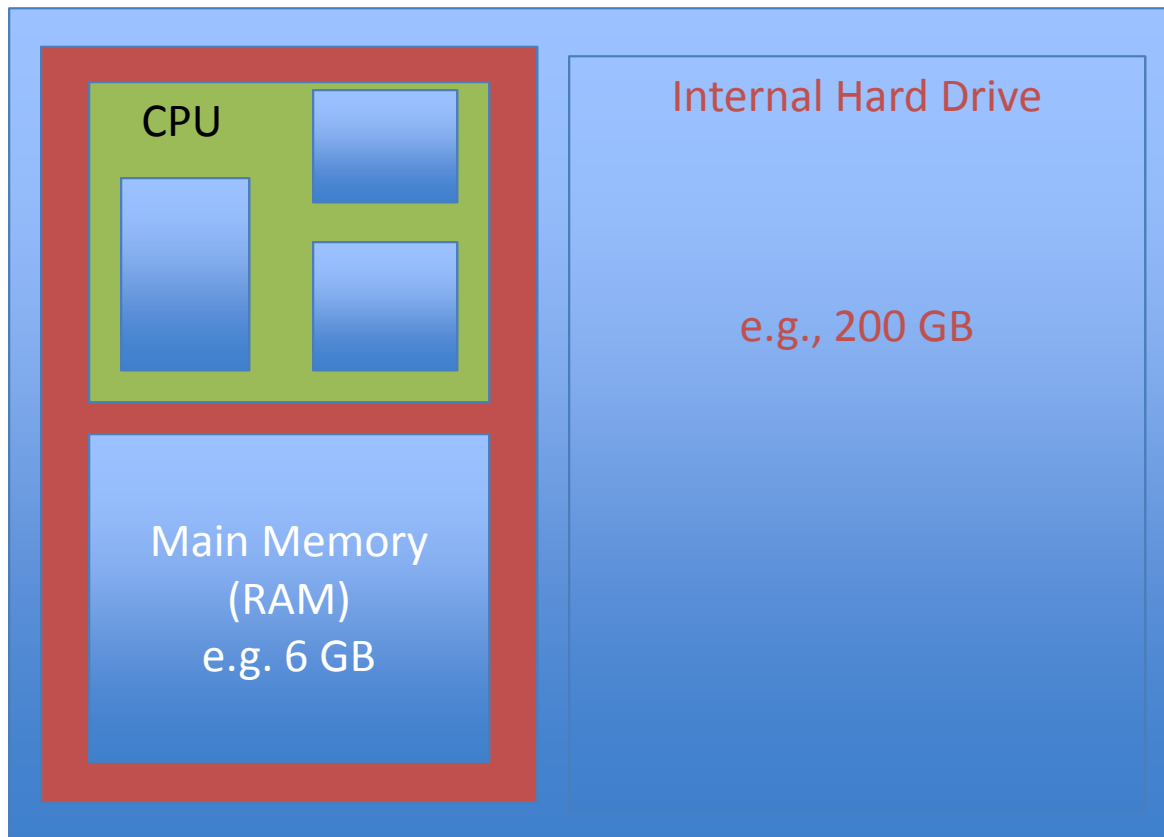


Computer Organization

Computer Organization

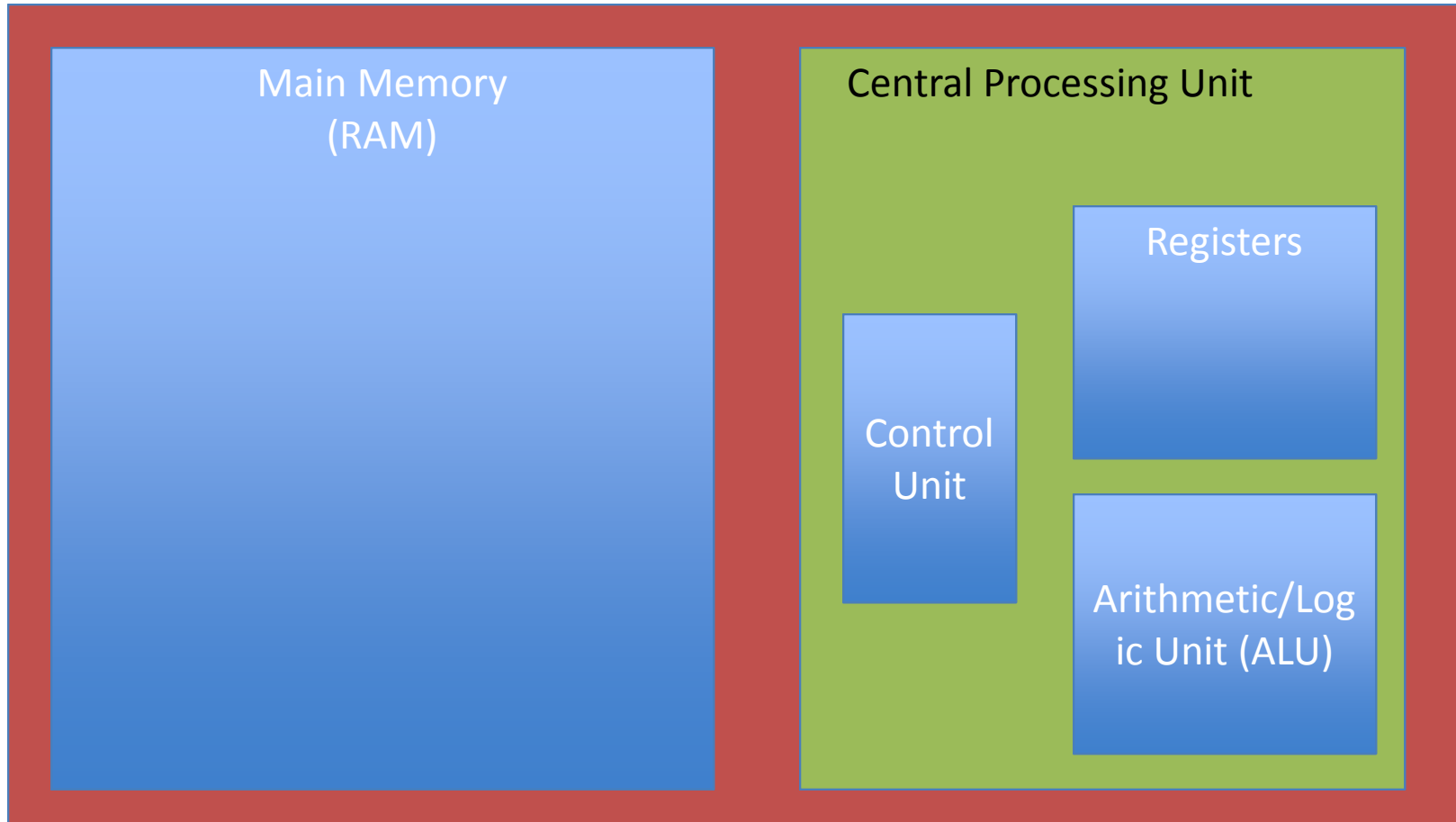
Computer



External Devices

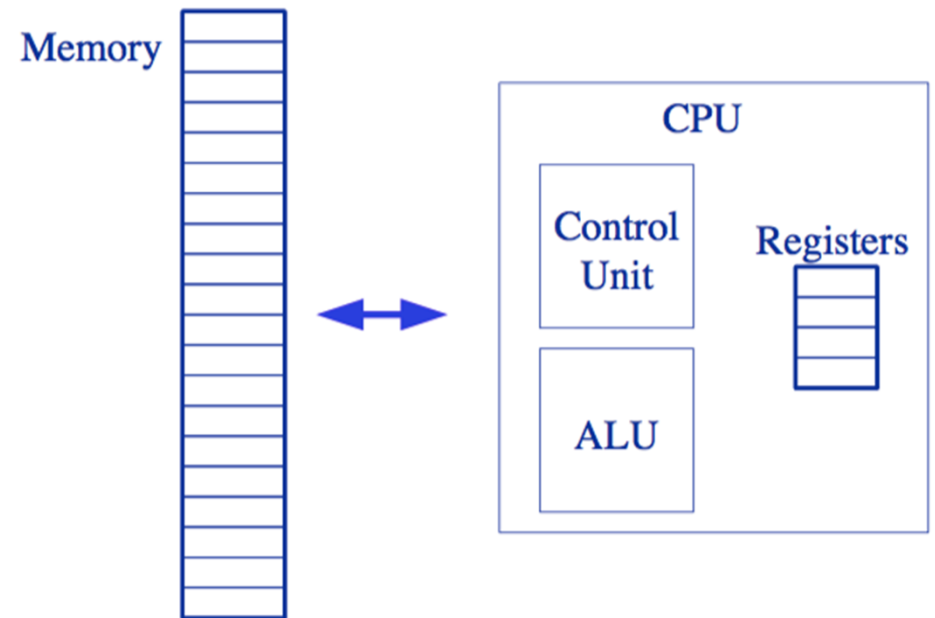


CPU and Memory



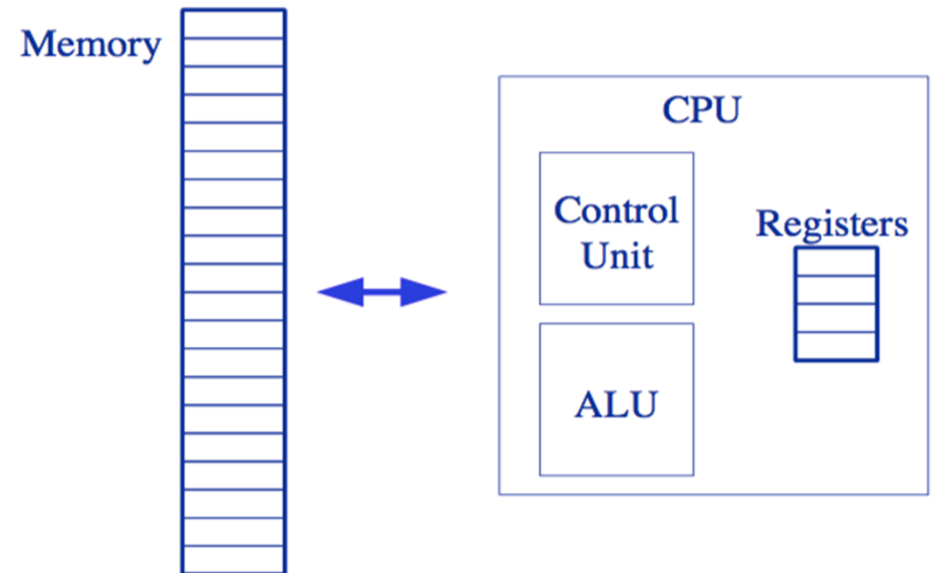
Von Neumann Architecture

- Program and Data are both stored in memory (Stored Program concept)
- Fetch / Execute cycle...



What can a CPU do?

- Fetch an instruction from memory
- Execute:
 - Copy data from/to memory to/from registers
 - Basic operations: add/subtract 2 registers, etc
 - Other operations: shift bits, compare against 0, jump to a different part of the program



What do machine instructions look like?

