1) With a neat timing diagram, explain input transfer on a synchronous bus.

Ans:

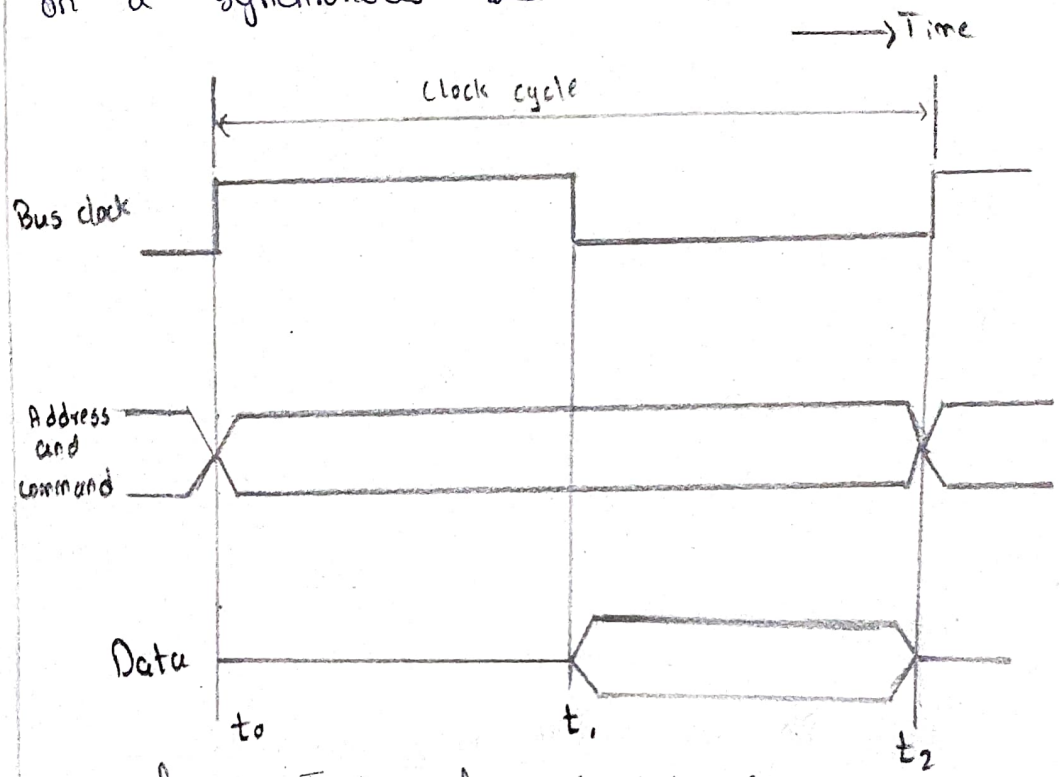


fig a: Timing of an input transfer on a synchronous bus.

