Intro. to Computer

CS-3503 COA

Waqas Majeed
CS, CCSE, KSU

Computer?

- Computer: A machine (programmable) which computes!
  - Machine (automation): physical, powered, converts input to output.
  - Program(adaptation): sequence of instructions to perform tasks.
  - Made up of hardware (provides speed) and software (provides flexibility).

- Computation: an arithmetic or logical operations-based calculation (using algorithm).
  - Data manipulation: an auxiliary operation to carry out computation.
  - What about counting?

Counting and numbers

- Counting example:

  Odometer

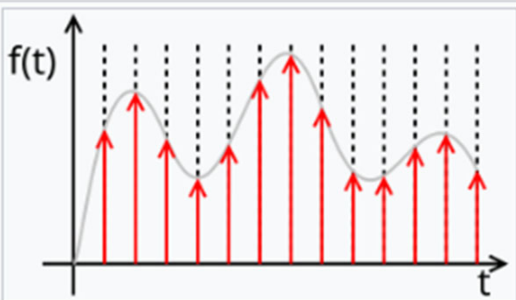
- Decimal numbering system
  - Example: $234.56

dimes cents

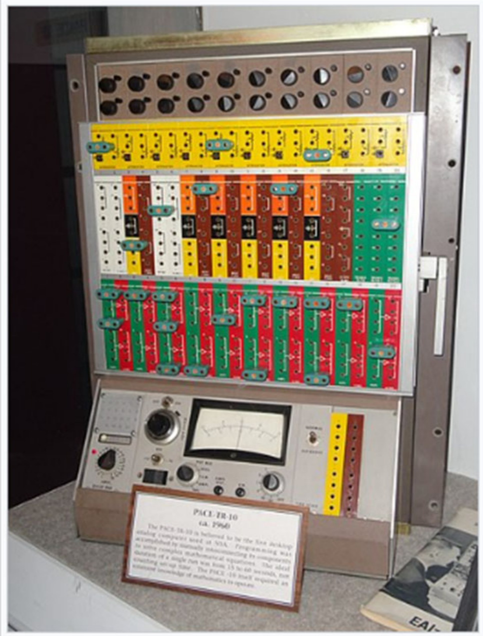
= 2 x $100 + 3 x $10 + 4 x $1 + 5 x $0.1 + 6 x $0.01

10^2 10^1 10^0 10^{-1} 10^{-2}

Modern Computer Types



Discrete sampled signal


TR-10 desktop analog computer of the late 1960s and early 1970s


EAI 8800 Analog computing system used for hardware-in-the-loop simulation of a Claas tractor (1986)

A digital computer has the discrete signals but only at two levels called high or low, or simply, 1 or 0.

Quantum Computer!

[Ref: Wikipedia]

Computer History

- BCs (Gen0)
- 1945 ENIAC (Gen1)

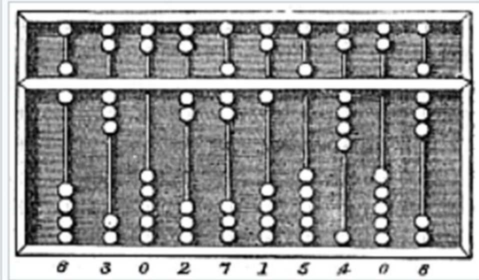
  vacuum tubes (electronic)

- 1964 IBM 360 (Gen2-4)

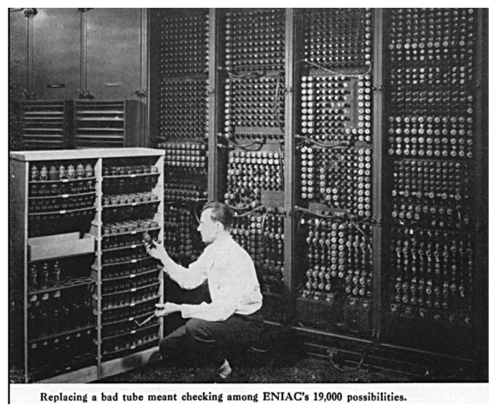
  Transistor --> uP-Electronics IC --> VLSIC, Embedded

- 2024 IBM Summit (9^{th} fastest; Gen5)

  Nano-electronics, AI, Quantum computing



The Chinese suanpan (算盘). The number represented on this abacus is 6,302,715,408.

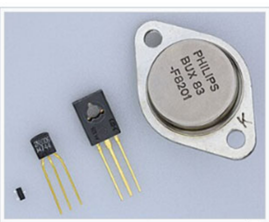


[Ref: Wikipedia]

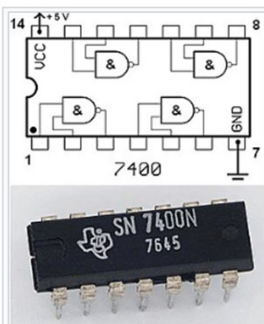
(counter-clockwise from top left)

Computer?

