Unit ⇒ 3 ⇒ Input Output Organisation

1.) Modes of data transfer
   a.) Program Controlled / Program input output
   b.) Intrupt driven
   c.) Direct memory ~~addre~~ access [DMA]
2.) Intrupt structure
3.) Input output interface
4.) Asynchronous data transfer [ADT]
5.) Input Output processor
6.) Data transfering approches and modes.

1.) Modes of data Transfer ⇒

⇒ Binary informations received from an external device is usually stored in memory for later processing. information transfer from the Central Computer into an external device originates in the memory unit. The CPU executes the input output instruction and may accept the data on temphory basis. but the ultimate Source or destination is the memory Unit. data transfer b/w the Central Computer an input output device may be handeld in variety of modes. Some modes use the CPU as an intermediate path other transfer the data to and from memory unit directly. there are three ~~posi~~ possible ways to transfer the data.

i) Program Controlled
ii) Intrupt initiated input /output
iii) Direct memory access [DMA]

i) Program Controlled ⇒ Program input output operation are the result of input output instruction written in the Computer program, each data transfer in initiated by an instruction in the program usually the transfer is to & from the CPU Register and other peripheral Components, other

instruction are needed to transfer the data to & from CPU memory, transfering data under program controlled required constant mone monitering of the peripheral by the CPU. Once data transfer is initiated the CPU is required to moniter the interface. to see when a transfer can again be made. It is upto the program instruction executed in the CPU to keep close. Tabs on everythings that is taking place in the interface unit and the input output device.

i) Programmed input output mode for input data transfer :-

a.) Each input is read after first testing wheter the device is ready with the input [A state reflected by a bit in a status Register.

b.) The program waits for the ready status by Repeated testing the status bit and till all targeted bytes are read from and the input device.

c.) The program is in busy [non waiting], bus state only after the device gets ready in wait state.

ii) Programmed input output mode for output data transfer :-

a.) Each output return after the first testing wheter device is ready to accept the byte at its output register or output buffer is empty.

b.) The program waits for the ready state by repeated testing the status bits end till all the targeted bytes are return to the device.

c.) The program in busy [non waiting] state only after the device gets ready else wait state.

ii) Interupt input output :-

⇒ An interupt input output is a process of data transfer in which an external device or a peripheral informs the CPU that it is ready for communications and Request the attention of the CPU.

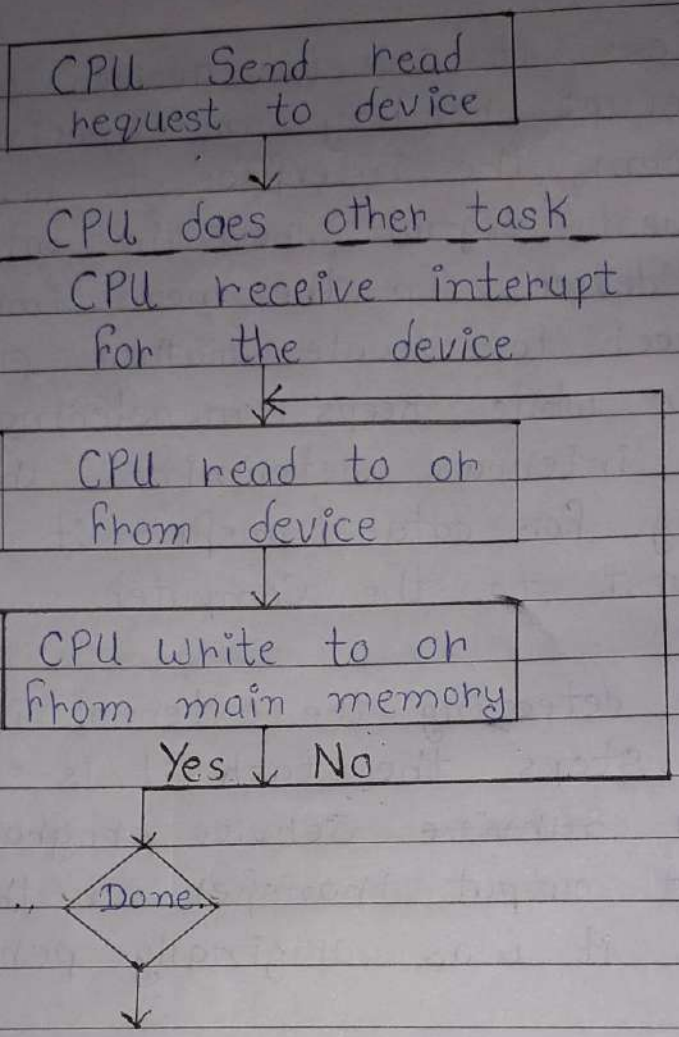
⇒ In the program controlled method the CPU stays in a program loop Until the input output unit indicats that it is ready for data transfer this is a time Consuming

process it can be avoided by using an ~~inter~~ interupt facility and special commands to inform the interface to issue an interupt request signal when the data are available from the device. In the mean time the CPU can proceed to execute another program the interface mean while keeps monetering the device when the interface determines that the device is ready for data transfer it generates an interupt request to the Computer.

⇒ Upon detecting the external interupt signal the CPU stops the task it is processing branches to a ~~survive~~ Service program to process the input output transfer an then return to the task it was originally performing.

⇒ For input the device interupt the CPU when new data has arrived and is ready to be reterived by the System processor the actual action to perform depend on whete the device uses input output pord an ~~m~~ memory mapping.

⇒ For output the device delevers an interupt either when it is ready to accept new data or to ~~acko~~ acknowledge a successful data transfer.



⇒ Advantages ⇒

1.) It is faster than programmed input output mode of data transfer.

2.) It is more ~~effective~~ effecient than program input output of data transfer.

