ICS 233

Computer Architecture and Assembly Language

College of Computer Sciences & Engineering
King Fahd University of Petroleum & Minerals

Weekly Lecture and Lab Breakdown

Spring 2007 (Term 062)

Proposed By

Dr. Muhamed F. Mudawar

Last Updated: Wednesday, March 7, 2007

W	Lecture	Lab
1	**Introduction** • Course information, objectives, grading, etc. • Introduction to computer architecture • Assembly and machine languages • Assembly language tools • Components of a computer system • Memory hierarchy • Fetch-execute cycle • Chip manufacturing process • Technology trends • Programmer's view of a computer system	**Introduction to the Lab** • Information about the lab • Lab objective, grading, etc. • Tour inside the personal computer, motherboard, CPU, memory, hard disk, input devices, and output devices
2	**Review of Data Representation** • Binary, decimal, and hexadecimal numbers • Signed integers and 2's complement • Binary and hexadecimal addition/subtraction • Carry and overflow • Characters and ASCII table	**Assembly language tools** • Tools that will be used in the lab • **MARS** or **PCSPIM** tool • Example of an assembly language program • Edit-Assemble-Run-Debug cycle • Trying different instructions • Viewing machine instructions • Viewing content of registers • Tracing program execution
3	**Instruction Set Architecture** • Overview of the MIPS Processor • MIPS registers Instruction formats • R-type arithmetic, logical, & shift instructions • I-type format and immediate constants • Jump and conditional branch instructions	
4	**Instruction Set Architecture – cont'd** • Unsigned and signed comparisons • Pseudo Instructions • Translating **if** statements • Load and Store instructions • Memory alignment and byte ordering • Translating loops • Array indexing, pointers, and traversal • Addressing modes • RISC design principles	

5	**MIPS Assembly Language Programming** • Assembly language tools • Assembly language program template • Directives, text, data, and stack segments • Defining data, arrays, and strings • Indirect jump and jump table **Console Input/Output** • System services and the syscall instruction • Reading and writing an integer • Reading and writing a string • Reading and writing a float/double • Exiting a program	**MIPS Assembly I**
6	**Procedures and the Runtime Stack** • Runtime stack and its applications • Defining a procedure • Procedure call and return address • Nested procedure calls • Passing arguments in registers • Value and reference parameters • Saving and restoring registers • Stack parameters • Accessing parameters on the stack • Local variables on the stack • Accessing local variables on the stack • Stack frames • Application: sorting an integer array	**MIPS Assembly II**
7	**Integer Multiplication and Division** • Shift-add multiplication algorithm • Sequential multiplication hardware • Signed multiplication • MIPS integer multiply instructions • HI and LO registers • Shift-subtract division algorithm • Sequential division hardware • Signed division • MIPS integer division instructions • Multiplication and division by powers of 2	**Procedure and the Stack**

7 8	**Floating-Point Arithmetic** • Floating-point representation • Normalized and de-normalized numbers • IEEE 754: zero, infinity, and NaN • FP addition algorithm and hardware • FP multiplication algorithm and hardware • Rounding and accurate FP arithmetic **MIPS Floating-Point Coprocessor** • Floating-point registers • Floating-point instructions • Floating-point applications	**Integer Multiplication / Division** •
8	**CPU Performance** • CPU performance and Metrics • CPI and performance equation • MIPS performance metric • Amdahl's law • Benchmarks & performance of recent CPUs	**Floating-Point Arithmetic** •
9	**Datapath Design** • Designing a processor • Register transfer logic • Datapath components • Register file design • Instruction and data memories • Clocking methodology • Single-cycle datapath	**Introducing Logisim Tool** **Designing a register file**
10	**Control Unit Implementation** • Main control signals • ALU control • Control signal values of various instructions • Logic equations for control signals • Implementing the control unit • Estimating the longest delay and clock cycle • Single-cycle CPU performance	**Datapath Design**
11	**Pipelined Datapath and Control** • Pipelining concepts • Pipeline timing and performance • 5-stage MIPS pipeline • Pipelined datapath design • Pipelined control	**Control Unit Design**

12	**Pipeline Hazards** • Structural hazards • Data dependencies and hazards • Forwarding datapath and control signals • Load delay and stalling the pipeline • Compiler scheduling • Control hazards • Branch delay • Dynamic branch prediction • Branch target and prediction buffer	**Pipelined Datapath Design**
13	**Memory System Design** • Memory hierarchy • SRAM memory structure & technology • DRAM memory structure & technology • DRAM memory operation • Pipelined and Interleaved memory • Memory performance	**Pipeline Hazards**
14	**Cache Memory Design** • Locality of reference • Cache memory organization • Handling cache misses • Write policy and write buffer • Replacement policy • Cache performance • Memory stall cycles • AMAT • Two-level cache memory	**Cache Memory**
15	**Disk Storage?** **Advanced Architectures?** • •	•

	Interrupts • Software exceptions • SYSCALL instruction • Hardware interrupts • Interrupt processing and handler • MIPS Coprocessor 0 **ALU Design** • Hardware adders • Barrel shifter • Multifunction ALU Design	