



IT111 - Introduction to Information Technology

Bachelor of Science in Information Technology

LECTURE 7 COMPUTER ORGANIZATION (CONTINUE)

Prof. Lamiaa Fattouh Ibrahim
Everyone Smart University
Faculty of Informatics
Spring 2023



Objectives

After studying this chapter, the student should be able to:

- ☐ **Types of Storage Technology Used**
- ☐ **Understand the interconnection of subsystems.**
- ☐ **Describe different methods of input/output addressing.**
- ☐ **Simple Computer**

Types of Storage Technology Used

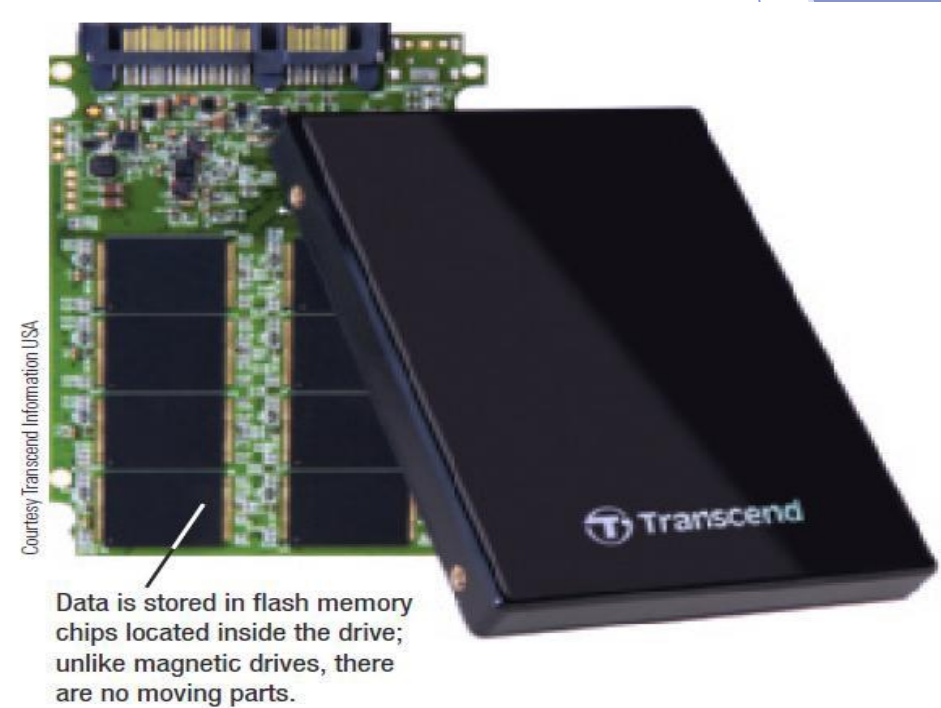
- Magnetic (conventional hard drives)
- Optical (optical discs)
- Electrons (flash memory media)

Internal and External Magnetic Hard Drives

- Internal hard drives
 - Permanent storage devices located inside the system unit
 - Removed only if a problem develops
- External hard drives
 - Commonly used to transport large amounts of data from one computer to another
 - Portable external hard drives are smaller and easier to transport
 - Most connect with a USB connection although some may be wireless

Hard Drives

- Solid State Drives (SSDs)
 - Use less power and have no moving parts
 - Particularly appropriate for portable computers and mobile devices



Optical Discs and Drives

- Major advantage: large capacity
 - CD discs are normally single layer and hold 650 or 700 MB
 - DVD discs hold 4.7 GB (single-layer) or 8.5 GB (dual-layer)
 - BD discs hold 25 GB (single-layer) or 50 GB (dual-layer)
 - Discs can also be double-sided
 - Read-Only Optical Discs: CD ROM, DVD ROM, and BD-ROM – Can be read from, but not written to, by the user
 - CD-R, DVD-R, DVD+R, BD-R Discs Recordable optical discs (also sometimes called write-once discs) can be written to, but the discs cannot be erased and reused.
 - CD-RW, DVD-RW, DVD+RW, and BD-RE Rewritable optical discs can be written to, erased, and overwritten

Electrons (flash memory media)

flash memory is a chip-based storage medium that represents data using electrons. It is used in a variety of storage systems, such as the SSDs and hybrid hard drives.



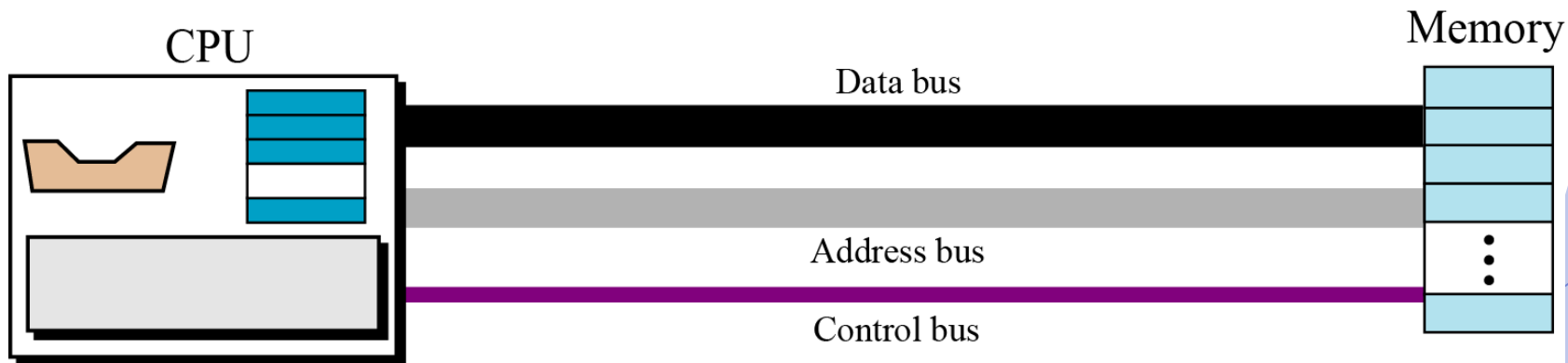
5-5 SUBSYSTEM INTERCONNECTION

The previous sections outlined the characteristics of the three subsystems (CPU, main memory, and I/O) in a stand-alone computer. In this section, we explore how these three subsystems are interconnected. The interconnection plays an important role because information needs to be exchanged between the three subsystems.

5.5.1 Connecting CPU and memory

The CPU and memory are normally connected by three groups of connections, each called a **bus**: *data bus*, *address bus*, and *control bus* (Figure 5.12).

Figure 5.12 Connecting CPU and memory using three buses



Data bus

The data bus is made of several connections, each carrying 1 bit at a time. The number of connections depends on the size of the word used by the computer. If the word is 32 bits (4 bytes), we need a data bus with 32 connections so that all 32 bits of a word can be transmitted at the same time.

Address bus

Address bus The address bus allows access to a particular word in memory. The number of connections in the address bus depends on the address space of the memory. If the memory has 2^n words, the address bus needs to carry n bits at a time. Therefore, it must have n connections.

Control bus

The **control bus** carries communication between the CPU and memory. For example, there must be a code, sent from the CPU to memory, to specify a read or write operation. The number of connections used in the control bus depends on the total number of control commands a computer needs. If a computer has 2^m control actions, we need m connections (wires) for the control bus, because m bits can define 2^m different operations.

Example 1

If the memory has 256 words

How many wires need to have the address bus?

