Chapter 7. Input and Output Interface

7.1 Serial and Parallel communication

Most of the microprocessors are designed for parallel communication. In parallel communication number of lines required to transfer data depends on the number of bits to be transferred. For example, to transfer a byte of data, 8 lines are required and all 8 bits are transferred simultaneously. Thus for transmitting data over a long distance, using parallel communication is impractical due to the increase in cost of cabling. In such situation serial communication is used. In serial communication one bit is transferred at a time over a single line.

Figure 7.1 shows the serial and parallel communication.

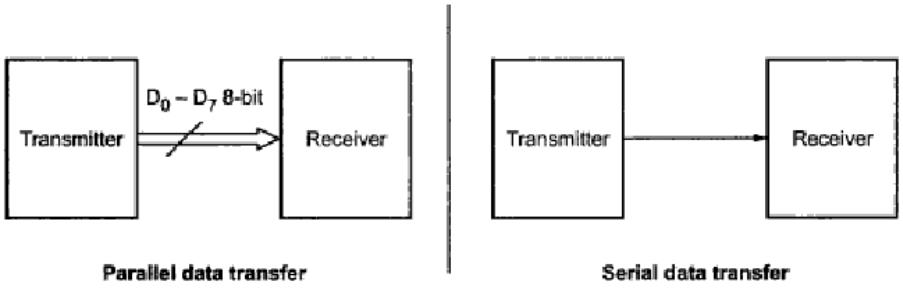


Figure 7-1: Parallel and serial data transfer

Parallel Communication

In parallel transmission, multiple bits are sent simultaneously on different channels (wires, frequency channels) within the same cable, or radio path and synchronized to a clock. Parallel devices have a wider data bus than serial devices and can, therefore, transfer data in words of one or more bytes at a time. As a result, there is a speedup in parallel transmission bit rate over serial transmission bit rate. However, this speedup is a trade-off versus cost since multiple wires cost more than a single wire and as a parallel cable gets longer, the synchronization timing between multiple channels becomes more sensitive to distance. The timing for parallel transmission is provided by a constant clocking signals sent over a separate wire within the parallel cable; thus parallel transmission is considered synchronous.
Parallel transfer may take place:

- Between a microprocessor and memory
- Between a microprocessor and IO chip
- Between IO chip and peripheral device

7.2.Methods of Parallel Communication

Simple Input and Output:

When you need to get digital data from a simple switch, such as a thermostat, into a microprocessor, all you have to do is connect the switch to an input port line and read the port. The thermostat data is always present and ready, so you can read it at any time.

Likewise, when you need to output data to a 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 logic level required to turn on the light. The LED is always there and ready, so you can send data to it at any time. The timing waveform in fig. 7-2, represents this situation. The crossed lines on the waveform represents 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 dependent on any other signals.

Simple 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. An example of this is 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. As shown in fig. 7-2, you can connect this strobe line to an input port line and poll it to determine when you can input valid data from the keyboard. Another alternative, is to connect the strobe line to an interrupt input on the processor and have an interrupt service procedure read in the data when the processor receives an interrupt. The scheme behind this arrangement is that the transfer is time dependent. Thus you can only read data when a strobe pulse tells you that the data is valid.

Figure 7-2 shows the timing waveform which represent this type of operation. The sending device, such as a keyboard, output parallel data on the data lines, and then outputs an $\overline{STB}$ signal to let you know that valid data is present.

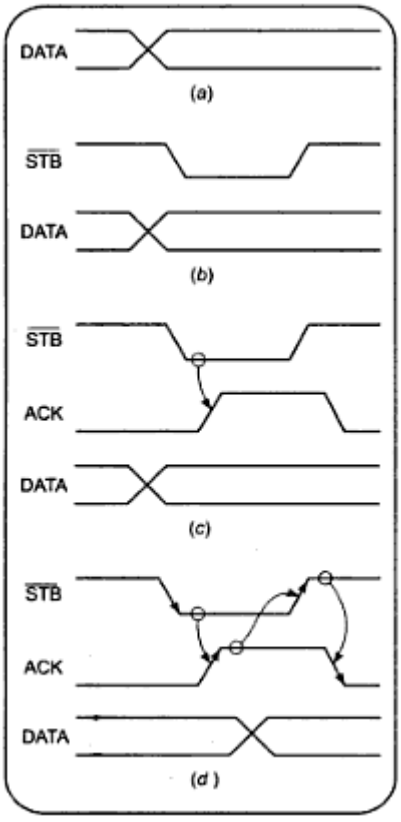


Figure 7-2: Parallel data transfer (a)Simple output (b) Simple strobe I/O (c) Single handshake I/O (d) Double handshake I/O

Single handshake I/O

In single handshake, a peripheral device first sends a "Strobe signal" to the microprocessor to indicate that it is ready to send data. The microprocessor, upon detecting the strobe signal, opens up its input port and receives the data. After receiving data, it sends an "Acknowledge signal" to

the peripheral to indicate that transmission has been completed. A transmission session has been completed.

