Computer Architecture and Organization

The IAS Computer

Computer Component

Lecture 2

Dr.Monia Al-hobishi

The IAS Computer

In 1946, von Neumann and his colleagues began the design of a new stored program computer, referred to as the IAS computer, at the Princeton **Institute for Advanced Studies**. The IAS computer, although not completed until 1952, is the prototype of all subsequent general-purpose computers.

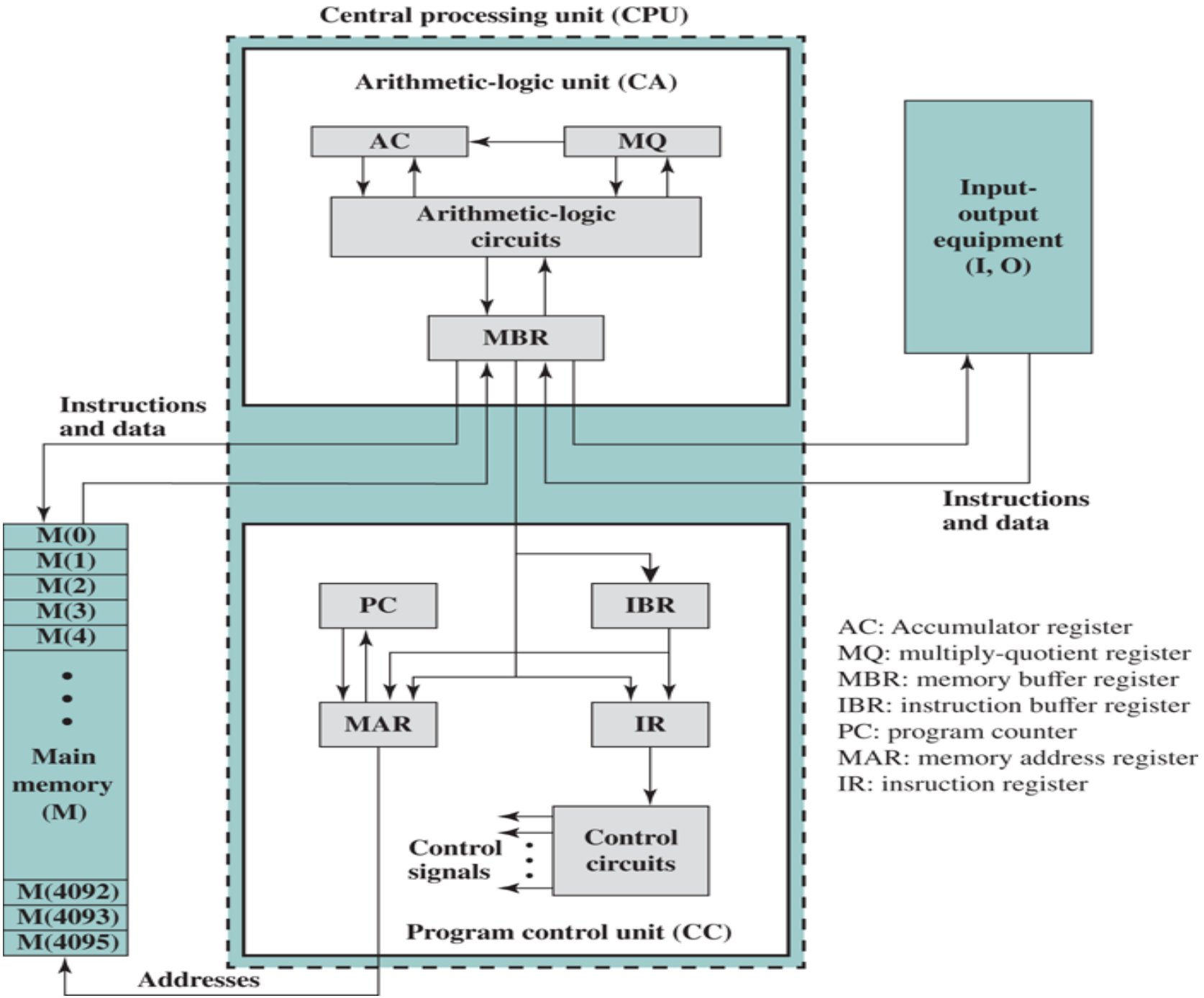
Structure of IAS Computer

It consists of :

- A **main memory** , which stores both data and instructions .

(the term instruction refers to a machine instruction that is directly interpreted and executed by the processor) .

