UNIT-4

11/1/19

Input/Output Organization.

→ Accessing I/o devices, Interrupts Hardware, enabling and disabling Interrupts, Handling Multiple devices, Direct memory acess (DMA),

Buses:- Synchronous Bus, Asynchronous Bus, interface circuit, Standard Input/output Interface: peripheral component Interconnect (PCI) BUS, universal serial BUS (USB)

overview:-

12/1/19 → I/o devices are used to exchange data.

→ other modern i/o devices are sensor's, other than monitors, keyboards etc..,

→ In automater running car when a signal of redcolour is seen in the traffic the sensor gets the red signal as input to the computer system (i.e; input to processor) output is car is getting stopped.

conclusion:

→ Sensor instruction is another kind of Input instruction format

Accessing I/o devices:-

→ using a single-bus arrangement we can acess I/o devices

→ This bus enabels all the devices connected to it to exchange information.

→ This bus contains 3 types of lines:-

(i) Address lines

(ii) Data lines

(iii) Control lines

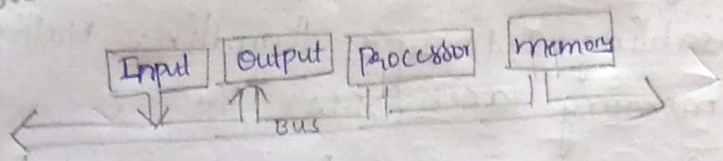
→ Each I/o device is assigned with unique address inside memory.

→ If memory and processor needs to send output informa

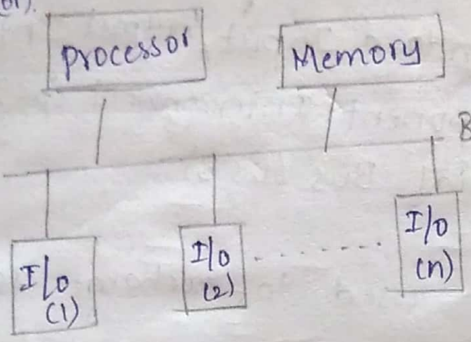
-tion. Intially the information is sent to output devices with it's address.

25/1/19

Single-Bus structure:



(or).



→ I/o devices interaction with system (i.e, processor, memory

Memory mapped I/o :-

→ when you type a character, processor knows which output Input address is vacant.

→ In memory there is always a seperate space for I/o devices. (Input-out memory space)

eg:- Move DATAIN, R0
Move R0, DATAOUT

* How I/o devices interact with bus?

→ I/o devices interact with bus using I/o interface.

→ I/o interface is collection of 3 hardware components:-

(i) Address Decoder
(ii) control circuits
(iii) Data & status Registers

(i) → Every I/o device is assigned with unique address

→ Address decoder is used to identify corresponding I/o device.

→ Address decoder enables the device to recognise it's address when this address appears of Address line.

(iii) Data Register contains data which has to exchange to (or) from processor.

i.e, Data Register (DATA IN, DATA OUT)

In STATUS Register means the third bit in an 8 bit register. that 3rd bit register is SIN flag if SIN is 0 processor has to wait. if SIN is 1 processor move. (Refer Branch Instructions)

(ii) control circuits:-

→ Gives Idea about synchronisation, program controll -ed I/O.

Input example

Problem:- Until SIN = 1 processor has to be in wait condition (or) For every unit of clk cycle processor looks up into SIN (or) polls with SIN until SIN = 1

→ This is time waste for processor

→ In this waiting time processor can execute another Instruction. so that system performance can increase.

another name interrupt

Solution:- When data is ready send a signal from control circuit to the processor. then processor reads data from data in register. so this is called Synchronisation. This also called Program controlled I/O

