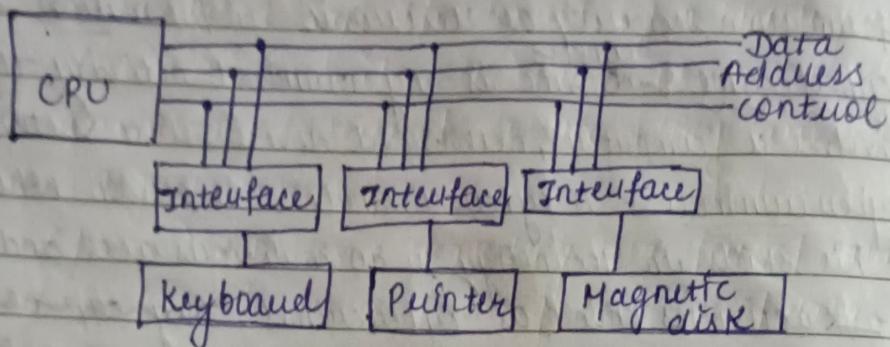


Unit - 05

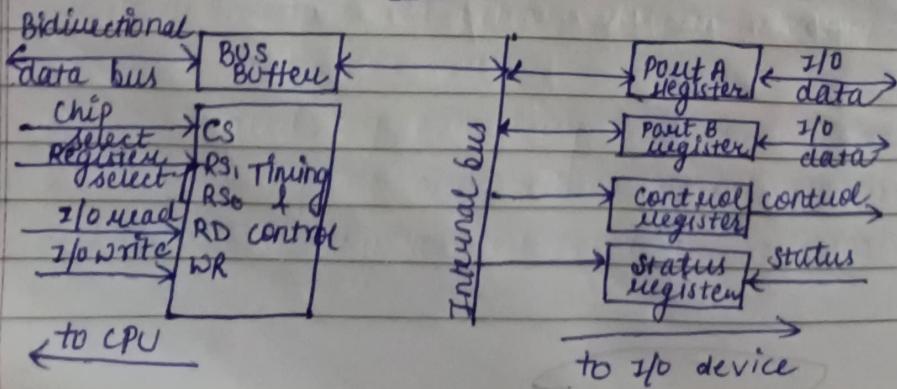
I/O Interface :- It is a hardware circuit used between CPU & I/O devices to supervise & synchronize all I/P & O/P transfers.



Requirement / Need of I/O interface :-

- ① I/O devices are electromechanical & electro-magnetic while CPU & memory are electronic devices. So they have different signals so conversion of signals is required.
- ② Data transfer rate of CPU is faster as compared to I/O devices. So synchronization is needed.
- ③ Data format of I/O devices are different from word format of CPU & memory.
- ④ Operating modes of I/O devices are different from each other and so each I/O device must be controlled so that it will not disturb the operation of other I/O devices to CPU.