Solution:

256 word = 2^8 words need 8 wires address bus

Example 2

How many operations a control bus with 16 wires

Solution:

Number of operations = $2^{16} = 65536$ operations

Example 3

How many wires in data bus with memory word equal 4 bytes

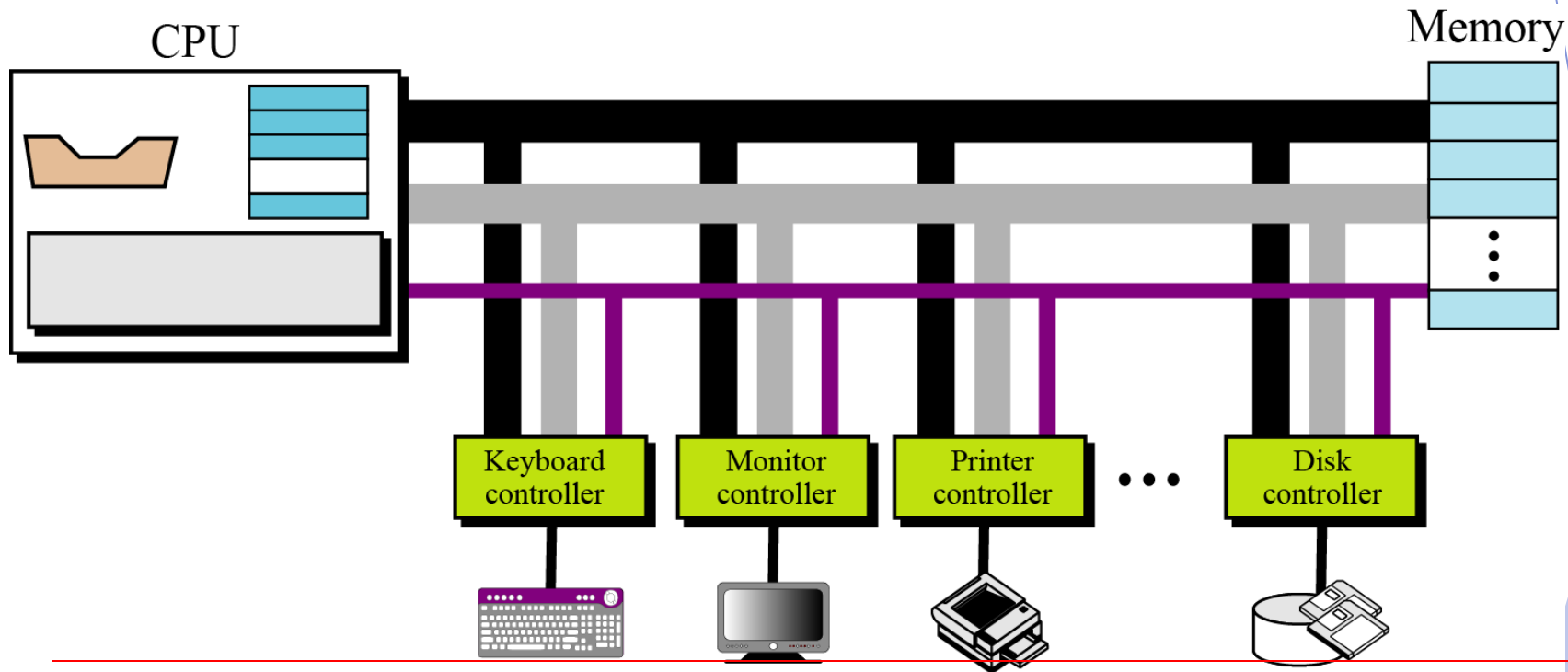
Solution:

Data bus= $4 * 8 = 32$ wires

5.5.2 Connecting I/O devices

I/O devices cannot be connected directly to the buses that connect the CPU and memory because the nature of I/O devices is different from the nature of CPU and memory. I/O devices are electromechanical, magnetic, or optical devices, whereas the CPU and memory are electronic devices. I/O devices also operate at a much slower speed than the CPU/memory. There is a need for some sort of intermediary to handle this difference. Input/output devices are therefore attached to the buses through input/output controllers or interfaces. There is one specific controller for each input/output device (Figure 5.13).

Figure 5.13 Connecting I/O devices to the buses

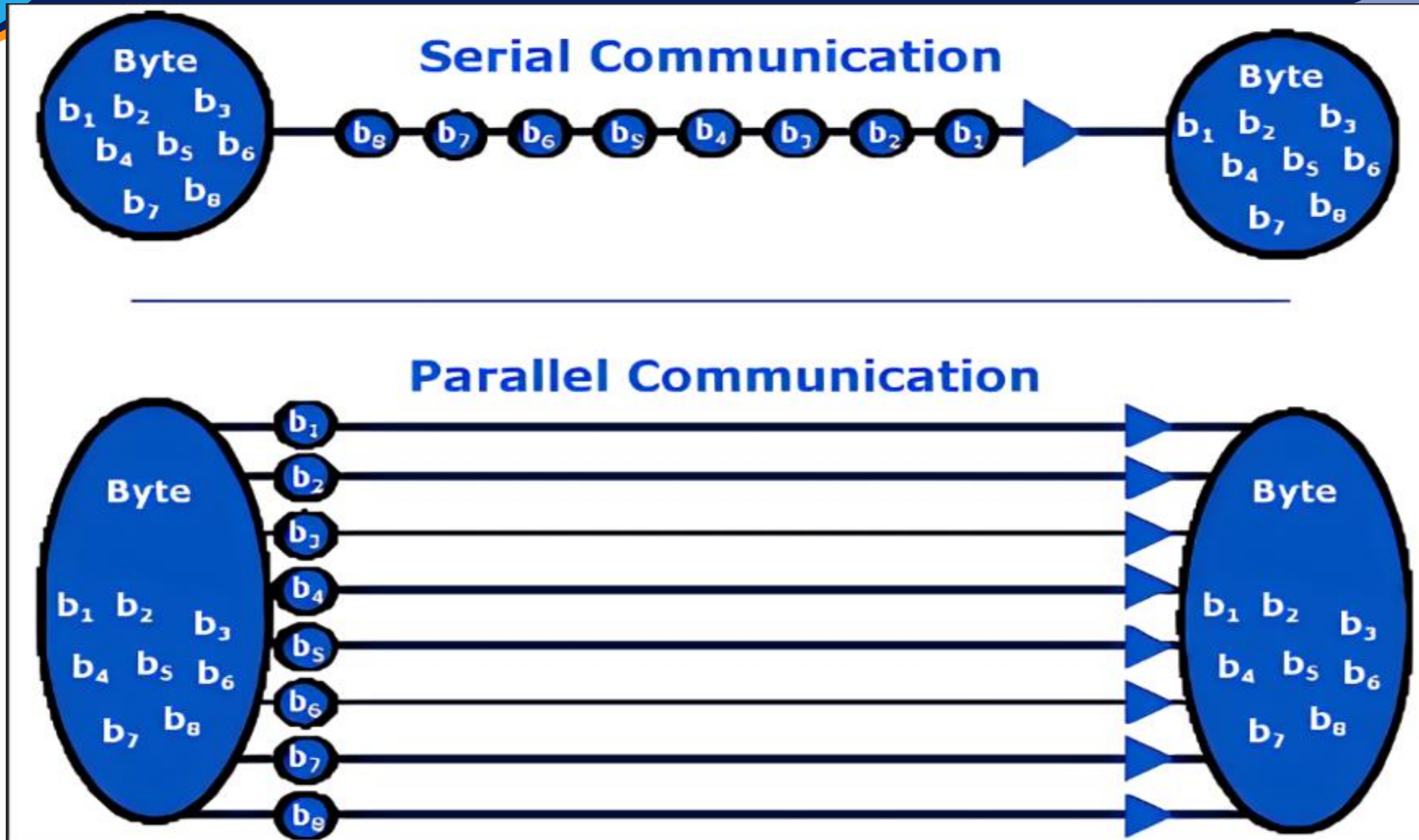


Controllers

Controllers or interfaces bridge the gap between the nature of the I/O device and the CPU and memory. A controller can be a serial or parallel device. A serial controller has only one data wire, while a parallel controller has several data connections so that several bits can be transferred at a time.

