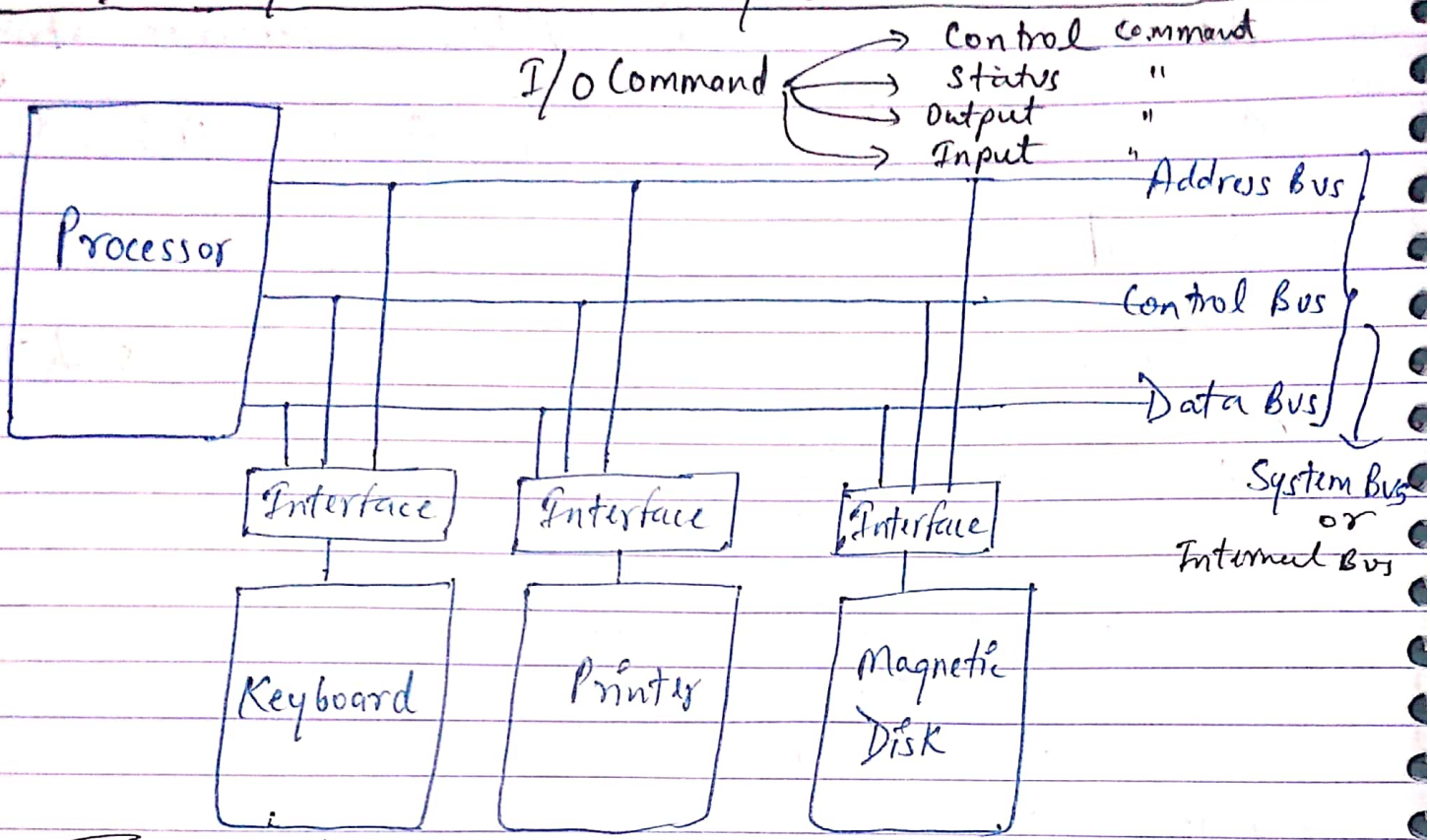
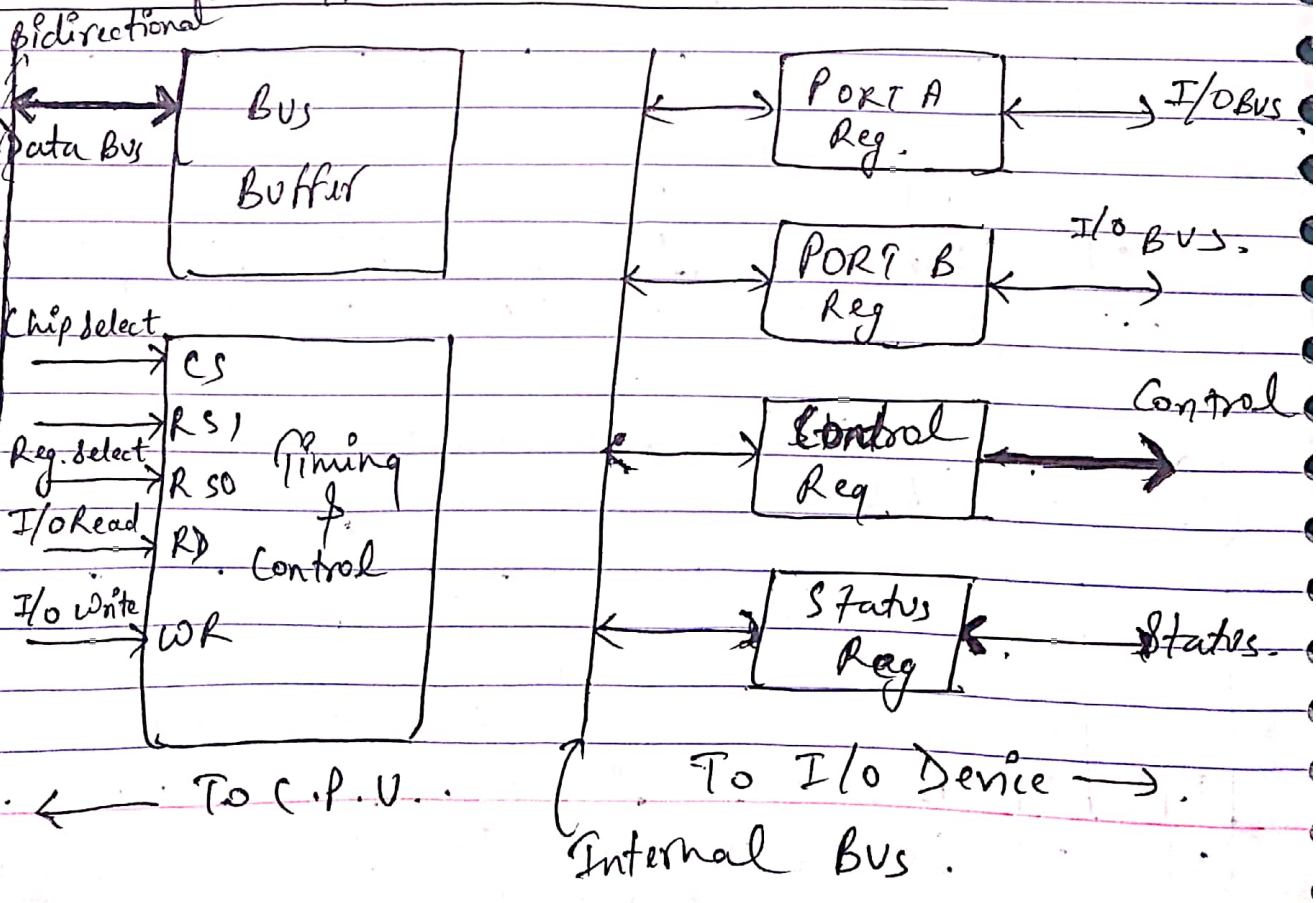
UNIT-5