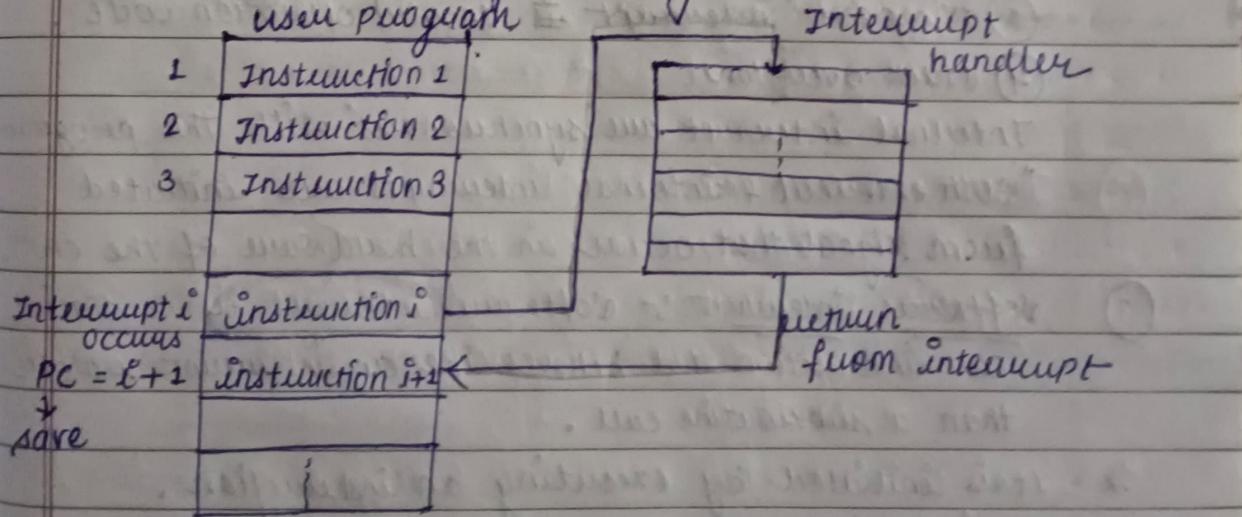
Example of I/O Interface :-



CS	RS <sub>1</sub>	RS <sub>0</sub>	Register select
0	X	X	None
1	0	0	Port A register
1	0	1	Port B register
1	1	0	control register
1	1	1	Status register

Program interrupt :- Program interrupt refers to the transfer of program control from a currently running program to another service program as a result of an external or internal generated request.

- \* Control returns to the original program after the service is executed.
- \* When an interrupt is initiated, the state of CPU containing the following information saved.
  - ① Content of the program counter (PC)
  - ② Content of all CPU registers
  - ③ Content of status bit condition
- \* CPU does not respond to an interrupt until the execution of currently running instruction ends.



Transfer of program control via interrupt

\* Before going to next fetch cycle, it checks for an interrupt signal, if there is an interrupt.

### Types of program interrupt :-

(1) External interrupt :- External interrupt are

generated by I/O devices, by a timing device from a circuit monitoring the power supply or from any external source. They are "asynchronous".

e.g :- ① I/O device requesting transfer of data.

② I/O device finished transfer of data.

③ Timeout interrupt is generated from program that goes into endless loop.

④ Power failure circuit also causes interrupt.

(2) Internal interrupt :- Internal interrupt are

generated from illegal or erroneous use of an instruction also called "Traps".

① Due to register overflow error.

② Divide by zero errors.

③ Timeout interrupt  $\leftrightarrow$  invalid operation code.

④ stack overflow

Internal interrupt are "synchronous" with the program.

"Both external & internal interrupts are initiated from signal that occurs in the hardware of the CPU".

(3) Software interrupt :- Software interrupt is a special call instruction that behaves like an interrupt rather than a subroutine call.

\* It is initiated by executing an instruction.

\* It is used by the programmer to initiate an interrupt procedure at any desired point in the program.

Eg :- Supervisor call instruction generates a software interrupt to switch from user mode to supervisor mode.

### Vectored Interrupt

- ① Vectored interrupt is an interrupt in which the address of the service routine is hardware.
- ② The address of the subroutine is already known to the CPU.
- ③ In 8085 microprocessor RST 5.5, RST 6.5, RST 7.5 + TRAP are vectorized interrupt.
- ④ An interrupt for which the internal hardware automatically transfers the program control to a specific memory location is called vectorized interrupt.
- ⑤ Find the address of these vectorized interrupt is very easy.

### Non-vectorized interrupt

- ① Non-vectorized interrupt is an interrupt in which the address of the address routine is supplied externally by the device.
- ② The device will have to supply the address of the subroutine to the microprocessor.
- ③ In 8085-microprocessor, INPR is the non-vectorized interrupt.
- ④ An interrupt for which an external hardware (eg I/O device) is required to provide the address at which the program control needs to be transferred is called non-vectorized interrupt.
- ⑤ The address in non-vectorized interrupt is not pre-defined.

