COMPUTER ORGANIZATION AND ARCHITECTURE (COA)

EET 2211
4TH SEMESTER – CSE & CSIT
CHAPTER 7, LECTURE 26

INPUT / OUTPUT

TOPICS TO BE COVERED

1. External Devices
2. I/O Modules
3. Programmed I/O
4. Interrupt-Driven I/O
5. Direct Memory Access
6. Direct Cache Access

LEARNING OBJECTIVES

- Explain the use of I/O modules as part of a computer organization.
- Understand programmed I/O and discuss its relative merits.
- Present an overview of the operation of direct memory access.
- Present an overview of direct cache access.

INTRODUCTION

- In addition to the processor and a set of memory modules, the third key element of a computer system is a set of I/O modules.

- Each module interfaces to the system bus or central switch and controls one or more peripheral devices.

- An I/O module is not simply a set of mechanical connectors that wire a device into the system bus. Rather, the I/O module contains logic for performing a communication function between the peripheral and the bus.

Why peripherals are not connected directly to the system bus ?

- There are a wide variety of peripherals with various methods of operation. It would be impractical to incorporate the necessary logic within the processor to control a range of devices.

- The data transfer rate of peripherals is often much slower than that of the memory or processor. Thus, it is impractical to use the high-speed system bus to communicate directly with a peripheral.

- The data transfer rate of some peripherals is faster than that of the memory or processor. Again, the mismatch would lead to inefficiencies if not managed properly.

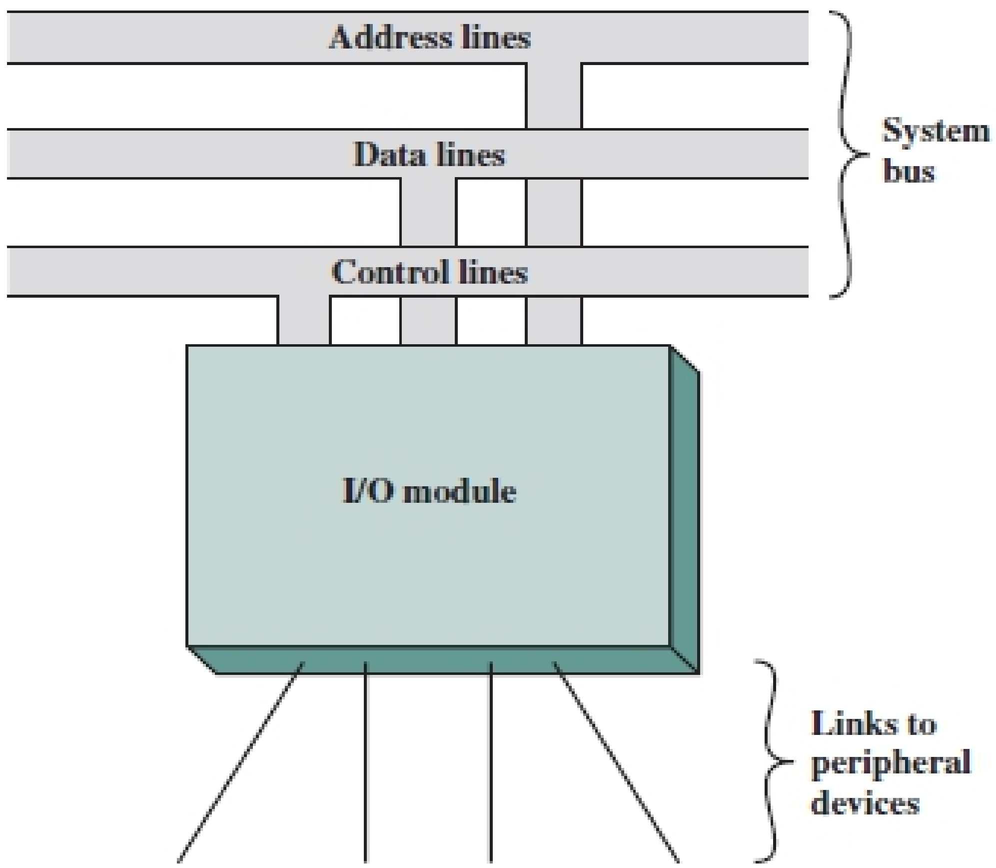
- Peripherals often use different data formats and word lengths than the computer to which they are attached.

