Knight Foundation School of Computing and Information Sciences

Course Title: Computer Architecture **Date:** 01/15/2026

Course Number: CDA 3102

Number of Credits: 3

Subject Area: Computer Organization
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. Prerequisites: COT 3100 or MAD 2104. Corequisite: COP 3337
Textbook: Computer Organization and Design MIPS Edition: The Hardware/Software Interface, David A. Patterson and John L. Hennessy, 6th Edition, Morgan Kaufmann, Dec. 2020, ISBN: 978-0128201091
References: Digital Design and Computer Architecture, David Money Harris and Sarah L. Harris, Morgan Kaufmann, Oct. 2021, ISBN: 978-0128200643
Prerequisites Courses: COT 3100 or MAD 2104
Corequisites Courses: COP 3337

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
- Number conversion among binary, octal, decimal, and hex. systems

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

Course Outcomes:

1. **Explain** the instruction execution cycle of a simple von Neumann architecture by traversing the data path of various machine instructions [Understand]
2. **Develop** simple assembly language programs to perform basic computations [Apply]
3. **Apply** high-level programming constructs such as selection, iteration, and function call/return in low-level assembly representations [Apply]
4. **Describe** the purpose of interrupts and traps including their behaviors. [Understand]
5. **Design** digitals logic circuits including combinational and sequential circuits [Create]
6. **Design** system components including memory and arithmetic/logic units (ALUs) [Create]
7. **Design** a control unit for a simple processor [Create]
8. **Explain** how modern processor performance is enhanced through cache organization, instruction-level parallelism mechanisms, and control-flow optimization techniques (e.g. branch prediction and multi-instruction scheduling). [Understand]
9. **Use** AI-assisted tools to deepen understanding of instruction-level behavior (e.g. exploring, translating and debugging assembly language programs). [Apply]

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 & CPU Building Blocks o Logic gates & combinational circuits o Von Neumann model o Instruction execution cycle o Multiplexers, decoders, encoders o ALU and shifter o Latches, flip-flops, registers o Memory organization o Clocks & counters o Bus protocols, DMA o Datapath & control signals o AI-assisted visualization and verification of datapath/control behavior	14	1, 5, 6, 7, 9
• Assembly-level machine organization o Assembly language programming o Addressing modes o Subroutines & stack behavior o System routines o Interrupts & traps o Bit-level operations o Assembly process & linking o AI guided code translation, debugging, & register-tracking	14	2, 3, 4, 9
• Performance enhancement & Memory Systems o Simple machine architecture o Instruction prefetch o Pipelining & hazards o Different cache architecture types o Branch prediction o Dynamic scheduling o Speculative execution o AI-assisted analysis of pipeline behavior and memory access patterns	9.5	8, 9

Tentative Weekly Schedule

Week	Topics	Course Outcome(s)
1	Introduction to Computer Architecture Levels of Computer Organization Von Neumann Model	1
2	Logic Gates and Combinational Circuits Multiplexers, Decoders, Encoders	5
3	Instruction Execution Cycle Datapath Overview Control Signals	1, 7
4	ALU and Shifter Design Registers and Register Files	6
5	Sequential Logic Latches, Flip-Flops, and Registers Clocks and Counters	5
6	Memory Organization Buses, Bus Protocols, and DMA	6
7	Datapath & Control Integration AI-Assisted Visualization of Datapath Behavior	1, 7, 9
8	Assembly-Level Machine Organization Instruction Formats and Addressing Modes	2
9	Assembly Language Programming Selection and Iteration in Assembly	2, 3
10	Subroutines and Stack Behavior Function Call / Return Mechanisms	3
11	Interrupts and Traps System Routines and Bit-Level Operations	4
12	Assembly Process and Linking AI-Guided Translation and Debugging of Assembly Code	2, 3, 9
13	Processor Performance Fundamentals Pipelining and Pipeline Hazards	8
14	Cache Organization Memory Hierarchy and Cache Architectures	8
15	Advanced Performance Techniques Branch Prediction, Dynamic Scheduling, Speculative Execution AI-Assisted Analysis of Pipeline & Memory Behavior	8, 9
16	Final Exam	1–9