Date - 19/09/19

INPUT/OUTPUT BUS & INTERFACE UNITS -



BLOCK DIAGRAM OF INTERFACE -



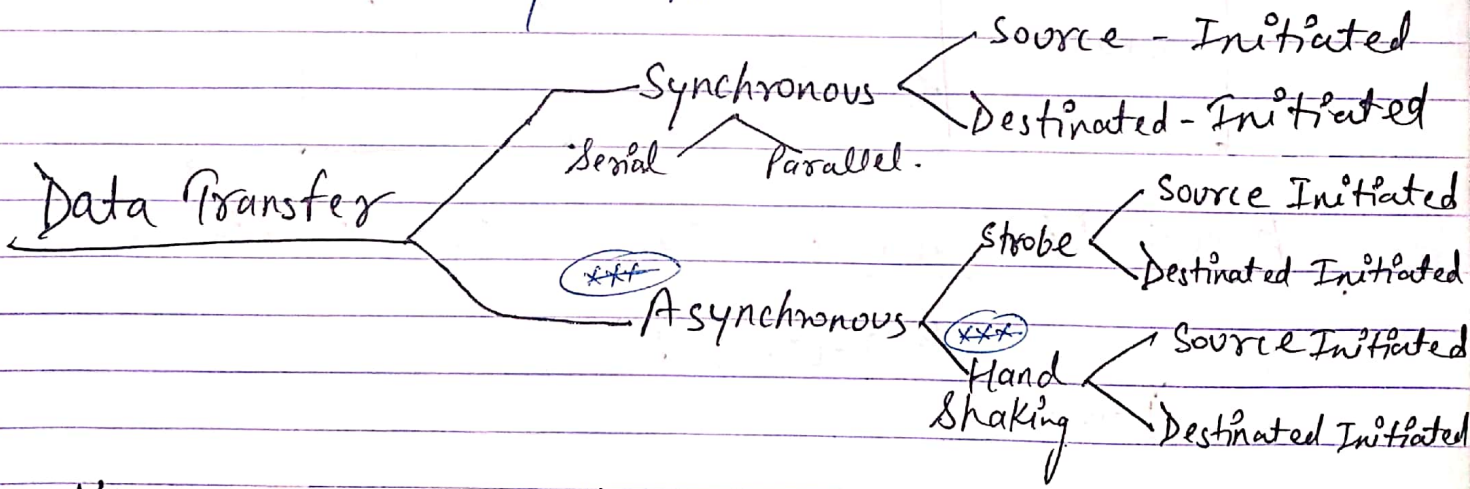
CS	RS1	RS0	Reg. Selected.
0	X	X	None
1	0	0	Port A Reg
1	0	1	Port B Reg.
1	1	0	Control Reg.
1	1	1	Status Reg.

COMMUNICATION B/W C.P.U & IO/MEMORY (3-methods)

(i) Seperate data, Address & control lines of IO Memory (then I/O Processor

(ii) Common Data & Address lines but seperate control ling (Isolated I/O)

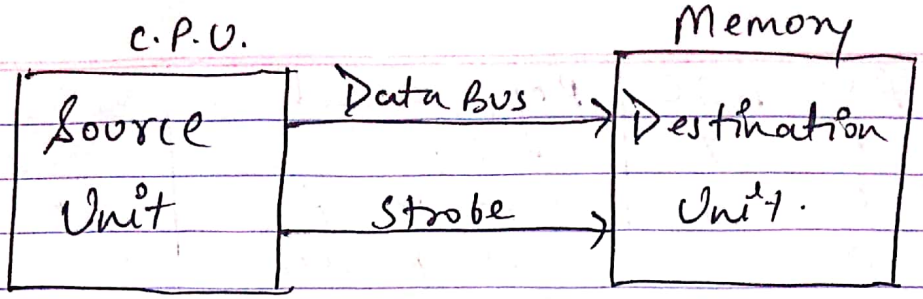
(iii) Common Data, Address & control Buses for IO & memory (Memory-mapped I/O)



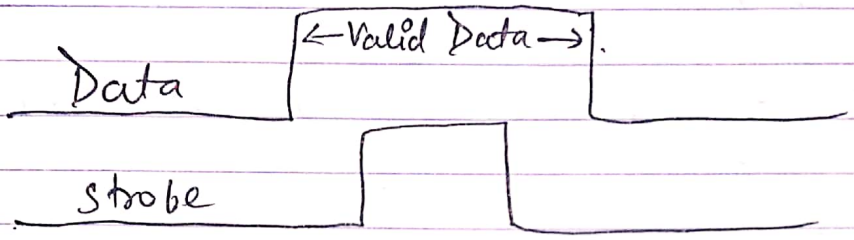
ASYNCHRONOUS DATA TRANSFER -
(ex- E-mail, message, whatsapp

* Strobe - (It is a single control line).

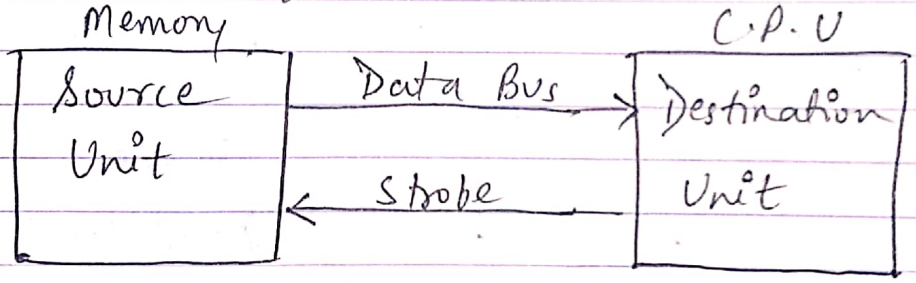
* Strobe -



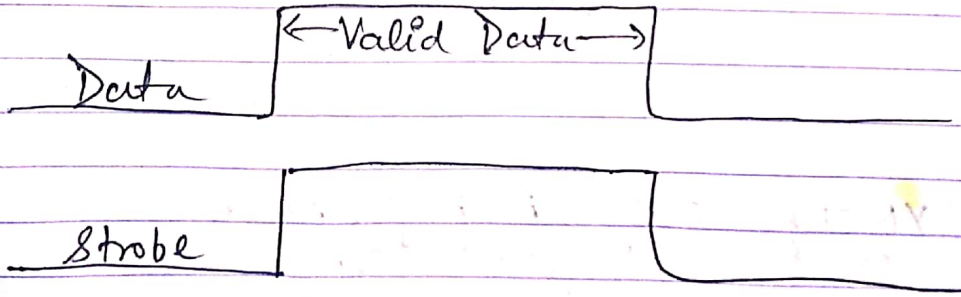
Block Diagram of Source - Initiated



Timing Diagram of Source - Initiated.



Block Diagram of Destination - Initiated.

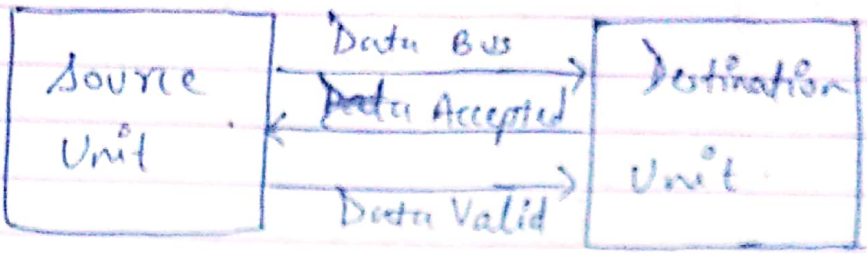


Timing Diagram of Destination - Initiated.

