Computer Organization & Assembly Language

Lecture 1

(Course Introduction)

Course Introduction

- Instructor: Sadia Zaman Mishu
- e-mail: sadia.cse09@gmail.com

- Credit Hours: 3

Course Introduction

- **Text Book :**

Assembly Language Programming and Organization of the IBM PC, Ytha Yu and Charles Marut

Lecture Outline

- Part I: Computer Organization
- Part II: Assembly Language

Part I: Computer organization

- Main hardware components and their relation to the software.
- What the computer does when it executes an instruction.

Topics to be covered

- **Part I: Topics to be covered**
- The Components of a Microcomputer System
  I. Memory
  II. The CPU
  III. I/O Ports
- Instruction Cycle
- I/O Devices
- Programming Languages

Microcomputer Systems:

- I/O devices are also called peripherals.
- Integrated circuit , also known as chips , digital circuits.
- IC circuits are known as digital circuits because to operates on discrete voltage signals levels typically a high voltage and a low voltage represent by 0 and 1’s.
- These symbols are called binary digits or bits.
- All information processed by the computer is represented by strings of 0’s and 1’s that is by bit string.

Motherboard:

- The **motherboard** is a printed circuit board that is the foundation of a computer, located at the bottom of the computer case.
- It allocates power to the CPU, RAM, and all other computer hardware components.
- Most importantly, the motherboard allows hardware components to communicate with one another.

Bytes and Words

- Information processed by the computer is stored in its memory.
- The memory circuits are usually organized into groups that can store eight bits of data.
- A string of eight bits is called a **byte**.
- Each **memory byte** is identified by a number that is called **address**.

Bytes and Words

- The data stored in a memory byte are called its **contents**.
- The address of a memory byte is fixed and is different from the address of any other memory byte in the computer.
- The contents of a memory byte are not unique and are subject to change, because they denote the data currently being stored.