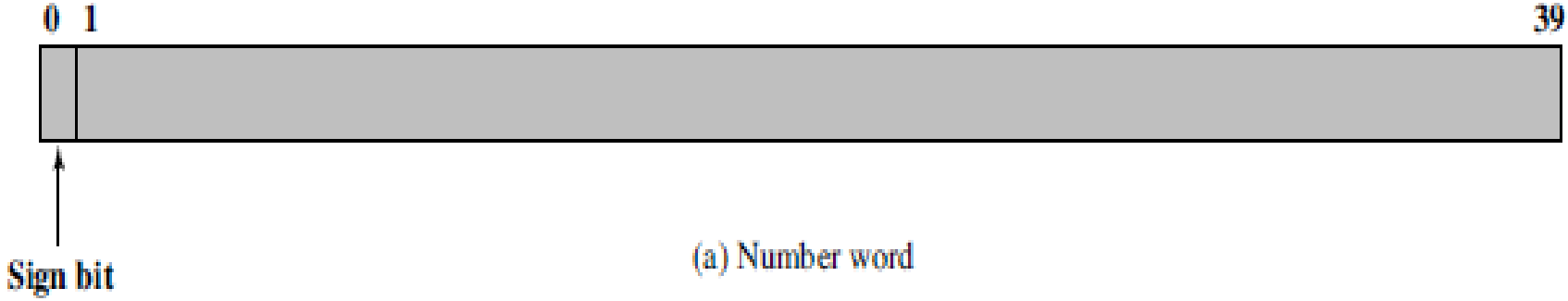
- **An arithmetic and logic unit** (ALU) capable of operating on binary data
- **A control unit** , which interprets the instructions in memory and causes them to be executed .
- **Input–output (I/O)** equipment operated by the control unit .

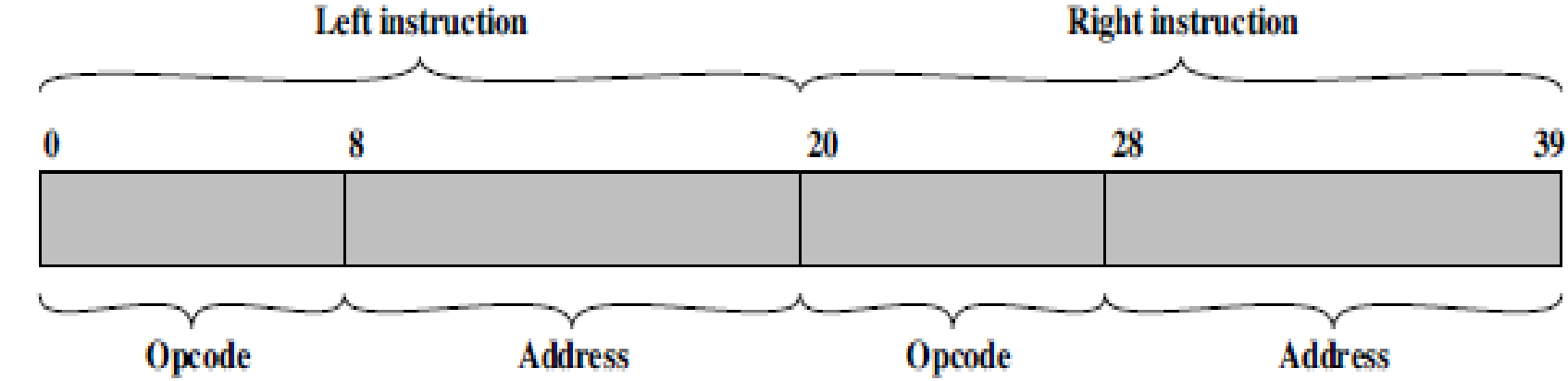

Central processing unit (CPU)
Arithmetic-logic unit (CA)
AC
MQ
Arithmetic-logic circuits
MBR
Input-output equipment (I, O)
Instructions and data
Instructions and data
M(0)
M(1)
M(2)
M(3)
M(4)
Main memory (M)
M(4092)
M(4093)
M(4095)
PC
IBR
MAR
IR
Control signals
Control circuits
Program control unit (CC)
Addresses
AC: Accumulator register
MQ: multiply-quotient register
MBR: memory buffer register
IBR: instruction buffer register
PC: program counter
MAR: memory address register
IR: insruction register

IAS Memory Formats

- The memory of the IAS consists of 1000 storage locations, called **words,** of 40 binary digits (bits).
- **Numbers** are represented in **binary form**, and each **instruction** is a **binary code** .
- A **word** may also contain two **20-bit** instructions, with each instruction consisting of an **8-bit operation code** (**opcode**) specifying the operation to be performed and a **12-bit address** designating one of the words in memory.

IAS Memory Formats





Structure of IAS Computer

- The **control unit and the ALU** contain storage locations, called **registers**, defined as follows:

  - **Memory buffer register (MBR):** Contains a word to be stored in memory or sent to the I/O unit, or is used to receive a word from memory or from the I/O unit.
  - **Memory address register (MAR):** Specifies the address in memory of the word to be written from or read into the **MBR**.
  - **Instruction register (IR):** Contains the 8-bit opcode instruction being executed.

Structure of IAS Computer

- **Program counter (PC):** Contains the address of the next instruction to be fetched from memory.

- **Instruction buffer register (IBR):** Employed to hold temporarily the right-hand instruction from a word in memory.

-**Accumulator (AC) and multiplier quotient (MQ):** Employed to hold temporarily operands and results of ALU operations.

Von Neumann architecture

Is based on three key concepts:

- Data and instructions are stored in a single read–write memory.
- The contents of this memory are addressable by location .
- Execution occurs in a sequential fashion from one instruction to the next.

