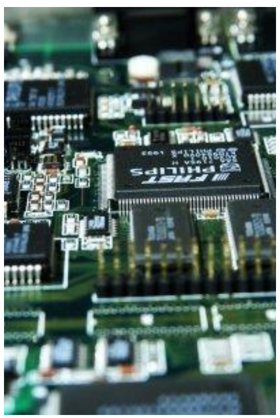
CSE 401

Computer Engineering (2)

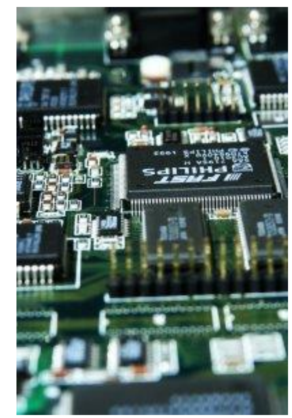
هندسة الحاسبات (2)



4th year, Comm. Engineering

Winter 2016

Lecture #6



Dr. Hazem Ibrahim Shehata
Dept. of Computer & Systems Engineering

Credits to Dr. Ahmed Abdul-Monem Ahmed for the slides

Adminstrivia

- Assignment #2:
  - To be released next week.

Website: http://hshehata.github.io/courses/zu/cse401/
Office hours: Monday 11:30am – 12:30pm

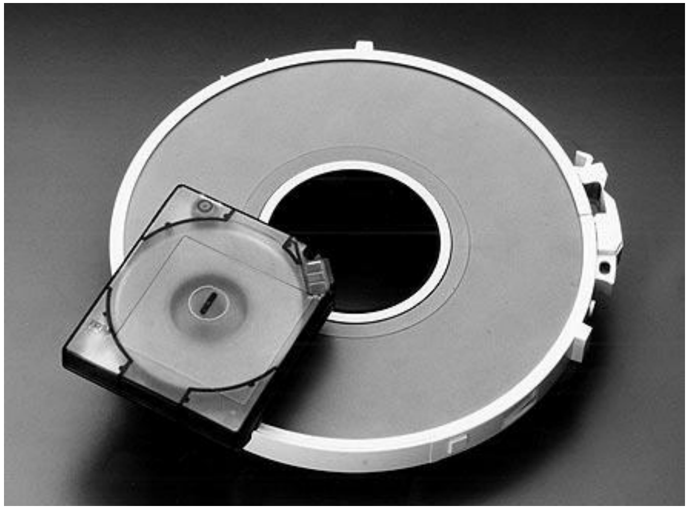
Chapter 6. External Memory (*Cont.*)

Types of External Memory

- Magnetic Disk
- Redundant Array of Independent Disks (RAID)
- Solid-State Drive (SSD)
- Optical Disk
- Magnetic Tape

Magnetic Tape

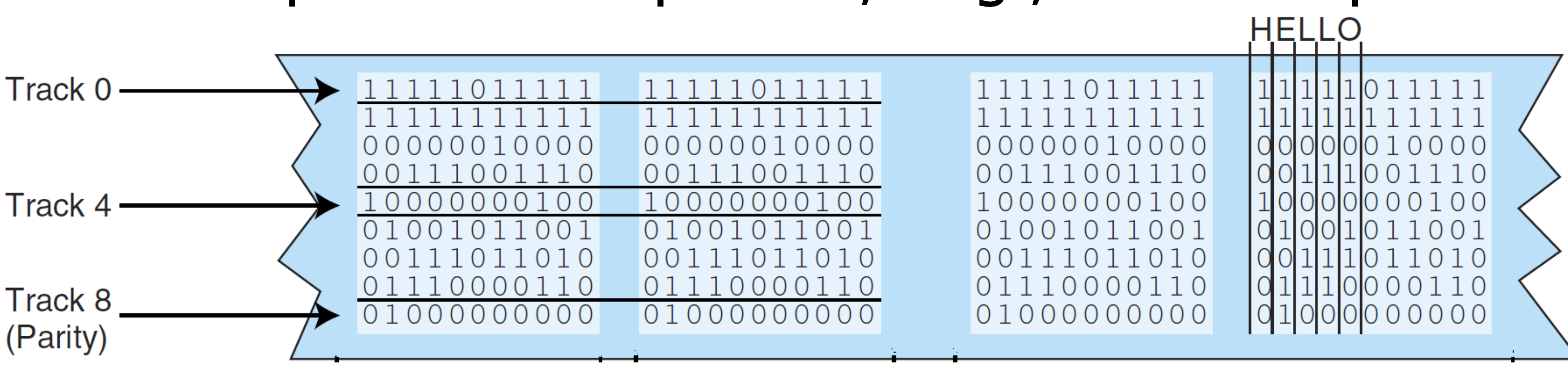
- Features:
  - —Oldest secondary memory (1951).
  - —Slowest-speed & lowest-cost in memory hierarchy.
- Usage: offline storage (i.e., backup).
- Access method: sequential ➔ slow!
- Medium: flexible polyester tape coated with magnetizable material (metal particles).
- Tape width: 0.38cm - 1.27cm.
- Tape package:
  - —Used to be open reels.
  - —Now housed in cartridges.



