

Unit 5 : Input / Output

Input / Output : Peripheral devices, I/O interface, I/O ports, Interrupts: input hardware, types of interrupts and exceptions.

Modes of Data Transfer : Programmed I/O, interrupt initiated I/O and Direct Memory Access, I/O channels and processes.

Serial communication: synchronous and Asynchronous communication, standard communication interfaces.

Peripheral Device : All devices which are connected to the CPU that called peripheral device.

Peripheral Input Device

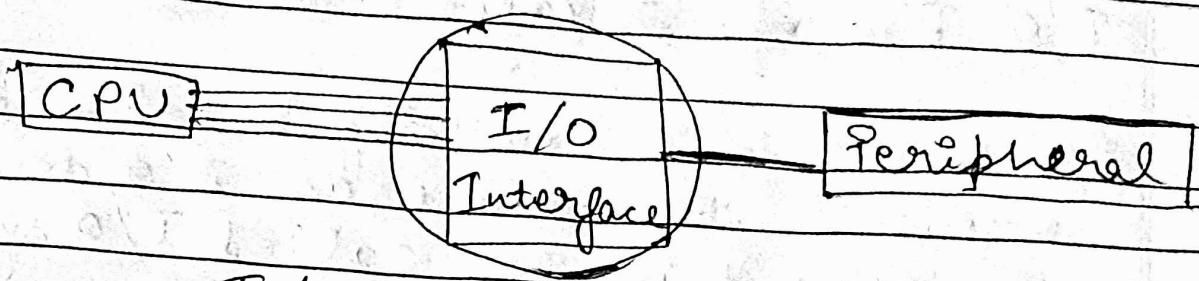
- (a) Keyboard
- (b) Mouse
- (c) Touch Screen
- (d) Trackball
- (e) Trackpads
- (f) Joystick
- (g) Light pen

Output device

- (a) Monitors
- (b) LCD
- (c) Printers
- (d) Laser Printers
- CRT, LCD
- Scanner

I/O Interface

- (i) I/O interface enables transfer of data b/w internal storage and external I/O device.



I/O Interface

In order to interface peripherals with the CPU, special communication links are required. Due to this communication link, the difference b/w the CPU and peripherals such as data transfer speed, mode of operation are overcome.

Peripherals

CPU

1. These are electro-mechanical and electromagnetic device. It is an electronic device.
2. Data transfer rate is slower than that of the CPU. Data transfer rate is faster than that of peripherals.

3. Data is form of codes.

Data is in word format.

I/O Bus and Interface Modules

I/O interface is nothing but the hardware required to connect an I/O device to the bus. It is also called I/O system. The major requirements of an I/O interface are :

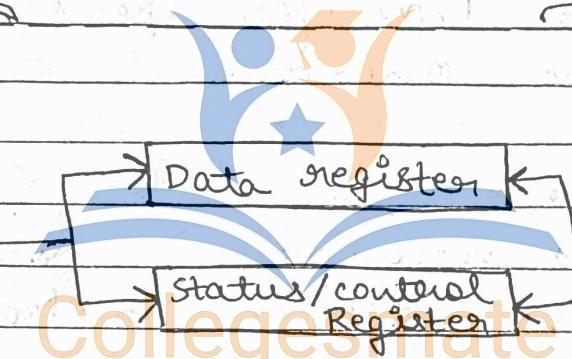
- (a) Control and Timing
- (b) Processor Communication
- (c) Device Communication
- (d) Data Buffering
- (e) Error detector