⇒ Disadvantages ⇒

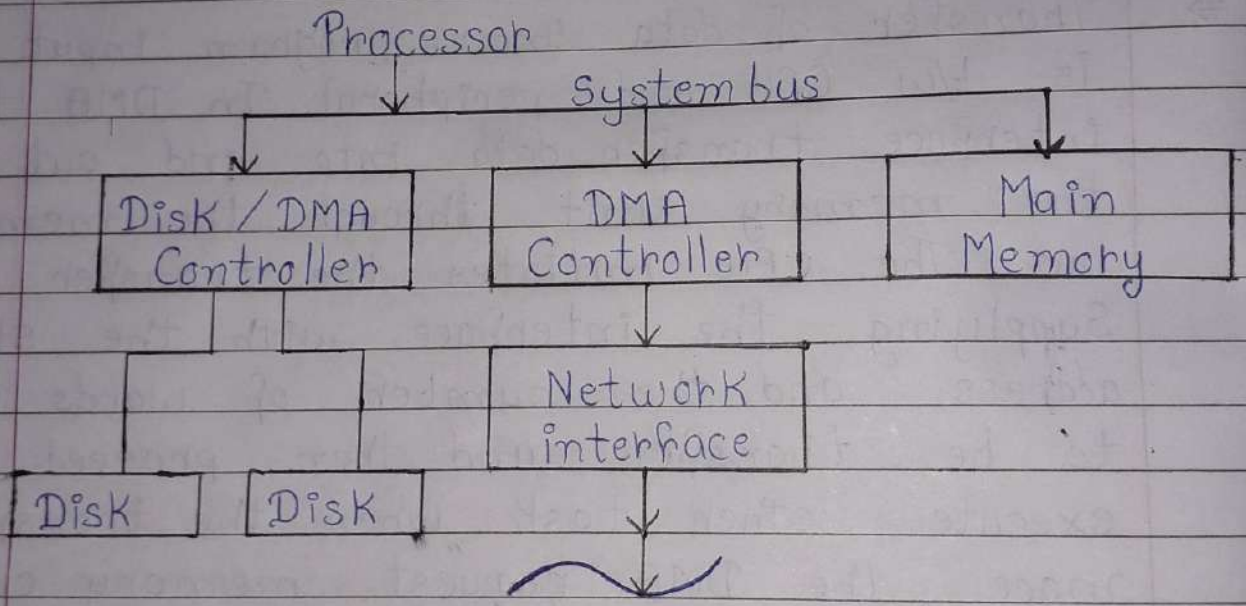
1.) For the low level language it becomes more Complicated.

2.) It can be tough to get various pieces to work together.

iii) Direct Memory access [DMA] ⇒

⇒ Transfer of data under program input output is b/w CPU and peripheral. In DMA the interface transfer data into and out of the memory unit through the memory bus. The CPU initiates the transfer by supplying the interface with the starting address and the number of words needed to be transfer and then proceed to execute other task. When the transfer is made the DMA request memory cycle through the memory bus when the reques is granted by the memory controller the DMA transfer the data directly into memor Input output device are connected to ~~System~~ System bus through a special interface circuit called "DMA Controller".

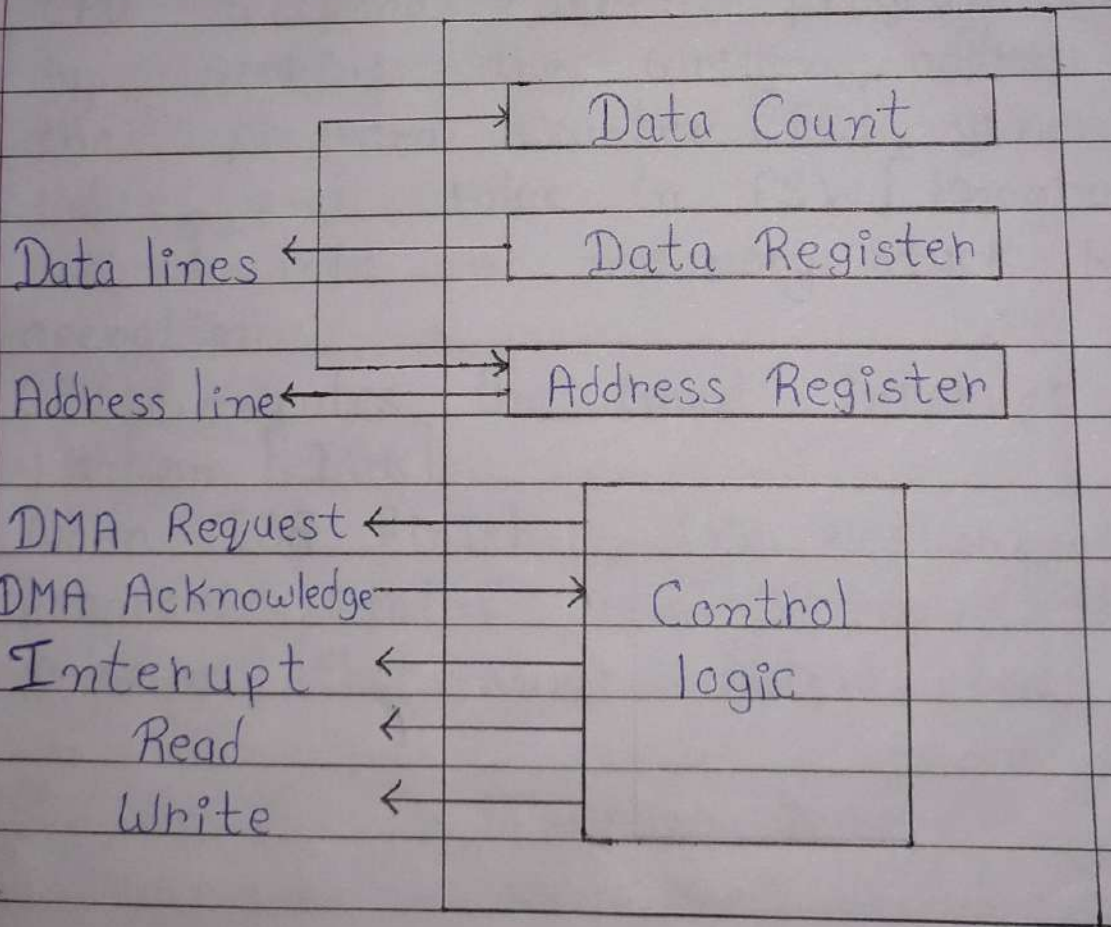
⇒ DMA Controller ⇒ DMA Uses hardware for accessing the memory that hardware is called a DMA Controller. It has the work of transfering the data between input output device and main memory with very less intraction. The DMA Controller is a controller Unit which has the work of transfering data.