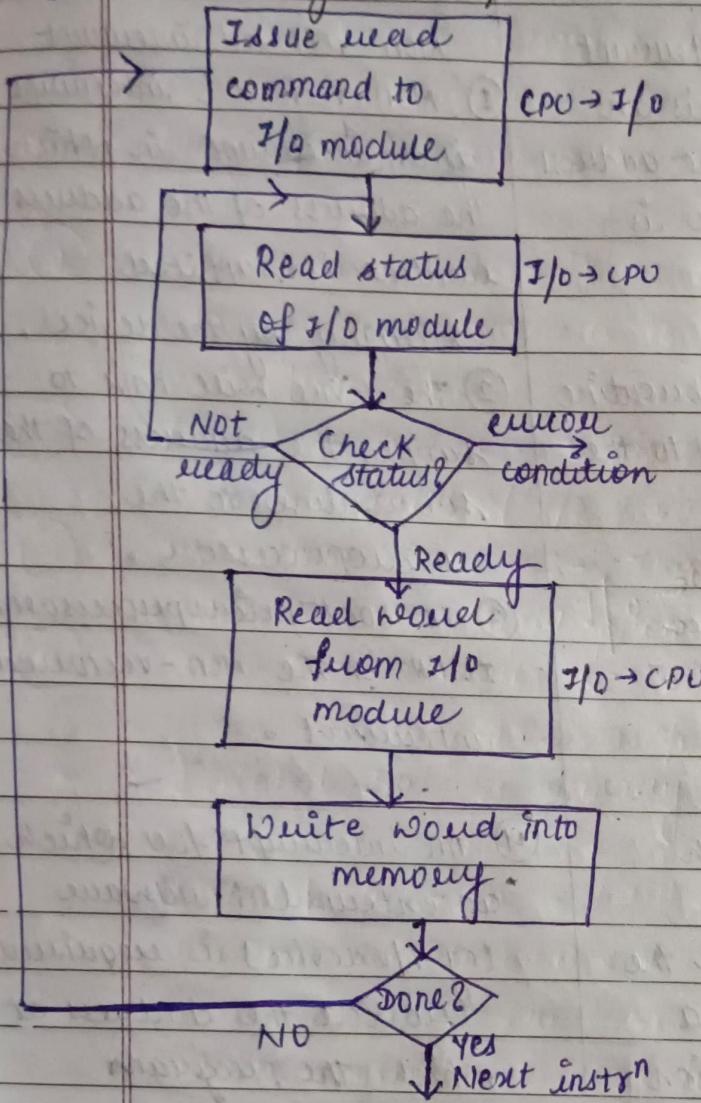
TRAP > RST 7.5 > RST 6.5 > RST 5.5 > INTR

vectorized interrupt have higher priority than non-vectorized interrupt.

