8255 Data Transfer Modes :

8255 ports can work in three different modes.

1) Mode 0 : Simple I/O :

In simple I/O mode, all the three ports are available for data transfer. But data transfer is not reliable as there is no handshaking. Each port can be programmed as i/p or o/p port individually.

2) Mode 1 : Handshake I/O

Both Mode 1 and Mode 2 involve handshaking.

A port programmed in Mode 1 will perform handshaking in one direction for the rest of its life.

Either a port will be programmed for i/p handshaking ie it will be connected to i/p devices like keyboard, mouse etc.

Or

A port can be programmed for o/p handshaking ie lifelong it will perform output handshaking only.

In mode 2, port can do i/p handshaking or o/p handshaking.

Suppose in your computer, a port has been created specifically for keyboard. Back in time, there were ports specifically for keyboard, mouse etc. Today we use Universal Serial Bus (USB)

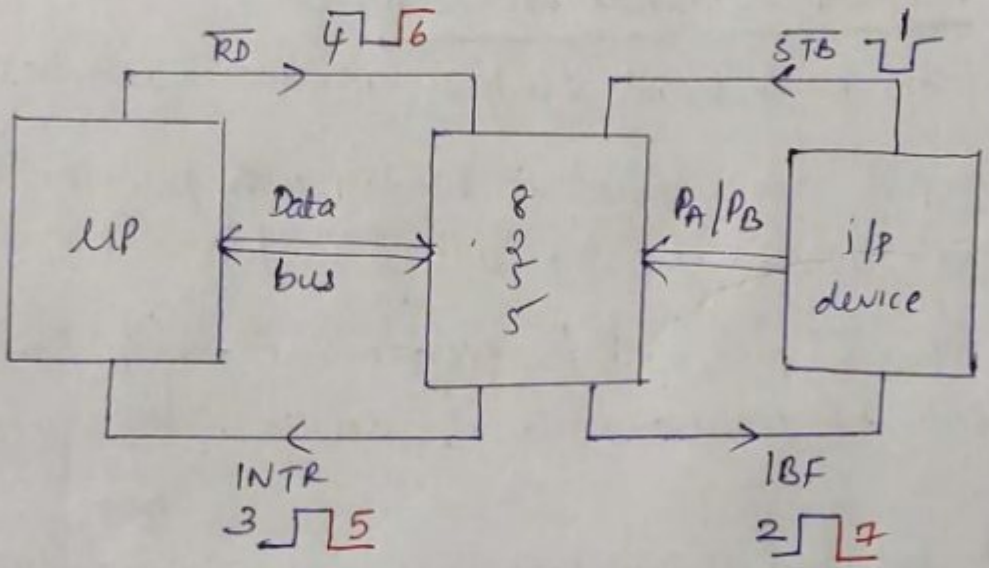
Semester: IV

On the former case, the port uses M1 – I/p h/s.

If a port is created specifically for a speaker, it uses M1 – O/p h/s.

If a port is created for pendrive : M2
(ie both i/p & o/p h/s is done).

Mode 1 Input Handshaking :



µP is connected to 8255 through data bus. On 8255 some I/p device is connected. This device will be connected on Port A or Port B.

Note:- It cannot be connected on Port C as port C is sacrificed for handshaking.

Input device send some data to 8255. 8255 will send that data to µP.

Handshaking takes place in the following manner:

① The i/p device gives a signal called strobe ($\overline{STB}$). $\overline{STB}$ is an indication to 8255 that i/p device is going to send some data now.

If i/p device wants to send data, why can't it directly send the data? Why it has to inform 8255 before sending the data?

Whenever an i/p device is going to send data or µP is sending data to a device, it has to inform, otherwise the data will be treated as garbage. When computer is on, it means there is power supply and there will always be some data on the port which is garbage. So before sending data, the device has to inform.

② The data sent by the i/p device gets accepted by the port and gets stored in a temporary register called I/p Buffer Register.
When the register is full, 8255 gives a signal IBF (Intend Buffer Full). IBF goes high informing the I/p device that it has to stop sending further data and hence data loss is prevented.

Semester: IV

(3) 8255 now interrupts the µP through the INTR line thereby informing µP that it has data waiting to be read.

