

# **Computer Organization and Architecture**





# Topics to be Covered

- Basic Structure of Computers.
- Machine Instructions and Programs.
- Basic Processing Unit.
- Arithmetic.
- The memory system.
- Input/Output Organization.
- Computer Peripherals.
- Pipelining.

Topics may be changed slightly during the course.

# Books

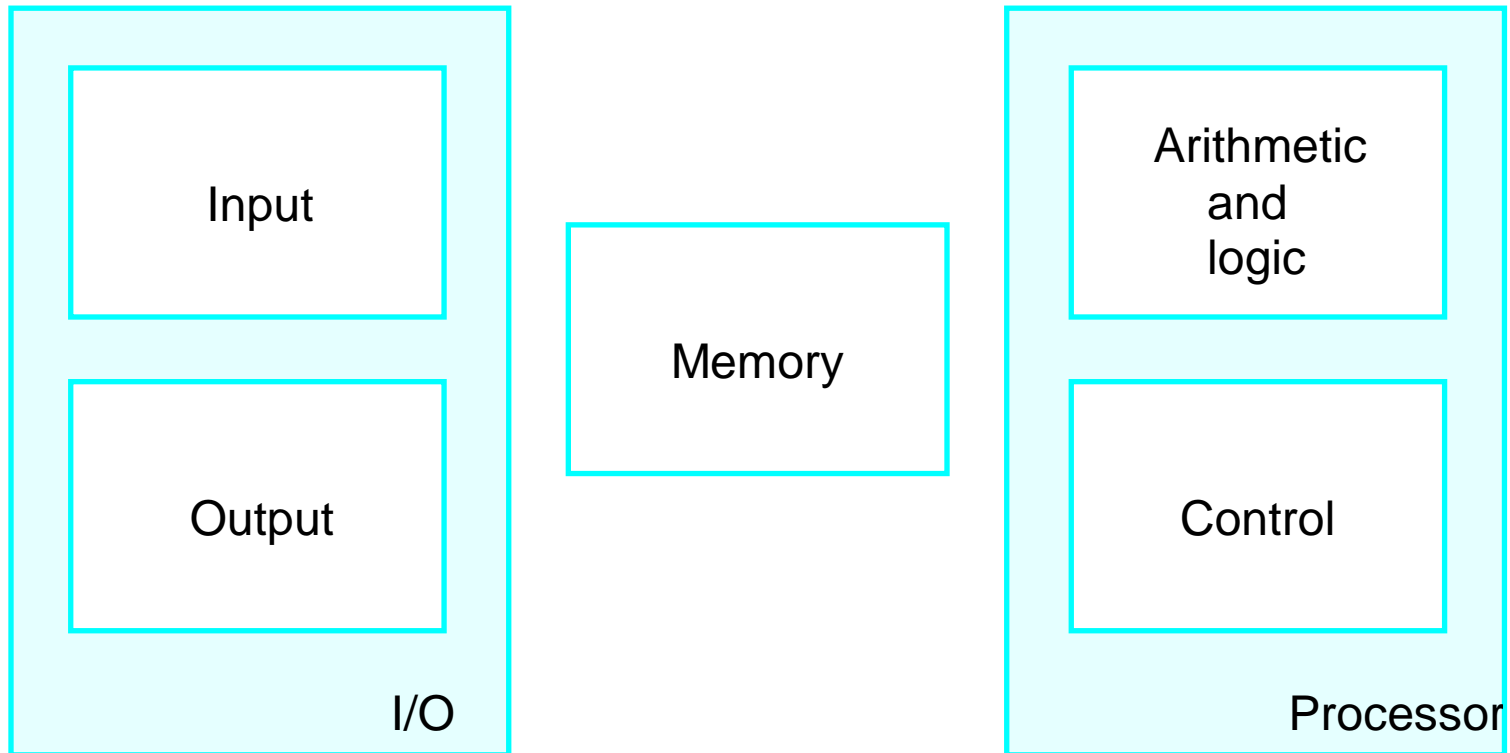
- *Computer Organization* by Hamacher, Vranesic and Zaky.
- *Computer System Architecture* by M. Morris Mano.