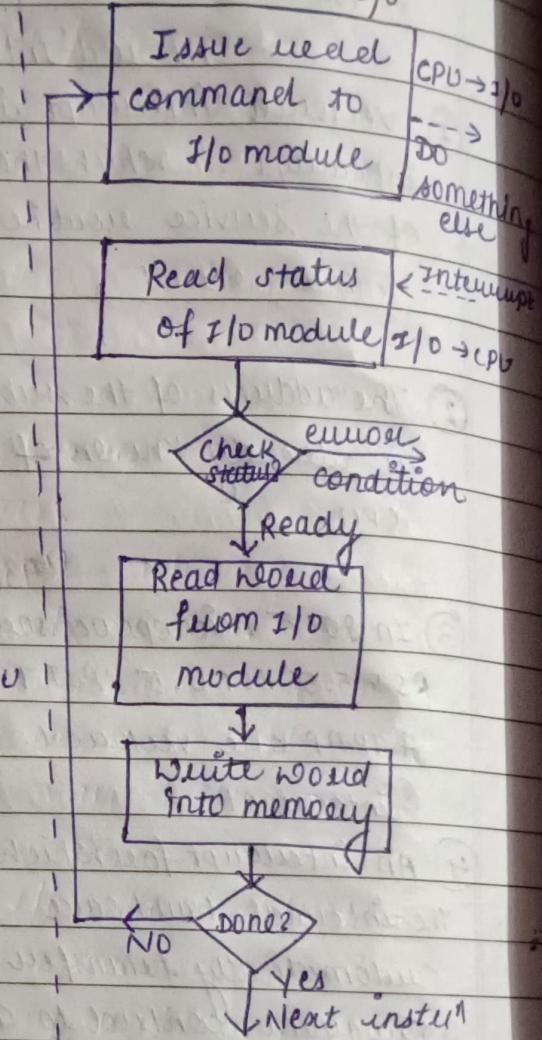
## \* Modes of transfer :-

(1)

Programmed I/O



(2) Interrupt driven I/O



(3)

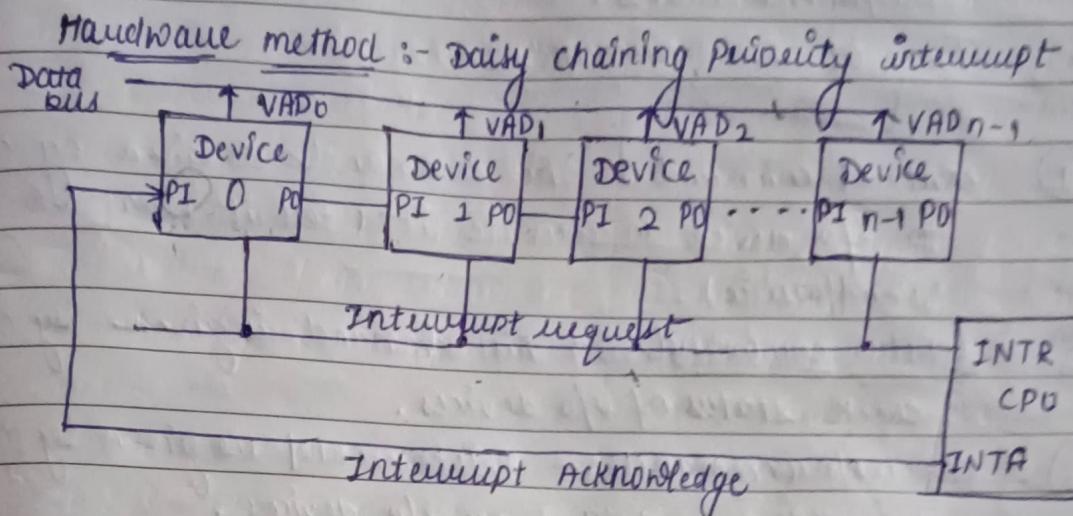
Direct memory access (DMA)

Priority interrupt :- A priority interrupt is a system in which a priority is established over a number of interrupt requests that which interrupt to serve first.

\*

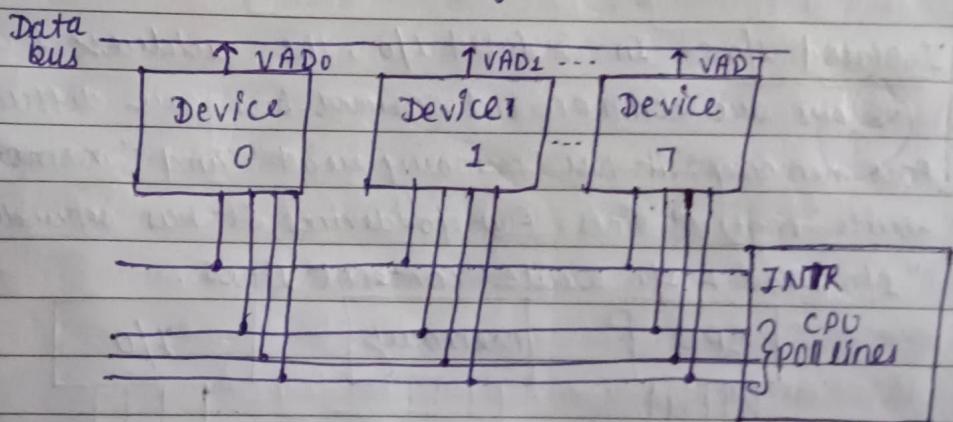
Devices with high speed transfers (e.g. magnetic disk, magnetic tape) in comparison to low speed devices (e.g. keyboard).

