Tishk International University
Engineering Faculty
Computer Engineering Department
Computer Organization CMPE 352/A
3rd Grade - Spring Semester 2019-2020
Lecture 3_Part1

Computer Organization:
Top Level View of Computer Function and Interconnection

Dr. Rand Basil Alhashimie

rand.basil@tiu.edu.iq

Lecture Objectives

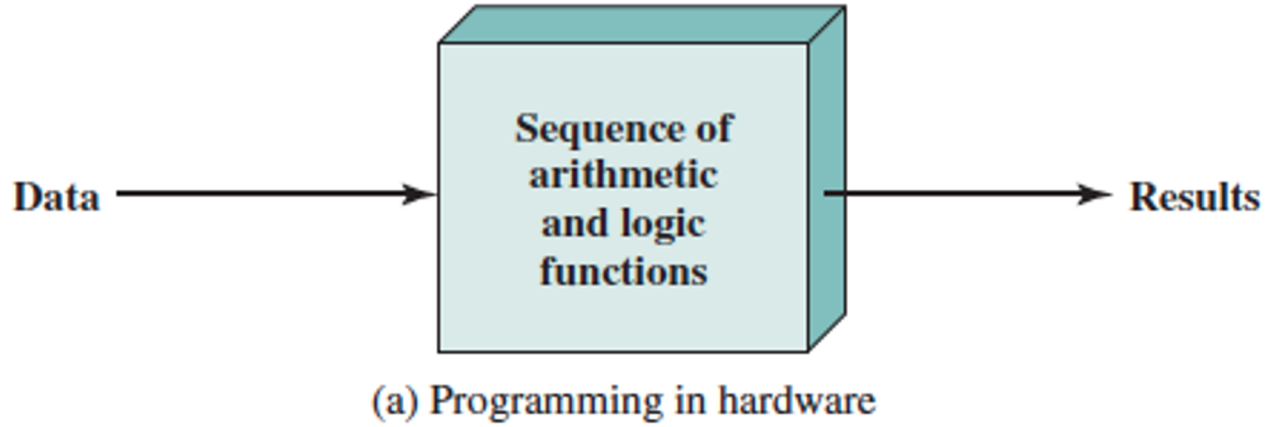
After studying this lecture, you should be able to:

- Understand the basic elements of an instruction cycle and the role of interrupts.
- Describe the concept of interconnection within a computer system.
- Understand the difference between synchronous and asynchronous bus timing.

Lecture Outline

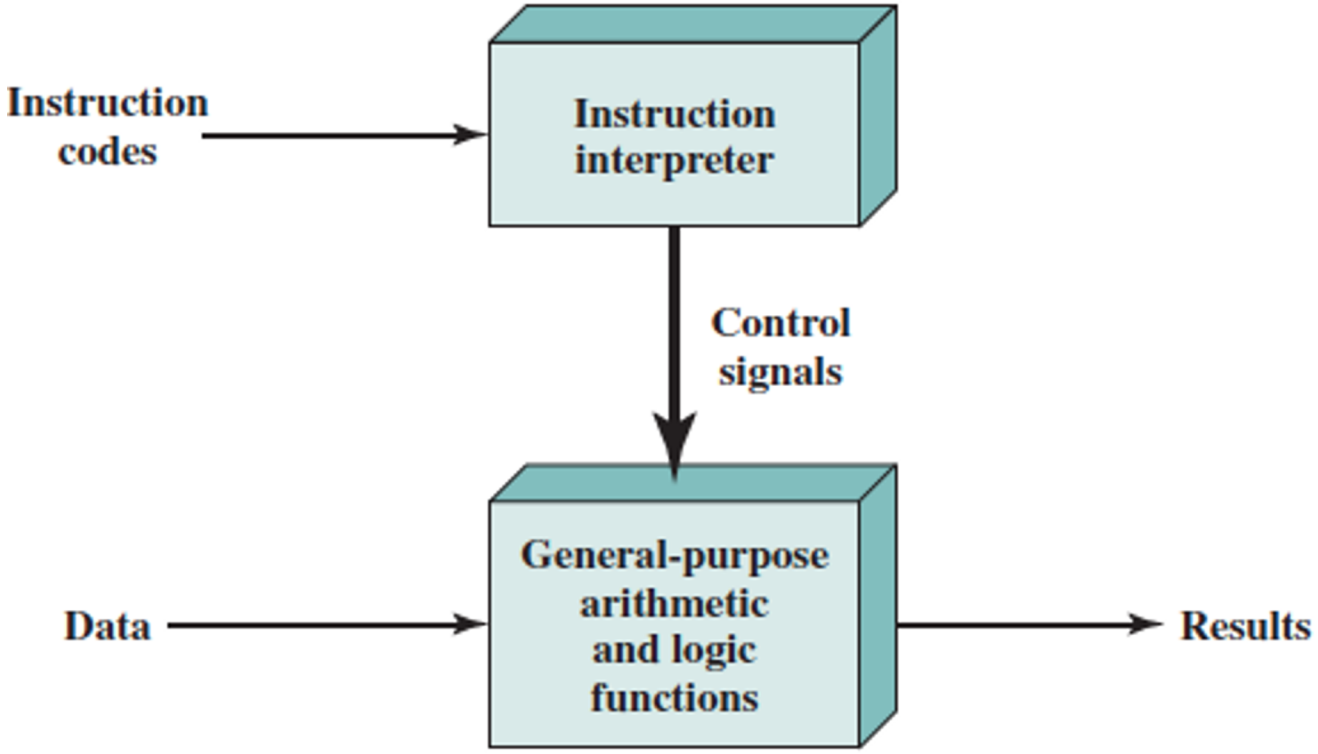
- Program concept
- Computer Top level
  - CPU
  - Memory
  - I/O interconnections

