INTRODUCTION TO COMPUTER SCIENCE

Computer model: Von Neumann Model

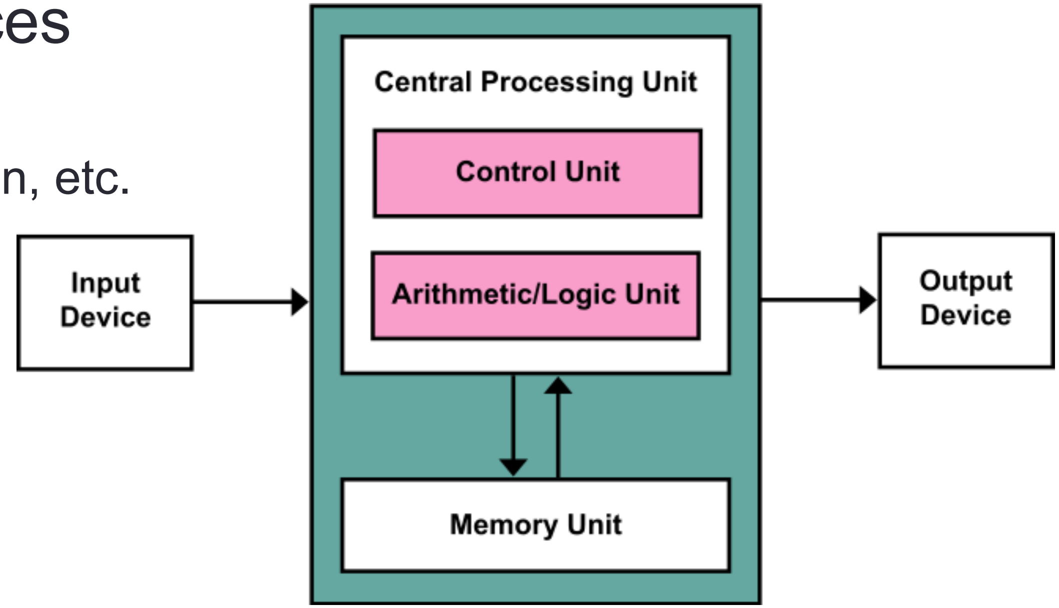
How programs and data are stored: Binary System

