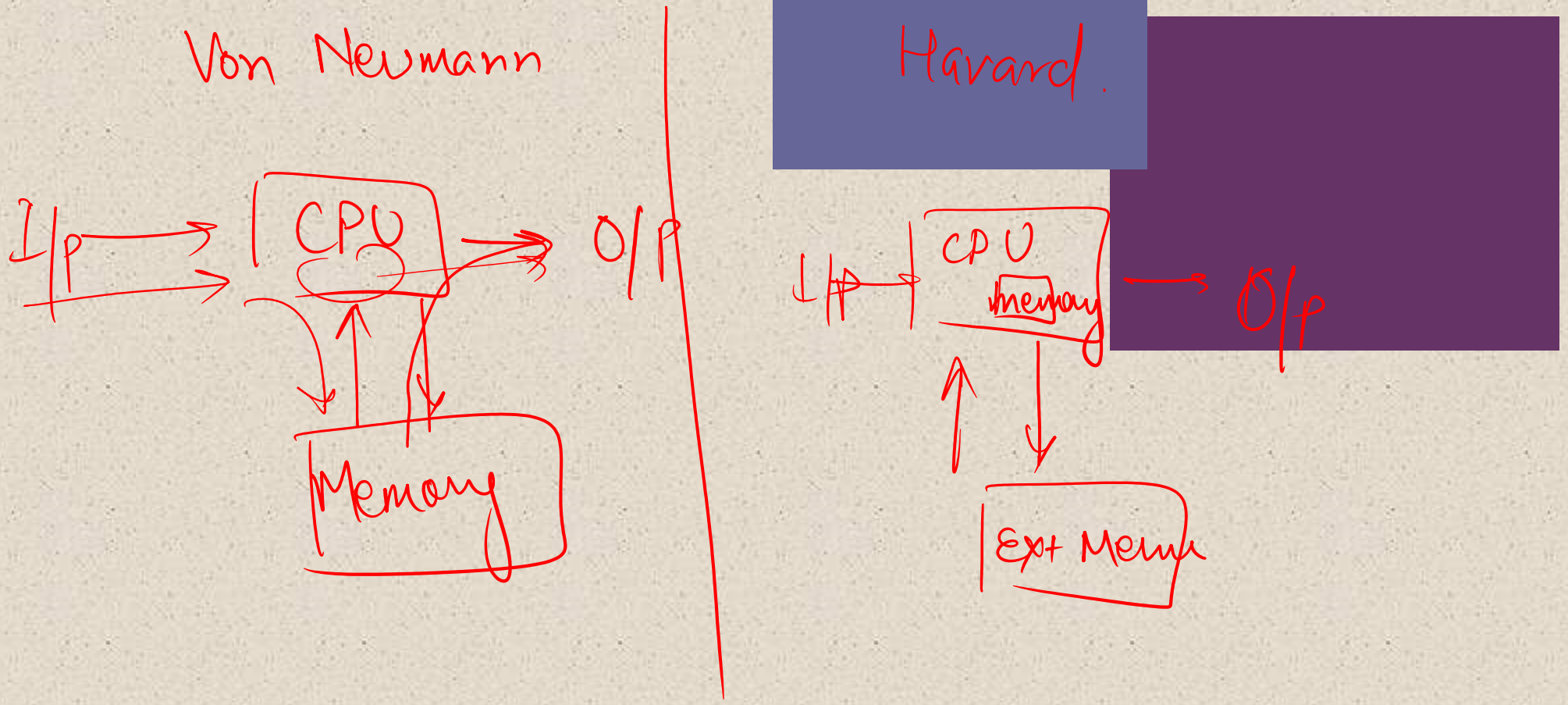
Input/Output



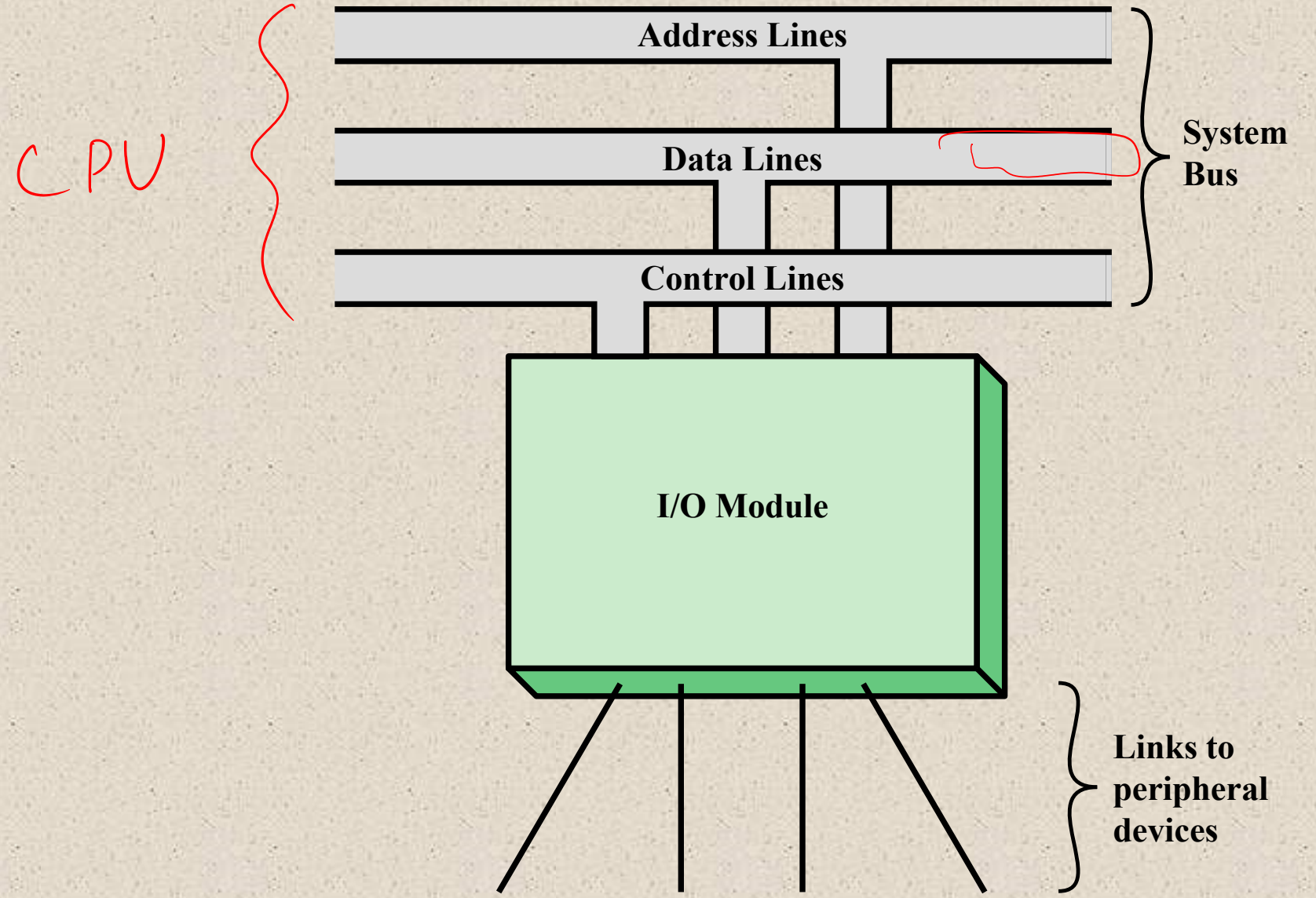


Figure 7.1 Generic Model of an I/O Module

External Devices

- Provide a means of exchanging data between the external environment and the computer
- Attach 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
- *Peripheral device*
  - An external device connected to an I/O module

Three categories:

- Human readable
  - Suitable for communicating with the computer user
  - Video display terminals (VDTs), printers
- Machine readable
  - Suitable for communicating with equipment
  - Magnetic disk and tape systems, sensors and actuators
- Communication
  - Suitable for communicating with remote devices such as a terminal, a machine readable device, or another computer