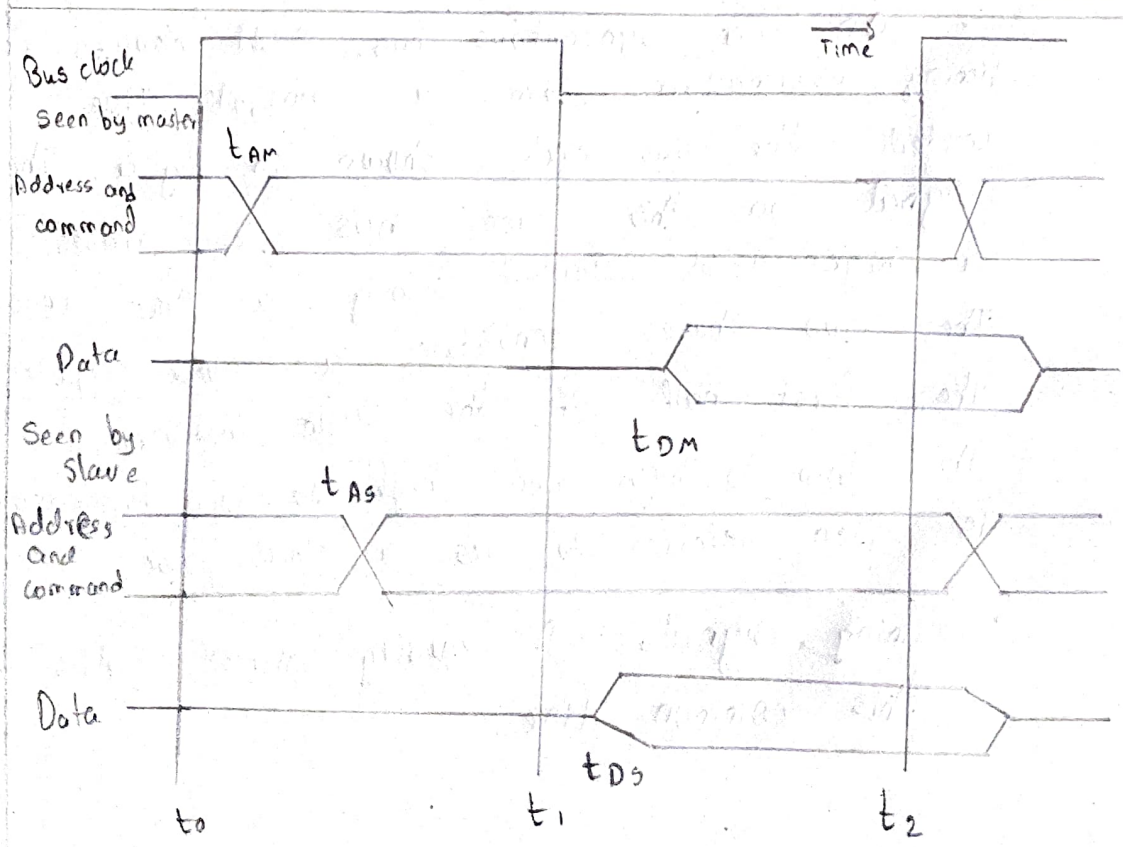


fig b: A detailed timing diagram for the input transfer

02

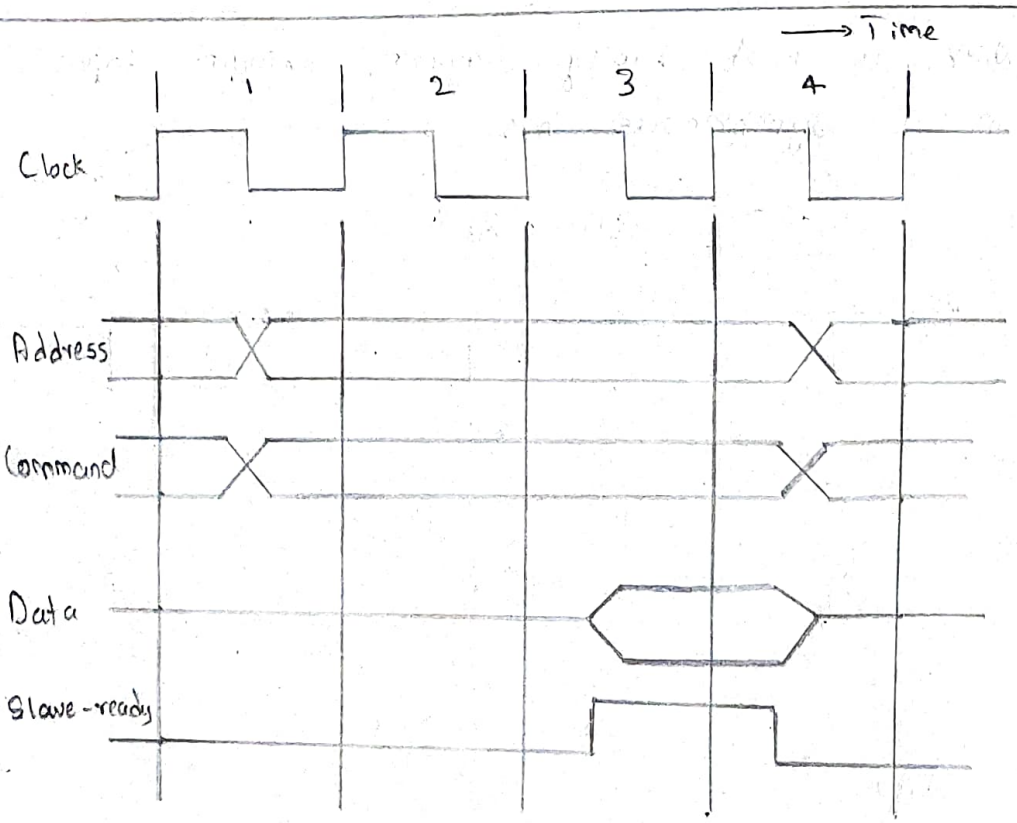


fig c: An input transfer using multiple clock cycles

- On a synchronous bus, all devices derive timing information from a control line called the 'bus clock', shown in fig a. The signal on this line has 2 phases: a high level followed by a low level. The two phases constitute a clock cycle. The first half of the cycle between the low-to-high and high-to-low transitions is often referred to as a clock pulse.

- Timing signals: A equally placed pulses on this common line.

- Bus cycle : Two or more timing intervals in which single data transfer takes place.
- The signal has 2 phases : a high level followed by a low level. The 2 phases constitute a clock cycle.
- At time t_0, the processor places the device address on address lines and sends an appropriate command on control lines as shown in fig a.
- The command will indicate an i/p operation and specify the length of the operand to be read.
