UNIT-05

I/O ORGANIZATION

DATE: / /
PAGE:

* Peripheral devices :—
A peripheral device is defined as the device which provides input/output functions for a computer. A peripheral device is a device that is connected to a computer system but is not part of the core computer system architecture.

* Classification of Peripheral devices :—
It is generally classified into 3 basic categories:

1. Input Devices:
Example:
Keyboard, mouse, scanner, microphone etc.

2. Output Devices:
Example :—
Monitor, headphone, printer etc.

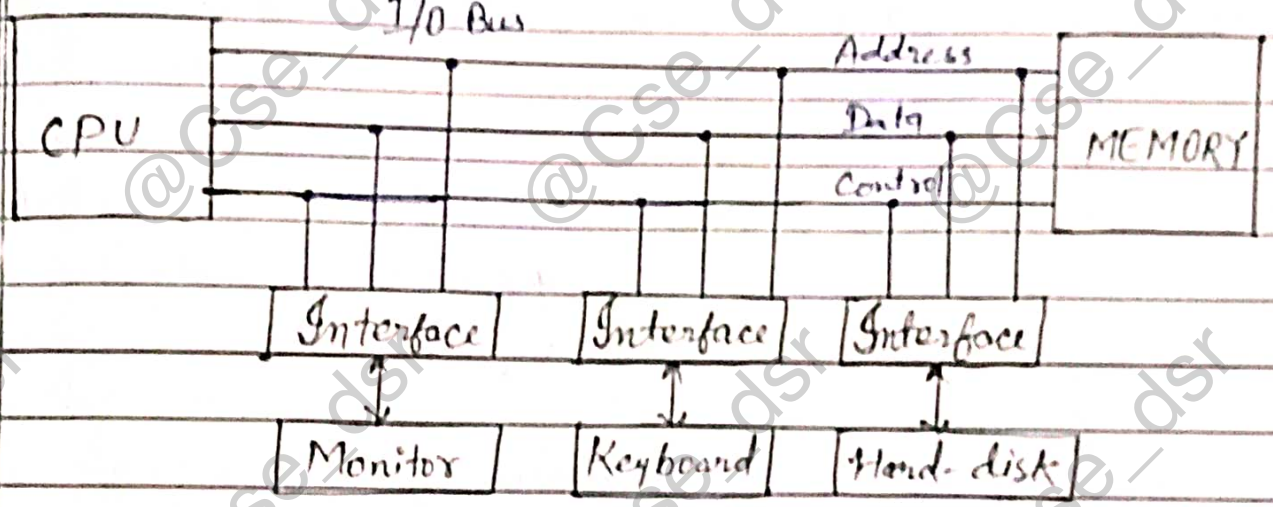
3. Storage Devices:
Example :—
External Hard disk, magnetic tape, pen drive etc.

* Advantage of Peripheral Devices:
↳ It is ~~taking for~~ helpful for taking input very easily.

↳ It is also provided a specific output.
↳ It has a storage device for storing info^{n} or data.
↳ It also improves the efficiency of the system.

* Input - Output Interface:

Input - Output Interface is used as an method which helps in transferring of information between the internal storage devices and the external peripheral device.



* Functions of Input - Output Interface:

↳ It is used to synchornize the speed of CPU with respect to input-output devices.
↳ It selects the input-output device which is appropriate for the interpretation.
↳ It is capable of providing signals like control

DATE: __/__/__
PAGE: _____

and timing signals.

↳ There are various error detectors inside.

↳ It converts serial data into parallel data and vice-versa.

↳ It also convert digital data into analog signal and vice-versa.

* Interrupt :—

↳ An interrupt is a signal that requests the processor to suspend its current execution and service the occured interrupt.

↳ To service the interrupt the processor executes the corresponding interrupt service routine (ISR).

↳ After the execution of ISR, the processor resumes the execution of the suspended program.

↳ Interrupt is of two types :—

① Hardware interrupt
   ↳ Maskable
   ↳ Non-Maskable

② Software interrupt
   ↳ Normal interrupt
   ↳ Exception

DATE : __/__/__
PAGE : ____

① Hardware interrupt :—
If a processor receives the interrupt request from an external . I/O device , it is termed as h/w interrupt.