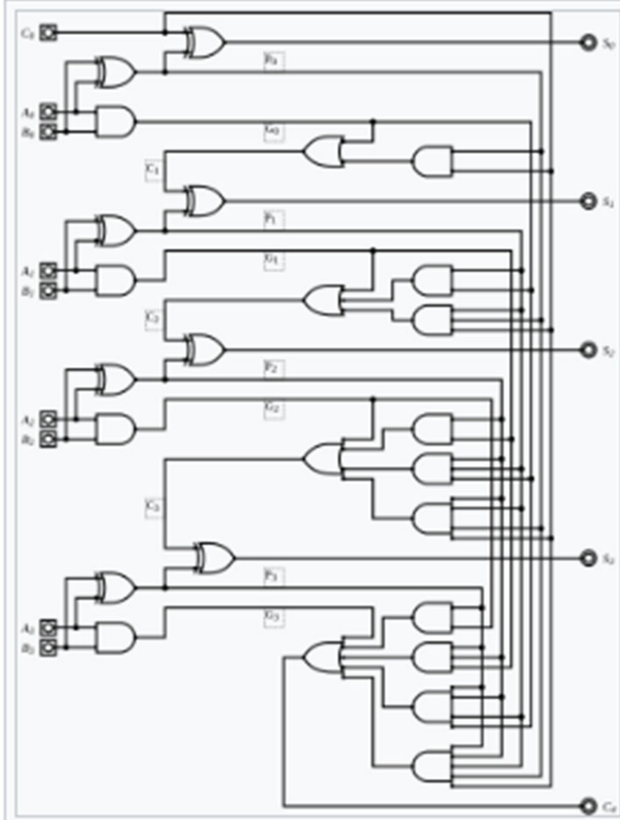
- Transistor (digital electronics switch)
- Digital logic gates
- Computer:
  - Memory
  - CPU



Size comparison of bipolar junction transistor packages, including (from left to right): SOT-23, TO-92, TO-126, and TO-3



The 7400 chip, containing four NANDs. The two additional pins supply power (+5 V) and connect the ground.



A logic circuit diagram for a 4-bit carry lookahead binary adder design using only the AND, OR, and XOR logic gates.

[Ref: Wikipedia]

Digital Computer implementation concept

Human World

Commands

Decimal numbers and Natural language

Design,
Software,
Customizability

Instruction Set
Architecture
(ISA)

format, access,
addressing, I/O, CPU
& Memory design, ...

Data Typification &
Conversion Methods

Computer World

Operation Codes

Binary representations

Implementation,
Purposeful Hardware,
Computing speed

Computer
Organization

Memory, control, signaling,
computation circuit realization
& operationalization

Processed information or knowledge

Computation: by example

- Compute: 5+4-3x2? Use PEMDAS algorithm. Hard coded!

  Imagine a computer implemented for this computation with

  hardwires for both the numbers and the operations.

- Imagine 100 such problems but each time with different numbers, i.e. same program/computation but different constants.

Soft coded!

a=7, b=8, c=2, d=3 <<< Data

a + b – c x d <<< Program

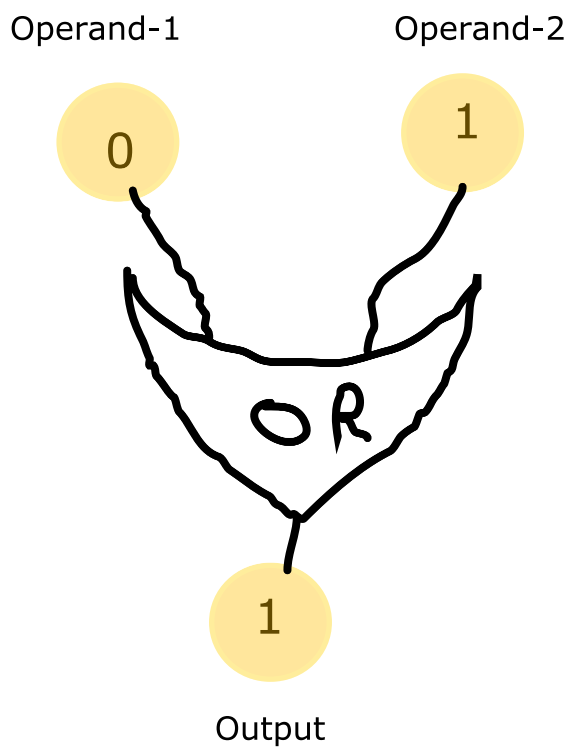
Computer Memory

(now, think of a large program with the data above which is used numerous times all over the place in this program)

Imagine a general-purpose computer hardware-implemented for the mathematical operations and any kind of numbers that can appear in this program, and the memory contents are easily changeable using software.

This will bring great computing time efficiency, power saving, cost reduction, human effort and tediousness reduction.

