

---

# Chapter 1 - Introduction

# Chapter 1 Objectives

---

- ❑ Know the concentration of computer organization and computer architecture
- ❑ Understand **units** of measure common to computer systems
- ❑ Understand the computer as a **layered & structured** system
- ❑ Be able to explain the **von Neumann/Princeton** architecture and the function of basic computer components.

# Computer Organization and Architecture

---

## □ Computer organization

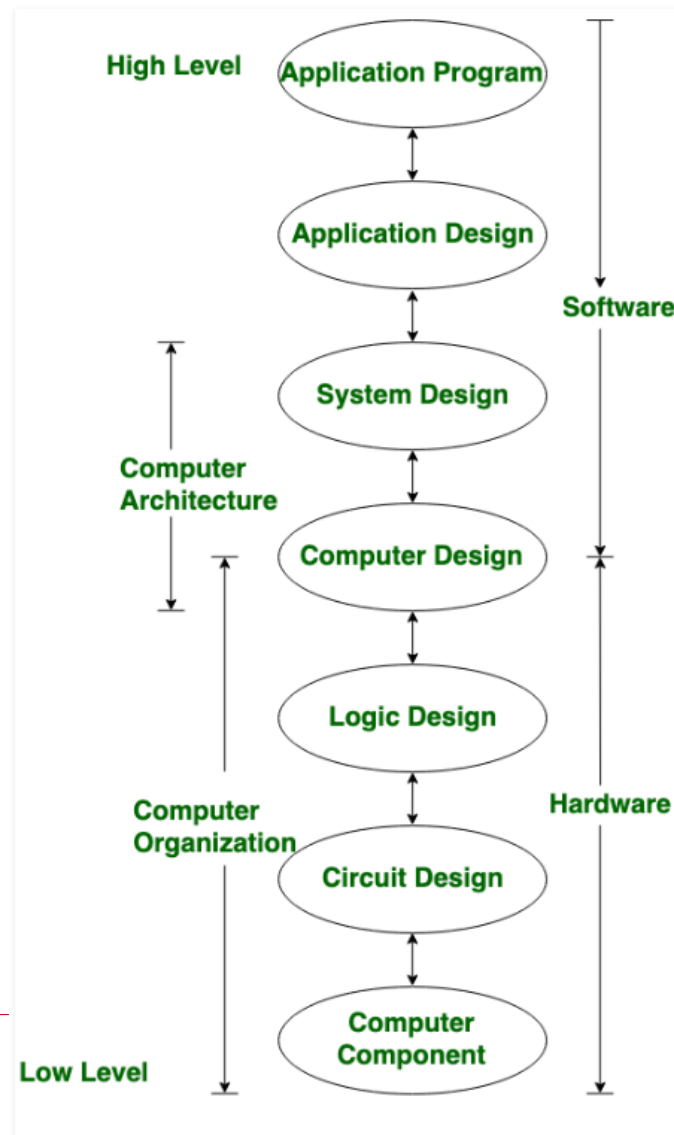
- It focuses on the working **mechanism** of all **physical** aspects of computer systems
  - e.g., circuit design, control signals, memory types, etc.
- Try to answer the question: How does a computer work?

## □ Computer architecture

- It focuses on the **structure** and **behavior** of the computer systems. It affects the logical execution of programs.
  - e.g., instruction sets, instruction formats, data types, addressing modes, etc.
- Try to answer the question: How do I design a computer?

# Computer Organization and Architecture

---



# Chapter 1 Objectives

---

- ❑ Know the **concentration of** computer organization and computer architecture
- ❑ Understand **units** of measure common to computer systems
- ❑ Understand the computer as a **layered** system
- ❑ Be able to explain the **von Neumann** architecture and the function of basic computer components.

# Consider This Advertisement



	Acer Aspire S3-951	MacBook Air (13.3 inch)
Price	\$899.00	\$1,299.00
Screen Size	13.3 inches, LED backlit	13.3 inches, LED backlit
Screen Resolution	1366-by-768	1440-by-900
Processor	1.66 GHz Intel Core i5	1.7 GHz Intel Core i5
Graphics	Intel HD Graphics 3000	Intel HD Graphics 3000
Storage	20GB SSD (for OS), 320 GB HDD	128GB flash storage
Memory	4GB, 1066 MHz DDR3	4GB, 1333 MHz DDR3
Battery Life (promised)	up to 6 hours, 50 days standby	up to 7 hours, 30 days standby
Connectivity	802.11 b/g/n Wi-Fi; Bluetooth 4.0	802.11 a/b/g/n Wi-Fi; Bluetooth 4.0
Webcam	1.3 Megapixel	FaceTime camera (MP undisclosed)
Ports	USB 2.0 (2); HDMI (1); SD Card reader (1)	USB 2.0 (2); Thunderbolt (1); SD Card Read
Operating System	Windows 7 Home Premium	Mac OS X 10.7, Lion
Base Weight	2.98 pounds	2.96 pounds
Height	0.68 inch	0.68 inch

L1 Cache??

GHz??

GB??

PCI??

USB??