How computers are built: Logic Gates

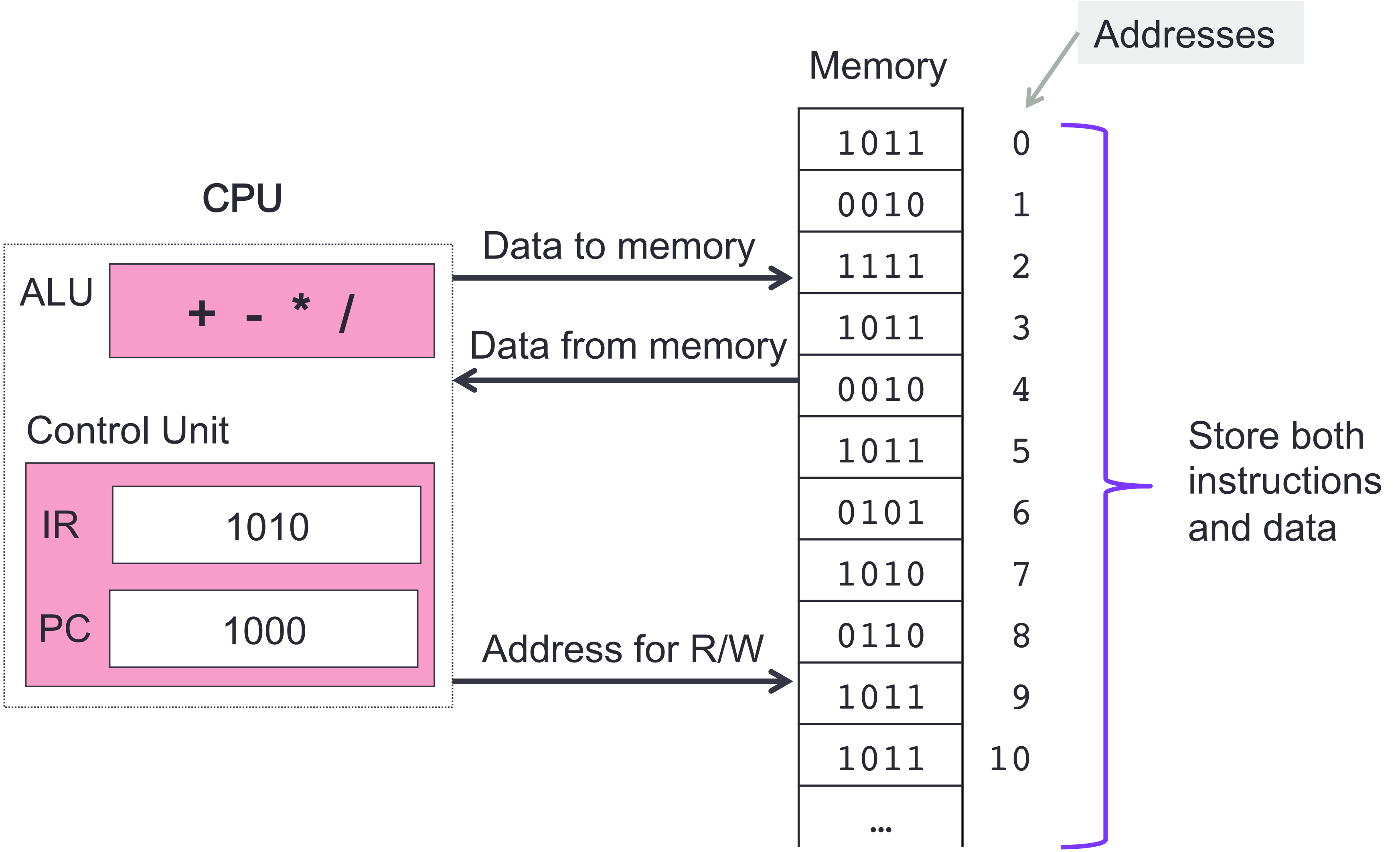
From higher level languages to machine language

Von Neumann Model

- Basic model of a computer architecture
- Processing Unit
  - ALU and processor registers
  - Control Unit: Program Counter and Instruction Register
  - Memory: holds data and instructions
- Input and output devices
  - Human Interface
    - Mouse, keyboard, screen, etc.
  - Storage
  - Networking
  - Graphics

