SENG 11232 · Engineering Foundation
STUDENT PRE-LECTURE NOTES · LECTURE 06

Computer Architecture & the Execution Model

Module 2 — Systems & Digital Foundations

HOW TO USE Read these notes BEFORE the lecture. Highlight terms you don't recognise. Bring your questions to class. Use the blank spaces for your own notes.

Learning Objectives

By the end of this lecture, you should be able to:

- 1. Describe the Von Neumann architecture and its five components
- 2. Trace the fetch-decode-execute cycle for a simple instruction
- 3. Explain the memory hierarchy and its performance implications
- 4. Explain how source code becomes executable machine instructions

Von Neumann Architecture

Five Components

- CPU: performs computation (ALU + Control Unit + Registers)
- Memory (RAM): stores both instructions and data
- Input: keyboard, sensors, network — data enters system
- Output: display, speakers, network — results leave system
- System Bus: address bus + data bus + control bus

KEY Von Neumann bottleneck: CPU and memory share the same bus. Memory access speed limits overall system performance.

Fetch-Decode-Execute Cycle

- FETCH: read instruction at address in PC from RAM into IR; increment PC
- DECODE: CU interprets opcode and identifies operands
- EXECUTE: ALU performs operation; result stored; flags updated

Your notes:

_____

_____

_____

Memory Hierarchy & Compilation

Memory Hierarchy (fastest to slowest)

- Registers: < 1 ns, < 1 KB — inside CPU — hold current operands

- L1 Cache: ~ 1 ns, 32-64 KB per core
- L2 Cache: ~ 5 ns, 256 KB - 1 MB
- L3 Cache: ~ 20 ns, 4-64 MB shared
- RAM: ~ 60-100 ns, 8-64 GB
- SSD: ~ 100 µs | HDD: ~ 10 ms

NOTE	Cache-friendly code accesses data sequentially (arrays). Random pointer chasing (linked lists) causes cache misses — major performance impact.

From Source to Execution

- Compiled (C, C++, Go): entire program translated to machine code before execution
- Interpreted (Python, Ruby): translated and executed line by line at runtime
- JIT compiled (Java, C#, JS): bytecode compiled to machine code at runtime

Your notes:

_____

_____

_____

Key Terms for This Lecture

Term	Your Definition (fill in during/after lecture)
Von Neumann Architecture	
CPU	
ALU	
Control Unit	
Program Counter	
Instruction Register	
Cache	
RAM	
Fetch-Decode-Execute	
JIT Compilation	
System Call	
Operating System	

Self-Check Questions

Answer these after the lecture to test your understanding:

- Q1. Draw and label the five components of the Von Neumann architecture.

Answer: _____

- Q2. Trace the fetch-decode-execute cycle for an ADD R1, R2 instruction.

Answer: _____

- Q3. Why is data accessed sequentially (in arrays) faster than data accessed randomly (linked lists)?

Answer: _____

- Q4. What is the difference between a compiled and an interpreted language? Name one advantage of each.

Answer: _____