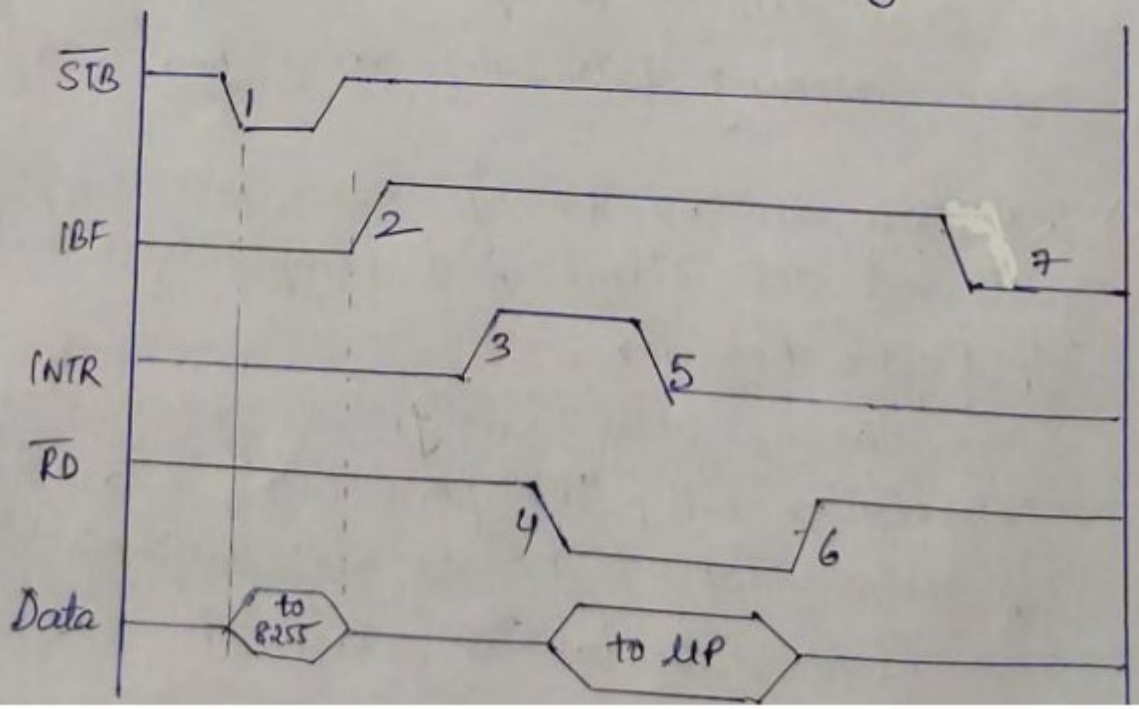
(4) On response to the interrupt, µP when free issues the $\overline{RD}$ signal and reads the data.

(5) When µP starts reading the data, INTR will go low immediately.

(6) Eventually $\overline{RD}$ will go high which means µP has successfully read the data.

(7) Now 8255 will make IBF go low informing I/P device that it can send more data in the above sequence.

Note:- By default IBF is low, indicating 8255 is free.

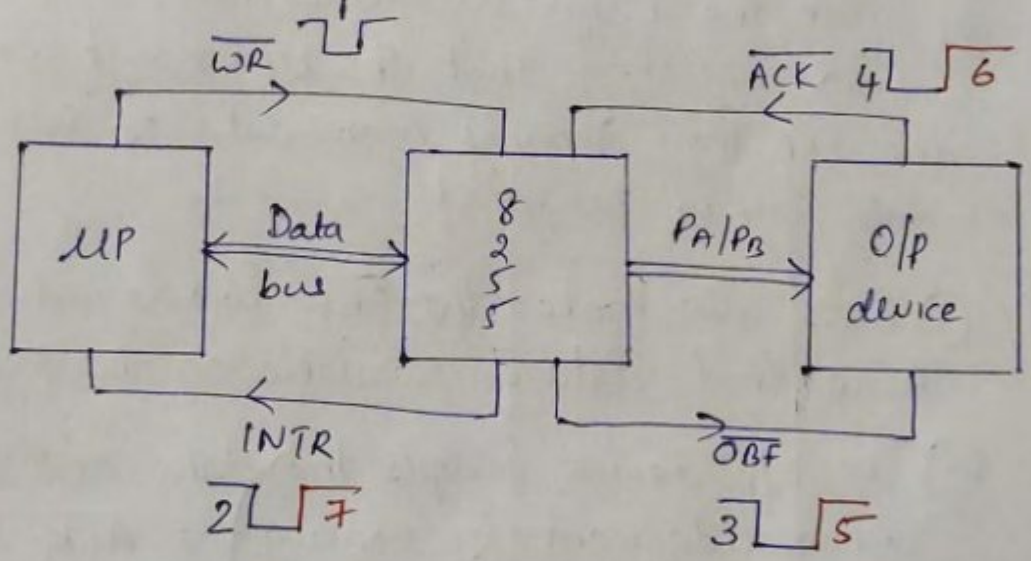


Handshaking requires 4 signals. But all these 4 lines are not taken from Port C.

$\overline{RD}$ and $\overline{WR}$ are physically present on 8255, so $\overline{RD}$ and $\overline{WR}$ lines are not taken from Port C.

To give $\overline{STB}$, IBF and INTR, lines are taken from Port C.

Mode 1 - O/p Handshaking:



When μP wants to send data to O/p device, μP will send data to 8255 and 8255 sends it to O/p device.