- Information travels at speed determined by physical and electrical characteristics.
- Clock pulse width should be long to allow the devices to decode the address and control signal. Clock pulse width must be longer than maximum propagation delay.
- As signals are changing states b/w ~~to~~ t_0 to t_1, information on bus is unreliable.
- Slave places requested i/p data at t_1.
- At end of clock cycle (at t_2), master strobes data on data lines into its input buffer.
- The figure b shows 2 views, signal seen by master and seen by slave.

Master sends the address and command line at to. (fig b).

These donot actually visible on bus until t_{AM}

- At t_{AS}, signal reaches slave. Interface, slave decodes it.
- At t_1, slave sends requested data and at t_2, master loads data into its i/p buffer. The data must be continued to be valid after t_2, for a period equal to the hold time of the buffers.

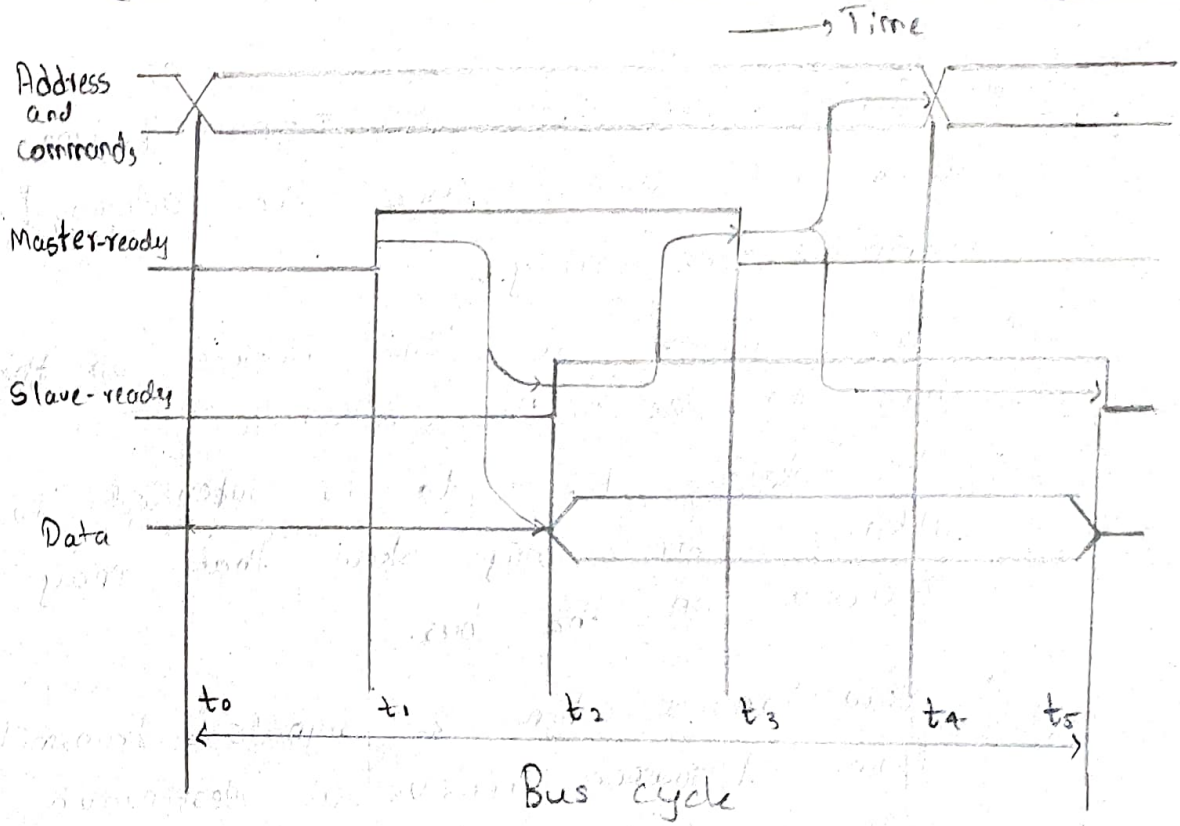
An input transfer using multiple clock cycles: (fig c).