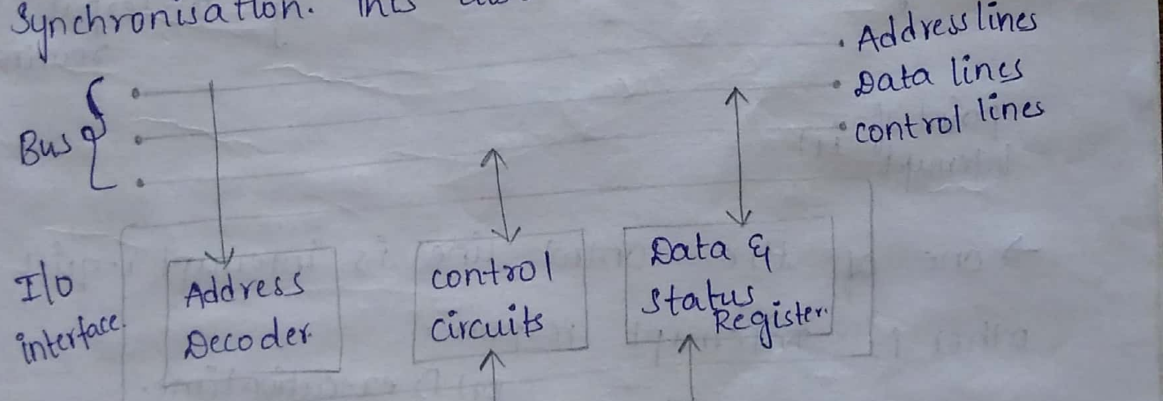


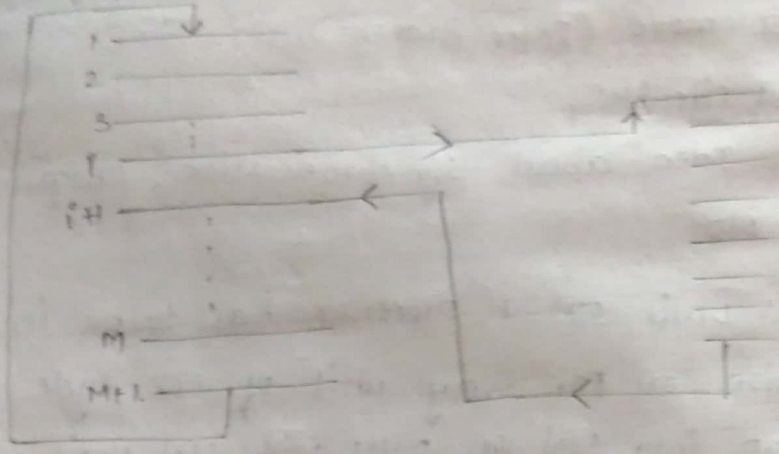
Fig: I/O interface for an input device.

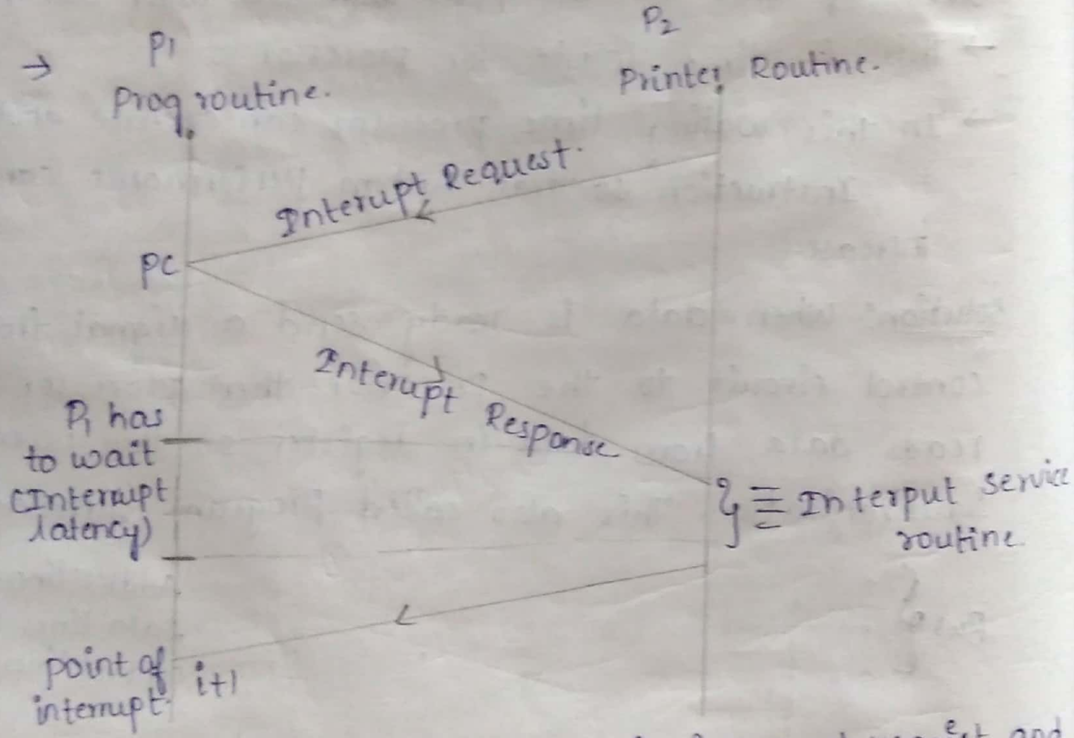
i/output example
7/2/19 eg:- Step 1:- program ~~1~~ P_1 (program routine)/computer routine.

Step 2:- printer software (or) print routine. this is P_2

P_1 (computer routine) P_2 (printer routine)

computer program





→ one of the control line is interrupt requist and other is Interrupt response.

→ P_1 is the waiting time, i.e, Interrupt latency. (or) P_2 execution time.

→ processor is busy with P_1. suddenly interrupt reque has come from printer P_2 with highest priority

→ If processor is at middle of i statement it comp -te's that 'i' and then it goes to P_2 (interrup service routine / printer software) by storing PC value (i+1) & variables in processor registers (or) stack memory organisation.

→ After executing ISR processor returns back to (i+1) i.e., point of Interrupt.

→ Interrupt latency should be very very less.

* Another application

OS executes all it's responsiblites using signals

eg: real time applications railway signal, traffic control signal.

