COE 205 Computer Organization & Assembly Language

Course Learning Outcomes Table

Course Learning Outcomes	Outcome Indicators and Details	Assessment Methods and Metrics	ABET 2000 Criteria
1. Ability to analyze, design, implement, and test assembly language programs.	• Instruction Set Architecture • Number (unsigned and signed) and character representation • Addressing modes • Syntax, semantics, and effect on flags of Pentium instructions. • Input/output. • Arithmetic and logic operations. • Flow-control structures. • Procedures. • Macros. • String manipulation. • Interrupt mechanism. • Implementation of Pseudo code algorithms in assembly language.	• Assignments • Quizzes • Exams • Project	C(H)
2. Ability to use tools and skills in analyzing and debugging assembly language programs.	• Assembly language vs. machine language. • Assembling and linking assembly programs (including use of multiple files). • Use of debugger to analyze and debug programs. • Use of libraries.	• Lab work	K(L)

3. Ability to design the datapath and control unit of a simple CPU.	• Fetch-execute cycle • Data, address and control busses • Register transfer • Data path design: 1-bus, 2-bus and 3-bus CPU. • Derivation of control steps for assembly instructions. • Hardwired Control unit design • Microprogrammed control unit design. • Fixed vs. variable instruction format.	• Assignments • Quizzes • Exams	C(M)
4. Ability to demonstrate self-learning capability.	• Ability to learn a course topic alone (e.g. Macros) • Course Project may involve topics not studied in the course	• Assignments • Quizzes	I(L)
5. Ability to work in a team.	• Project is divided into separate parts that will be integrated for project completion.	• Project	D(L)

Course Learning Outcomes Evaluation Table

Outcome	Outcome Min. Weight	Assessment Method							
		Assignments	**Quizzes**	**Exam I**	**Exam II**	**Final Exam**	**Lab Work**	**Project**	**Total**
O1	**55%**	**15%**	**8%**	**15%**	**20%**		**5%**	**8%**	**71%** **60%**
	Average			9.5%	12.1%		4.1%	7%	50.1%(70.6%)
	Evidence	#1-4	#1-4, 6	Q1-5	Q1-5		#1-13	Report	
O2	**4%**						**5%**		**5%** **80%**
	Average						4.1%		4.1% (82%)
	Evidence						#1-13		
O3	**15%**					**20%**			**20%** **75%**
	Average					14.2%			11.8% (59%)
	Evidence					Final Ex			
O4	**2%**		**2%**						**2%** **60%**
	Average		1.3%						1.3% (65%)
	Evidence		#5						
O5	**2%**							**2%**	**2%** **55%**
	Average							1%	1% (50%)
	Evidence							Report	
Weight		**15%**	**10%**	**15%**	**20%**	**20%**	**10%**	**10%**	**100%**
Average		**12.1%**	**6.6%**	**9.5%**	**12.1%**	**11.8%**	**8.2%**	**8%**	**68.3%**

Outcome Evaluation Example:

Class average for an outcome is computed by adding the average weights obtained from each assessment method used for the outcome divided by the total outcome weight. An example is shown below:

Class Average of Outcome 1 = [Assignments (12.1) + Quizzes (5.3) + Exam I (9.5) + Exam II (12.1) + Lab Work (4.1) + Project (7)] / 71 * 100 = 50.1/71*100 = 70.6.

Instructor Comments and Feedback:

Instructors need to comment here about outcomes that they think were not achieved or there is a need for improvement. The instructor needs to suggest ways for improving outcome achievement in next course offerings.

_____

_____

_____

_____

_____

_____

_____

COE 308 Computer Architecture
Course Learning Outcomes Table

Course Learning Outcomes	Outcome Indicators and Details	Assessment Methods and Metrics	Min. Weight	ABET 2000 Criteria
O1. Ability to apply knowledge of mathematics, probability, and statistics in computer analysis and design.	• Integer representation, addition, and multiplication • Floating-point representation, rounding, normalization, addition, and multiplication. • Program and instruction execution times and stall cycles. • Speedup computation • Evaluation of the average performance of I-pipelining and memory system	• Assignments • Quizzes • Exams	40%	A(H)
O2. Ability to design the datapath and control of a processor.	• Design generic datapath based on Instruction Set requirements. • Identify datapath components and clocking methodology. • Design a detailed single-cycle integer datapath, Muxes, and PC updating. • Identify control signals and design control logic • Design inter-stage buffers and clocking for multicycle datapath. • Design multi-cycle control states and logic. • Design pipelined datapath and control • Detect and eliminate structural hazards • Detect data hazards and implement forwarding • Handle control hazards and predict branches	• Assignments • Quizzes • Exams • Project	20%	C (H)

O3. Ability to identify, formulate, and solve computer architecture problems.	• Assess design methodologies in single-cycle, multi-cycle, and multiple-issue datapaths. • Assess tradeoffs in cache design, page size, bus width, degree of associativity, cache capacity, and main memory access time. • Assess tradeoffs in address translation, virtual page size, TLB size, sequential versus concurrent TLB and cache access. • Assess scalability issues in shared-memory and distributed-memory systems.	• Assignments • Quizzes • Exams	5%	E (L)
O4. Ability to use simulator tools.	• Ability to set up a simulator. • Set up simulation runs based on some design specifications. • Run simulations and collect results and statistics • Ability to analyze simulation results and modify design specifications to improve performance.	• Assignments • Mini-group Projects	2%	K(L)
O5. Ability to engage in self-learning.	• Demonstrates reading, writing, listening and speaking skills • Identifying, retrieving, and organizing information • Following a learning plan • Demonstrate critical thinking skills such as applying the facts, formulas, theories, etc. to everyday situations.	• Mini-group Projects • Assignments • Quizzes • Exams	2%	I(L)