- \* When no. of devices interrupt the CPU the devices with high priority served first.
- \* Both software & hardware method are used to establish priority.



PI = Priority I/O, PO = Priority o/p  
 VAD = Vectorized address

Software method :- polling : let no. of I/O devices = 8



Polling software method  
for priority interrupt

Poll lines depends on no. of devices connected

## I/O commands

\* Definition :- commands are the instructions given by the CPU to I/O interface or I/O devices to perform specific operations.

\* Types of I/O commands :-

(1) control command :- This command is given by CPU to I/O device to activate it & tell it what to do (Read/write).

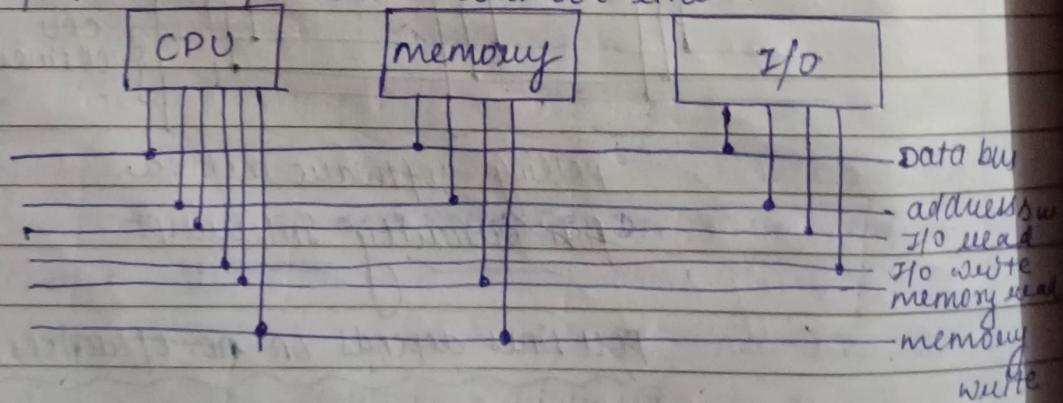
(2) Status command :- This command is given by CPU to check status of I/O devices.

(3) Data I/O Command :- This command is given by CPU to read any data from I/O devices.

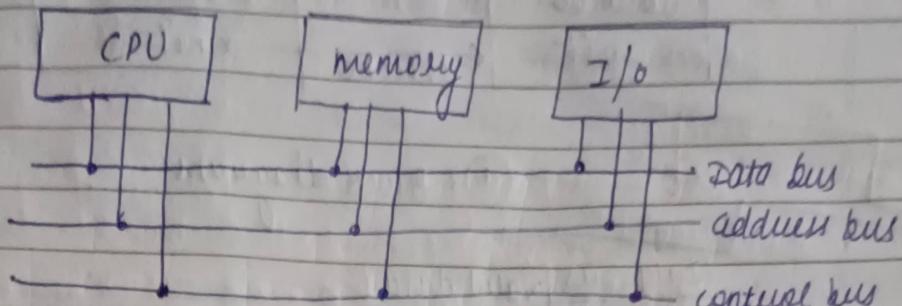
(4) Data Out command :- This command is given by CPU to write any data to I/O devices.

\* Isolated I/O vs memory mapped I/O :-

Isolated I/O :- In isolated I/O, the address & data bus are common but control bus are different. For memory, it has "memory read" and "memory write" control lines. For I/O devices it has separate "I/O read & I/O write" control lines.



memory mapped I/O :- address, data & control bus are common for memory & I/O devices.