Thus, an I/O module is required.

Major functions of I/O Module

- Interface to the processor and memory via the system bus or central switch.
- Interface to one or more peripheral devices by tailored data links.

Generic Model of an I/O Module



EXTERNAL DEVICES

- I/O operations are accomplished through a wide assortment of external devices that provide a means of exchanging data between the external environment and the computer.

- An external device attaches to the computer by a link to an I/O module.

- The link is used to exchange control, status, and data between the I/O module and the external device.

- An external device connected to an I/O module is often referred to as a *peripheral device or, simply, a peripheral*.

Classification of external devices

- **Human readable**: Suitable for communicating with the computer user;

Example- video display terminals (VDTs) and printers.

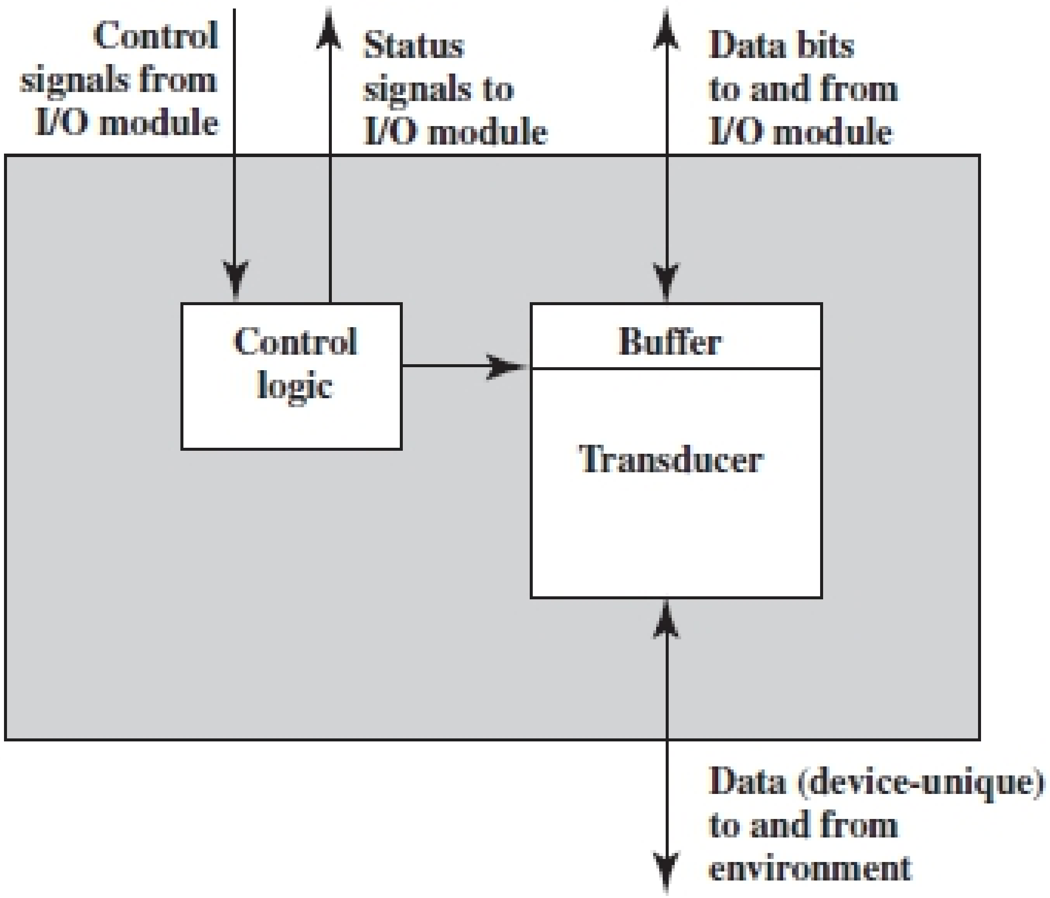
- **Machine readable**: Suitable for communicating with equipment;

Example- magnetic disk and tape systems, sensors and actuators, which are used in a robotics application

- **Communication**: Suitable for communicating with remote devices.