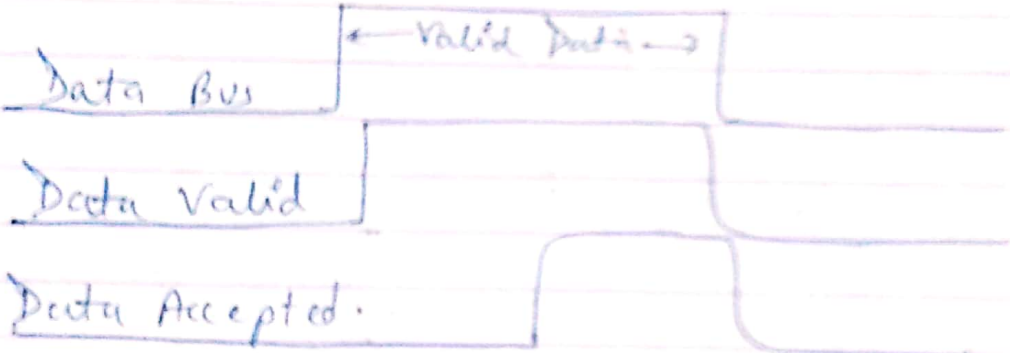
Imp ***

HANDSHAKING (Asynchronous Data Transfer) -

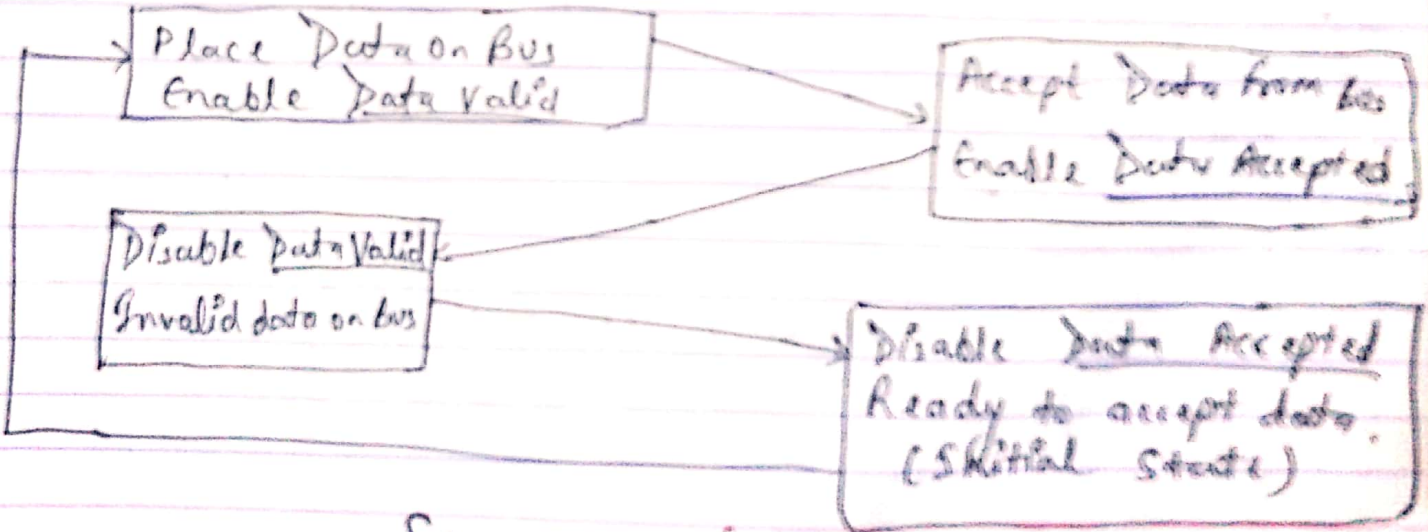
* Source - Initiated -



Block Diagram

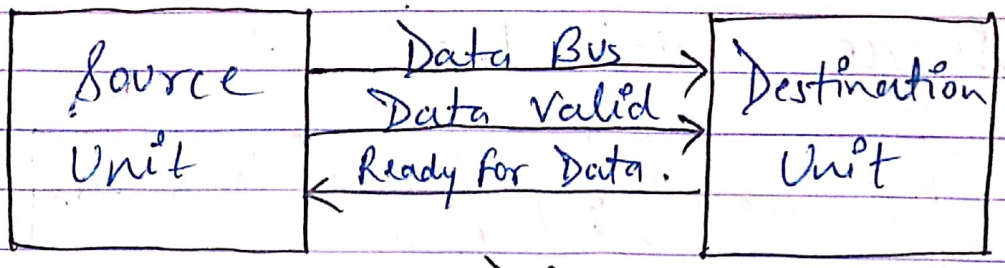


Timing Diagram

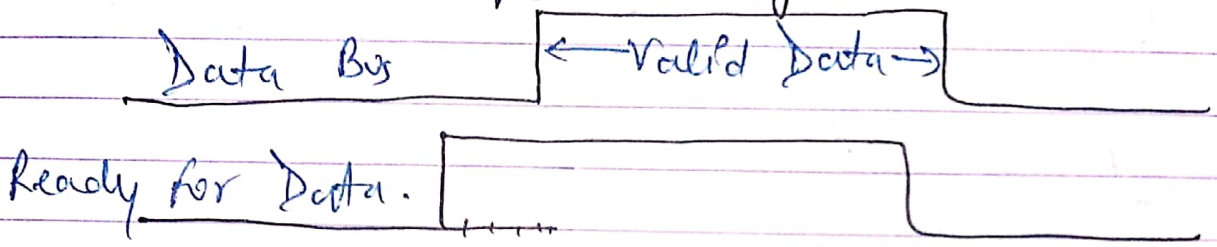


Sequence of Events

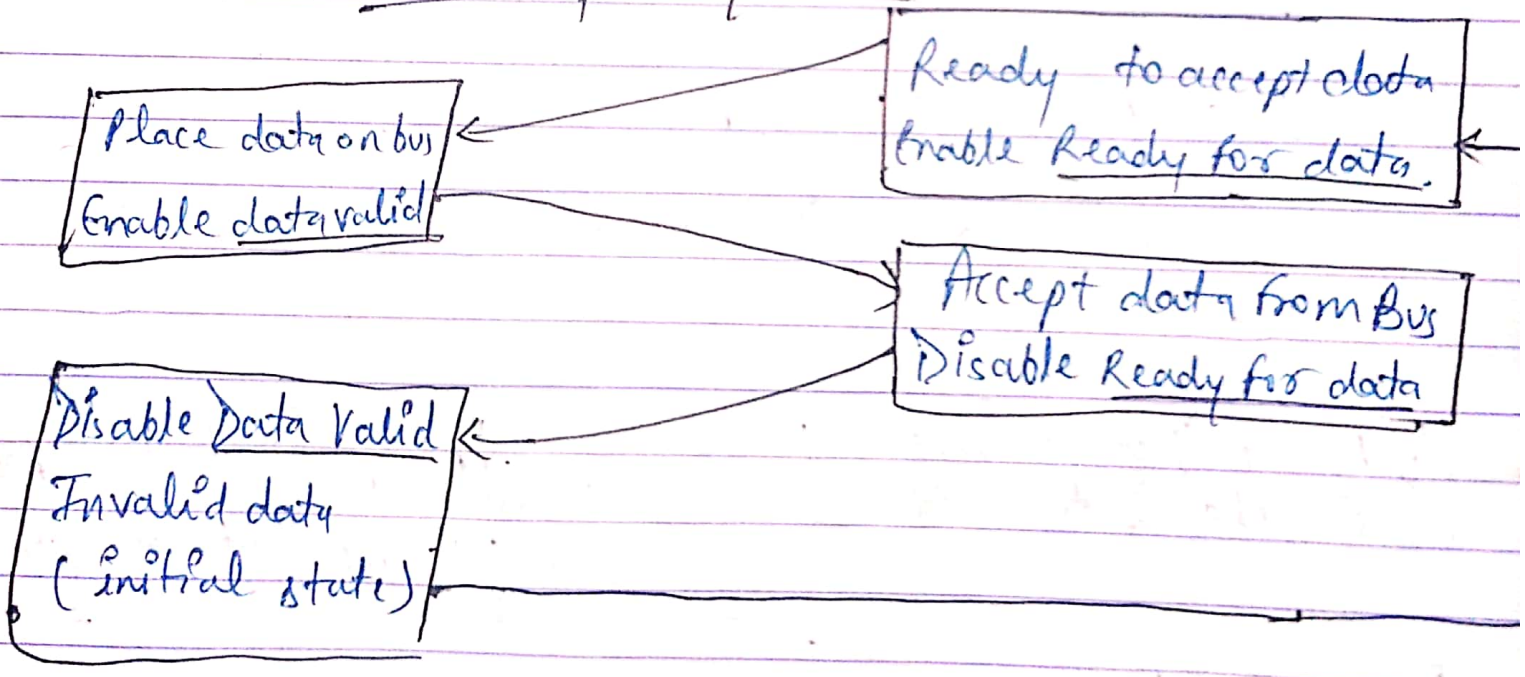
* Destination - Initiated -



Block Diagram



Timing Diagram

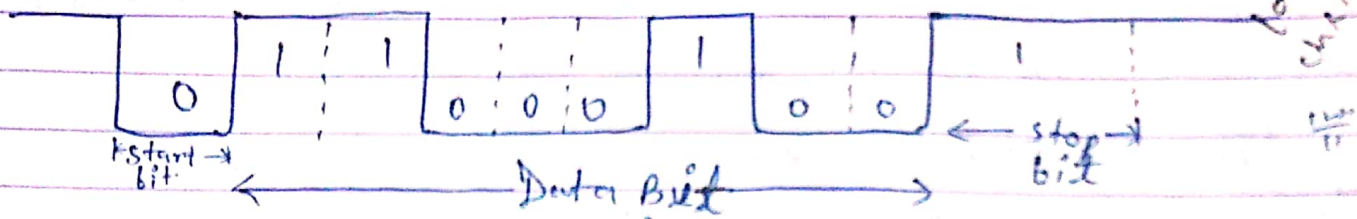


Sequence of Events

NOTE- The transfer occurs b/w C.P.U & Interface Unit strobe method is used & b/w interface unit & I/O device handshaking method is used.

* SERIAL ASYNCHRONOUS TRANSFER -

(ex- Keyboard, printer)



Serial Asynchronous data transmission

→ Start Bit (1-bit / 0)
→ Character Bit (n- bits (8/16 bits))
→ Stop Bit (1 or 2 bits / 1)