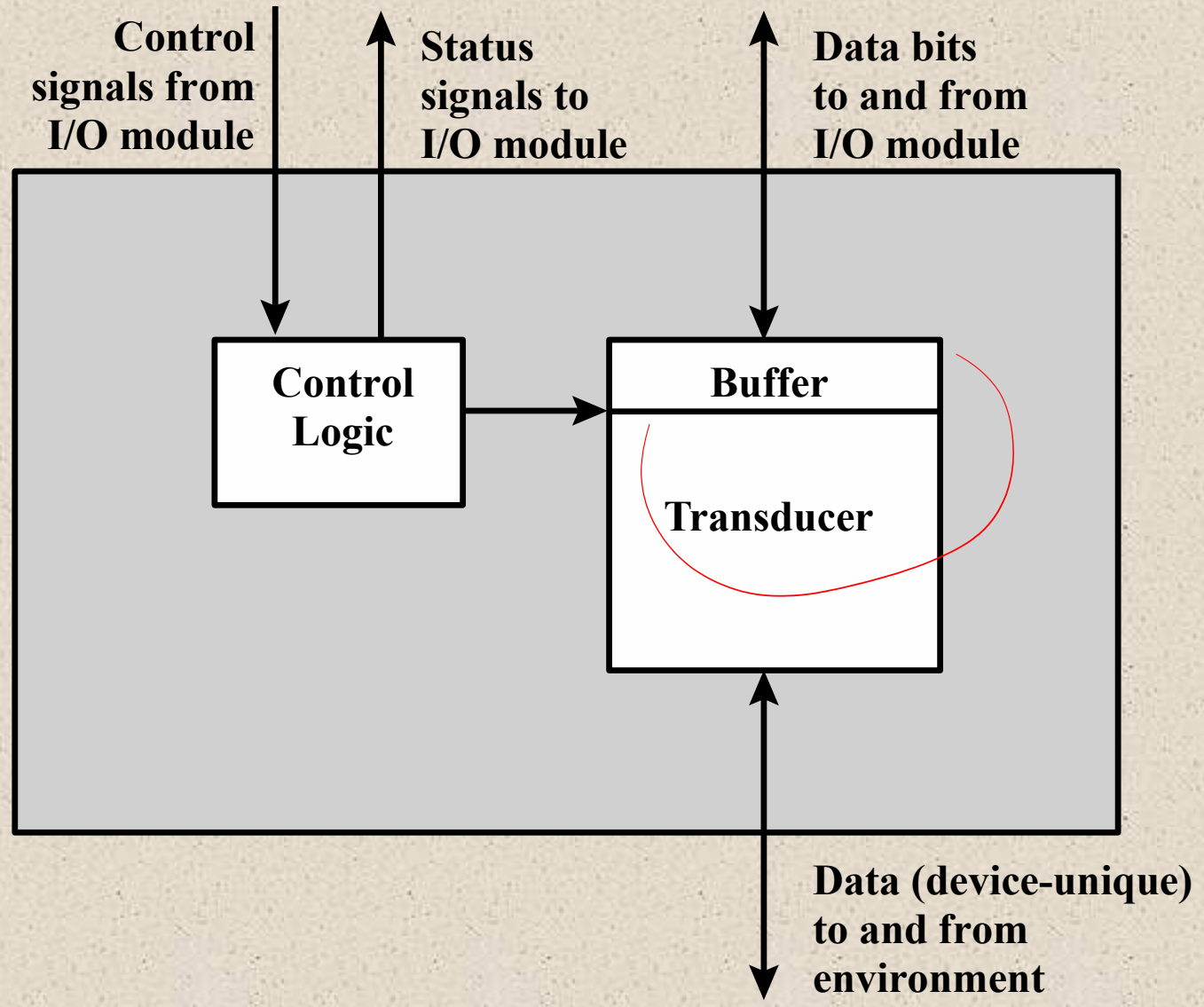


Figure 7.2 Block Diagram of an External Device

Keyboard/Monitor

Most common means of computer/user interaction

User provides input through the keyboard

The monitor displays data provided by the computer

International Reference Alphabet (IRA)

- Basic unit of exchange is the character
  - Associated with each character is a code
  - Each character in this code is represented by a unique 7-bit binary code
    - 128 different characters can be represented
- Characters are of two types:
  - Printable
    - Alphabetic, numeric, and special characters that can be printed on paper or displayed on a screen
  - Control
    - Have to do with controlling the printing or displaying of characters
    - Example is carriage return
    - Other control characters are concerned with communications procedures

Keyboard Codes

- When the user depresses a key it generates an electronic signal that is interpreted by the transducer in the keyboard and translated into the bit pattern of the corresponding IRA code
- This bit pattern is transmitted to the I/O module in the computer
- On output, IRA code characters are transmitted to an external device from the I/O module
- The transducer interprets the code and sends the required electronic signals to the output device either to display the indicated character or perform the requested control function

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

Control and timing

- Coordinates the flow of traffic between internal resources and external devices

Processor communication

- Involves command decoding, data, status reporting, address recognition

Device communication

- Involves commands, status information, and data

Data buffering

- Performs the needed buffering operation to balance device and memory speeds

Error detection

- Detects and reports transmission errors

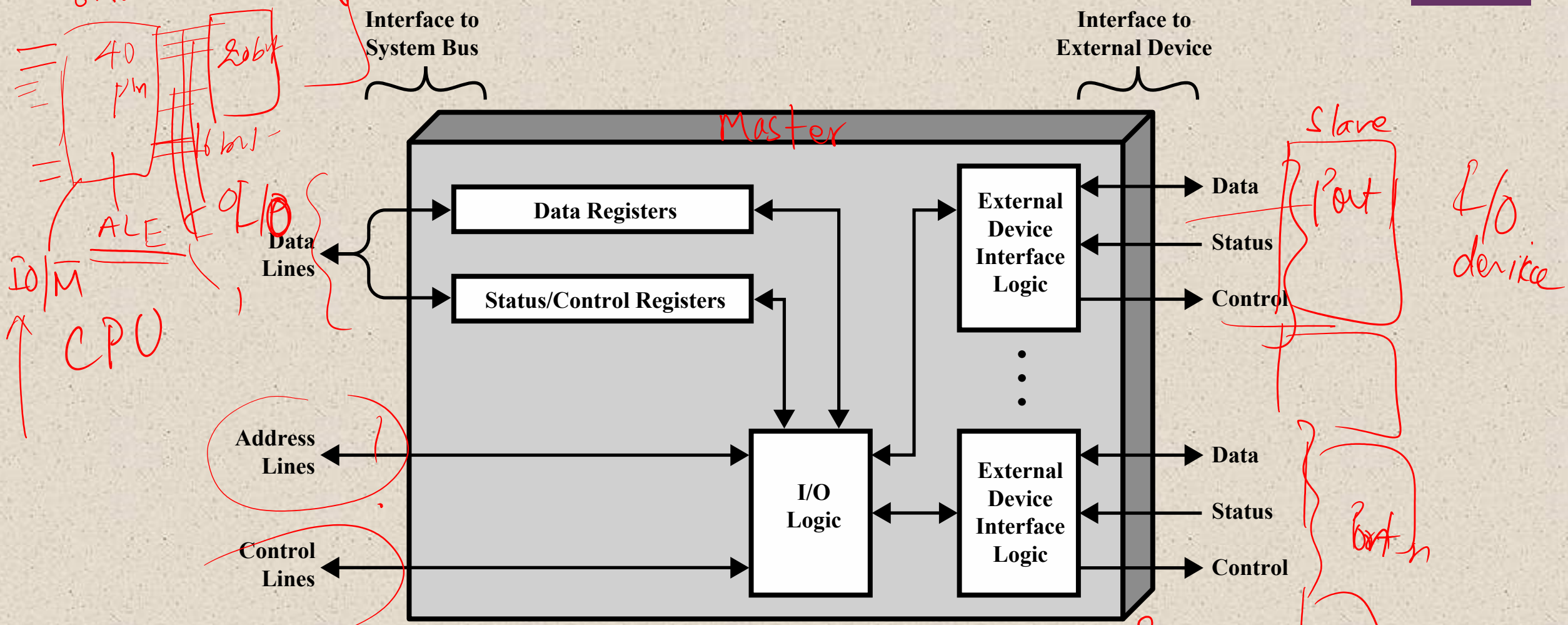


Figure 7.3 Block Diagram of an I/O Module

I/O Instructions

With programmed I/O there is a close correspondence between the I/O-related instructions that the processor fetches from memory and the I/O commands that the processor issues to an I/O module to execute the instructions

The form of the instruction depends on the way in which external devices are addressed

Each I/O device connected through I/O modules is given a unique identifier or address

When the processor issues an I/O command, the command contains the address of the desired device

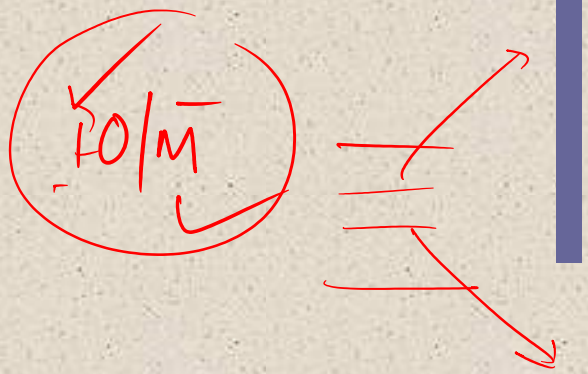
Thus each I/O module must interpret the address lines to determine if the command is for itself