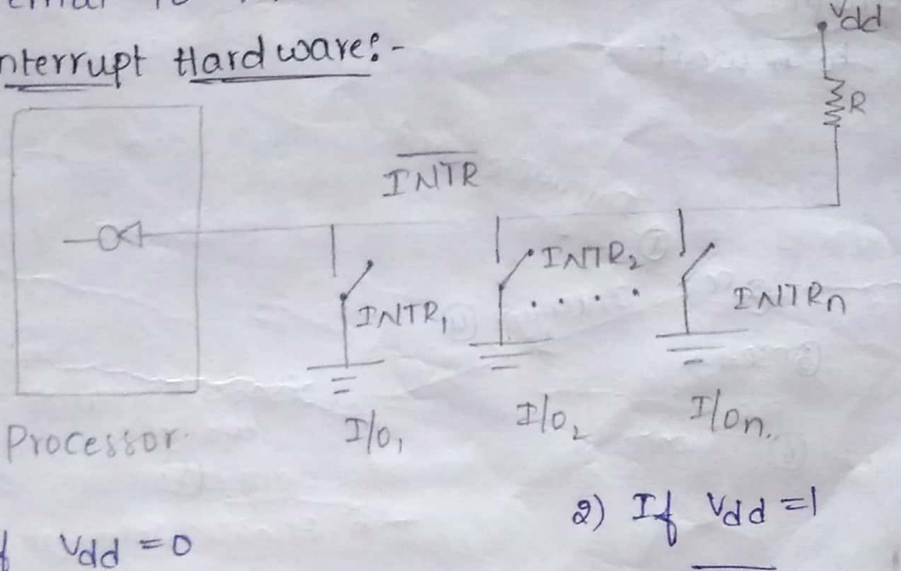
Interrupt:-

→ An ~~iterr~~ interrupt is a signal which stops current execution of the.

(or)

Interrupts enable transfer of control from one Program to another to be intiated by an event external to the computer.

8/2/19 Interrupt Hardware:-



1) If Vdd = 0

INTR = 1

Closed circuit - on.

2) If Vdd = 1

$\overline{INTR}$ = 0

open circuit - off.

3) All I/o devices associated with switches.

4) To generate interrupt for I/o 1

$\overline{INTR}$ = 1, Vdd = 0

INTR₁ =1 (close switch)

(5) All the switches should be grounded.

(6) INTR = INTR1 + INTR2 + - - - - + INTRN.

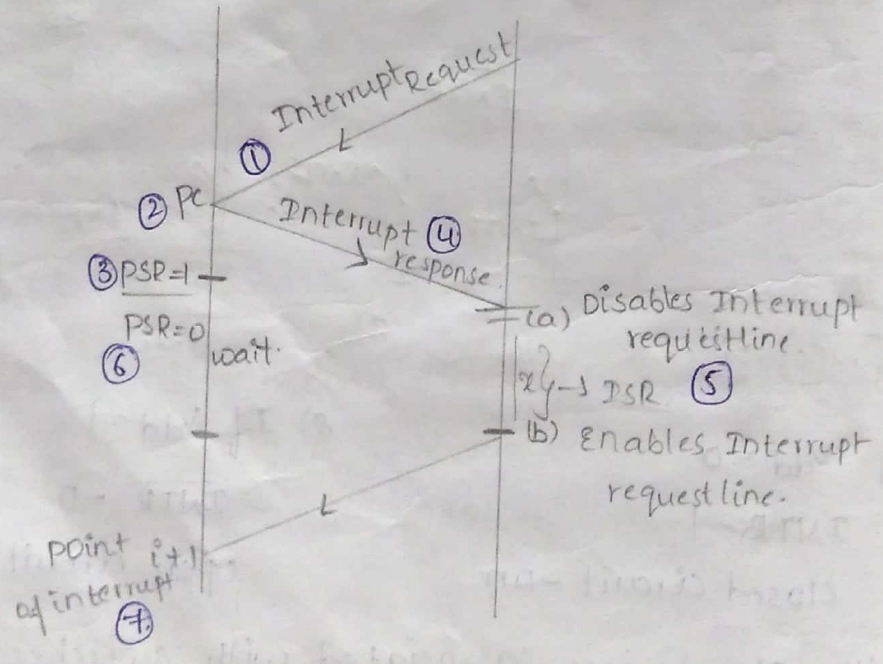
1 = 1 to

Enabling and Disabling Interrupts:-

→ ① The first statement in the ISR is that it dis-ables the Interrupt request. The last statement in ISR it enables the Interrupt request

② If processor register is zero, (enable) Every processor. maintains process status Register if PSR =1 it indicates that the processor is busy with processing instructions (disable).

③ If interrupt hardware is designed with edge trigged circuit it can handle only one interrupt at a time such that other I/o devices has to ~~r~~ wait.



Summary:-

Step1:- The device raises an interrupt request

Step 2:- The processor ~~interacts~~ stops the program beg in executed.

(waiting state).

stept 3:- Interrupts (Interrupt request line) are disabled by changing the control bits in the psR (except in the case of edge triggerd Interrupts)

④ The device is informed that it's request has been recognised. and in response. it deactives the interrupt requset signal.

