CS 352 – Computer Architecture

Instructor: Dr. Jack Tan
Office: Phillips 135
Office Hours: click here
Email: tanjs@uwec.edu
Tel: 836-2408

Course Description:
A study of hardware organization and architecture including performance metrics, machine data and instruction representations, computer arithmetic, assembly language programming, ALU design, CPU design and control, micro-programming, pipelining, cache memory systems, and I/O peripherals.

Credits: 3

Learning Resources:
• Computer Organization and Design MIPS Edition, Patterson and Hennessy, Morgan Kaufmann.

Evaluation
Lab & Homework: 50 %
Exams 50%

Students must pass both lab/homework assignments AND all exams to get a passing grade.

Course Content
Topic Number of weeks
• Gate and Logic Design 1.5
• Combinational Design and Circuits 1.5
• Arithmetic Design and Circuits 1
• Instructions: Language of the Computer 1
• Processor design: Control and Datapath 3
• Large and Fast: Exploiting Memory Hierarchy 1
• Storage and Other I/O Topics 1
• Multicores, Multiprocessors, and Clusters 1
• Assemblers, Linkers, and the SPIM Simulator 2
• Mapping Control to Hardware 1

Course Learning Outcomes
1. Map the high-level description of a digital system into a binary description of it (ABET Outcomes: A,M)
2. Analyze and design combinational systems using standard gates and minimization methods (such as Karnaugh maps) (ABET Outcomes: A,C,N)
3. Analyze and design simple synchronous sequential systems (ABET Outcomes: A,C,N)
4. Analyze and design sequential systems composed of standard sequential modules, such as counters and registers (ABET Outcomes: A,C,N)
5. Perform basic arithmetic operations with signed integers represented in binary (ABET Outcomes: A,M,N)
6. Use various metrics to calculate the performance of a computer system (ABET Outcomes: A, J)
7. Identify the addressing mode of instructions (ABET Outcomes: A)
8. Determine which hardware blocks and control lines are used for specific instructions (ABET Outcomes: A, J)
9. Demonstrate how to add and multiply integers and floating-point numbers using two’s complement and IEEE floating point representation (ABET Outcomes: A, J)
10. Analyze clock periods, performance, and instruction throughput of single-cycle, multi-cycle, and pipelined implementations of a simple instruction set (ABET Outcomes: A)
11. Detect pipeline hazards and identify possible solutions to those hazards (ABET Outcomes: A, J)
12. Show how cache design parameters affect cache hit rate (ABET Outcomes: A, J)
13. Map a virtual address into a physical address (ABET Outcomes: A)

**American Disability Act**
Any student who has a disability and is in need of classroom accommodations, please contact the instructor and the Services for Students with Disabilities Office in Old Library 2136 at the beginning of the semester.

**Scholastic Dishonesty**
Asking for help in understanding a problem or lending assistance to explain difficult points is encouraged. However, the copying of another student's assignment, or the common solution of written or programming assignments, or changing variable names of programming assignments, will be considered as cheating, unless group solution is specifically allowed. The purpose of assignments is to provide individual evaluation as well as a tool for learning and exploration of material. Note that the operational word in the definition of cheating is copying, not submission.

Students found guilty of academic dishonesty will be subjected to disciplinary action as prescribed by the Computer Science Department's prescribed disciplinary procedures. Disciplinary action for this course includes, but is not limited to, failure for the course.