Memory-mapped I/O

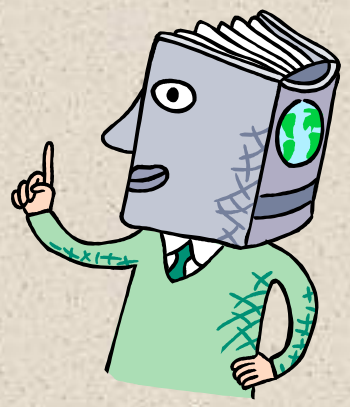
There is a single address space for memory locations and I/O devices

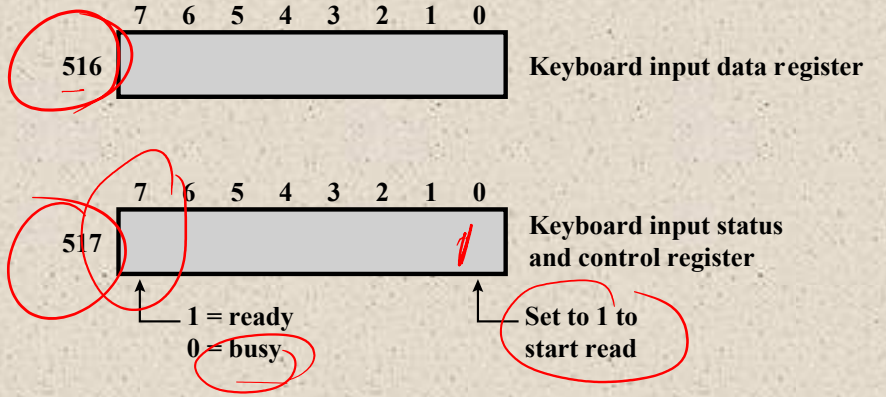
A single read line and a single write line are needed on the bus

I/O Mapping Summary



- Memory mapped I/O
  - Devices and memory share an address space
  - I/O looks just like memory read/write
  - No special commands for I/O
    - Large selection of memory access commands available
- Isolated I/O
  - Separate address spaces
  - Need I/O or memory select lines
  - Special commands for I/O
    - Limited set





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

Figure 7.5 Memory-Mapped and Isolated I/O

Programmed I/O

Three techniques are possible for I/O operations:

- Programmed I/O
  - Data are exchanged between the processor and the I/O module
  - Processor executes a program that gives it direct control of the I/O operation
  - When the processor issues a command it must wait until the I/O operation is complete
  - If the processor is faster than the I/O module this is wasteful of processor time
- Interrupt-driven I/O
  - 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
- Direct memory access (DMA)
  - The I/O module and main memory exchange data directly without processor involvement

Table 7.1
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)

I/O Commands

- There are four types of I/O commands that an I/O module may receive when it is addressed by a processor:

1) Control

- used to activate a peripheral and tell it what to do

2) Test

- used to test various status conditions associated with an I/O module and its peripherals

3) Read

- causes the I/O module to obtain an item of data from the peripheral and place it in an internal buffer

4) Write

- causes the I/O module to take an item of data from the data bus and subsequently transmit that data item to the peripheral

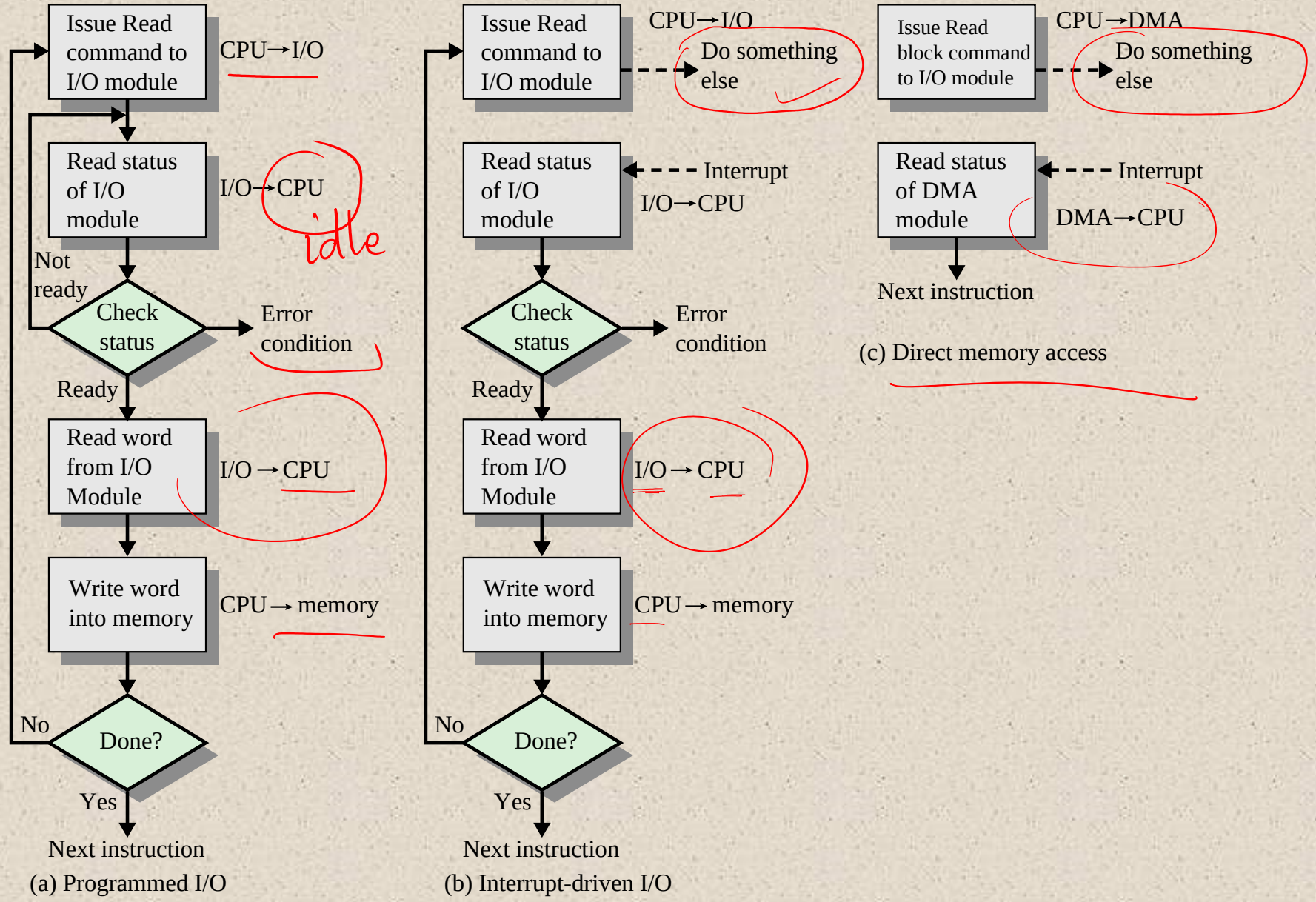


Figure 7.4 Three Techniques for Input of a Block of Data

Interrupt-Driven I/O

The problem with programmed I/O is that the processor has to wait a long time for the I/O module to be ready for either reception or transmission of data

An alternative is for the processor to issue an I/O command to a module and then go on to do some other useful work

The I/O module will then interrupt the processor to request service when it is ready to exchange data with the processor