⑤ The action requested by Interrupt is performed by ISR

⑥ Interrupts are enabled and exectution of the interrupted program is resumed from point of Interrupt.

1/2/19 **Handling Multiple devices:-** refer text book

→ when multiple interrupts occurs. the problems that arise is polling, the solution for this is vectored ~~Interpolation~~ In~~puts~~terrupts. , processor priority arbitary circuit.

→ when multiple interrupts in a single unit of clock cycle simultaneously this is Daisy chain, problem.

→ In this all the interrupts, (nested Interrupts) having same priority. are grouped and on this shortest Job & Round robbin, (time slice) FCFS algorithms are performed.

→ In Daisy chain problem which interrupt (device) which is closer to processor. (i.e; electrically closest) ~~th~~ will execute first..

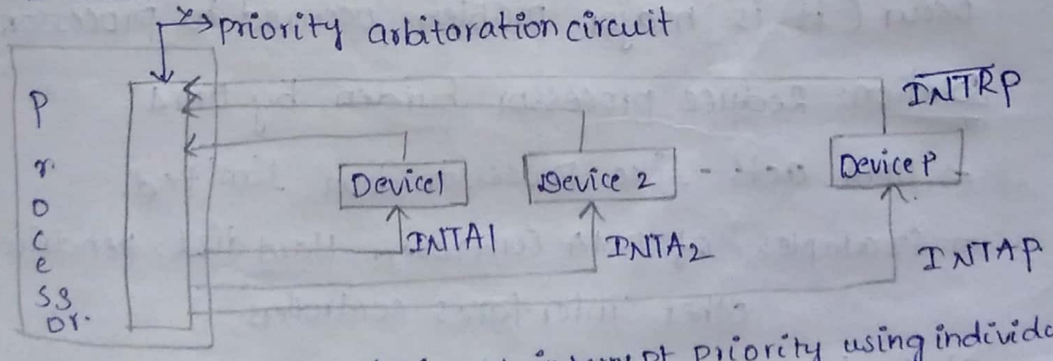
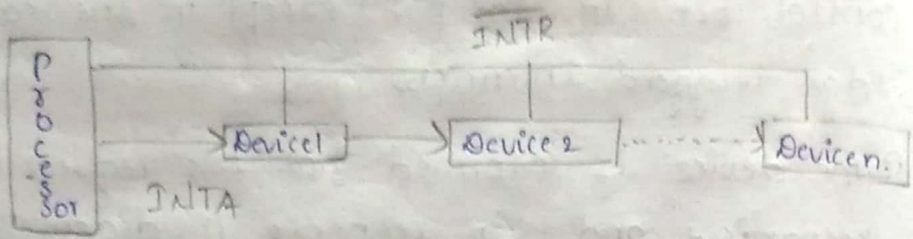
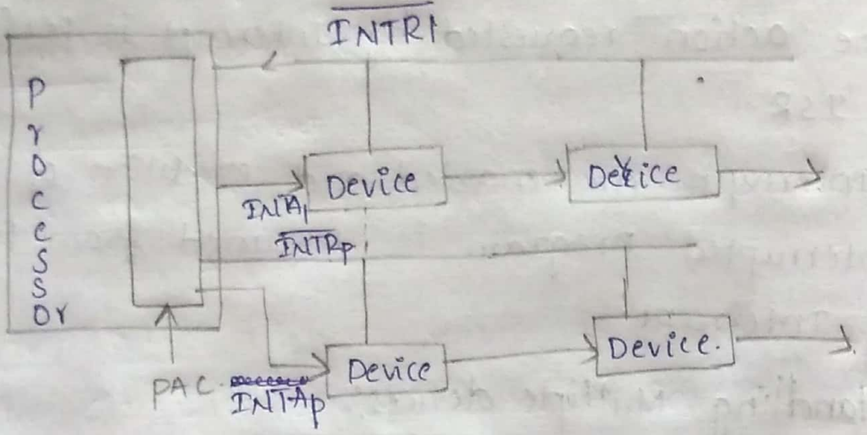


Fig: Implementation of interrupt priority using individual interrupt request and acknowledgement line.

→ one of the cntrl line must be acknowledgement line.



(a) Daisy chain



(b) Addressing priority group.

14/2/19 Direct Memory Access / DMA controller.

i) suppose, the inputs to the processor is a huge. amount of data.

(ii) Eg:- we want to see a movie from hard disk., how will it run (700 MB) Huge blocks of load.

⇒ 700 mb should be input to processor then it is mapped to memory.

⇒ Generally processor is busy with several inputs

⇒ processor feels like burden & some time dead lock. occur (it is major problem occured to processor)

Solution: Reduce processor burden by DMA

⇒ DMA acts like processor but limited.

⇒ Examples of DMA controller: Hard dist, pendrive some other interfaces controllers.