Data Layout and Recording Techniques

- Data are laid out on tape along parallel tracks running lengthwise ➔ linear recording.
- Tape moves underneath magnetic head in drive.
- Head has multiple (8-32) read/write elements ➔ read/write multiple (8-32) tracks simultaneously.
- There are two types of linear recording:

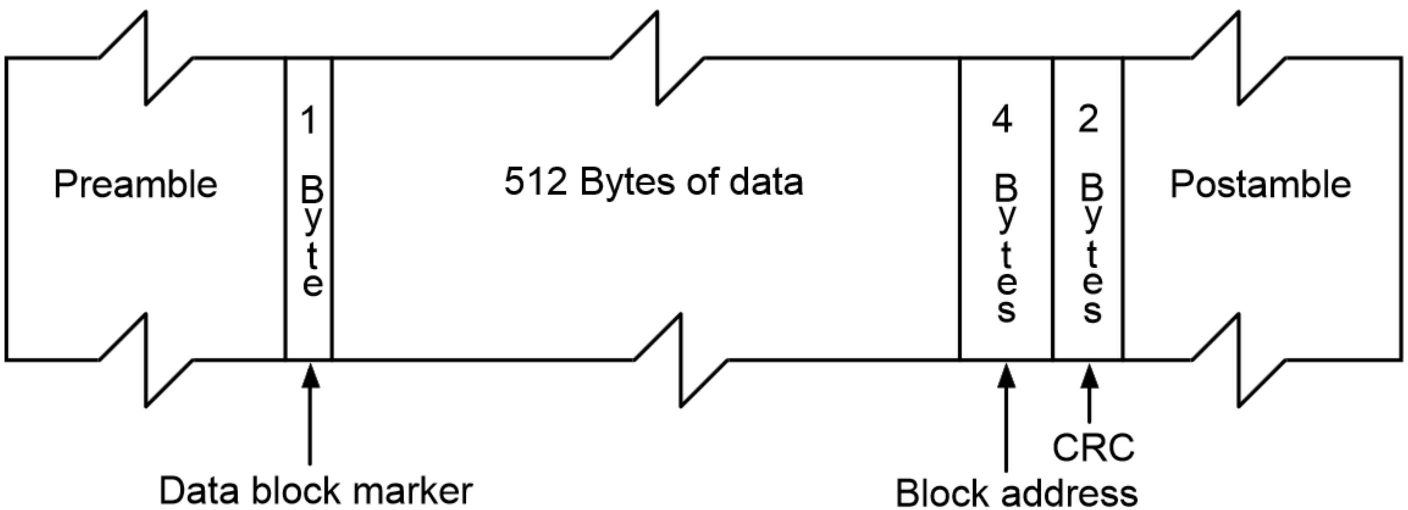
1. Parallel recording: Each byte/word is stored on multiple tracks in parallel, .e.g., 9-track tapes.



Serial Recording

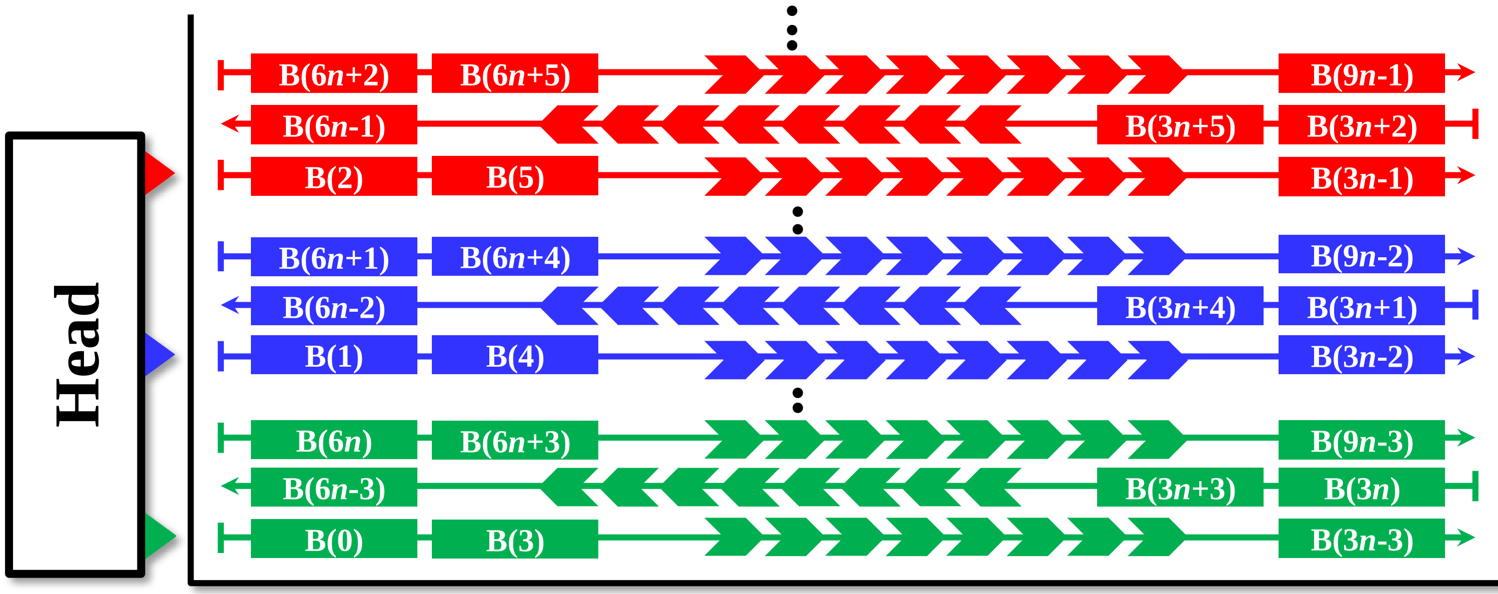
- … types of linear recording … (cont.):

