Peripheral Devices

Input Devices

- Keyboard
- Optical input devices
  - Card Reader
  - Paper Tape Reader
  - Bar code reader
  - Digitizer
  - Optical Mark Reader
- Magnetic Input Devices
  - Magnetic Stripe Reader
- Screen Input Devices
  - Touch Screen
  - Light Pen
  - Mouse
- Analog Input Devices

Output Devices

- Card Puncher, Paper Tape Puncher
- CRT
- Printer (Impact, Ink Jet, Laser, Dot Matrix)
- Plotter
- Analog
- Voice

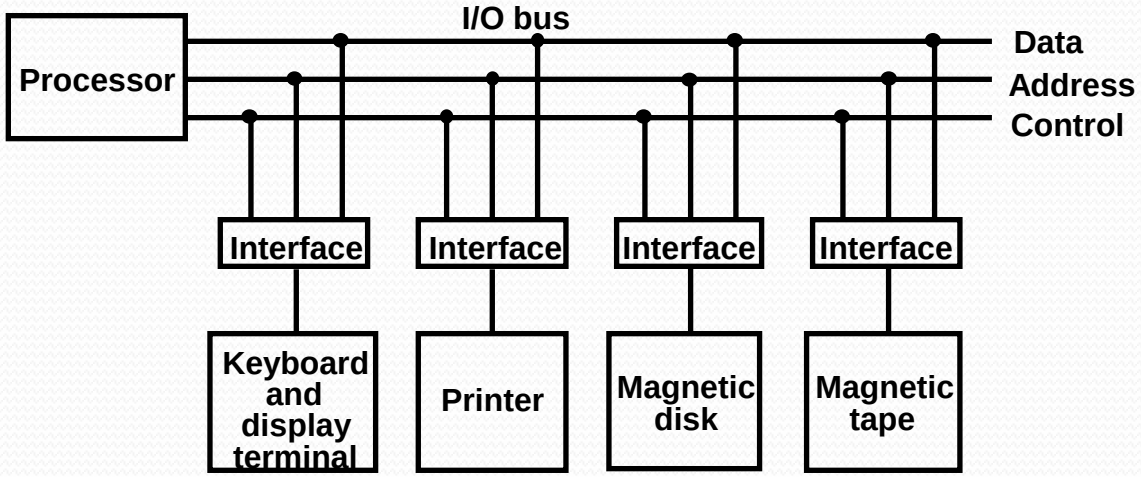
I/O Interface

- **Provides a method for transferring information between internal storage (such as memory and CPU registers) and external I/O devices**
- **Resolves the *differences* between the computer and peripheral devices**
  - **Peripherals - Electromechanical Devices**
  - **CPU or Memory - Electronic Device**
  - **Data Transfer Rate**
    - **Peripherals - Usually slower**
    - **CPU or Memory - Usually faster than peripherals**
      - **Some kinds of Synchronization mechanism may be needed**
  - **Unit of Information**
    - **Peripherals – Byte, Block, …**
    - **CPU or Memory – Word**
  - **Data representations may differ**

Peripherals are usually slower as compared to CPU.

True
False

I/O Bus and Interface



Each peripheral has an interface module associated with it

Interface

- **Decodes the device address (device code)**
- **Decodes the commands (operation)**
- **Provides signals for the peripheral controller**
- **Synchronizes the data flow and supervises the transfer rate between peripheral and CPU or Memory**

Typical I/O instruction

Op. code	Device address	Function code
		(Command)

I/O Bus and Memory Bus

Functions of Buses

- ***MEMORY BUS*** **is for information transfers between CPU and the MM**
- ***I/O BUS*** **is for information transfers between CPU and I/O devices through their I/O interface**
- **Many computers use a common single bus system for both memory and I/O interface units**
  - **Use one common bus but separate control lines for each function**
  - **Use one common bus with common control lines for both functions**
- **Some computer systems use two separate buses, one to communicate with memory and the other with I/O interfaces**
  - **Communication between CPU and all interface units is via a common I/O Bus**
  - **An interface connected to a peripheral device may have a number of *data registers* , a *control register*, and a *status register***
  - **A command is passed to the peripheral by sending to the appropriate interface register**
  - **Function code and sense lines are not needed (Transfer of data, control, and status information is always via the common I/O Bus)**