Program Concept



(a) Programming in hardware

Hardware and Software Approaches



(b) Programming in software

Program Concept

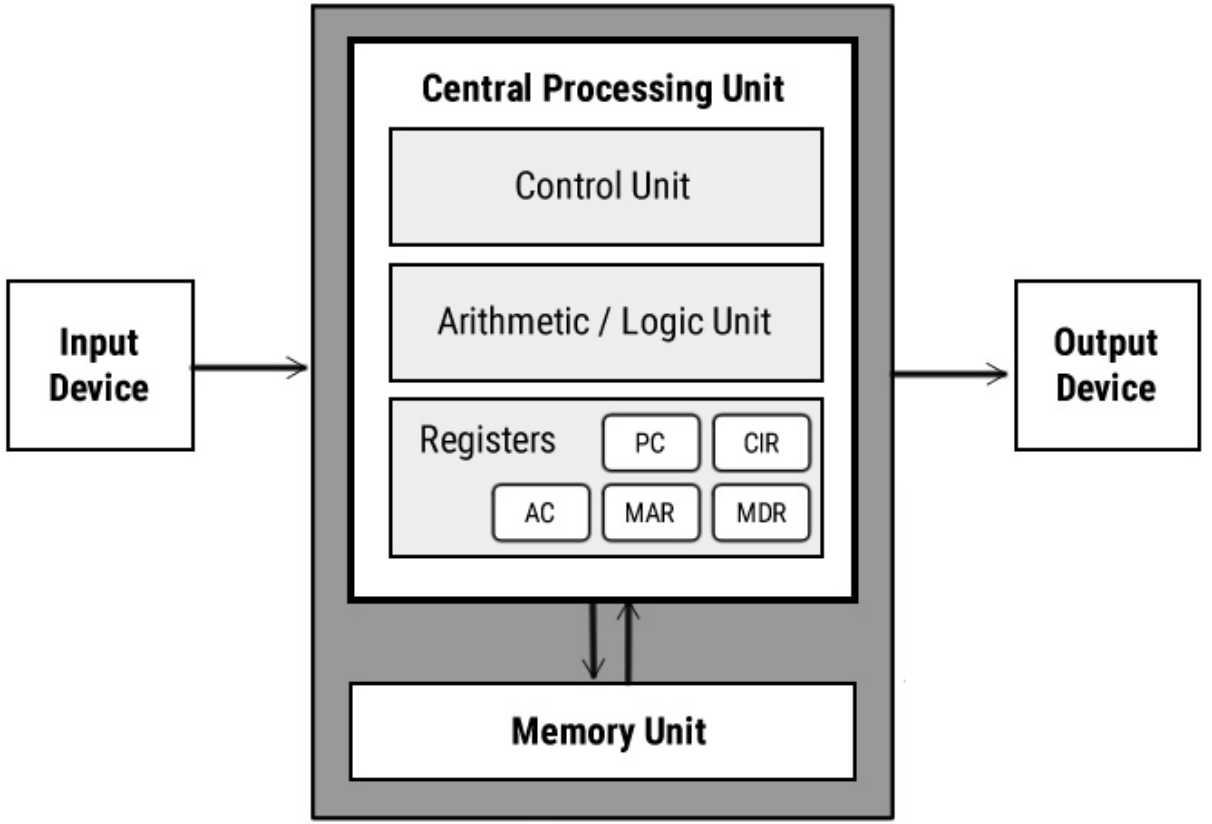
- Hardwired systems are inflexible
- General purpose hardware can do different tasks, given correct control signals.
- Instead of re-wiring, supply a new set of control signals

What is a program?