Computer Components

- Computer consists of **CPU** (central processing unit), memory, and **I/O** components, with one or more modules of each type.
- These components are interconnected in some fashion to achieve the basic function of the computer, which is to execute programs.

Computer Components

we can characterize a computer system by describing :

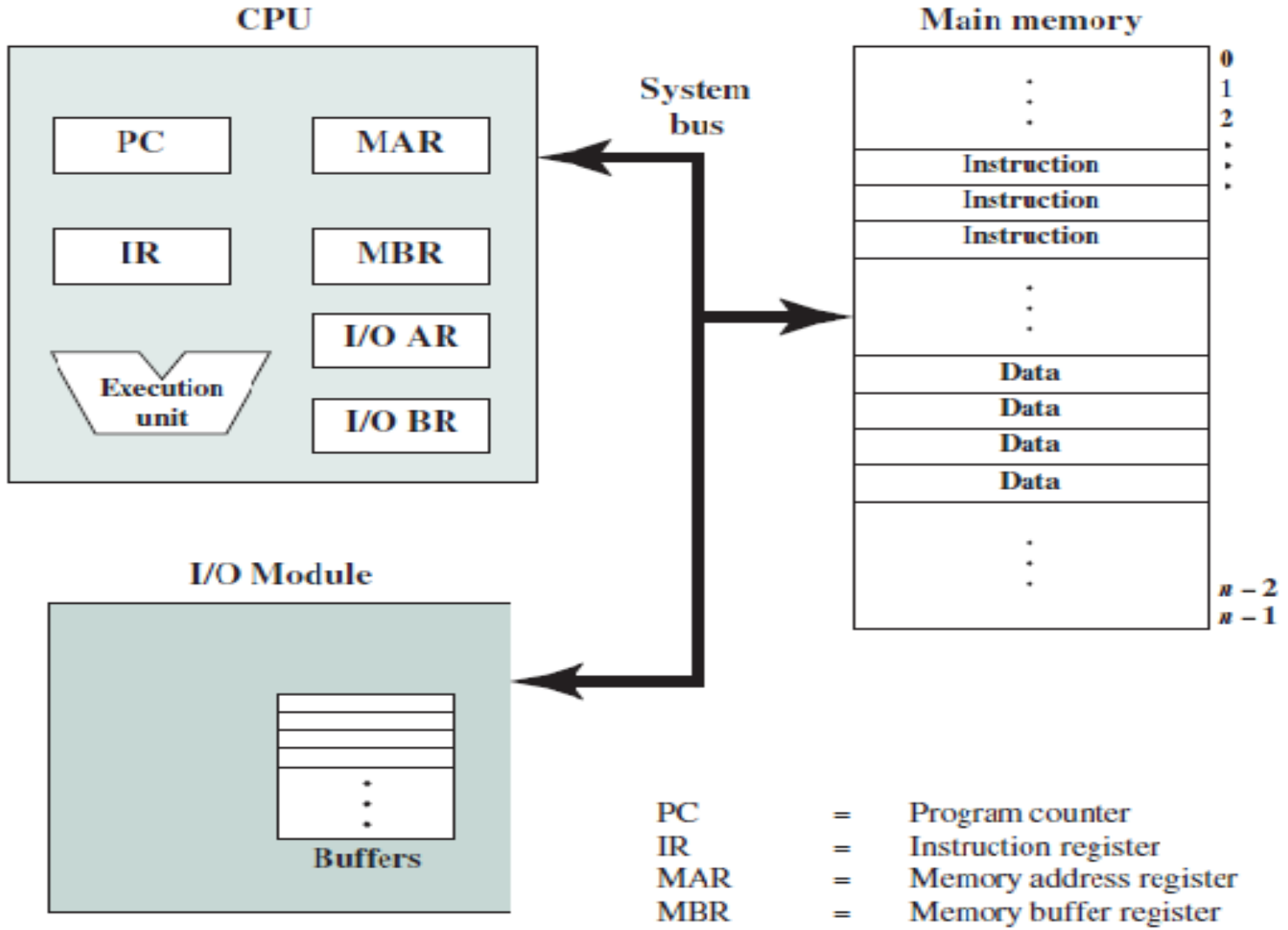
(1) - The external behavior of each component, that is, the data and control signals that it exchanges with other components .

(2) - The interconnection structure and the controls required to manage the use of the interconnection structure.

Internal Organization of a Processor

- All **CPU** design includes a set of **register**.
- **Register** is a temporary storage space on the CPU that can be accessed very quickly.
- Registers are the most important components of CPU.
- The registers used by the CPU are often termed as **Processor registers**.
- A processor register may hold an instruction, a storage address, or any data (such as bit sequence or individual characters).

Computer Components: Top-Level View



PC = Program counter
IR = Instruction register
MAR = Memory address register
MBR = Memory buffer register
I/O AR = Input/output address register
I/O BR = Input/output buffer register

CPU REGISTERS

- **Memory Address Register (MAR)-** It stores address of data or instructions to be fetched from memory.
- **Memory buffer register (MBR)-** which contains the data to be written into memory or receives the data read from memory.
- **I/O address register (I/OAR) –** specifies particular I/O device.
- **I/O buffer register (I/OBR)** is used for the exchange of data between an **I/O module** and the **CPU**.