Example- Human-readable device, such as a terminal, a machine-readable device, or even another computer.

Block Diagram of an External Device



- ***Control logic*** *associated with the device controls the device's operation in* response to direction from the I/O module.

- ***Transducer*** **-**The ***transducer*** *converts data from electrical* to other forms of energy during output and from other forms to electrical during input.

- **Buffer-** Typically, a **buffer** is associated with the transducer to temporarily hold data being transferred between the I/O module and the external environment.

- A buffer size of 8 to 16 bits is common for serial devices, whereas block-oriented devices such as disk drive controllers may have much larger buffers.

- The interface to the I/O module is in the form of control, data, and status signals.

- **Control signals** *determine the function that the device will perform, such as*
  1) send data to the I/O module (**INPUT or READ**)
  2) accept data from the I/O module (**OUTPUT or WRITE**)
  3) report status
  4) perform some control function particular to the device
  (e.g., **position a disk head**)

- ***Data*** *are in the form of a set of bits to be sent to or received from* the I/O module.

- ***Status signals*** *indicate the state of the device. Examples are* ***READY/NOT-*** **READY** to show whether the device is ready for data transfer.

Keyboard/Monitor

- The most common means of computer/user interaction is a keyboard/monitor arrangement.
- The user provides input through the keyboard, the input is then transmitted to the computer and may also be displayed on the monitor.
- In addition, the monitor displays data provided by the computer.
- The basic unit of exchange is the **character**.

- A code is associated with each character, typically 7 or 8 bits in length.
- The most commonly used text code is the **International Reference Alphabet (IRA).**
- Each character in this code is represented by a unique 7-bit binary code; thus, 128 different characters can be represented.
- Characters are of two types: printable and control.
- **Printable characters** are the alphabetic, numeric, and special characters that can be printed on paper or displayed on a screen.
- Some of the **control characters** have to do with controlling the printing or displaying of characters; an example is carriage return. Other control characters are concerned with communications procedures.

Working Principle

- For keyboard input, when the user depresses a key, this generates an electronic signal that is interpreted by the transducer in the keyboard and translated into the bit pattern of the corresponding IRA **(International Reference Alphabet)**code.
- This bit pattern is then transmitted to the I/O module in the computer.
- At the computer, the text can be stored in the same IRA code.
- On output, IRA code characters are transmitted to an external device from the I/O module.
- The transducer at the device interprets this code and sends the required electronic signals to the output device either to display the indicated character or perform the requested control function.

Disk Drive

- A disk drive contains electronics for exchanging data, control, and status signals with an I/O module plus the electronics for controlling the disk read/write mechanism.

- In a fixed-head disk, the transducer is capable of converting between the magnetic patterns on the moving disk surface and bits in the device's buffer.

- A moving-head disk must also be able to cause the disk arm to move radially in and out across the disk's surface.

I/O MODULES

Module Function

The major functions or requirements for an I/O module fall into the following categories:

- Control and timing
- Processor communication
- Device communication
- Data buffering
- Error detection

Control and timing

- During any period of time, the processor may communicate with one or more external devices in unpredictable patterns, depending on the program's need for I/O.
- The internal resources, such as main memory and the system bus, must be shared among a number of activities, including data I/O.
- Thus, the I/O function includes a **control and timing requirement, to coordinate the flow of traffic** between internal resources and external devices.

