

## **Programmable parallel ports and Handshake input/output**

### **Parallel Communication**

- It is a method of conveying multiple binary digits (bits) simultaneously i.e., all bits of word are transferred at a time.
- It is faster.
- Hardware requirement is complex.
- E.g., 8255A PPI

### **Methods of parallel data transfer**

#### **i) Simple I/O**

- When you need to get digital data from simple switch, such as thermostat, into microprocessor, all you have to do is connect the switch to an I/O port line and read the port.

- Likewise, when you need to output data to simple display device, such as LED, all you have to do is connect the input of the LED buffer on an output port pin and output the logical level required to turn on the light.

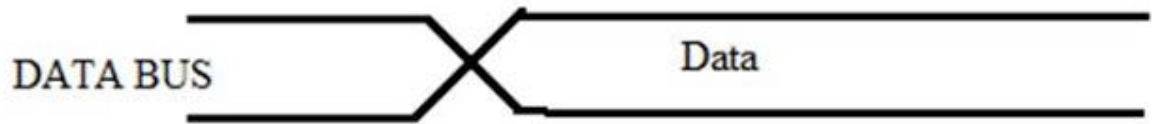


Fig: Simple I/O

The timing waveform represents the situation.

- The crossed lines on the waveform represent the time at which a new data byte becomes valid on the output lines of the port.
- The absence of other waveforms indicates that this output operation is not directly dependant on any other signal.

## ii) Strobe I/O

- In many applications, valid data is present on an external device only at a certain time, so it must be read in at that time.
- E.g. the ASCII-encoded keyboard. When a key is pressed, circuitry on the keyboard sends out the ASCII code for the pressed key on eight parallel data lines, and then sends out a strobe signal on another line to indicate that valid data is present on the eight data lines.

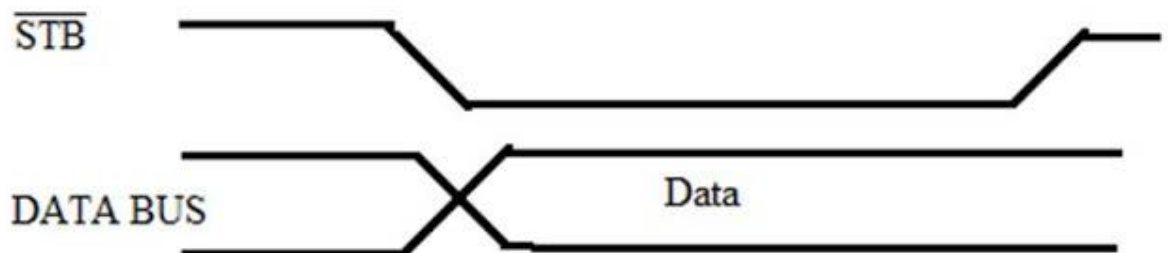
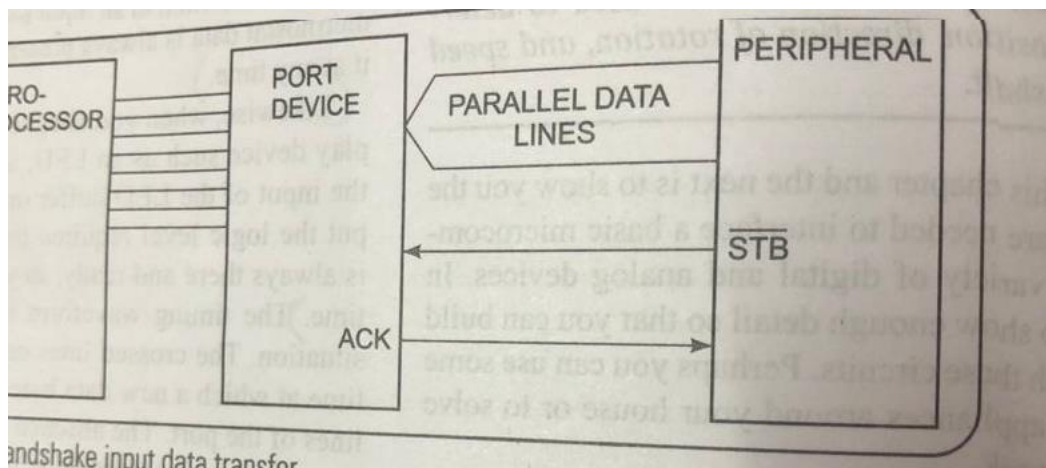


Fig: Strobe I/O

- This timing waveform represents strobe I/O.
- The sending device, such as a keyboard, outputs a parallel data on the data lines, and then outputs an STB signal to let you know that valid data is present.
- For low rates of data transfer, such as from a keyboard to a MP, a simple strobe transfer works well.
- However, for higher speed data transfer, this method does not work because there is no signal which tells the sending device when it is safe to send the next data byte.
- In other words, the sending system might send data bytes faster than the receiving system could read them.
- To prevent this problem, a handshake data transfer scheme is used.



