CS 2521 : Computer Organization and Architecture (4)

Catalog Description:
Internal representation of programs and data. Computer organization and introduction to computer architecture. Machine and assembly language programming. Data and procedural structures. Addressing methods. Systems software including linking and loading. Introduction to hardware performance analysis and measurements.

Textbook: David Patterson and John Hennessy, *Computer Organization and Design: The Hardware/Software Interface, 4th Revised Ed.,* Morgan Kaufmann, 2009.

Course Goals:
This class is an introduction to computer organization and design. A student who successfully completes this course will understand how to carry out and interpret quantitative performance evaluations of computer systems, how programs and data are represented in a computer, the foundations of computer arithmetic, the design and use of single-cycle, multi-cycle and pipelined processors. In addition, students will be exposed to current issues in computer architecture such as superscalar and VLIW designs.

Prerequisites by Course & Topics
CS 1521: Computer Science II – recursion, linked lists
ECE 1315: Digital System Design – combinational and sequential logic circuits
MA 1296: Calculus I – mathematical maturity, critical thinking

Major Topics Covered in the Course
- Assembly language
- Von Neumann model of computing
- Instruction Set Architecture (RISC and CISC)
- Computer Arithmetic
- Single Cycle Architecture
- Multi Cycle Architecture
- Pipelining
- SuperScalar/VLIW

Class/Laboratory Schedule: Lecture: 3 hours per week, Laboratory: 1

Course Outcomes
1. Become proficient in quantitative performance evaluation of computer systems.
   a. Analyze system performance at an overall level based on throughput and response time.
   b. Critique computer architecture design decisions based on measures such as cycles-per-instruction and instructions-per-cycle.
2. Understand how programs and data are stored and represented in a computer system.
   a. Engage in detailed study of instruction sets of modern computer architectures including MIPS and Intel x86.
   b. Master Von Neumann model of computation, compare with alternatives such as Harvard architecture.
   c. Understand memory hierarchy both with respect to physical organization and virtual memory as provided in modern operating systems.
   d. Understand scope and life span of activation records in a variety of contexts.
3. Master the foundations of computer arithmetic.
   a. Ability to use binary and hexadecimal number systems.
   b. Understand and implement fast integer multiplication methods such as Booth's algorithm.
   c. Detailed knowledge of floating point representation and arithmetic, including discussion of rounding and precision errors.
4. Understand the design and implementation of single-cycle, multi-cycle, pipelined, and super-scalar architectures.

a. Compare the performance of different computer architectures, and observe the effect on standard performance measures of different hardware implementation decisions.

5. Develop proficiency in assembly language programming.
   a. Knowledge of assembly language addressing modes and instruction formats.
   b. Understanding of processor functional units such as register file and arithmetic-logical unit.
   c. Ability to translate an assembly language program into machine language.

Relationship to Program Outcomes

CS 2521 is a required core course that is taken after successful completion of the introductory sequence, digital logic, and Calculus I. This course contributes to meeting the following program outcomes:

1. *Students understand the mathematics and statistics that underlie scientific applications.*
   CS 2521 provides students with an in-depth understanding of how integer and floating point computations are actually carried out at the hardware level. This allows the student to understand the limits of what can actually be computed both with respect to precision and range. Course outcomes 2 and 3 map to this program outcome.
3. *Students understand the fundamentals of computer organization and architecture, data structures and related algorithms, and programming languages.*
   CS 2521 provides a comprehensive introduction to computer organization and architecture. Students understand the motivation and design principles underlying the MIPS instruction set architecture (a RISC architecture), and as such synthesize the major design developments of the last 20 years in computer architecture. Course outcomes 1-5 map to this program outcome.
4. *Students can apply computer science principles and practices to a variety of problems.*
   CS 2521 requires extensive programming at the assembly language level, and as such requires students to employ their existing analysis and design skills in a totally new environment. Course outcomes 1-5 map to this program outcome.

Assessment Plan for Course:

This course is assessed every other year by the instructor and a course assessment document covering all of the course outcomes and their effect on the program outcomes is prepared.

Estimate CSAB Category Content

	CORE	ADVANCED		CORE	ADVANCED
Data Structures			Computer Organization and Architecture	3	
Algorithms			Concept of Programming Languages	1	
Software Design					

Coordinator/Prepared by: C. Prince