- During clock cycle (1), master sends address and commands info. to the bus requesting 'read' operation. Slave receives and decodes it.
- At beginning of clock cycle - 2, it makes access to respond immediately.
- The data becomes ready and are placed in the bus at clock cycle (3). Here, the slave asserts a control signal called slave ready.
- The master strobes the data to its input - buffer at the end of clock cycle (3).
- The bus transfer operation is now complete and master sends a new address to start a new transfer in clock cycle (4).

2) Explain handshake control of data transfer during an input operation.

Ans.

Handshake control of data transfer during an input operation:



- This asynchronous bus uses handshake signals b/w master and slave for coordinating data-transfers.
- Master ready: control signal used to indicate that master is ready for a transaction (MR).
- Slave ready: Control signal used to indicate that slave is ready for a transaction (SR).

At t_0 - master places the address and command information on the bus and all devices on the bus decode this information.

- At t_1, master sets MR - signal line to inform all devices that the address / command information is ready.
- This causes all other devices on the bus to decode the address.
- The delay $t_1 - t_0$ is intended to allow for any skew that may occurs on the bus.
- Skew occurs when 2 signals transmitted from 1 source arrive at destination at different time.
- Therefore, the delay $t_1 - t_0$ should by larger than the maximum possible bus skew.
- At t_2, slave performs the i/p operation and sets SR signal to 1 to inform all the devices that is ready.