*What does it all mean??*

# The Measures of Speed and Capacity

---

- Kilo- (K) = 1 thousand =  $10^3$  and  $2^{10}$
- Mega- (M) = 1 million =  $10^6$  and  $2^{20}$
- Giga- (G) = 1 billion =  $10^9$  and  $2^{30}$
- Tera- (T) = 1 trillion =  $10^{12}$  and  $2^{40}$
- Peta- (P) = 1 quadrillion =  $10^{15}$  and  $2^{50}$
- Exa- (E) = 1 quintillion =  $10^{18}$  and  $2^{60}$
- Zetta- (Z) = 1 sextillion =  $10^{21}$  and  $2^{70}$
- Yotta- (Y) = 1 septillion =  $10^{24}$  and  $2^{80}$

Where a metric refers to a power of ten or a power of two **typically** depends upon what is being measured.

# Measures of Speed and Capacity (`cont.)

---

- Byte = a unit of storage
  - $1\text{KB} = 2^{10} = 1024 \text{ Bytes}$
  - $1\text{MB} = 2^{20} = 1,048,576 \text{ Bytes}$
  - 1G
- So,
  - $1\text{KB} \neq 1000 \text{ Bytes}$
  - $1\text{KB} = 1024 \text{ Bytes}$

# Measures of Speed and Capacity (`cont.)

---

## □ Hertz

- It is the unit of frequency and is defined as **one cycle per second**
  - So, time is the reciprocal of frequency
- $1\text{MHz} = 1,000,000\text{Hz} \rightarrow 1\text{M cycle per second}$
- Processor speeds are measured in MHz or GHz

# Measures of Speed and Capacity (`cont.)

---

- A CPU operates at 133MHz
  - What's the duration of one cycle (in sec.)?

$$\begin{aligned} & 1 / (133,000,000 \text{ cycles/second}) \\ & = 7.52\text{ns/cycle} \end{aligned}$$

# Measures of Time and Space

---

- Milli- (m) = 1 thousandth =  $10^{-3}$
- Micro- ( $\mu$ ) = 1 millionth =  $10^{-6}$
- Nano- (n) = 1 billionth =  $10^{-9}$
- Pico- (p) = 1 trillionth =  $10^{-12}$
- Femto- (f) = 1 quadrillionth =  $10^{-15}$
- Atto- (a) = 1 quintillionth =  $10^{-18}$
- Zepto- (z) = 1 sextillionth =  $10^{-21}$
- Yocto- (y) = 1 septillionth =  $10^{-24}$

# Measures of Time and Space: Examples

---

- Millisecond = 1 thousandth of a second
  - Hard disk drive access times are often 10 to 20 milliseconds.
- Nanosecond = 1 billionth of a second
  - Main memory access times are often 50 to 70 nanoseconds.
- Micron (micrometer) = 1 millionth of a meter
  - Circuits on computer chips are measured in microns.

# Chapter 1 Objectives

---

- ❑ Know the **concentration of** computer organization and computer architecture
- ❑ Understand **units** of measure common to computer systems
- ❑ Understand the computer as a **layered** system
- ❑ Be able to explain the **von Neumann** architecture and the function of basic computer components.

# The Computer Level Hierarchy

---

- Computer system consists of hardware and software.
- Build complex systems requires a “*divide and conquer*” strategy
  - Each program module solves a smaller problem.
- Complex computer systems employ a similar technique through a series of *virtual machine* layers.

# The Concept of Virtual Machine

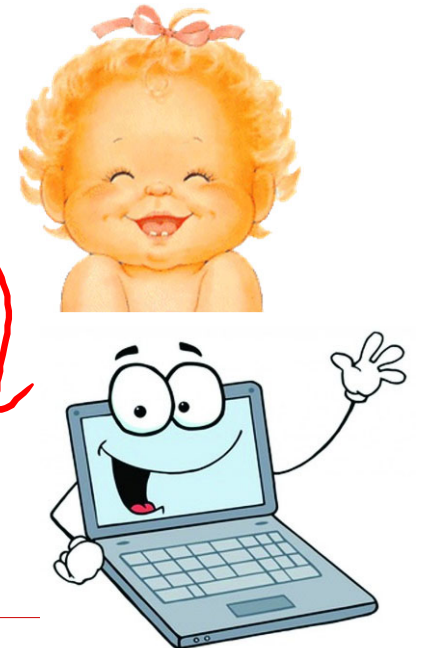
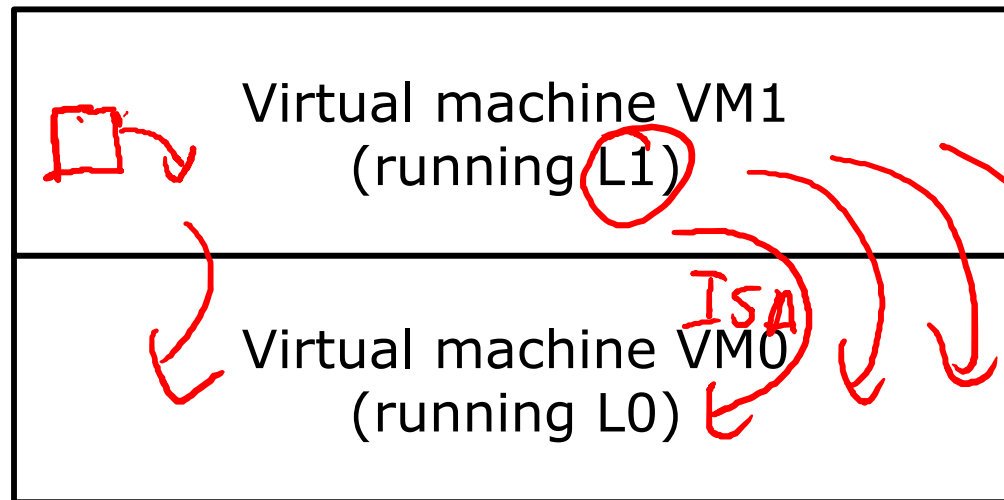
---

- Let's assume that a computer can execute programs written in its native **machine language**
  - Each instruction of this language can be executed by using some electric circuit ---We call this language as **L0**
- However, if we can design a relative simpler (human-friendly) language, namely **L1**, we would like to use it
  - If we can write an easier program in **L1**, and the program can be transformed into **L0**, we are done!

