CMPS 375 – Computer Architecture

Last Updated: 02/07/2008

Catalog Description: Prerequisite: Computer Science 293. Hardware organization and implementation of computer architecture. Instruction set considerations and addressing modes. System control concepts. CPU control, microprogramming, I/O interface and memory organization. Parallel and data flow architecture.

Minimum Topics:
- Principle of equivalence of hardware and software
- Moore’s law
- Computer level hierarchy
- Von Neumann model
- Data representation in computer systems
- Digital logic fundamentals
- Instruction set architectures
- Memory management

Learning Objectives: Students will be able to:
- Demonstrate understanding of computing hardware technology trends
- Demonstrate understanding of how programs are executed at the machine level
- Demonstrate understanding of data representations such as unsigned integer, signed integer (signed magnitude, one’s complement, and two’s complement), floating-point (single and double precisions), and character codes (Binary-Coded Decimal, ASCII, and Unicode)
- Demonstrate understanding of computer arithmetic such as binary addition, binary subtraction, binary multiplication, and binary division
- Demonstrate understanding of error detection (Cyclic Redundancy Check) and correction (Hamming codes).
- Demonstrate familiarity with Boolean algebra (Boolean expression, Boolean identities, Boolean function), logic gates (AND, OR, NOT, and universal gates), and digital components (simple IC).
- Demonstrate ability to use Karnaugh maps to simplify a digital circuit
- Distinguish differences between combinational circuits (adder, decoder, multiplexer, shifter, ALU) and sequential circuits (Flip-Flops, registers, memory)
- Demonstrate understanding of a simple computer that includes a CPU, bus, clock, I/O system, memory organization and addressing (byte-addressable, word-addressable, alignment, high-order memory interleaving, and low-order memory interleaving), and interrupts
- Demonstrate understanding of instruction set architectures such as instruction formats (little endian versus big endian, number of operands, instruction length, opcodes), addressing mode (immediate, direct, register, indirect, register indirect, indexed addressing, based addressing and stack addressing), instruction-level pipelining
(fetch-decode-execute cycle), and real world examples (Intel, MIPS, and Java Virtual Machine)

- Demonstrate understanding of various types of memory (DRAM, SRAM, ROM), cache memory (direct mapped cache, fully associative cache, set associative cache, replacement policies, and write policies), virtual memory (paging and segmentation), and real world examples of memory management

Relevance to Program Learning Outcomes and Evaluation:

a. An ability to apply knowledge of computing and mathematics appropriate to the discipline
   Justification: data representation in computer systems
   Measured by: assignments and tests

b. An ability to analyze a problem, and identify and define the computing requirements appropriate to its solution
   Justification: digital logic; writing MARIE code
   Measured by: assignments and tests

c. An ability to design, implement and evaluate a computer-based system, process, component, or program to meet desired needs
   Justification: An introduction to a simple computer, Intel, MIPS architecture
   Measured by: assignments and tests

g. An ability to analyze the impact of computing on individuals, organizations and society, including ethical, legal, security and global policy issues
   Justification: Topic in the "Moore's Law"
   Measured by: assignments

h. Recognition of the need for, and an ability to engage in, continuing professional development
   Justification: Topic in the "Principle of Equivalence of Hardware and Software"
   Measured by: assignments

i. An ability to use current techniques, skills, and tools necessary for computing practice
   Justification: Altera's Educational package (software + hardware)
   Measured by: assignments and projects

Units Covered:
AR1 Digital logic and digital systems (6/6)
AR2 Machine level representation of data (3/3)
AR3 Assembly level machine organization (9/9)
AR4 Memory system organization and architecture (5/5)
AR5 Interfacing and communication (3/3)
AR6 Functional organization (7/7)
AR7 Multiprocessing and alternative architectures (3/3)