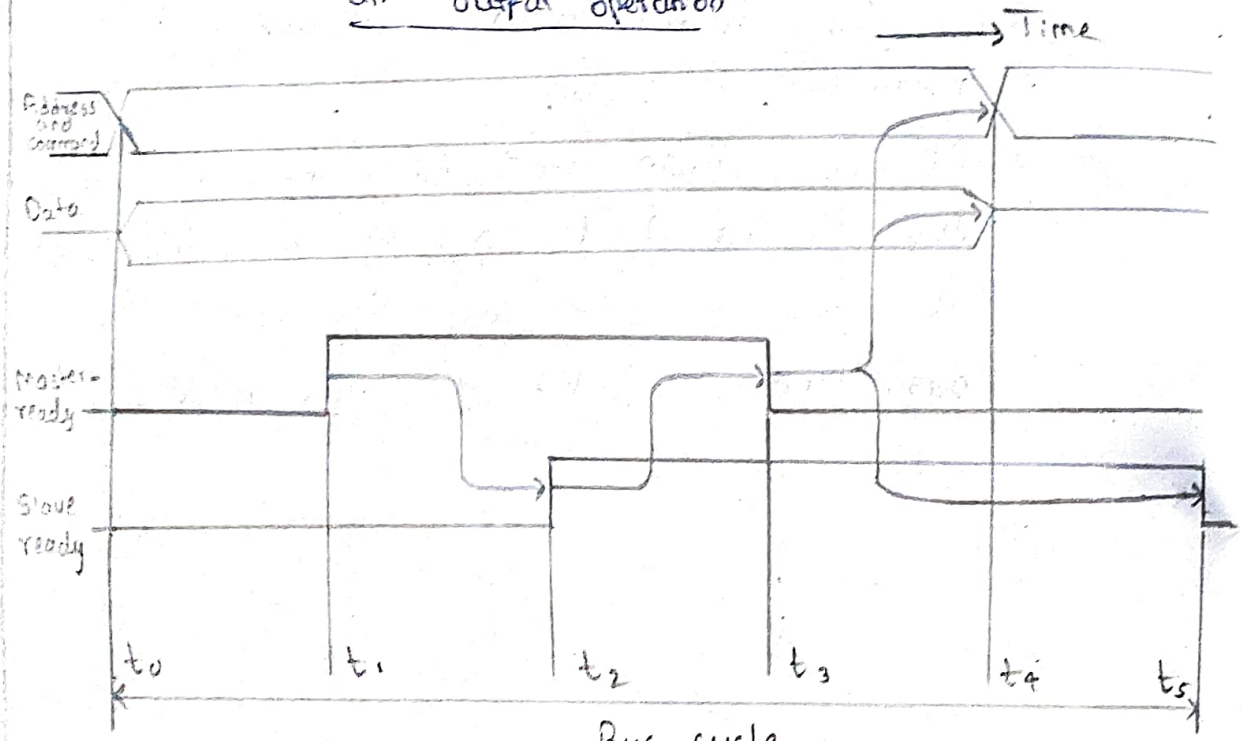
- At t_3, SR signal arrives at master indicating that the i/p data are available on bus.
- At t_4, master removes address/ command information from bus.
- At t_5, when the device interface receives the 1 to 0 transmission of MR signal, it removes data and SR signal from the bus. This completes the i/p transfer.

08

3) Explain handshake control of data transfer during an output operation

Ans

Handshake control of data transfer during an output operation



- The timing for an output operation is illustrated in above figure.
- It is essentially the same as for an input operation. It also contains 2 control lines:
  1) MR (Master - Ready) :- Used to indicate that master is ready for transaction.
  2) Slave Ready (SR) :- used to indicate that slave is ready for transaction.

- At t_0, master places address or command on bus, ie master places the output data on the data lines.
- At t_1, master sets MR signal to 1 to inform all devices that the address / command info. is ready.
- MR signal = 1 causes all devices on the bus to decode the address.
- The delay t_1 to t_0 is intended to allow for any skew that may occurs on the bus.
- Skew occurs when signals transmitted from 1 source arrive at destination at different time.
- Therefore the delay $t_1 - t_0$ should be larger than the maximum possible bus skew.
- At t_2, slave performs required output operation and sets SR signal to 1 to inform all devices that it is ready.
- At t_3, SR signal arrives at master indicating that the output data are available on bus.

At t_4, master removes address / command information from bus.

At t_5, when device interface receives the 1 to 0 transition of MR signal. it removes data and SR signal from the bus. This completes output operation.

4) Define the process of Bus arbitration. Explain with neat diagram

Ans:
- The device that is allowed to initiate data transfer on the bus at any given time is called the bus master. Bus arbitration is the process by which the next device to become the bus master is selected and bus mastership is transferred to it.
- A device that wishes to use the bus sends a request to the arbiter. When multiple requests arrive at same time, the arbiter selects on request and grants the bus to corresponding device.
- If there is no particular urgency, the arbiter may grant bus using a simple round robin scheme.