Q If 10 character / second transfer. How many bits / sec transfer?

⇒ 100 bits / sec or 100 baud.

Exercise -

Q How many characters per second can be transmitted over a 1200 baud line in each of the following modes. (Assume a character code of 8 bits)

(a) Synchronous Serial Transmission
(b) Asynchronous Serial Transmission with two stop bits
(c) Asynchronous Serial Transmission with one stop bitts.

Solution (i) 1200 baud = 1200 bits/sec

$ch \times 8 = 1200$

$ch = \frac{1200}{8} = 150$ character/sec.

(ii) 1200 baud = 1200 bits/sec.

data (8+1+2): 1 start, 2 stop

$ch \times 11 = 1200$

$ch = \frac{1200}{11} = 109.9 \simeq 110$ character/sec

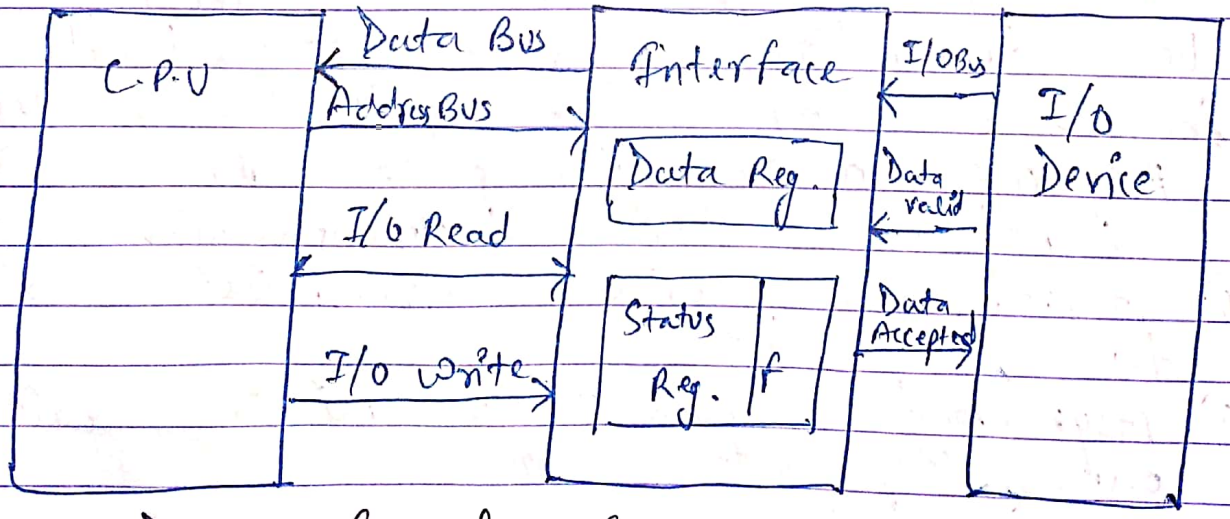
(iii) 1200 baud = 1200 bits/sec.

$ch \times 10 = 1200$

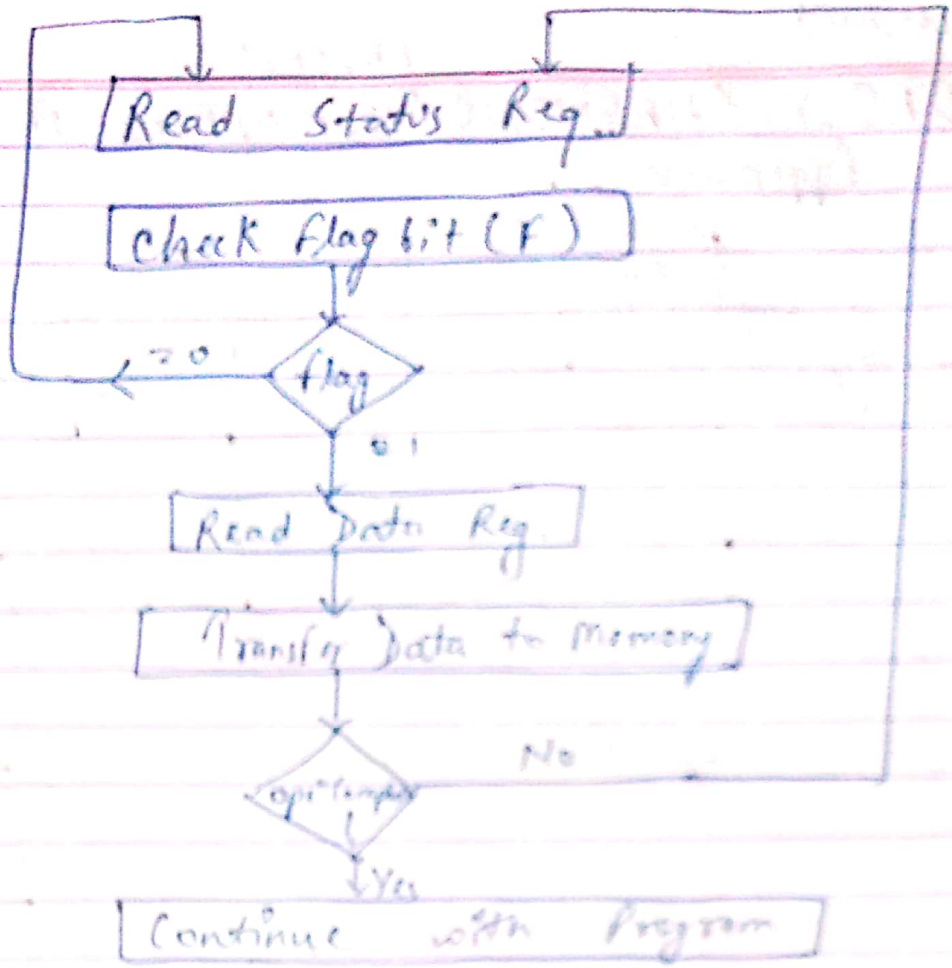
$ch = \frac{1200}{10} = 120$ characters/sec