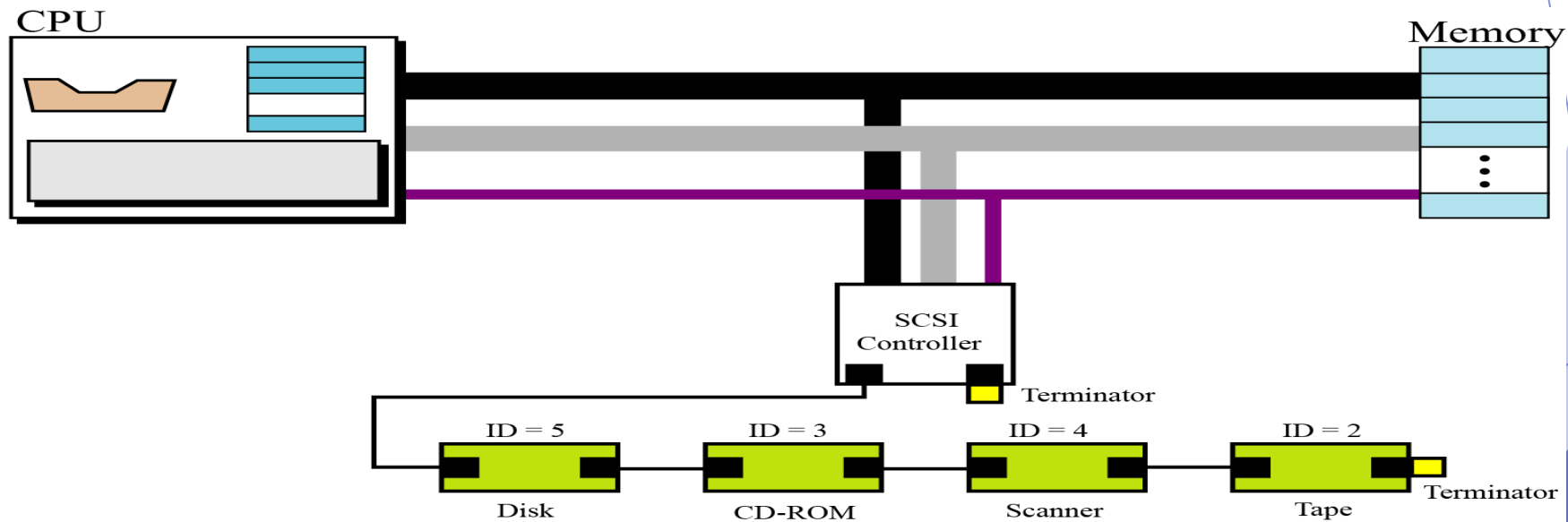
Kinds of controllers are:

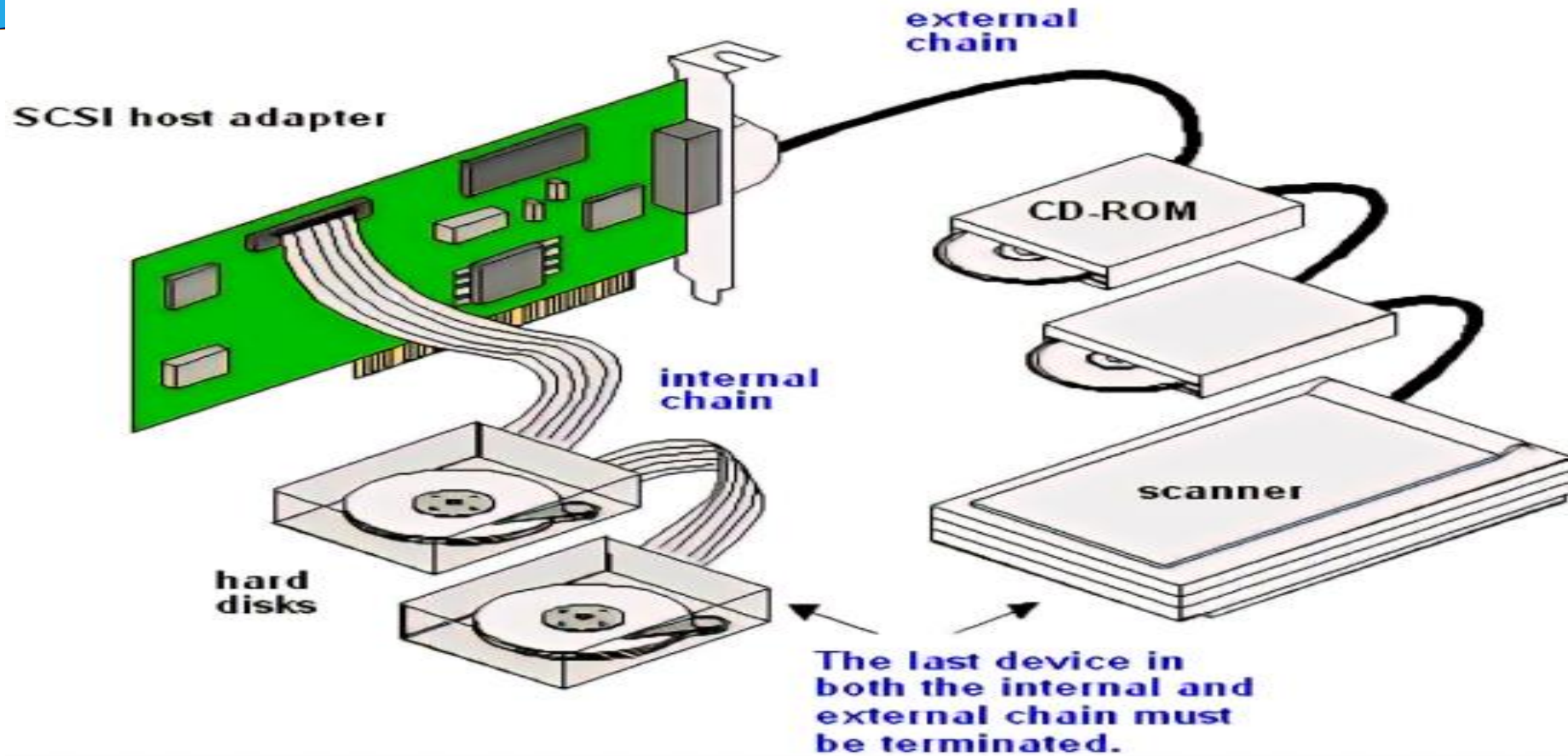
- SCSI
- Fire-Wire
- USB.



SCSI Controller

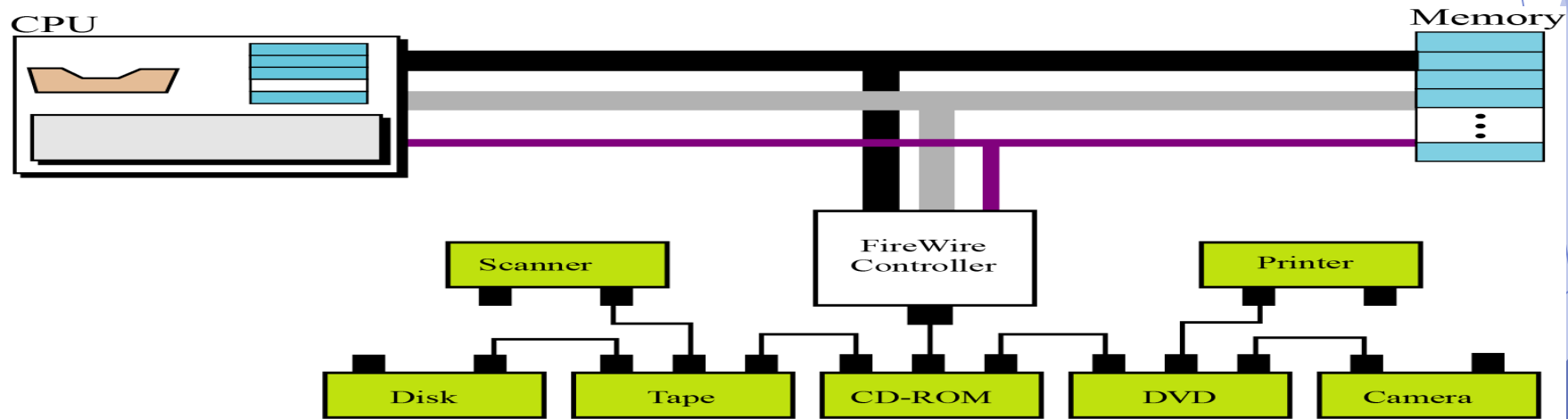
SCSI The small computer system interface (SCSI) has a parallel interface with 8, 16, or 32 connections. The SCSI interface provides a daisy-chained connection. Both ends of the chain must be connected to a special device called a terminator, and each device must have a unique address (target ID).