### iii) Single handshake I/O

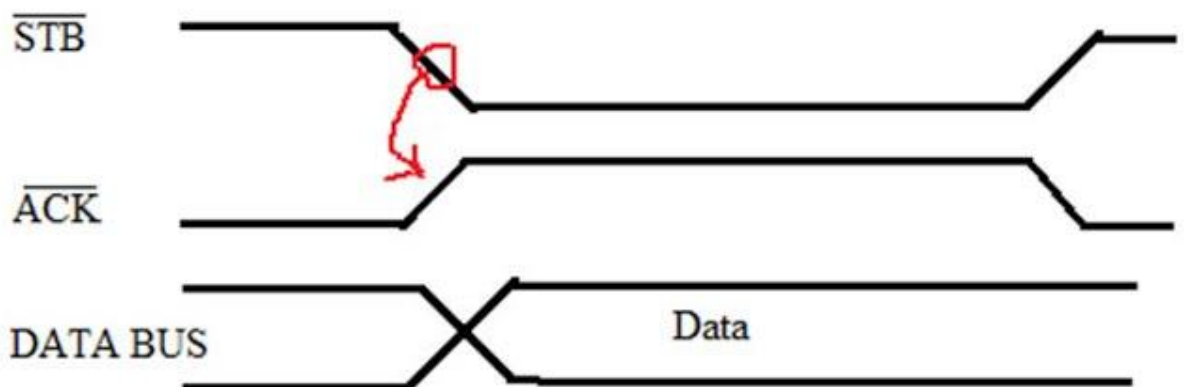


Fig: Single Handshake I/O

- It shows the timing waveform for a handshake data transfer from a peripheral device to a MP.
- The peripheral outputs some parallel data and sends an STB signal to the MP.
- The MP detects the asserted STB signal on a polled or interrupts basis and reads in the bytes of data.
- Then, the MP sends ACK (acknowledge) signal to the peripheral to indicate that the data has been read and that the peripheral can send next byte of data.
- The point of this method is that the sending device or system is designed so that it does not send the next byte until the receiving device or system indicates with an ACK signal that it is ready to receive the next byte.

#### iv) Double handshake I/O

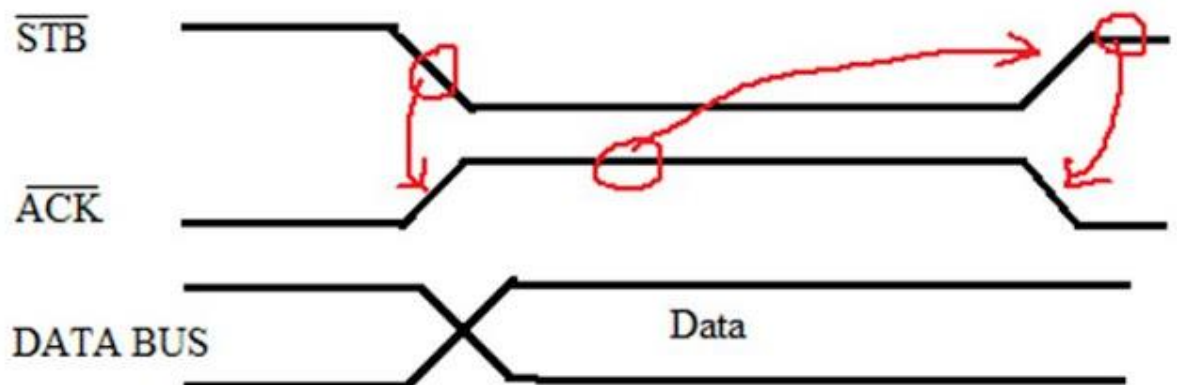


Fig: Double Handshake I/O

- For data transfer where even more coordination is required between the sending system and the receiving system, a double handshake is used.
- The sending (peripheral) device asserts its STB line low to ask the receiving device whether it is ready or not for data reception.

- The receiving system raises its ACK line high to indicate that it is ready.
- The peripheral device then sends the byte of data and raises its STB line high to assure that the valid data is available for the receiving device (MP).
- When MP reads the data, it drops its ACK line low to indicate that it has received the data and requests the sending system to send next byte of data.