2. Serial Recording: data laid out as sequence of bits along each track, similar to magnetic disks.
   - Data read/written in contagious blocks called records.
   - Records are separated by inter-record gaps.
   - Tape is formatted to assist in locating records.



Serpentine Recording

- Typical recording technique in serial tapes.
  - First set of bits is recorded on tracks in forward direction.
  - When tape reaches end, rd/wr elements are repositioned to new tracks & recording is resumed in opposite direction.
  - Process continues, back and forth, until tape is full.
  - Data still recorded serially along individual tracks, but blocks in sequence are stored in adjacent tracks.
  - Adv.: more tracks than head/elements ➔ increasing capacity.



Linear Tape-Open (LTO)

- Dominant magnetic tape technology today.
- Developed late 1990s as an open source alternative to proprietary tape systems.

	LTO-1	LTO-2	LTO-3	LTO-4	LTO-5	LTO-6	LTO-7	LTO-8
Release date	2000	2003	2005	2007	2010	2012	2015	TBA
Compressed capacity	200 GB	400 GB	800 GB	1600 GB	3000 GB	6250 GB	16 TB	32 TB
Compressed transfer rate	40 MB/s	80 MB/s	160 MB/s	240 MB/s	280 MB/s	400 MB/s	788 MB/s	1.18 GB/s
Linear density (bits/mm)	4880	7398	9638	13250	15142	15143		
Tape tracks	384	512	704	896	1280	2176	3584	
Tape length (m)	609	609	680	820	846	846	960	
Tape width (cm)	1.27	1.27	1.27	1.27	1.27	1.27	1.27	
Write elements	8	8	16	16	16	16	32	
WORM?	No	No	Yes	Yes	Yes	Yes	Yes	Yes
Encryption Capable?	No	No	No	Yes	Yes	Yes	Yes	Yes
Partitioning?	No	No	No	No	Yes	Yes	Yes	Yes

Chapter 7. Input / Output

Outline

- External Devices
  - Types
  - Structure
- I/O Modules
  - Function
  - Structure
- I/O Techniques
  - Programmed I/O
  - Interrupt-Driven I/O
  - Direct Memory Access
- I/O Channels & Processors

