# Lecture 9

## Stored Program Concept: The Hardware View

# The von-Neumann Architecture

- Processor

  carries out instructions

- Memory

  temporary storage

- Input/Output

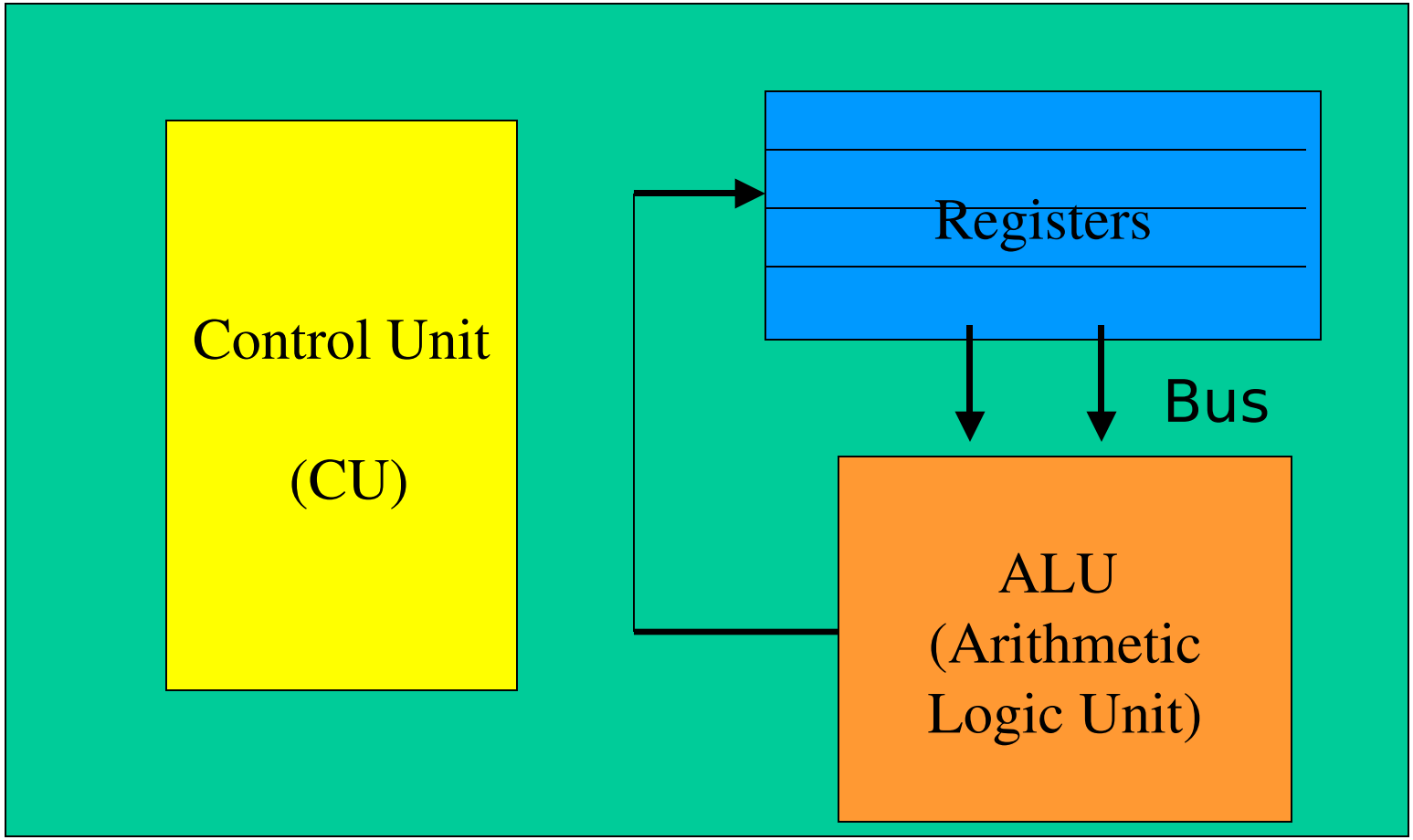
  allows user interaction

# CPU – Central Processing Unit

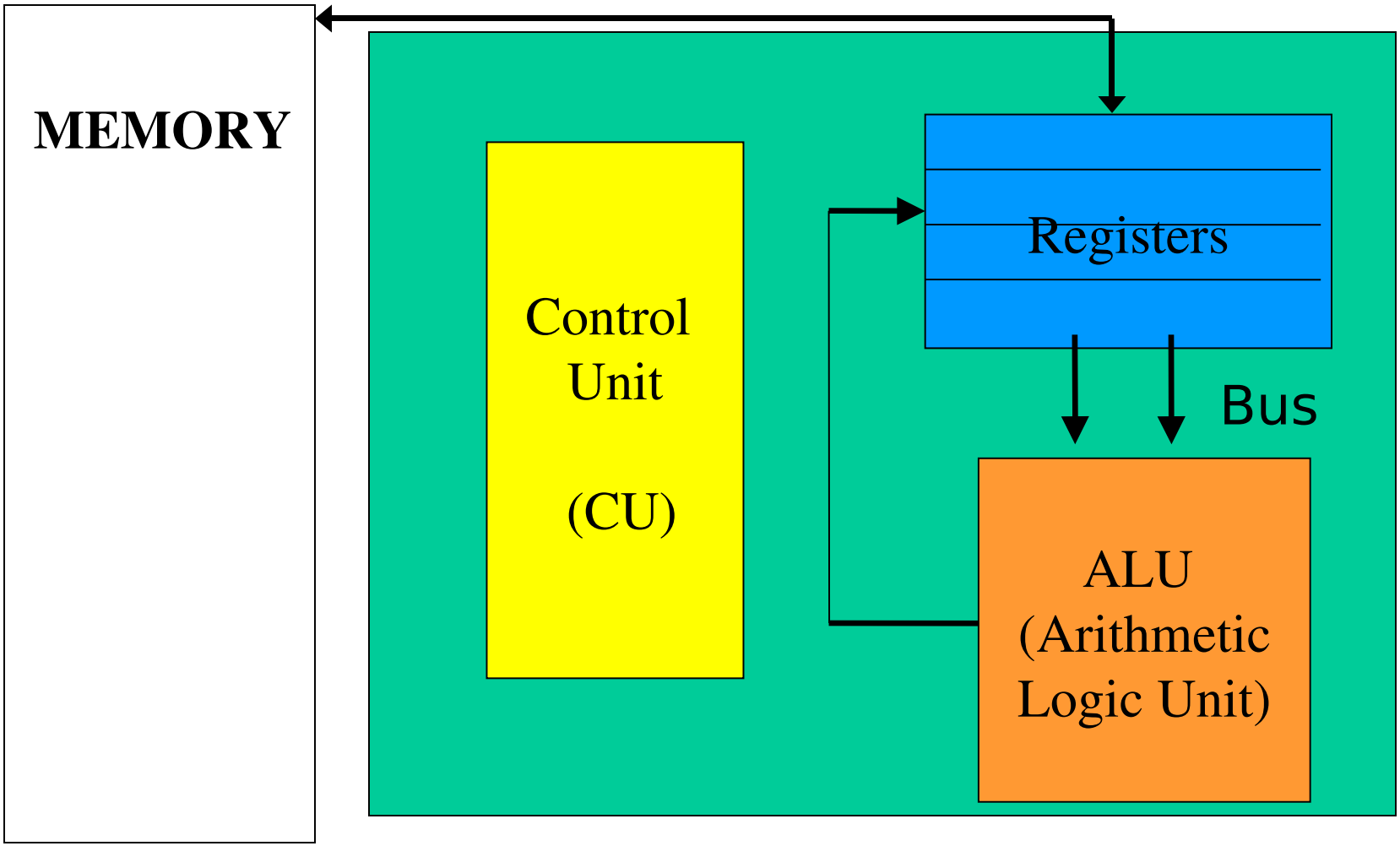
- ALU – arithmetic and Logic Unit

  performs operations on data (e.g. add, subtract, AND, OR)
- Control Unit – oversees functions of CPU
- Registers – memory locations built into the CPU for current data

# CPU – Central Processing Unit



# Memory and the CPU



# What is stored in Memory?

- Instructions
- Data

Example of instructions:

ADD R0, R2, R1

SUB MEM7, MEM1, MEM9

# Executing an Instruction

Example: SUB MEM7, MEM1, MEM9

3. Copy the value from MEM1 to a register (say R1).
4. Copy the value from MEM9 to a register (say R2).
5. Tell the ALU to subtract R1-R2.
6. Put the sum into location 7 in memory.

# Memory vs. Registers

- Registers hold the values that the ALU is currently working with. Registers are accessed very fast.
- Memory is larger (holds more data), is more permanent than registers, and slower.

# Control Unit

1. Fetch instruction from memory

3. Decode instruction

5. Execute instruction

# 1: Fetch instruction from memory

Copy the next instruction into a special register call the instruction register.

How do we know which is the next instruction?

Its address is in a special register called the program counter.

# 1: Fetch instruction from Memory

Example:

Program Counter (PC) has the value 10.

The control unit fetches the instruction sitting at location 10 in memory, and copies it into the instruction register.

# 2: Decode the instruction

Break up the instruction into its parts:

- Operation (add, sub, etc)
- Data (copy data from memory if necessary)

# 3: Execute

Configure the ALU and the buses to carry out the instruction.

(the actual operation is executed by the ALU)

# Example of Fetch-Decode-Execute

- Say PC=10, the control unit copies an instruction from location 10 in memory into the IR.
- Decode the instruction into its parts:  SUB R0 R1 R2
- Set the ALU to subtract and set the buses to send registers R1 and R2 to the ALU

# Changing the program counter

There are two ways to change the program counter:

The normal way is that after executing an instruction, the control unit increments the Program counter

PC=PC+1

The following instruction in memory will be executed next.

# Changing the PC (#2)

A jump instruction sets the PC to a certain location in memory.

Example:  JUMP MEM87

This instruction means: jump to memory location 87 and continue executing.

PC = 87

# “High-Level” View

- A user writes a program in a high-level language (e.g. C++)
- The program is translated by a compiler into machine language
- The CPU executes the machine language instructions

  (fetch-decode-execute cycles)