FireWire

IEEE standard 1394 defines a **serial interface** commonly called FireWire. It is a high-speed serial interface that transfers data in packets, achieving a transfer rate of up to 50 MB/sec, or double that in the most recent version. It can be used to connect up to **63 devices** in a daisy chain or a tree connection (using only one connection). There is no need for termination as there is for SCSI.



FireWire



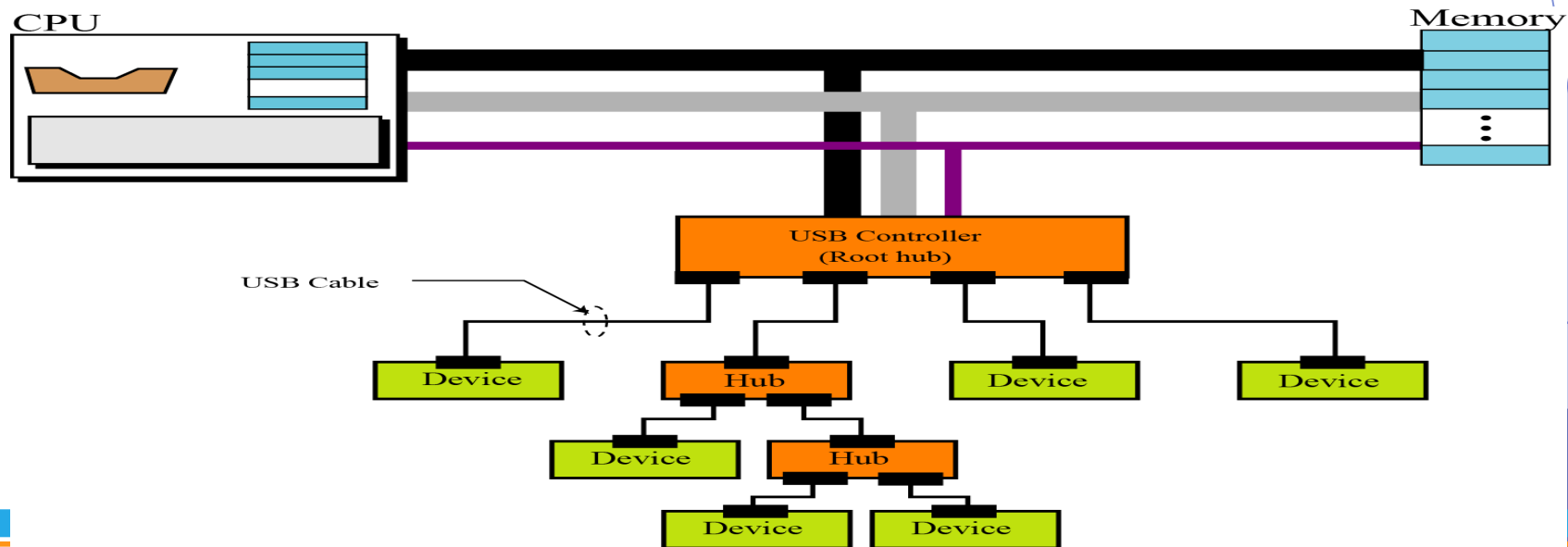
USB controller

Universal Serial Bus (USB) is a competitor for FireWire. Although the terminology uses the term bus, USB is a serial controller that connects both low-and high-speed devices to the computer bus.

Data is transferred over USB in packets. Each packet contains:

- an address part (device identifier).
- a control part.
- part of the data to be transmitted to that device.

All devices will receive the same packet, but only those devices with the address defined in the packet will accept it.



USB controller

- Multiple devices can be connected to a USB controller, which is also referred to as a **root hub**.
- **USB-2** (USB Version 2.0) allows up to **127 devices** to be connected to a USB controller using a tree-like topology with the controller as the root of the tree, hubs as the inter-mediate nodes, and the devices as the end nodes.
- The difference between **the controller (root hub)** and the other **hubs** is that the controller is aware of the presence of other hubs in the tree, but other hubs are passive devices that simply pass the data. Devices can easily be removed or attached to the tree without powering down the computer. This is referred to as **hot-swappable**.
- When a hub is removed from the system, all devices and other hubs connected to it are also removed.

A USB hub is used to connect multiple USB devices to a single USB port.



Receptacles Types		USB 2.0	USB 3.1 Gen1	USB 3.1 Gen2
Type-C				
Type-B	Standard			N/A
	Mini		N/A	N/A
	Micro			
Type-A	Standard			
	Mini		N/A	N/A
	Micro		N/A	N/A

USB Controller

USB-2 provides three data transfer rates:

- 1.5 Mbps (megabits per second): can be used with slow devices such as keyboards and mice.
- 12 Mbps: can be used with printers
- 480 Mbps: can be used with mass storage devices

USB 3.0 adds a new transfer mode called SuperSpeed capable of transferring data at up to 4.8 Gbit/s.

It is promised to update USB 3.0 to 10 Gbit/s.

POWER CONNECTOR
Connects the computer to a power outlet.

FIREWIRE PORT
Connects FireWire devices.

VGA MONITOR PORT
Connects a VGA monitor.

NETWORK PORT
Connects the computer to a network.

USB PORTS
Connect a keyboard, mouse, scanner, flash memory drive, printer, digital camera, or other USB devices.

AUDIO PORTS
Connect speakers, headphones, and a microphone.

HDMI PORT
Connects a high-definition monitor.

EMPTY SLOTS
Ports located on new expansion cards added to the computer will be accessible here.

CONNECTORS



Monitor (VGA)



USB



Monitor (HDMI)



FireWire



Network (RJ-45)



Audio (3.5mm)

High-Definition Multimedia

HDMI (High-Definition Multimedia Interface) is a digital replacement for existing analogue video standards. It can be used for transferring video data digital audio data from a source to a compatible computer monitor, video projector, digital television, or digital audio device.

There are a number of HDMI-standard cables available:

- standard,
- enhanced,
- high definition,
- 3D video signals;
- up to eight channels of compressed or uncompressed digital audio;
- a CEC (Consumer Electronics Control) connection;
- an Ethernet data connection.

5.5.3 Addressing input/output devices

The CPU usually uses the **same bus** to read data from or write data to main memory and I/O device.