Interface to
System
Bus

Interface to external
device

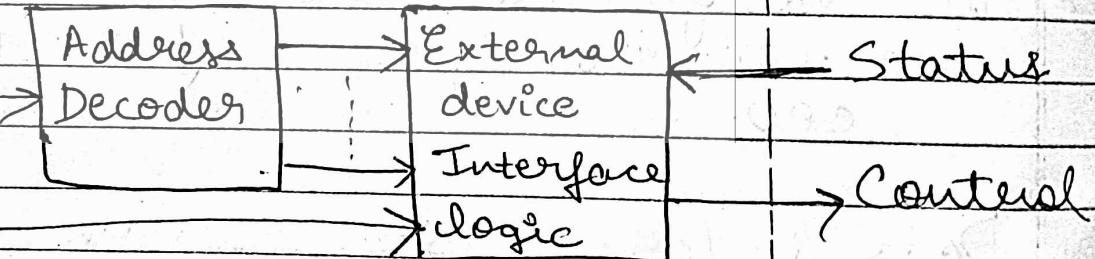
Data
Lines

→ Data



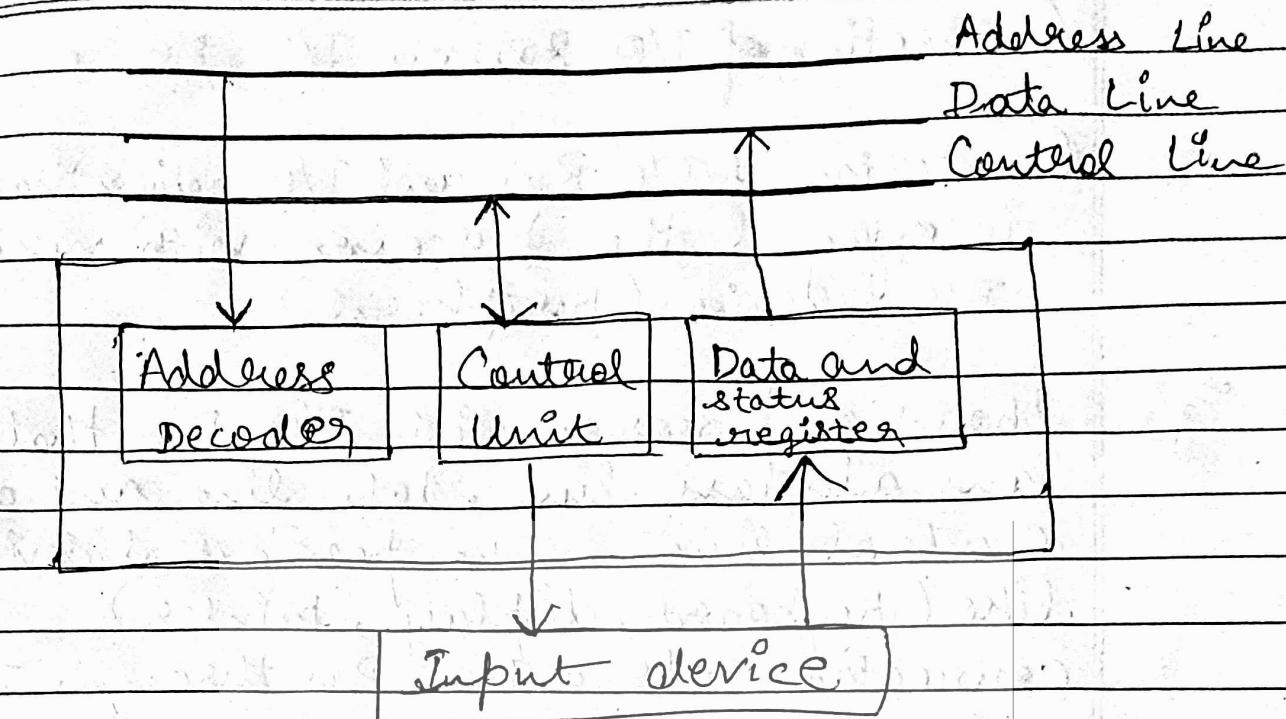
EDUCATIONAL SUPPORT SERVICES

Address
line
Control
line

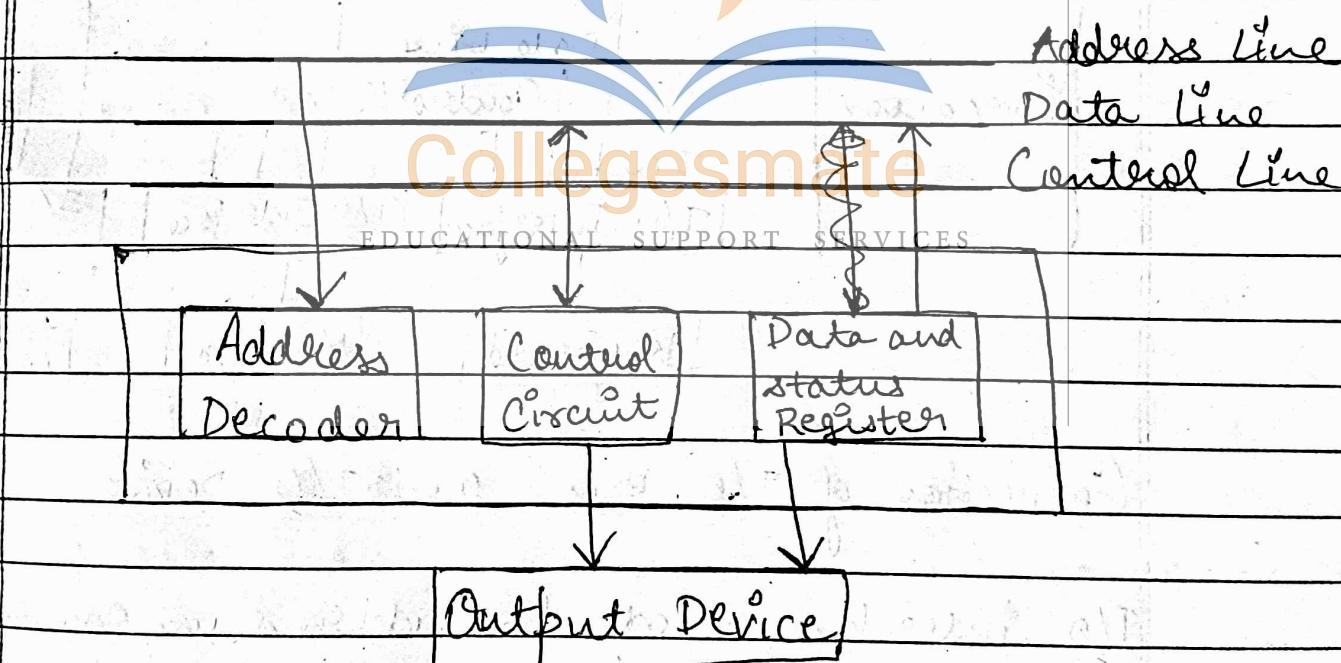


I/O Interface

Fig: Block diagram of I/O



(a) I/O Interface for I/P device

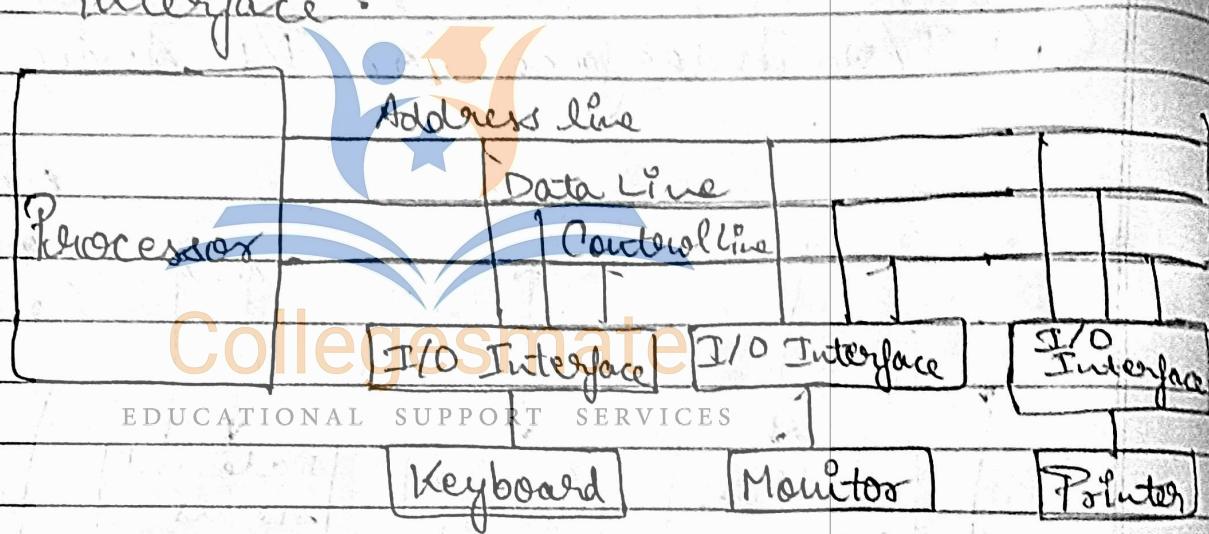


(b) I/O Interface for O/P device

Connection of I/O Bus to I/O Device

Connection of I/O Bus and I/O Device that interface the processor with various I/O devices (peripherals).