MODES OF TRANSFER

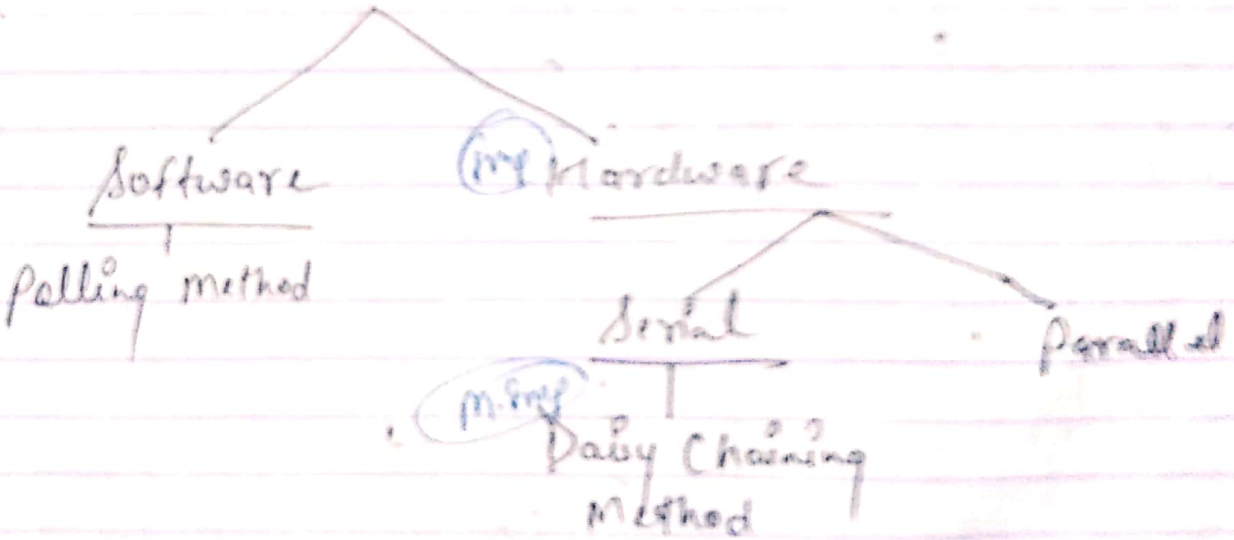
- Programmed I/O
- Interrupt I/O
- DMA (Direct Memory Access)