# Computer Architecture vs. Organization

- Architecture
  - Abstract model
  - Programmer's view of Instruction sets, addressing modes, registers etc.
  - **WHAT ?**
- Organization
  - Realization and implementation of architecture
  - Hardware
  - **HOW ?**

# Functional Units

# Functional Units



Basic functional units of a computer.



# Information Handled by a Computer

- **Instructions**

- Govern the transfer of information within a computer as well as between the computer and its I/O devices
- Specify the arithmetic and logic operations to be performed
- Program

- **Data**

- Used as operands by the instructions
- Source program

- Encoded in binary code – 0 and 1

# Input Unit

- Accepts inputs from outside world.
  - Keyboard
  - Mouse
  - Microphone
  - Joystick.

# Memory Unit

- Store programs and data
- Two classes of storage
  - Primary storage (main memory)
    - ❖ Fast, volatile.
    - ❖ Programs must be stored in memory while they are being executed
    - ❖ Large number of semiconductor storage cells
    - ❖ Cells are processed (read or written) in groups called **words**
    - ❖ Address of words
    - ❖ RAM and memory access time
    - ❖ Memory hierarchy – cache, main memory
  - Secondary storage – larger and cheaper

# Arithmetic and Logic Unit (ALU)

- Most computer operations are executed in ALU of the processor.
- Load the operands into memory – bring them to the processor – perform operation in ALU – store the result back to memory or retain in the processor.
- Registers
- Fast control of ALU

# Output Unit

- Sends the processed results to the outside world.
  - Speaker
  - Monitor
  - Printer

# Control Unit

- All computer operations are controlled by the control unit.
- The timing signals that govern the I/O transfers are also generated by the control unit.
- Control unit is usually distributed throughout the machine instead of standing alone.
- **Operations of a computer:**
  - Accept information in the form of programs and data through an input unit and store it in the memory
  - Fetch the information stored in the memory, under program control, into an ALU, where the information is processed
  - Output the processed information through an output unit
  - Control all activities inside the machine through a control unit

# Basic Operational Concepts

# A Typical Instruction

- **Add LOCA, R0**
- Add the operand at memory location LOCA to the operand in a register R0 in the processor.
- Place the sum into register R0.
- The original contents of LOCA are preserved.
- The original contents of R0 is overwritten.
- Instruction is fetched from the memory into the processor – the operand at LOCA is fetched and added to the contents of R0 – the resulting sum is stored in register R0.



# Separate Memory Access and ALU Operation

- Load LOCA, R1
- Add R1, R0
- Whose contents will be overwritten?

# Connection Between the Processor and the Memory

# Registers

- Instruction register (IR)
- Program counter (PC)
- General-purpose register ( $R_0 - R_{n-1}$ )
- Memory address register (MAR)
- Memory data register (MDR)

# Typical Operating Steps

- Programs reside in the memory through input devices
- PC is set to point to the first instruction
- The contents of PC are transferred to MAR
- A Read signal is sent to the memory
- The first instruction is read out and loaded into MDR
- The contents of MDR are transferred to IR
- Decode and execute the instruction

# Typical Operating Steps (Cont')

- Get operands for ALU
  - General-purpose register
  - Memory (address to MAR – Read – MDR to ALU)
- Perform operation in ALU
- Store the result back
  - To general-purpose register
  - To memory (address to MAR, result to MDR – Write)
- During the execution, PC is incremented to the next instruction

# Interrupt

- Normal execution of programs may be preempted if some device requires urgent servicing.
- The normal execution of the current program must be interrupted – the device raises an *interrupt* signal.
- Interrupt-service routine
- Current system information backup and restore (PC, general-purpose registers, control information, specific information)

# Bus Structures

- There are many ways to connect different parts inside a computer together.
- A group of lines that serves as a connecting path for several devices is called a *bus*.
- Address/data/control

# Bus Structure

- Single-bus



# Speed Issue

- Different devices have different transfer/operate speed.
- If the speed of bus is bounded by the slowest device connected to it, the efficiency will be very low.
- How to solve this?
- A common approach – use buffers.