KAU CS 212 • 2007

Computer Organization

Dr. Muhammad Al-Hashimi

Credits 3

Homepage www.hashimi.ws/cs212

Email cs212@hashimi.ws

Phone 695-1223 • 695-2000 ✆ Ext. 51223

Office Faculty of Science • Room 577

Prerequisite CS 111 • basic C programming.

ACM/IEEE CC2001 AR2–AR6, AR8.

This sophomore-level course explains how computers are designed and how they work. We study modern computer design principles using a typical processor. We then learn how efficient memory systems are designed to work closely with the processor. Finally, we study input/output (I/O) systems which bring the processor and memory together with a wide range of devices. The course emphasizes system-level issues, understanding program performance, and the use of abstraction as a tool to manage complexity.

Topics Material designed for 14 teaching weeks. Check the homepage for current teaching schedule.

- ✎ **Preliminary** Programming concepts review, abstraction and computer design.
- ✎ **Performance Evaluation** Performance measurement and reporting, execution times and rates, Amdahl's law.
- ✎ **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 integer arithmetic, floating point numbers design, arithmetic, and accuracy, arithmetic instructions.
- ✎ **Processors** Basic integer ALU, datapath design, hard-wired and microprogrammed control, processor performance, exceptions and interrupts, instruction overlapping, pipelined execution, pipeline hazards, multiple-issue, superscalar and dynamically-scheduled pipelines.
- ✎ **Memory** Locality of reference, memory hierarchies, cache design, interfacing cache with main memory, cache performance, virtual memory, address mapping [=translation], page management, page faults, VM performance, integration with cache.
- ✎ **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), I/O performance.

Textbook David A. Patterson and John L. Hennessy, *Computer Organization & Design: The Hardware/Software Interface*, 3rd edition, Morgan Kaufmann Publishers, 2004. ISBN: 1558606041

Assessment Personal best 3 out of 4 tests, programming and group discussion assignments. Details in FAQ on the homepage.

40%	Tests
10%	Programming assignment
10%	Reading assignments
40%	Final exam

Learning Resources Check the homepage for the latest lecture schedule, summary and slides, discussion forums, audio clips, software tools, textbook resources, online guides and supporting web links.

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

- ✎ Examine how instructions and data are stored and processed from high level to machine level.
- ✎ Examine the assembly interface to hardware.
- ✎ Show how processors implement instruction set architecture specifications.
- ✎ Explain how locality of reference is utilized to build effective memory systems from different storage technologies.
- ✎ Examine how memory systems interact with the processor to execute instructions efficiently.
- ✎ Describe how I/O devices interface to the processor and memory.
- ✎ Identify key performance issues and how they influence design and implementation
- ✎ Appreciate how abstraction is used to manage design complexity.

References Check homepage for complete list.

1. Nicholas Carter, *Schaum's Outlines of Computer Architecture*, McGraw-Hill, 2002. ISBN: 007136207X
2. James Good and Karen Miller, *A Programmer's View of Computer Architecture with Assembly Language Examples from the MIPS RISC Architecture*, Oxford University Press, 1993. ISBN: 0195131096