CPU REGISTERS

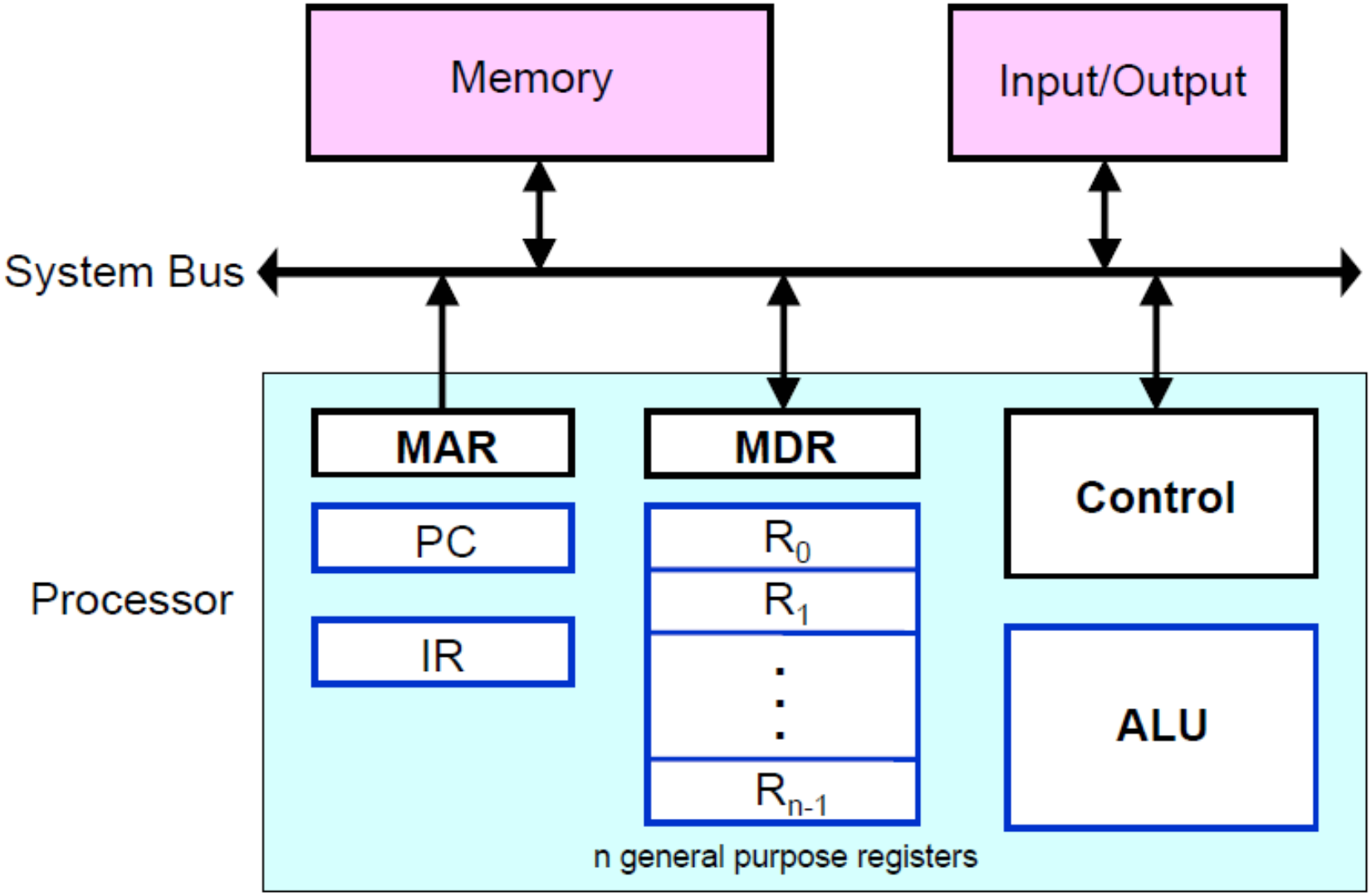
- **Instruction register(IR)** -The fetched instruction is loaded in the IR for execution.
- **Program Counter(PC) -** It counts instructions, is the register that contains the address of the next instruction to be fetched.
- A **memory module** consists of a set of locations, defined by sequentially 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. It contains internal buffers for temporarily holding these data until they can be sent on.

- Memory Data Register (MDR) - contains the data to be stored in the computer storage (e.g. RAM), or the data after a fetch from the computer storage.
- It acts like a buffer and holds anything that is copied from the memory ready for the processor to use it.
- General Purpose registers: General purpose registers are used to store data and intermediate results during program execution. Its contents can be accessed through assembly programming.

CPU REGISTERS

- The **Accumulator (AC) register** is a general purpose processing register.
- The **Temporary Register (TR)** is used for holding the temporary data during the processing.
- The **Input Registers (INR)** holds the input characters given by the user.
- The **Output Registers (OR)** holds the output after processing the input data.