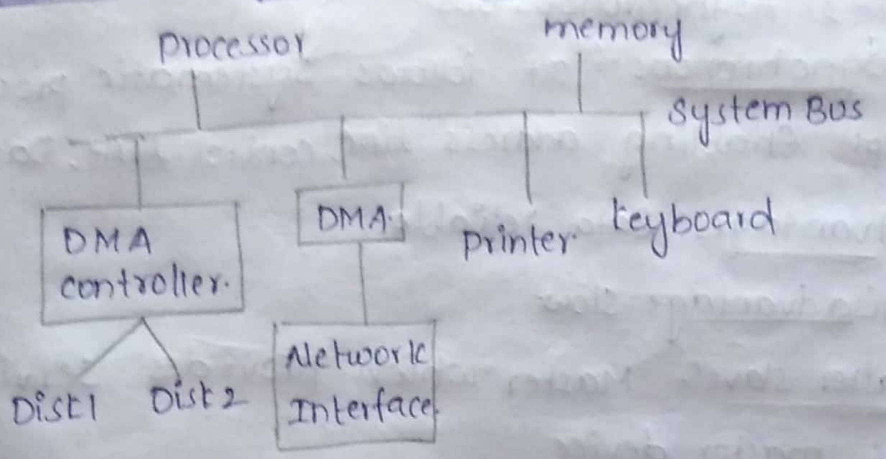
⇒ keyboards, printers donot use DMA.

⇒ DMA is hardware.

⇒ DMA will run in DMA Block mode (or) Burst mode.

(DMA) enables data (huge blocks of data) from harddisk to main memory)

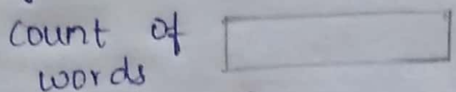
Implementation of DMA controllers in System.

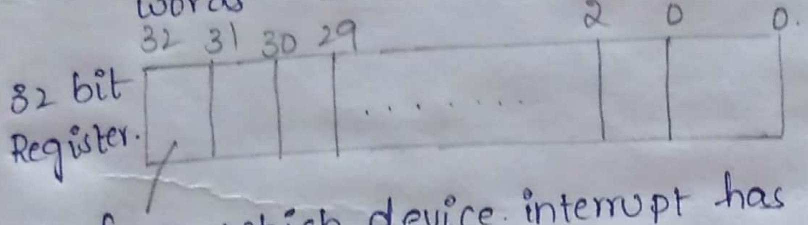


Hardware components of DMAC -

⇒ Register which contains starting address of word where memory is vacant.

⇒ Register which contains count of words or Blocks





from which device interrupt has been occurred.

31 - 32 ⇒ interrupt request.

29 - 30 ⇒ interrupt is enable or not

1 - 2 ⇒ Read/write.

0 -1 ⇒ work is done or not.

⇒ Before handling any interrupt, DMA has to take permission from processor

16/2/19 Buses:- These are used to transfer data b/w hardw-are components.

⇒ there are 2y ways of transfers

1) Synchronous Bus

2) Asynchronous Bus

⇒ Synchronous:- It follows systematic procedure.

step1) Enabling address line, control line, Data line.

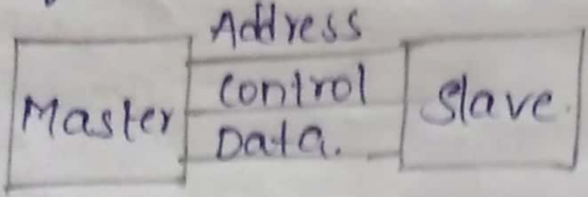
Advantages: Data reliable

disadvantage - slow

Master-slave:- Master which provides service is called as master device.

slave - which utilises service is called as slave devic

→ The classification of Master or slave depends on mode of communication.



25/2/19 Asynchronous Bus:-

→ In an Asynchronous bus address line, control line, data line can be enabled (or) disabled in any sequence in any unit of clock cycle.

→ Master & slave donot work with respect to address control & Data lines.

1st master will on next slave will on next data get transfa

→ They only work with a small Hand shake procedure.

→ The data get's transferred even control line is off.

→ Asynchronous bus will not ge preference to Bus clock

→ Modern processors follow Asynchronous bus data transfer because Asynchronous is fast.

→ synchronous bus is slow compared to Asynchronous bus & It is more reliable.

→ Asynchronous bus disadvantage is less relaibility.