- A sequence of steps
- For each step, an arithmetic or logical operation is done
- For each operation:
  - a different set of control signals is needed
  - a unique code is provided (e.g. ADD, MOVE)

Von Neumann Architecture

- Von Neumann architecture was first published by John von Neumann in 1945.
- His computer architecture design consists of a **Control Unit, Arithmetic and Logic Unit (ALU), Memory Unit, Registers and Inputs/Outputs**.
- Von Neumann architecture is based on the stored-program computer concept, where instruction data and program data are stored in the same memory. This design is still used in most computers produced today.



Von Neumann Architecture

- Data and instructions are stored in a single read–write memory.
- The contents of this memory are addressable by location, without regard to the type of data contained there.
- Execution occurs in a sequential fashion (unless explicitly modified) from one instruction to the next.
- A particular set of hardware will perform various functions on data depending on control signals applied to the hardware.

Von Neumann Architecture

How shall control signals be supplied?

- The entire program is actually a sequence of steps. At each step, some arithmetic or logical operation is performed on some data.
- Programming is now much easier. Instead of rewiring the hardware for each new program, all we need to do is provide a new sequence of codes.
- Each code is, in effect, an instruction, and part of the hardware interprets each instruction and generates control signals.
- To distinguish this new method of programming, a sequence of codes or instructions is called *software*.

Computer Function: Top Level View

- Computer consists of:
  - CPU
  - Memory
  - I/O interconnections
- These components are connected to achieve the main function of the computer, which is to execute program.