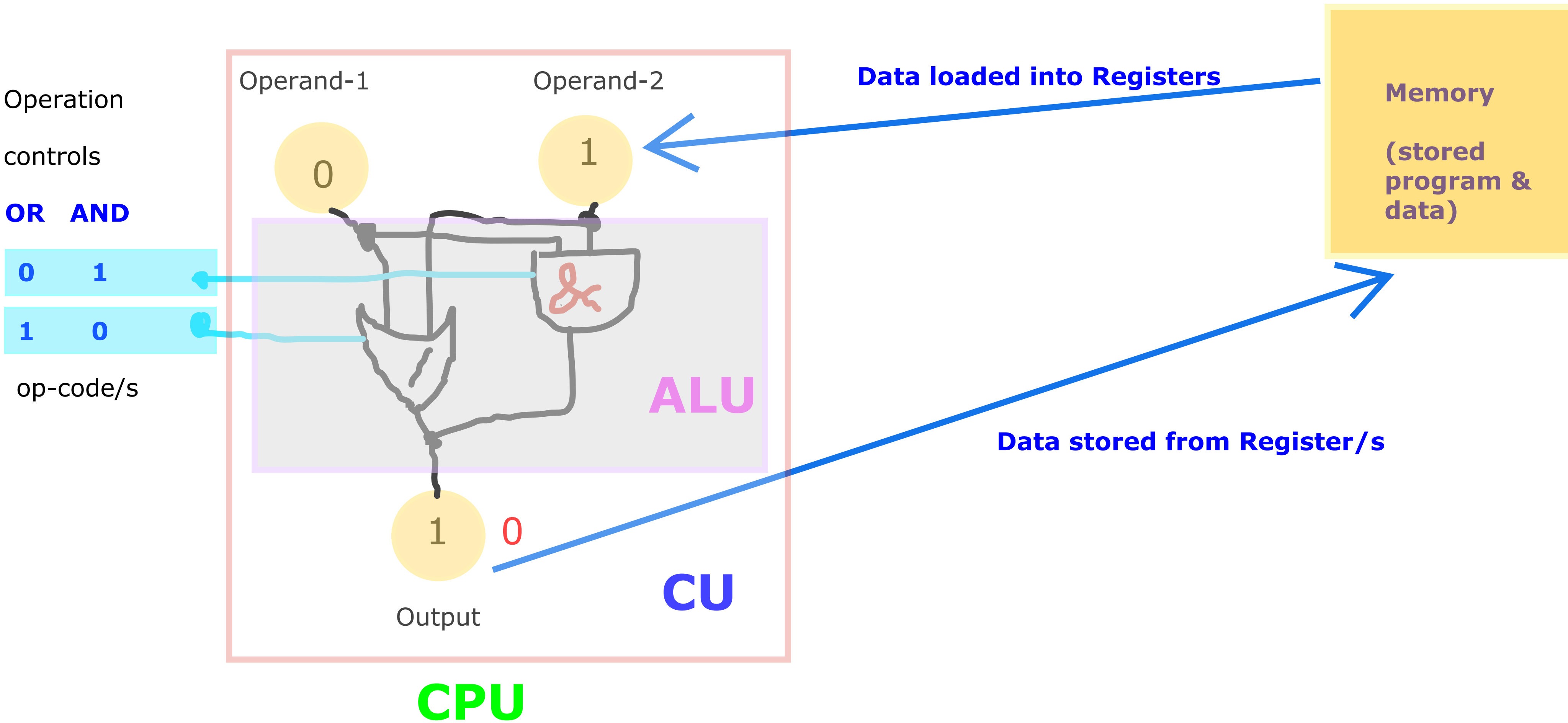
1-bit Digital Computer (Simplified)



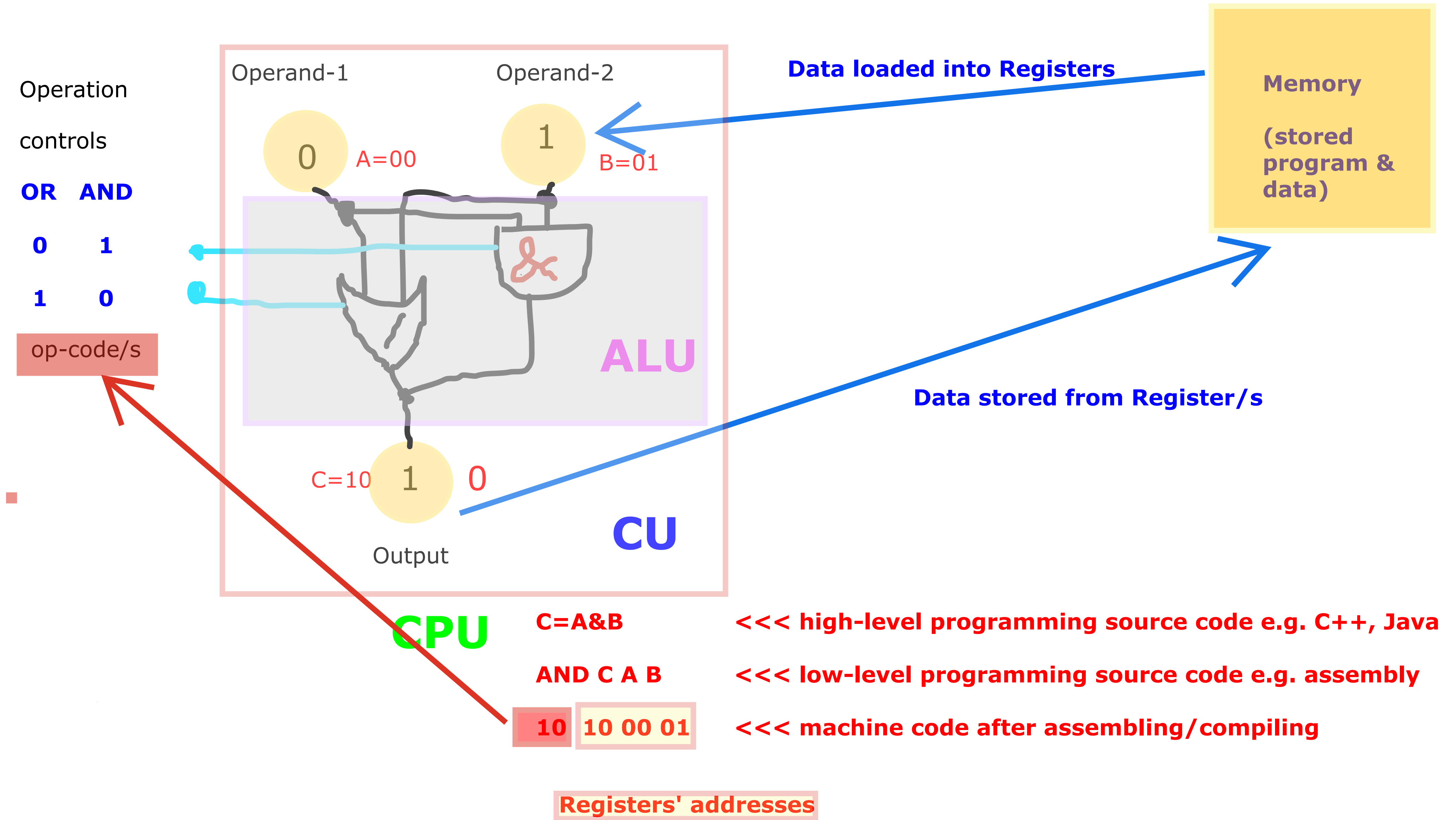
1. How to **speedup** and **automate** the different **data manipulation** into these input-operands and out of the output which are called **Registers**?

