. Specify the functional definition of various building blocks, pieces, and units used in selected datapath designs.
13. Trace the execution of instructions in the example datapaths.
14. Identify the components of CPU performance based on the CPU performance equation.
15. Explain the basic principles behind pipelined execution.
16. Determine control and selected data hazards in a MIPS code fragment.
17. Compute the CPI for a code fragment given selected pipelining scenarios.
18. Compare memory technologies available for computing systems.
19. Give examples of spatial and temporal locality in both data and code (instructions).
20. Explain the basic principle behind a memory hierarchy.
21. Describe the various ways to organize a cache memory.
22. Explain the steps and issues involved in performing reads and writes through cache.
23. Determine the location in cache given a memory address and a cache organization.
24. Calculate the storage requirements of a cache given its size and organization.
25. Calculate memory stall cycles due to cache given cache miss rate and miss penalty.
26. Determine the effects of various cache design parameters on cache performance.
27. Determine the physical address given a virtual address and page size.
28. Describe the steps, structures and issues involved in virtual address translation.
29. Identify the design characteristics and choices involved in page management.
30. Explain the steps and issues involved in page faults.
31. Describe the role of the operating system in managing virtual memory.
32. Identify issues, requirements and challenges involved in designing I/O systems.
33. Explain how dependability can be achieved in a computing system.
34. Describe the characteristics and operation of a typical I/O device (hard disk is selected).
35. Compare shared bus and communication hub-based, point-to-point I/O interconnect systems.
36. Describe the characteristics of communication over typical bus interconnect.
37. Describe the steps and issues involved in completing a program I/O request.
38. Describe the various methods used to address and interface I/O devices.
39. Describe the main goals, issues and difficulties faced by multiprocessors.
40. Compare the characteristics of shared memory and message-passing multiprocessors.

References Check homepage for complete list.

1. Nicholas Carter, *Schaum's Outlines of Computer Architecture*, McGraw-Hill, 2002.
   ISBN: 007136207X

2. Sivarama P. Dandamudi, *Guide to RISC Processors for Programmers and Engineers*, Springer 2005. ISBN: 0387210172

Rev 1.0