ⓐ Maskable interrupt :— The h/w interrupt that can be ignored or delayed for sometime . If the processor is executing a program with higher priority.

ⓑ Non- Maskable interrupt :— The h/w interrupt that can neither be ignored nor delayed and must immediately be serviced by the processor are termed as non-maskable interrupt.

② Software interrupt :—
The interrupt that occur when condition is met or a system call occurs. This interrupt signal is generated by the s/w programs .

ⓐ Normal interrupt :— The interrupt that is caused by the s/w instruction or system call is called normal software interrupt.

DATE: / /
PAGE

(b) Exception :- It is an unplanned interruption while executing a program. Ex:- while executing a program, if we got a value that is divided by zero is called an exception.

* Priority interrupt :-

↳ A priority interrupt is an interrupt which gets service on the basis of its priority.

↳ The system has authority to decide which conditions are allowed to interrupt the CPU, while some other execution is going.

↳ When two or more devices interrupt the computer simultaneously, the computer services the device with the higher priority first.

↳ Generally, devices with high speed transfer such as magnetic disks are given high priority and slow devices such as keyboards are given low priority.

DATE : __/__/____
PAGE : _____

* Interrupt handling :-

After every inst^n cycle the processor will check for interrupts. If there is no interrupt then next inst^n cycle will be executed.

If there is an interrupt present then it will trigger or call the interrupt handler, the handler will stop the present inst^n which is executing and save the configuration in register, and load the program counter by interrupt location. After processing the interrupt by the processor, interrupt handler will load the inst^n & its configuration from saved register process will start its processing where its left.

↳ The interrupt handler is also called ISR.

↳ There are different types of interrupt handler which will handle different interrupt. For example :- for clock in system will have its own interrupt handler, keyboard will have its own etc.

↳ ISR handles both maskable & non-maskable interrupt.

↳ All the processes which are performed by an interrupt handler are called interrupt handling.

* Modes of data transfer:-

Data transfer b/w the memory, CPU & I/O devices may be handled in variety of modes.

CPU as an intermediate
{ ① programmed I/O
② Interrupt initiated I/O
③ Direct Memory Access (DMA)

① Programmed I/O :-

Programmed I/O op^{n}s are result of I/O inst^{n}s written in computer program.

↳ Each data transfer is initiated by an I/O inst^n in the program to access register or memory.

↳ In this, to transfer data, the CPU requires constant monitoring of I/O devices.

