Subject:
Date:

Chap 07 Input/Output problems

I/O Module – is a hardware interface that connects the CPU and main memory to peripheral devices (such as keyboard, printers, disk etc)

Why it is necessary?

1. There are a wide variety of peripherals with various methods of operation.
2. The data transfer rate of peripherals is much slower than that of the processor or memory.
3. data transfer of some peripherals is faster than that of the memory or processor.
4. Peripherals often use different data formats and word lengths than the computer to which they are attached.

It would be impractical to connect peripherals directly to the system bus.

So an I/O module is required.

I/O Module has two major function

1. Interface to the processor and memory
2. Interface to one or more peripheral devices.

Subject:

Date :

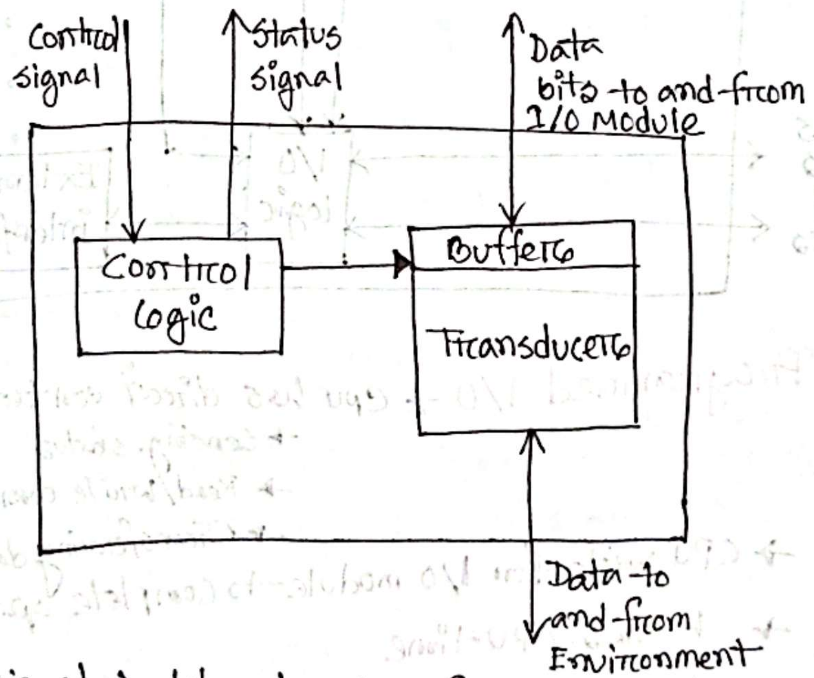
Peripheral – An external device connected to the CPU and memory through I/O Module is called peripheral device.

Classify pheripheral device into 3 categories:

① Human Readable – Suitable for communicating with the computer user. (Screen, printer, keyboard)

② Machine Readable – Suitable for communicating with equipment. (Monitoring and control)

③ Communication – Suitable for communicating with remote device (Modem, NIC)