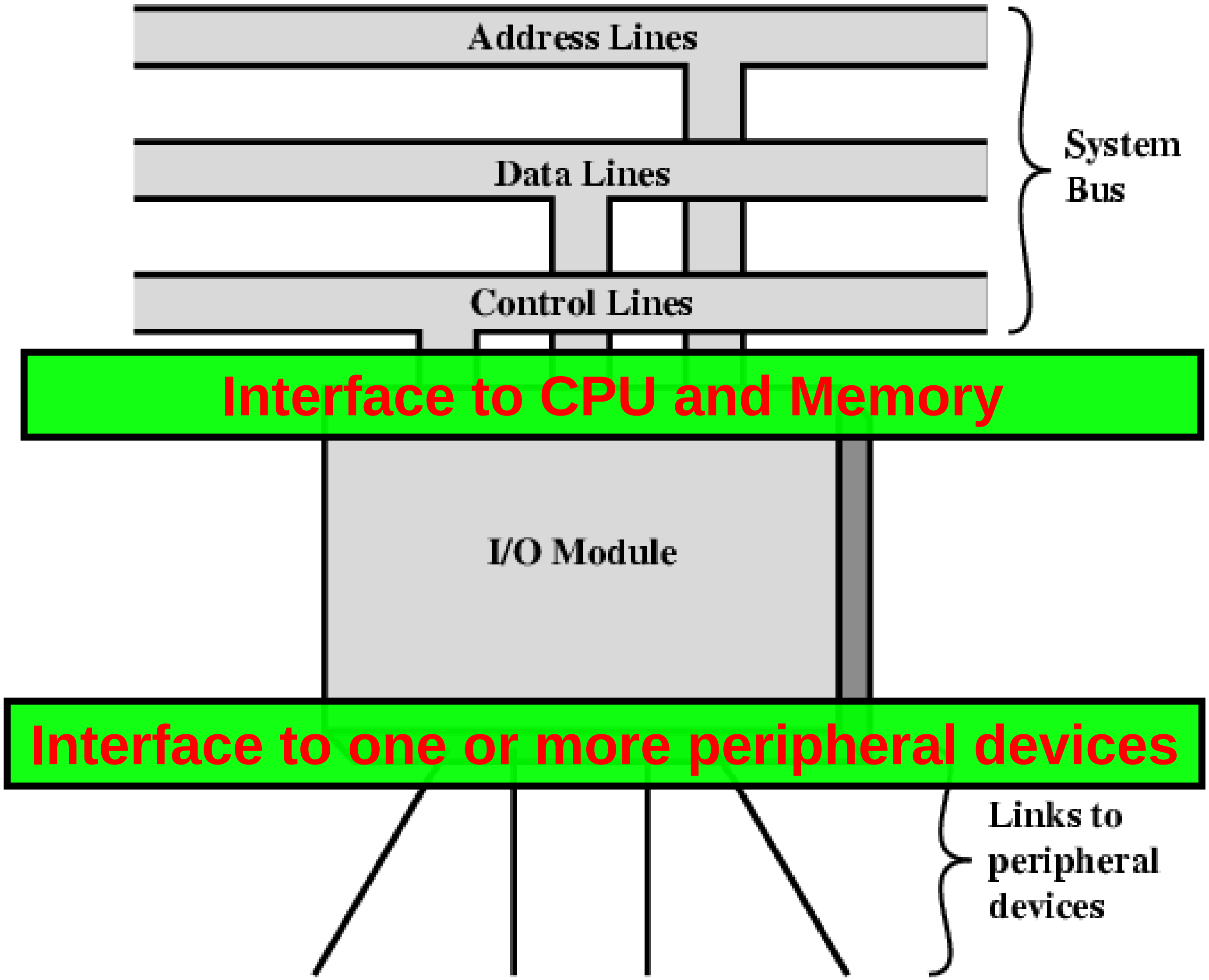
Terms

- Essential Computer Units
  - CPU and Memory
- Peripheral (or External or I/O) devices
  - Any device attached to a computer in order to increase its functionality.
    - Input: keyboard, mouse, scanner, … etc.
    - Output: printers, speakers, … etc.
    - Input and output: hard disk, modem, … etc.
- I/O (Input/Output) Operations
  - Transfer of data to/from computer from/to peripheral device (done by program, operation, or device).
  - Input: from a device to the computer
  - Output: from the computer to a device.

Input/Output Problems

- There is a wide variety of peripherals!
  - Different methods of operation (H/W).
  - Delivering different amounts of data.
  - At different speeds (which are also different from CPU and memory).
  - In different formats (e.g., word length).
- Conclusion: Hard to connect such variety of different devices directly to same Bus!!
- Solution: **I/O Module**