Eg:- Shows a processor with I/O Bus that has an address line, data line and a control line. The peripheral device like (Keyboard, display, printer) connected to a I/O Bus through I/O interface.



Connection of I/O Bus and I/O Device

I/O Interface decodes the address and control signals received from I/O Bus and forwards the decoded signals to the peripheral controllers. It is also responsible for synchronization of data flow and data transfer b/w the peripheral and processor.

The function code is known as I/O command and cause exchange of instruction b/w the I/O Interface and its respective peripheral.

Types of commands

- ↳ Control Command

- ↳ Status Control

- ↳ Data O/P command

- ↳ Data Input command

* Control Command : Activates the peripheral and tells the peripheral what is to be done.

* Status Command : Used to test various status condition in the interface and the peripheral.

* Data output Command : It enables the interface to respond i.e. data transfer takes place from the I/O but to one of the interface Registers.

* Data Input Command : It is exact opposite to the data O/P command. Here interface ~~recieve~~ receives data from the peripheral and is placed in the in the interface buffer Register.

I/O Vs Memory Bus and I/O interfacing Techniques

- ① Memory mapped I/O
- ② Isolated mapped I/O
- ③ Memory bus

Isolated Vs Memory Mapped

Memory Mapped I/O

Isolated Mapped I/O

- | | |
|---|---|
| 1. Memory and I/O shared the entire address range of processor. | Processor provide separate address lines for memory and I/O device. |
| 2. Usually processor provides more address line for accessing memory. | Usually processor provides less address lines for access memory. |
| 3. Memory control signals are used to control read/write I/O opn. | I/O control signals are used to control read and write I/O opn. |

I/O Port

- 66 The circuit which provides a data path with its associated controls to transfer data b/w the interface and the I/O device is

Called I/O port."

Types of I/O Ports

The port can be classified as serial port or parallel port.

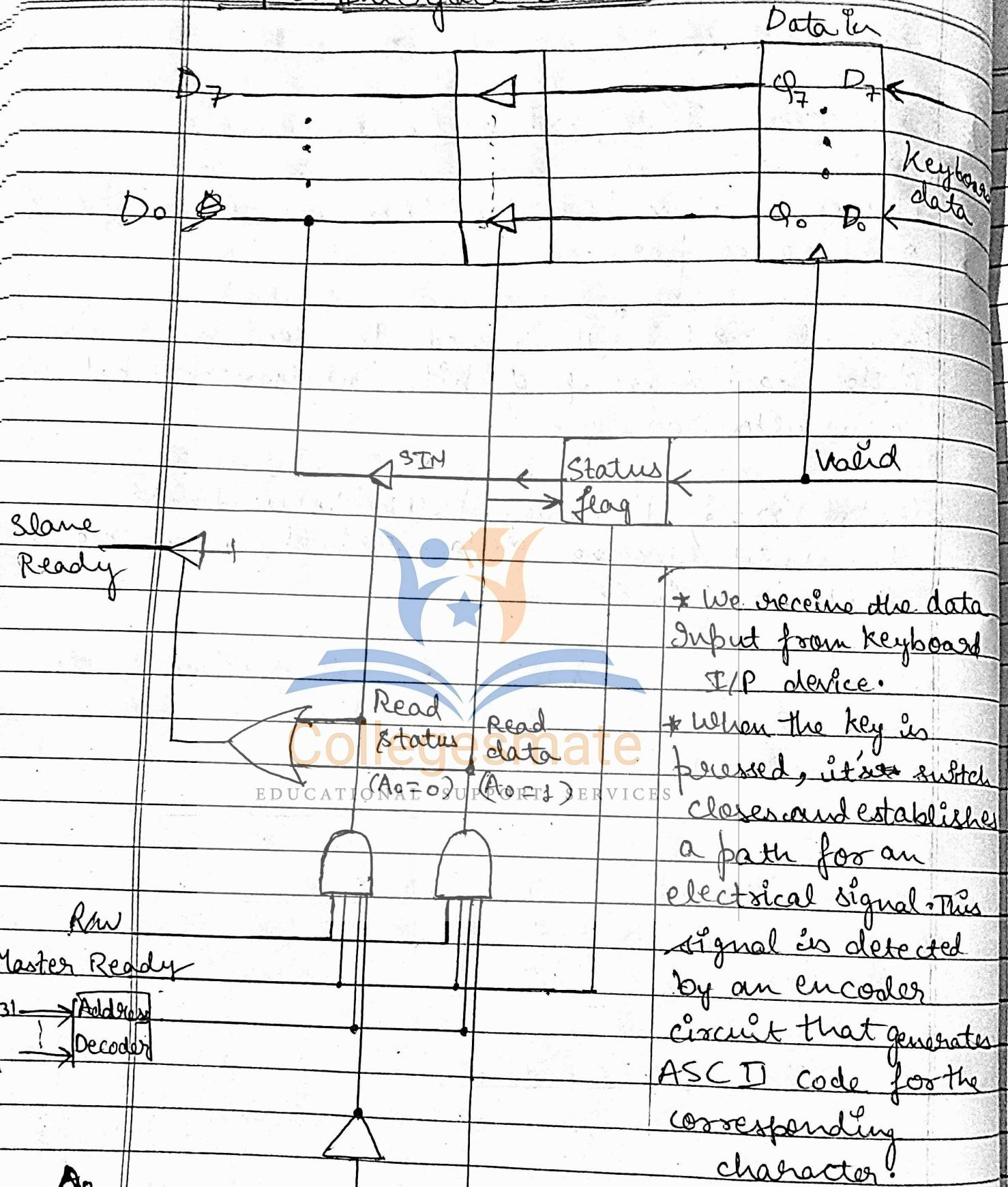
Parallel Port: "It is used to send or receive data having group of bits (8 bits or 16 bits) simultaneously.