Data transfer from I/O Device to C.P.U



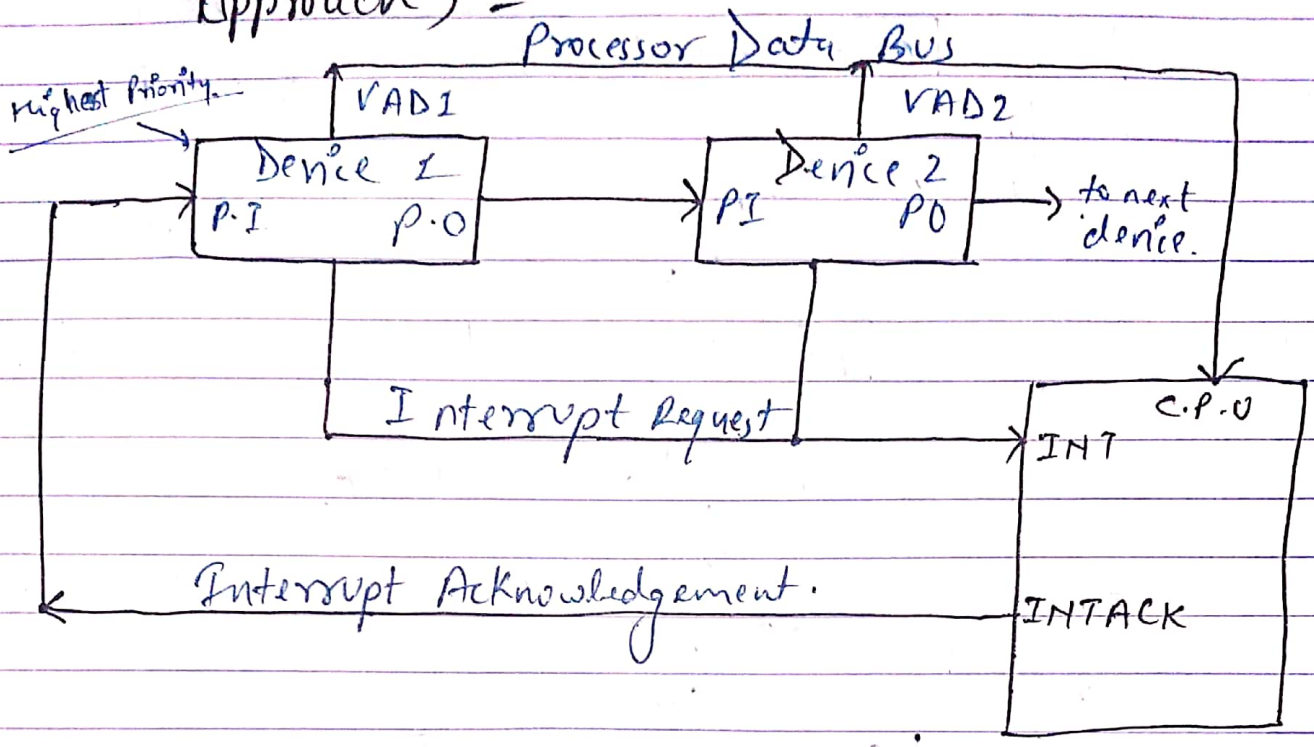
* INTERRUPT INITIATED I/O

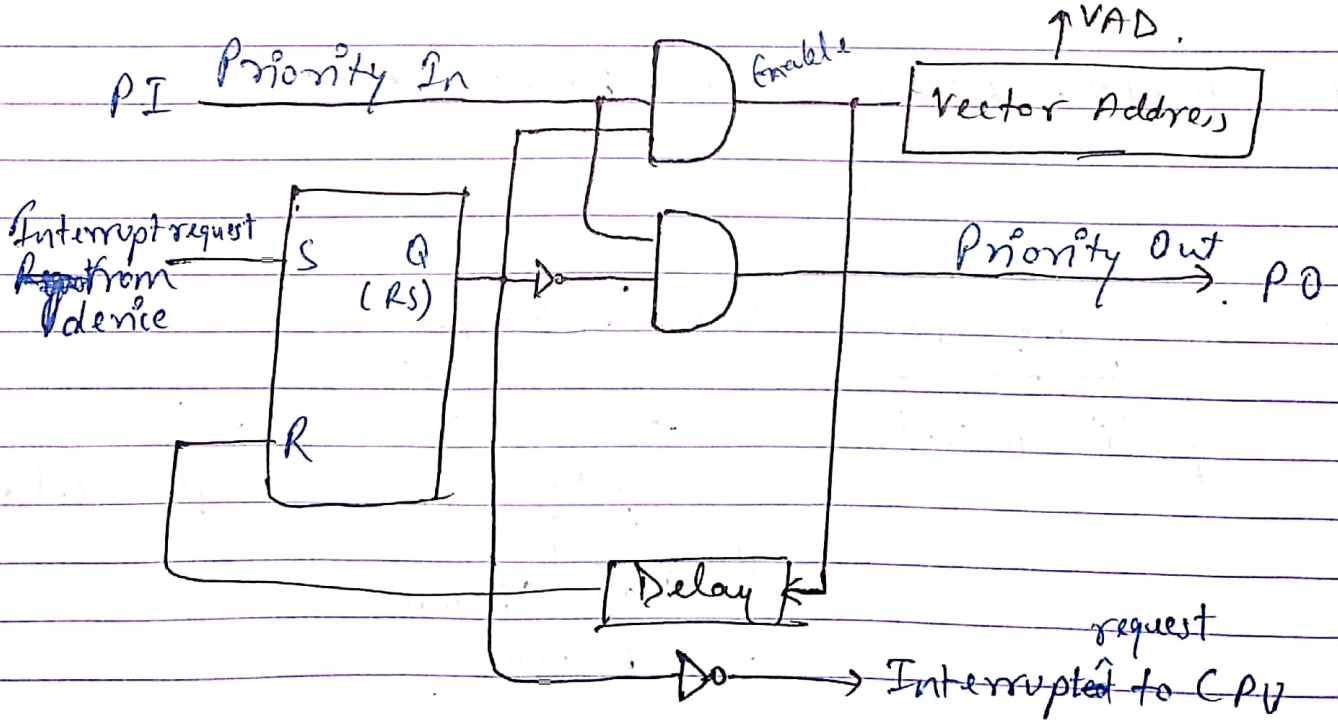


Speed → Priority

PRIORITY

* DAISY CHAINING / INTERRUPT METHOD (Serial Approach) -



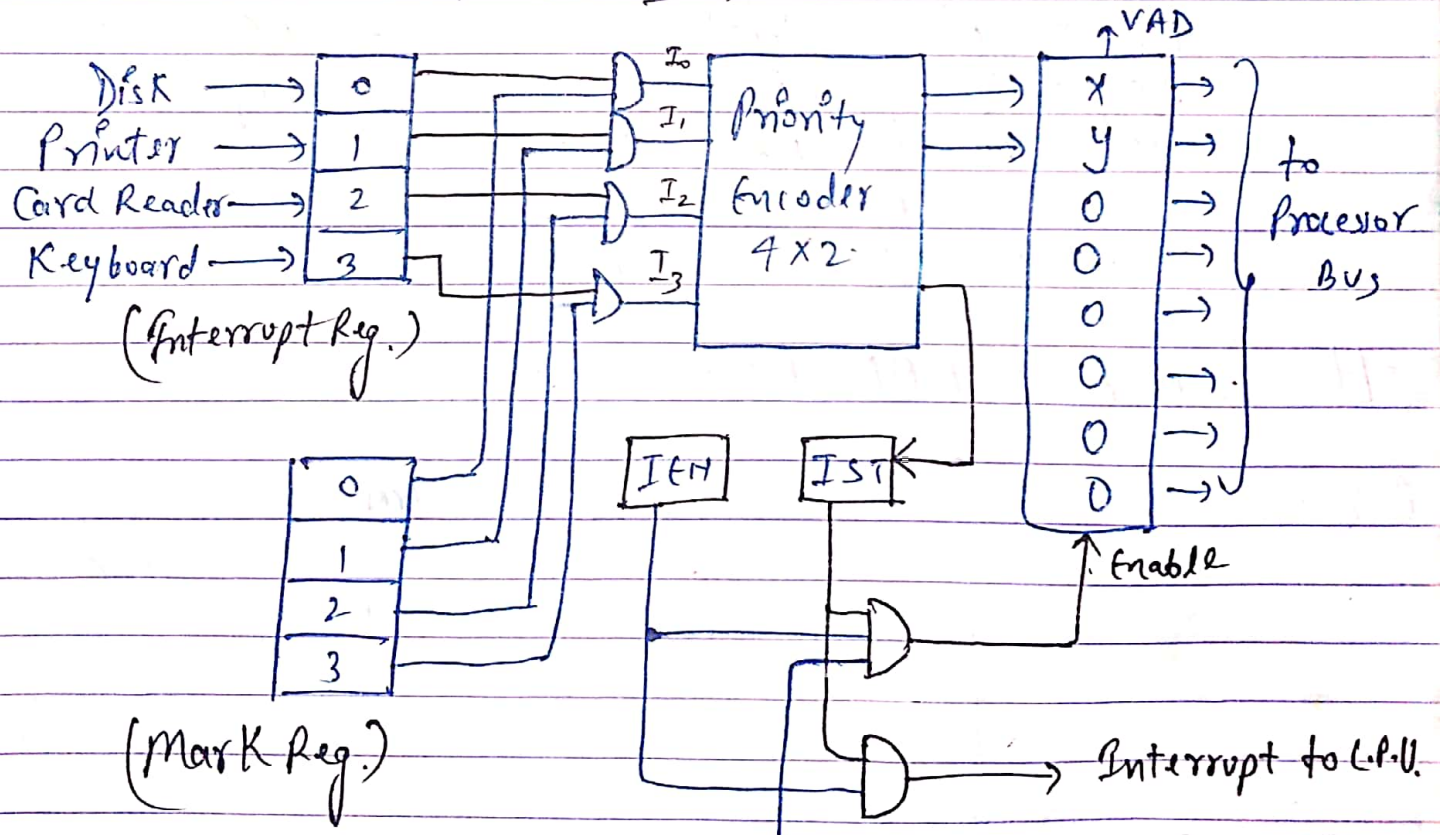


Each device issues a signal called interrupt request. & then C.P.U issues a signal called interrupt acknowledgement. to Device 1. (PI = 1 & PO = 0) Enable VAD

PI	RS	PO	Enable
0	0	0	0
0	1	0	0
1	0	1	0
1	1	0	1

* Advantages - Simplicity & Scalability.
* Disadvantages - * Priority depends on the physical position of devices.
  * Propagation delay increases.
  * failure of any device fails entire system.

PARALLEL PRIORITY INTERRUPT -



IST → Interrupt Status f/f
IEN → Interrupt Enable f/f (f/f always store one bit).
(overall control in the entire system)

Most Imp.