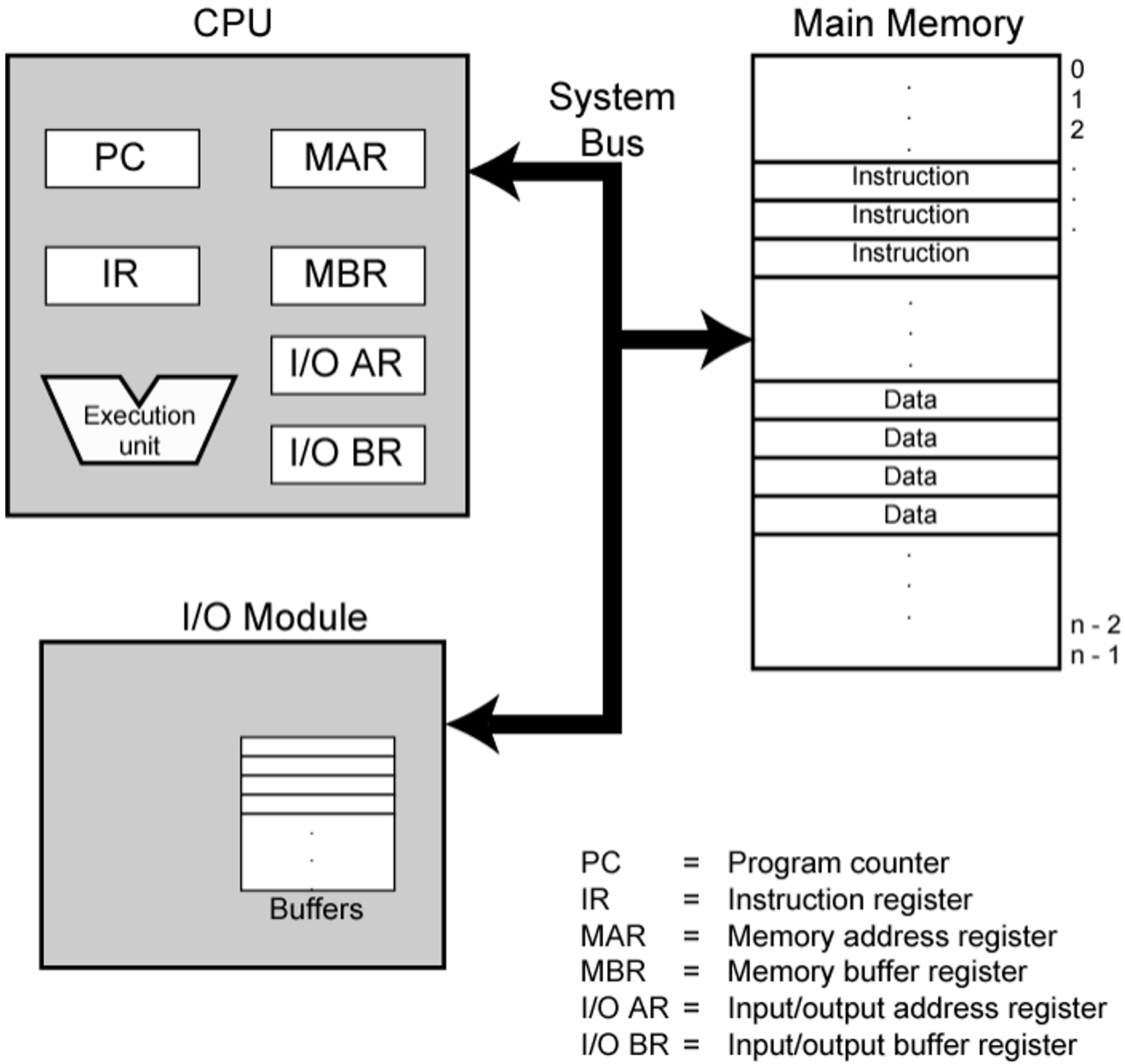


Figure 2

Components

- The Control Unit and the Arithmetic and Logic Unit constitute the Central Processing Unit.
- Data and instructions need to get into the system and results out
  - Input/output
- Temporary storage of code and results is needed
  - Main memory

I/O Components

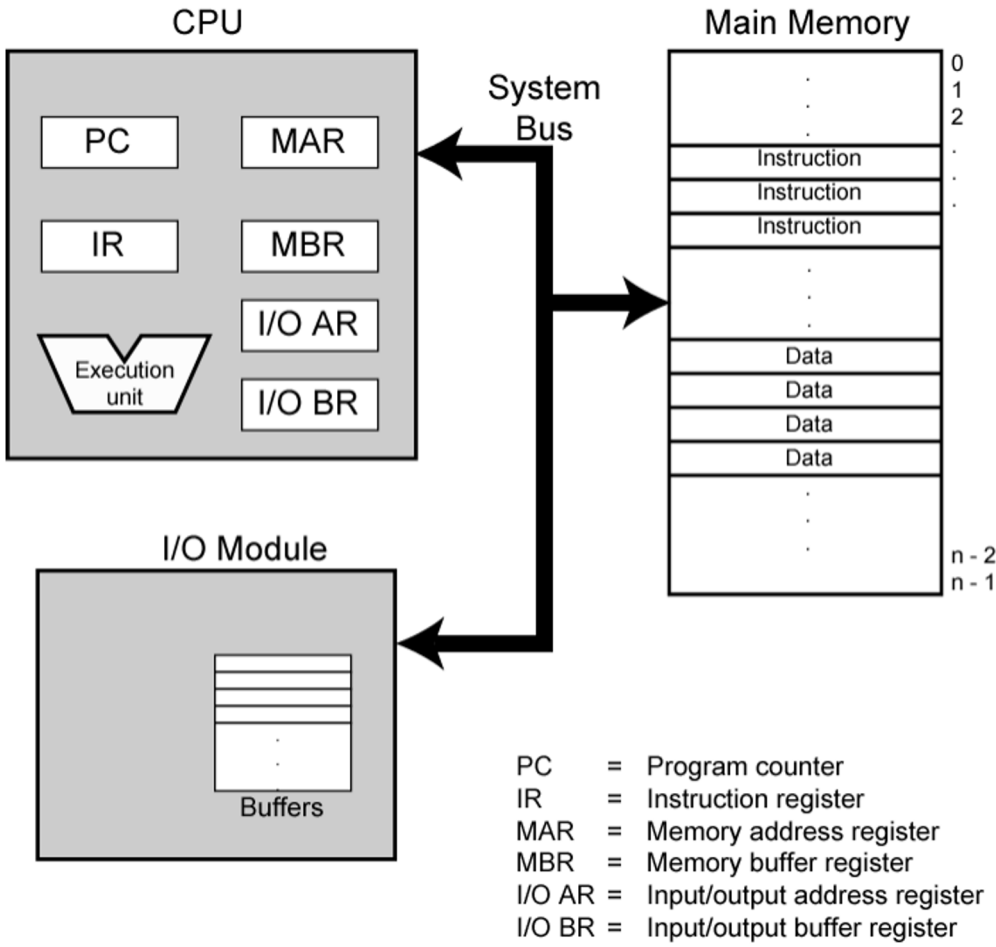
- Data and instructions must be put into the system. For this we need some sort of input module.
- This module contains basic components for accepting data and instructions in some form and converting them into an internal form of signals usable by the system.
- A means of reporting results is needed, and this is in the form of an output module. Taken together, these are referred to as *I/O components*.

