Knight Foundation School of Computing and Information Sciences

Course Title: Computer Architecture **Date:** 9/23/2019

Course Number: CDA 3102

Number of Credits: 3

Subject Area: Computer Organization	**Subject Area Coordinator:** Dong Chen **email:** dochen@cs.fiu.edu
Catalog Description: Covers the levels of organization in a computer: digital logic; machine and assembly language programming, design of memory, buses, ALU, CPU; virtual memory, I/O	
Textbook: Computer Organization and Design: The Hardware/Software Interface David A. Patterson, John L. Hennessy Morgan Kaufmann	
References: Digital Design and Computer Architecture David Money Harris and Sarah L. Harris Morgan Kaufmann	
Prerequisites Courses: COP 3337 and COT 3100 or MAD 2104	
Corequisites Courses: None	

Type: Required for CS Major

Prerequisites Topics:

- High level programming language constructs
- Function call/return
- Parameters of a function(method)
- Boolean algebra
- Fundamental data structures

Knight Foundation School of Computing and Information Sciences
CDA 3102
Computer Architecture

Course Outcomes:

1. Master the data path of a simple von Neumann architecture and its relation to the instruction execution cycle
2. Master simple machine and assembly language programming
3. Master the implementation of high-level language constructs in lower levels: selection, iteration, function call/return
4. Be familiar with interrupts and traps
5. Master the design of combinational and sequential circuits
6. Master the design of memory and the ALU.
7. Master control unit design
8. Be familiar with cache architectures, branch predictions, scheduling of multiple instruction issue and flow control.

Relationship between Course Outcomes and Program Outcomes

BS in CS: Program Outcomes	Course Outcomes
a) Demonstrate proficiency in the foundation areas of Computer Science including mathematics, discrete structures, logic and the theory of algorithms	1, 5
b) Demonstrate proficiency in various areas of Computer Science including data structures and algorithms, concepts of programming languages and computer systems.	1, 2, 3, 5, 6, 7, 8
c) Demonstrate proficiency in problem solving and application of software engineering techniques	5
d) Demonstrate mastery of at least one modern programming language and proficiency in at least one other.	
e) Demonstrate understanding of the social and ethical concerns of the practicing computer scientist.	
f) Demonstrate the ability to work cooperatively in teams.	
g) Demonstrate effective communication skills.	

Knight Foundation School of Computing and Information Sciences
CDA 3102
Computer Architecture

Assessment Plan for the Course & how Data in the Course are used to assess Program Outcomes

Student and Instructor Course Outcome Surveys are administered at the conclusion of each offering, and are evaluated as described in the School's Assessment Plan: https://abet.cs.fiu.edu/csassessment/

Outline

Topic	Number of Lecture Hours	Outcome
• Digital logic: Design of o Data representation o Fundamental building blocks (logic gates, combinational circuits) o Von Neumann model o Instruction execution cycle o Multiplexer, demultiplexer, encoder, decoder o Arithmetic Logic Unit, Shifter o Latch, flip-flop, register, memory organization o Clocks, counters o Bus protocols, arbitration, DMA o Data path, control unit o Microprogram	14	1,5,6,7
• Assembly level machine organization o Instruction sets and types o Assembly language programming o Addressing modes o Subroutines and system routines o I/O and interrupts o Bit level manipulation o Assembly process and linking	14	2,3,4
• Performance enhancement o Interpretation and translation o Simple machine architecture o Instruction prefetch o Pipelining, pipeline hazards o Cache architecture o Branch prediction o Dynamic scheduling of instructions o Speculative execution	11	8

Course Outcomes Emphasized in Laboratory Projects / Assignments

	Outcome	Number of Weeks
1	Digital circuit design Outcomes: 5	2
2	Machine and assembly language programming Outcomes: 2,3	3
3	Control unit and CPU design Outcomes: 1,4,7	4
4	Memory Outcomes: 6,8	2
5	Pipelining Outcomes: 8	2

Oral and Written Communication

No significant coverage

Written Reports		Oral Presentations	
Number Required	Approx. Number of pages	Number Required	Approx. Time for each
0	0	0	0

Social and Ethical Implications of Computing Topics

No significant coverage

Topic	Class time	student performance measures

Approximate number of credit hours devoted to fundamental CS topics

Fundamental CS Area	Core Hours	Advanced Hours
Algorithms:		
Software Design:		
Computer Organization and Architecture:	**2.5**	
Data Structures:		
Concepts of Programming Languages	**0.5**	

Theoretical Contents

Topic	Class time

Problem Analysis Experiences

1. Instruction set analysis, Implementation of high level programming language constructs in low level languages

Solution Design Experiences

1. Digital circuit design
2. Assembly language programming
3. Microprogram design

The Coverage of Knowledge Units within Computer Science Body of Knowledge[1]

Knowledge Unit	Topic	Lecture Hours
PL2	Virtual machine, hierarchy of virtual machines, intermediate languages	6
AR1	History of computer architecture, fundamental logic circuits, gate delays	6
AR2	Bits, bytes, and words, numeric data representation, fixed- and floating-point systems, signed and twos-complement representations, nonnumeric data (character codes, graphical data), representation of records and arrays	2
AR3	von Neumann machine, control unit; instruction fetch, decode, and execution, instruction sets and types (data manipulation, control, I/O), assembly/machine language programming, instruction formats, addressing modes, subroutine call and return mechanisms, I/O and interrupts	12
AR4	Storage systems, coding, data integrity, memory organization, latency, cycle time, cache memories	4
AR5	I/O fundamentals, external storage, RAID architectures, bus protocols, bus arbitration, DMA	2
AR6	Implementation of simple datapath, control unit, pipelining, instruction level parallelism	3
AR7	SIMD, MIMD, VLIW, interconnection networks, shared memory systems, cache coherence	2
AR8	Superscalar, superpipelining, branch prediction, prefetching, speculative execution, multiple instruction issue	2

[1]See https://www.acm.org/binaries/content/assets/education/cs2013_web_final.pdf for a description of Computer Science Knowledge units