# The Concept of Virtual Machine

---

- ❑ We call the virtual machine that runs L0 as **VM0**, whereas the virtual machine that runs L1 as **VM1**
- ❑ The structure of VM0 and VM1 is the following:



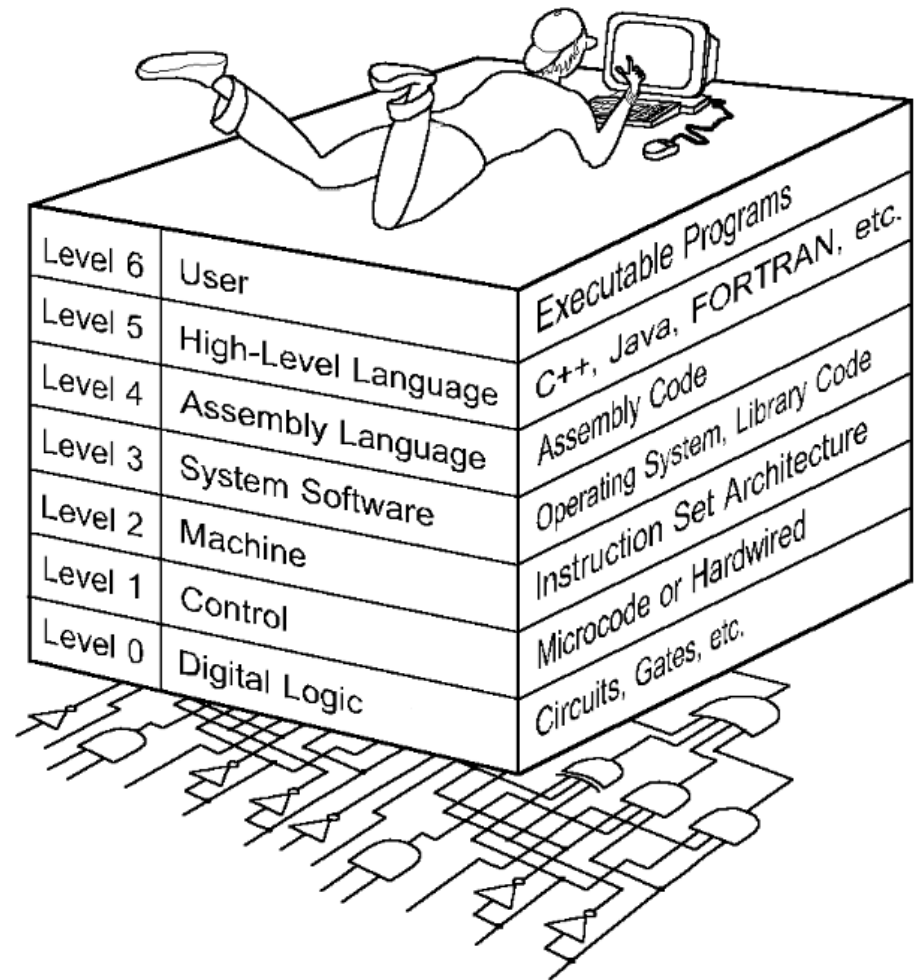
# The Concept of Virtual Machine

---

- No matter how does the VM been constructed, it is always possible to convert **L1** to **L0**, or **Li** to **L0** ( $i \geq 1$ )
- There are two important conversion technologies
  - **Interpretation** – As **L1** program is running, each of its instruction can be **decoded** and **executed** by a program written in **L0**
  - **Translation/Compilation** – converting the entire **L1** program into an **L0** program

# The Computer Level Hierarchy

- ❑ The computer architecture can be divided into **at least 7 levels**
- ❑ Each level execute their own **particular instructions**, calling upon machines at **lower levels** to perform tasks as required.
- ❑ **Computer circuits** ultimately carry out the work.



# The Computer Level Hierarchy

---

## □ Level 6: The User Level

- Program execution and user interface level.
- The level with which we are most familiar.

## □ Level 5: High-Level Language Level

- The level with which we interact when we write programs in languages such as C, Java, SQL, etc.

# The Computer Level Hierarchy

---

## □ Level 4: **Assembly Language Level**

- Acts upon assembly language produced from Level 5, as well as instructions programmed directly at this level.

## □ Level 3: **System Software Level (OS!)**

- Controls executing processes on the system.
- Protects system resources.
- Assembly language instructions often **pass-through Level 3** without modification.

# The Computer Level Hierarchy

---

## □ Level 2: Machine Level

- Also known as the *Instruction Set Architecture* (ISA) Level.
- Consists of instructions that are particularly designed for the architecture of the machine.
- Programs written in machine language does not need compilers, interpreters, or assemblers.