The processor executes the data transfer and resumes its former processing

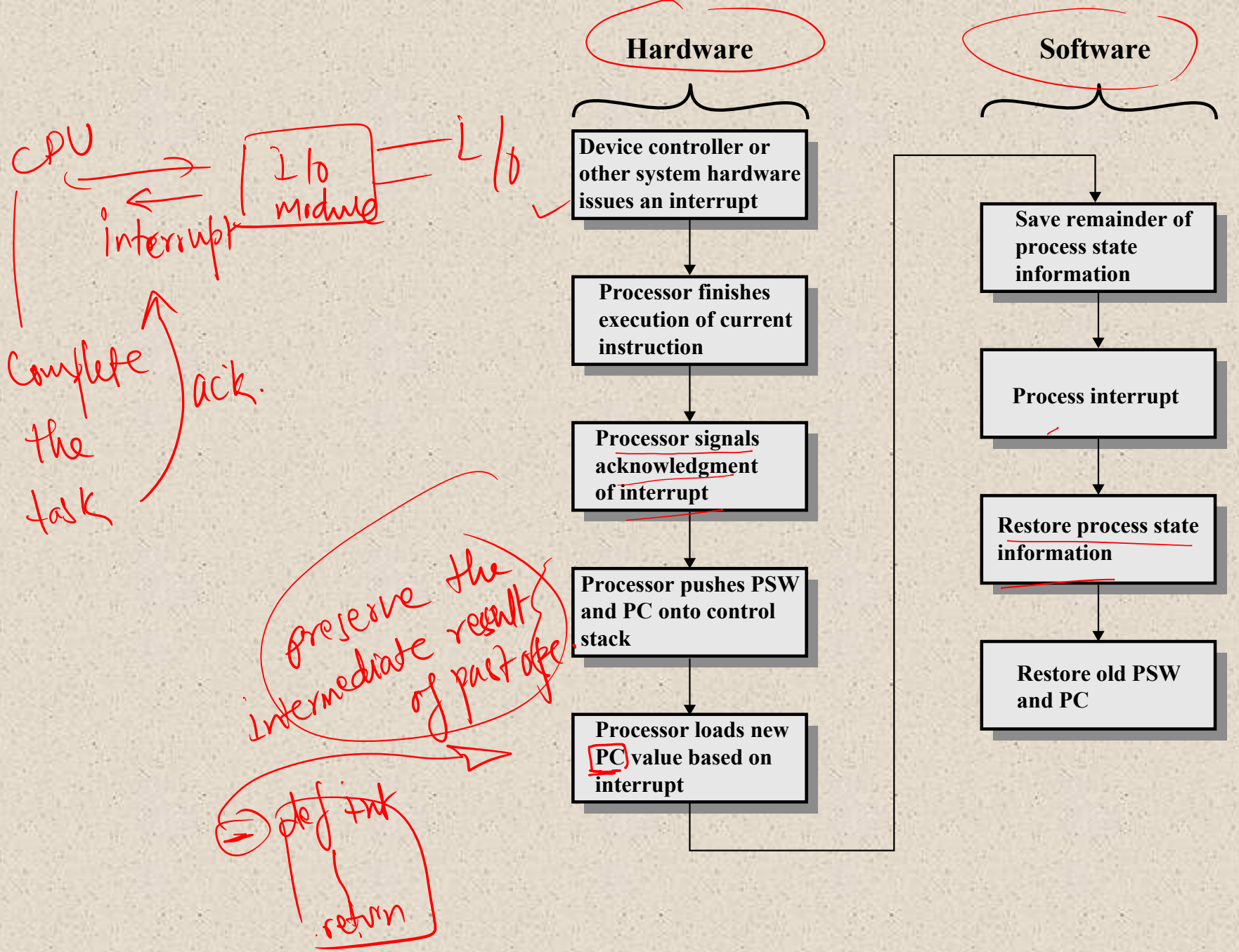


Figure 7.6 Simple Interrupt Processing

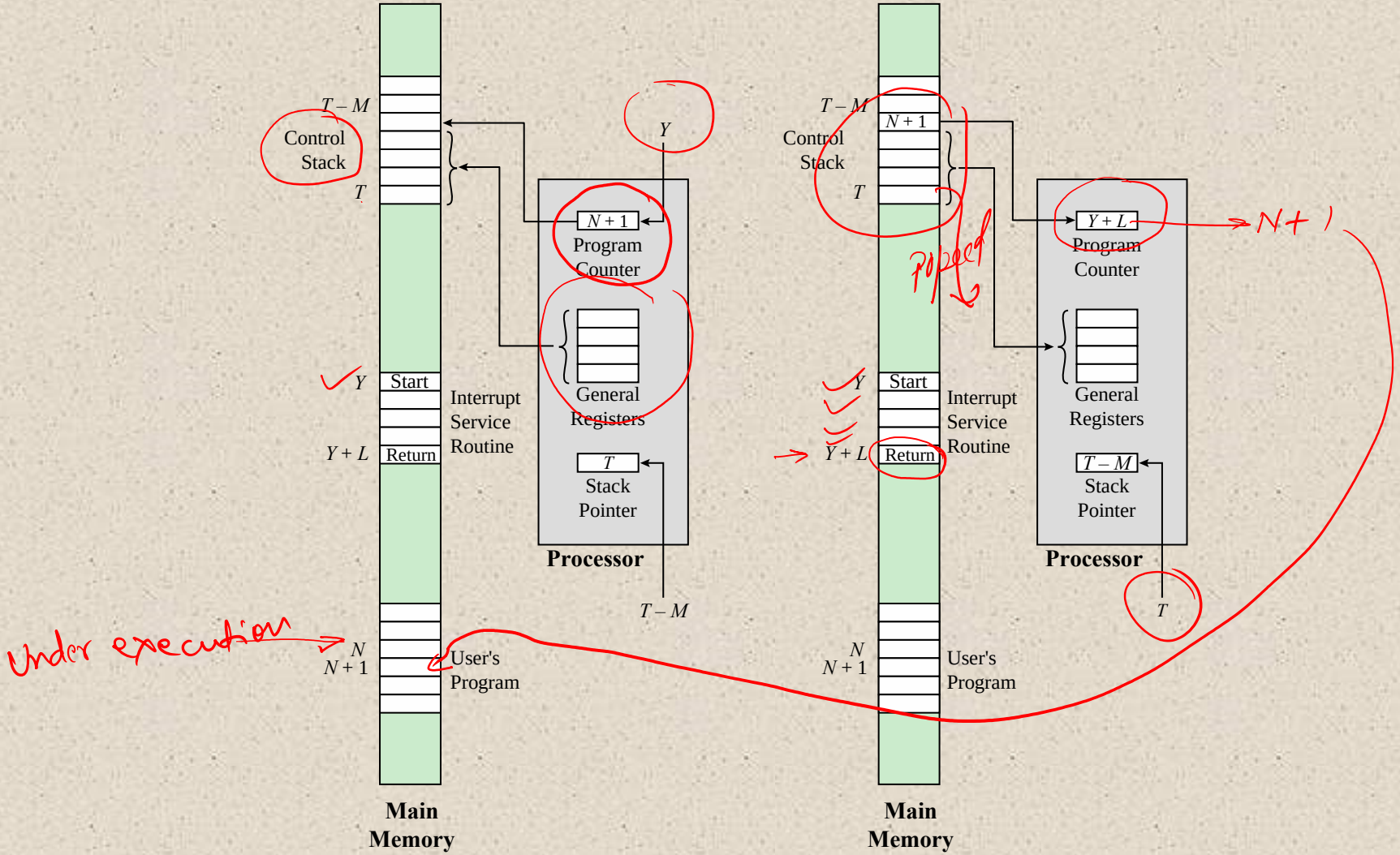
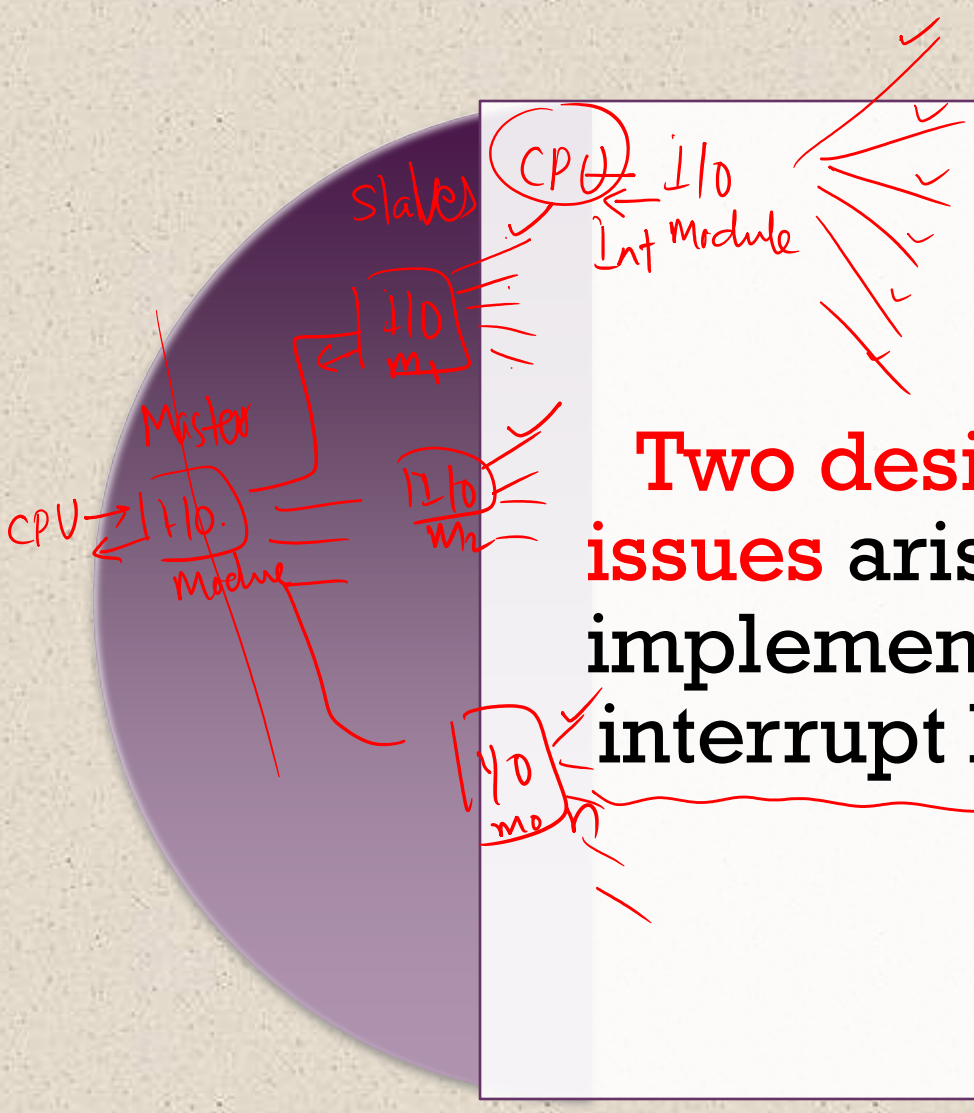