Example

The control of the transfer of data from an external device to the processor might involve the following sequence of steps:

1. The processor interrogates the I/O module to check the status of the attached device.
2. The I/O module returns the device status.
3. If the device is operational and ready to transmit, the processor requests the transfer of data, by means of a command to the I/O module.
4. The I/O module obtains a unit of data (e.g., 8 or 16 bits) from the external device.
5. The data are transferred from the I/O module to the processor.

Processor communication

- **Command decoding:** The I/O module accepts commands from the processor, typically sent as signals on the control bus. For example, an I/O module for a disk drive might accept the following commands: READ SECTOR, WRITE SECTOR, SEEK track number, and SCAN record ID. The latter two commands each include a parameter that is sent on the data bus.

- **Data:** Data are exchanged between the processor and the I/O module over the data bus.

- **Status reporting:** Because peripherals are so slow, it is important to know the status of the I/O module. For example, if an I/O module is asked to send data to the processor (read), it may not be ready to do so because it is still working on the previous I/O command. This fact can be reported with a status signal. Common status signals are BUSY and READY. There may also be signals to report various error conditions.

- **Address recognition:** Just as each word of memory has an address, so does each I/O device. Thus, an I/O module must recognize one unique address for each peripheral it controls.

DEVICE COMMUNICATION

This communication involves commands, status information, and data.