Double Handshaking I/O

In double handshake, first the peripheral device sends a strobe signal, the microprocessor, sends the acknowledge signal to indicate that it is ready to receive data. After which data is received. After sending data, the peripheral sends a strobe signal to indicate data transmission completion, due to which, the microprocessor drops its acknowledge signal and a session has been completed.

The only difference in the two is that, in double handshake, the peripheral is informed about the microprocessor's readiness to receive data. This is doesn't happen in single handshake. So the name follows "double handshake", literally meaning "double confirmation".

7.3.Serial transmission

In serial transmission, bits are sent sequentially on the same channel (wire) which reduces cost for wire but in turn slows the speed of transmission. For serial transmission, some additional time is required because bits are to be assembled and sent as a unit and then disassembled at the receiver.

Serial transmission can be either synchronous or asynchronous. In synchronous transmission, groups of bits are combined into frames and frames are sent continuously with or without data to be transmitted. In asynchronous transmission, groups of bits are sent as independent units with start/ stop flags and no data link synchronization, to allow arbitrary size gaps between frames. However, start/stop bits maintain physical bit level synchronization once detected.

Example of serial mode transmission includes transmission between two computers or from a computer to an external modem using RS-232 protocol.

Parallel Vs Serial data Transfer

Parallel data transfer	Serial data transfer
Multiple bits (generally 8 bits) of data is transferred at a time	1-bit of data is transferred at a time
9 lines are required to be connected between two points (including GND)	Only two lines are required to be connected
Data transfer is fast	Data transfer is slow
More expensive	Cheaper
Used over small distances	Used over long distances

7.4. Types of data communication systems

The data transmission can also be categorized on the basis of how transmission occurs as:

1. Simplex 2. Duplex

1. Simplex:

In simplex, the hardware exists such that the data transfer takes place only in one direction. There is no possibility in the other direction. A typical example is transmission from a computer to the printer.

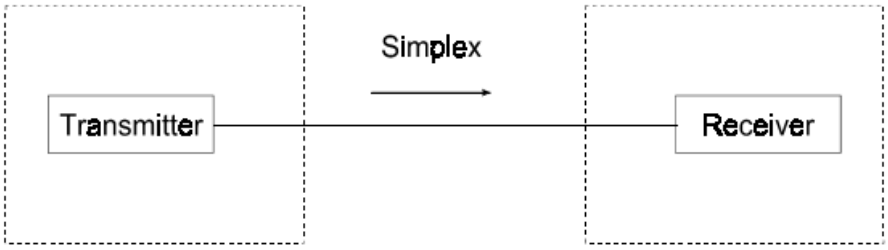


Figure 7-2: Simplex Transmission

2. Duplex:

In duplex the transmission is in both directions. It is further divided into two groups: half duplex and full duplex

Half Duplex: In half duplex transmission allows the data transfer in both directions, but no simultaneously. A typical example is a walkie-talkie.

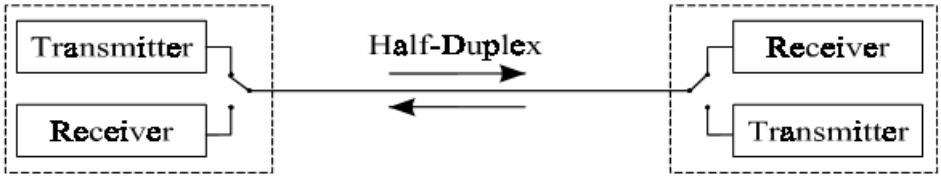


Figure 7-3: half duplex transmission

Full Duplex: The full duplex transmission allows the data transfer in both direction simultaneously. The typical example is transmission through telephone lines.

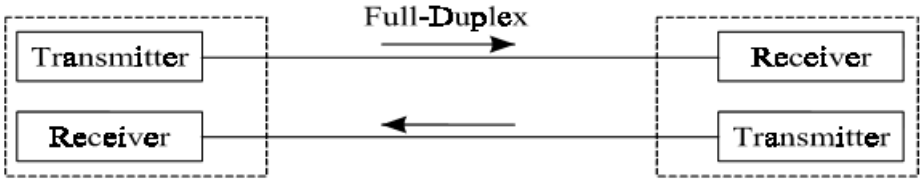


Figure 7-4: full duplex transmission

7.5.Serial Transmission Formats

The data in the serial communication may be sent in two formats.

a) Asynchronous B) Synchronous

Asynchronous serial data transfer

Asynchronous format are character oriented. Asynchronous system sends data bytes between the sender and receiver by packing the data in an envelope. This envelop helps transport the character across the transmission link that separates the sender and receiver. The transmitter creates the envelope and the receiver uses the envelope to extract the data. Each character (data byte) that the sender transmits is preceded with a start bit and suffixed with stop bit. These extra bits serve to synchronize the receiver with the sender.