### Asynchronous Data Transfer

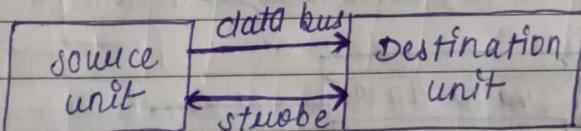
- A data transfer is said to be asynchronous data transfer, when two independent units uses their own internal clock. These independent unit does not make use of common clock.
- Synchronous data transfer is the one which has make use of common clock.

#### Types of asynchronous data transfer :-

- ① Strobe control
- ② Handshaking

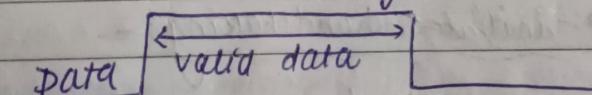
Strobe control :- In strobe control there is no knowledge about whether receiver receives the data or not.

#### (i) source initiated strobe

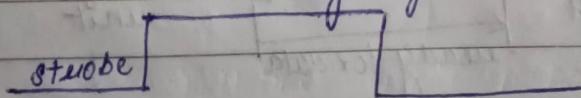


Method :-

(a) Block diagram



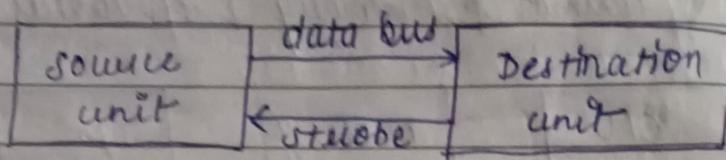
(b) Timing diagram



(c) Timing diagram

There is no acknowledge method in strobe control method.

(ii) Destination initiated strobe :-



① Block diagram

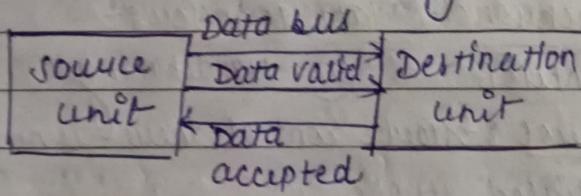
Data bus / valid data

strobe

② timing signal

Handshaking methods :- strobe control + acknowledgement signal

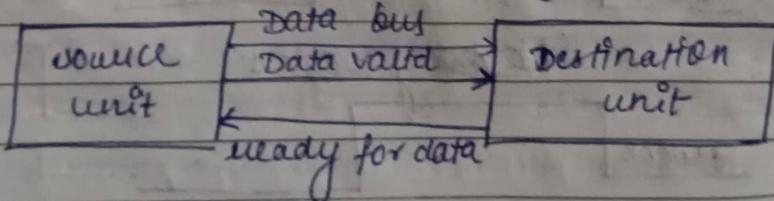
source - initiated handshaking :-



① Block diagram

Data bus      valid data  
Data valid        
Data accepted     

Destination - initiated handshaking :-



② Block diagram

Ready for data

Data bus

Data valid

### (b) Timing diagram

Aynchronous serial Transfer (communication)

In serial ~~trans~~ transmission, binary information is sent only when it is available and the line remain idle, when there is no information to be transmitted.

- \* Both sender & receiver have their own clocks and these clocks works independent to each other.

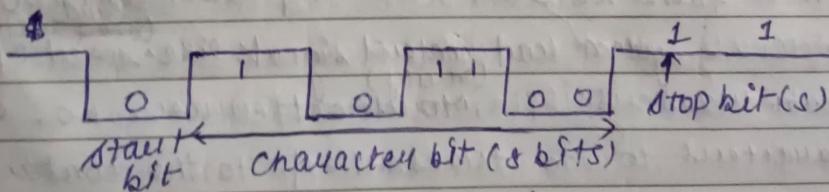
- \* The transmission process as follows :-

- i) When a character is not being sent, the line is kept in the 1st state.

- ii) The initiation of a character transmission is detected from the start bit, which is always 0.

- iii) The character bit always follows the start bit.

- iv) After the last bit of the character is transmitted a stop bit(s) is detected which is always "1".