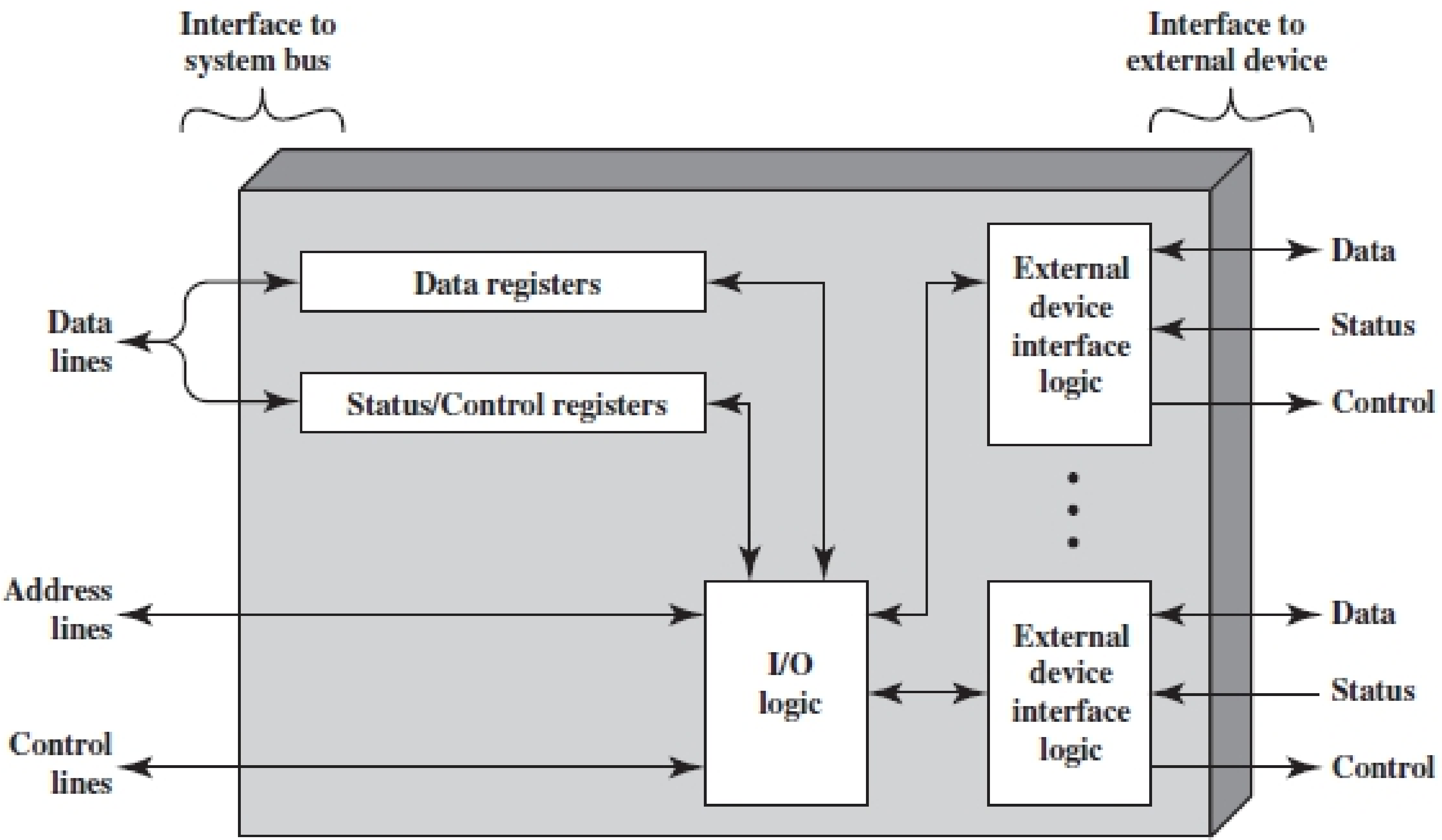
Data buffering

- The transfer rate into and out of main memory or the processor is quite high, but its rate is lower for many peripheral devices.
- Data coming from main memory are sent to an I/O module in a rapid burst.
- The data are buffered in the I/O module and then sent to the peripheral device at its data rate.
- In the opposite direction, data are buffered so as not to tie up the memory in a slow transfer operation.
- So I/O module must be able to operate at both device and memory speeds.

Error detection

- I/O module is often responsible for **error detection and for subsequently** reporting errors to the processor.
- One class of errors includes mechanical and electrical malfunctions reported by the device (e.g., paper jam, bad disk track).
- Another class consists of unintentional changes to the bit pattern as it is transmitted from device to I/O module. Some form of error-detecting code is often used to detect transmission errors.

I/O MODULE STRUCTURE



- An I/O module that takes on most of the detailed processing burden, presenting a high-level interface to the processor, is usually referred to as an I/O channel or *I/O processor*.

- *An I/O module that is quite primitive and requires detailed control* is usually referred to as an *I/O controller or device controller*. *I/O controllers are* commonly seen on microcomputers, whereas I/O channels are used on mainframes.

PROGRAMMED I/O

Programmed I/O

- With *programmed I/O,* data are exchanged between the processor and the I/O module.

- The processor executes a program that gives it direct control of the I/O operation, including sensing device status, sending a read or write command, and transferring the data.

- When the processor issues a command to the I/O module, it must wait until the I/O operation is complete. If the processor is faster than the I/O module, this is waste of processor time.

Interrupt-driven I/O

- With interrupt-driven I/O, the processor issues an *I/O command, continues* to execute other instructions, and is interrupted by the I/O module when the latter has completed its work.

- With both programmed and *interrupt I/O, the processor is* responsible for extracting data from main memory for output and storing data in main memory for input.

DIRECT MEMORY ACCESS (DMA)

In this mode, the I/O module and main memory exchange data directly, without processor involvement.

I/O Techniques

	No Interrupts	Use of Interrupts
I/O-to-memory transfer through processor	Programmed I/O	Interrupt-driven I/O
Direct I/O-to-memory transfer		Direct memory access (DMA)

Overview of Programmed I/O

- When the processor is executing a program and encounters an instruction relating to I/O, it executes that instruction by issuing a command to the appropriate I/O module.

- With programmed I/O, the I/O module will perform the requested action and then set the appropriate bits in the I/O status register. The I/O module takes no further action to alert the processor. In particular, it does not interrupt the processor.

- Thus, it is the responsibility of the processor to periodically check the status of the I/O module until it finds that the operation is complete.