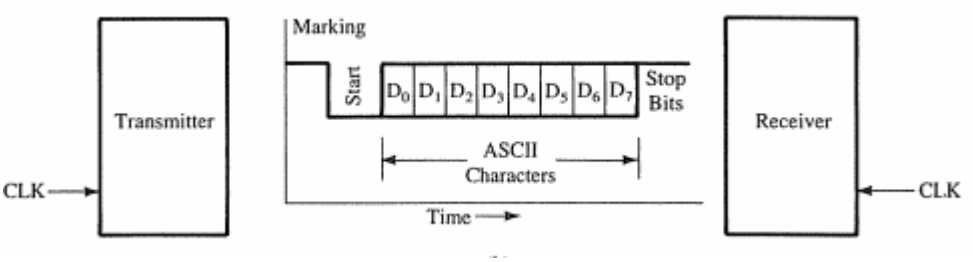


Figure 7-5: Transmission format for asynchronous transmission

Normally, the output line of transmitter is maintained high i.e. in marking state. Now if one wants to send a byte of data then first the output line is made low for 1 clock pulse to indicate start of data byte so it is called *start bit*. After that the data bits are transferred serially D0, D1…D7.

After D7, i.e. last bit, the output is maintained high for specific period to indicate the end of data and these bits are called *stop bits*.

The transmission begins with one start bit (low), followed by a character and one or two stop bits (high). This is known as *framing.*

(b) Synchronous serial data transfer

The start and stop bits in each frame of asynchronous format represents wasted overhead bytes that reduce the overall character rate. These, start and stop bits can be eliminated by synchronizing receiver and transmitter. They can be synchronized by having a common clock signal. Such a communication is called synchronous serial communication. Fig. 7-6 shows the transmission format of synchronous serial communication. In this transmission synchronous bits are inserted instead of start and stop bits.

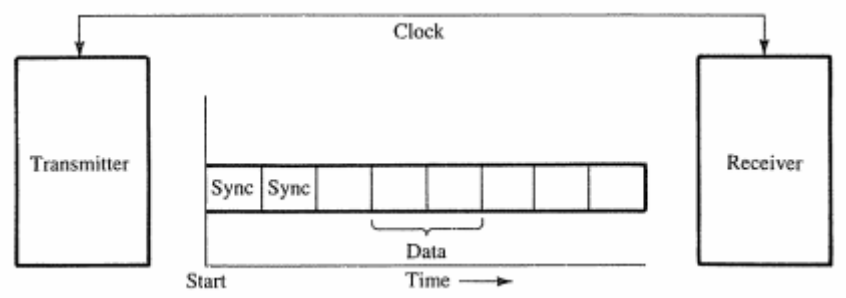


Figure 7-6: synchronous transmission format

Asynchronous vs synchronous serial data transfer

Asynchronous serial data transfer	Synchronous serial data transfer
Used to transfer one character at a time	Used to transfer group of characters at a time
Synchronous characters are not transmitted along with characters	Synchronous characters are transmitted along with group of characters
Start and stop bit for each character which forms a frame	No start and stop bit for each character
Two Clocks are used for Tx and Rx	Single clock is used for both Tx and Rx.
Software implementation is possible	Software implementation is not possible
Used for data rates $\leq$ 32 kbits/sec	Used for data rate $\geq$ 20 kbits/sec

Baud Rate

In serial communication, the rate at which data bits are transmitted is termed as *Baud rate.* The Baud rate may be defined as characters/seconds or change in voltage level/second. Generally, the rate at which bits are transmitted is given in terms of bits/second or baud.

Baud = Bits transmitted/second.

7.6. Introduction to programmable communication interface 8251

The 8251 is a USART (Universal Synchronous Asynchronous Receiver Transmitter) for serial data communication. As a peripheral device of a microcomputer system, the 8251 receives parallel data from the CPU and transmits serial data after conversion. This device also receives serial data from the outside, converts it into parallel data and then transmits parallel data to the CPU after conversion.

Features of 8251 USART

1. 28-pin DIP package, all inputs and outputs are TTL compatible.