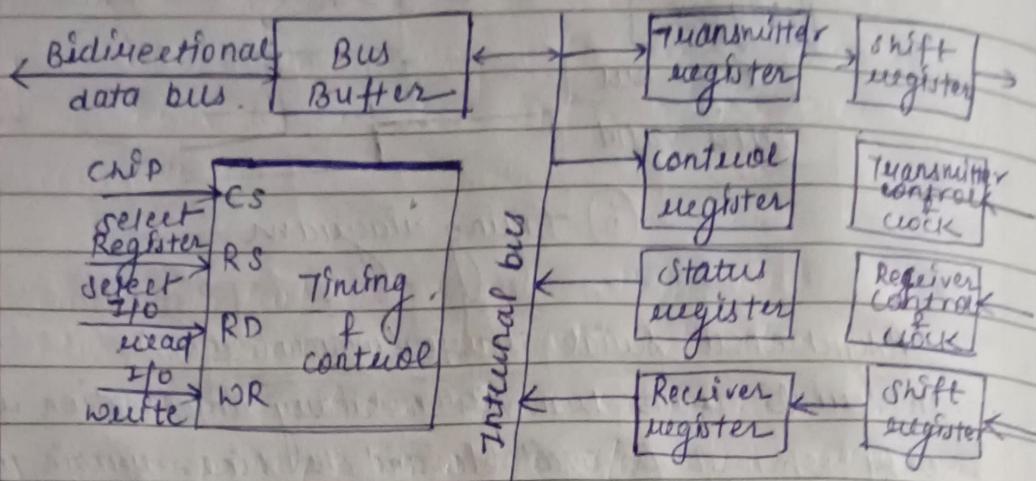


Asynchronous communication Interface

standard communication interface

UART (universal asynchronous ~~trans~~ communication interface)

works both as transmitter & receiver.



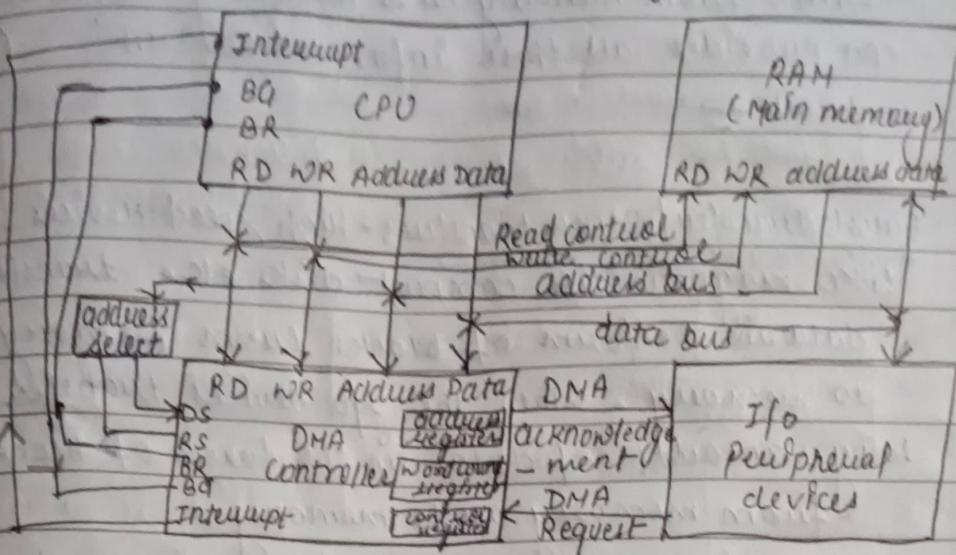
CS	RS	Operation	Register selected
0	X	X	None
1	0	WR	Transmitter register
1	1	WR	control register
1	0	RD	Receiver register
1	1	RD	status register

→ (write) ← (read)

- When transmission has been done from one computer to another first we select chsp and select control register. System load control signals like speed of data transmission, to detect errors, no. of characters to be transmitted to the control register. Now system check status register whether transmitter register is empty or not and if it is empty then by using bidirectional bus data has been transmitted to transmitter register.