⇒ DMA Controller provides an interface b/w bus and the input output device it transfer data without intervention of processor, the processor initiates the DMA Controller by sending the starting address, number of words in the data block and direction of transfer of data from input output device to the memory or from

main memory to input output devices. more than one external device can be connected to the DMA Controller.

⇒ Working of DMA Controller ⇒ DMA Controller Contain an address unit for generating address and selecting input output device for transfer it also contain the Control Unit and data count for keeping counts of number of blocks transferd and indicating the direction of data transfer when the transfer is completed DMA informs the processor by raising an interupt.



leave for

Diagram

2.) Interrupt structure ⇒

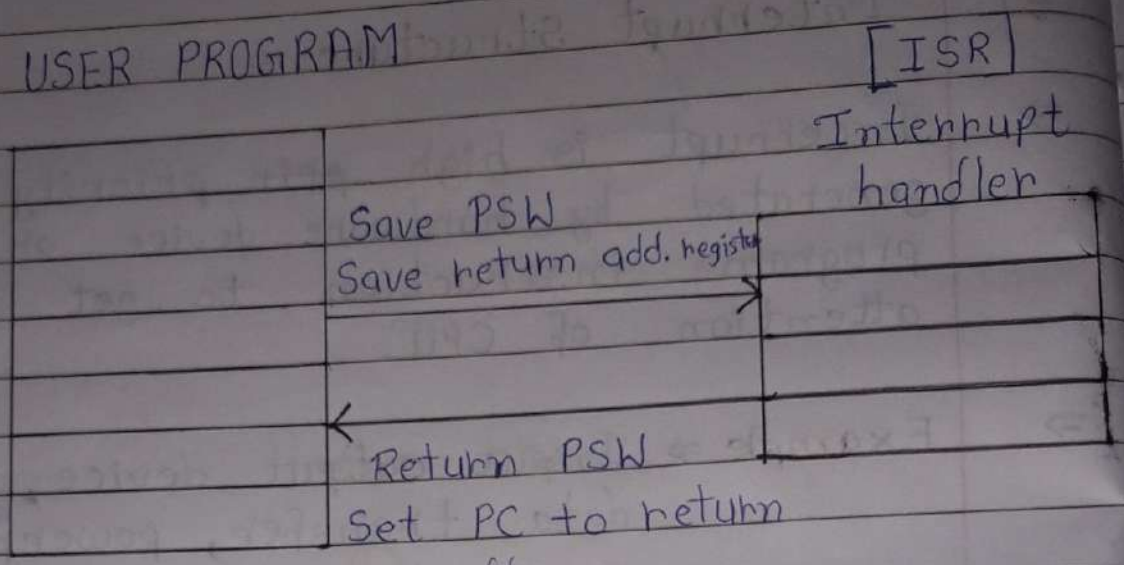
⇒ Interrupt is high ~~prir~~ priority instruction generated by hardware device or software program instruction to get immediate attention of CPU.

⇒ Example ⇒ Input output device, request for data transfer, powerfailure.

Interrupt handling mechanism ⇒

1.) Interrupt is generated by hardware device or software program.
2.) CPU Suspend current program executing by storing the written address from the program Counter [PC], general register value and value in PSW [Program status word] into a memory stack by push operation.
3.) CPU executes the ~~input~~ interupt Service routine [ISR].
4.) When ISR finished, CPU restors the previo program status and Continue the original program that was interrupted.

Diagram is on next Page.



Types of interrupt ⇒

1.) Hardware interrupt ⇒ If the interupt signal for the processor generated from the external device or hardware device are called hardware interupt

Ex ⇒ We press the key with keyboards, keyboards will generated a signal which is given to the processor to do action Such interupts are called hardware interupt.

⇒ Hardware interupt can be classified into two types.

i) Maskable interrupt ⇒ The hardware interupt that can be delayed

When a highest priority interrupt has occur to the processor are called Maskable interrupt.

ii) Non-Maskable interrupt ⇒ The hardware interrupt that can not be delayed an immediatly serviced by the processor are called non-maskable interrupt

2.) Software interrupt ⇒ The interrupt Signal generated from the internal device or software program are called Software interrupt.

⇒ Software interrupt divide into two types.-

i) Normal interrupt ⇒ The interrupts that are Caused by Software instruction are called normal interrupt.

ii) Exception interrupt ⇒ Exception is nothing but 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.

3.) Vectored & Non-Vectored interrupt ⇒

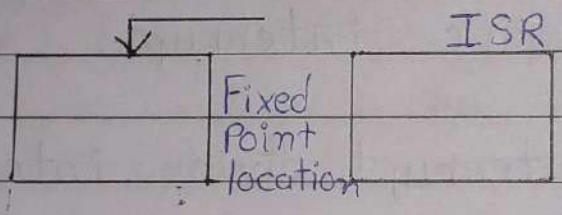
⇒ The way the processor [CPU] chooses the branch address of interrupt service routine [ISR].