2. It has built in baud rate generator, and it also provides different baud rate for transmitter and receiver.

3. Single +5 V supply

4. Compatible with 8085 CPU

5. Supports both synchronous and asynchronous mode of operations.

6. Error detection – Parity, overrun, framing

7. Transmitter and receiver contain full duplex, double buffered system.

7.7.Block Diagram Description of 8251

The functional block diagram of 825 1A consists five sections. They are:

- Read/Write control logic
- Transmitter
- Receiver
- Data bus buffer
- Modem control

The functional block diagram is shown in fig 7-7:

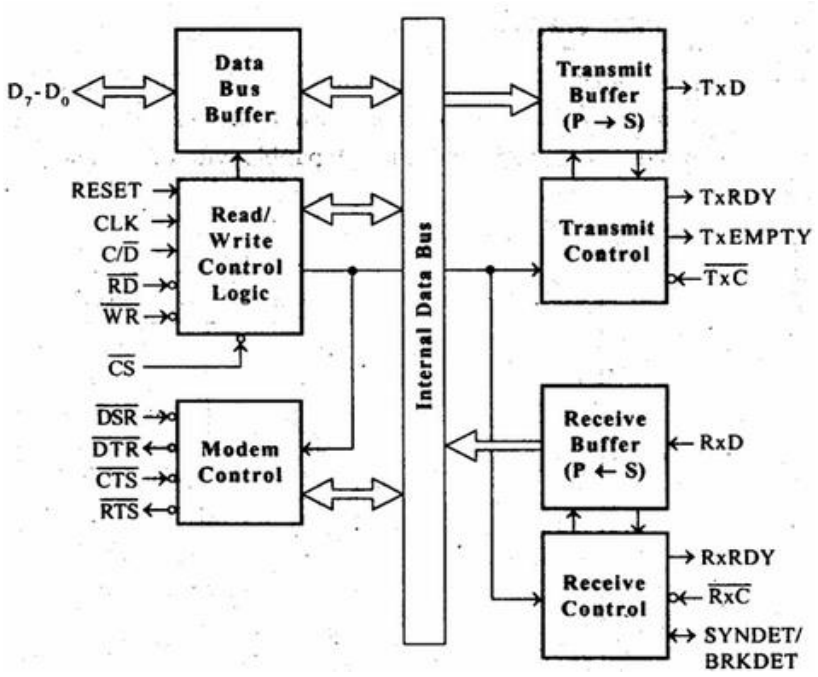


Figure 7-7: Block Diagram of 8251

Read/Write control logic:

- The Read/Write Control logic interfaces the 8251A with CPU, determines the functions of the 8251A according to the control word written into its control register.
- It monitors the data flow.
- This section has three registers and they are control register, status register and data buffer.
- The active low signals RD^{-}, WR^{-}, CS^{-} and C/D^{-} are used for read/write operations with these three registers.

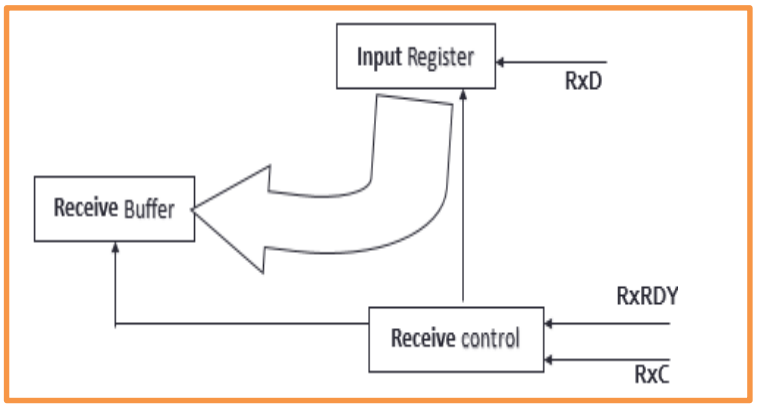
- When C/D¯ is high, the control register is selected for writing control word or reading status word.
- When C/D¯ is low, the data buffer is selected for read/write operation.
- When the reset is high, it forces 8251A into the idle mode.
- The clock input is necessary for 8251A for communication with CPU and this clock does not control either the serial transmission or the reception rate.

Transmitter section:

- The transmitter section accepts parallel data from CPU and converts them into serial data.
- The transmitter section is double buffered, i.e., it has a buffer register to hold an 8-bit parallel data and another register called output register to convert the parallel data into serial bits.
- When output register is empty, the data is transferred from buffer to output register. Now the processor can again load another data in buffer register.
- If buffer register is empty, then TxRDY is goes to high.
- If output register is empty then TxEMPTY goes to high.
- The clock signal, TxC¯ controls the rate at which the bits are transmitted by the USART.
- The clock frequency can be 1, 16 or 64 times the baud rate.

Receiver Section:

- Accepts serial data on the RxD pin and converts them to parallel data.
- Has two registers :
  - Receiver input register
  - Buffer register



- When RxD goes low, the control logic assumes it is a start bit, waits for half bit time, and samples the line again. If the line is still low, the input register accepts the following data, and loads it into buffer register at the rate determined by the receiver clock.