- Example:

100011001010

100011011011

100000100001

101011100110

111111111111

Machine Language

(each type of CPU has its own machine language)

- What does it mean?

What do machine instructions look like?

- Example:

100011001010	Machine Language
100011011011	
100000100001	
101011100110	
111111111111	

- What does it mean?

100011 00 1010	LOAD R0 10
100011 01 1011	LOAD R1 11
100000 10 00 01	ADD R2 R0 R1
101011 10 0110	STORE R2 12
111111111111	HALT

What do machine instructions look like?

- Example:

100011001010	Machine Language
100011011011	
100000100001	
101011100110	
111111111111	

- What does it mean?

100011 00 1010	LOAD R0 10	Assembly Language
100011 01 1011	LOAD R1 11	
100000 10 00 01	ADD R2 R0 R1	
101011 10 0110	STORE R2 12	
111111111111	HALT	

Assembly Language

- Set of mnemonic names for the instructions in a particular computer's machine language.
- Works on registers and memory locations in the computer.
- Translates directly into machine language (binary instructions).

But what about [JavaScript/Python]?

- Programming languages cannot be interpreted directly by the computer – they are too high-level, and are not in binary. They need to be translated to machine language.
- Compilers/Interpreters:
 - Compiler translates the entire program at one go.
 - Interpreter translates the program one statement at a time.