# The Computer Level Hierarchy

---

## □ Level 1: Control Level

- A ***control unit*** decodes and executes instructions and moves data through the system.
- Control units can be ***micro-programmed*** or ***hardwired***.
  - A ***microprogram*** is a program written in a low-level language that is implemented by the hardware.
  - ***Hardwired control units*** consist of hardware that directly executes machine instructions.

# The Computer Level Hierarchy

---

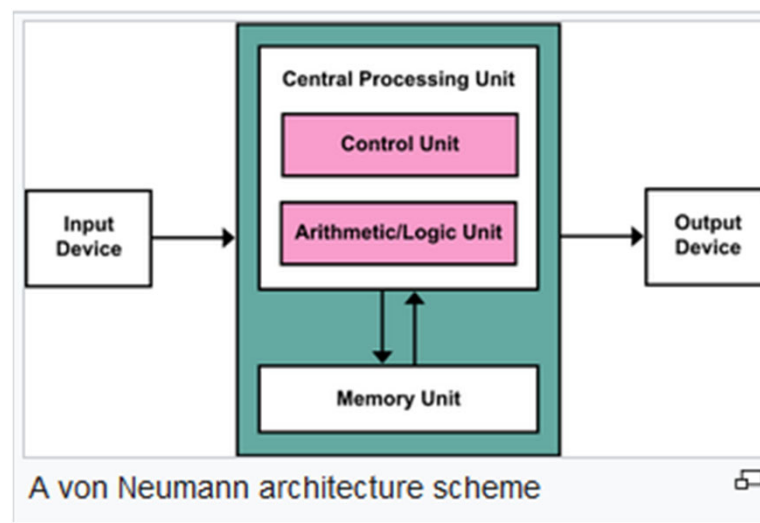
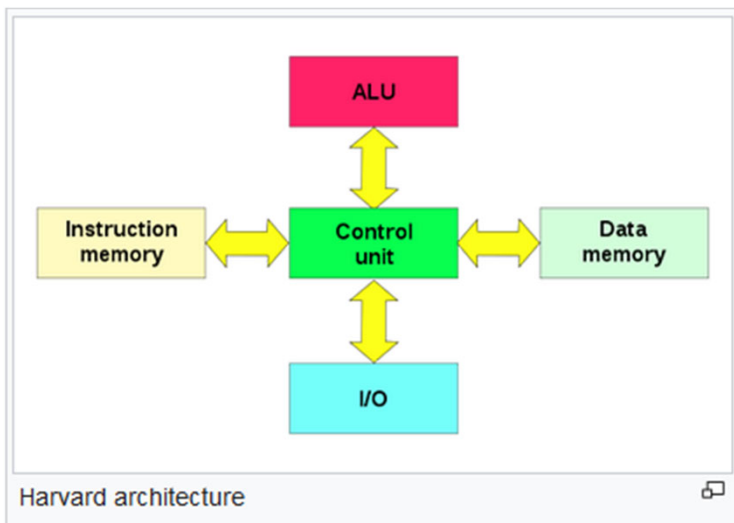
## □ Level 0: Digital Logic Level

- This level is where we find digital circuits (the chips).
- Digital circuits consist of gates and wires.
- These components implement the mathematical logic of all other levels.

# Chapter 1 Objectives

---

- ❑ Know the **concentration of** computer organization and computer architecture
- ❑ Understand **units** of measure common to computer systems
- ❑ Understand the computer as a **layered** system
- ❑ Can explain the **von Neumann/Princeton** architecture and the function of basic computer components.



[Wikipedia]

# The von Neumann Model

---

- On the first generation of computer, all programming was done at the *digital logic level*.
  - Programming the computer involved *moving plugs and wires*!
  - To solve a unique problem, it is required to use *different hardware configuration*!
- This kind of computer is known as “stored-program” computers
  - Which become known as, **von Neumann Architecture** systems.

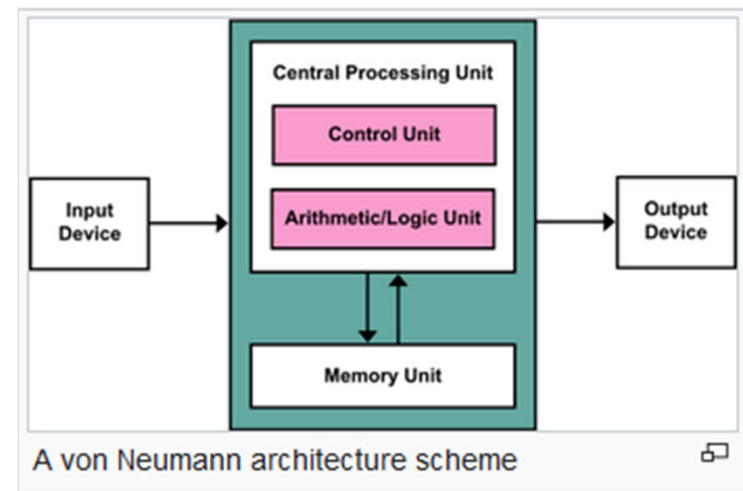
# The von Neumann Computer

---

- The von Neumann computers have the following characteristics:

- ***Three hardware systems:***

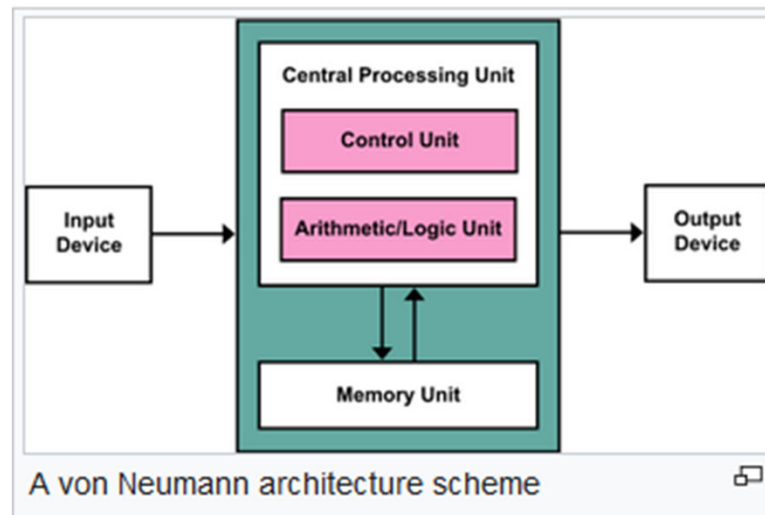
1. A central processing unit (CPU)
  - i. CU (Control Unit)
  - ii. ALU (Arithmetic and Logic Unit)
  - iii. Registers
2. Main memory
3. I/O sub-system



# The von Neumann Computer (`Cont.)

---

- ❑ A single data path between the CPU and main memory.
  - ❑ This single path is known as the *Bus*.



# An Example

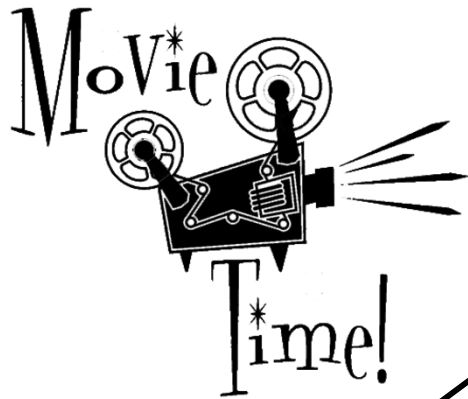
---

- Let's say you watched a movie and would like to share your ideas on Facebook

2. Memorize the content of the movie in your memory (Memory)

3. Organize your thoughts (CU & ALU)

4. "Tell" / Type your ideas (Output)



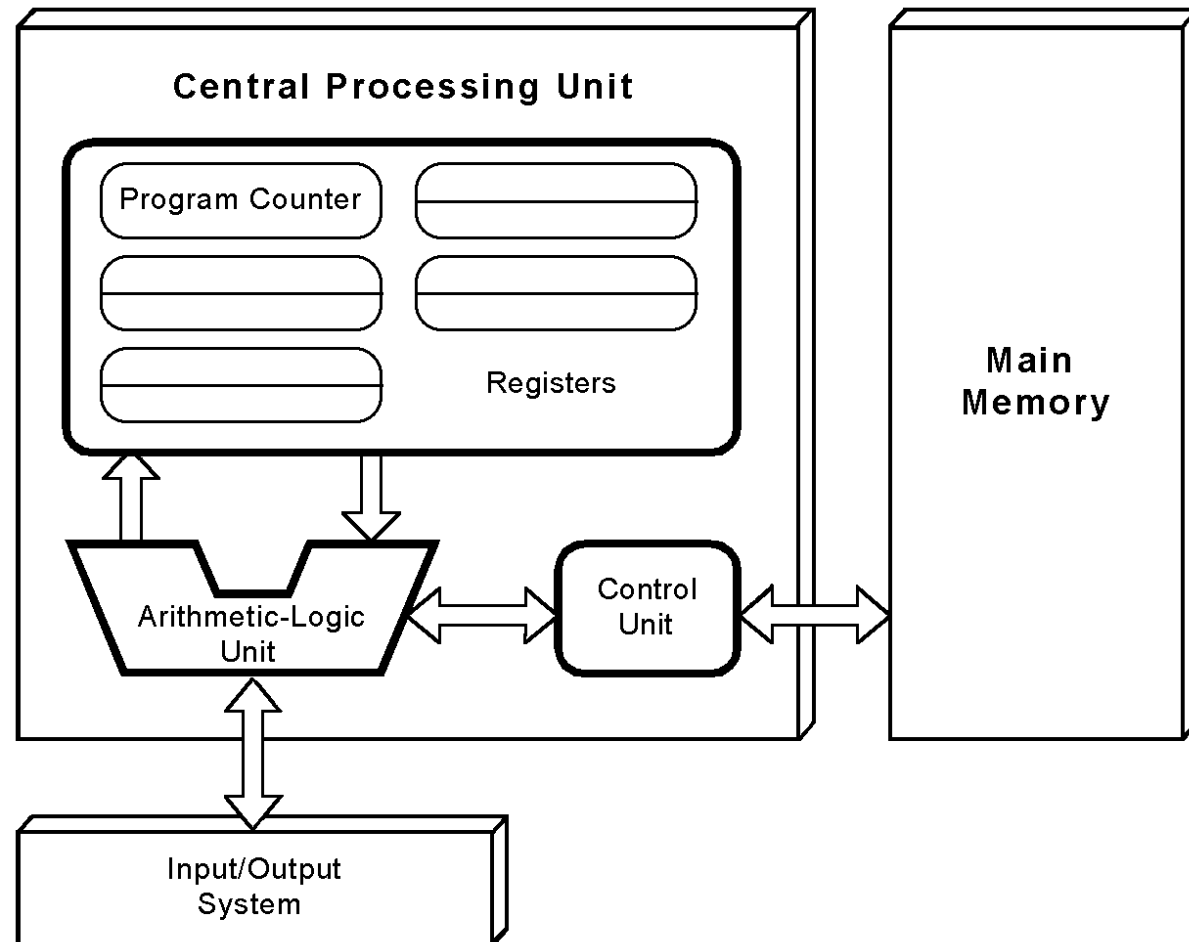
1. Watch the movie (Input)