I/O Module



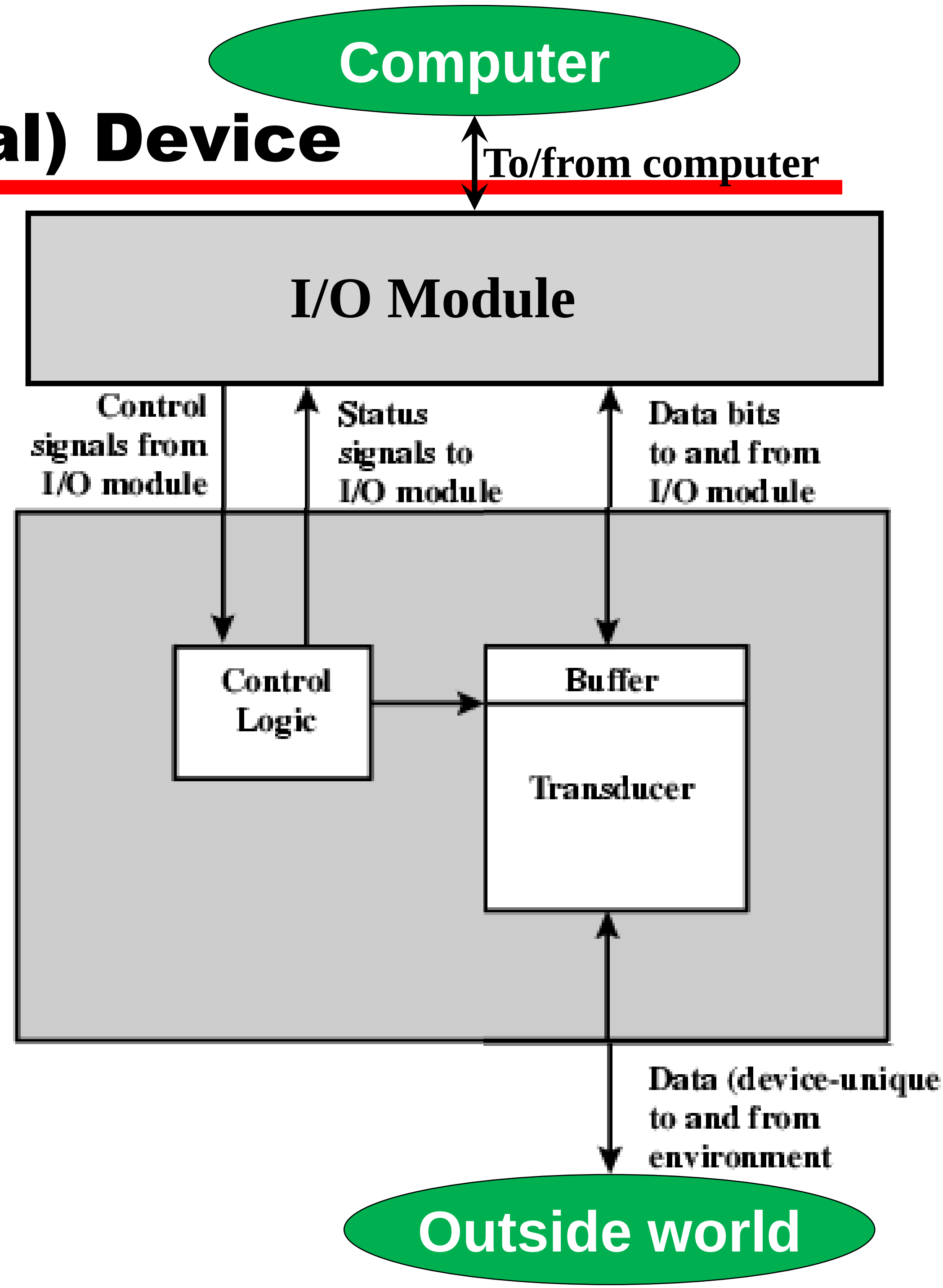
Address Lines
Data Lines
Control Lines
System Bus
Interface to CPU and Memory
I/O Module
Interface to one or more peripheral devices
Links to peripheral devices

Types of Peripherals

- Human readable
  - Screen, printer, keyboard, … etc.
- Machine readable
  - Magnetic disk, tape, … etc.
- Communication
  - Modem, Network Interface Card (NIC), Wireless Network Adaptor, … etc.

Peripheral (External) Device

- Control Signals
  - Send data to module, receive data from module, send status, position disk head.
- Status Signals
  - READY, NOT READY, …
- Buffer
  - Temporarily hold data being transferred. Size: x bytes ➔ x Kbytes!!
- Transducer
  - Converts energy: electrical ←→ other.
- Control logic
  - Controls operation.



Examples: Keyboard/Monitor, and Disk Drive

- Keyboard (input)
  - A key is pressed.
  - Transducer translates signal into ASCII.
  - ASCII is transmitted to I/O module in the computer.
  - Text can be stored as ASCII in the computer.
- Monitor (output)
  - Computer sends ASCII to I/O module. I/O module sends ASCII to external device (monitor).
  - Transducer at the monitor sends electronic signals to display the character.
- Hard Disk Drive (input/output)
  - Head moves in and out across disk surface.
  - Transducer converts magnetic patterns to/from bits.

Functions of I/O Module

1. Control & Timing.
2. CPU Communication.
3. Device Communication.
4. Data Buffering.
5. Error Detection.

1. Control & Timing

- I/O includes control & timing requirement to coordinate the flow of traffic between internal resources (CPU, MM, …) and external devices.
- Ex.: Transfer data from input device to CPU:
  1. CPU checks I/O module device status.
  2. I/O module returns status.
  3. If ready, CPU requests data transfer (command to I/O module).
  4. I/O module gets data from external device.
  5. I/O module transfers data to CPU.
- If transfer goes through bus, each CPU/module interaction may involve 1^+ bus arbitrations.