Figure 7.7 Changes in Memory and Registers for an Interrupt

Design Issues

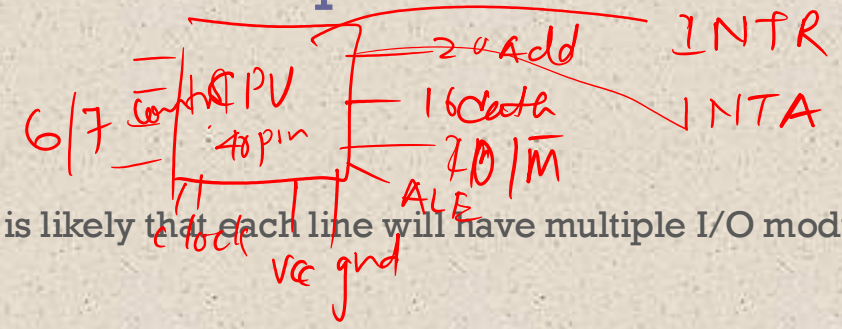


Two design issues arise in implementing interrupt I/O:

- Because there will be multiple I/O modules how does the processor determine which device issued the interrupt?
- If multiple interrupts have occurred how does the processor decide which one to process?

Device Identification

Four general categories of techniques are in common use:



- **Multiple interrupt lines**
  - Between the processor and the I/O modules
  - Most straightforward approach to the problem
  - Consequently even if multiple lines are used, it is likely that each line will have multiple I/O modules attached to it
- **Software poll**
  - When processor detects an interrupt it branches to an interrupt-service routine whose job is to poll each I/O module to determine which module caused the interrupt
  - Time consuming
- **Daisy chain (hardware poll, vectored)**
  - The interrupt acknowledge line is daisy chained through the modules
  - Vector – address of the I/O module or unique identifier. A device requesting an interrupt identifies itself directly by sending a special code to the processor over the bus. This enables the processor to identify the device that generated the interrupt.
  - Vectored interrupt – processor uses the vector as a pointer to the appropriate device-service routine, avoiding the need to execute a general interrupt-service routine first
- **Bus arbitration (vectored)**
  - An I/O module must first gain control of the bus before it can raise the interrupt request line
  - When the processor detects the interrupt it responds on the interrupt acknowledge line
  - Then the requesting module places its vector on the data lines