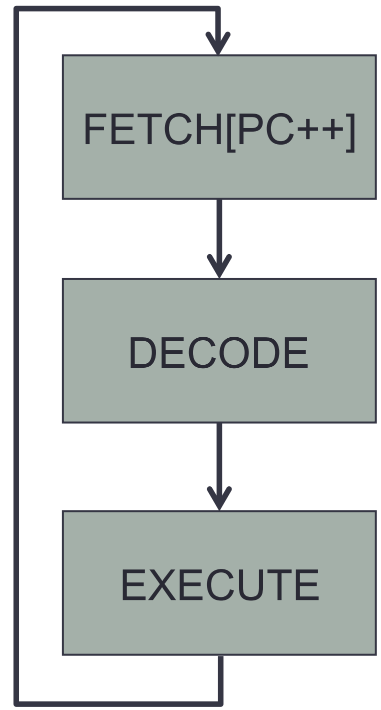


Von Neumann Model: closer look



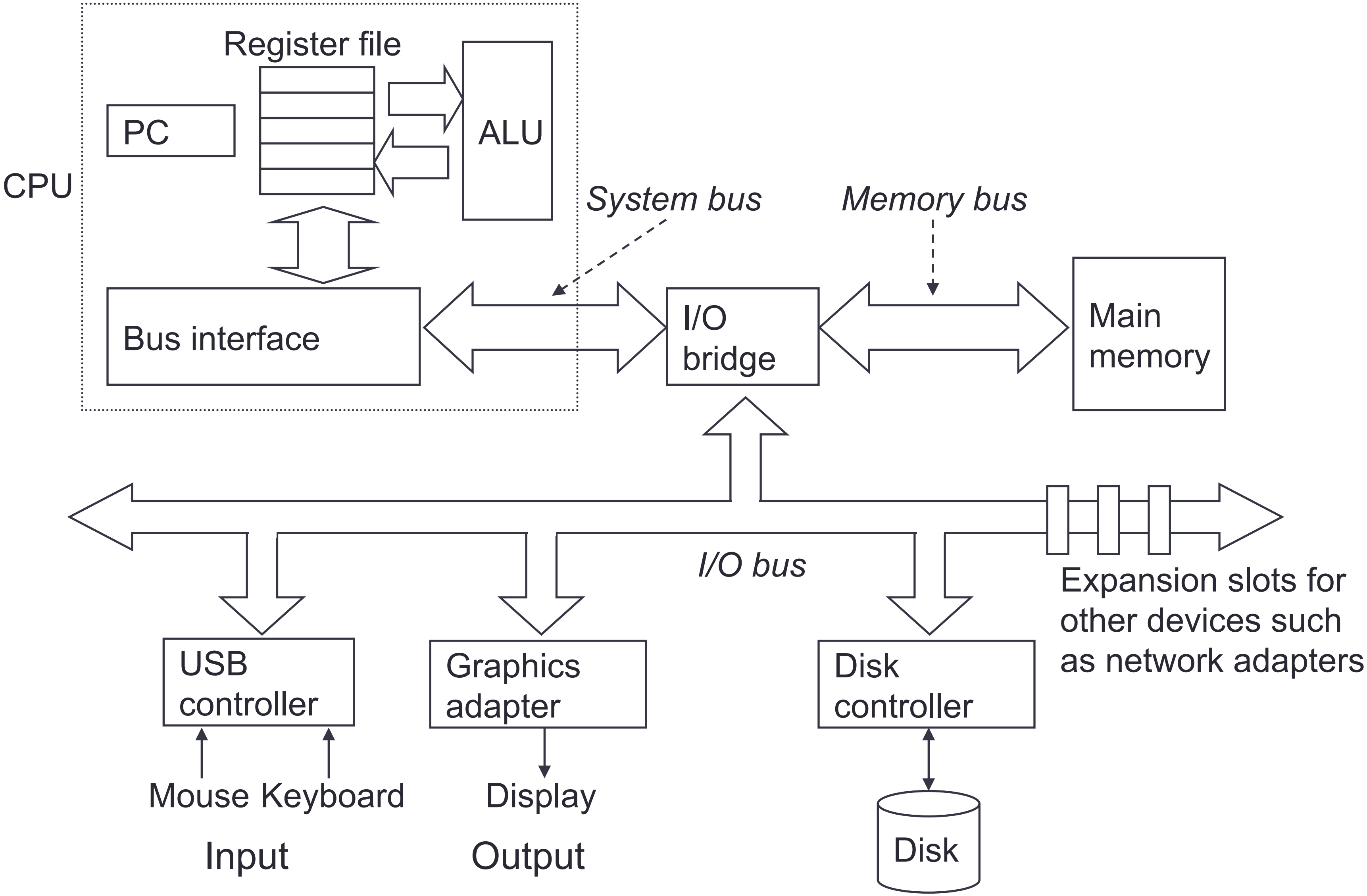
CPU Fetch-and-Execute Cycle

- Programs
  - Written in a high level language
  - Translated into machine language that can be executed by the CPU
- CPU executing a program
  - Program is in main memory



Arithmetic: +, -, *, /
Logic: bre, jmp

Von Neumann Model: in practice



How data is stored?