I/O Commands

- **Control: Used to activate a peripheral and tell it what to do.**

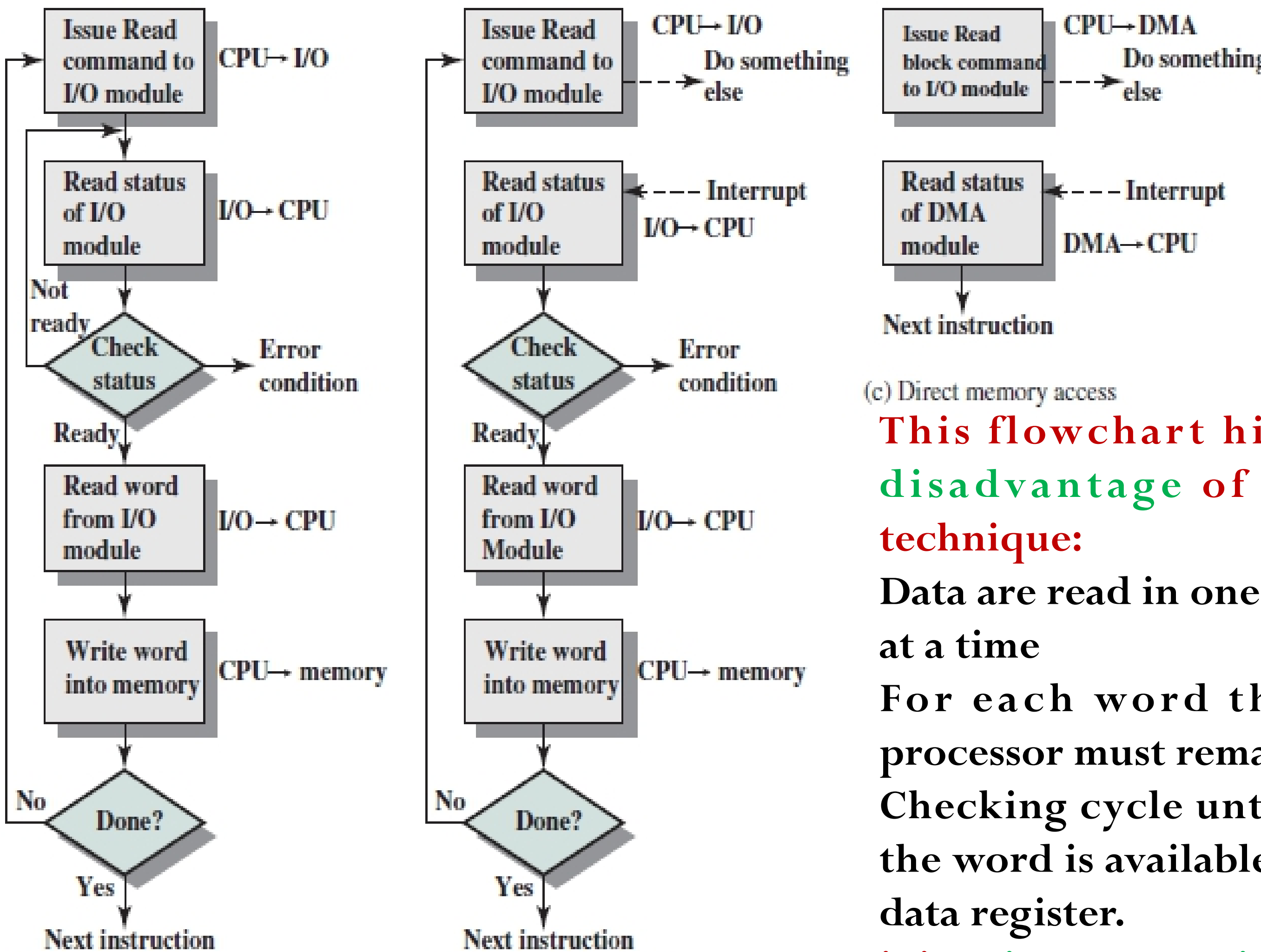
  For example, a Magnetic-tape unit may be instructed to rewind or to move forward one record. These commands are tailored to the particular type of peripheral device.