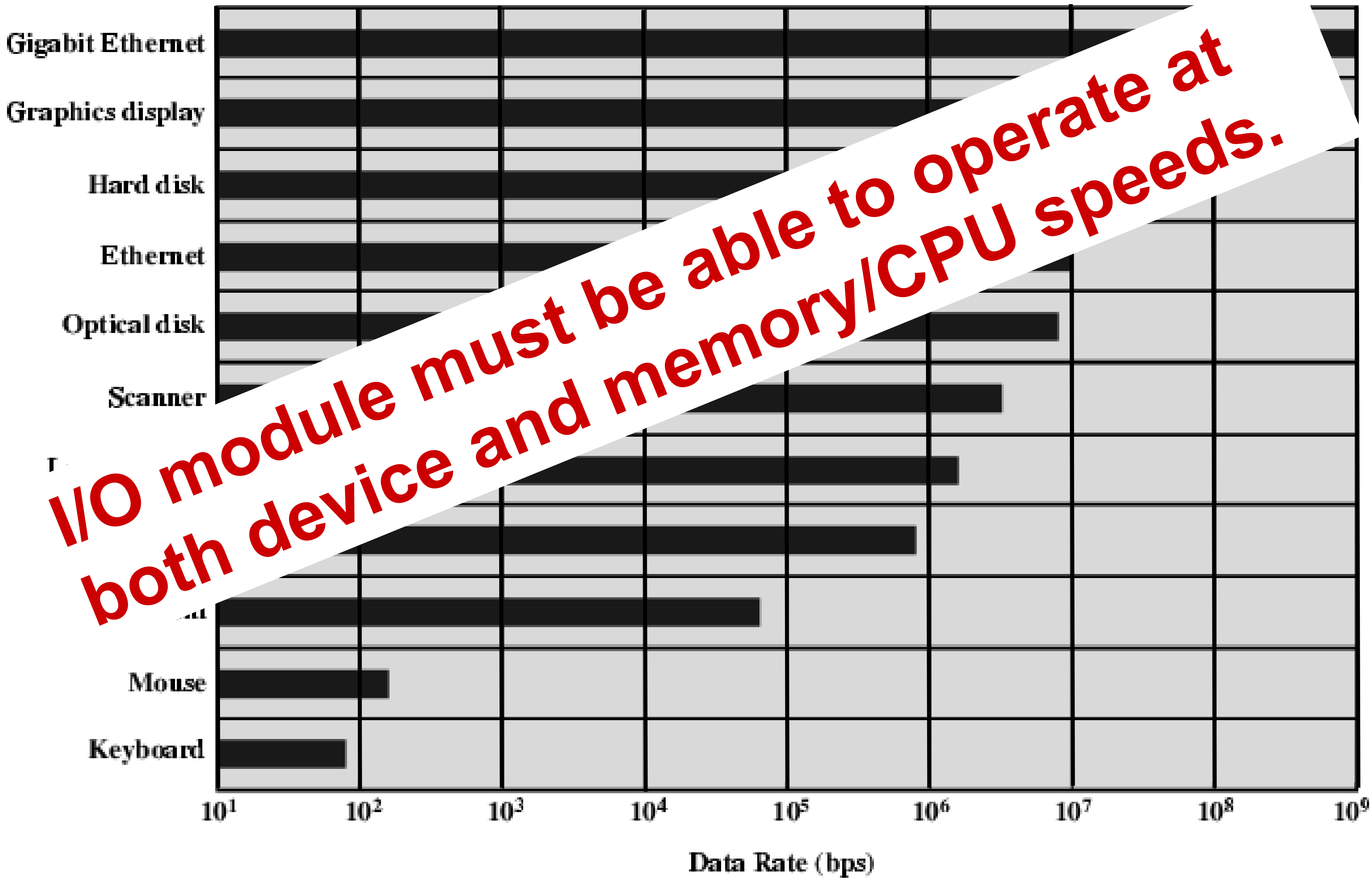
2. CPU Communication

- CPU communication involves the following:
  - Command decoding
    - Module accepts commands from CPU on control lines.
    - Command parameters can be sent over data line.
    - e.g., SEEK track in a disk drive: SEEK command sent on control lines and track # sent on data lines.
  - Address recognition
    - One unique address for each peripheral it controls.
  - Data exchange
    - Between CPU and device over the data bus.
  - Status reporting
    - BUSY, READY, or some error conditions.

3. Device Communication

- I/O module must also be able to do device communication:
  - Commands
  - Status information
  - Data