2. How to **quickly** (billion of times per second) **change** the mathematical **operations** to be carried out on the data **time to time**?

1-bit Digital Computer (General-purpose)

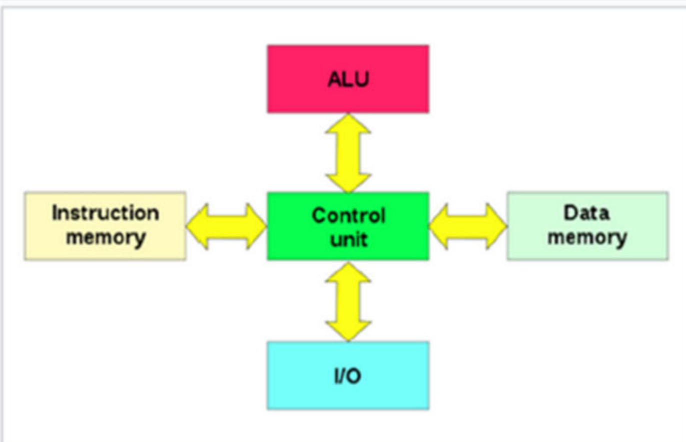


1-bit Digital Computer (General-purpose)

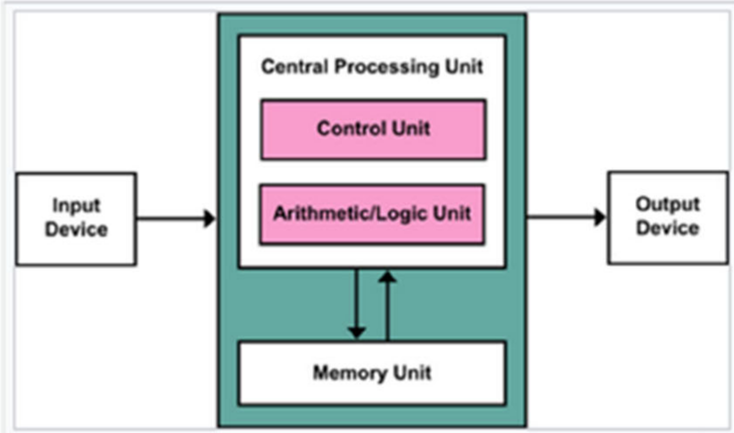


In this course, we will extrapolate the ideas learnt so far along with other modern computing concept to build or learn about bigger and more complex computers.

Computer Architectures

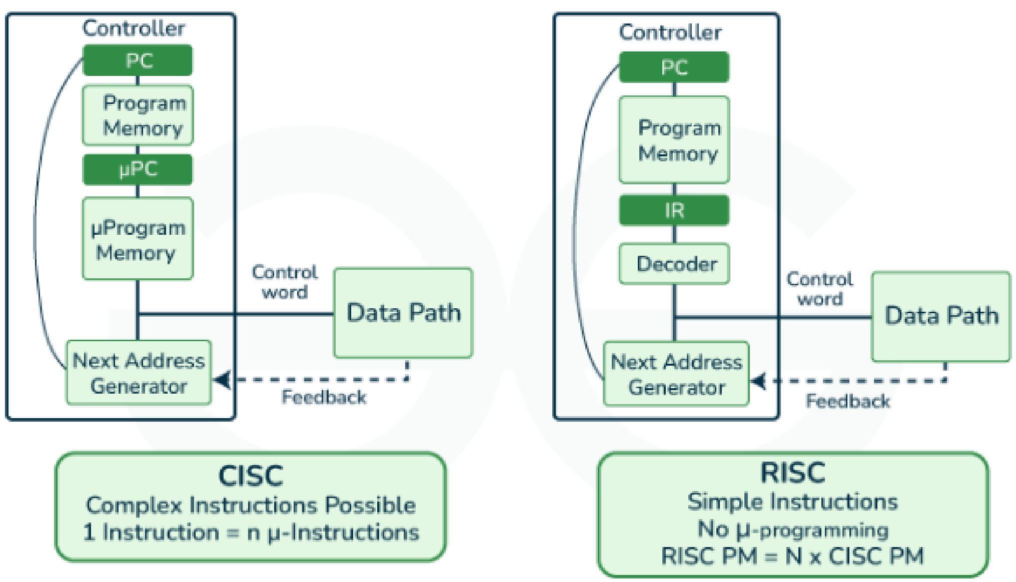


Harvard architecture



A von Neumann architecture scheme /Princeton

[Wikipedia]



[geeksforgeeks.org]