Serial Port: A serial port is used to transmit/receive data serially i.e. one bit at a time.

 Collegesmate

A not-for-profit educational support services.

Input Interface Circuit

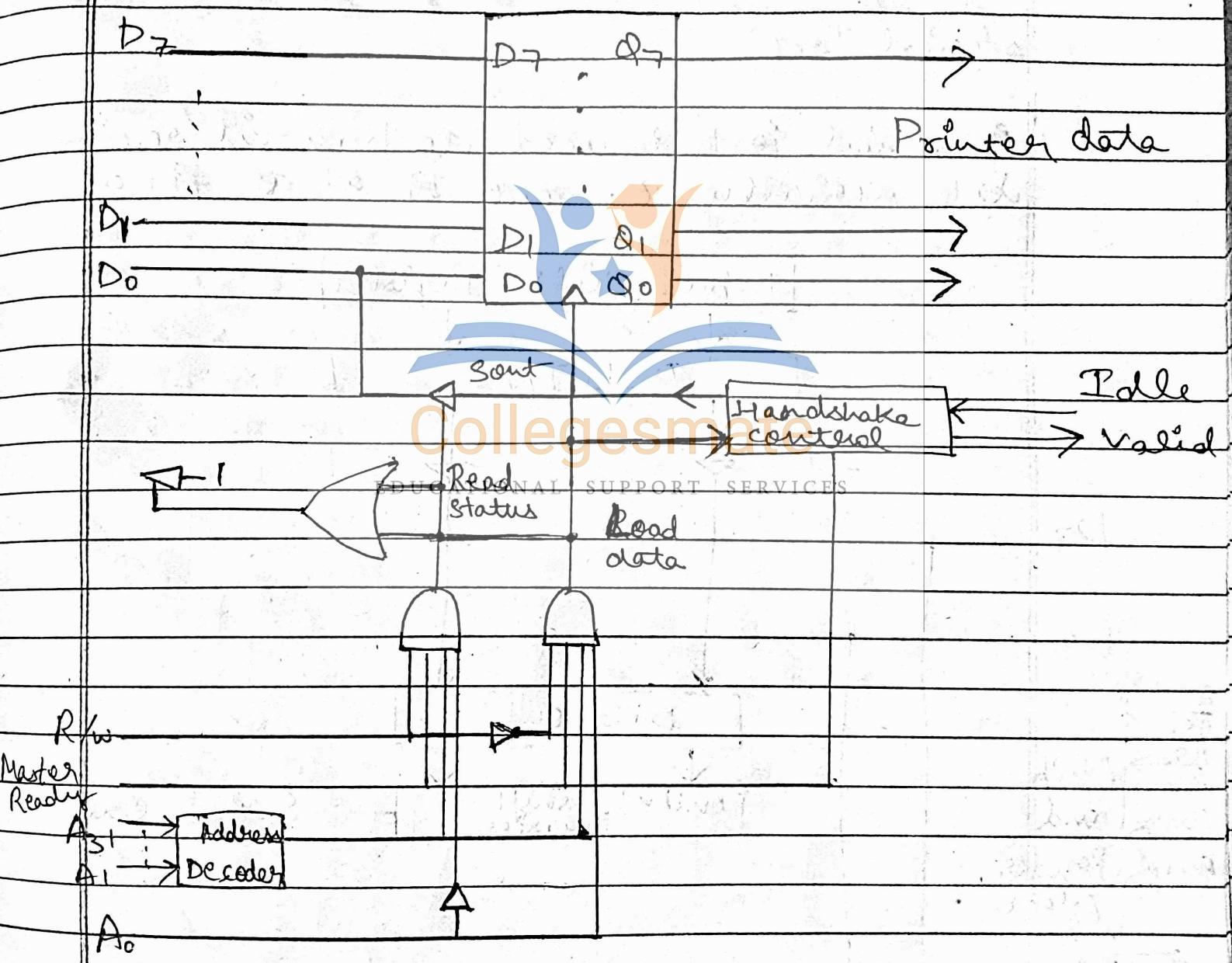


- * We receive the data input from keyboard I/P device.
- * When the key is pressed, it's switch closes and establishes a path for an electrical signal. This signal is detected by an encoder circuit that generates ASCII code for the corresponding character!

Fig : Input Interface Circuit

The I/P port consists of data register. Data IN and status flag, SIN, when a key is pressed, the valid signal activates and causes the ASCII code to be loaded into Data IN and SIN to be set to 1. The status flag SIN is cleared to 0.

Output Interface Circuit



Output Interface Circuit

The O/P port contains a data register, Data out and a status flag, sent. The dont flag is set to 1 when the pointer is ready to accept another character and it is cleared to 0 when a new character is loaded into Data out by the processor.

Serial Port

A serial Port is used to transmit/receive data serially i.e. one bit at a time.