Bus arbitration:

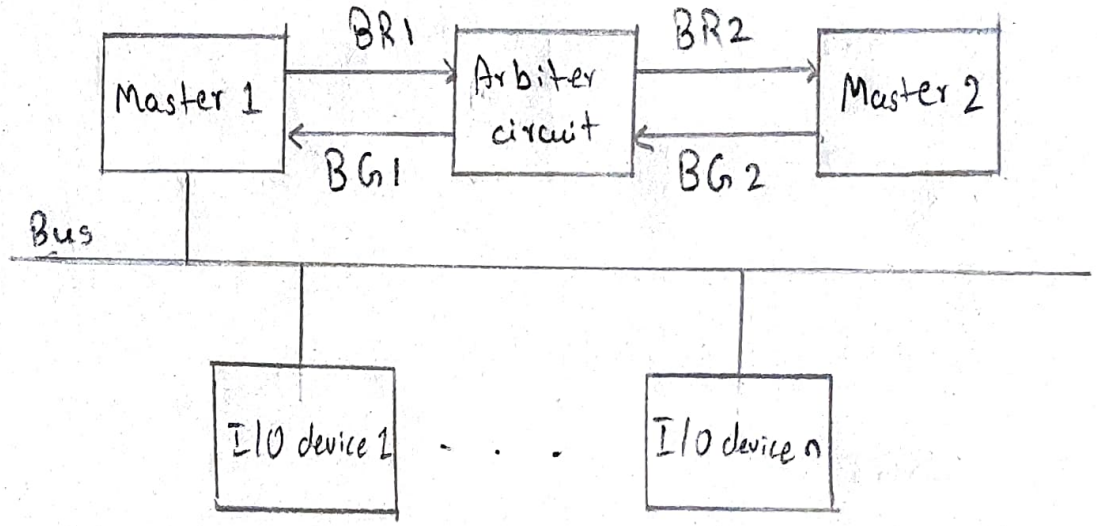


Fig. a

- Fig. a illustrates an arrangement for bus arbitration involving 2 masters.
- BR1 and BR2 → Bus - request lines.
- BG1 and BG2 → Bus - grant lines.
- Master requests use of the bus by activating its bus request line.
- If single BR is sent, the arbiter activates the corresponding bus grant.
- Once the transfer is completed, the master deactivates its bus - request and the arbiter deactivates its bus - grant.