⇒ The processor uses two method for it.

i) Non vectored interrupt ⇒

⇒ The address of ISR is assigned to a fixed location in memory.

⇒ Device does not provide the address of ISR.



ii) Vectored interrupt ⇒

⇒ The Source (device) that interrupted supplies the address information of ISR to the CPU. This information is called interrupt vector.

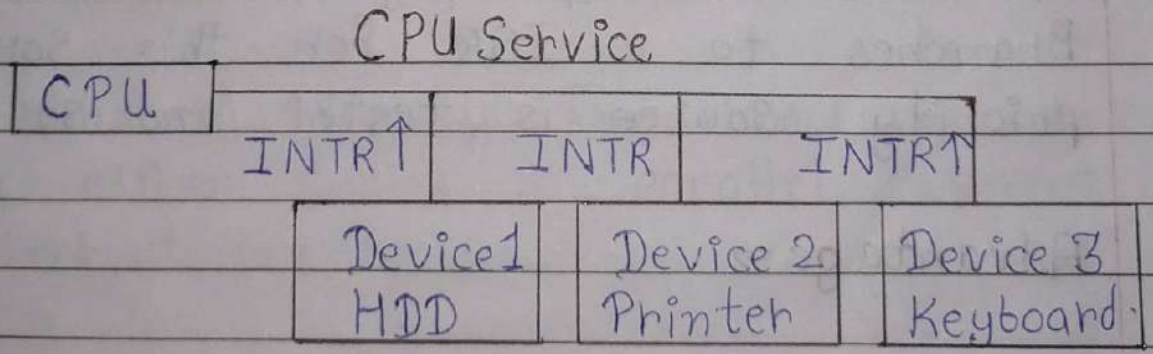
⇒ It is common when CPU has multiple peripheral connected to the System bus.

4) Priority Interrupt ⇒

⇒ There are number of Input output device attached to the computer They all are capable of generating the interrupt. When the interrupt is generated from more than one device simultaneously priority interrupt system is used to determine which device is to be serviced first.

⇒ A priority interrupt is a system that established a priority over the various source to determine which condition is to be serviced first when two or more request arrive simultaneously.

⇒ Device with high speed transfers are given higher priority. for example Hardisk, slow device are given low priority for example Keyboard.



⇒ Establishing priority of Simultaneous Interrupt ⇒

i) Using Software Method [Polling] ⇒

⇒ Polling is software method of establishing priority of simultaneous in interrupt it branches to an common ISR whose job is to poll each input output device to determine which device caused the interrupt.

⇒ In this method there is one common branch address for all interrupt.

⇒ Branch address contain program code that polls the interrupt sources in sequence i.e check each source whether whether they have generated the interrupt.

⇒ Highest priority source is tested first if its interrupt signal is ON control Branches to a ISR for this source priority source is tested and so on.

⇒ Advantage ⇒

a.) Flexible since established by Software.

b.) low cost since it needs a very low hardware

⇒ Disadvantage ⇒

a.) Very slow.
b.) Time overhead to handle many interrupt can be excessive.

ii) Priority interrupt by hardware ⇒

⇒ Require a priority interrupt manager which accepts all the interrupt require to determine the highest priority request fast. Since identification of highest priority interrupt request is identified by hardware

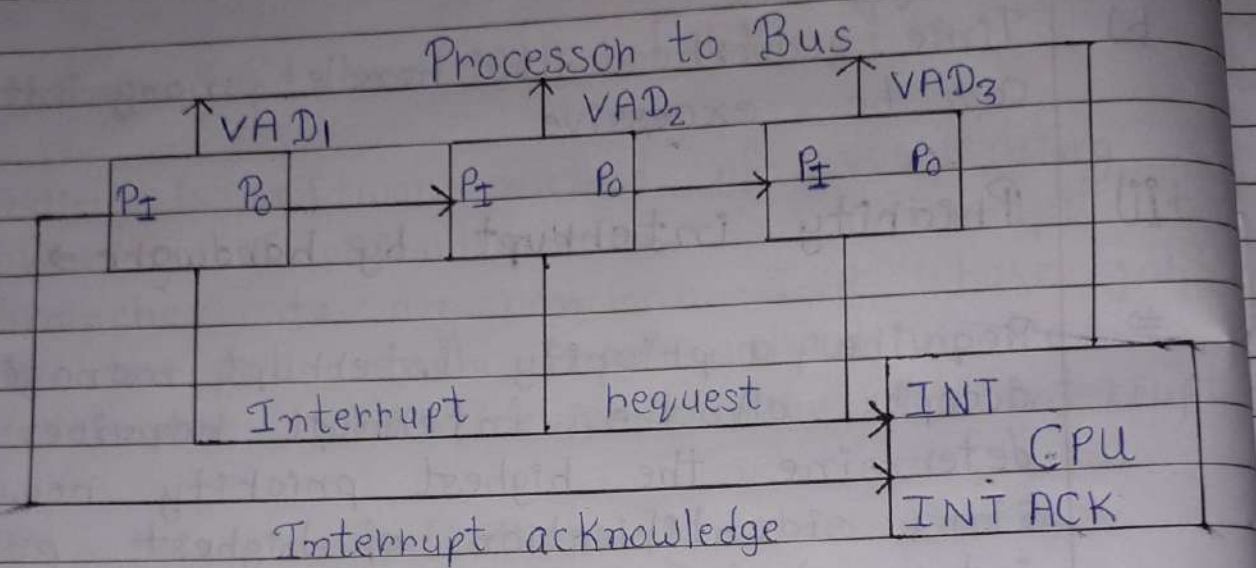
⇒ To speed up the operation, each interrupt source has its own interrupt vector to access its own service routine directly.