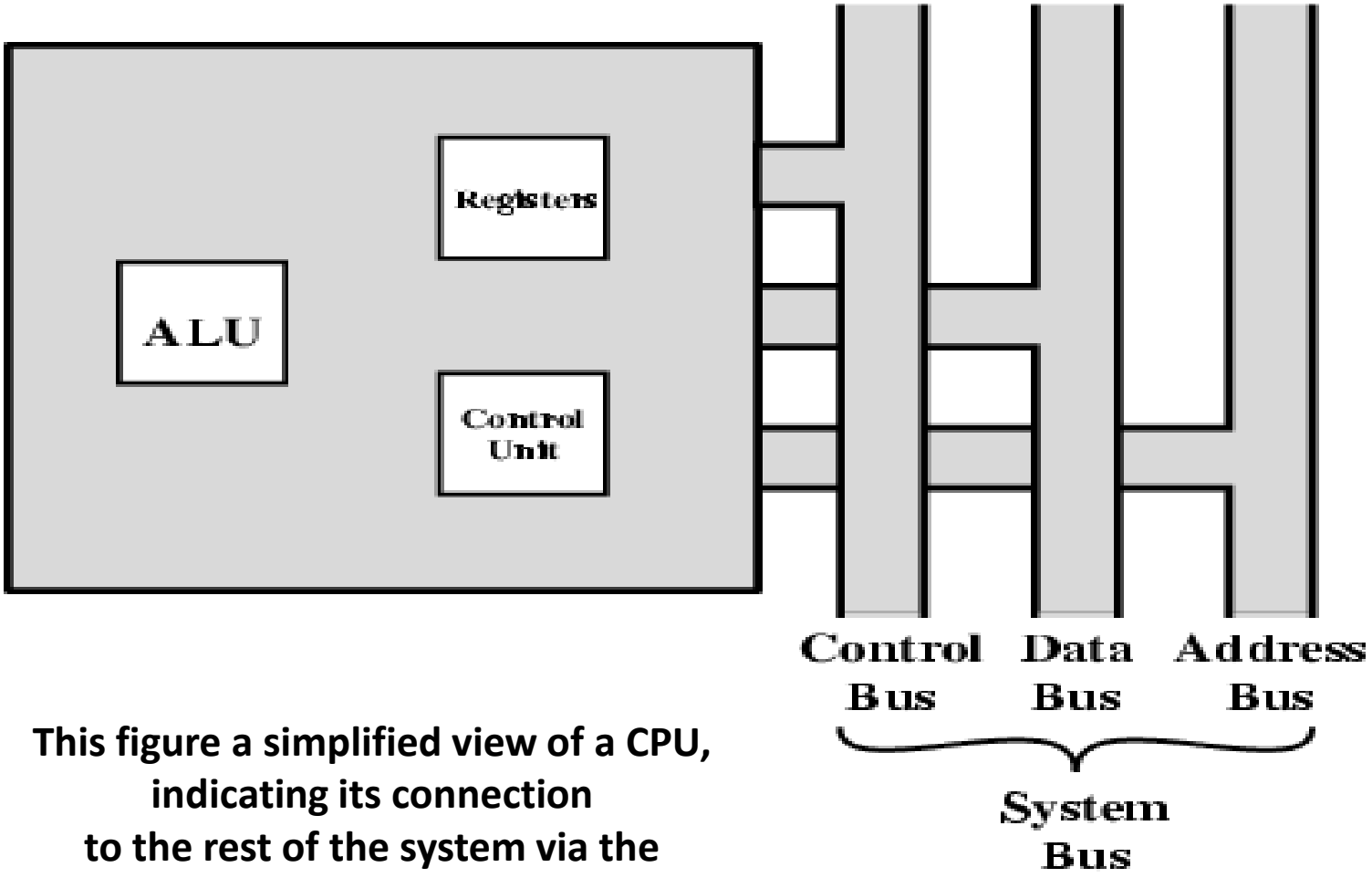
Computer Components: Top-Level View



CPU REGISTERS

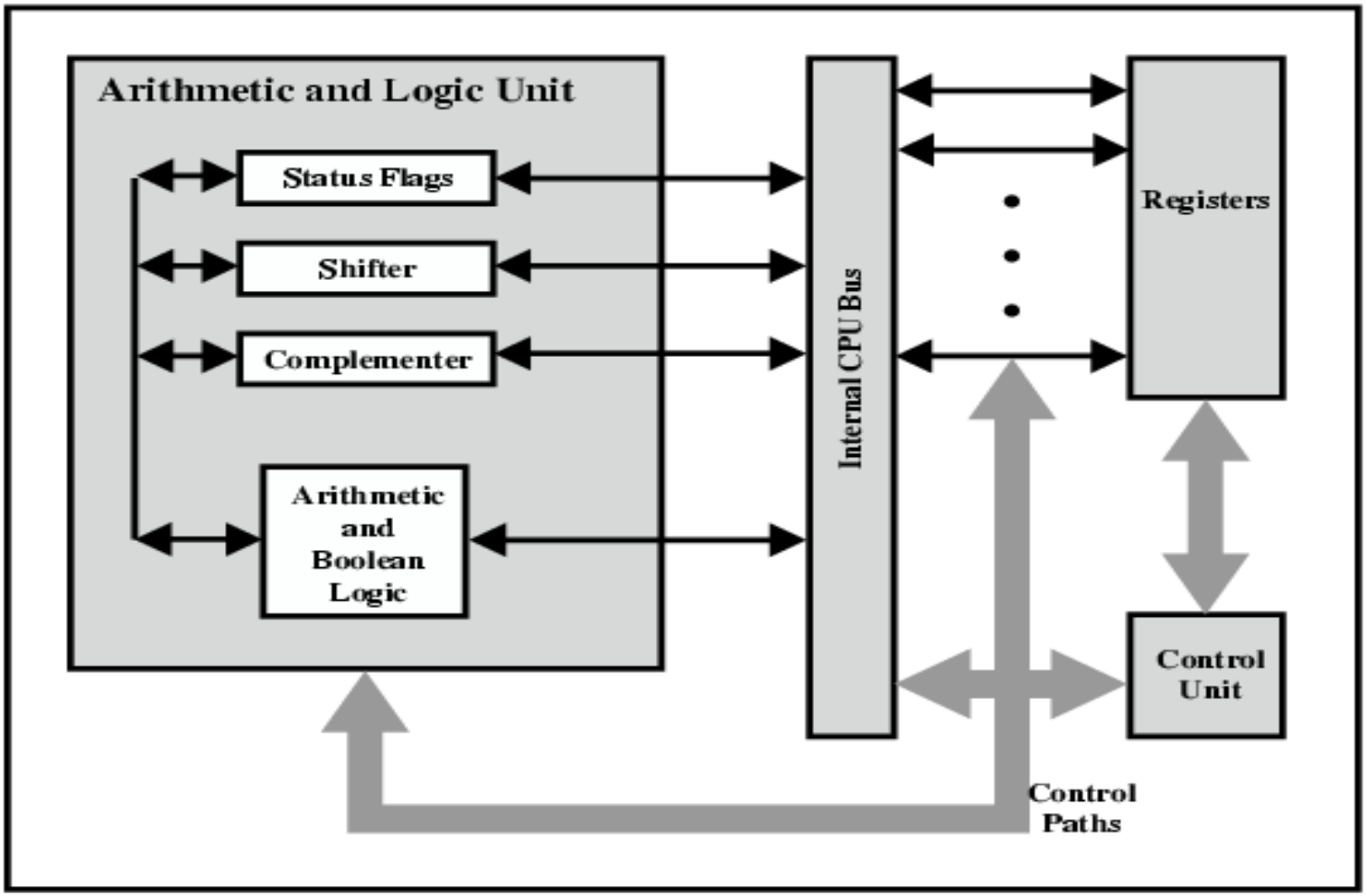
Register	Symbol	Number of bits	Function
Data register	DR	16	Holds memory operand
Address register	AR	12	Holds address for the memory
Accumulator	AC	16	Processor register
Instruction register	IR	16	Holds instruction code
Program counter	PC	12	Holds address of the instruction
Temporary register	TR	16	Holds temporary data
Input register	INPR	8	Carries input character
Output register	OUTR	8	Carries output character

CPU With System Bus



This figure a simplified view of a CPU, indicating its connection to the rest of the system via the system bus.