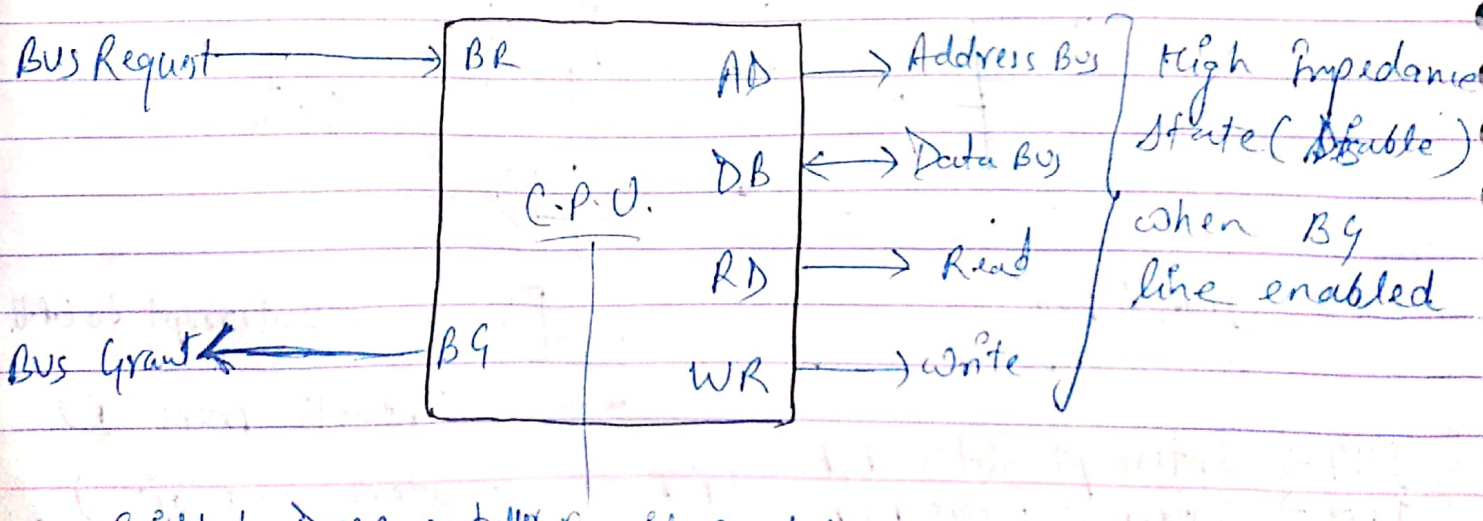
INTERRUPT CYCLE -

At the end of each instruction cycle the C.P.U. checks IEN & the interrupt signal from IST. If either is equal to zero, control continues with the next instruction. If both IEN & IST are equal to one (1) the C.P.U. goes to an interrupt cycle. During interrupt cycle the C.P.U. performs the following sequence of micro-operations.

$SP \leftarrow SP-1$	Decrement Stack Pointer
$M[SP] \leftarrow PC$	Push PC into stack.
$INTACK \leftarrow 1$	Enable interrupt acknowledgement.
$PC \leftarrow VA$	Transfer Vector address to PC
$IEN \leftarrow 0$	Disable further interrupts.
Goto fetch next instruction.	

PC → Programme counter
SP → Stack Pointer

DMA TRANSFER -



DMA Transfer → Burst (Bulk) Transfer.
→ Cycle Stealing.

When the DMA takes control of the bus system (buses), it communicates directly with the memory. The transfer can be made in several ways. In DMA burst transfer a block sequence consisting of a no. of memory words is transferred continous burst. while the DMA controller is master of the memory busses. This mode of transfer needed for fast devices such as magnetic disc, where data transmission can not be stopped or slow down untill a entire block is transmitted.

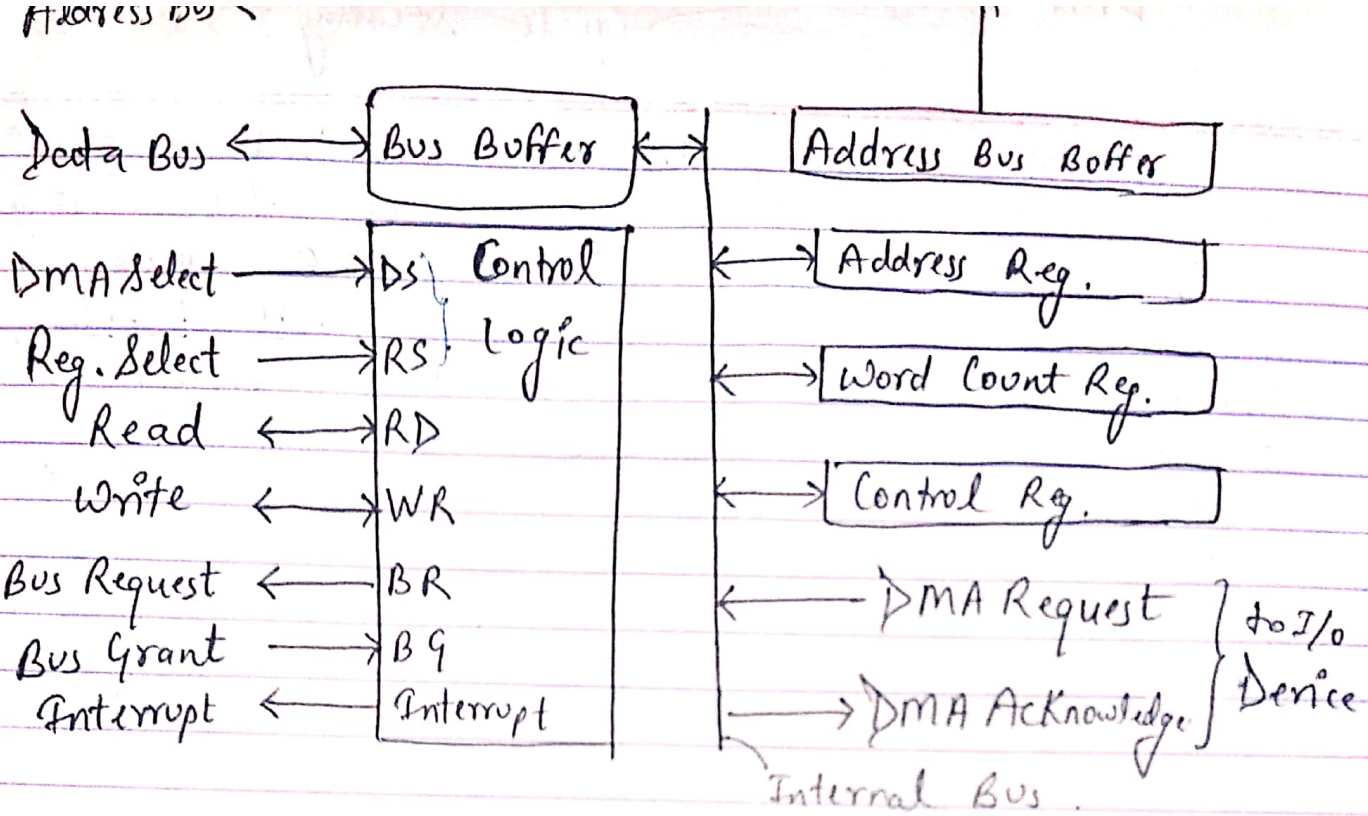
An alternative technique called cycle stealing. Allows the DMA controller to transfer one data word at a time after which it must return control of the busses to the C.P.U. The C.P.U. only delays it's operation for one memory cycle to allow the direct memory I/O transfer. to steal one memory cycle.

M. Pmp

DMA CONTROLLER -

The C.P.U first initializes the DMA by sending the following information through the data bus -

(i) The starting address of the P-I-O memory block where data are available (for read) or where data are to be



Block Diagram of DMA Controller.