Chapter 1 - Introduction

> Glimpse of the whole computer, and its history
> Overview of the course

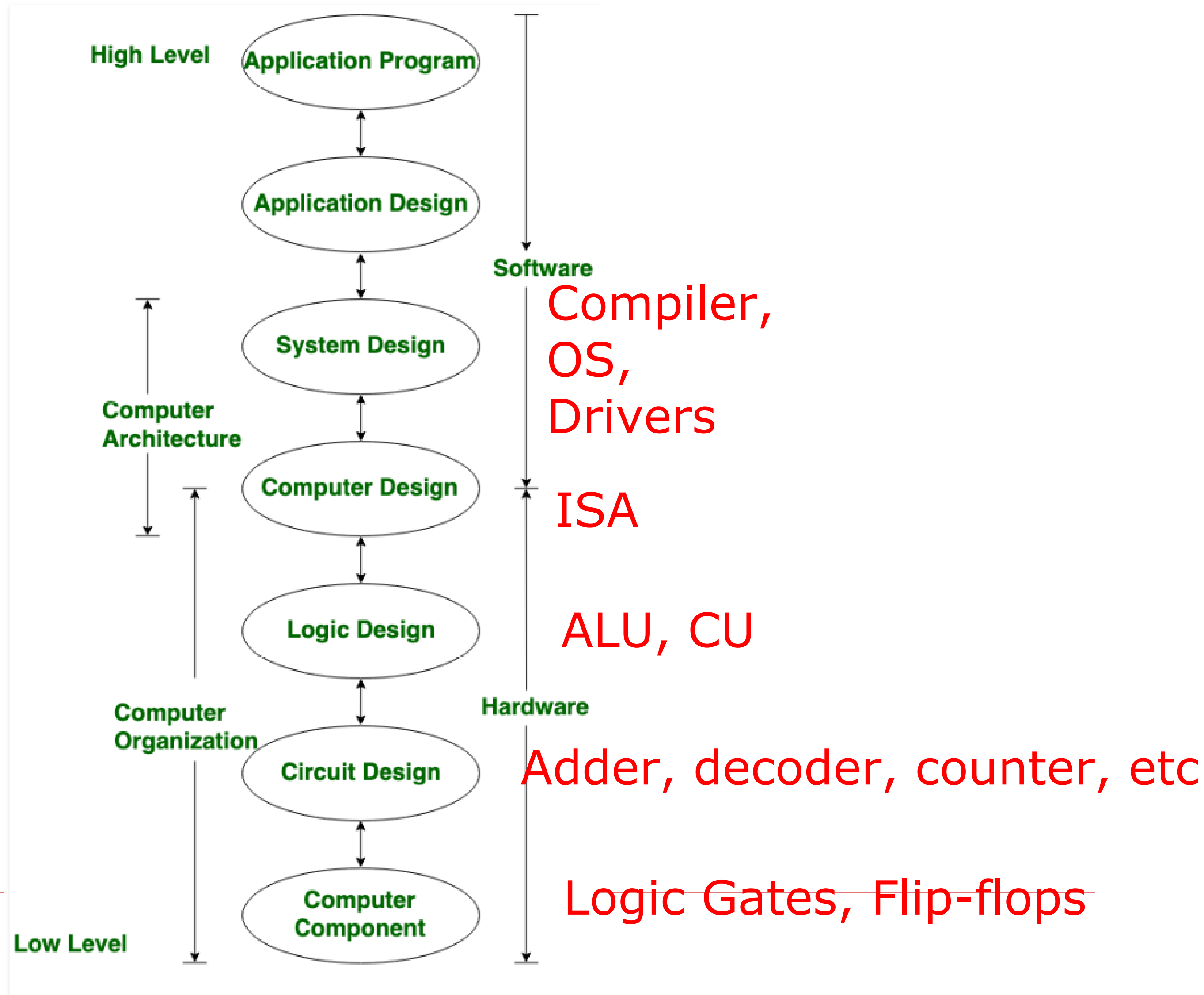
Chapter 1 Objectives

- 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



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??

USB??

PCI??

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
  - 1KB = 2^{10} = 1024 Bytes
  - 1MB = 2^{20} = 1,048,576 Bytes
  - 1G
- So,
  - 1KB $\neq$ 1000 Bytes
  - 1KB = 1024 Bytes

Kibibyte

Kibibyte is the unit of data with IEC binary prefix *Kibi*.

$$\text{Kibi} = 2^{10}$$

1 Kibibyte = 2^{10} bytes

1 Kibibyte = 1,024 bytes

IEC: International Electrotechnical Commission.

[https://www.gbmb.org/kib-to-kb]

Measures of Speed and Capacity (‘cont.)

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

1/ (133,000,000 cycles/second)
= 7.52ns/cycle

Measures of Time and Space

- Milli- (m) = 1 thousandth = 10^{-3}
- Micro- (μ) = 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.

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:

Low/High-level Code	Virtual machine VM1 (running L1)
Assembler, Compiler, Interpreter	ISA
Machine code	Virtual machine VM0 (running L0)