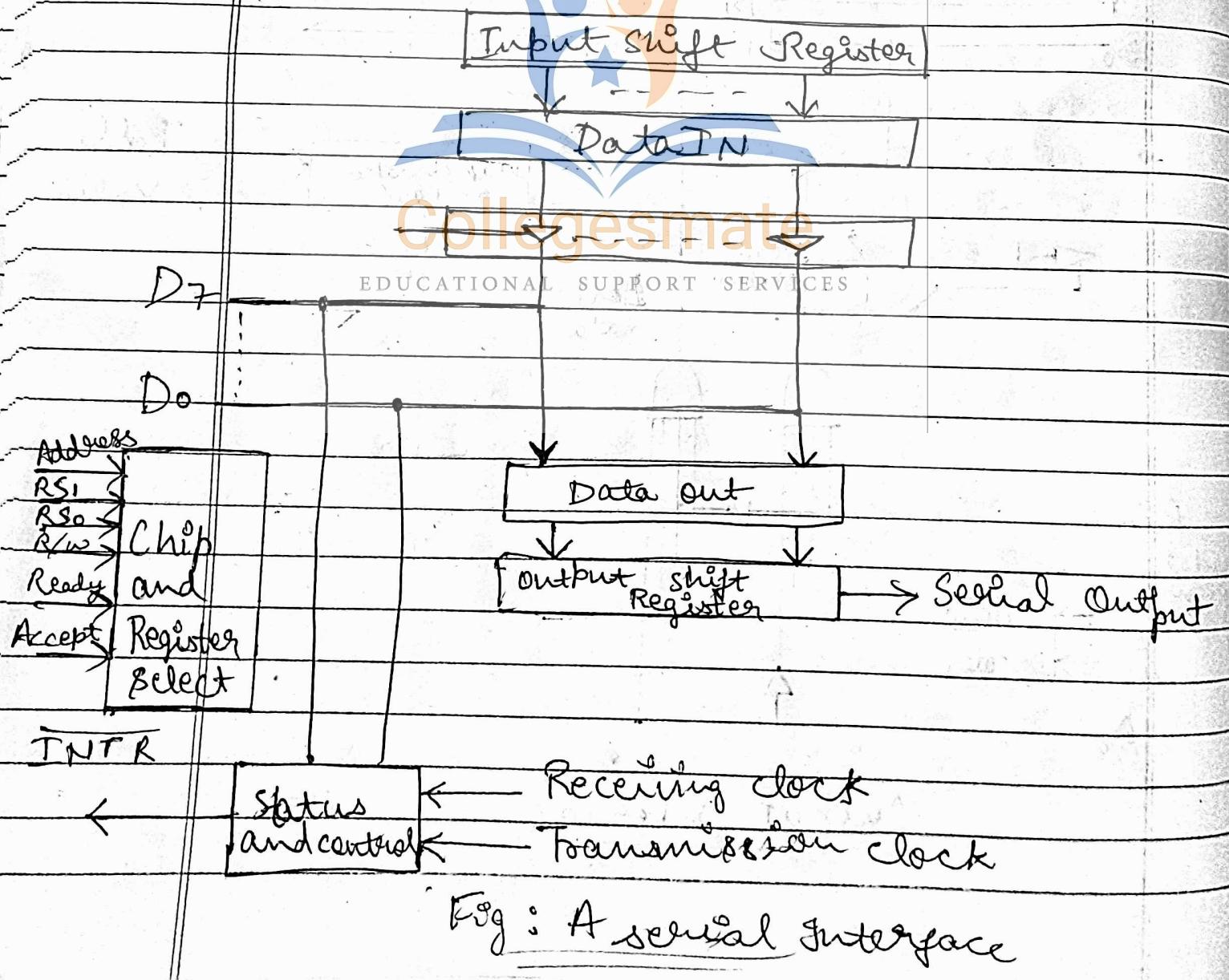


Fig : A serial Interface

The Input shift Register accepts serial data bit by bit and converts it into the parallel data. The converted parallel data is loaded in the data register and it is then read by the processor using data Bus.

I/O Processor (IOP)

- ↳ IOP can fetch and execute its own instruction.
- ↳ It can have complete control over I/O op
- ↳ It can perform arithmetic and logic op, branches, searching and translation.
- ↳ IOP can transfer data from an 8-bit source to 16 bit destination and viceversa.
- ↳ Communication b/w IOP and CPU is through Memory based control Blocks.

Block diagram of IOP

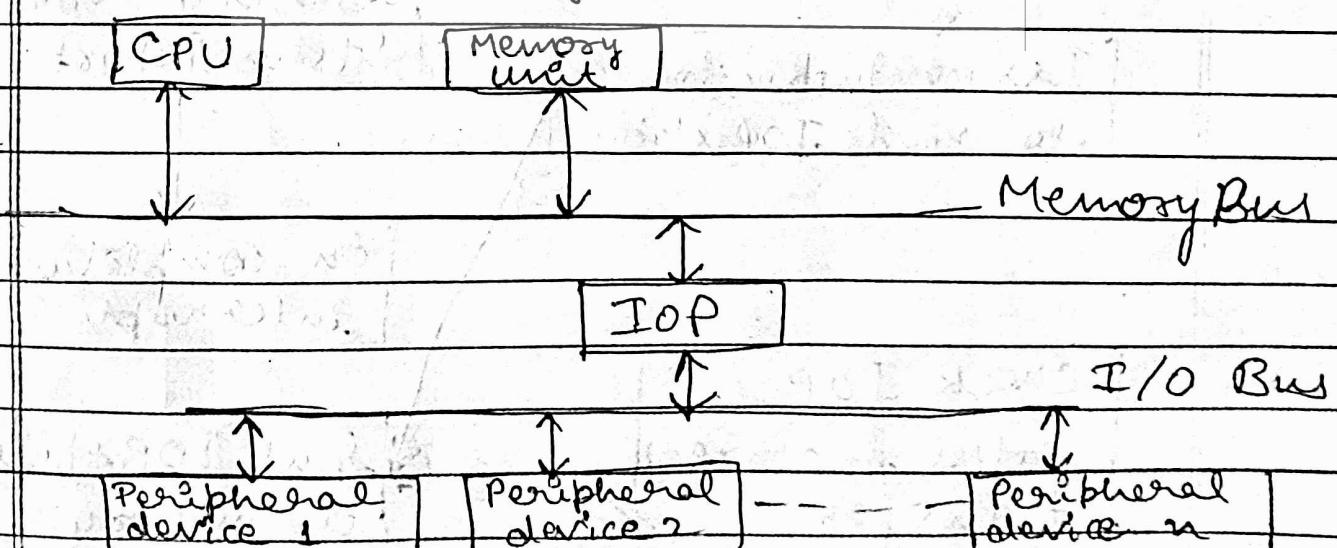


Fig: Computer with IOP

The CPU and IOP work independently used to communicate with each other using centrally located memory and DMA. IOP does the data transfer B/w various peripheral devices and the memory unit.