⇒ This no polling is required because all decisions are established by the hardware priority interrupt unit.

⇒ Hardware priority function can be establish by either serial or parallel connect of interrupt lines.

⇒ There are two types ⇒

⇒ Daisy channing Priority [serial priority] ⇒



⇒ If the device requested the interrupt, it place VAD (Vector address) on the bus and it blocks the ~~signed by~~ signal by placing 0 in Po (Priority out)

$P_I = 1$ & $P_I = 0$

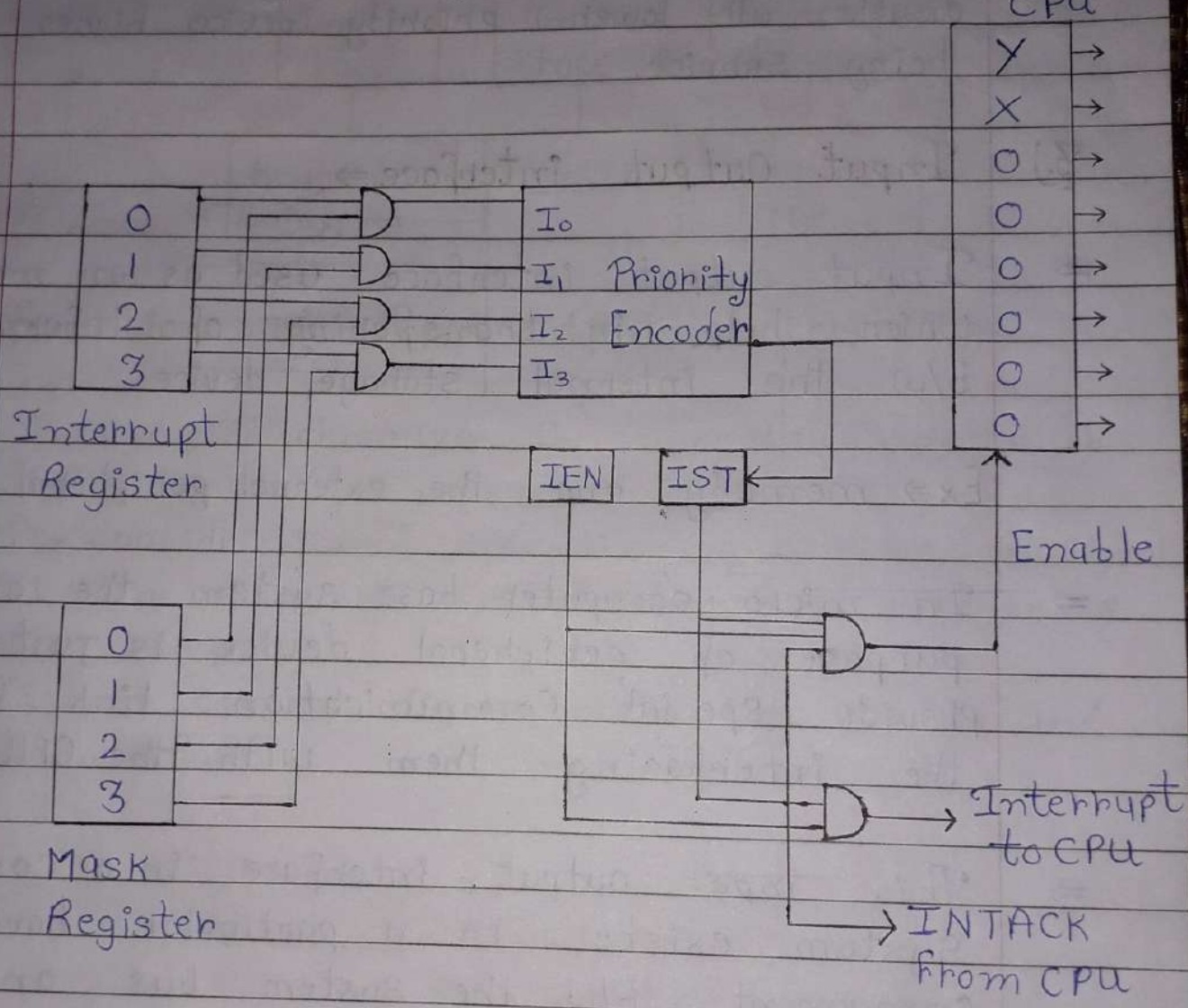
⇒ If device had no requested interrupt it pass the acknowledge signal INTACK

⇒ This process continous until appropriate is found.

⇒ If device gets 0 in its PI input it generates 0 at the PO output tells other device that

acKnowledge Signal [INTACK] has been blocked.

⇒ Parallel priority interrupt ⇒



⇒ It Consist of interrupt register whose bits are set separately by two interrupting device.

⇒ Priority is established according to the position of the bits in the register.

⇒ Mask register is used to provide facility for highest priority device to interrupt when lower device is being serviced or disable all lower priority device highes being service.