The Concept of Virtual Machine

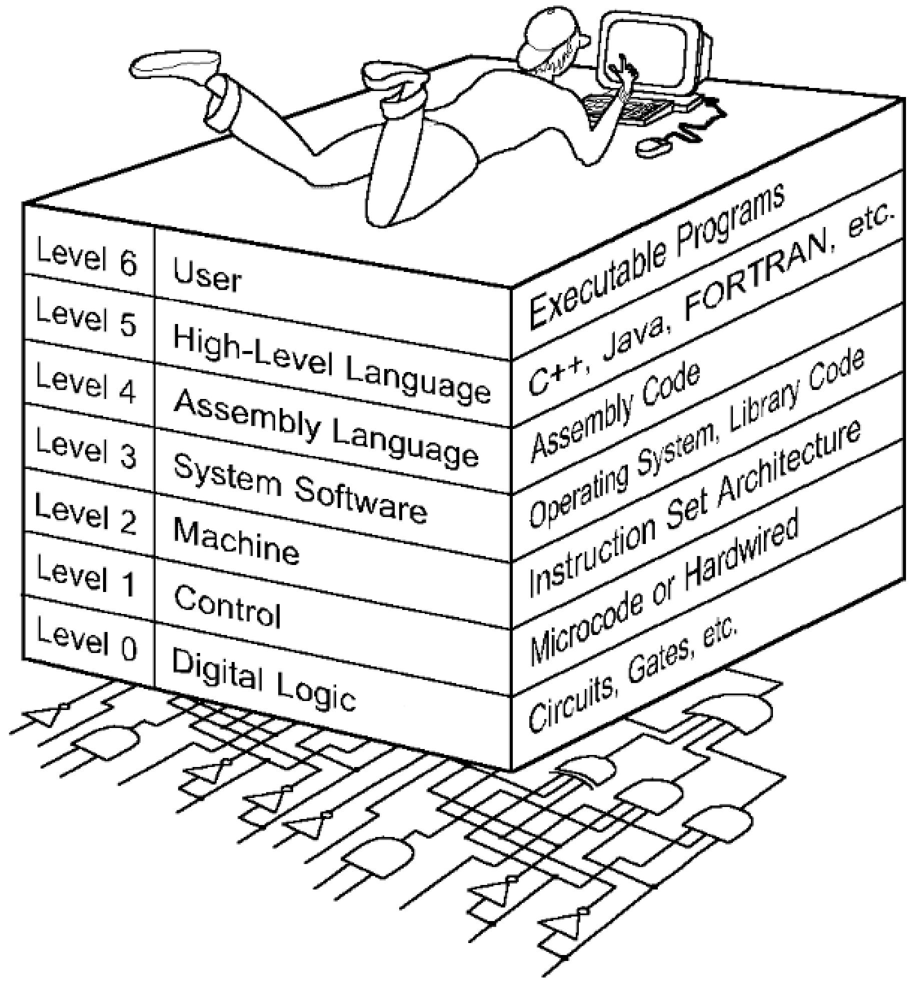
- No matter how does the VM been constructed, it is always possible to convert L1 to L0, or Li to L0 (i>=1)
- There are two important conversion technologies
  - **Interpretation** – As L1 program is running, each of its instruction can be decoded and executed by a program written in L0
  - **Translation/Compilation**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

The Computer Level Hierarchy

- 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.

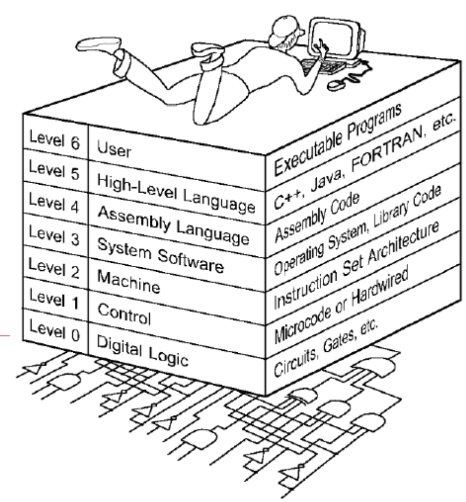
The Computer Level Hierarchy

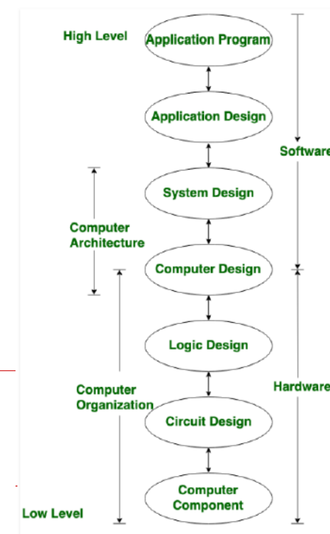
- The computer architecture can be divided into at least 7 levels
- Each level execute their own particular instructions, calling upon machines at lower levels to perform tasks as required.
- Computer circuits ultimately carry out the work.



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.



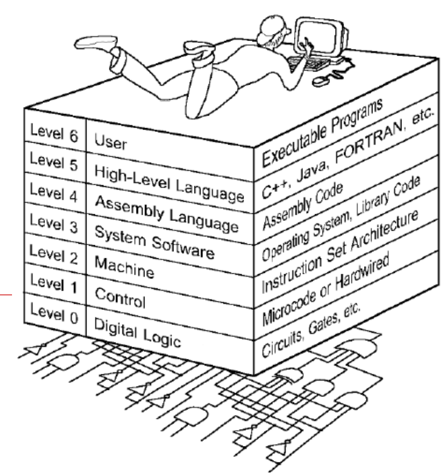


The Computer Level Hierarchy

- 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***.

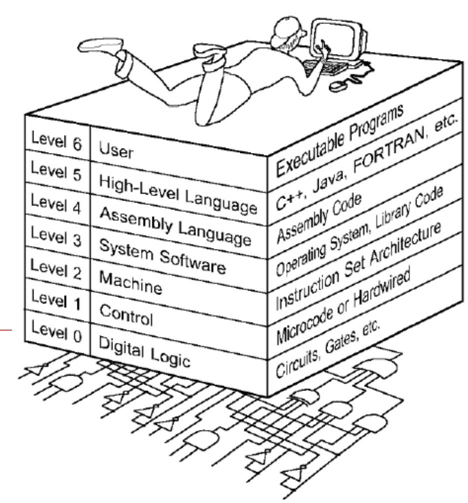
e.g. multiplication

- using sums

    - A ***microprogram*** is a program written in a low-level language that is implemented by the hardware.

