The University of Georgia
Department of Computer Science

Course Information Sheet
CSCI 4720
Computer Architecture and Organization

Brief Course Description (50-words or less)	Design and analysis of the structure and function of modern computing systems. Topics studied include combinational and sequential logic, number systems and computer arithmetic, hardware design and organization of CPU, I/O systems and memory systems, instruction set and assembly language design, performance characterization and measurement, and current trends and developments in computer architecture and organization.
Extended Course Description / Comments	A hierarchical and holistic view of computer organization and architecture is presented - from the logic design level right up to the virtual machine level. Tradeoffs associated with design choices at each level of abstraction are identified and quantified. Tradeoff parameters such as performance (speed), hardware complexity (cost), memory footprint and power consumption are analyzed in juxtaposition. The impact of the design of the instruction set architecture on performance and complexity of compiler design, impact of various organizational features on the operating systems overhead, relationship between locality and latency in the context of hierarchical memory design and the impact of the design of the instruction set on locality are quantified and analyzed.
Pre-Requisites and/or Co-Requisites	CSCI 2670: *Introduction to Theory of Computation* or CSEE 2220: *Fundamentals of Logic Design* CSCI 1730: *Systems Programming*
Required, Elective or Selected Elective	Selected Elective Course
Approved Textbooks (if more than one listed, the textbook used is up to the instructor's discretion)	Author(s): David A. Patterson, John L. Hennessy Title: *Computer Organization and Design: The Hardware/Software Interface* Edition: Fifth Edition, Elsevier, 2014, ISBN: 978-0-12-407726-3.

Specific Learning Outcomes (Performance Indicators)

The overall goal of the course is to enable students to analyze and design the structure and function of various components of modern computing systems. By the end of the semester, all students will be able to do the following:

1. Design a combinational logic circuit using logic gates and programmable logic arrays (PLAs) given a functional description.
2. Design and functional analysis of common combinational logic circuits such as adders, decoders, encoders, multiplexors, demultiplexors and switches/routers.
3. Design a sequential logic circuit using flip flops and combinational logic given a functional description of a finite state automaton.
4. Design and perform functional analysis of common sequential logic circuits such as sequence detectors and counters.
5. Design memory elements such as registers and RAM using flip flops.
6. Design hierarchical memory using register files, caches and RAM

modules.

7. Analyze the performance of computer systems in terms of commonly used metrics such as CPU execution time, MIPS, MFLOPS, power consumption and reliability and the speedup resulting from system optimization using Amdahl's law.
8. Analyze the tradeoffs in Instruction Set Architecture design using the MIPS assembly language as an example.
9. Design and analyze algorithms for fixed-point and floating-point binary arithmetic.
10. Design and analyze the single cycle and pipelined datapath and CPU control for a subset of the MIPS assembly language.
11. Discuss multicore computing architectures such as Graphical Processing Unit (GPU) and their applications

ABET Learning Outcomes

a. Analyze a complex computing problem and to apply principles of computing and other relevant disciplines to identify solutions.
b. Design, implement, and evaluate a computing-based solution to meet a given set of computing requirements in the context of the program's discipline.
c. Communicate effectively in a variety of professional contexts.
d. Recognize professional responsibilities and make informed judgments in computing practice based on legal and ethical principles.
e. Function effectively as a member or leader of a team engaged in activities appropriate to the program's discipline.
f. Apply computer science theory and software development fundamentals to produce computing-based solutions.

Relationship Between Student Outcomes and Learning Outcomes

	ABET Learning Outcomes						
Specific Learning Outcomes		a	b	c	d	e	f
	1	●	●				
	2	●	●				
	3	●	●				
	4	●	●				
	5	●	●				●
	6	●	●				●
	7	●	●		●		●
	8	●	●				●
	9	●	●				●
	10	●	●				●
	11	●	●				●

Major Topics Covered (Approximate Course Hours) 3 credit hours = 37.5 contact hours 4 credit hours = 50 contact hours Note: Exams count as a major topic covered	Combinational Logic Design(4 hours) Sequential Logic Design (4 hours) Hierarchical Memory Design (6 hours) Performance Analysis of Computer Systems (4 hours) Test 1 (1.5 hours) Binary Computer Arithmetic (4 hours) Instruction Set Architecture Design (8 hours) MIPS Assembly Language Programming using SPIM/XSPIM (4 hours) Test 2 (1.5 hours) CPU Datapath Design (4 hours) CPU Controller Design (4 hours) Emerging Trends in Computer Systems Design (2 hours) Final Exam (3 hours)
Course Master	Dr. Suchendra Bhandarkar