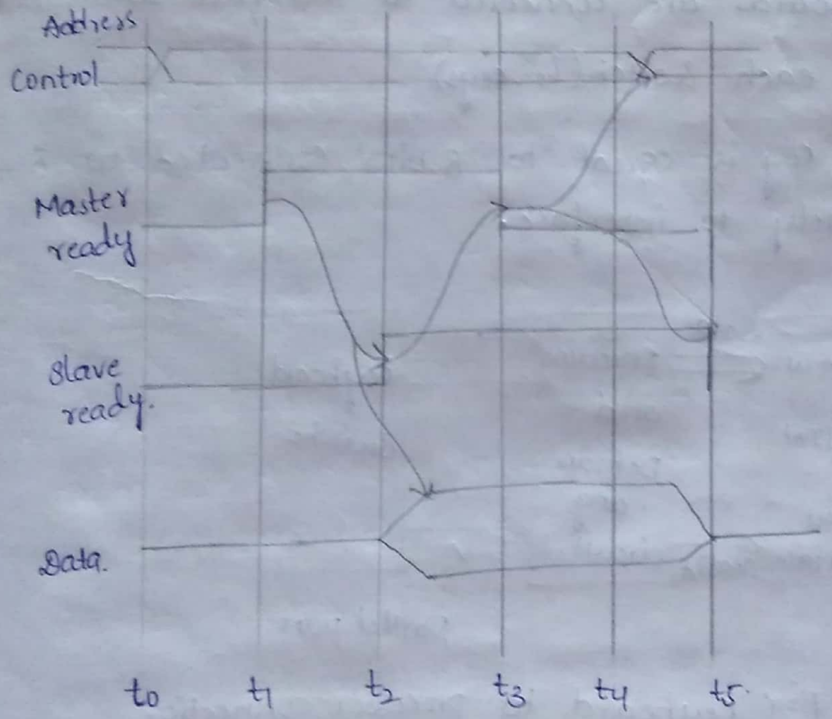


Fig:- Hand shake control of data transfer during an input operation.

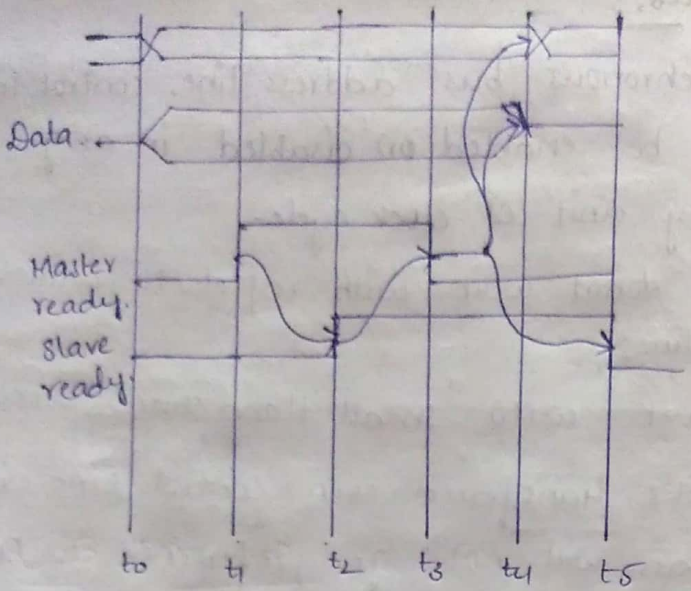


Fig:- Handshake control of data transfer during output operation.

Interface circuit:-

→ keyboard - processor Interface circuit

→ keyboards are connected to interface using Parllel p

(for each switch (i.e., keys))

→ one key is equal to 8 bits connected to 8 wires parllely to interface.

27/2/19

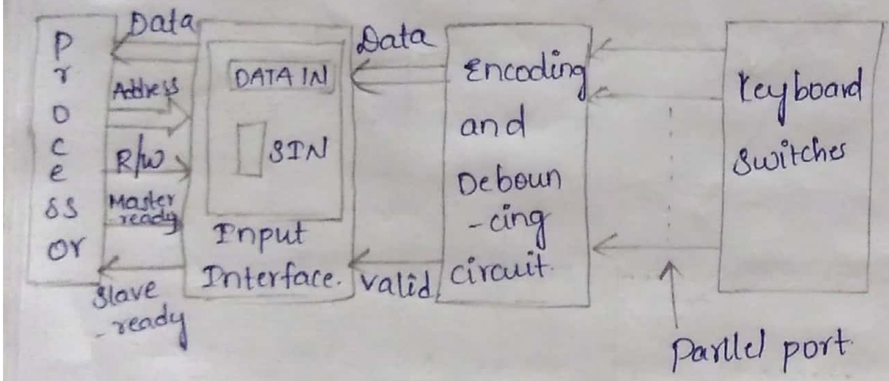


Fig:- keyboard to processor connection.

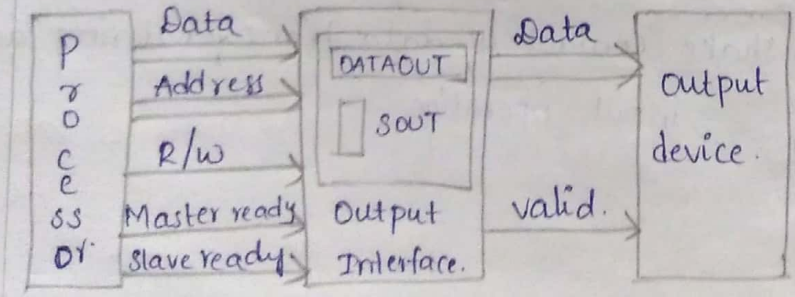


Fig:- Printer to Processor connection.

28/2/17 Standard Input/output Interface:-

-> To transfer data we need buses.

(i) PCI Bus (pheripheral component Internal)

(ii) SCSI Bus (Small computer System Interface)

(iii) USB (universal Serial Bus)

iv) ISA (Industry standard Architecture)

-> Based on the required speed of computer the. above bus Interfaces can be included inside. System.