# The von Neumann Model

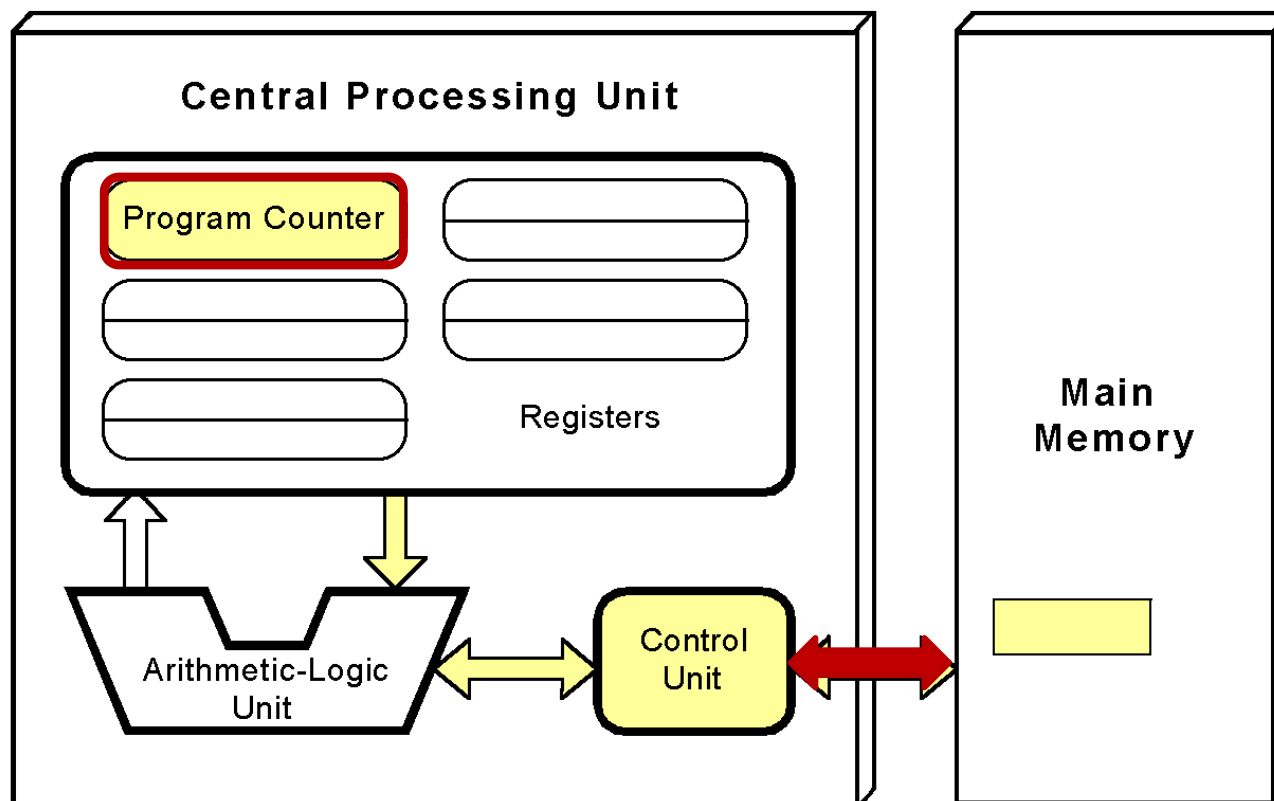
---

- These computers employ a *fetch-decode-execute cycle* to run programs



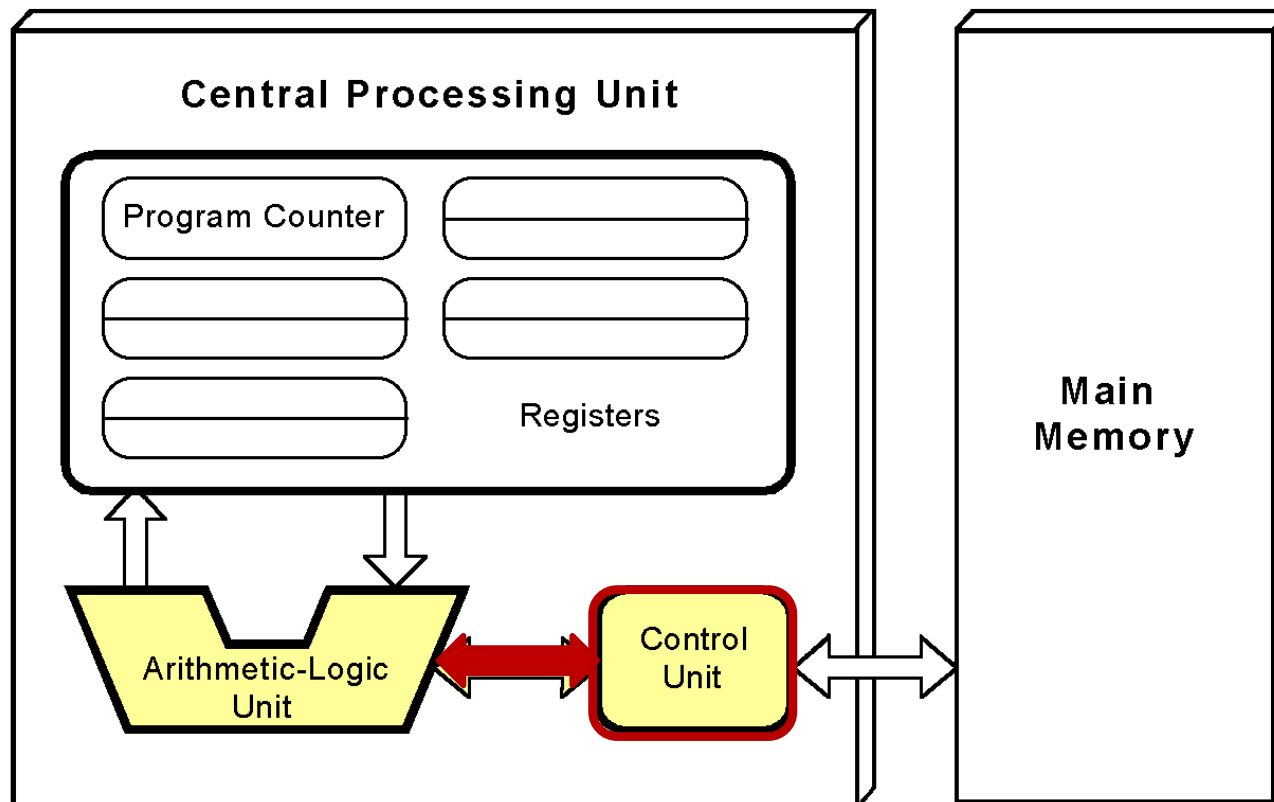
# Fetch-decode-execute Cycle (1)

- ❑ The **Control Unit** (CU) fetches the an instruction from memory