Communication B/w CPU and IOP

CPU operation

Issues Instruction
test IOP Path

If status is OK,
sends Instruction to
start I/O transfer

Continues with
another Program

Issues Instruction
to read IOP status

Check IOP

status for correct
transfer

IOP operation

Transfer status
word into memory

Accesses memory
for I/O Program

Perform I/O transfer
using DMA prepare
Status Report

On completion
interrupts CPU

Send IOP status
to memory

Fig: CPU and IOP Communication

I/O Channels

- ↳ An I/O channel has a special purpose processor. This processor has an ability to execute I/O instruction and it can have control over I/O operations.
- ↳ The I/O instruction are stored in main memory when I/O transfer is required the CPU initiates and I/O transfer by instructing the I/O Channel to execute and I/O program stored in main memory.

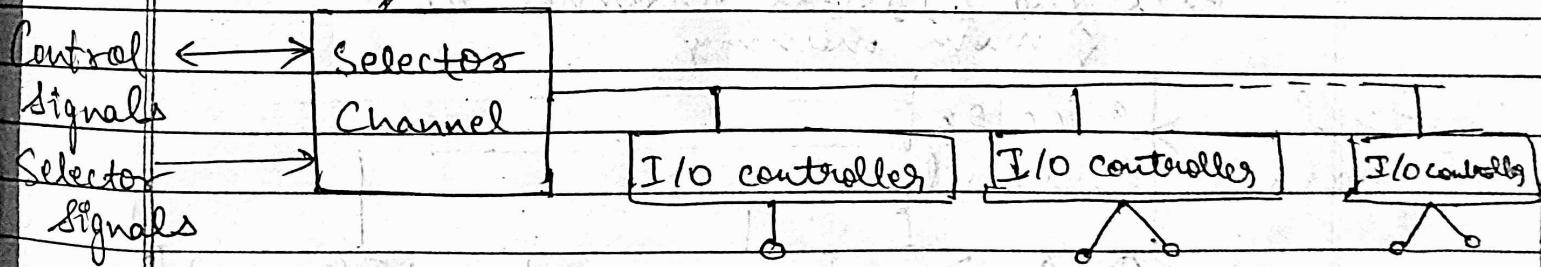
Types of I/O Channels

Two main types of I/O channels:

- ↳ Selector Channel
- ↳ Multiplexer Channel

Selector Channel

→ Data and Address Channels to main memory

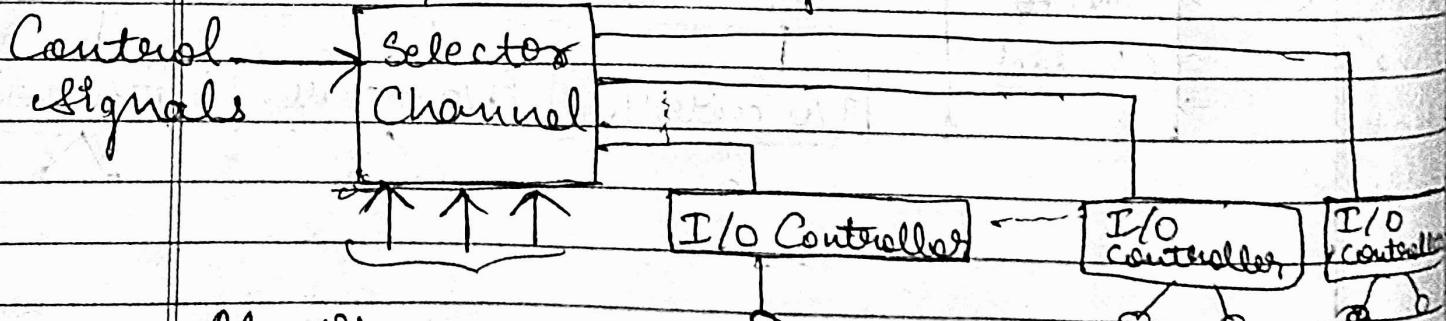


- ⇒ A selector channel controls multiple high-speed devices. However at a time it can control the data transfer of only one device. It selects one device at a time and does the data transfer.
- ⇒ Each device, or a small set of device is handled by a controller, or I/O module as shown in previous figure.

Multiplexer channel

- ↳ A multiplexer channel can handle I/O with multiple device service at the same time.
- ↳ The multiplexer channel does the time multiplexing and communicates with each I/O controller in a allotted time slot.
- ↳ In multiplexer channel different device can have different speeds.

Data and address Channel to
↓ main memory



Multiplexer
select line

Fig: Multiplexer channel

Modes of Transfer

① Programmed I/O

- ⇒ I/O operation will mean a data transfer b/w an I/O device and memory or b/w an I/O device and the processor.
- ⇒ If in any computer system I/O op's are completely controlled by the Processor, that system is said to be using "Programmed I/O".
- ⇒ When such a technique is used, processor executes programs that initiates direct and terminate the I/O operation, including sensing device status for sending a read or write command and transferring the data.
- ⇒ It is the responsibility of the processor to periodically check the status of the I/O system until it finds that the operation is complete.

Ex: This is a simple program which services I/O ports A, B and C. The routine checks the status of I/O ports in proper sequence. If the request bit is set the corresponding device routine is called to complete the I/O request.