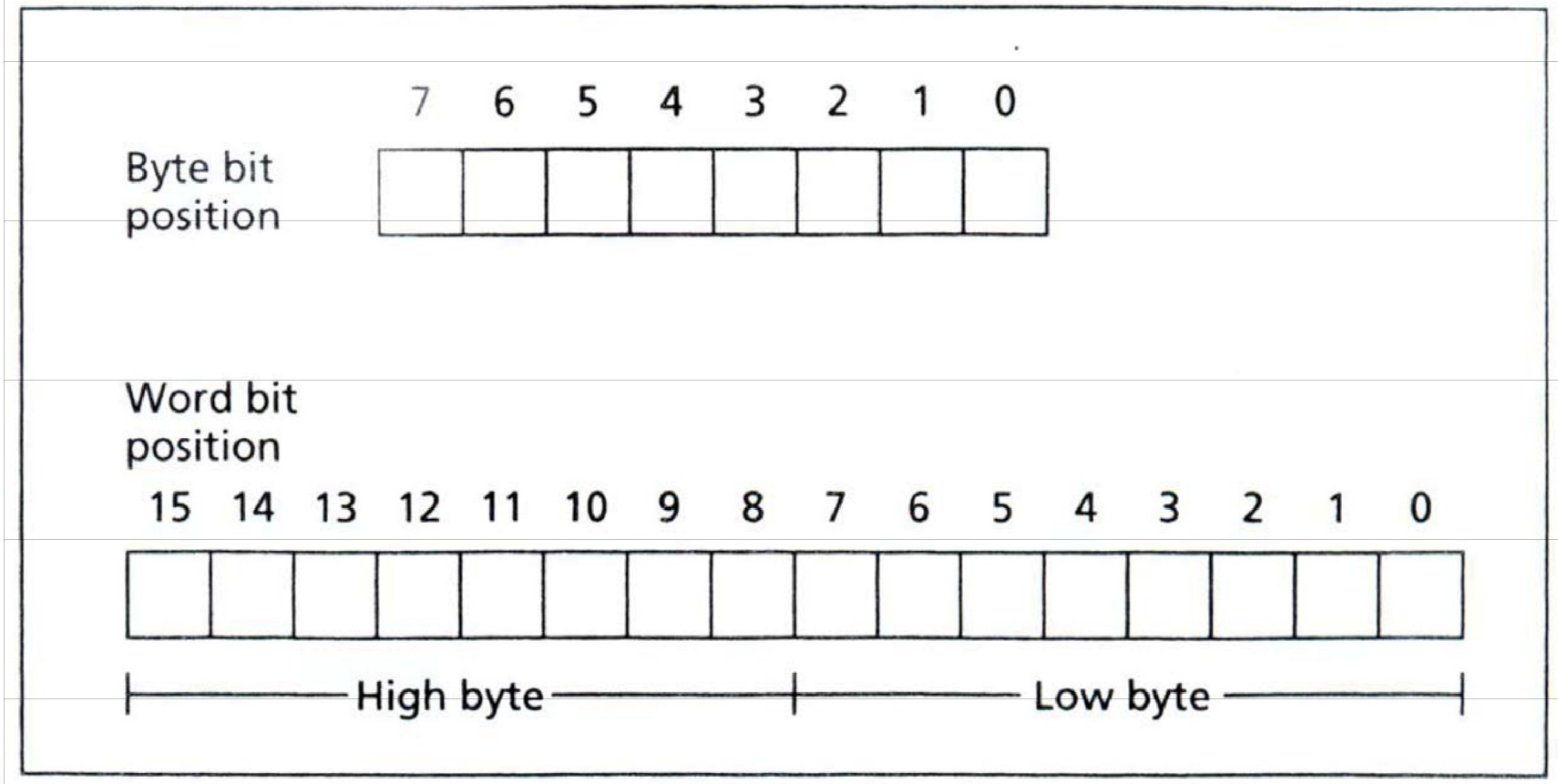
Memory Represented as Bytes

Address	Contents
.	
.	
.	
.	. . . .
7	0 0 1 0 1 1 0 1
6	1 1 0 0 1 1 1 0
5	0 0 0 0 1 1 0 1
4	1 1 1 0 1 1 0 1
3	0 0 0 0 0 0 0 0
2	1 1 1 1 1 1 1 1
1	0 1 0 1 1 1 1 0
0	0 1 1 0 0 0 0 1

Bit Position

- The positions are numbered from right to left, starting with 0.
- In a word, the bits 0 to 7 form the **low byte** and the bits 8 to 15 form the **high byte**.
- For a word stored in memory, its low byte comes from the memory byte with the lower address and its high byte is from the memory byte with the higher address.