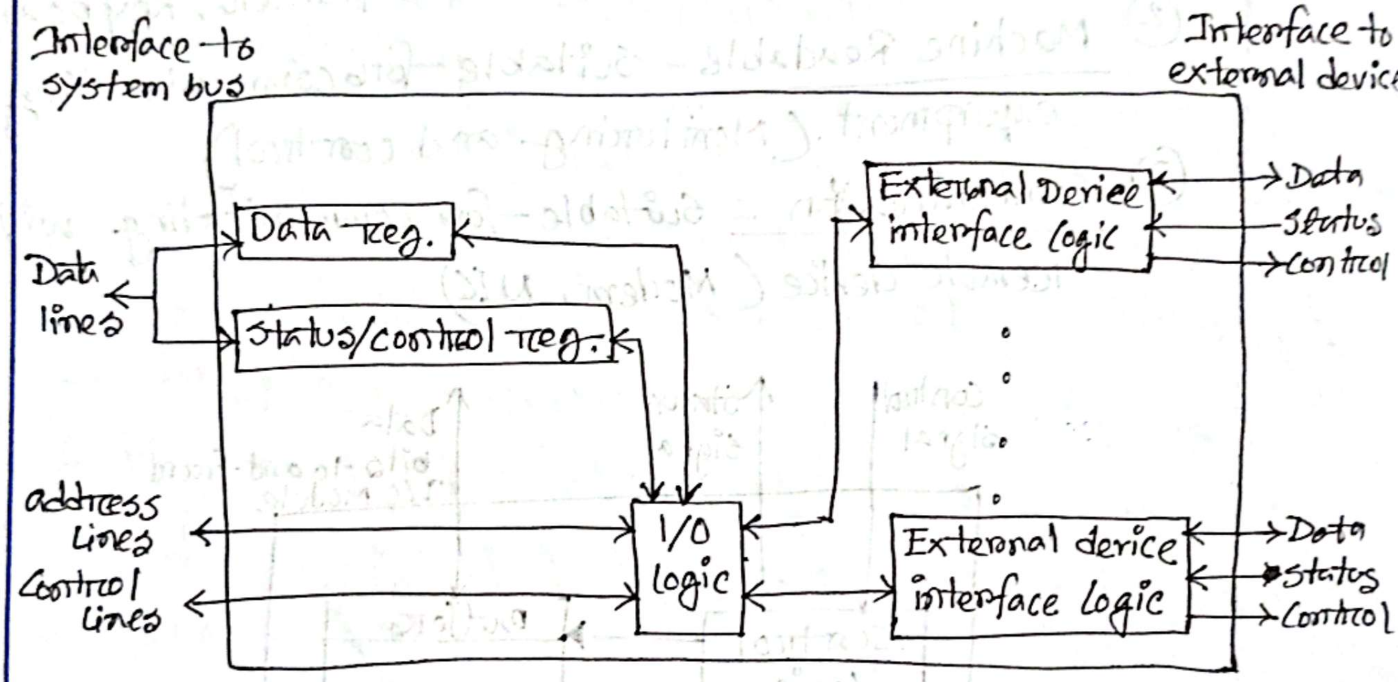
Control signal → determine the function that the device will perform.

Status signal → indicate the state of the device.

Subject:
Date :

I/O Module function :

1. Control and timing
2. Processor communication
3. Device communication
4. Data buffering
5. Error detection



I Programmed I/O → CPU has direct control over I/O
→ Sensing status
→ Read/write command
→ Transferring data

→ CPU waits for I/O module to complete operation

→ Wastes CPU time

II. CPU requests I/O operation → I/O module performs operation
→ I/O module sets status bits → CPU checks status bits periodically
→ I/O module does not inform CPU directly. → I/O module does not interrupt CPU → CPU may wait or come back later.

Subject:.................................
Date :.................................

- I/O Commands – there are 4 I/O commands that an I/O module may receive when addressed by processor.

(1) Control – Used to activate a peripheral and tell what to do

(2) Test – Used to test various status conditions

(3) Read – Causes the I/O module to obtain an item of data from the peripheral

(4) Write – Causes the I/O module to take an item of data from the data bus

- I/O Mapping

(1) Memory mapped I/O – Device and memory share an address space
  - I/O Looks Just Like memory read/write
  - No specific command for I/O

(2) Isolated I/O – Separate address space
  - Need I/O or memory select line
  - Special commands for I/O

- Interrupt Driven I/O – Overcomes CPU wait
  - No repeated CPU checking of device
  - I/O module interrupts when ready

- CPU issues read command → I/O module gets data from peripheral, CPU does other work → I/O module interrupt CPU → CPU requests data → module transfers data

Subject:

Date :

Design Issue

- ① How does the processor determine which device issued interrupt
- ② if multiple interrupt occurred, how does the processor decide which one to proce

4 general categories of techniques are in common use:

① Multiple interrupt lines between the processor and the I/O modules (impractical)

② Software poll, when the processor detects an interrupt, it branches to an interrupt-service routine whose job it is to poll each I/O module to determine which module caused the interrupt. (time consuming)

③ Daisy Chain - CPU when senses an interrupt, it send a interrupt acknowledge, the requesting module responds by placing a vector on the data lines,

Vector → address of the ~~CPU~~ I/O module or unique identifier.

④ Bus Master - I/O module must first gain control of the bus, Only one module can raise the line at a time.

The requesting module than places its vector on the data lines.

~~When more 1~~

Subject:

Date :

□ PC BUS –

→ 80X86 has 1 interrupt line

→ 8086 use 8259A interrupt controller

→ 8259A has 8 interrupt lines

8259A accepts interrupts → determines priority → raises INTR line → CPU acknowledges → 8259A put correct vector on data bus → CPU process interrupt