-> All the combinations of Interfaces can be included inside computers.

(i) PCI Bus:-

PCI Bus can be seen in mother Board.

-> In mother Board we can see a Bridge called PCI Bridge.

-> processor, memory & Input/output devices are communicated using PCI bus and it's PCI bridge

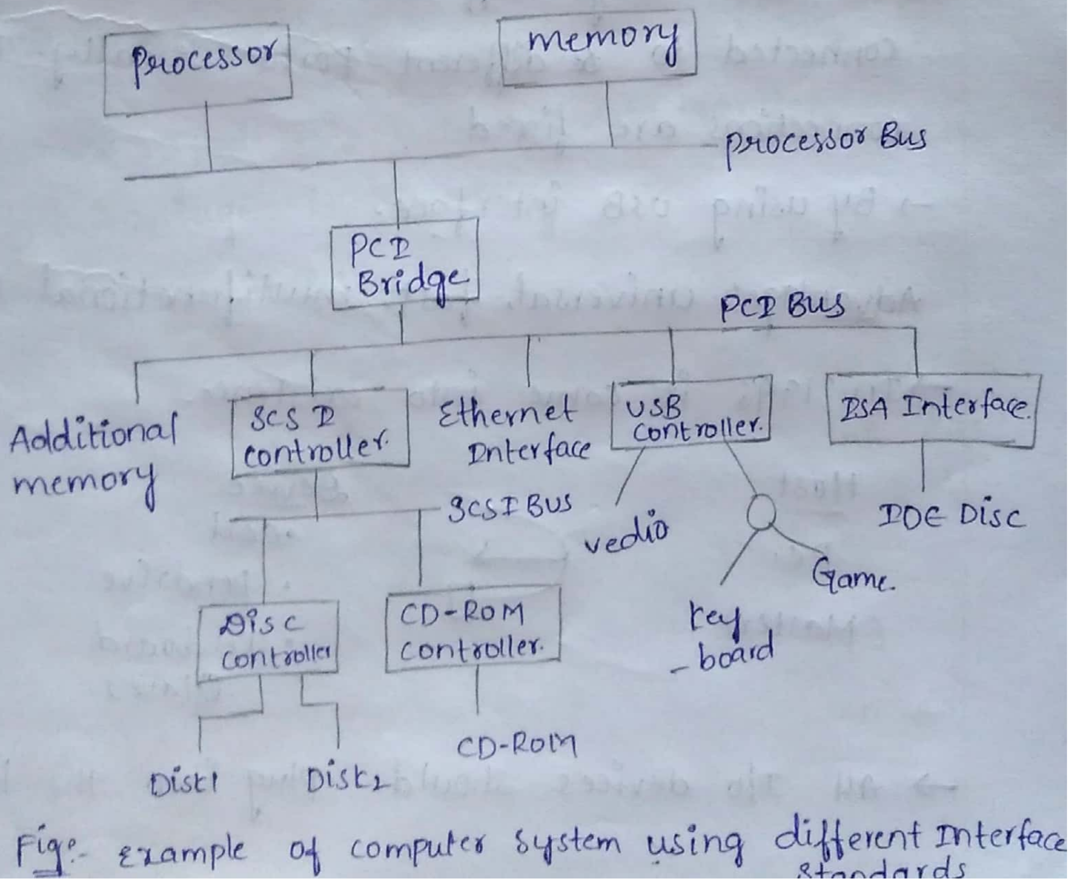


Fig:- example of computer system using different Interface standards

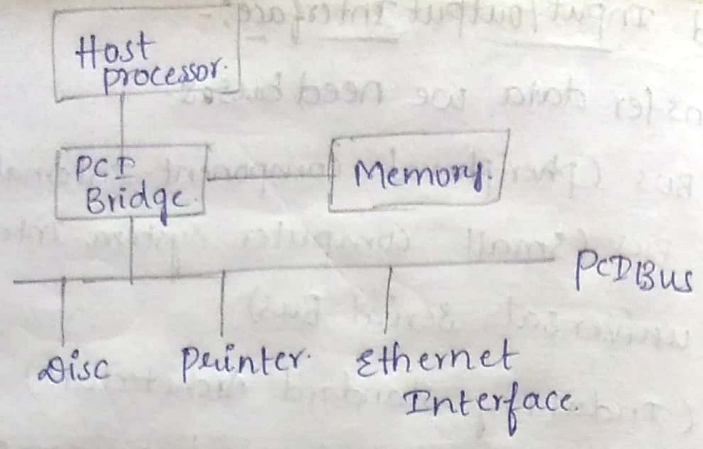
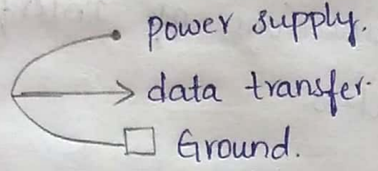
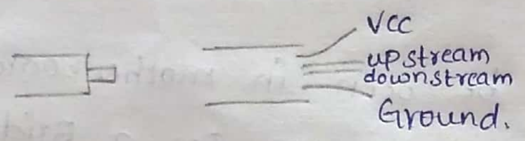


Fig:- use of PCI Bus in a computer System

USB Interface:-/USB port:-





→ USB Interface supports Parllel Data transfer

→ Before 1993 it use serial Data transfer. it takes 4 to 5 seconds to get a character on screen after we click on keyboard

→ In olden days different Input/output devices are connected to ~~se~~ different ports serially. Those connections are fixed

→ By using USB interface.