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

Component	%
Exams	70
Projects	20
Homework	10

Proposed Letter Grade Scale

Letter Grade	Percentage Range	Letter Grade	Percentage Range
A	95 or above	C+	70 – 74.99
A-	90 – 94.99	C	65 – 69.99
B+	85 – 89.99	D	60 – 64.99
B	80 – 84.99	F	59.99or below
B-	75 – 79.99		

Sample Assignment

Part 1 — Write a Short MIPS Program

Write a MIPS program that:

1. Reads an integer n.
2. Computes the sum of all even numbers from 0 to n using:
    - A loop
    - A conditional branch
    - A procedure call with stack discipline
    - A bitwise operation to test evenness

Your compute_even_sum procedure must return the sum in $v0.

Part 2 — AI-Assisted Review & Debugging

Using a generative AI tool of your choice:

A. Translation Check

- Ask the AI to explain your entire program in plain English.
- Identify two inaccuracies or omissions in the AI's interpretation and provide corrected explanations (2–4 sentences each).

B. Debugging Assistance

- Introduce one small bug (e.g., register misuse, wrong branch condition).
- Ask the AI to help debug it.
- Report:
    - The bug you inserted
    - The AI's suggestion
    - Whether the suggestion was correct
    - Your corrected code (3–5 lines max)

C. Responsible Use Reflection

Write 5–7 sentences describing:

- How you verified AI output
- Why assembly generated by AI must be checked
- How AI helped you better understand register flow, control signals, or instruction behavior

Part 3 — Instruction Behavior (Datapath & Control Signals)

Select two instructions from your program (e.g., addi, and, lw, beq, jal) and for each:

1. Provide a brief datapath description (3–4 sentences) identifying source registers, ALU role, memory access (if any), and result path.
2. List the key control signals that would be active (e.g., RegDst, ALUSrc, MemRead, Branch, RegWrite, PCSrc).

(Diagrams optional but not required.)

Submission

Submit one PDF/DOCX containing:

- Program code
- AI explanation comparison
- Debugging documentation
- Responsible-use reflection
- Datapath + control signal analysis

Sample Rubric (100 points total)

1. MIPS Program Correctness — 30 pts

Grading Item	Complete (Full Credit)	Partial Credit	No Credit
Program functionality (even-sum logic, loop, branch, bitwise) – 12 pts	Program works correctly and meets all requirements	Minor logic or output errors	Major errors; requirements not met
Procedure call & stack usage – 8 pts	Correct prologue/epilogue; proper use of `$sp`, `$ra`	Small mistakes but mostly correct	Missing or incorrect usage
Code clarity & comments – 10 pts	Clear structure, meaningful comments	Some comments missing or unclear	Poor readability; no comments

2. AI-Assisted Analysis (CLO 9) — 30 pts

Grading Item	Complete (Full Credit)	Partial Credit	No Credit
AI explanation comparison – 10 pts	Identifies ≥2 inaccuracies and provides correct clarifications	Identifies only one or gives limited corrections	No comparison or incorrect analysis
Debugging using AI – 10 pts	Bug described; AI suggestion evaluated accurately; corrected	Some details missing	No debugging analysis

Grading Item	Complete (Full Credit)	Partial Credit	No Credit
	code provided		
Responsible-use reflection – 10 pts	5–7 thoughtful sentences on validation, AI limitations, and learning impact	Superficial or <5 sentences	Missing or irrelevant response

3. Instruction Behavior (Datapath & Control Signals) — 25 pts

Grading Item	Complete (Full Credit)	Partial Credit	No Credit
Datapath description for two instructions – 12 pts	Clear explanation of register flow, ALU role, and memory interaction	Some technical inaccuracies	Missing or incorrect
Control-signal listing – 8 pts	Correct key control signals for both instructions	Partially correct	Missing or wrong
Instruction cycle connection – 5 pts	Clear link to fetch/decode/execute behavior	Partial explanation	No explanation

4. Professionalism & Completeness — 15 pts

Grading Item	Complete (Full Credit)	Partial Credit	No Credit
Organization and clarity of report – 10 pts	Well-structured, readable submission	Minor formatting issues	Difficult to read or incomplete
Evidence of AI interaction – 5 pts	Interaction included (text or screenshots)	Partially included	No evidence provided