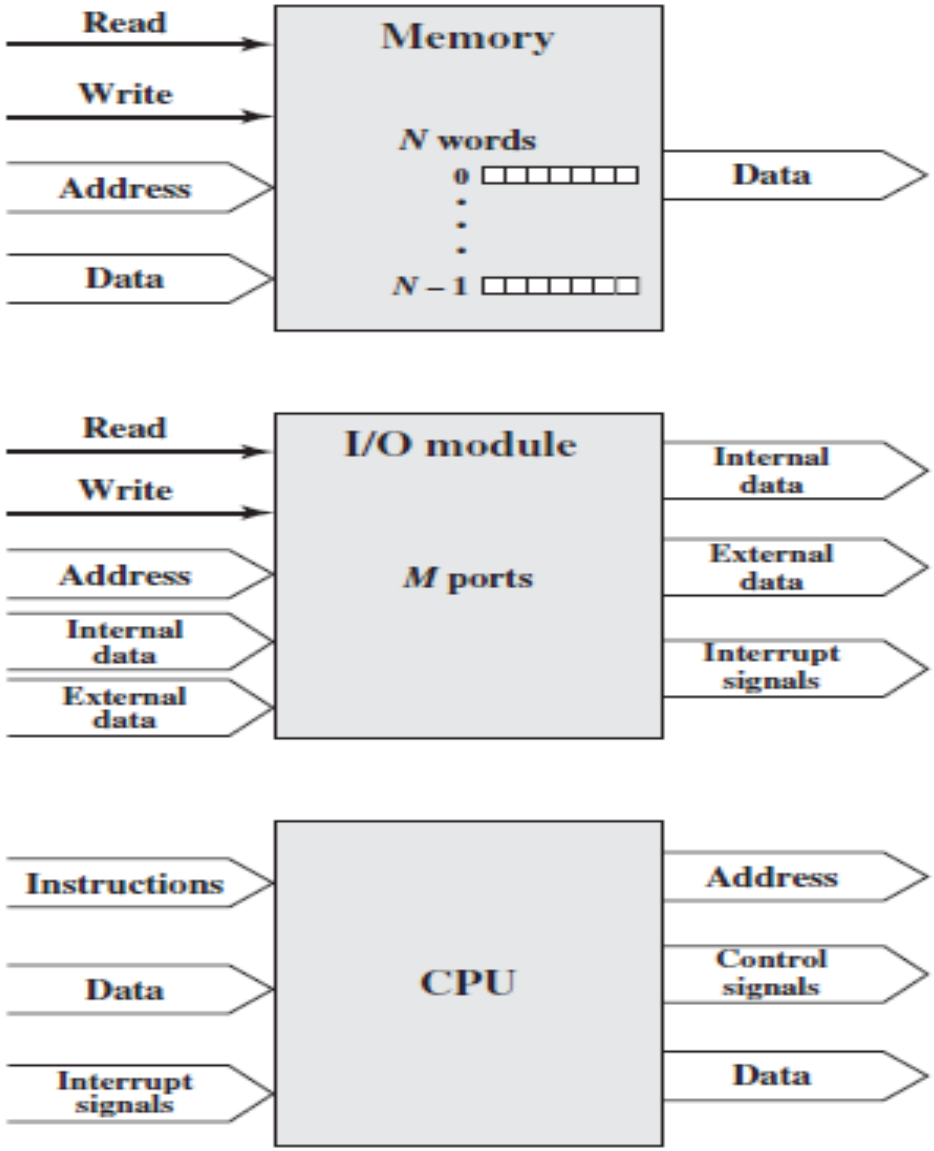
CPU Internal Structure



INTERCONNECTION STRUCTURES

- A computer consists of a set of components or modules of three basic types (processor, memory, I/O) that communicate with each other.
- The collection of paths connecting the various modules is called the **interconnection structure**.
- In Figure (**Computer Modules**) suggests the types of exchanges that are needed by indicating the major forms of input and output for each module type:

Computer Modules



INTERCONNECTION STRUCTURES

- **Memory:** a memory module will consist of N words of equal length . each word is assigned a unique numerical address (0, 1, . . . ,N – 1). A word of data can be read from or written into the memory .The nature of the operation is indicated by read and write control signals.
- **I/O module:** I/O is functionally similar to memory. There are two operations, read and write . I/O module may control more than one external device. We can refer to each of the interfaces to an external device as a port and give each a unique address (e.g., 0, 1, . . . ,M– 1).

INTERCONNECTION STRUCTURES

- **Processor:** The processor reads in instructions and data, writes out data after processing, and uses control signals to control the overall operation of the system .It also receives interrupt signals.

The **interconnection structure** must support the following types of transfers :

- **Memory to processor :** The processor reads an instruction or a unit of data from memory.
- **Processor to memory:** The processor writes a unit of data to memory.

INTERCONNECTION STRUCTURES

- **Processor to I/O :** The processor sends data to the I/O device.
- **I/O to or from memory :** I/O module is allowed to exchange data directly with memory, without going through the processor, using direct memory access (DMA) .

BUS INTERCONNECTION

- A **bus** is a communication pathways , or lines connecting two or more devices.
- A key characteristic of a bus is that it is a **shared transmission medium .**
- **Multiple devices** connect to the bus, and a signal transmitted by any one device is available for reception by all other devices attached to the bus.

If two devices transmit during the same time period, their signals will overlap and become garbled.

Thus, only one device at time can successfully transmit.