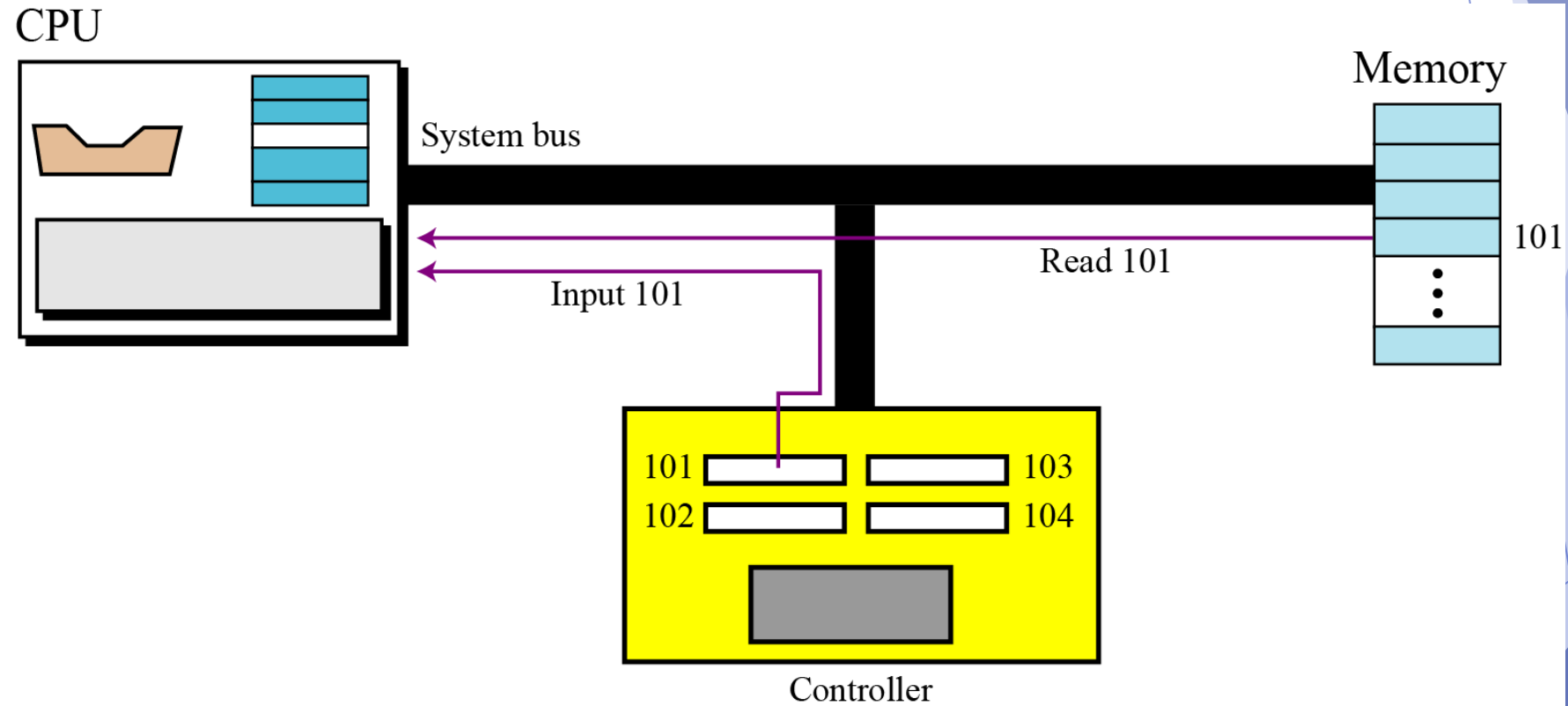
There are two methods for handling the addressing of I/O devices:

- isolated I/O.
- memory-mapped I/O.

Isolated I/O addressing

- In the isolated I/O method, the instructions used to read/write memory are totally different than the instructions used to read/write I/O devices.
- There are instructions to test, control, read from, and write to I/O devices. Each I/O device has its own address. The I/O addresses can overlap with memory addresses without any ambiguity because the instruction itself is different.
- For example, the CPU can use a command 'Read 101' to read from memory word 101, and it can use a command 'Input 101' to read from I/O device 101. There is no confusion, because the read command is for reading from memory and the input command is for reading from an I/O device

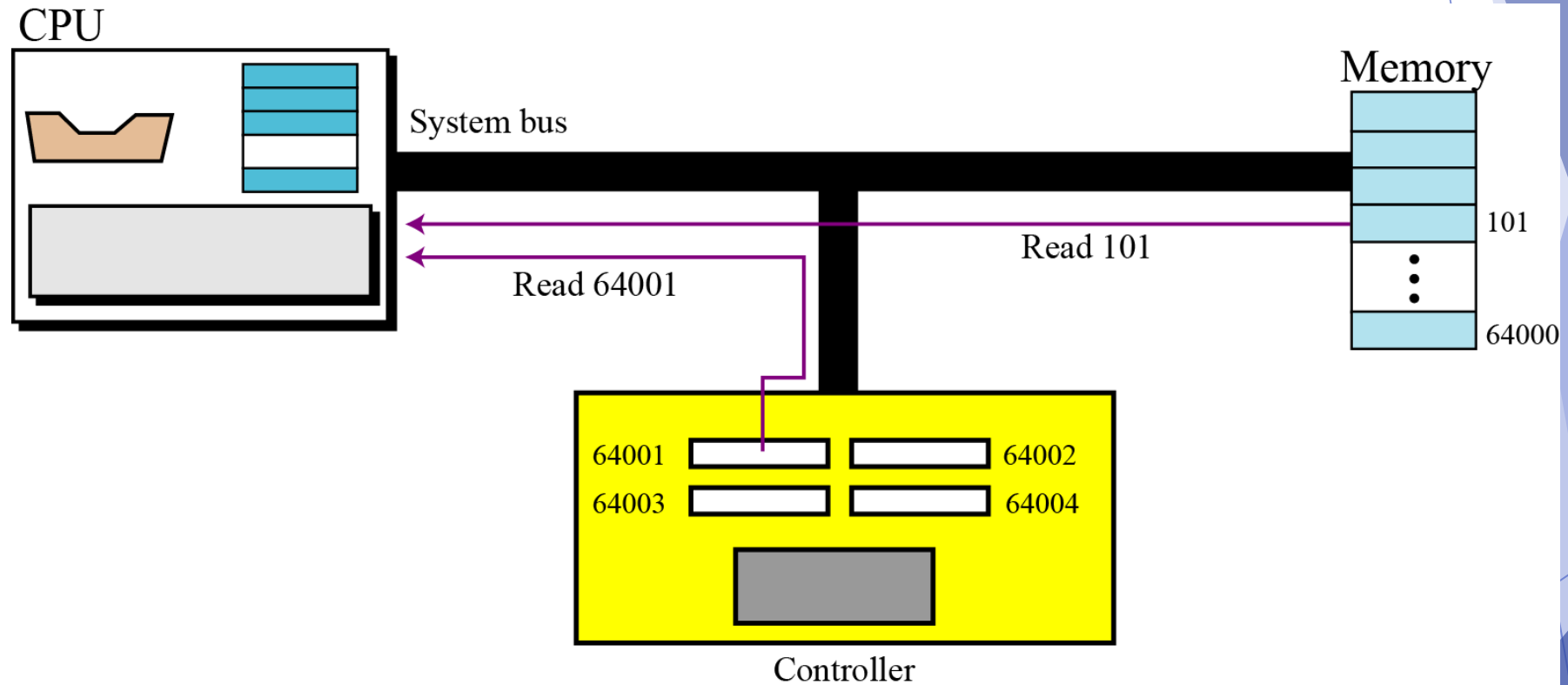
Isolated I/O addressing



Memory-mapped I/O addressing

- In the memory-mapped I/O method, the CPU treats each register in the I/O controller as a word in memory. In other words, the CPU does not have separate instructions for transferring data from memory and I/O devices. For example, there is only one 'Read' instruction. If the address defines a word from memory, the data is read from that word. If the address defines a register from an I/O device, the data is read from that register.
- The advantage of the memory-mapped configuration is a smaller number of instructions: all the memory instructions can be used by I/O devices. The disadvantage is that part of the memory address space is allocated to registers in I/O controllers.
- For example, if we have five I/O controllers and each has four registers, 20 addresses are used for this purpose. The size of the memory is reduced by 20 words.