4. Data Buffering (Speed Mismatch)



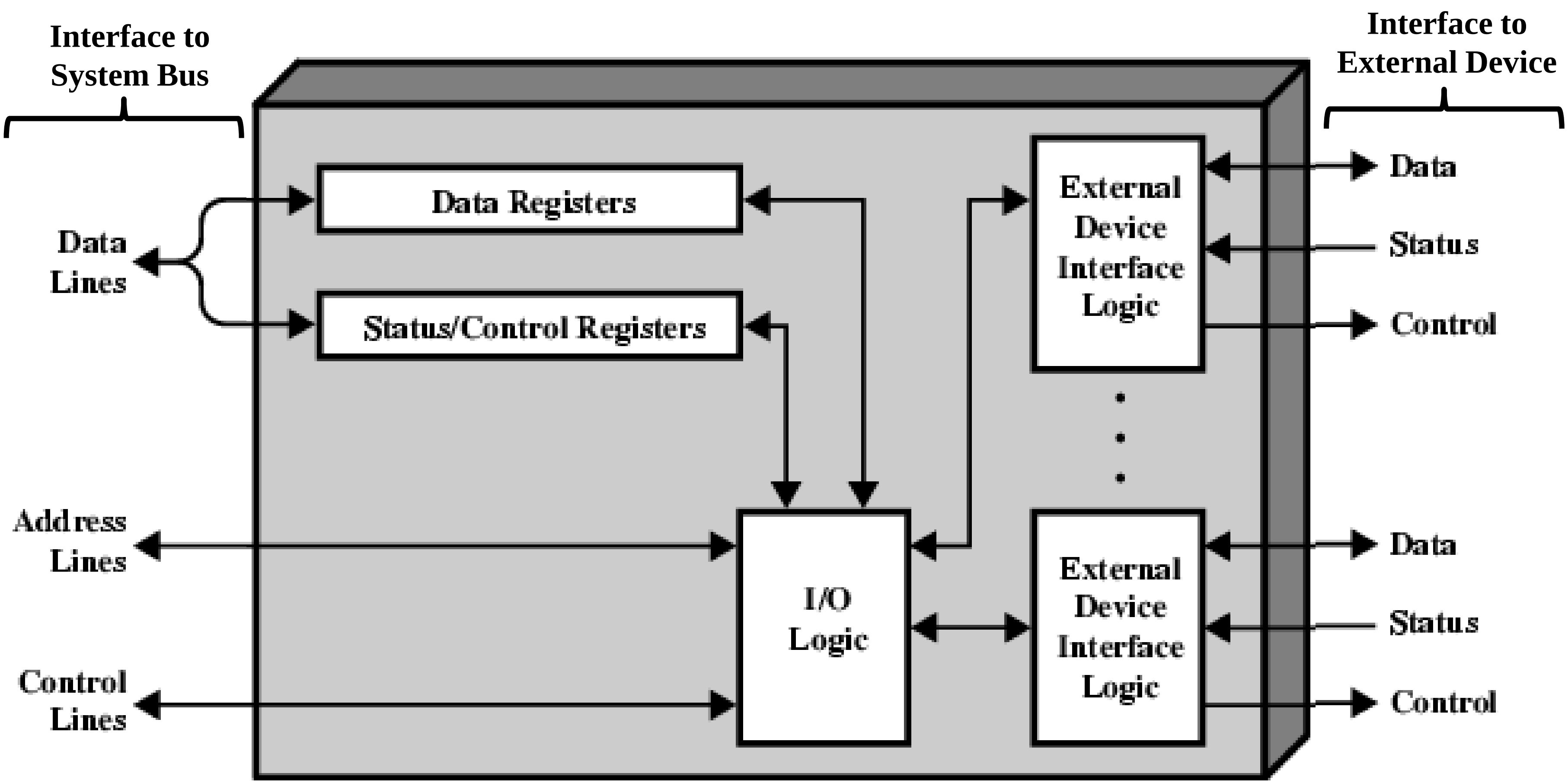
I/O module must be able to operate at both device and memory/CPU speeds.

5. Error Detection

- Mechanical and electrical malfunctions
  - Report to CPU.
  - e.g., paper jam, bad disk sector/track.
- Unintentional changes to transmitted bit pattern
  - Detected using error-detecting codes.
  - e.g., parity bit (ASCII).

I/O Module Structure

- St./Ctrl. registers: hold device status or accept control info from CPU.
- CPU issues commands to I/O module via control lines.
- Some control lines are also used by I/O module for bus arbitration.
- Module controlling more than 1 device has a set of unique addresses.



I/O Module Design Decisions

- I/O module lets CPU view a wide range of devices in a simple-minded way.
- Module may hide device properties from CPU:
  - —Quite complex module design.
  - —Simple CPU commands (e.g., render object).
  - —Referred to as **I/O channel** (**I/O processor**).
  - —Common in mainframes.
- Module may reveal device properties to CPU:
  - —Relatively simple module design.
  - —Detailed CPU commands (e.g., rewind tape).
  - —Referred to as **I/O controller** (**device controller**).
  - —Common in microcomputers.