- Computers use the binary system to represent data.
- The *binary digit*, or *bit*, is the unit of computer memory.
- Any data from numbers, alphabet to images are represented using the binary system
  - Register file
  - Disk
  - Memory
  - Network

Binary Numbers

- Base 2
  - Symbols = {0,1} often called {false, true} or {off, on}
- Numbers are written as $d_n \ldots d_2 d_1 d_0$
- The decimal value of a binary number is $\sum_{i=0}^{n} d_i \times 2^i$
  - 101

1	0	1
2^2	2^1	2^0

$\rightarrow$ $2^2 + 2^0 = 5$

  - 1101

1	1	0	1
2^3	2^2	2^1	2^0

$\rightarrow$ $2^3 + 2^2 + 2^0 = 13$

Each position has a power of two value

- Binary representation is used in computers
- Bit and byte

How Many Binary Patterns from N Bits

Number of Bits	Number of Patterns	Number of Patterns as Power of Two
1	2	2^1
2	4	2^2
3	8	2^3
4	16	2^4
…	…	…
10	1024	2^{10}

Number of possible patterns of N bits = 2^N

1024 occurs often in Computer Science:

- 2^{10} bytes = 1024 bytes → 1 Kilobyte
- 2^{20} bytes = $2^{10} \times 2^{10}$ → 1024 Kilobytes (1 Megabytes)
- 2^{30} bytes = $2^{10} \times 2^{20}$ → 1024 Megabytes (1 Gigabytes)
- 2^{40} bytes = $2^{10} \times 2^{30}$ → 1024 Gigabytes (1 Terabytes)
- 2^{50} bytes = $2^{10} \times 2^{40}$ → 1024 Terabytes (1 Petabytes)

N-Bit Binary Addition

Binary Addition
0 + 0 = 0
0 + 1 = 1
1 + 0 = 1
1 + 1 = 0 (carry 1)

- Simple circuit
  Few basic logic gates

```
    2        010
+   4    +   100
  ---       ----
    6        110
```

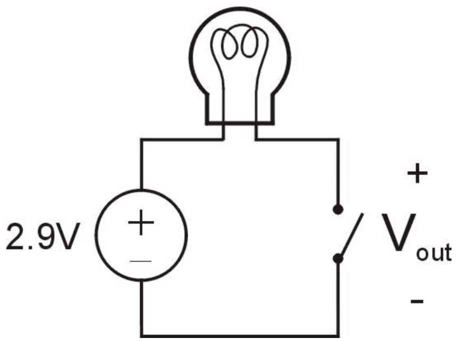
```
             11   ← carry
    3        011
+   1    +   001
  ---       ----
    4        100
```

So far we only know how to represent *unsigned* integers

- How to represent negative integers using the binary representation?

Transistor: Building Block of Computers

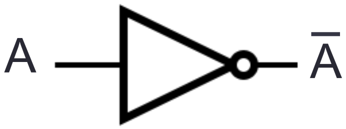
- Microprocessors contain millions (billions) of transistors
  - Intel Pentium 4 (2000): 48 million
  - IBM PowerPC 750FX (2002): 38 million
  - IBM/Apple PowerPC G5 (2003): 58 million
- Logically, each transistor acts as a switch
- Combine transistors to implement logic gates
  - AND, OR, NOT, NAND, NOR, XOR
- Combine gates to build higher-level structures
  - Adder, multiplexer, decoder, register, …
- Combine higher-level structures to build processor, memory and peripherals