- RxRDY - Receiver Ready Output: Output signal, goes high when the USART has a character in the buffer register & is ready to transfer it to the MPU.
- RxD - Receive Data Input : Bits are received serially on this line & converted into a parallel byte in the receiver input register.
- RxC - Receiver Clock Input : Clock signal that controls the rate at which bits are received by the USART.

MODEM Control:

- The MODEM control unit allows to interface a MODEM to 8251A and to establish data communication through MODEM over telephone lines.
- This unit takes care of handshake signals for MODEM interface.

7.8.Introduction to PPI device 8255A

Intel 8255 is a programmable peripheral interface (PPI) chip. It means that it is a programmable chip used for interfacing or connecting peripheral devices. Peripheral device is another name for I/O device. Also, it is well known that I/O ports are used for connecting I/O devices. Thus, in very simple words, 8255 is a programmable I/O port chip.

8255 is a general purpose programmable I/O device used for parallel data transfer. It is 40 pin IC which operate on +5V power supply. It is designed to work in two mode, i.e. IO modes and bit set reset (BSR) modes. It consists three 8-bit IO ports named as Port A, Port B, Port C. Port A and B always behave as 8-bit ports, whereas, port C can be used as a single 8 bit port, or as two 4-bit IO port or we can use the individual pins of port C as individual lines. These three ports are divided into two groups, i.e. group A and group B. Group A consists of port A and port C upper (PC_U), whereas, group B consists of port B and port C lower (PC_L). Here, port C upper (PCU) means the signals PC4, PC5. PC6, and PC7 and port C lower (PCL) means the pins PC0, PC1, PC2, and PC3. In which mode these ports operate, is decided by the D7 bit of the control word register (CWR). If the D7 bit of CWR is 0 then they will operate in BSR mode and if the D7 bit is 1 then they will operate in IO mode. The IO mode further classified as:

Mode 0- Basic or simple input/output (Group A, Group B)

Mode 1- Strobe input/output (Group A, Group B)

Mode 2 – Bidirectional bus (Port A only)

Figure 6-1 shows the basic modes of 8255.

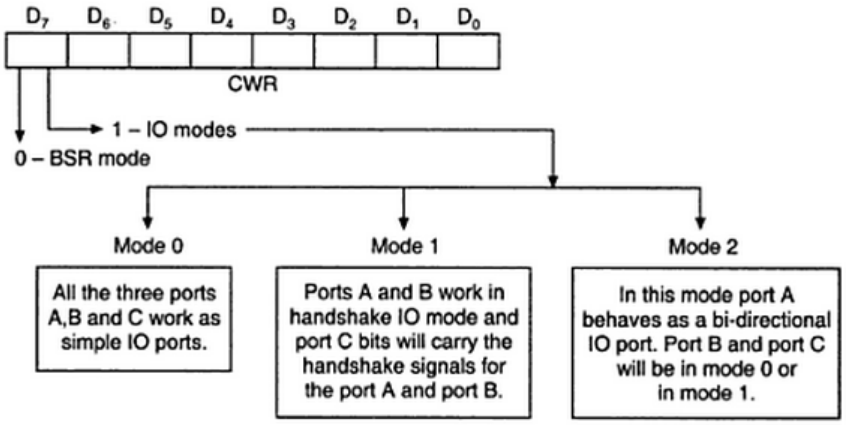


Figure 6-1 Basic modes of 8255

7.9.Pin Description of 8255

Pin	Description
D_0 - D_7	Data lines
RESET	Reset input
$\overline{CS}$	Chip select
$\overline{RD}$	Read control
$\overline{WR}$	Write control
A_0, A_1	Internal address
PA_7 - PA_0	Port-A pins
PB_7 - PB_0	Port-B pins
PC_7 - PC_0	Port-C pins
V_{CC}	+5V
V_{SS}	0V (GND)