Isolated vs. Memory Mapped I/O

Isolated I/O

- Separate I/O read/write control lines in addition to memory read/write control lines
- Separate (isolated) memory and I/O address spaces
- Distinct input and output instructions

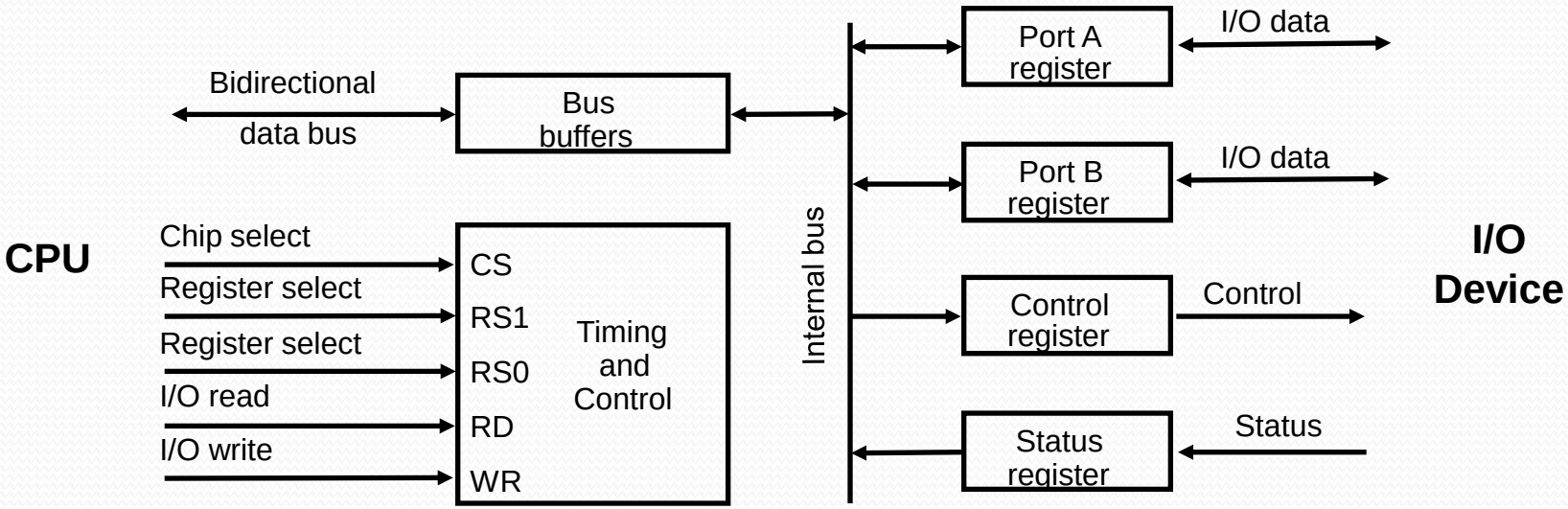
Memory-mapped I/O

- A single set of read/write control lines
  (no distinction between memory and I/O transfer)
- Memory and I/O addresses share the common address space
  -> reduces memory address range available
- No specific input or output instruction
  -> The same memory reference instructions can be used for I/O transfers
- Considerable flexibility in handling I/O operations

…….. Provides signals for the peripheral controller

a) CPU
b) MEMORY
c) REGISTER
d) INTERFACE

I/O Interface



CS	RS1	RS0	Register selected
0	x	x	None - data bus in high-impedence
1	0	0	Port A register
1	0	1	Port B register
1	1	0	Control register
1	1	1	Status register

- CPU initializes(loads) each port by transferring a byte to the Control Register
→ Allows CPU can define the mode of operation of each port
→ *Programmable Port*: By changing the bits in the control register, it is possible to change the interface characteristics