- Switch open:
  - Light is off
- Switch closed:
  - Light is on

Logic Gates

Symbol



NOT

AND

OR

Truth Table

A	$\bar{A}$
0	1
1	0

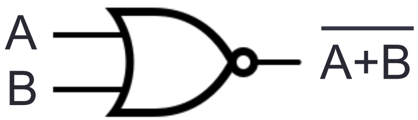
A	B	AB
0	0	0
0	1	0
1	0	0
1	1	1

A	B	A+B
0	0	0
0	1	1
1	0	1
1	1	1

Logic Gates

Symbol

NAND



NOR

XOR

Truth Table

A	B	$\overline{AB}$
0	0	1
0	1	1
1	0	1
1	1	0

A	B	$\overline{A+B}$
0	0	1
0	1	0
1	0	0
1	1	0

A	B	$A \oplus B$
0	0	0
0	1	1
1	0	1
1	1	0

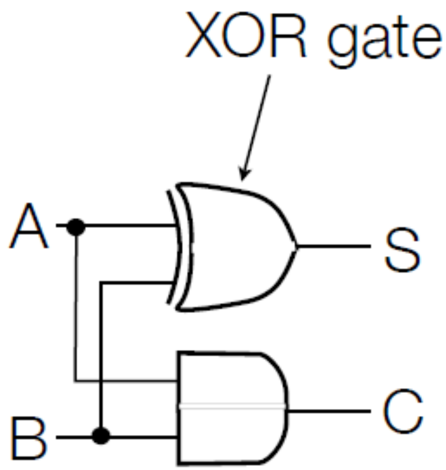
Addition: The Half Adder

- Addition of 2 bits: A & B produces summand (S) and carry (C)

A	B	S	C
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

$$S = A \oplus B$$

$$C = AB$$



- But to do addition, we need 3 bits at a time (to account for carries)

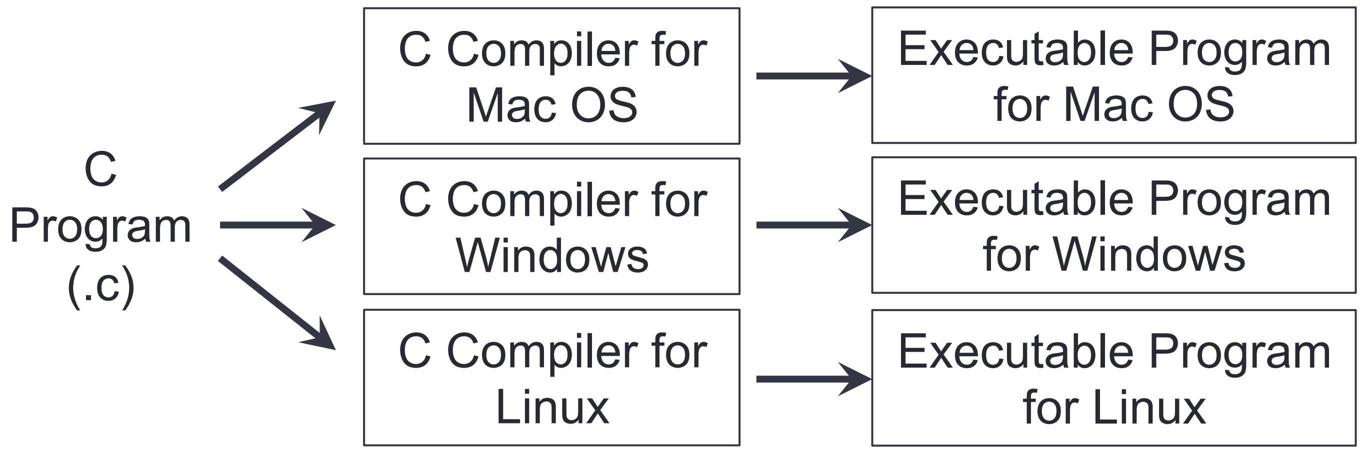
```
  011  ← carry bits
+ 1011
  1001
 10101
```

Program Meets Hardware

- Programs are written in higher level language
  - Java, C, C++, Perl, Python
- The CPU can execute very simple machine language instructions
  - Add, Sub, Jmp
- How to obtain runnable code from a program written in some programming language?
  - Compiler: translates a higher level language program into machine language program (executable). The executable program can be executed many times.
  - Interpreter: executes the computation written on a higher level language program.