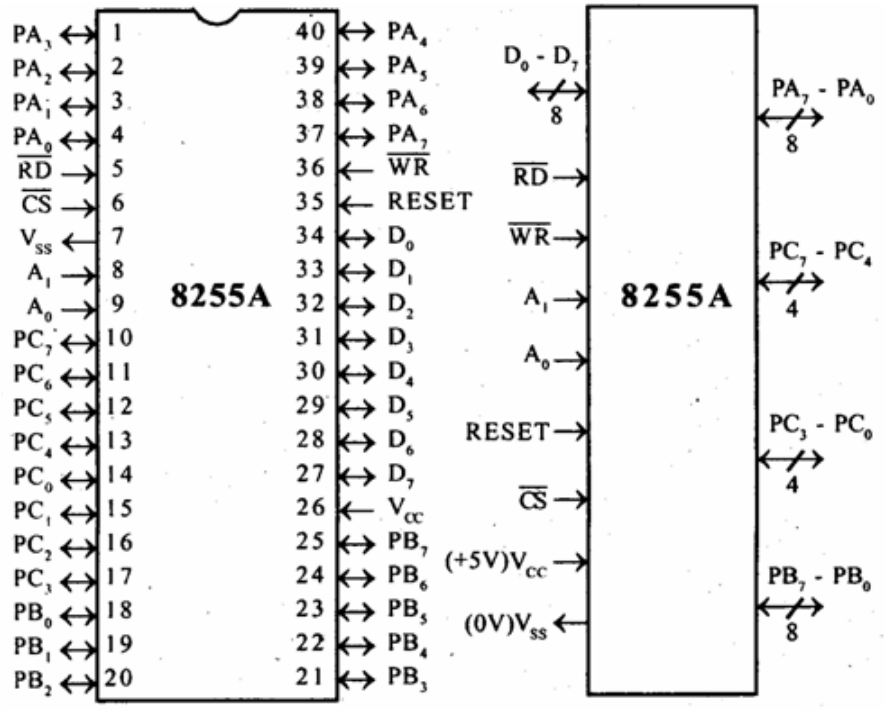


Figure 6-2: Pin diagram of 8255H

Functions of Pins

Data bus (D0-D7): These are 8-bit bi-directional buses, connected to 8085 data bus for transferring data.

CS: This is Active Low signal. When it is low, then data is transfer from 8085.

RD^{-}: This is Active Low signal, when it is Low read operation will be start.

WR^{-}: This is Active Low signal, when it is Low Write operation will be start.

Address (A0-A1): This is used to select the ports as shown:

A1	A0	SELECTION
0	0	PORT A
0	1	PORT B
1	0	PORT C
1	1	CONTROL REGISTER

RESET: This is used to reset the device. That means clear control registers.

PA0-PA7: It is the 8-bit bi-directional I/O pins used to send the data to peripheral or to receive the data from peripheral.

PB0-PB7: Similar to PA

PC0-PC7: This is also 8-bit bidirectional I/O pins. These lines are divided into two groups.

- ✓ PC0 to PC3 (Lower Groups)
- ✓ PC4 to PC7 (Higher groups)

These two groups working in separately using 4 data's.

7.10. Block Diagram of the 8255:

The block diagram of 8255 consists of mainly four parts as shown in fig. 6-3. These are:

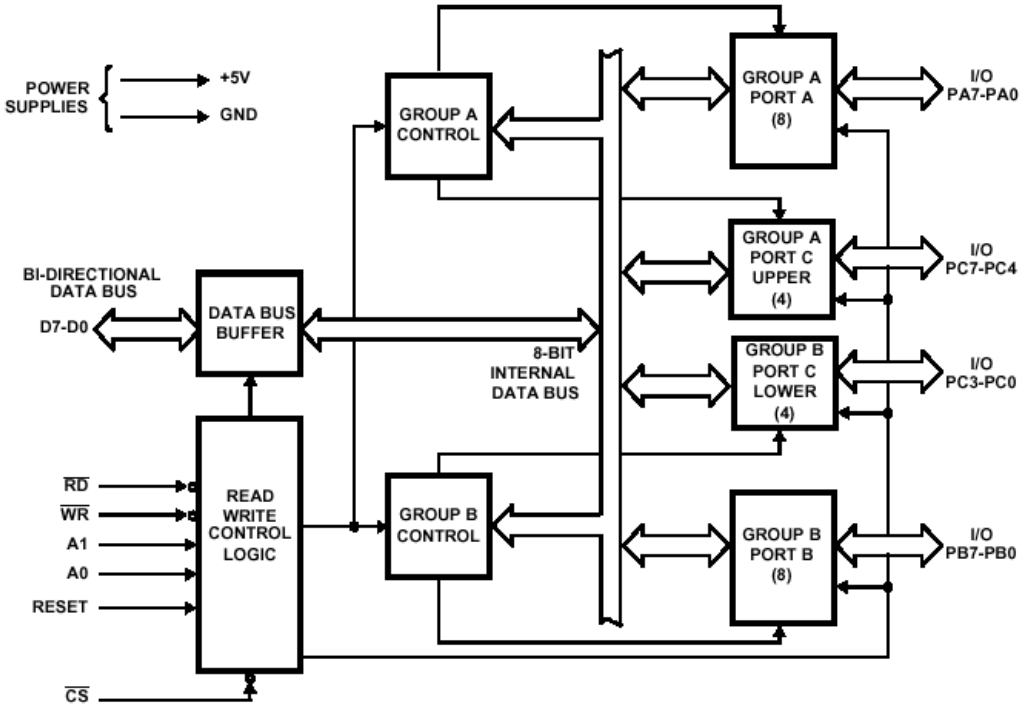


Figure 6-3: 8255 Block Diagram

1. Data bus buffer

2. Read/Write control logic

3. IO ports

4. Group A and group B control logic

1. Data bus buffer:

This three-state bi-directional 8-bit buffer is used to interface the 8255 to the system data bus. Data is transmitted or received by the buffer upon execution of input or output instructions by the CPU. Control words and status information are also transferred through the data bus buffer.