- **Test: Used to test various status conditions associated with an I/O module and its peripherals.** The processor will want to know that the peripheral of interest is powered on and available for use. It will also want to know if the most recent I/O operation is completed and if any errors occurred.

- **Read: Causes the I/O module to obtain an item of data from the peripheral and place it in an internal buffer.** The processor can then obtain the data item by requesting that the I/O module place it on the data bus.

- **Write: Causes the I/O module to take an item of data (byte or word) from the data bus and subsequently transmit that data item to the peripheral.**

Three Techniques for Input of a Block of Data



(a) Programmed I/O

(b) Interrupt-driven I/O

(c) Direct memory access

This flowchart highlights the main disadvantage of programmed I/O technique:

Data are read in one word (e.g., 16 bits) at a time

For each word that is read in, the processor must remain in a status-Checking cycle until it determines that the word is available in the I/O module's data register.

it is a time-consuming process that keeps the processor busy needlessly.

I/O Instructions

- I/O-related instructions are instructions that the processor fetches from memory

- I/O commands are commands that the processor issues to an I/O module to execute the instructions.

- When the processor, main memory, and I/O share a common bus, two modes of addressing are possible:
- Memory mapped
- Isolated.

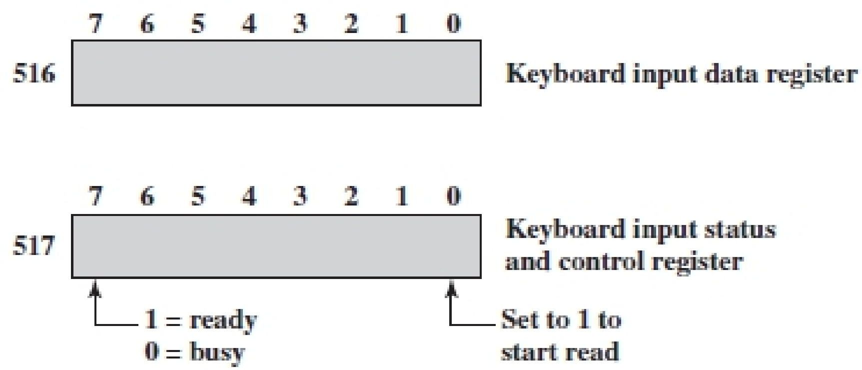
Memory-mapped I/O

- With **memory-mapped I/O,** there is a single address space for memory locations and I/O devices.
- The **processor** treats the status and data registers of I/O modules as memory locations and uses the same machine instructions to access both memory and I/O devices.
- So, for example, with 10 address lines, a combined total of $2^{10} = 1024$ memory locations and I/O addresses can be supported, in any combination.
- With memory-mapped I/O, a single read line and a single write line are needed on the bus.

Isolated I/O

- The bus may be equipped with memory read and write plus input and output command lines.
- The command line specifies whether the address refers to a memory location or an I/O device.
- The full range of addresses may be available for both.
- Again, with 10 address lines, the system may now support both 1024 memory locations and 1024 I/O addresses.
- Because the address space for I/O is isolated from that for memory, this is referred to as **isolated I/O**.

I/O Instructions(Memory- Mapped and Isolated I/O)



ADDRESS	INSTRUCTION	OPERAND	COMMENT
200	Load AC	"1"	Load accumulator
	Store AC	517	Initiate keyboard read
202	Load AC	517	Get status byte
	Branch if Sign = 0	202	Loop until ready
	Load AC	516	Load data byte

(a) Memory-mapped I/O

ADDRESS	INSTRUCTION	OPERAND	COMMENT
200	Load I/O	5	Initiate keyboard read
201	Test I/O	5	Check for completion
	Branch Not Ready	201	Loop until complete
	In	5	Load data byte

(b) Isolated I/O

THANK YOU