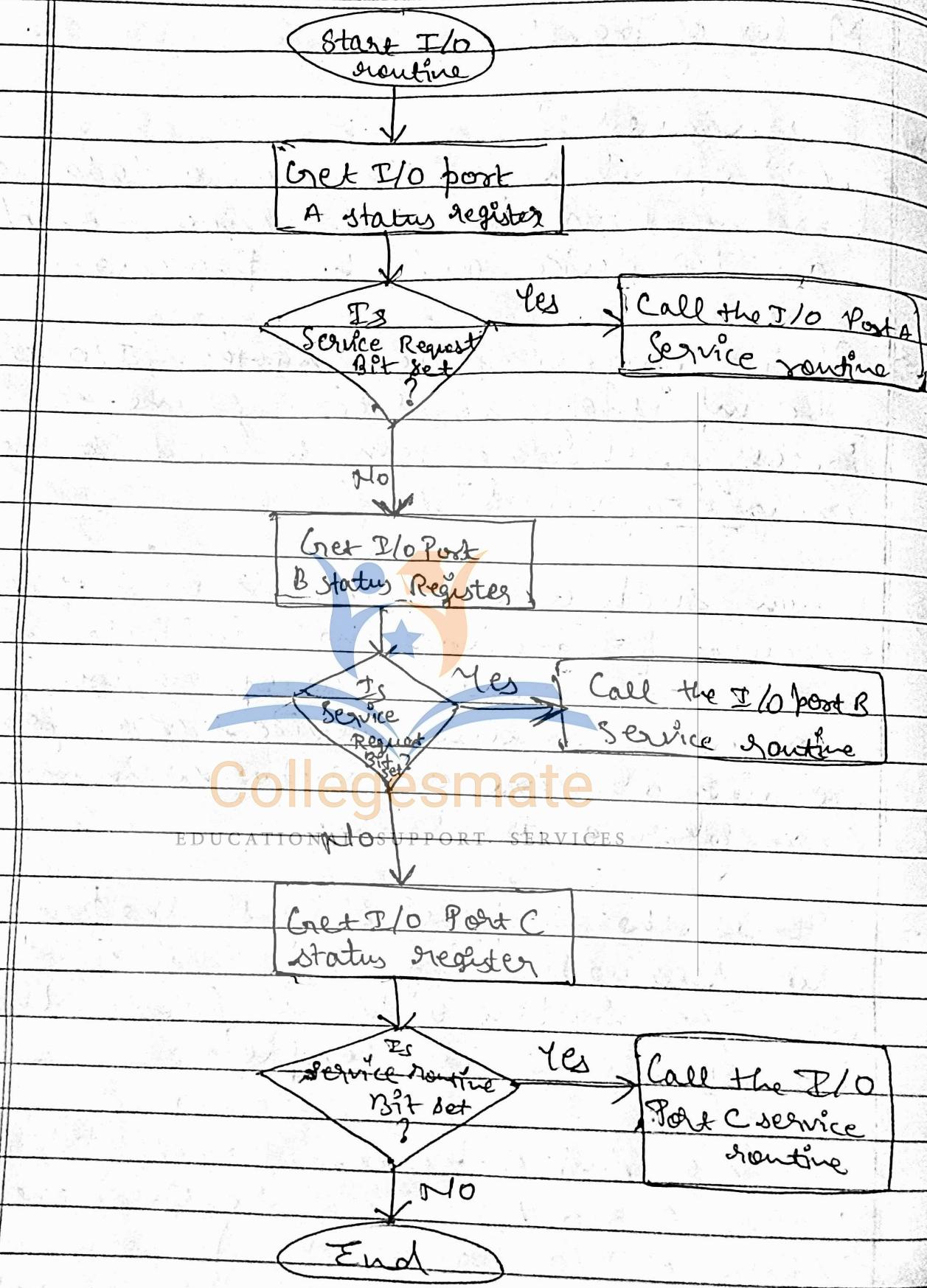
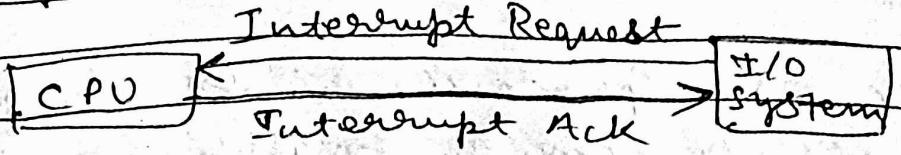


Fig : Flowchart for I/O Service routine

⑥ Interrupt - Initiated I/O



This method provides an external asynchronous input that would inform the processor that it should complete whatever instruction that is currently being executed, and fetch a new routine (Interrupt Service Routine) that will service the requesting device.

After this receiving is completed, the processor would resume exactly where it left off. The event that causes the interruption is called interrupt and the special routine executed to service the interrupt (ISR) is called the Interrupt Service Routine (ISR).

Hardware Interrupt

An interrupt caused by an External signal is referred as a Hardware interrupt.

Software Interrupt

Conditional interrupts or interrupts caused by special instruction are called software interrupt.

Maskable Interrupt

In the processor those interrupts which can be masked under software control are called maskable interrupts.

Non Maskable Interrupt

The interrupts which can not be masked under software control are called non maskable interrupts.

Vector Interrupt

If the internal control circuit of the processor produces a call to pre determined memory location which is the starting address of interrupt service routine, then that address is called vector address and such interrupts are called vector interrupts.

There are 2 ways to support vector interrupts:

1. Fixed Vector address
2. Programmable vector address

The processor which supports programmable vector address approach maintains the table in memory called the

Maskable Interrupt

In the processor, those interrupts which can be masked under software control are called maskable interrupt.

Non Maskable Interrupt

The interrupts which can not be masked under software control are called non maskable interrupt.

Vector Interrupt

If the internal control circuit of the processor produces a call to pre determined memory location which is the starting address of interrupt service routine, then that address is called vector address and such interrupts are called vector interrupts.

There are 2 ways to support vector interrupts:

1. Fixed Vector Address
2. Programmable vector address

The processor which supports programmable vector address approach maintains the table in memory called the

interrupt vector table.

The programmers are allowed to change the vector address in the interrupt vector table. Thus this approach is known as programmable vector address.

Nested Interrupts

A system of interrupts that allows an interrupt service routine to be interrupted is known as Nested Interrupts.