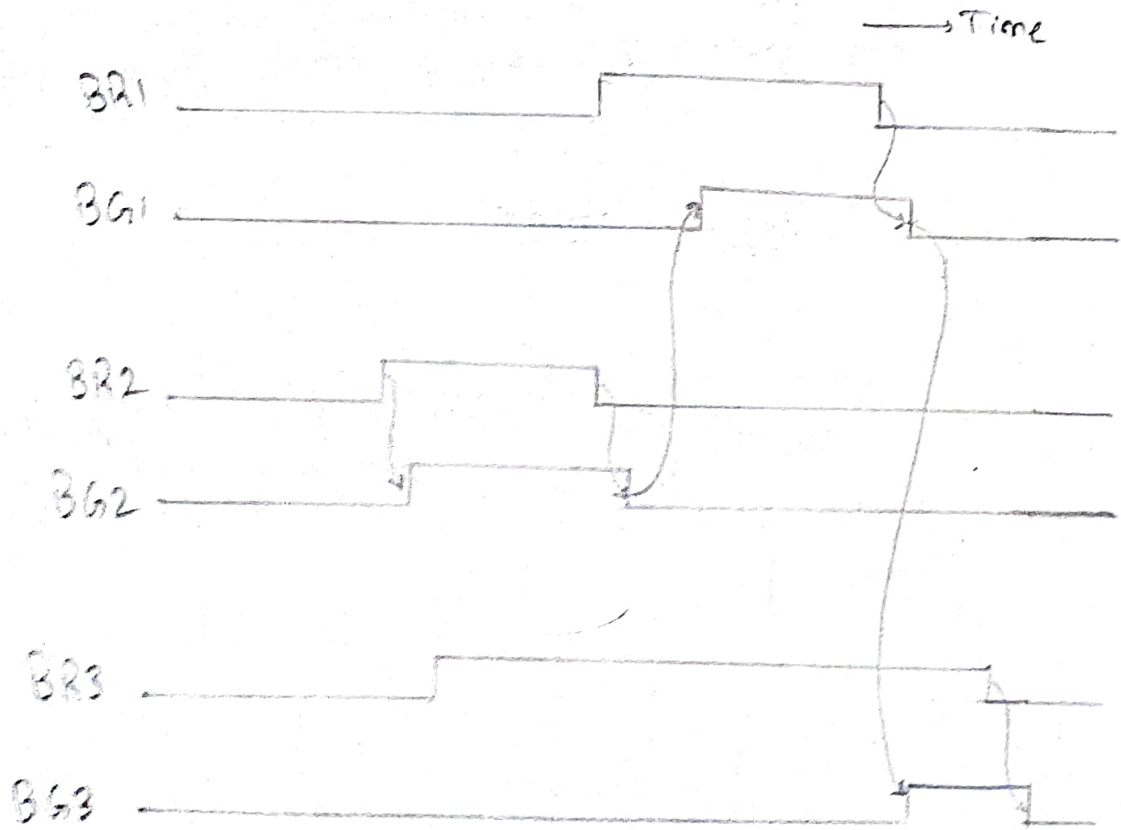


Fig. b

- The fig. b illustrates a possible sequence of events for the case of 3 masters.
- Let say, master 2 sends a request to use the bus first. Since there is no other request, the arbiter grants the bus to this master by asserting bus 2. and it releases bus by deactivating BR 2.
- By this time, both master 1 and 3 have activated their request lines.
- Here, since master 1 has higher priority than that of master 3, the arbiter activates BG 1 granting the bus to master 1.
- When master 1 releases the bus by deactivating BR 1, the arbiter deactivates BG 1 and activates BG 0 to grant the bus to master 3.
- It is important to note that the bus is granted to master 1 before master 3 even though master 3 activated its request line before master 1.

5) Define bus. Explain I/O interface for an input device with suitable diagrams.

Ans: Bus is defined as set of parallel wires used for data communication b/w different parts of computer.

- The below figure a, only 1 source / destination pair of units can use this bus to transfer data at any 1 time.
- The bus consists of 3 sets of lines used to carry address, data and control signals. I/O devices interfaces are connected to these lines.