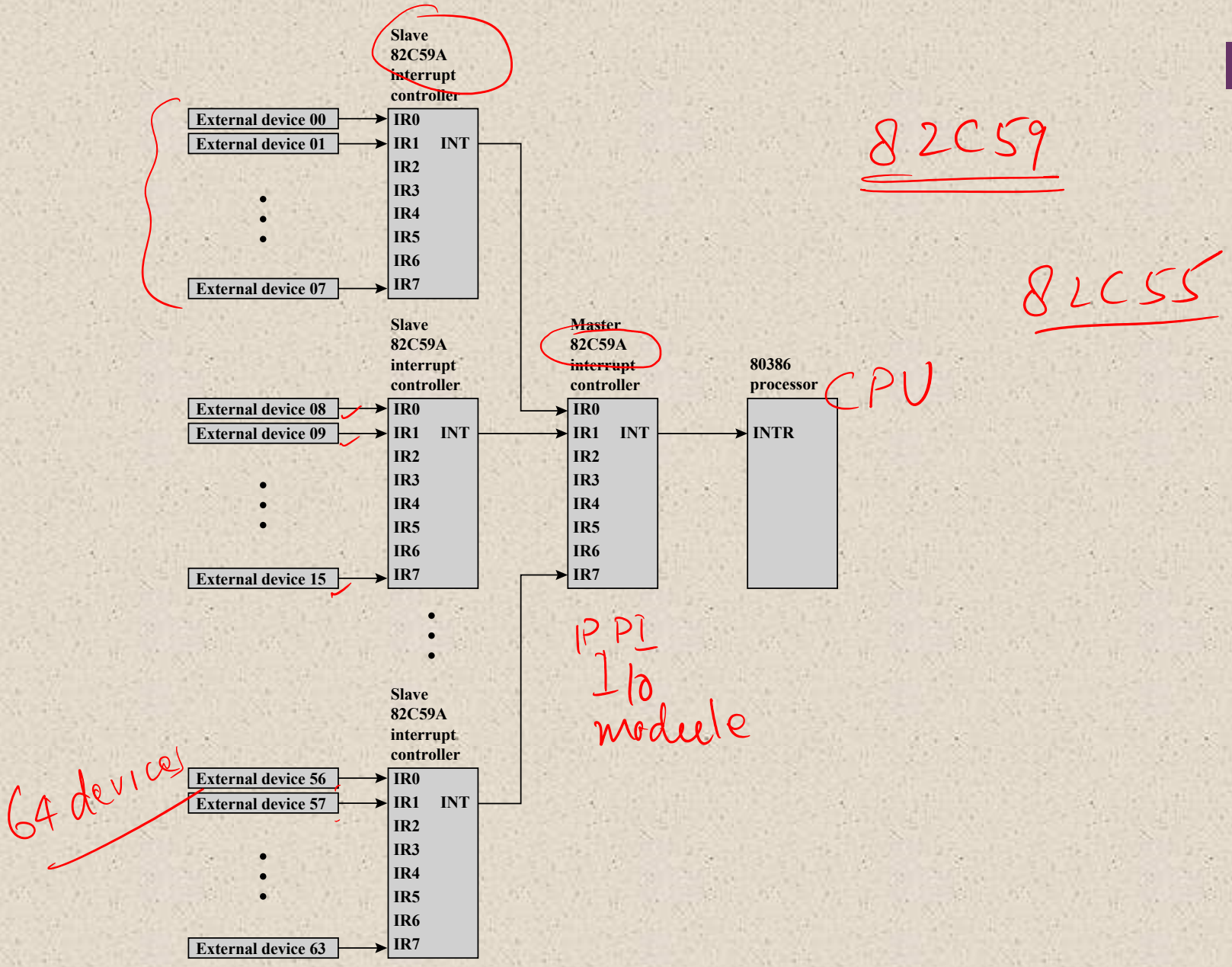


Figure 7.8 Use of the 82C59A Interrupt Controller

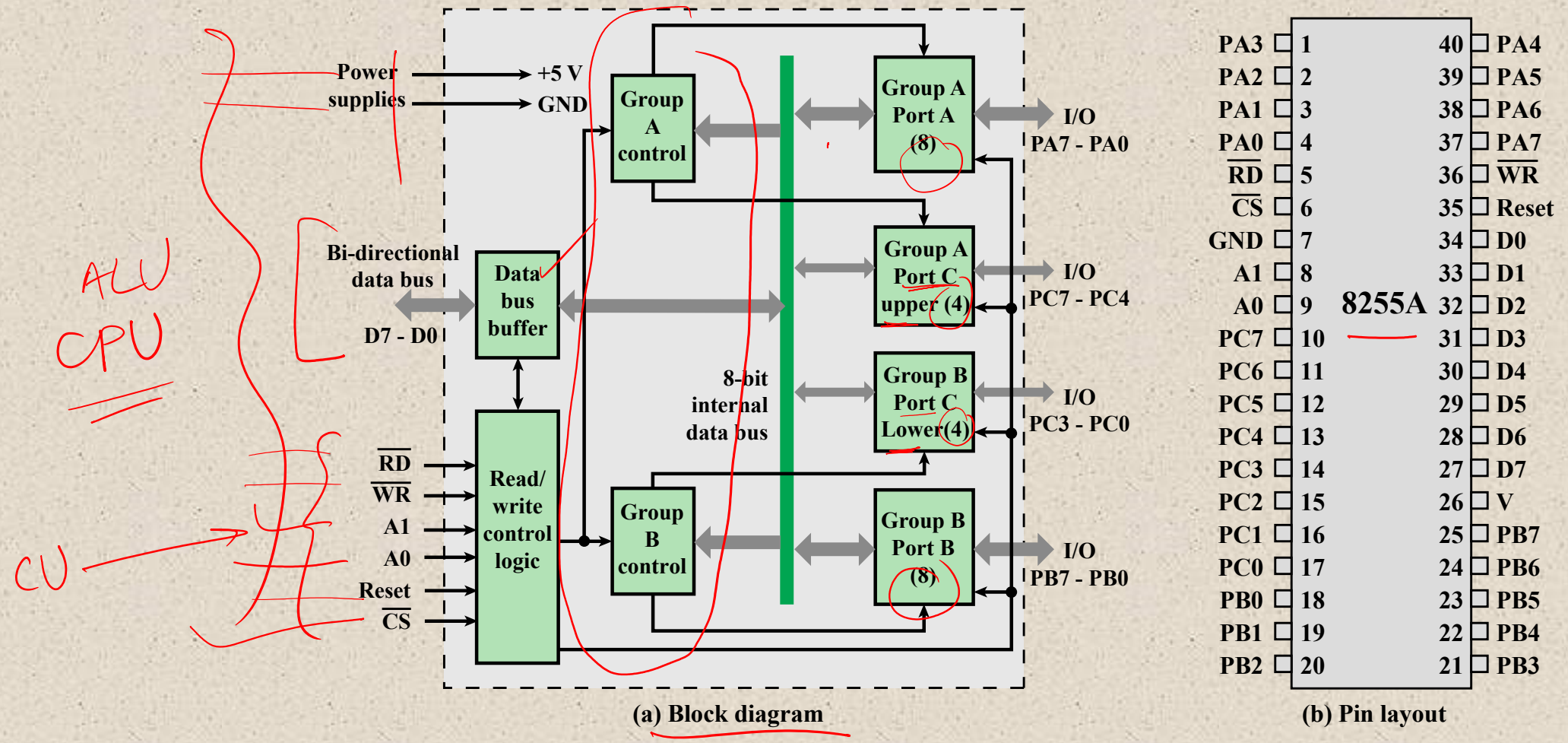
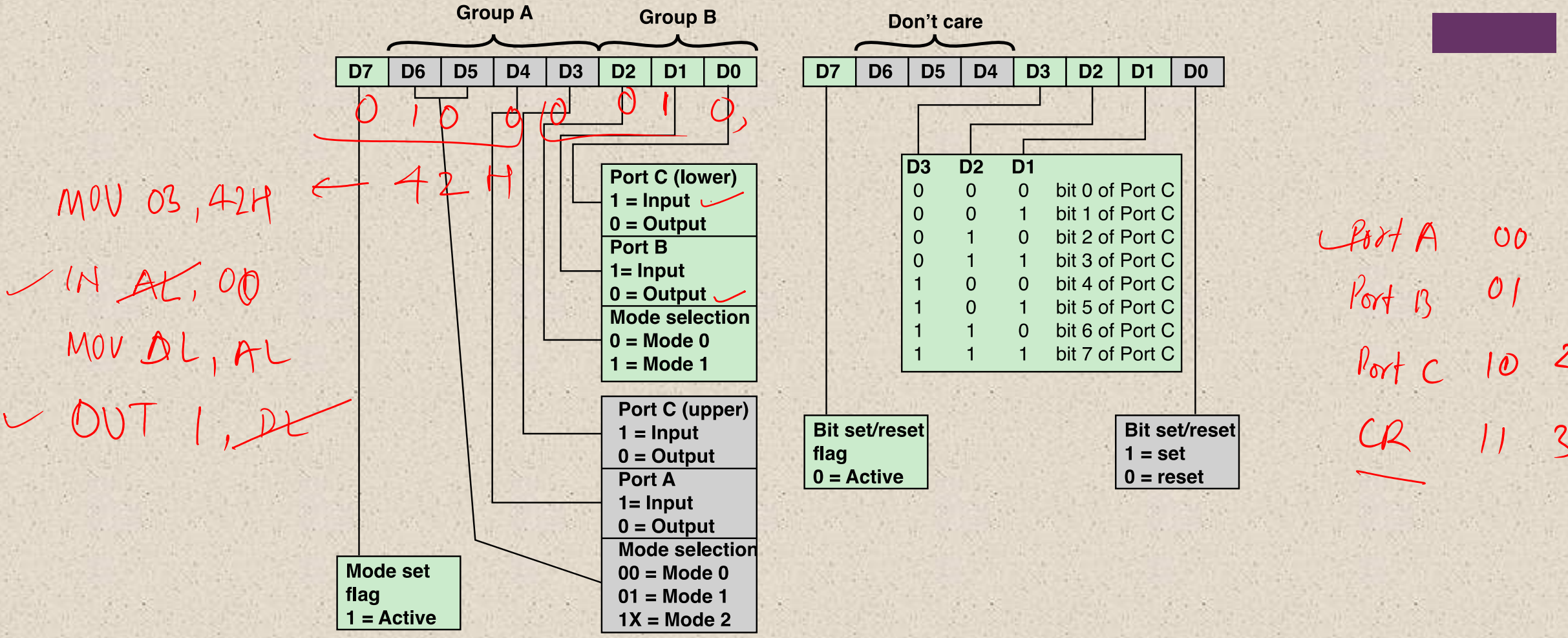


Figure 7.9 The Intel 8255A Programmable Peripheral Interface



(a) Mode definition of the 8255 control register to configure the 8255

(b) Bit definitions of the 8255 control register to modify single bits of port C

Figure 7.10 The Intel 8255A Control Word

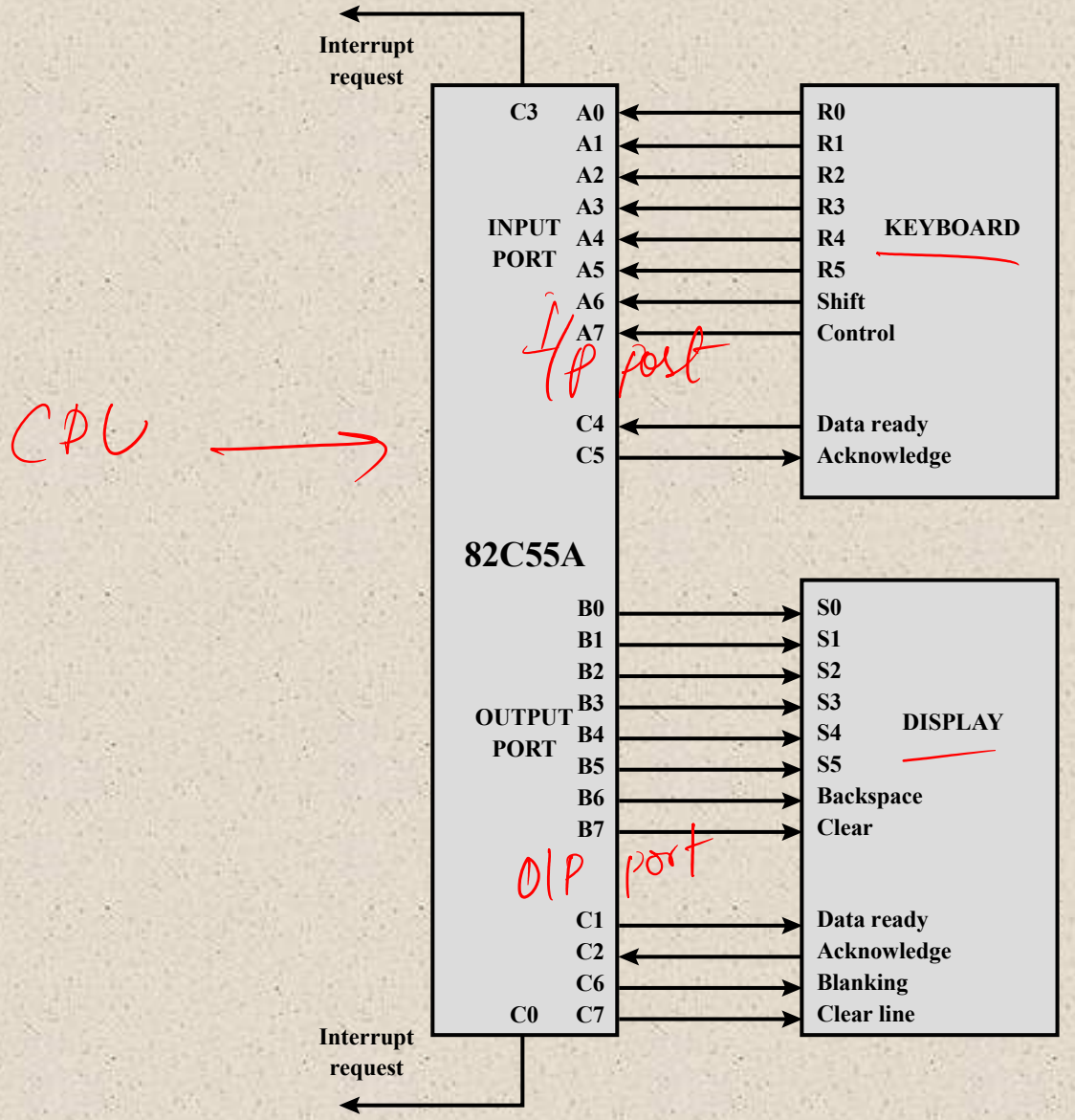


Figure 7.11 Keyboard/Display Interface to 82C55A

Drawbacks of Programmed and Interrupt-Driven I/O

- Both forms of I/O suffer from two inherent drawbacks:

  1) The I/O transfer rate is limited by the speed with which the processor can test and service a device

  2) The processor is tied up in managing an I/O transfer; a number of instructions must be executed for each I/O transfer

- When large volumes of data are to be moved a more efficient technique is *direct memory access* (DMA)

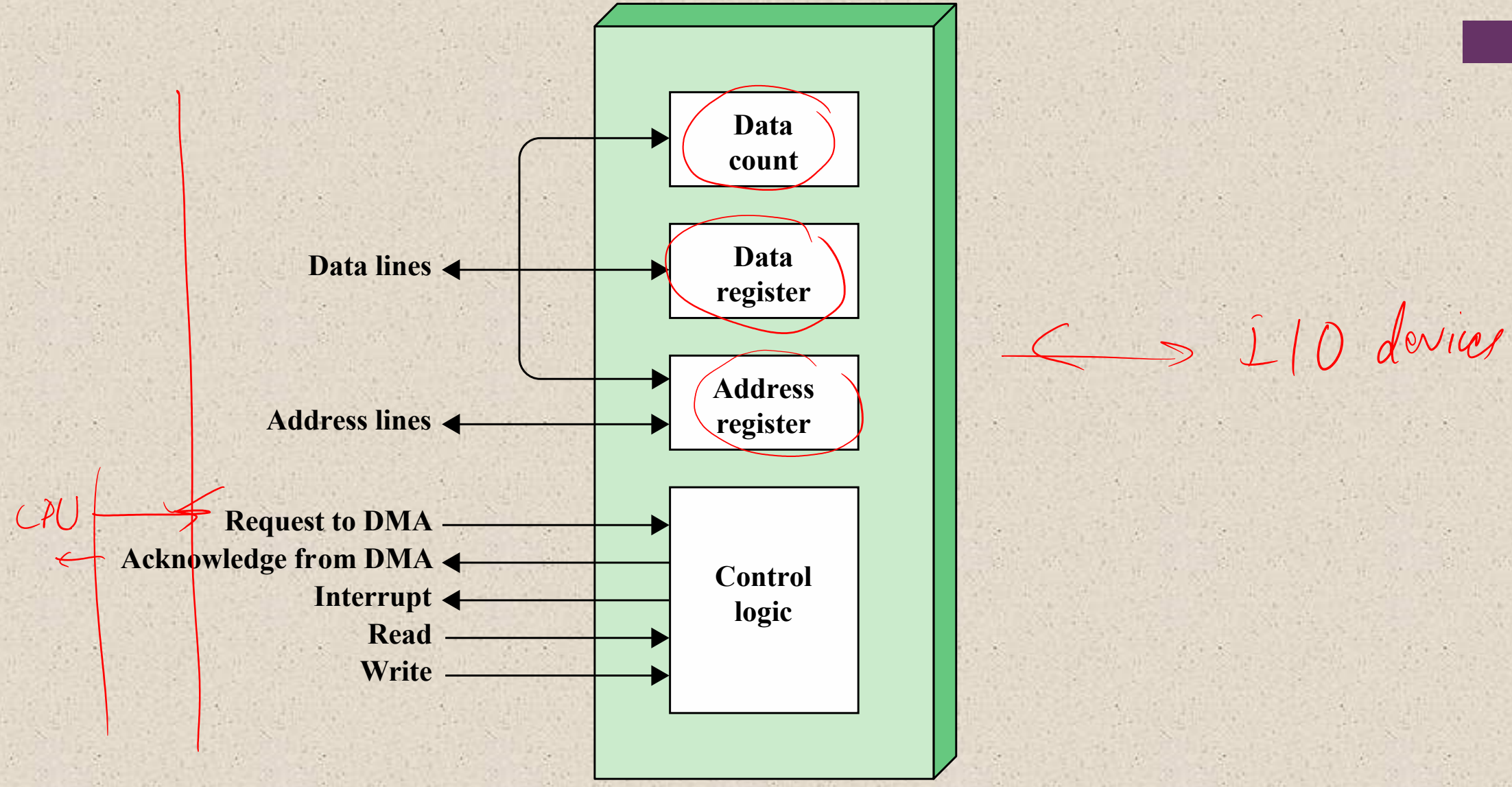
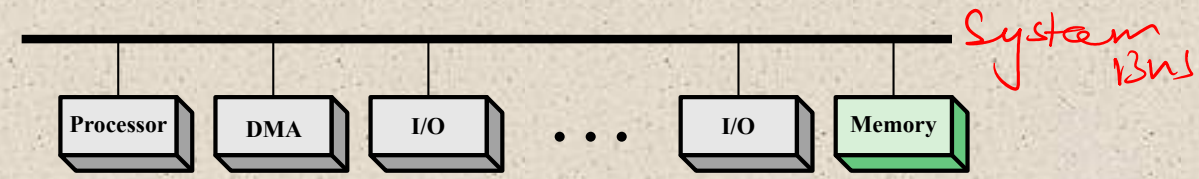
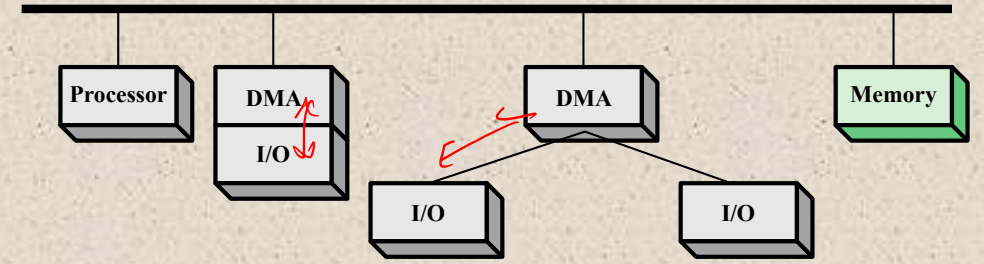


