CIS 371 Computer Organization and Design

Course Number & Title (A.1)	**CIS 371 Computer Organization and Design**
Credit Units (A.2)	1 CU (3 hours of lecture per week)
Instructor (A.3)	Milo M. K. Martin, Associate Professor, milom@cis.upenn.edu, 606 Levine Hall
Text(s)/Required Materials (A.4)	Patterson and Hennessy, "Computer Organization and Design: The Hardware Software Interface" Course lecture notes made available via the course webpage
Catalog Description (A.5a)	This is the second computer organization course and focuses on computer hardware design. Topics covered are: (1) basic digital system design including finite state machines, (2) instruction set design and simple RISC assembly programming, (3) quantitative evaluation of computer performance, (4) circuits for integer and floating-point arithmetic, (5) datapath and control, (6) micro-programming, (7) pipelining, (8) storage hierarchy and virtual memory, (9) input/output, (10) different forms of parallelism including instruction level parallelism, data-level parallelism using both vectors and message-passing multi-processors, and thread-level parallelism using shared memory multiprocessors. Basic cache coherence and synchronization.
Prerequisites (A.5b)	CIS 240, knowledge of at least one programming language
Course Satisfies (A.5c)	[ ] Math [ ] Science [**x**] Engineering [ ] Technical Elective [ ] TBS **(check only one, UG curric impact only)** **Required**
Course Web	http://www.cis.upenn.edu/~cis371/
Course Outcomes (A.6a)	• Demonstrate understanding of processor design foundations, including: instruction set architecture design, fast arithmetic, pipelining, prediction, caching, scheduling, virtual memory, and explicit and implicit parallelism. • Demonstrate ability to analyze performance impacts of architectural features (branch prediction, load-to-use penalty) • Demonstrate understanding of basic hardware design tradeoffs: performance, area/cost, and methods for evaluating these trade-offs. • Demonstrate the ability to design (including a design document and testing plan) and implement a pipelined processor in HDL • Demonstrate the ability to work as a team to design and implement a pipelined processor • Demonstrate the ability to perform modular design decomposition & component testing • Demonstrate the ability to use Verilog HDL (hardware description language) and ability to test the design using FPGAs (field-programmable gate arrays). • Demonstrate the ability to implement a processor in HDL based on a reference design • Demonstrate the ability to use hardware development and debugging environments.
Contribution towards Program Outcomes (A.6b)	CSCI: a, b, c, d, i, j CMPE: a, b, c, d ,e, k
Topics Covered (A.7)	• Basic digital system design including finite state machines • Instruction set design and simple RISC assembly programming • Quantitative evaluation of computer performance • Circuits for integer arithmetic • Datapath and control • Micro-programming • Pipelining • Storage hierarchy and virtual memory • Input/output • Different forms of parallelism (instruction-level, data-level parallelism, and thread-level parallelism) • Basic cache coherence and synchronization
Grading Details	15% Homework 30% Labs 20% Midterm 35% Final
Prepared By/Date	Milo M. K. Martin, Revised May 2011