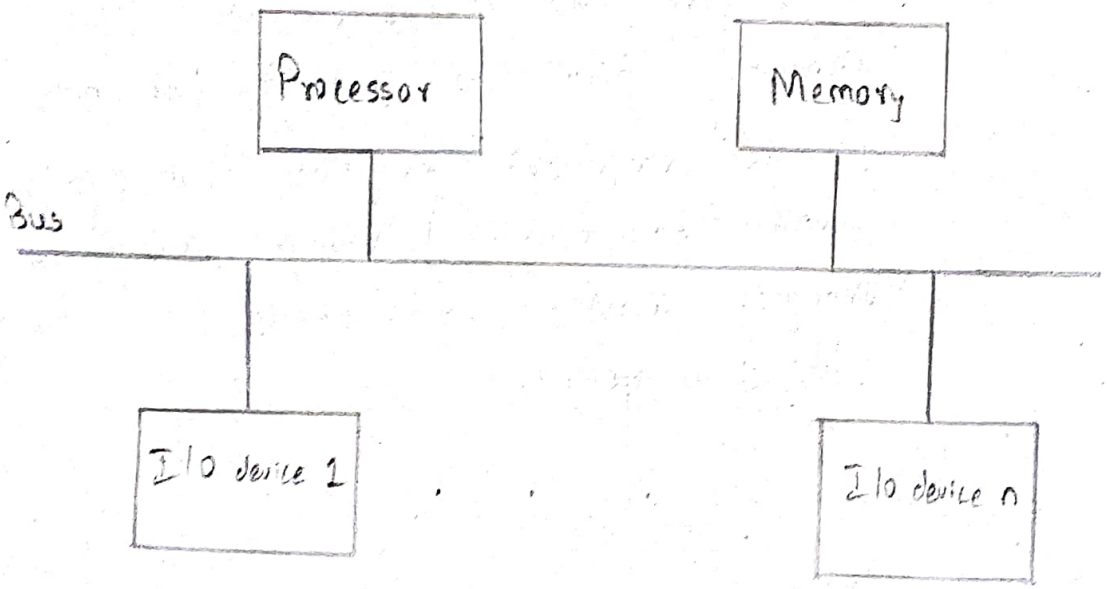


fig a. A single bus structure

- When the processor places a particular address on the address lines, it is examined by the address decoders of all devices on the bus.
- The devices that recognizes this address responds to the command issued on the control lines.
- The processor uses the control lines to request either read or write operation, and the requested data are transferred over the data lines.
- Any machine instruction that can access memory can be used to transfer data to / from an I/O device.
- For ex: if the i/p device in fig b is a keyboard, and if DATAIN is its data register, the instruction:

  Load R2, DATAIN.

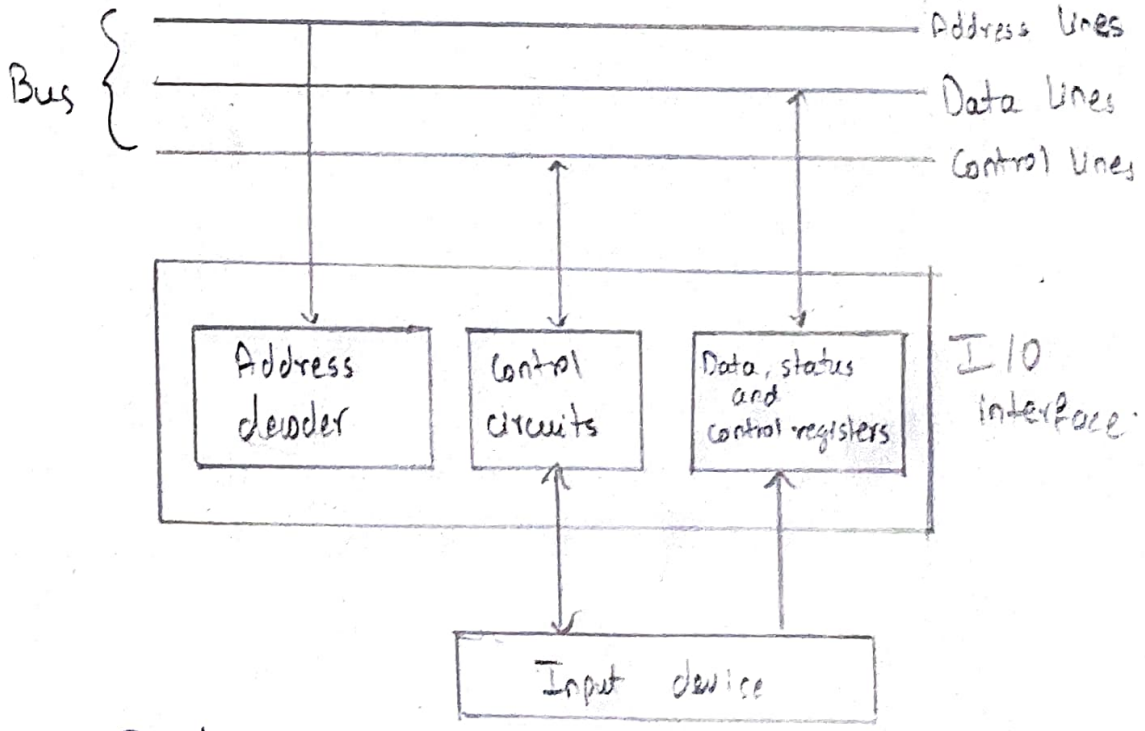


Fig b. I/O interface for an input device

- This instruction reads the data from DATAIN and stores them into processor register R2.
- Similarly the instruction

  Store R2, DATAOUT.

  sends the contents of register R2 to location DATAOUT, which may be that data register of a display device interface.
- The status and control registers contain information relevant to the operation of the I/O devices.
- The address decoder, the data and status registers and the control circuitry required to coordinate I/O transfers constitute the device's interface circuit.