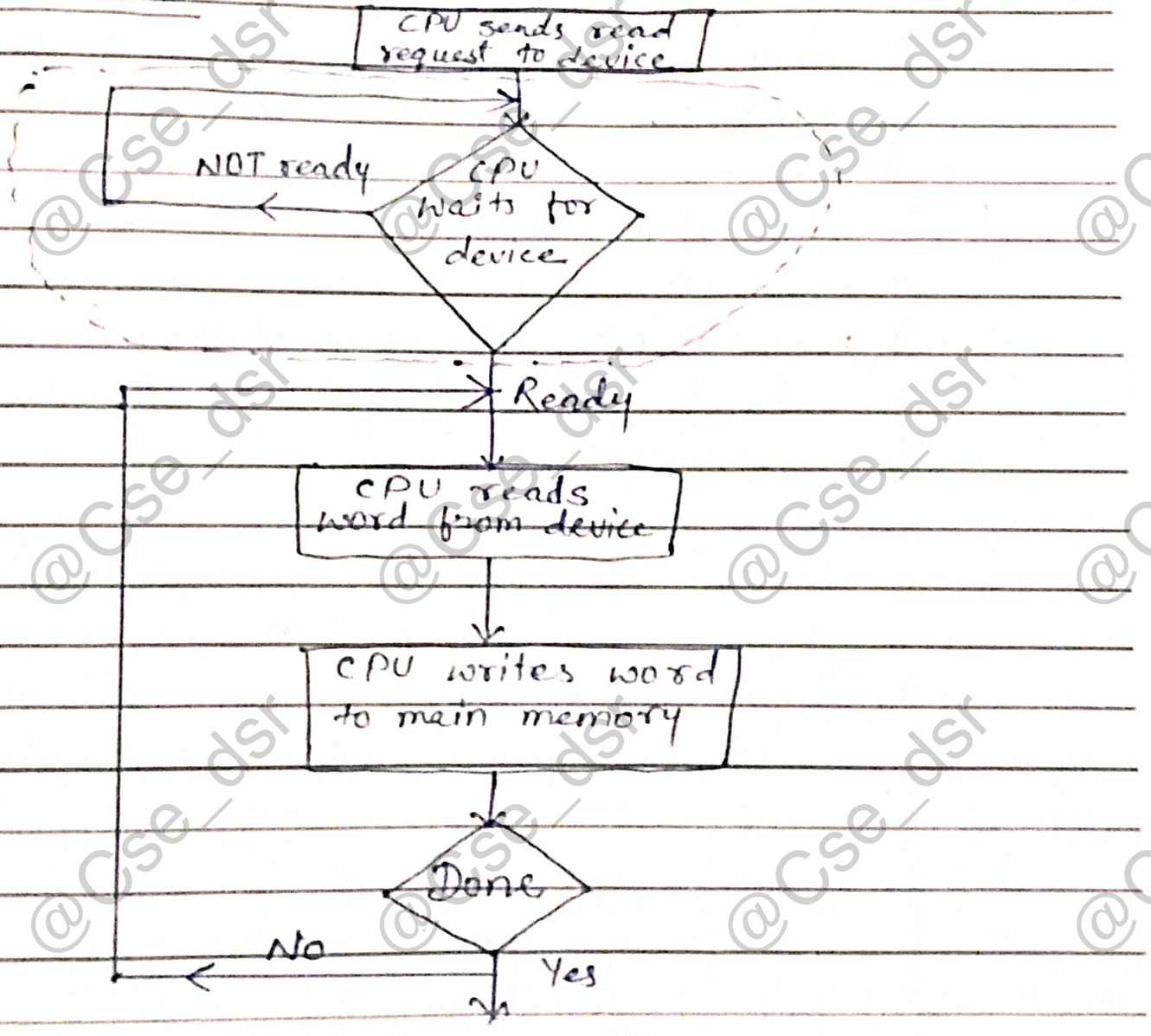
↳ Here, CPU makes a request & then CPU stays in program loop (called polling)

until the I/O devices is ready for data transfer. [The I/O devices takes no further action to alert the CPU.]

Disadvantage:-

This is time consuming process since it keep CPU busy. To avoid this problem interrupt initiated I/O mode is used.

Flow chart



DATE: / /
PAGE:

② Interrupt Initiated I/O:-

In this, instead of continous monitoring of CPU interface will be informed to issue an interrupt request signal, when I/O is free.

→ Meanwhile, CPU processes the other program & interface keep monitoring the device.