2. Read/Write control logic:

The function of this block is to manage all of the internal and external transfers of both Data and Control or Status words. It accepts inputs from the CPU Address and Control busses and in turn, issues commands to both of the Control Groups. It consists of inputs $\overline{RD}$, $\overline{WR}$, A0, A1, and $\overline{CS}$

$\overline{CS}$ (Chip Select): A "low" on this input pin enables the communication between the 8255 and the CPU.

$\overline{RD}$ (Read): A “low” on this input pin enables 8255 to send the data or status information to the CPU on the data bus. In essence, it allows the CPU to “read from” the 8255.

$\overline{WR}$ (Write): A “low” on this input pin enables the CPU to write data or control words into the 8255.

A0 and A1 (Port Select 0 and Port Select 1): These input signals, in conjunction with the $\overline{RD}$ and $\overline{WR}$ inputs, control the selection of one of the three ports or the control word register. They are normally connected to the least significant bits of the address bus (A0 and A1). Table 6-1 gives the basic operation

(RESET) Reset. A “high” on this input initializes the control register to 9Bh and all ports (A, B, C) are set to the input mode.

Table 6-1: IO Operations Definitions

A1	A0	$\overline{RD}$	$\overline{WR}$	$\overline{CS}$	Operations
0	0	0	1	0	Port A to Data bus(Read Operation)
0	1	0	1	0	Port B to Data bus(Read Operation)
1	0	0	1	0	Port C to Data bus(Read Operation)
0	0	1	0	0	Data bus to Port A (Write operation)
0	1	1	0	0	Data bus to Port B (Write operation)
1	0	1	0	0	Data bus to Port C (Write operation)
1	1	1	1	0	Data bus to control register

3. IO PORTS

The 8255 contains three 8-bit ports (A, B, and C). All can be configured to a wide variety of functional characteristics by the system software but each has its own special. All these ports are bidirectional in nature which are having internal output latch/buffers and input buffers.

Port A: One 8-bit data output latch/buffer and one 8-bit data input latch.

Port B: One 8-bit data input/output latch/buffer and one 8-bit data input buffer.

Port C: One 8-bit data output latch/buffer and one 8-bit data input buffer (no latch for input). This port can be divided into two 4-bit ports under the mode control. Each 4-bit port contains a 4-bit latch and it can be used for the control signal output and status signal inputs in conjunction with ports A and B.

4. Group A and Group B control logic

The functional configuration of each port is programmed by the systems software. In essence, the CPU "outputs" a control word to the 8255. The control word contains information such as "mode", "bit set", "bit reset", etc., that initializes the functional configuration of the 8255. Each of the Control blocks (Group A and Group B) accepts "commands" from the Read/Write Control logic, receives "control words" from the internal data bus and issues the proper commands to its associated ports.

7.11. Control Word

To specify the function of each port, a content is to be stored on a special register called control register and the content is called a control word. Bit D7 of the control register specifies either the I/O function or the bit set/reset function. If D7= 0, port C operates on bit set/reset (BSR) mode If D7=1, then bits D6-D0 determine I/O function in various modes. The BSR control word does not affect the functions of Port A and Port B.

To communicate with peripherals through the 8255, three steps are necessary:

1. Determine the address of ports A, B and C and of the control register according to the chip select logic and address lines A0 andA1.

2. Write the control word in the control register.

3. Write I/O instructions to communicate with peripherals through Ports A, B and C.

7.12. Operation modes

There are two basic operational modes of 8255:

1. Bit set/reset Mode (BSR Mode).

2. Input/Output Mode (I/O Mode).

The two modes are selected on the basis of the value present at the D7 bit of the Control Word Register. When D7 = 1, 8255 operates in I/O mode and when D7 = 0, it operates in the BSR mode.

Bit set/reset Mode (BSR Mode):

The Bit Set/Reset (BSR) mode is applicable to port C only. Each line of port C (PC0 - PC7) can be set/reset by suitably loading the control word register.

- D_7 bit is always 0 for BSR mode.
- Bits D_6, D_5 and D_4 are don't care bits.
- Bits D_3, D_2 and D_1 are used to select the pin of Port C.
- Bit D_0 is used to set/reset the selected pin of Port C.