- ❑ It uses a register **Program Counter** (PC) to determine where the instruction is located



# Fetch-decode-execute Cycle (2)

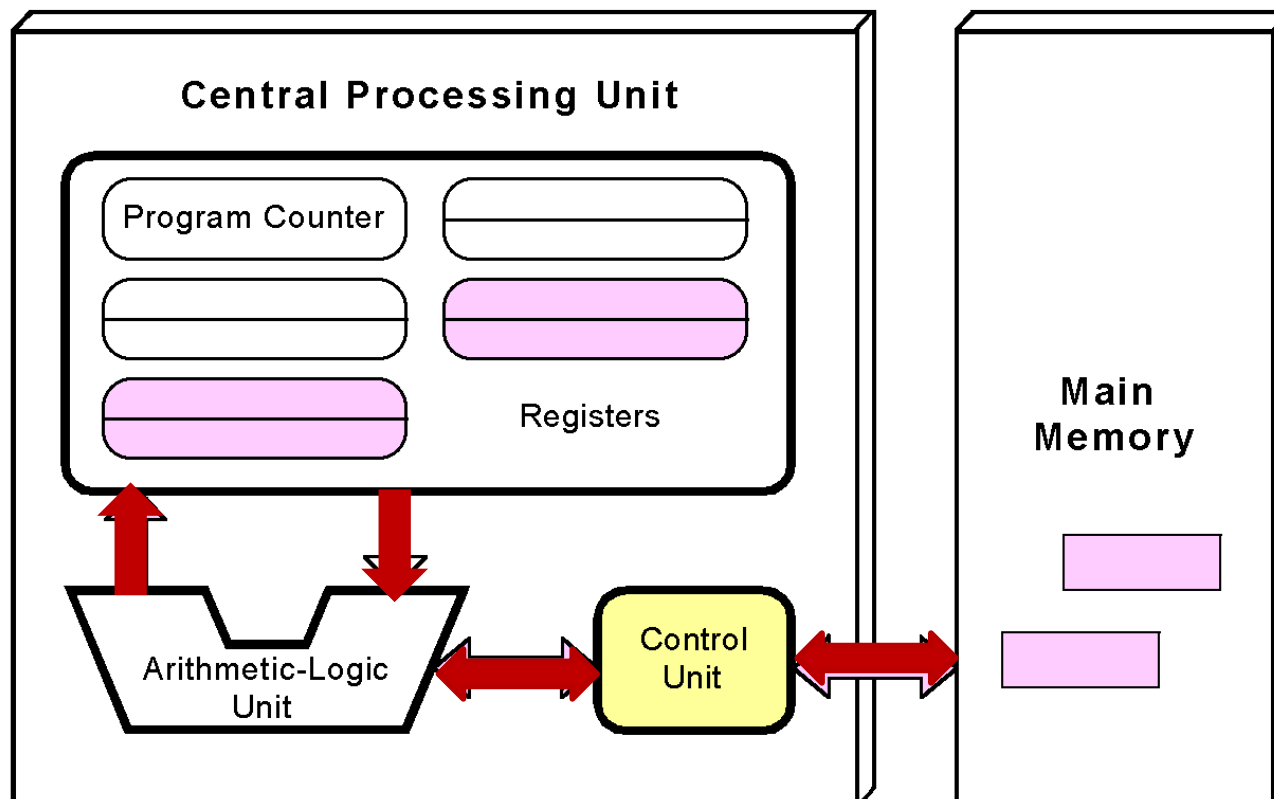
- The **CU** decodes the fetched instruction into a machine language that the **ALU** can “understand” and “reconfigure” itself