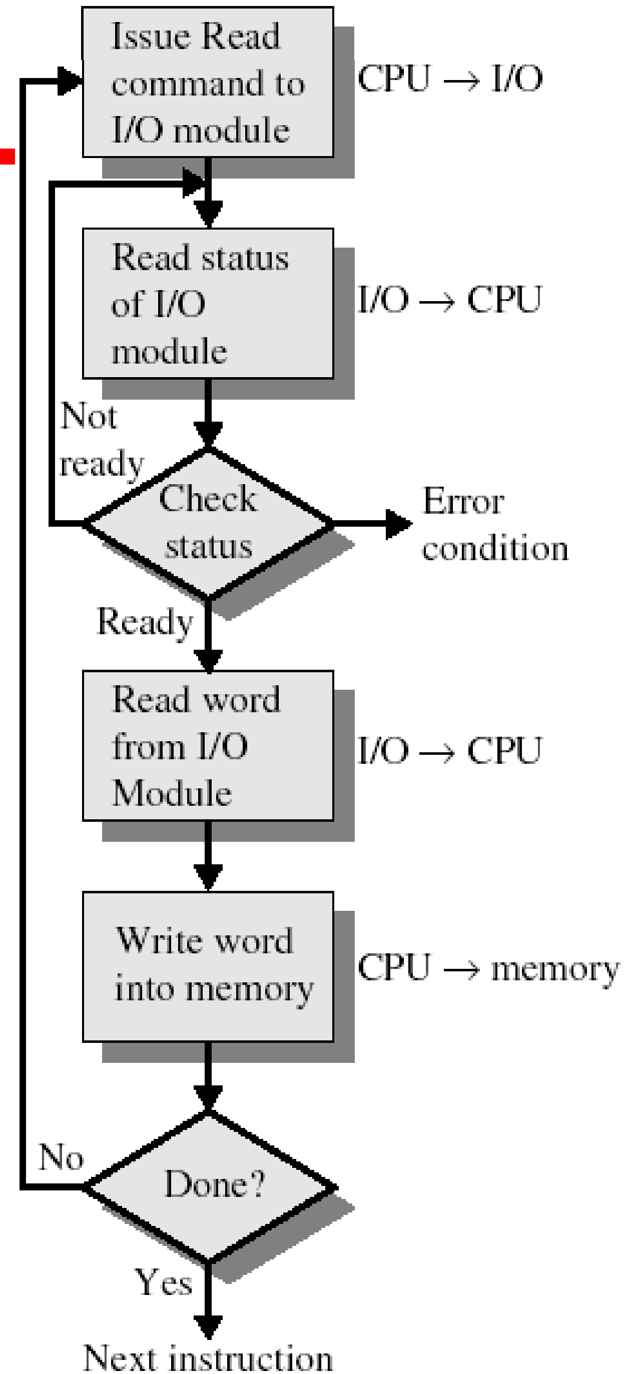
Input Output Techniques

- Programmed I/O.
- Interrupt-driven I/O.
- Direct Memory Access (DMA).

	No Interrupts	Use of Interrupts
I/O-to-memory transfer through CPU	Programmed I/O	Interrupt-driven I/O
Direct I/O-to-memory transfer		Direct Memory Access (DMA)

Programmed I/O

- CPU (program) has direct control over I/O.
  — Issuing commands
  — Sensing status
  — Transferring data
- CPU issues a command to I/O module.
- CPU checks status bits periodically.
  — This process is called: pooling.
- I/O module performs operation.
- I/O module sets status bits.
- CPU transfers data: device ←→ memory.
- Notes:
  — I/O module does not inform CPU directly.
    – I/O module does not interrupt CPU.
  — CPU waits for I/O operation to complete.
    – **Disadvantage**: Wasting CPU time!



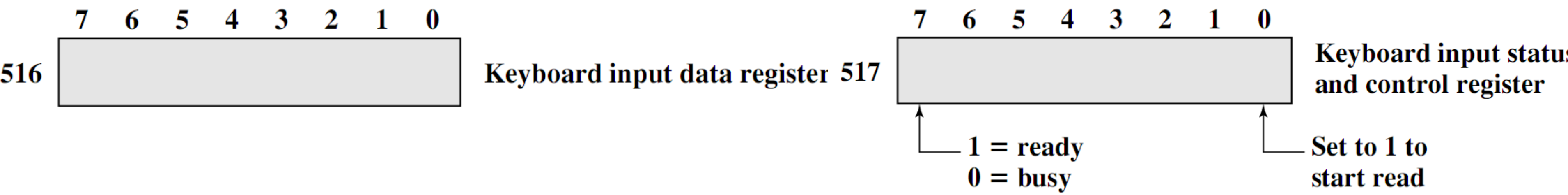
I/O Commands vs. I/O Instructions

- I/O Command
  - Signal: issued by (or Sent from) CPU to I/O module.
  - Types:
    - Control: activate device & tell it what to do (e.g., rewind tape).
    - Test: check status (e.g., is power on? is error detected?)
    - Read/Write: transfer data CPU ←→ buffer ←→ peripheral
- I/O instruction
  - Step in program: fetched from MM & executed by CPU.
  - To execute an I/O instruction: (1) CPU issues address of I/O module & device, (2) CPU issues an I/O command.
  - Instruction form depends on how devices are addressed.
- In programmed I/O, there is a one-to-one mapping between I/O instructions and I/O commands.

Addressing Techniques of I/O Devices

- Two ways to assign addresses to I/O devices:
  1. Memory-mapped I/O
     - Devices & memory share same address space.
     - I/O looks just like memory read/write.
     - No special instructions for I/O ➔ "load", "store", … etc.
     - Bus has one Read & one Write control line.
     - **Pros**: Large selection of memory access instructions.
     - **Cons**: Valuable memory address space is used up!
  2. Isolated I/O
     - Separate address space for devices.
     - Special instructions for I/O ➔ "in", "out", "test", … etc.
     - Need two Rd & two Wr control lines.
     - **Pros**: efficient use of memory address space.
     - **Cons**: Not so many I/O instructions.

I/O Mapping - Example



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

Reading Material

- Stallings, Chapter 6:
  - Pages 215-217
- Stallings, Chapter 7:
  - Pages 222-232