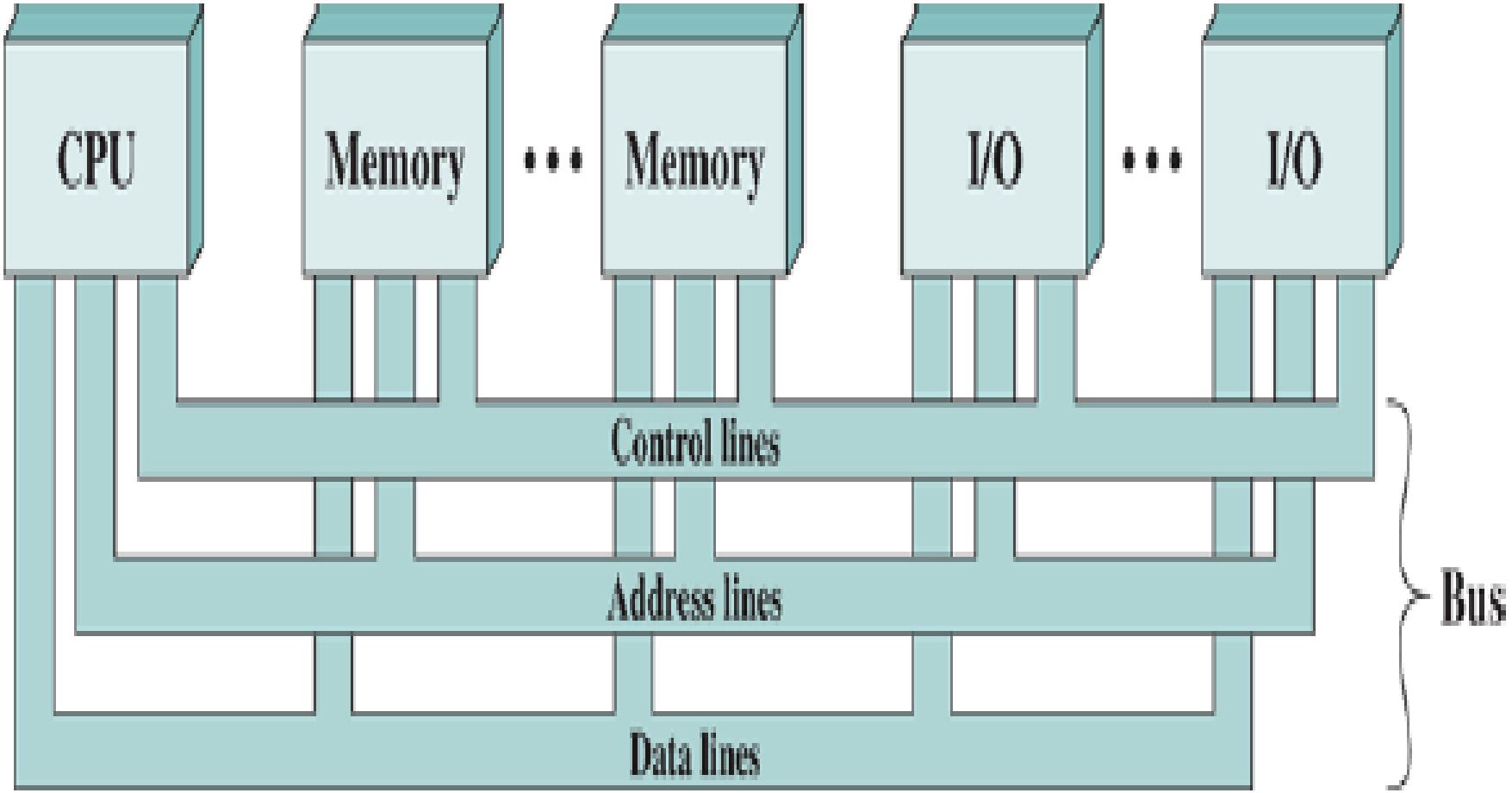
BUS INTERCONNECTION

- Each line is capable of transmitting signals representing **binary 1 and binary 0**.
- A bus that connects major computer components (**processor, memory, I/O**) is called a **system bus**.
- A **system bus** consists, typically, of from about fifty to hundreds of separate lines .

BUS INTERCONNECTION

- Each line is assigned a particular meaning or function.
- There are many different bus designs, on any bus the lines can be classified into three functional groups :**data, address, and control lines**.

Bus Interconnection Scheme



BUS INTERCONNECTION

- **Data lines** - provide a path for moving data among system modules. These lines, collectively, are called the **data bus** .The **data bus** may consist of **32, 64, 128**, or even more separate lines
- ***For example :*** if the data bus is 32 bits wide and each instruction is 64 bits long, then the processor must access the memory module twice during each instruction cycle.

BUS INTERCONNECTION

- **Address lines** are used to designate the source or destination of the data on the data bus.

  For example, if the processor wishes to read a word (8, 16, or 32 bits) of data from memory, it puts the address of the desired word on the address lines.

  The address lines are generally also used to **address I/O ports** .

BUS INTERCONNECTION

- **Control lines** : are used to control the access to and the use of the data and address lines. Because the data and address lines are shared by all components, there must be a means of controlling their use.

Control signals transmit both command and timing information among system modules. **Timing signals** indicate the validity of data and address information. **Command signals** specify operations to be performed.

BUS INTERCONNECTION

Typical control lines include :

- **Memory write:** Causes data on the bus to be written into the addressed location.
- **Memory read:** Causes data from the addressed location to be placed on the bus .
- **I/O write:** Causes data on the bus to be output to the addressed I/O port
- **I/O read:** Causes data from the addressed I/O port to be placed on the bus .
- **Transfer ACK:** Indicates that data have been accepted from or placed on the bus .

BUS INTERCONNECTION

Typical control lines include :

- **Bus request:** Indicates that a module needs to gain control of the bus .
- **Bus grant:** Indicates that a requesting module has been granted control of the bus .
- **Interrupt request:** Indicates that an interrupt is pending.
- **Interrupt ACK :** Acknowledges that the pending interrupt has been recognized.
- **Clock:** Is used to synchronize operations .
    - **Reset:** Initializes all modules .

BUS INTERCONNECTION

- The operation of the bus is as follows. If one module wishes to send data to another, it must do two things:

  1- obtain the use of the bus .

  2- transfer data via the bus.

  If one module wishes to request data from another module, it must :

  1- obtain the use of the bus .

  2- transfer a request to the other module over the appropriate control and address lines. It must then wait for that second module to send the data.

System Bus Types