Memory-mapped I/O addressing



5-6 PROGRAM EXECUTION

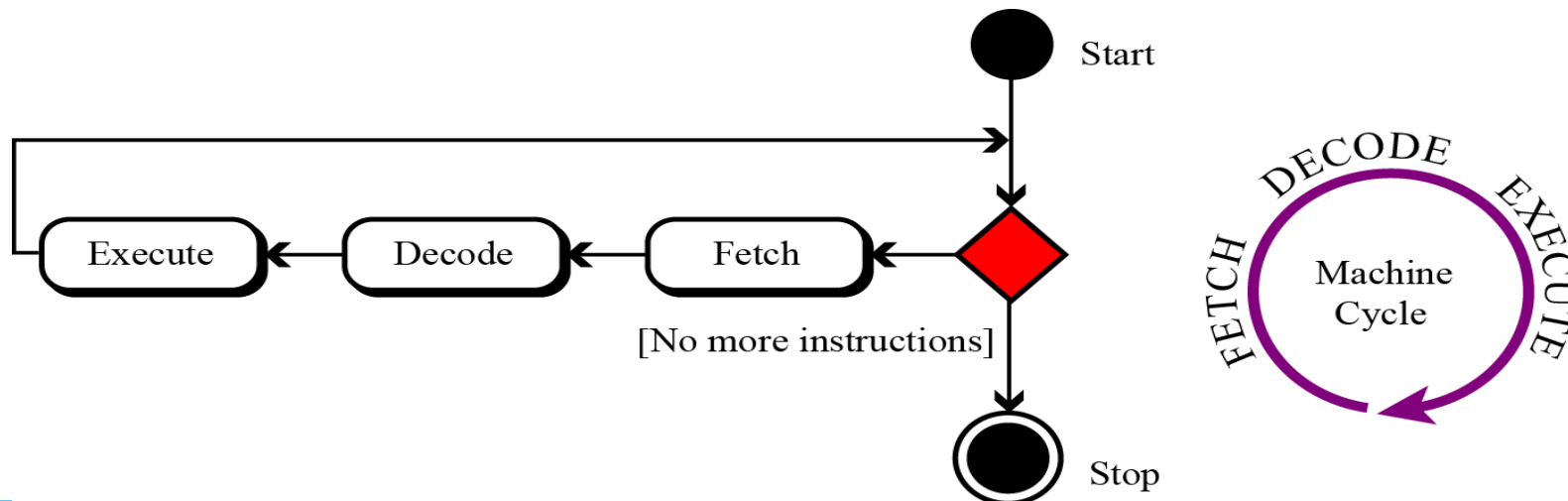
Today, general-purpose computers use a set of instructions called a program to process data. A computer executes the program to create output data from input data. Both the program and the data are stored in memory.

At the end of this chapter we give some examples of how a hypothetical simple computer executes a program.

5.6.1 Machine cycle

The CPU uses repeating **machine cycles** to execute instructions in the program, one by one, from beginning to end. A simplified cycle can consist of three phases: *fetch*, *decode*, and *execute* (Figure 5.19).

Figure 5.19 The steps of a cycle



Fetch- Decode-Execute

Fetch: In the fetch phase, the control unit orders the system to copy the next instruction into the instruction register in the CPU. The address of the instruction to be copied is held in the program counter register. After copying, the program counter is incremented to refer to the next instruction in memory.

Decode: The second phase in the cycle is the decode phase. When the instruction is in the instruction register, it is decoded by the control unit. The result of this decode step is the binary code for some operation that the system will perform

Execute: After the instruction is decoded, the control unit sends the task order to a component in the CPU. For example, the control unit can tell the system to load (read) a data item from memory, or the CPU can tell the ALU to add the contents of two input registers and put the result in an output register. This is the execute phase.

5.6.2 Input/output operation

Commands are required to transfer data from I/O devices to the CPU and memory. Because I/O devices operate at much slower speeds than the CPU, the operation of the CPU must be somehow synchronized with the I/O devices.

Three methods have been devised for this synchronization:

- ☐ **Programmed I/O**
- ☐ **Interrupt driven I/O**
- ☐ **Direct memory access (DMA)**

Programmed I/O

The transfer of data between the I/O device and the CPU is done by an instruction in the program. When the CPU encounters an I/O instruction, it does nothing else until the data transfer is complete.

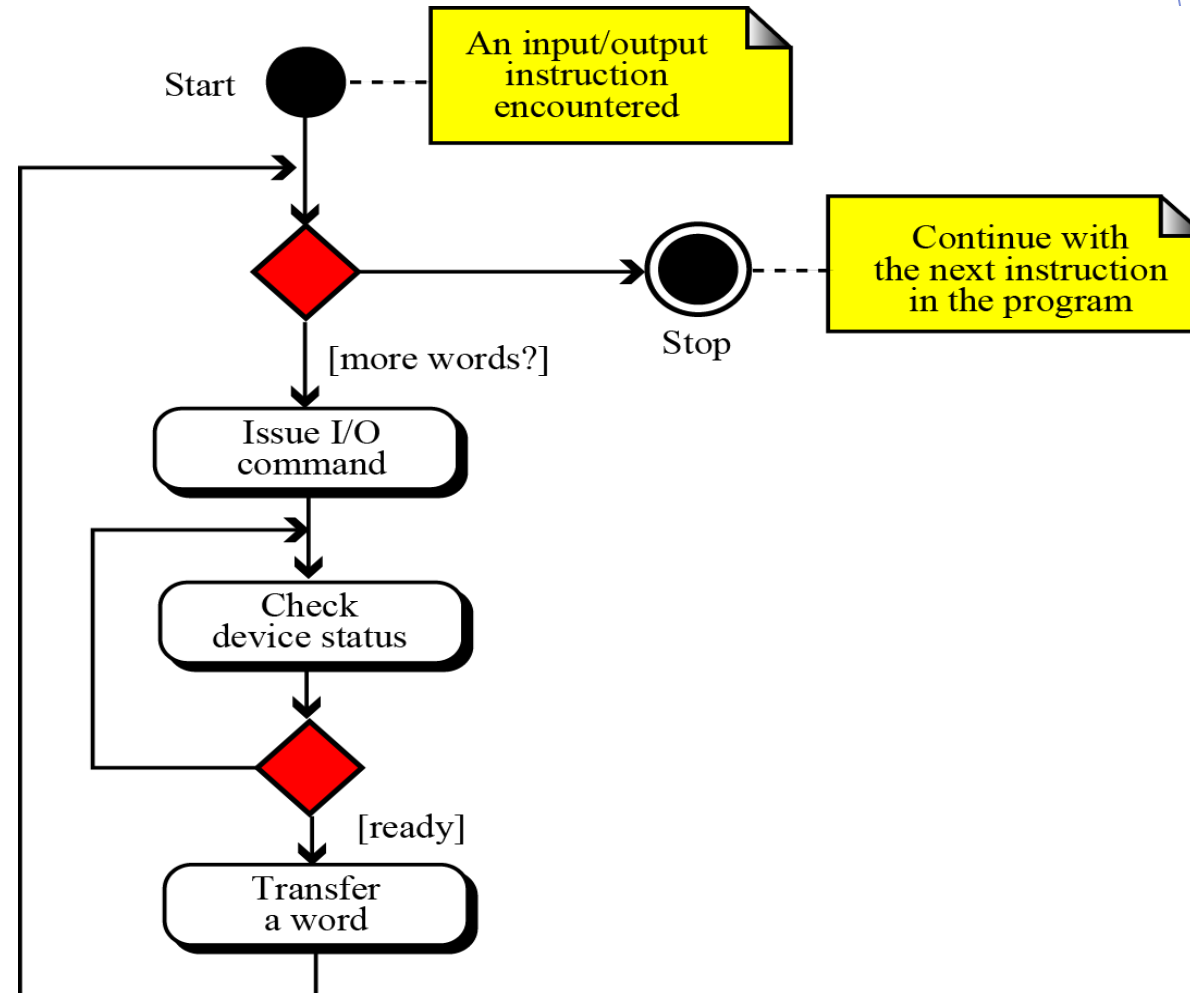
The CPU constantly checks the status of the I/O device:

- If the device is ready to transfer, data is transferred to the CPU.
- If the device is not ready, the CPU continues checking the device status until the I/O device is ready.
- The big issue here is that CPU time is wasted by checking the status of the I/O device for each unit of data to be transferred.

Note that data is transferred to memory after the input operation, while data is transferred from memory before the output operation.