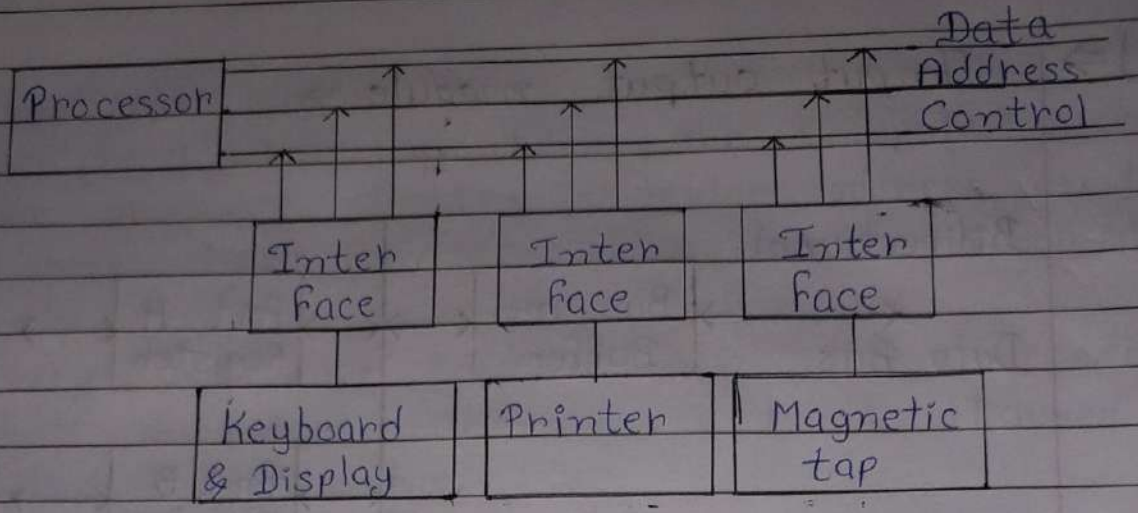
3) Input Output interface ⇒

⇒ Input output interface used as an method which help in transfering of information b/w the internal storage device.

Ex⇒ memory and the external peripheral device.

⇒ In micro computer base system the only purpose of peripheral device is just to provide special Communication link for the interfacing them with the CPU

⇒ This input output interface in a computer System exists in a particular hardware component b/w the system bus and peripheral. This Component is known as the "Interface Unit".

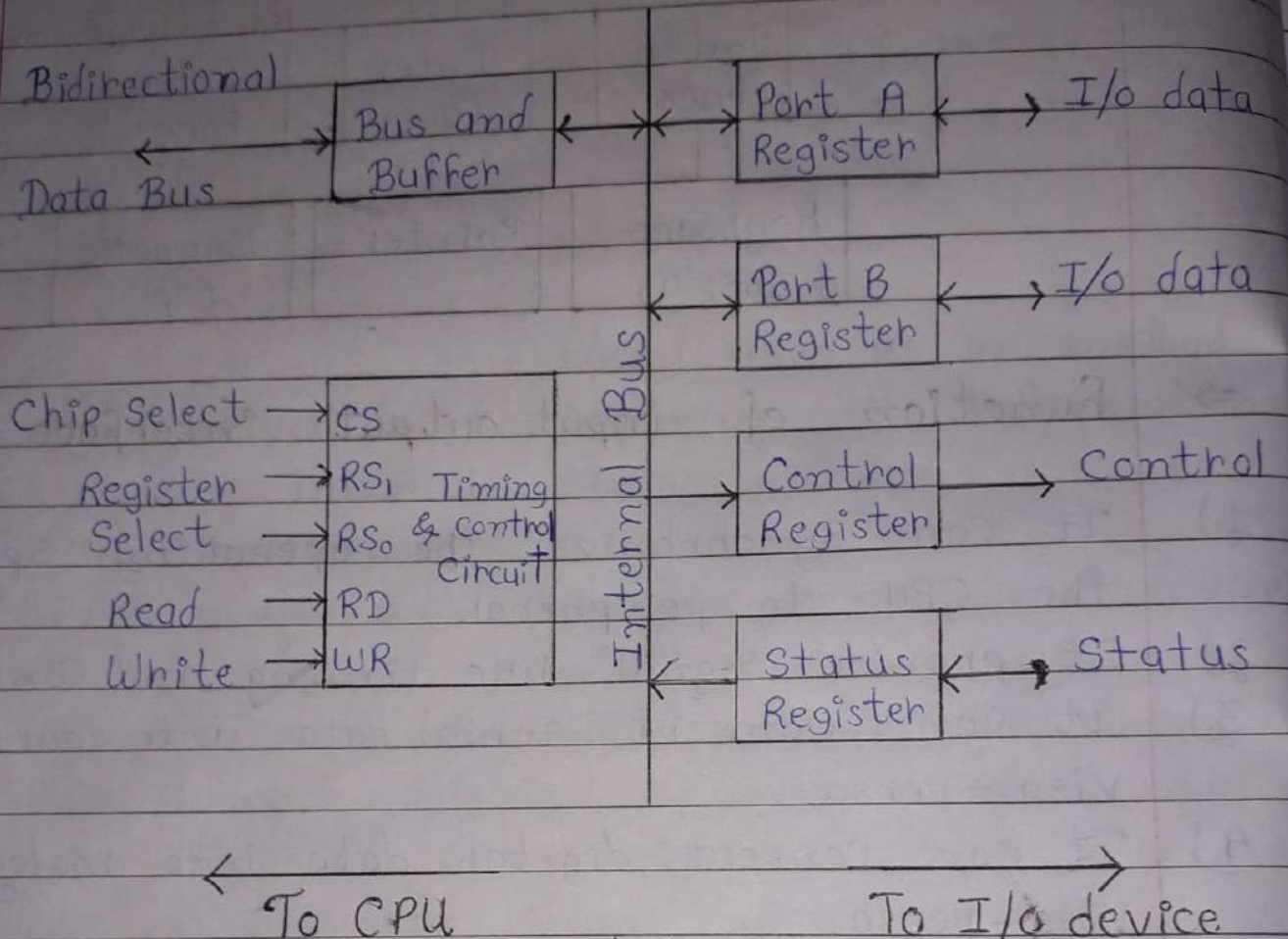


⇒ function of input output interface ⇒

1.) It can synchronize the operating speed of the CPU to peripheral.
2.) It provide signal like timing and control signal.
3.) It can convert serial data into parallel & visa-versa.
4.) It can convert digital data into analog and visa-versa.