# Fetch-decode-execute Cycle

## (3)

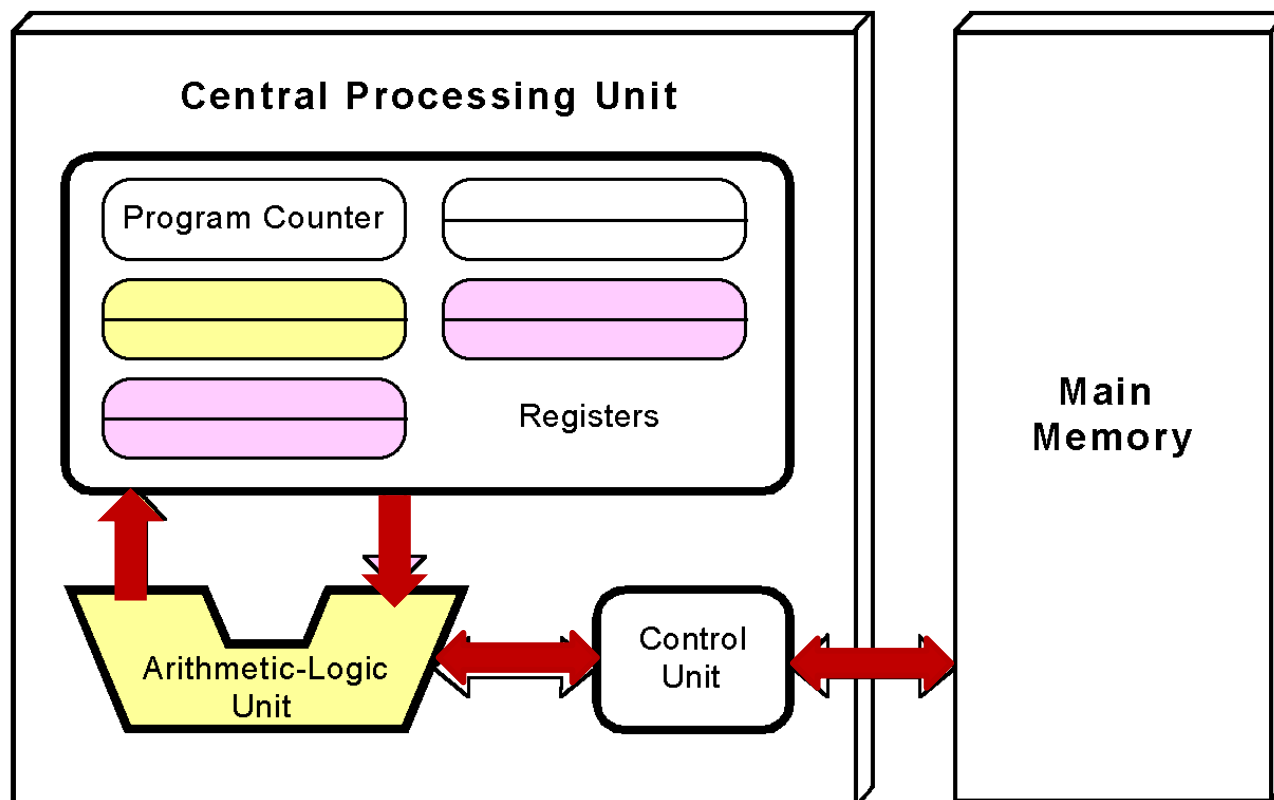
- Any *data operands* required to execute the instruction are fetched from *memory* and be placed into the *registers* inside the CPU



# Fetch-decode-execute Cycle

## (4)

- The **ALU** executes the instruction and places results in either the **registers** or back to the **memory**



# Summery: Fetch-decode-execute Cycle

---

- It is the fundamental cycle for the CPU to execute a program
  - The **Control Unit** (CU) fetches the next instruction from memory. It uses **Program Counter** (PC) to determine where the instruction is located.
  - The **CU** decodes the fetched instruction into a language that the **ALU** can understand.
  - Any data operands required to execute the instruction are fetched from **memory** and placed into the **registers** within the CPU.
  - The **ALU** executes the instruction and places results in either in the **registers** or the **memory**.

# Conclusion

---

- ❑ This chapter has given you an **overview** of the subject of computer **architecture**.
- ❑ You should now be sufficiently **familiar** with general system **structure** to guide your studies throughout the remainder of this course.
- ❑ Subsequent chapters will **explore and dig deeper** many of these topics in greater detail.

❑ Look out for HW1 on this Chapter-1; will post on D2L!



End of Chapter 1