After transmission status register shows that transmitter register is full i.e. it is ready for transmission.

## DMA (direct memory access) transfer



→ When there is a transmission of data from memory to I/O devices or from I/O devices to memory there is a need of CPU for transmission. During the I/O transmission there is a major involvement of CPU to minimize that involvement so that CPU can do another work we use DMA for the transmission.

→ DMA controller is a hardware device used to minimize the task done by the CPU from I/O devices or to I/O devices.  
Components of DMA controller :-

DS = DMA select, RS = Register select

RS is a combination of two R8 & R9 used to select b/w three registers address register, word count register & control register.

BR = Bus request, BG = Bus grant & interrupt are connected from CPU to DMA controller.

⇒ Address register :- used by the CPU to store the starting address from where the data transfer need to happen through DMA controller.

⇒ word count register :- Number of words to be transferred from memory to I/O devices or from I/O devices to memory.

# control register :- controls the data to be read or write from memory or to memory respectively. CPU provides all these info. to the DMA controller.

\* Burst transfer / Block transfer :- High speed devices like magnetic disk, compact disk etc. transfer data in the form of blocks from memory or to memory. When the I/O devices transfer large amount of data in the form of block from memory or to memory.

\* Cycle stealing :- cycle stealing is a process in which DMA controller over DMA process used to transfer single word from I/O peripheral devices to memory.

Process on working :-

- ① I/O peripheral devices request DMA controller through DMA request line.
- ② After that DMA controller sends bus request to CPU. Before bus grant CPU provides some basic info. to DMA for data transmission.
- ③ Through address line CPU first select DMA through address select and after that it select register from among three registers of DMA.
- ④ Then CPU provides the following registers info. to the DMA controller.
- ⑤ After providing all info. CPU grant bus to the DMA controller.
- ⑥ After that DMA acknowledge I/O peripheral devices to use read control, write control, data bus & address bus.

As, the word count decreases and when

register value

word count value reaches below "0", it tells DMA controller that data transfer has been done and DMA controller removes the DMA acknowledge from I/O peripheral devices.

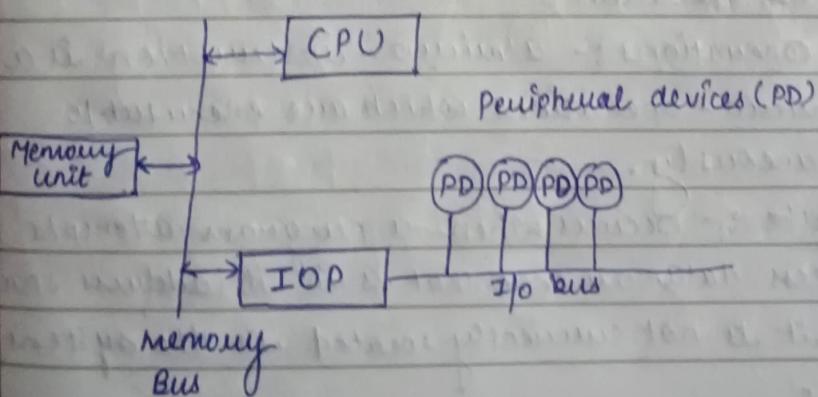
After all DMA controller sends interrupt signal to CPU to tell CPU that data transfer has been done by DMA controller and then bus grant signal has been deactivated by the CPU.

→ It is useful when there is a burst transfer of data.

### Input-output processor(IOP)

#### Input-output processor(IOP) :-

- ① The IOP is similar to CPU except that it is designed to handle only I/O processing.
- ② The IOP fetches & executes I/O instructions to facilitates I/O transfer. If required, it can perform all the functions of CPU.
- ③ Both CPU + IOP exists in the system however CPU is the master, while IOP is slave.
- ④ The CPU only initiates the I/O program, after that IOP operates independent of the CPU.



## CPU - IOP communication :-

