EE 361 Digital Systems and Computer Design

Designation: Required

Catalog Description: EE 361 Digital Systems and Computer Design (3) Design methodology, processor design, control design, memory organization, system organization. Pre: 213 and 260; or consent.

Credits: 3

Pre-and Co-requisites: Pre-requisites: EE 213 "Basic Circuit Analysis II" and EE 260 "Introduction to Digital Design"

Class/Lab Schedule: 3 lecture hours per week

Topics Covered:
• Principles of instruction set design, the role of compiler and optimizations, assembly language, machine level programs, how high level programs are implemented with machine instructions (10 hours)
• Review of digital circuits, elementary design methods, and hardware description language (HDL) (3 hours)
• Processor data path design (6 hours)
• Processor controller design (3 hours)
• Algorithms and circuits for integer arithmetic, and the arithmetic logic unit (ALU) (6 hours)
• Data representations: review of representing characters and integers (1 hour)
• Floating point representation and arithmetic (2 hours)
• Memory system design, implementing memory cells with transistors, RAM, memory hierarchy, caches (4 hours)
• I/O systems (2 hours)

Text Book and Other Required Materials: "Computer Organization and Design: The Hardware/Software Interface" by Patterson and Hennessey

Course Objectives and Their Relationship to Program Objectives:
A student should understand (i) computer organization, (ii) the principles of designing efficient computers, and (iii) the relationship between programs (software) and the hardware they run on. A student should be able to design efficient and complex digital systems, such as a RISC processor, computer arithmetic circuits, and control circuits. In addition, a student should master modern design methods for digital circuits, including appropriate computer-aided design (CAD) tools. [Program Objectives this course addresses: 1, 2, 3, 4, 5.]

Course Outcomes and Their Relationship to Program Outcomes:
The following are the course outcomes and the subset of Program Outcomes (numbered 1-11 in square braces "[ ]") they address:
• Design assembly language programs from simple programs written in a high-level language. [1,3,5]
• Design assembly language functions that can be called by programs written in a high-level
• Translate assembly language programs into machine language programs. [1,3,5]
• Design an I/O driver. [1,3,5]
• Understand the relationship between programs and the computer hardware they run on. [1]
• Understand how data is represented in computers [1]
• Understand algorithms for arithmetic and be able to design arithmetic circuits. [1,3,5]
• Design a data path for a computer. [3,5]
• Design a hardwired or micro-programmed controller for a computer. [3,5]
• Learn to use a hardware description language (HDL) to design complex circuits at the behavioral level [3,5,9,11].
• Learn to use an HDL functional simulation to verify and debug designs [9,11]
• Implement a computer using an HDL. [1,3,5,9,11]
• Understand memory hierarchy and the algorithms. [1]
• Use tools for assembly language (machine level) programming such as the SPIM simulator. [3,5,9,11]

**Contribution of Course to Meeting the Professional Component**

**Engineering Topics: 100%**

**Computer Usage:**
Students use the SPIM RISC processor simulator to run and debug assembly language programs. In addition, they design, simulate, and debug digital circuits using an HDL simulator software tool (e.g., veriwell or Xilinx Webpack Modelsim). The course also makes use of Internet services such as email, myuhportal.hawaii.edu, and the web, for references. The course has a web site, which has downloadable software and documents, as well as reference links. Approximately 30% of the assignments use computers. Note that not all programming and circuit design assignments use computers since they only require paper-and-pencil solutions.

**Design Credits and Features:**
EE 361 has 1 design credit. About 20% of the homework assignments are writing programs or fragments of programs in assembly or machine language. About another 15% are circuit design problems, which includes a 4 week task of implementing a single cycle MIPS processor using a hardware description language (HDL) such as verilog.

**Instructor(s):** T. Dobry, Y.F. Dong, L. Macchiarulo, G. Sasaki, X. Zhou.

**Person(s) Preparing Syllabus and Date:** G. Sasaki, November 25, 2008.