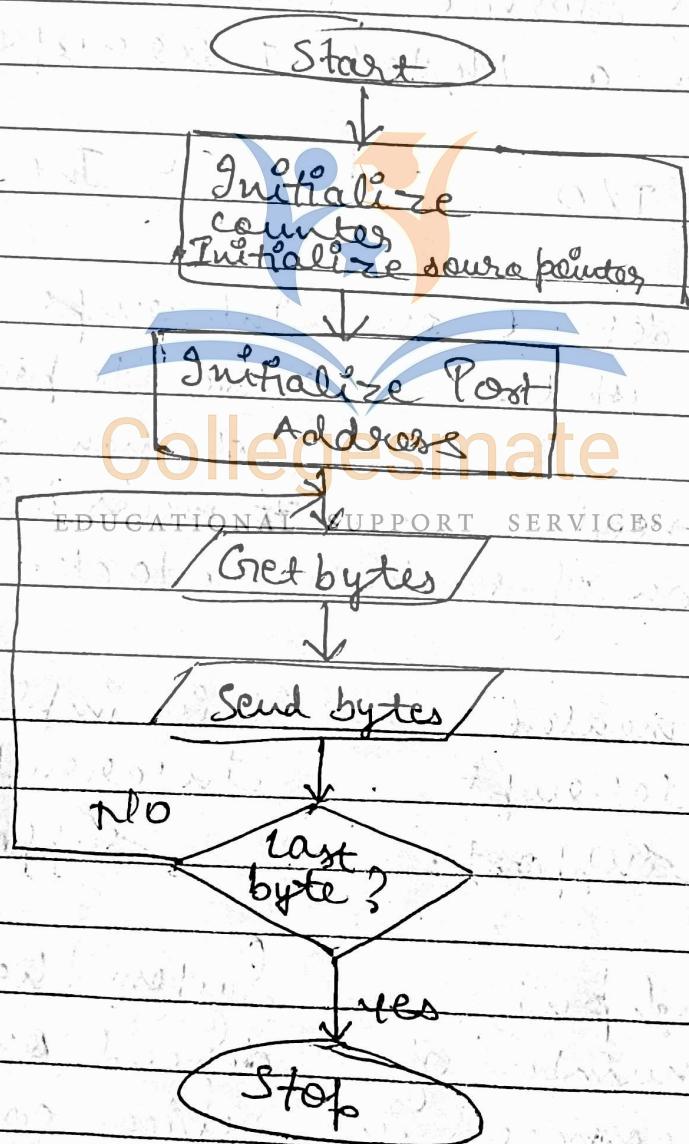
Programmed I/O

Interrupt driven I/O

- It does not depend on interrupt status.
 - It does not need initialization of stack.
 - It is implemented without interrupt hardware support.
 - System throughput decreases as number of I/O devices connected in the system decreases.
- Collegesmate** EDUCATIONAL SUPPORT SERVICES
- Interrupt must be enabled to process interrupt driven I/O.
 - It needs initialization of stack.
 - It is implemented using interrupt H/w support.
 - System throughput does not depend on number of I/O devices connected in the system.

DMA (Direct Memory Access) - (DMA)

- In SW control data transfer, processor executes a series of instruction to carry out data transfer for each instruction execution fetch, decode and execute phases are required. The flow chart to transfer data from memory to I/O device.



DMA Block diagram

DMA stands for ~~the~~ Direct Memory Access.
 It is a H/W controlled data transfer.
 DMA controller is used to carry-out data transfer. During data transfer data is not treated through processor.

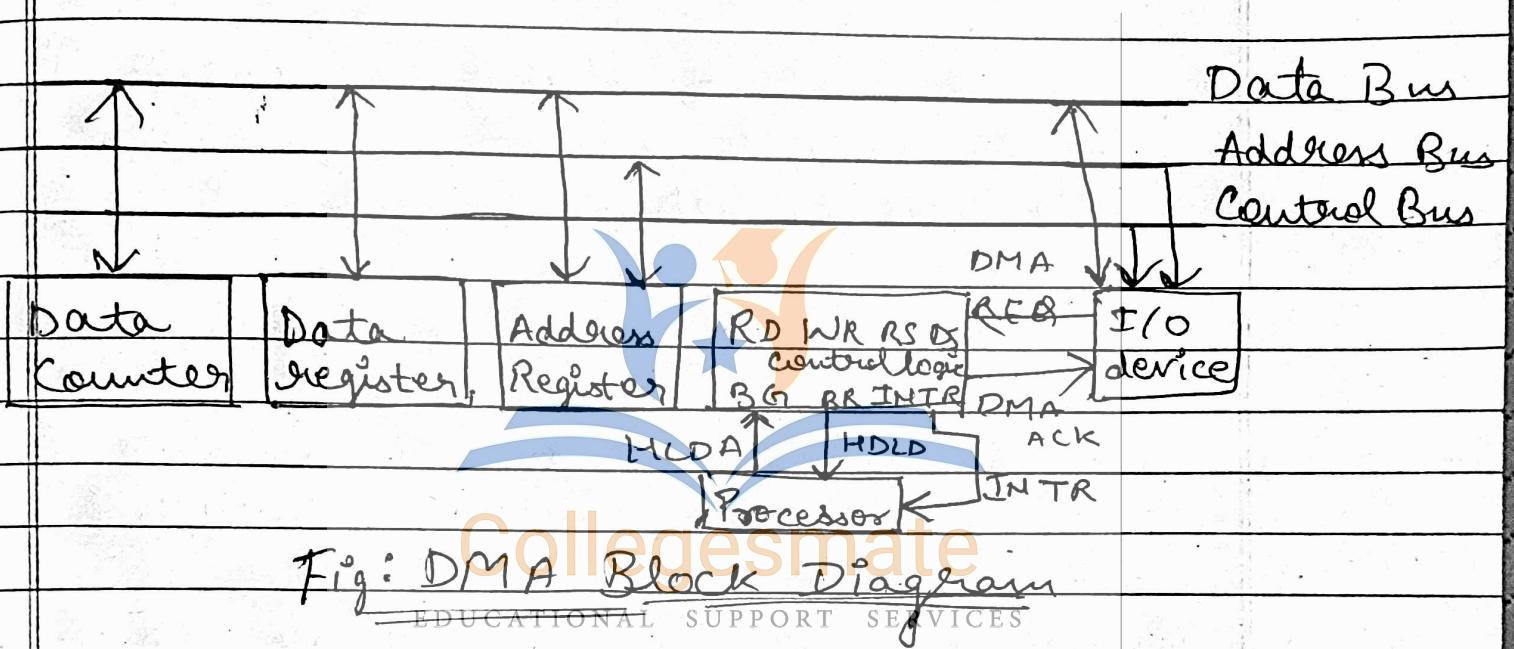


Fig: DMA Block Diagram

For performing the DMA operation, the Basic Blocks required in a DMA channel/controller are shown in fig.

- ④ DMA controller communicates with the CPU via the data Bus and control lines.

⑤