Diagram is on next Page

4.) ⇒ Input output module ⇒



CS	RS1	RS0	Register Selected
0	X	X	None - data duo in High Impedance
1	0	0	Port A register
1	0	1	Port B register
1	1	0	Control register
1	1	1	Status register

⇒ The input output buses are linked to all the peripheral interface from the Computer processor. The processor locates a device address on the address line for interaction with a specific device.

⇒ Every interface contains an address decoder that monitors the address lines attached to Input output bus.

5.) Asynchronous Data transfer [ADT] ⇒

⇒ The internal operations in an individual unit of a digital System are Synchronized using clock pulse.

⇒ If the registers in the input output interface share a common clock with CPU ~~registe~~ register then the transfer b/w the two unit is said to be Synchronous but in the most cases the internal timing in each unit is independent of each other, So each uses its private clock ~~after~~ for its internal registers. In this case two units are called asynchronous to each other and if data transfer occurs b/w them, this data transfer • called "asynchronous data transfer."

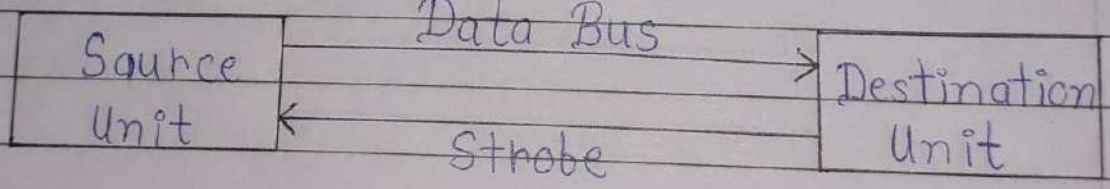
⇒ There are two methods can achieve this asynchronous data transfer.

i) Strobe Control method ⇒

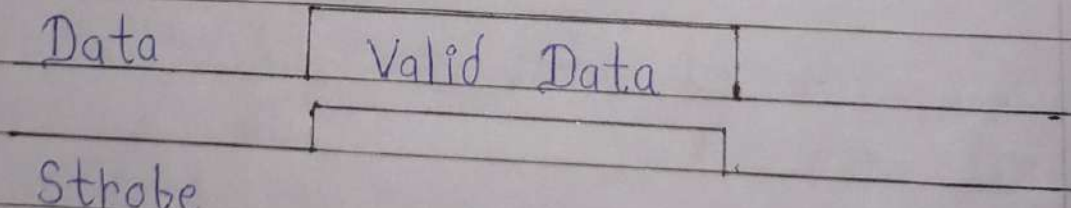
⇒ The strobe control method of asynchronous data transfer employs a single control line to time each transfer. This control line is also known as a strobe and it may be achieved either by source or destination depending on which initiate the transfer.

a.) Source initiated Strobe ⇒ In the source initiated strobe the strobe is initiated by source. The source unit first places the data on the data bus.

Block Diagram



Timing Diagram



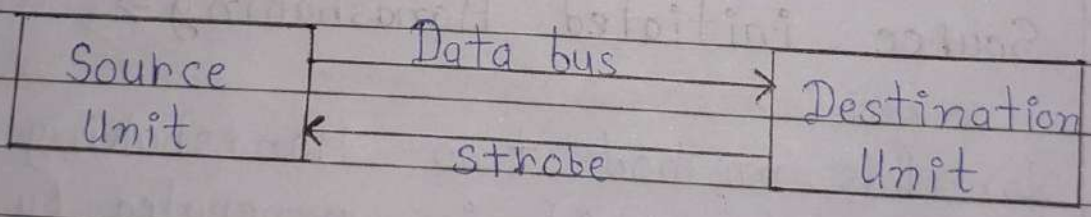
⇒ After a brief delay to ensure that the data resolve to a stable value the source activate a strobe pulse. The information on the data bus and strobe control signal remains in the active state for a sufficient time to allow the destination unit to receive the data.

⇒ The destination unit uses a falling edge of strobe control to transfer the contents of a data bus to one of its internal register. The source removes the data from the data bus after it disable its strobe pulse, Thus new valid data will be available only after the strobe is enabled again.

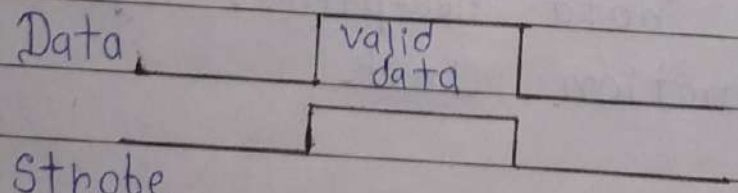
b.) Destination initiated strobe ⇒

⇒ In this method strobe is initiated by destination the destination unit first activates the strobe pulse, informing the source to provide the data.

Block Diagram



Timing Diagram



⇒ The source unit responds by placing the requested binary information on the data bus. The data must be valid and remain on the bus long enough for the destination unit to accept it.