Main Memory (Temporary Storage)

- An input device will bring instructions and data in sequentially. But a program is not invariably executed sequentially; it may jump around (e.g., the IAS jump instruction).
- Similarly, operations on data may require access to more than just one element at a time in a predetermined sequence. Thus, there must be a place to store temporarily both instructions and data.
- That module is called *memory*, or *main memory*, to distinguish it from external storage or peripheral devices. Von Neumann pointed out that the same memory could be used to store both instructions and data.

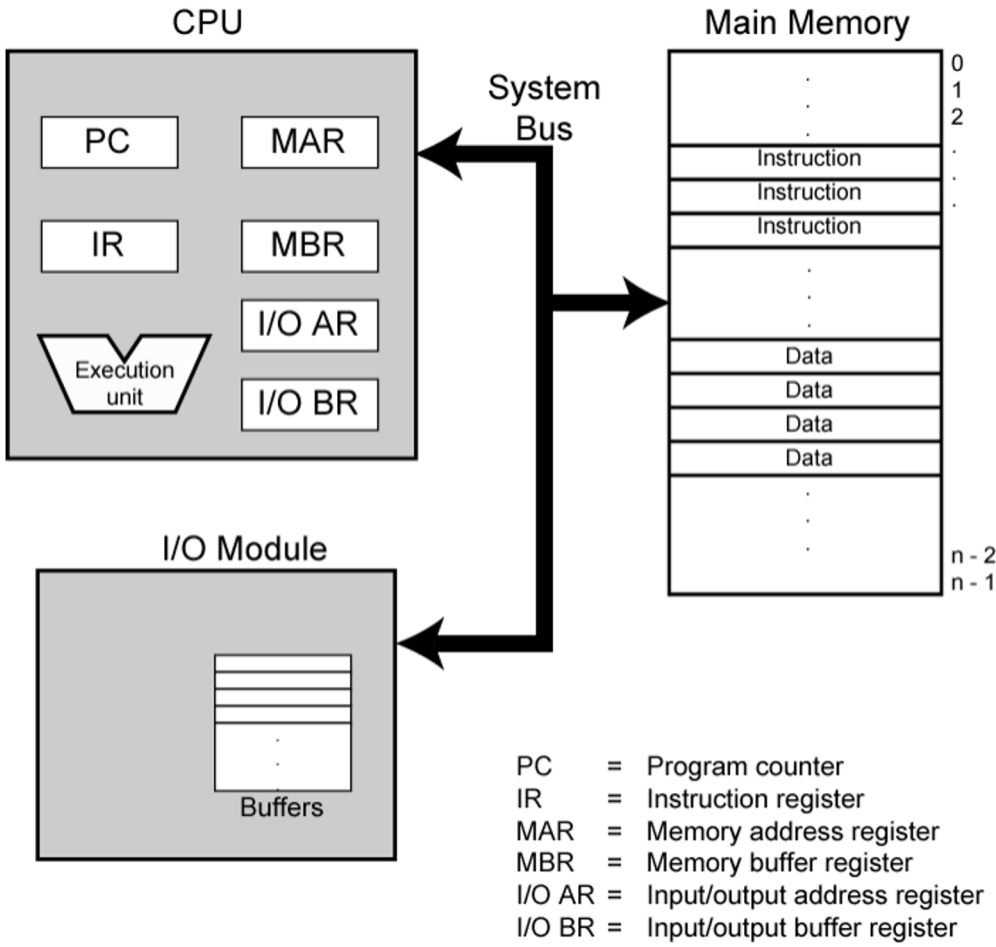
CPU Registers

- MAR (Memory Address Register)
- MBR (Memory Buffer Register)
- PC (Program Counter)
- IR (Instruction Register)
- AC (Accumulator-Temporary Register)
- I/O AR (Input-Output Address Register)
- I/O BR (Input-Output Buffer Register)



Memory Locations and I/O

- A memory module consists of a set of locations, defined by sequentially numbered addresses.
- Each location contains a binary number that can be interpreted as either an instruction or data.
- An I/O module transfers data from external devices to CPU and memory, and vice versa. It contains internal buffers for temporarily holding these data until they can be sent on.



MAR and MBR (CPU Data Exchange)

- The CPU exchanges data with memory. For this purpose, it typically makes use of two internal (to the CPU) registers:
  - **Memory address register (MAR)**, which specifies the address in memory for the next read or write
  - **Memory buffer register (MBR)**, which contains the data to be written into memory or receives the data read from memory.

I/O Address and Buffer Registers

- Similarly, an I/O address register (I/O AR) specifies a particular I/O device.
- An I/O buffer (I/O BR) register is used for the exchange of data between an I/O module and the CPU.

End of Part 1