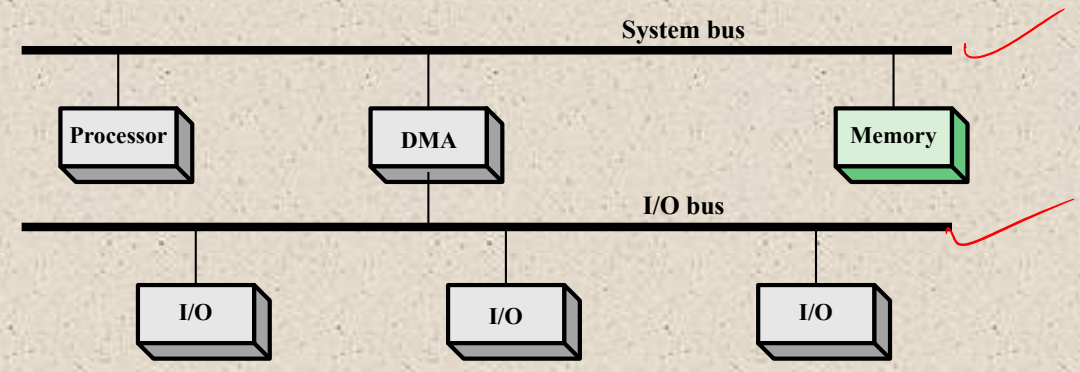
Figure 7.12 Typical DMA Block Diagram



(a) Single-bus, detached DMA

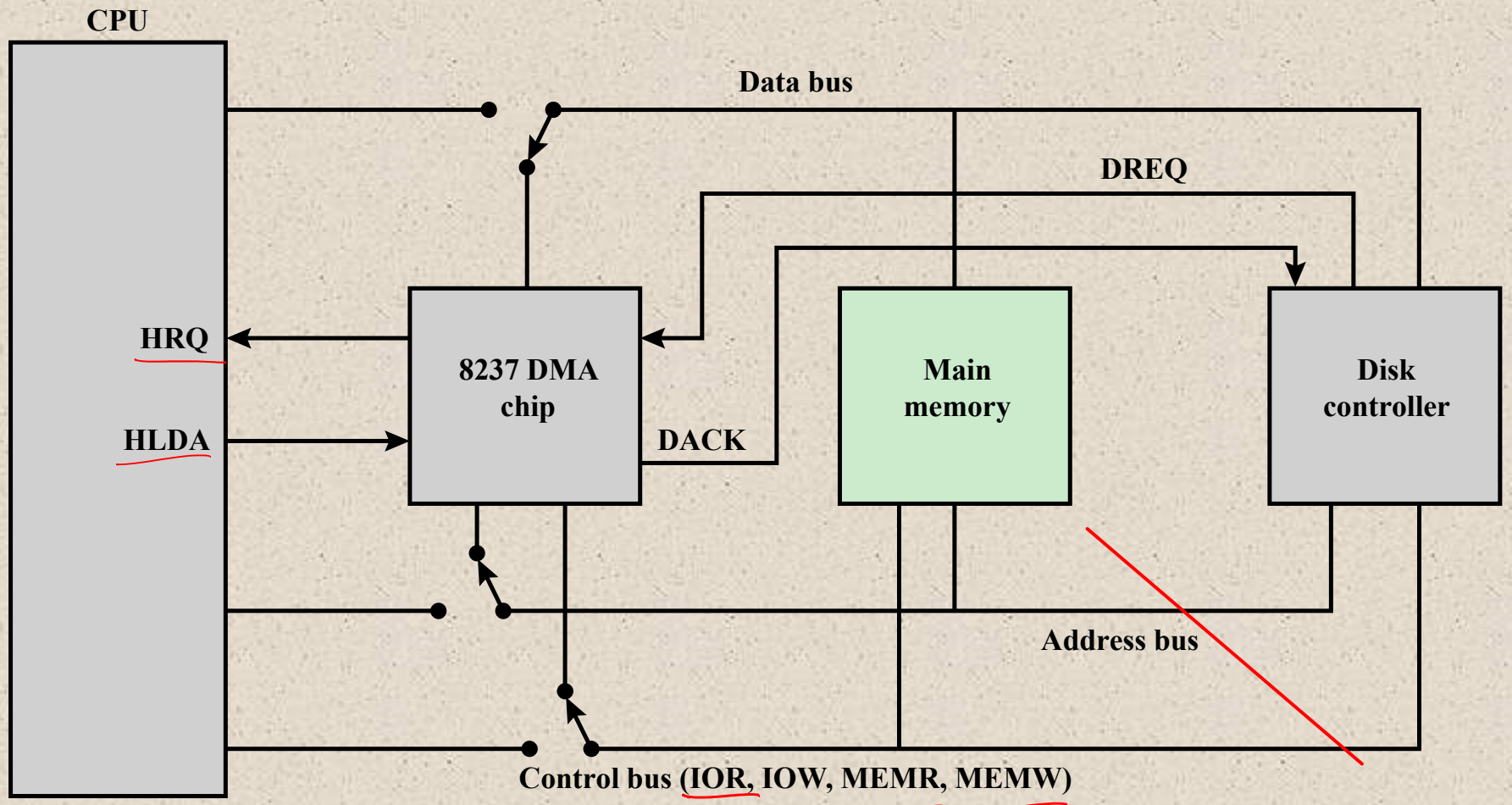


(b) Single-bus, Integrated DMA-I/O



(c) I/O bus

Figure 7.14 Alternative DMA Configurations



DACK = DMA acknowledge
DREQ = DMA request
HLDA = HOLD acknowledge
HRQ = HOLD request

Figure 7.15 8237 DMA Usage of System Bus

Modes of Data Transfer in DMA

There are 3 modes of data transfer in DMA that are described below.

1. **Burst Mode:** In Burst Mode, buses are handed over to the CPU by the DMA if the whole data is completely transferred, not before that.

2. **Cycle Stealing Mode:** In Cycle Stealing Mode, buses are handed over to the CPU by the DMA after the transfer of each byte. Continuous request for bus control is generated by this Data Transfer Mode. It works more easily for higher-priority tasks.

3. **Transparent Mode:** Transparent Mode in DMA does not require any bus in the transfer of the data as it works when the CPU is executing the transaction.

Gate 2020

Q.No. 3 Consider the following statements.

I. Daisy chaining is used to assign priorities in attending interrupts.

II. When a device raises a vectored interrupt, the CPU does polling to identify the source of interrupt.

III. In polling, the CPU periodically checks the status bits to know if any device needs its attention.

IV. During DMA, both the CPU and DMA controller can be bus masters at the same time.

Which of the above statements is/are TRUE?

(A) I and II only

(B) I and IV only

(C) I and III only

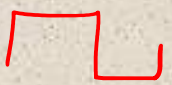
(D) III only

Gate 2024 Set 2

Consider a computer with a 4MHz processor. Its DMA controller can transfer 8 bytes in 1 cycle from a device to main memory through cycle stealing at regular intervals. Which one of the following is the data transfer rate *(in bits per second)* of the DMA controller if 1% of the processor cycles are used for DMA?

1. 2,56,000
2. 3.200
3. 3.25,60,000
4. 32,000

clock rate (Cycle Time) = $\frac{1}{4MHz}$



Data transfer time = 1CC = .25 μsec.

In .25 μsec (CPU + DMA) ⇒ 8 bytes ~~data~~ transferred.

~~In 1 sec how much~~ ⇒ Time taken in cycle stealing mode = $\frac{.25 \times 100}{1}$ = 25μs.

25 μsec = 8 byte data.

transfer rate 1 sec = $\frac{8 \times 8}{25\,\mu s} = \frac{64 \times 10^6}{25} = 2.56 \times 10^6$ bits/sec.

Which one of the following statements is FALSE?

A. In the cycle stealing mode of DMA, one word of data is transferred between an I/O device and main memory in a stolen cycle

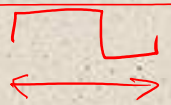
B. For bulk data transfer, the burst mode of DMA has a higher throughput than the cycle stealing mode

C. Programmed I/O mechanism has a better CPU utilization than the interrupt driven I/O mechanism

D. The CPU can start executing an interrupt service routine faster with vectored interrupts than with non-vectored interrupts

GATE Practice

A hard disk with a transfer rate of 1 Mbytes/ second is constantly transferring data to memory using DMA. The processor runs at 500 MHz, and takes 500 and 1000 clock cycles to initiate and complete DMA transfer respectively. If the size of the transfer is 1 Kbytes, what is the percentage of processor time consumed for the transfer operation?_____(Rounded off to three decimal)



CC time = $\frac{1}{500MHz}$

DMA (Ini + Compl^n) time = $\frac{500+1000}{500\ M} = \frac{3}{M} = 3\mu s$

transfer rate = 1 MBPS = 10^6 B/sec

transfer rate for 1 Byte = $\frac{1}{10^6}$, so Time take 1 KB data = $\frac{1024}{10^6} = 1024\ \mu s$

~~data for 1 sec =~~

~~1024 × 1024~~

$\frac{3\ \mu s}{3+1024} \times 100\ \%$

$\simeq 0.3\ \%$

GATE 2022

Which one of the following facilitates transfer of bulk data from hard disk to main memory with the highest throughput?

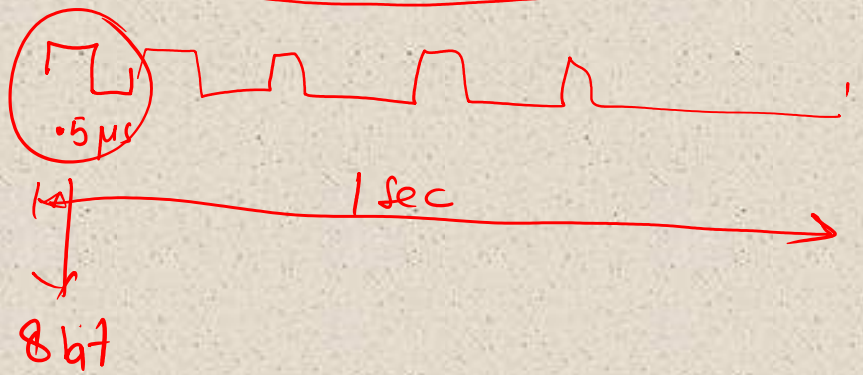
1. DMA based I/O transfer (Burst mode)
2. Interrupt driven I/O transfer
3. Polling based I/O transfer
4. Programmed I/O transfer

Thimothi williams MCQ in CSE

GATE 2021 set-2

Consider a computer system with DMA support. The DMA module is transferring one 8-bit character in one CPU cycle from a device to memory through *cycle stealing* at regular intervals. Consider a 2 MHz processor. If 0.5% processor cycles are used for DMA, the data transfer rate of the device is _____ bits per second.

Time of CC = $\frac{1}{2\,MHz}$ sec = .5 μsec



Total CC in 1 sec = $\frac{1}{.5\mu} = 2 \times 10^6$

No CC utilized in 0.5% ⇒ $\frac{0.5}{100} \times 2 \times 10^6 = 10000$ sec

8 bits to be transfer in 1 CPU cycle

transfer rate of device = 8 × 10000 bits/sec.