Before sending data to 8255, μP wants to know whether 8255 is free or not. How?

Option 1: μP keeps checking 8255 – Are you free? {Polling}

Option 2: 8255 tells μP when it is free.

If 8255 is free, INTR = 1 by default.

O/p Handshaking takes place in the following manner:

(1) Whenever µP wants to send data and the o/p port is empty (indicated by a high on the INTR line), µP writes data on the o/p port by giving the $\overline{WR}$ signal.
The data gets stored in the buffer register.

(2) When the buffer register is full, 8255 makes INTR low indicating that the µP should wait. This prevents the µP from sending more data to 8255 and hence data loss is prevented.

(3) 8255 also makes the $\overline{OBF}$ low to indicate the o/p device that data is available on the port.

(4) The o/p device accepts the data and sends an acknowledgement by making the $\overline{ACK}$ low. The data is thus transferred to the o/p device.

(5)(6) Now the $\overline{OBF}$ and $\overline{ACK}$ lines go high.

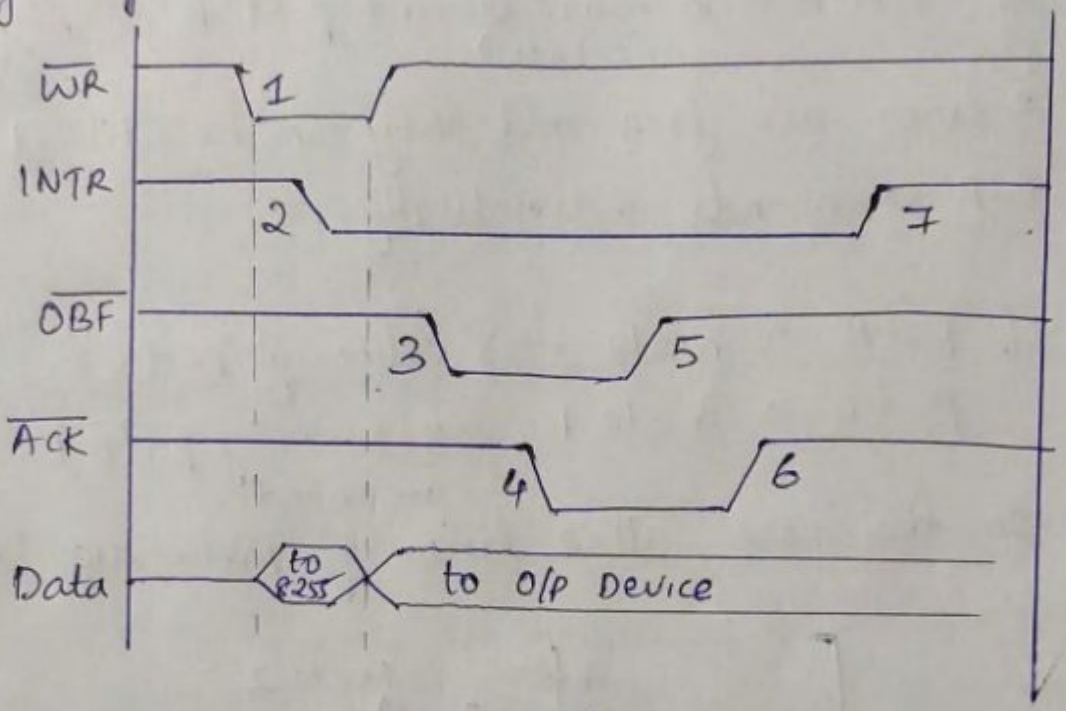
(7) The INTR line becomes high to allow the µP to transfer new data ie, the o/p port is free.

This process is repeated for further bytes.

In o/p handshaking, each port uses 3 lines of Port C (C for $\overline{OBF}$, $\overline{ACK}$ and INTR). $\overline{WR}$ of 8255 is also used.

Academic Year: 2022-23 **Semester: IV**
Class/Branch: SE **Subject: MP**

Timing Diagram :



Mode 2 Bidirectional Handshake

In mode 2, port A can anytime do I/P h/s and O/P h/s.

I/P h/s	O/P h/s
• $\overline{STB}$	• $\overline{OBF}$
• IBF	• $\overline{ACK}$
• INTR	• INTR

So total 5 signals are required for bidirectional handshake and they are taken from port C.

3 lines of port C are still free.

So port B has 2 options - to work in mode 0
to work in mode 1.

Academic Year: 2022-23 **Semester: IV**
Class/Branch: SE **Subject: MP**

→ If port A is in mode 2 and port B is in mode 0
(5 lines) (no hls)
3 lines are free and they can be utilized by BSR commands individually.

→ If port A → mode 2 {5 lines of port C}
port B → mode 1 {3 lines of port C}
are used.
In this case, all 8 lines of port C are fully utilized.