Programmed I/O

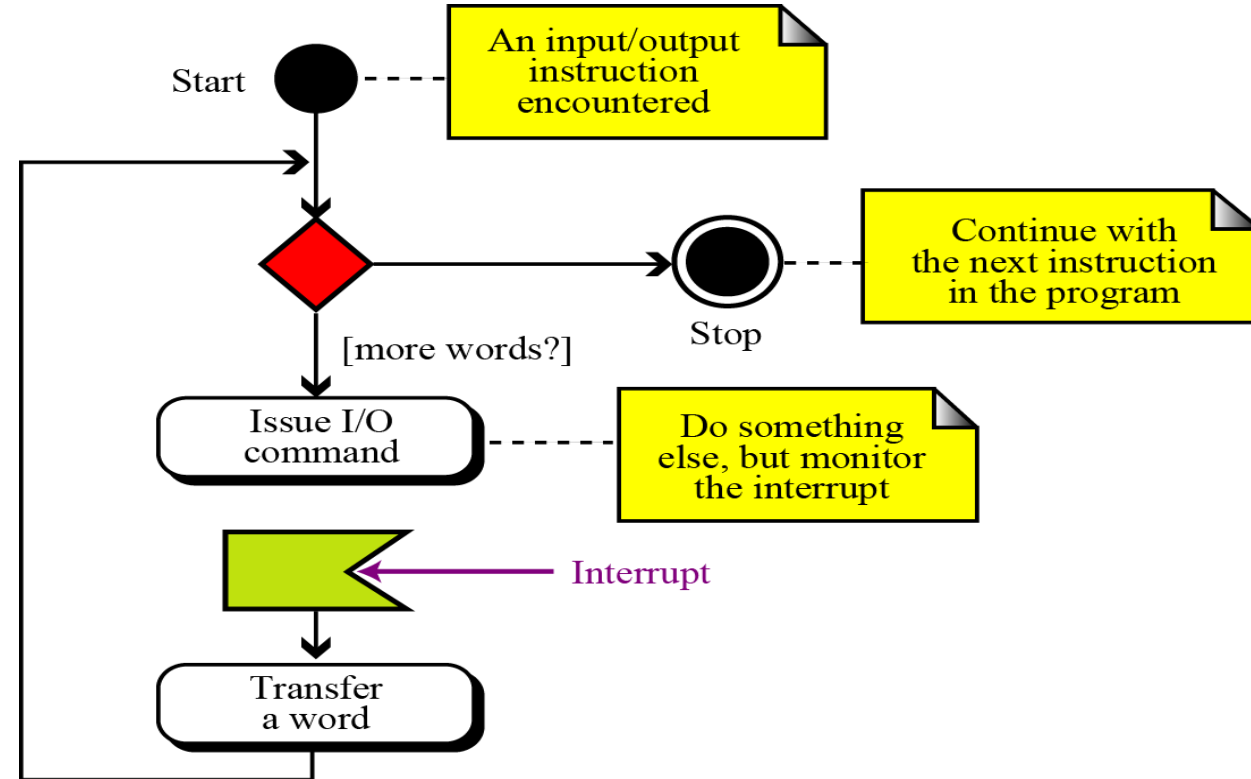
In the programmed I/O method, synchronization is very primitive: the CPU waits for the I/O device



Interrupt-driven I/O

- In the interrupt-driven I/O method, the CPU informs the I/O device that a transfer is going to happen, but it does not test the status of the I/O device continuously.
- The I/O device informs (interrupts) the CPU when it is ready.
- During this time, the CPU can do other jobs such as running other programs or transferring data from or to other I/O devices.
- In this method, CPU time is not wasted—the CPU can do something else while the slow I/O device is finishing a task.

Figure 5.21 Interrupt-driven I/O



Direct memory access (DMA)

- This method transfers a large block of data between a high-speed I/O device, such as a disk, and memory directly without passing it through the CPU.
- This requires a DMA controller that relieves the CPU of some of its functions.
- The DMA controller has registers to hold a block of data before and after memory transfer.
- Using this method for an I/O operation:
 1. The CPU sends a message to the DMA.
 2. The message contains the type of transfer (input or output).
 3. The beginning address of the memory location.
 4. The number of bytes to be transferred.
 5. The CPU is then available for other jobs.
 6. When ready to transfer data, the DMA controller informs the CPU that it needs to take control of the buses.
 7. The CPU stops using the buses and lets the controller use them.
 8. After data transfer directly between the DMA and memory, the CPU continues its normal operation.

DMA connection to the general bus

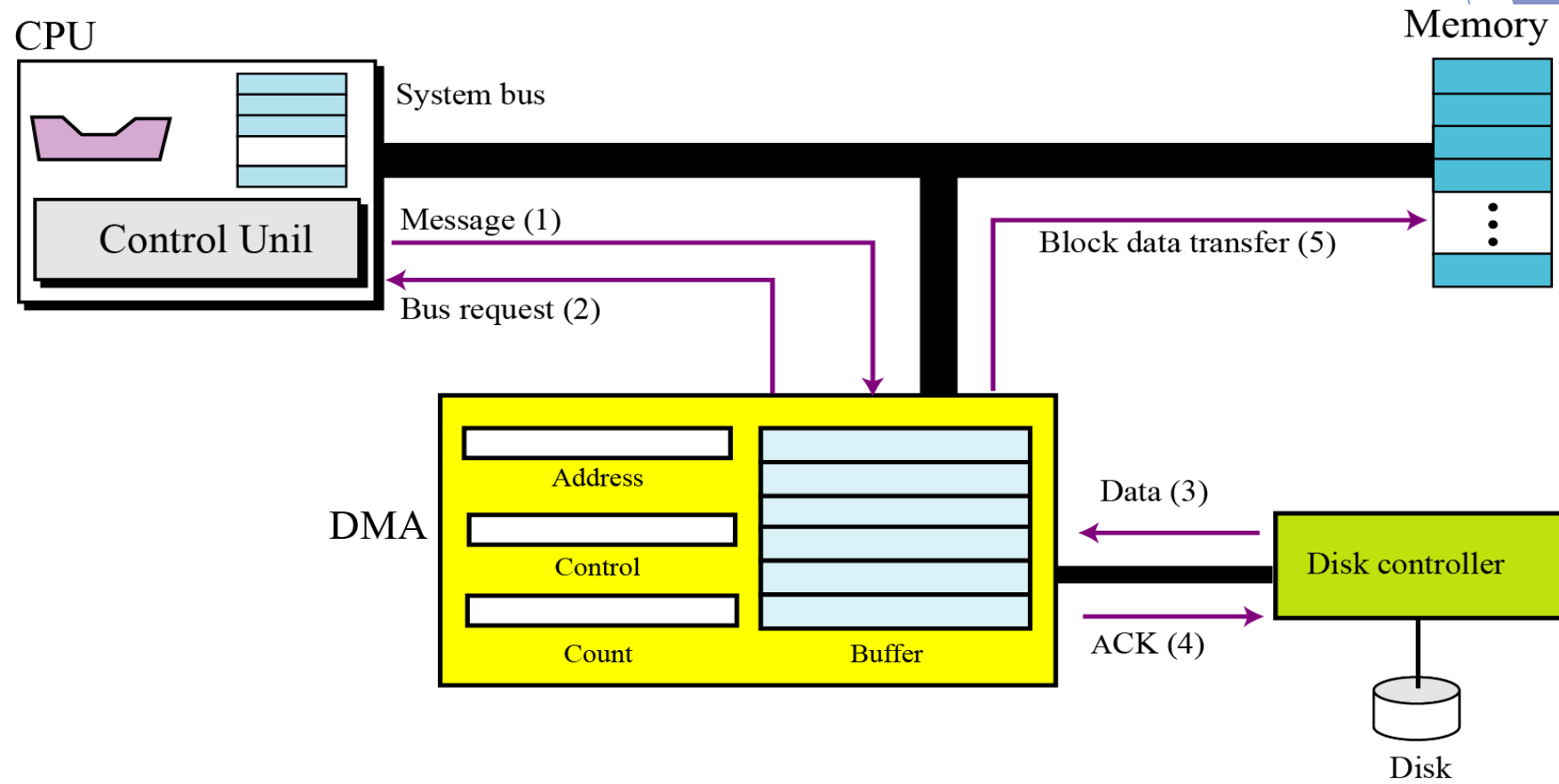
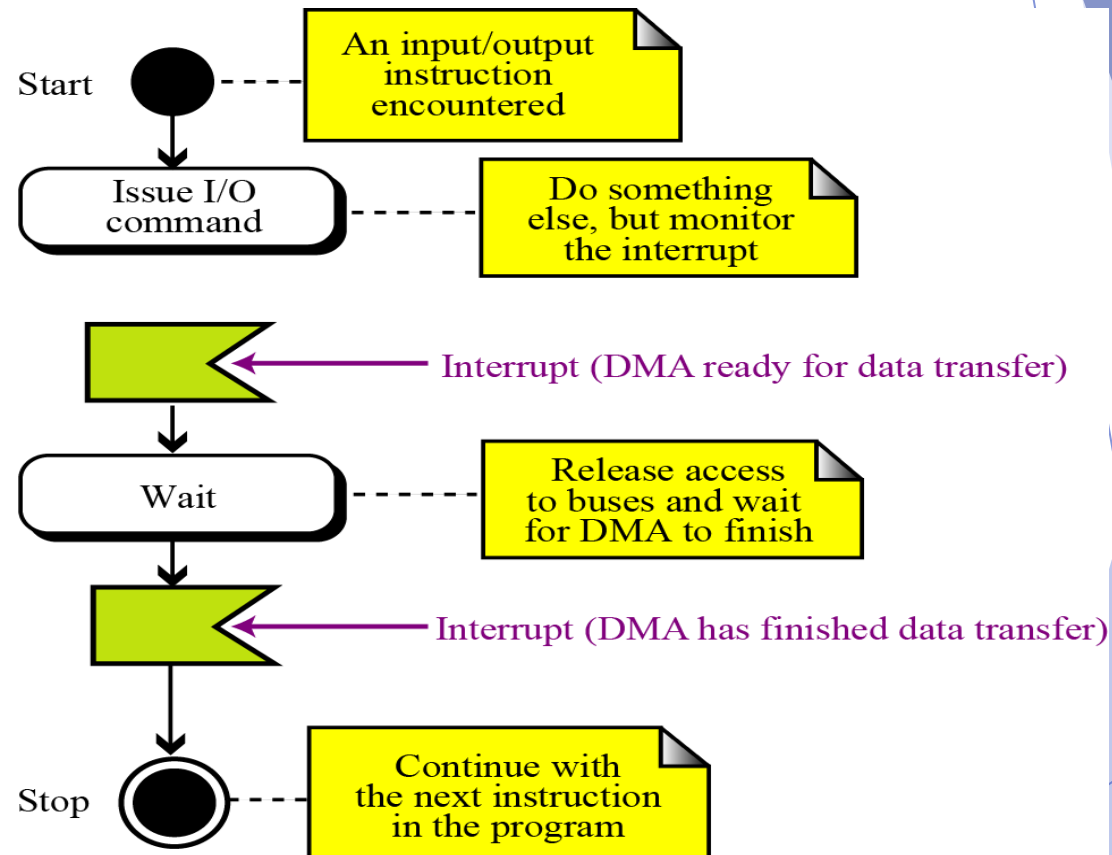