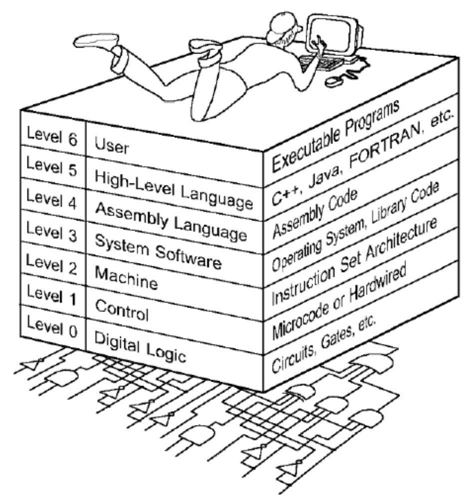
- using H/W

    - ***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.

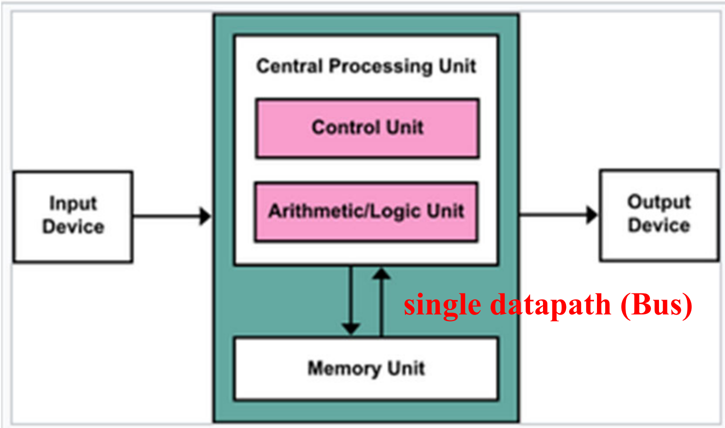


The von Neumann Model

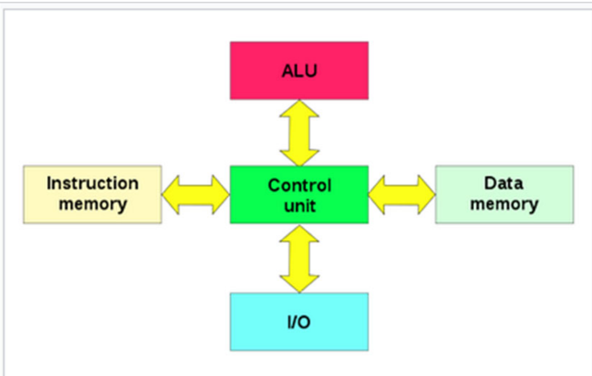
- On the first generation of computer, all programming was done at the *digital logic level*.
  - Programming the computer involved *moving plugs and wires*!
  - To solve a unique problem, it is required to use *different hardware configuration*!
- This kind of computer is known as "stored-program" computers
  - 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)
       1. CU (Control Unit)
       2. ALU (Arithmetic and Logic Unit)
       3. Registers
    2. Main memory
    3. I/O sub-system