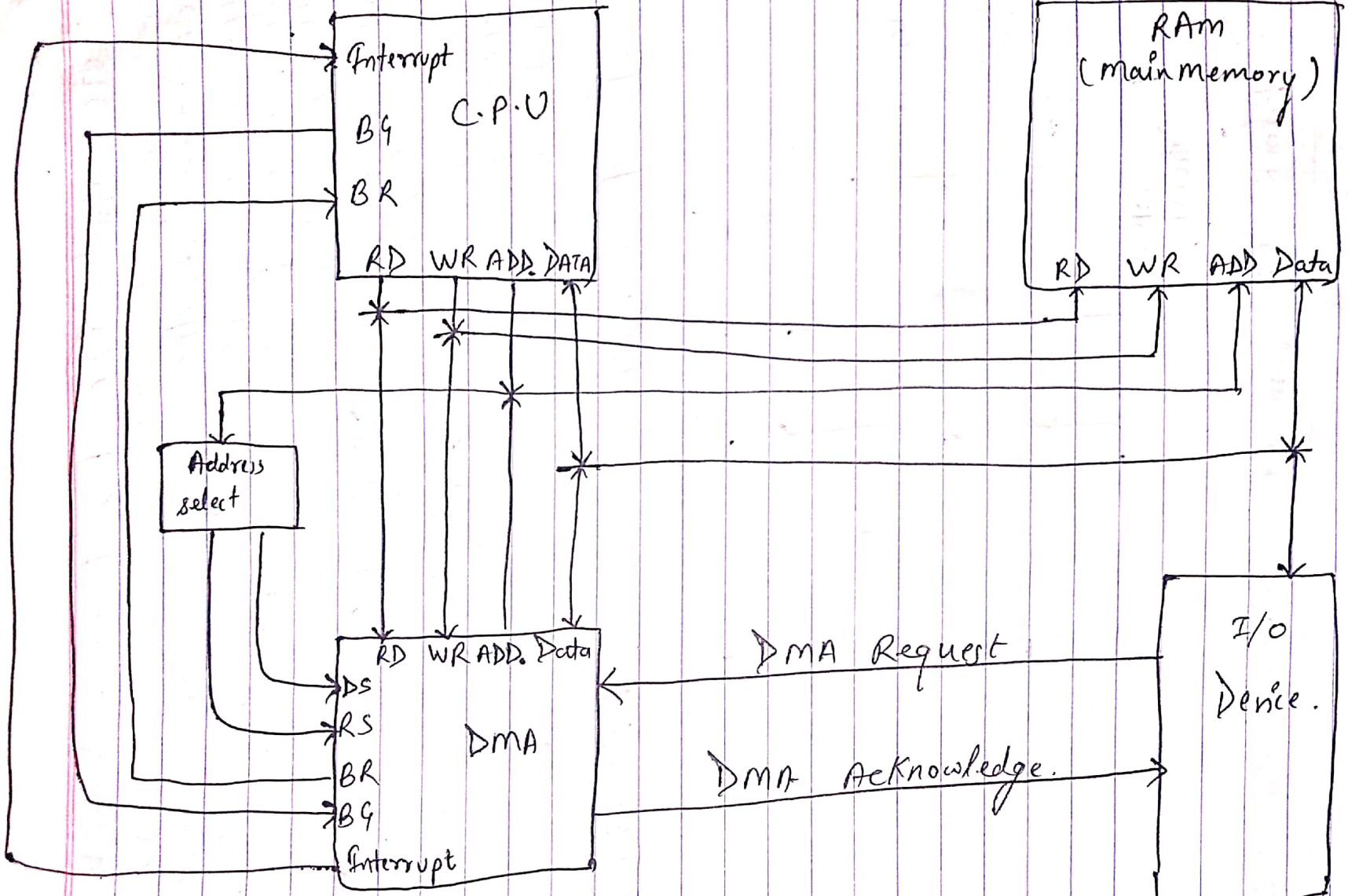
stored (for write).

(ii) The word count, which is the no. of words in the memory block.

(iii) Control to specify the mode of transfer such as read or write.

(iv) A control to start the DMA transfer.

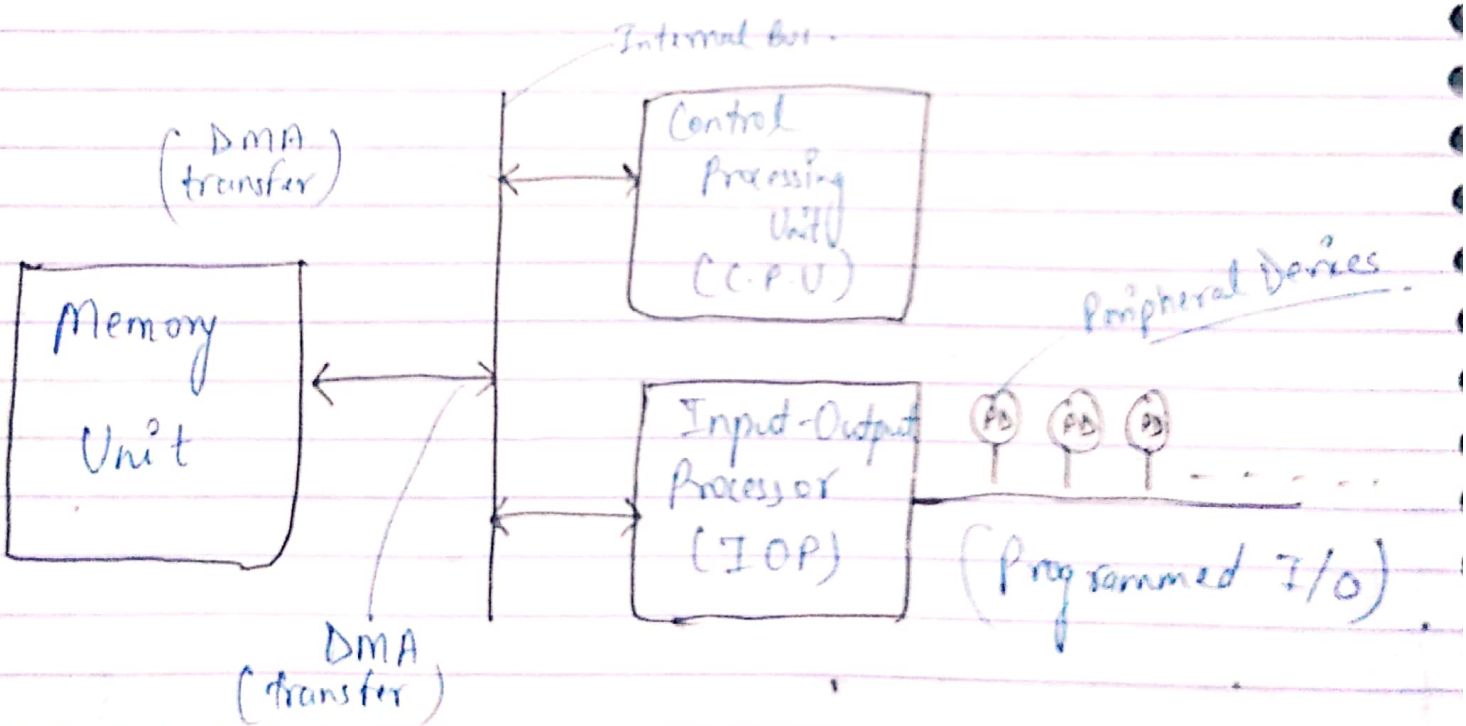
DMA TRANSFER



Block Diagram of DMA Transfer

Note: that - The RD (Read) & WR (Write) lines in the DMA controller are bidirectional. The direction of transfer depends on the status of the BG (Bus Grant) line. When BG = 0 the RD & WR lines allowing the C.P.U. to communicates with the internal DMA registers. When BG = 1 the RD & WR lines are output lines from the DMA controller to the Random Access Memory (RAM) to specify the read or write operation for the data.

IOP (INPUT-OUTPUT PROCESS) -



Block Diagram of a C.P.U with I.O.P.

CPU-IOP Communication.