Bit Positions in a Byte and a Word



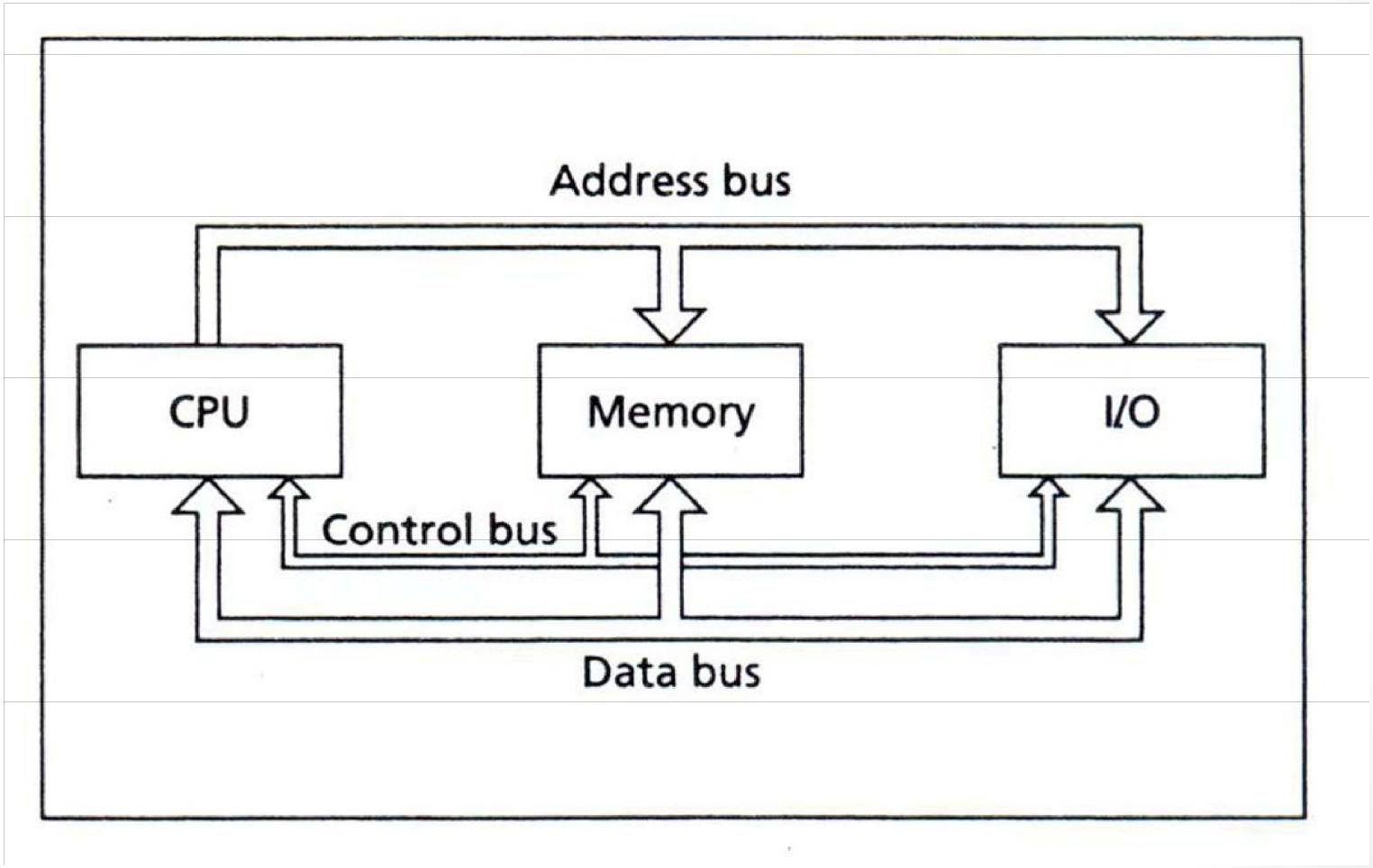
Buses

- A processor communicates with memory and I/O circuits by using signals that travel along a set of wires or connections called **buses**.
- There are three kinds of signals: address, data, and control.
- And there are three buses: **address bus**, **data bus**, and **control bus**.

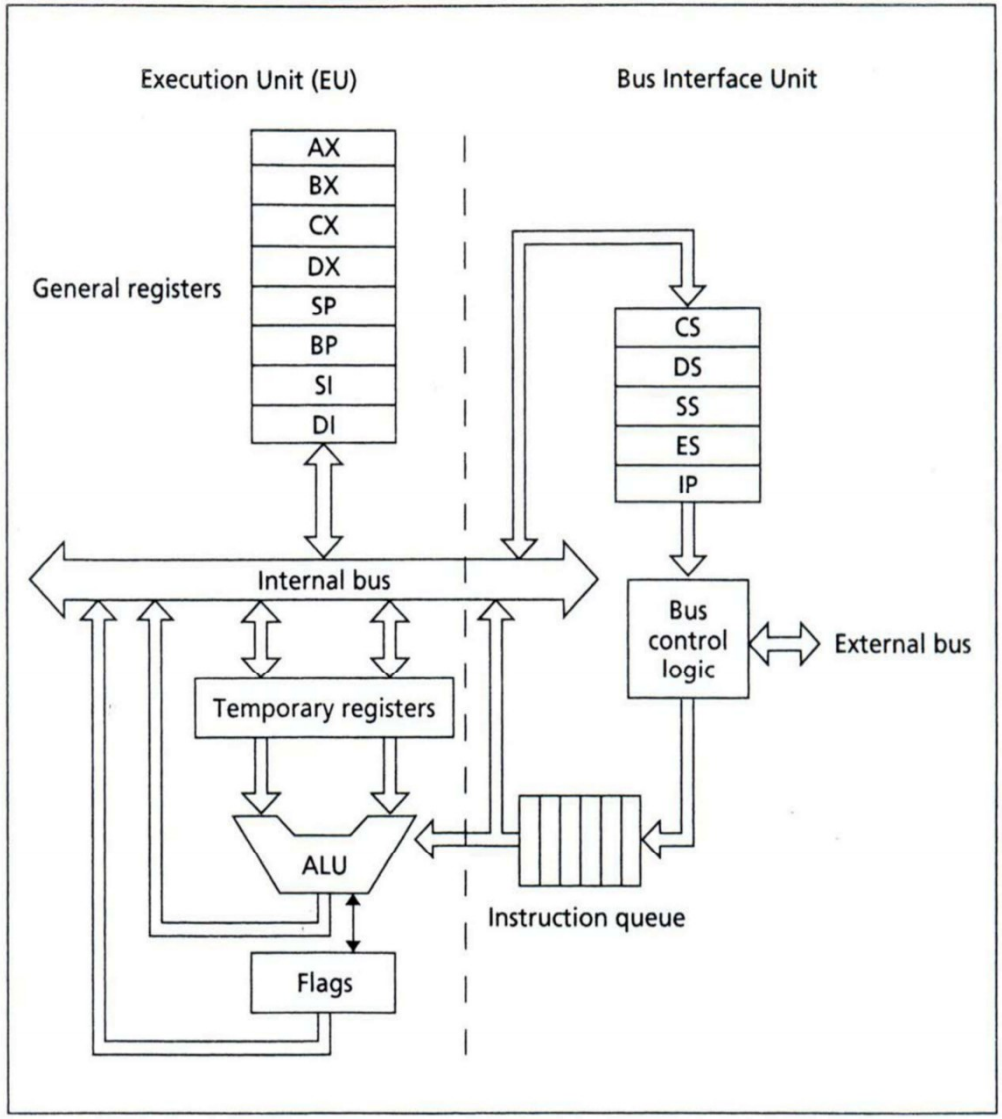
Buses:

- For example, to read the contents of a memory location, the CPU places the address of the memory location on the address bus, and it receives the data, sent by the memory circuits, on the data bus.
- A control signal is required to inform the memory to perform a read operation.
- The CPU sends the control signal on the control bus.

Bus Connections of a Microcomputer



Intel 8086 Microprocessor Organization



Intel 8086 Microprocessor Organization

- There are two main components of :
- Execution unit
- Bus interface unit

Execution Unit (EU):

- The purpose of the EU is to execute instructions.
- The **arithmetic and logic unit (ALU)** can perform arithmetic (+, -, x, /) and logic (AND, OR, NOT) operations.
- The data for the operations are stored in circuits called **registers**.
- The EU has 8 registers.
- The EU contains temporary registers for holding operands for the ALU and flag registers whose individual bits reflect the results of a computation.

Bus Interface Unit (BIU):

- The BIU facilitates communication between the EU and the memory or I/O circuits.
- The BIU is responsible for transmitting addresses, data, and control signals on the buses.
- The **instruction pointer (IP)** contains the address of the next instruction to be executed by the EU.

Instruction Prefetch:

- While the EU is executing an instruction, the BIU fetches up to six bytes of the next instruction and places them in the instruction queue.

Machine Instruction:

- The **Opcode** specifies the type of operation.
- The **Operands** are often given as memory addresses to the data to be operated on.

Fetch-Execute Cycle:

Fetch

- Fetch an instruction from memory.
- Decode the instruction to determine the operation.
- Fetch data from memory if necessary. **Execute**
- Perform the operation on the data.
- Store the result in memory if needed.

Programming Languages

- The operations of the computer's hardware are controlled by its software.
- When the computer is on, it is always in the process of executing instructions.

Machine Language:

- The CPU can only execute machine language instructions.
- They are bit strings.
- **Machine Instruction Operation**
- 10100001 00000000 00000000 Fetch the contents of memory word 0 and put it in register AX.
- 00000101 00000100 00000000 Add 4 to AX.
- 10100011 00000000 00000000 Store the contents of
- AX in memory word 0.

Assembly Language:

- A more convenient language to use is **assembly language**.
- In assembly language, we use symbolic names to represent operations, registers, and memory locations.
- If location 0 is symbolized by A, the preceding program expressed in IBM PC assembly language would look like this:

Assembly Language:

Assembly Instruction Comment

```
MOV AX, A          ; fetch the contents of
                   ; location A and put it in
                   ; register AX
ADD AX, 4          ; add 4 to AX

MOV A, AX          ; move the contents of AX
                   ; into location A
```