Program Meets Hardware

- C uses compilation



gcc …

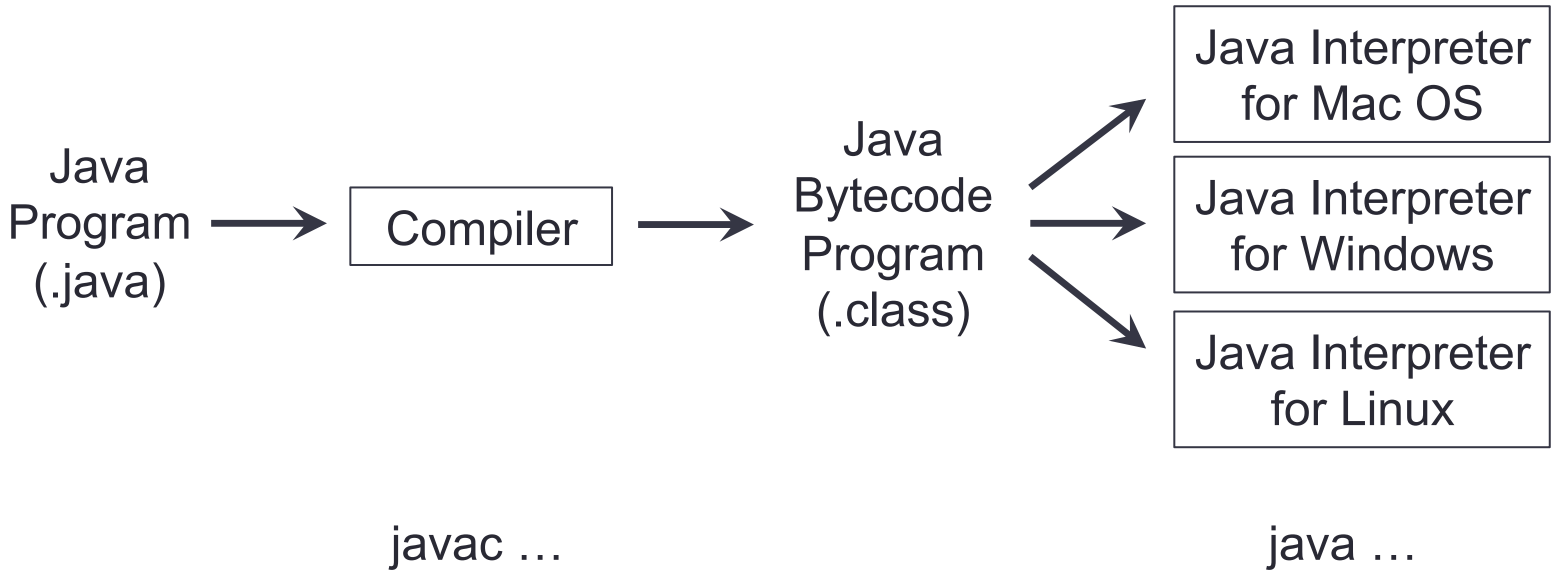
```
$ gcc Example.c

$ ./Example
```

Compiles Example.c into Example

Execute Example

Program Meets Hardware

- Java combines compilation and interpretation




```
$ javac Example.java

$ java Example.class
```

Compiles Example.java into Example.class

Interprets Example.class (JVM)

Wrapping Up

- Von Neumann Model
  - Some CS courses will dive into a piece of this model while others make use of the model as a whole
- We understand that computers use the binary system to represent data
- Basic building blocks of a computer
  - The adder inside the CPU is built from a XOR and a AND gates
- How programs written in higher level languages are executed by the CPU