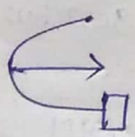
Advantage:- universal, fast, multifunctional simple.

→ In 1995 it came into existence.

Host

PC.

(Master)



Device

- cell
- pendrive
- keyboard (slave)

→ All I/O devices should plug into the host bus.

→ Host has 2 types of responsibilites:-

(i) Hardware responsibilites :- are providing electrical power, detect the USB connection, controlling data transfer.

(ii) Software responsibilites:- Handling connectivity configuring USB ~~drivers~~ devicers, Managing band width, running USB devicers.

→ There are mainly 2 types of USB plugs:-

(i) type A :- Ex:- Monitor port, requires more power.

(ii) type B ↳ has different hardware architecture.

↳ Micro A, Micro B, mini (types)

Examples of Micro plugs, keyboards, Pendrives with less power supply equal to 5V

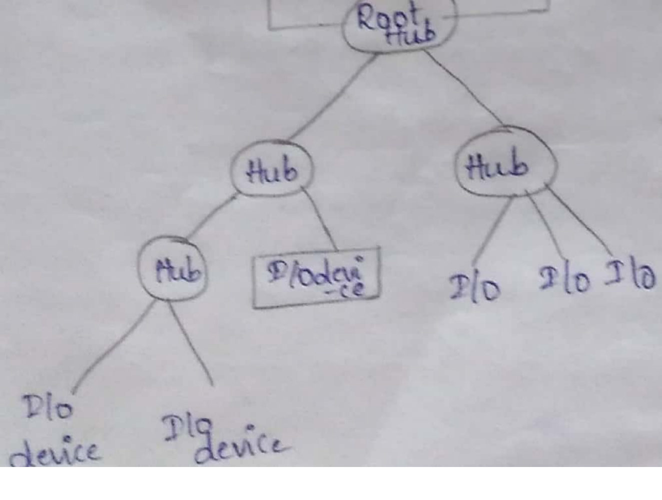
Advantage of type B:- is A plug with 5v supports Parllel Data transfer.

Mini is advantageous compared to Micro

→ Type A require more power, Type B require less Power.

→ Iphone, Ipod uses USB as data transfer. advantage is simplicity.

USB Bus tree Structure :-



→ we can connect I/O devices with different speeds to the Root hub.

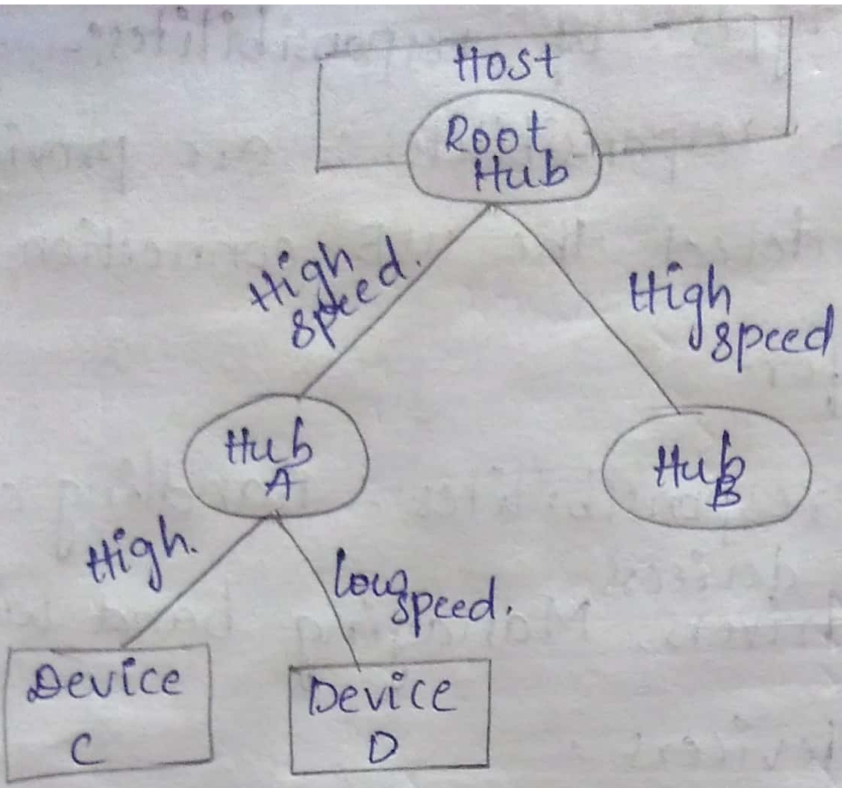


Fig:- ~~slip~~ split Bus ~~Arch~~ operation.

→ Here ~~slip~~ spliting is we split speed.