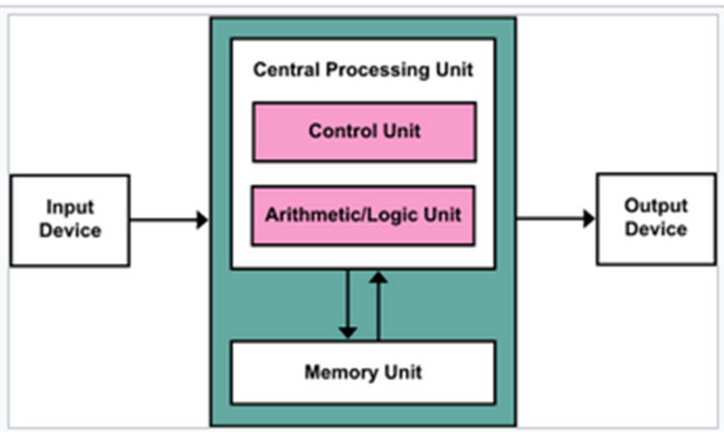
A von Neumann architecture scheme



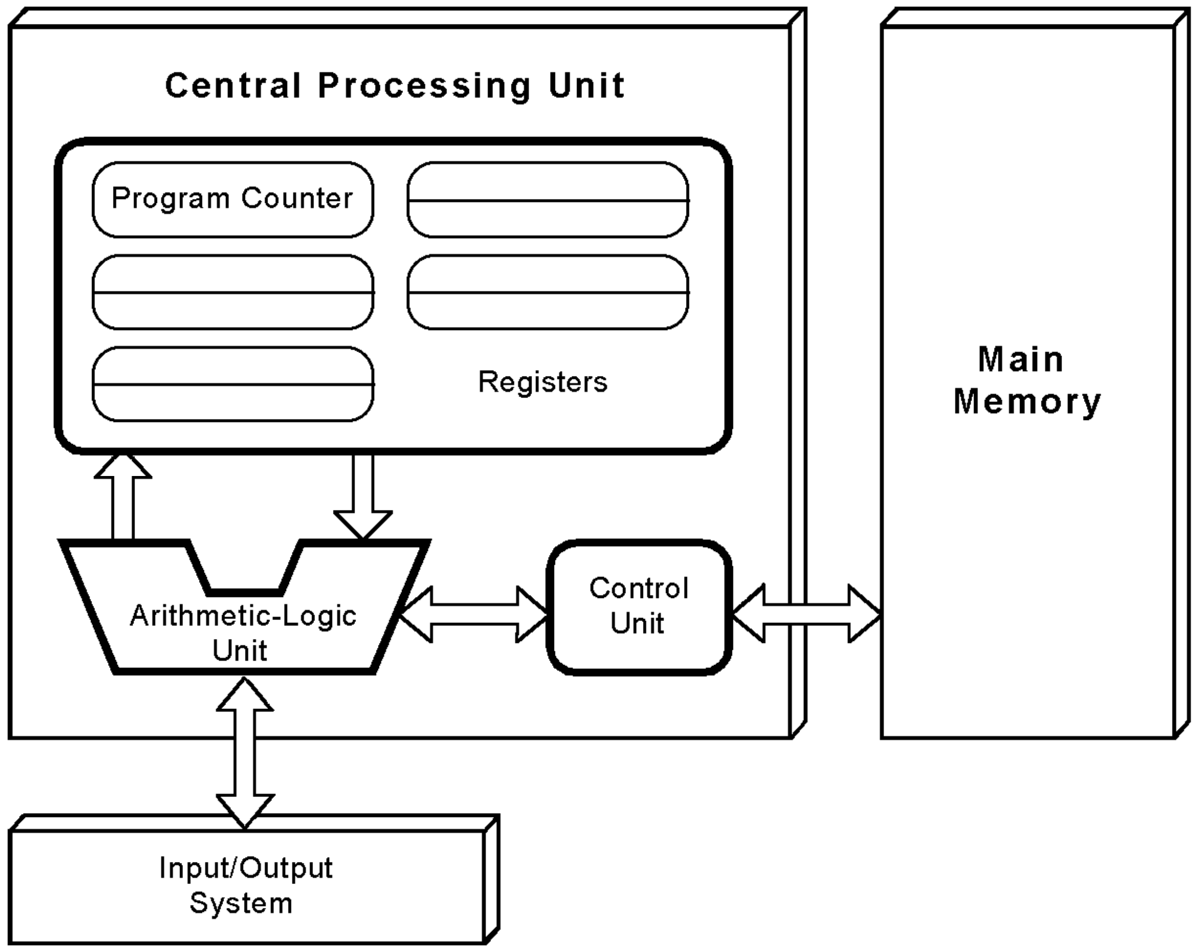
Harvard architecture

The von Neumann Model

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

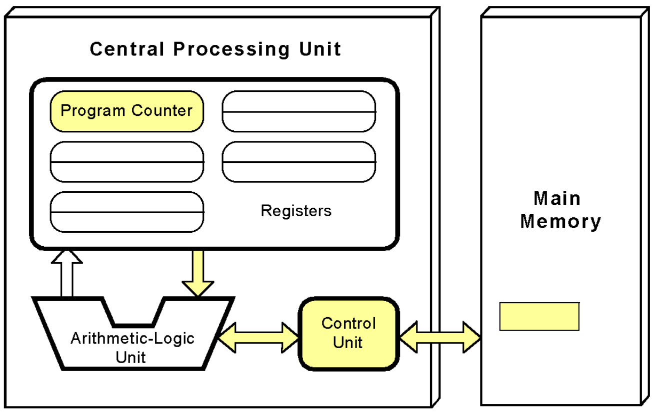


A von Neumann architecture scheme

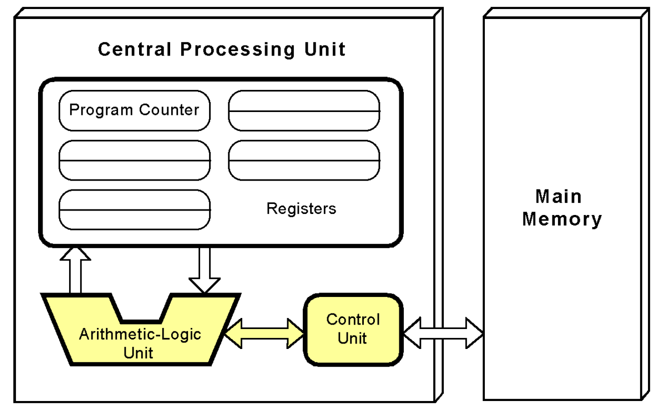


Fetch-decode-execute Cycle

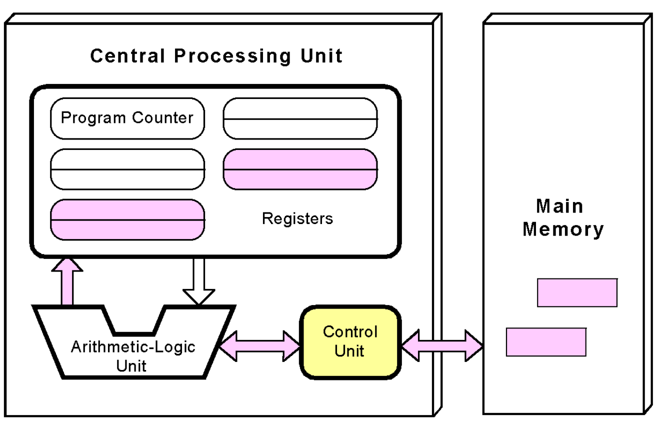
1. CU fetches an instruction



2. CU decodes the instruction & "reconfigures" the ALU



3. Needed data operands are fetched prior instruction execution



4. ALU executes the instruction