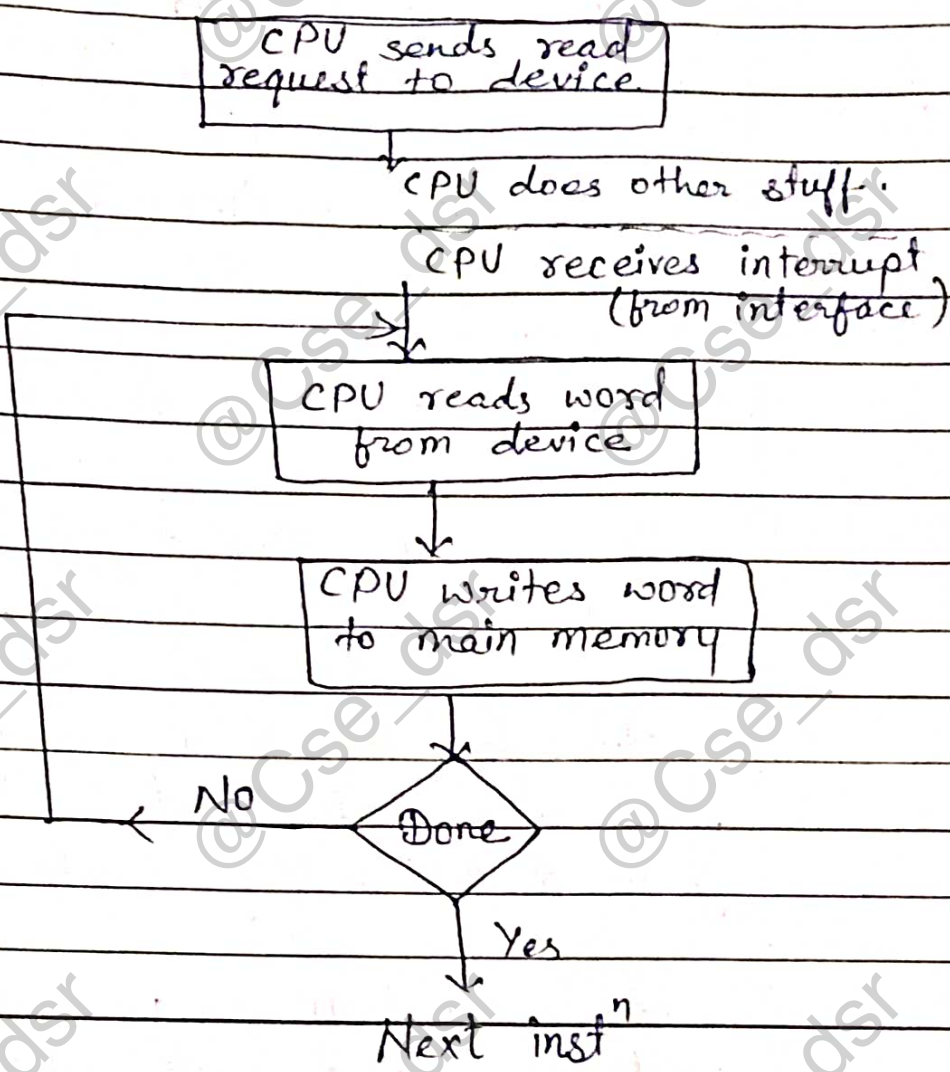
→ When device is ready for data transfer, interface generates interrupt request.

→ After getting external interrupt signal, the CPU stops the task it is performing, process the I/O data transfer & then resumes the original task it was performing.

→ Advantage of this is, the wait period of CPU is eliminated.

DATE : __/__/____
PAGE : _____

Flowchart :-



③ Direct Memory Access (DMA) :-

↳ To transfer large blocks of data at high speed between external devices & main memory, DMA approach is used.

↳ DMA allows data transfer directly b/w I/O devices and main memory with minimal intervation of CPU.

DATE: __/__/__
PAGE: _____

↳ DMA controller is nothing but the I/O interface to which CPU grants the authority to read & write memory without its involvement.

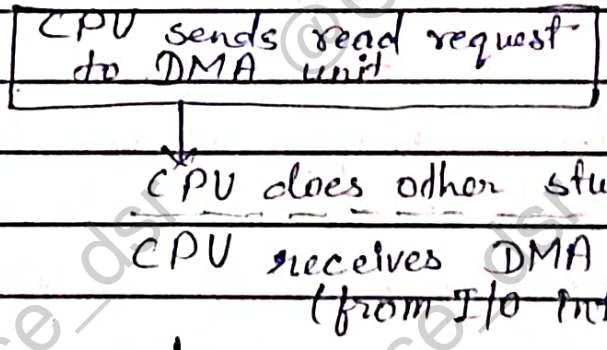
↳ DMA controller itself controls data transfer b/w main memory & I/O device.

↳ CPU is only involved in beginning & end of the transfer.

↳ CPU asks the DMA controller to transfer data b/w device and MM & then CPU proceed to execute other tasks.

↳ When data transfer is finished, the DMA controller interrupts the CPU.