Figure 5.23 DMA input/output

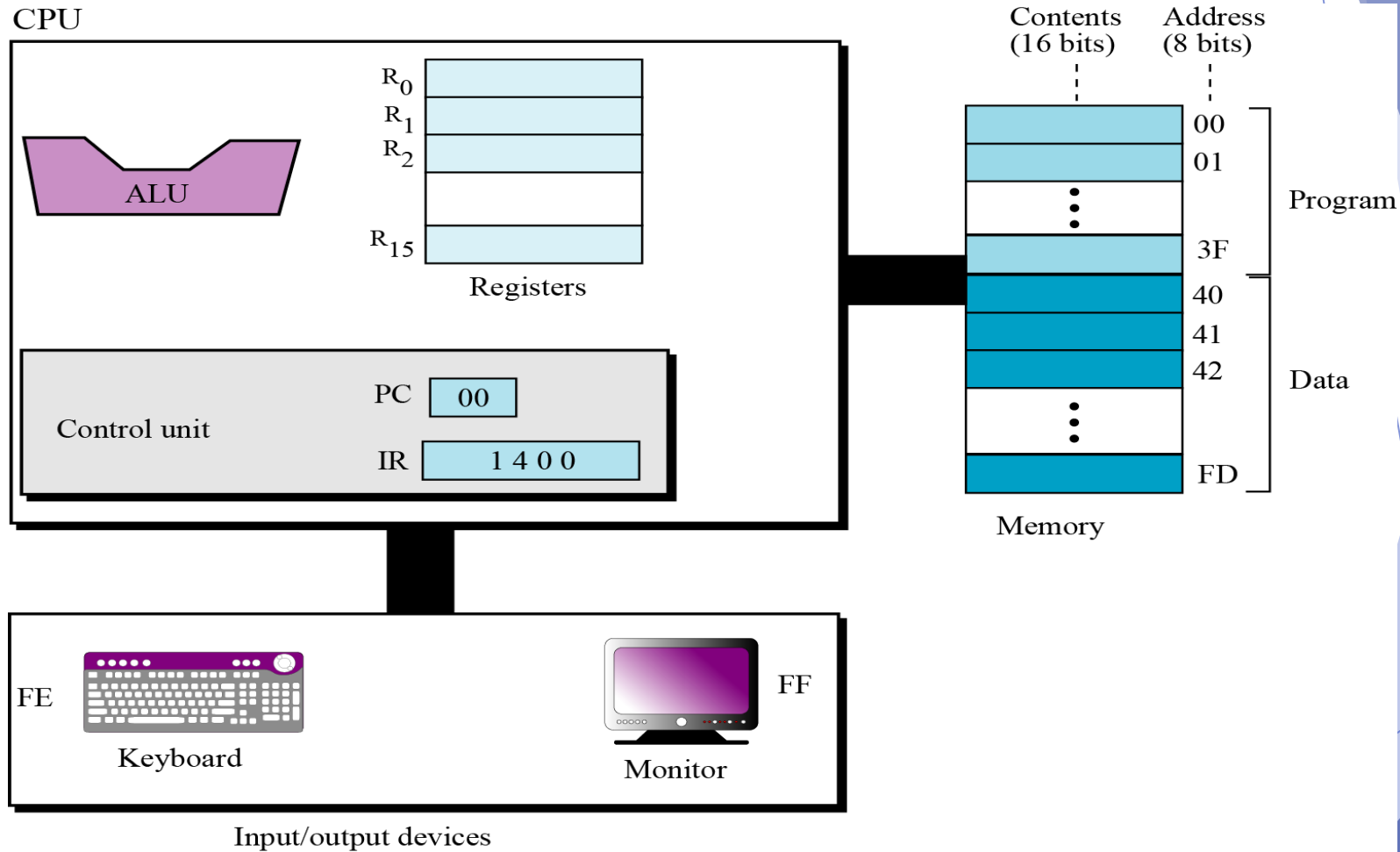
Note that, in this method, the CPU is idle for a time. However, the duration of this idle period is very short compared to other methods the CPU is idle only during the data transfer between the DMA and memory, not while the device prepares the data



5-8 A SIMPLE COMPUTER

To explain the architecture of computers as well as their instruction processing, we introduce a simple (unrealistic) computer, as shown in Figure 5.30. Our simple computer has three components: CPU, memory, and an input/output subsystem.

The Components of a Simple Computer



Summary

- ☐ Types of Storage Technology Used
- ☐ Understand the interconnection of subsystems.
 - ✓ Connecting CPU and memory
 - ✓ Connecting I/O devices
- ☐ Program Execution
- ☐ Describe different methods of input/output addressing.
- ☐ Simple Computer

Thank You