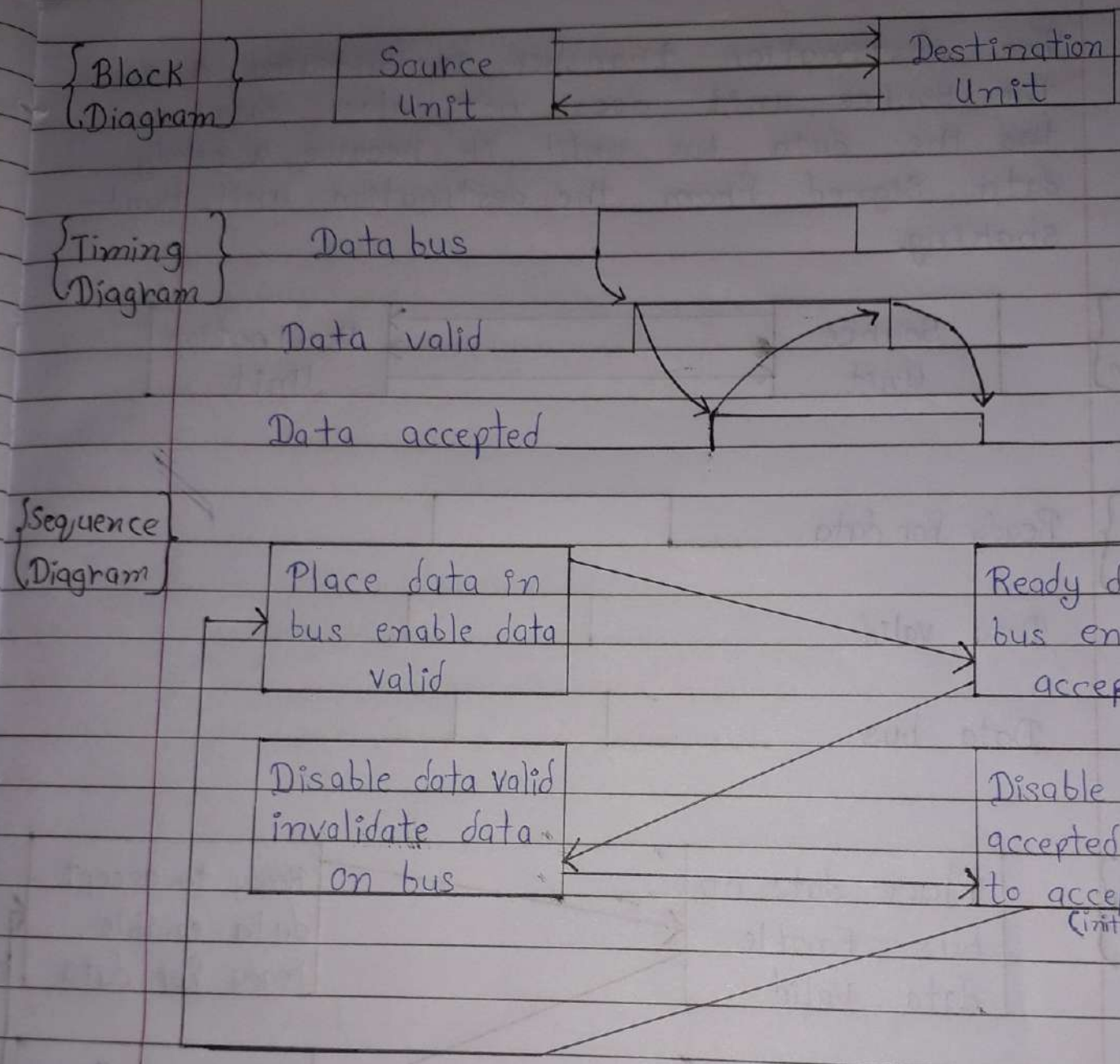
ii) Handshaking method ⇒

⇒ In handshaking method introduce a second control signal line that replays the Unit that initiates the transfer.

⇒ In this method one control line is in the same direction as the data flow in the bus from the source to the destination. The source unit uses it to inform the destination unit wheter these are called data in the bus. It means the sequence of control depends on whether the transfer is initiated by source and destination.

a.) Source initiated Handshaking ⇒

⇒ In this method two Handshaking lines are data valid which is generated by the source unit and "data accepted". generated by the destination Unit.

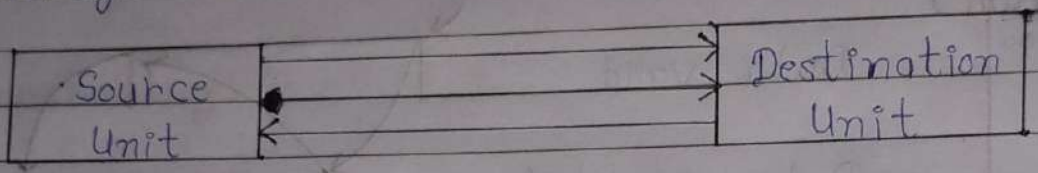


b.) Destination initiated handshaking ⇒

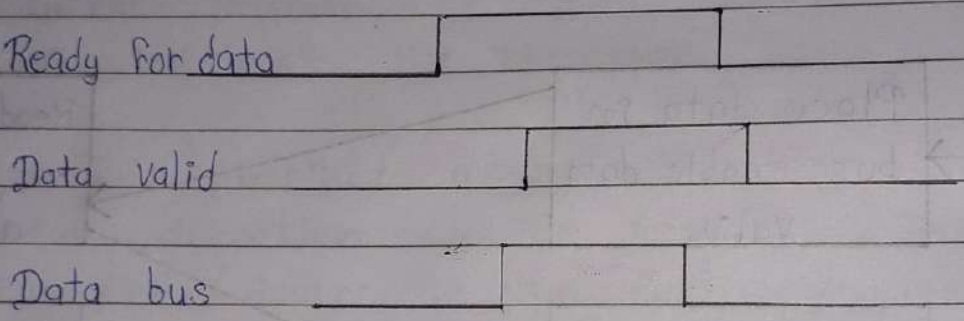
⇒ In this method two handshaking line are "data valid" generated by the Source Unit and "ready for data" generated by the destination Unit.

⇒ The destination transfer is initiated & so the source unit does not place data on the ~~the~~ the data bus until it receive a ready data signed from the destination unit. Hand-shaking.

{Block Diagram}



{Timing Diagram}



{Sequence Diagram}