Flowchart:-



DATE: __/__/__
PAGE: _____

* Input-Output Processor :-

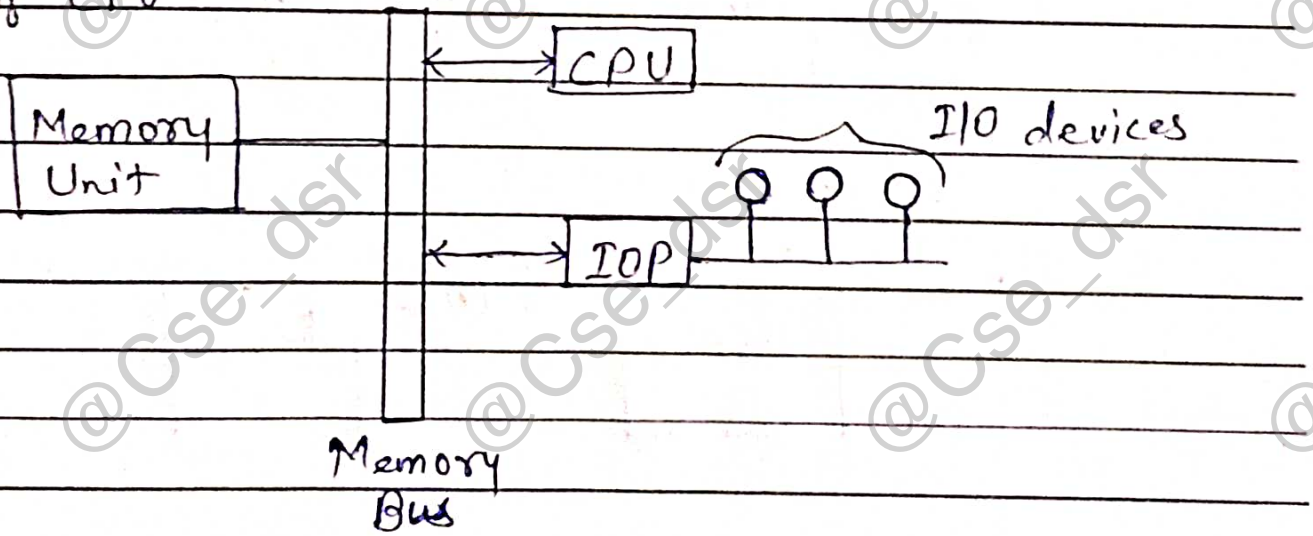
The Input Output Processor (IOP) is just like a CPU that handles the details of I/O operations.

↳ It is having more facilities than DMA controller.
It is enhancement of DMA controller.

↳ The IOP can fetch & execute its own inst^{n}s that are designed to characterize I/O transfers.
It can perform other processing tasks like arithmetic, logic, branching and code translation.

↳ It communicates with the process by the means of DMA.

↳ The advantage is the I/O devices can directly access main memory without intervention of CPU.



DATE: / /
PAGE:

* Asynchronous data transfer mode :—

↳ In this data transfer mode, the data transfer b/w sender & receiver is not synchronized with same clock pulse.

↳ This is used when the devices do not matches in speed.

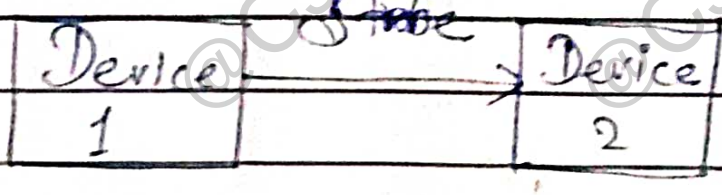
↳ In this all devices are independent and each have its own private clock.

↳ Asynchronous data transfer b/w two independent units requires control signals to be transmitted b/w communicating units. So, both devices can co-ordinate.

↳ In this two methods are used :—

i) Strobe control :—

A strobe signal is supplied by one of the units to indicate the other unit when the transfer has to occur.



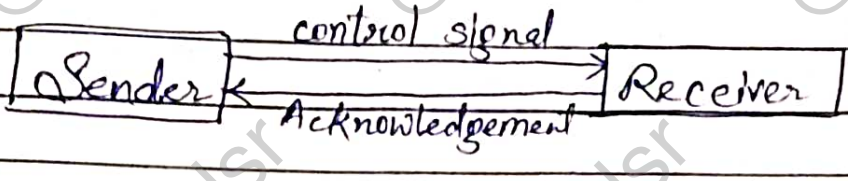
DATE : / /
PAGE :

ii) Handshaking method :—

↳ Here, a control signal is send by the sender to indicate data transfer is about to occur.

↳ Then the receiver responds with another control signal called acknowledgement.

↳ After that data transfer occurs.



* Synchronous data transfer mode :—

↳ In this mode, the units share a common clock pulse.

↳ The data transfer between sender and receiver is synchronized with same clock.

↳ This is used between the devices that matches in speed.

↳ It is faster than asynchronous data transfer mode because there is no need of any other extra control signal.

↳ The implementation of this move is costly.