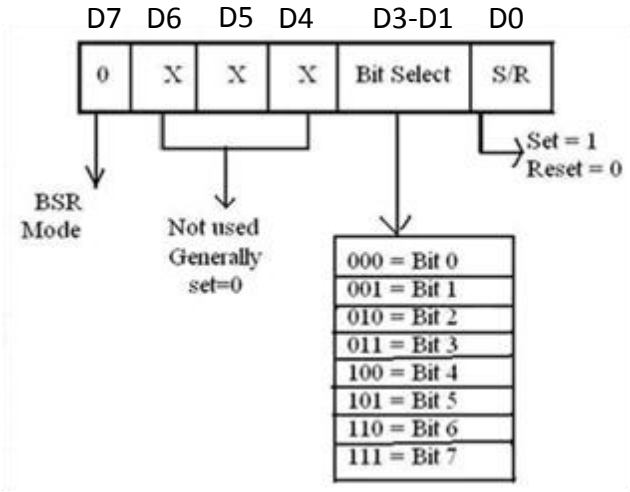


Figure 6-4: Control word format in the BSR mode

- As an example, if it is needed that PC_5 be set, then in the control word,
- Since it is BSR mode, **D_7 = '0'**.
- Since D_4, D_5, D_6 are not used, assume them to be '**0**'.
- PC_5 has to be selected, hence, **D_3 = '1', D_2 = '0', D_1 = '1'**.
- PC_5 has to be set, hence, **D0 = '1'**.
- Thus, as per the above values, 0B (Hex) will be loaded into the Control Word Register (CWR).

D7	D6	D5	D4	D3	D2	D1	D0
0	0	0	0	1	0	1	1

I/O mode:

This mode is selected when D7 bit of the Control Word Register is 1. There are three I/O modes:

1. Mode 0 - Simple I/O

2. Mode 1 - Strobed I/O

3. Mode 2 - Strobed Bi-directional I/O

The control word format for the I/O mode is as shown:

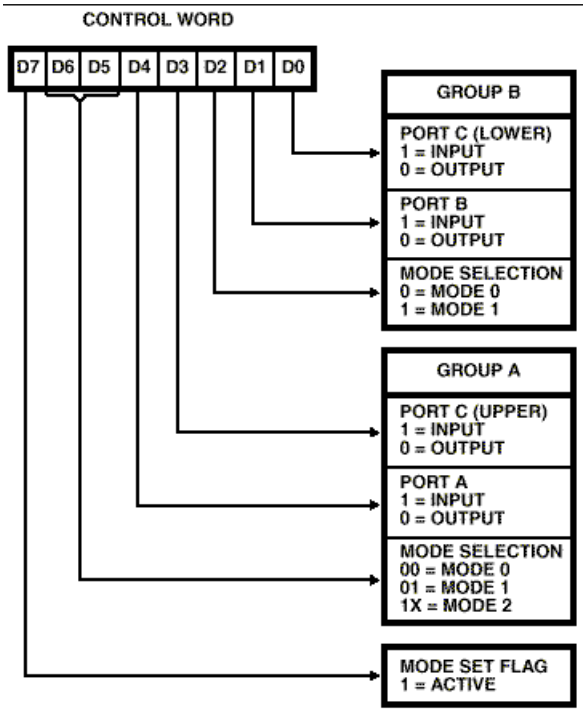


Figure 6-5: Control word format for I/O mode

Example1. Find the control word, if port B and upper port C have to be initialized as input ports and lower port C and port A as output ports (all in mode 0):

1. Since it is an I/O mode, D7 = 1.

2. Mode selection bits, D2, D5, D6 are all 0 for mode 0 operation.

3. Port B and upper port C should operate as Input ports, hence, D1 = D3 = 1.

4. Port A and lower port C should operate as Output ports, hence, D4 = D0 = 0.

Hence, for the desired operation, the control word register will have to be loaded with 8A (hex).

Example 2: Find the control word of the 8255 for the following configurations:

(a)All the ports of A, B, and C are output ports (mode 0).

(b) PA = in, PB = out, PCL = out, and PCH = out.

Solution: From Figure 6-5 we have:

(a) 1000 0000 = 80H (b) 1001 0000 = 90H

1. Mode 0: Simple Input /Output: In this mode, ports A, B are used as two simple 8-bit I/O ports and port C as two 4-bit ports. Each port can be programmed to function as simply an input port or an output port. The input/output features in Mode 0 are as follows.

1. Outputs are latched.

2. Inputs are not latched.

3. Ports don't have handshake or interrupt capability.

Question : Suppose 8255A is operating in mode 0. Write a program to accept the data input from port B and port C lower. Output the same read data to port A and port C upper respectively.

Solution: Steps:

1. Determine address of port A, port B, port C and control register

2. Write the control word in the control register.

3. Write I/O instructions (program) to communicate with peripherals.

Step1: *Determining address*

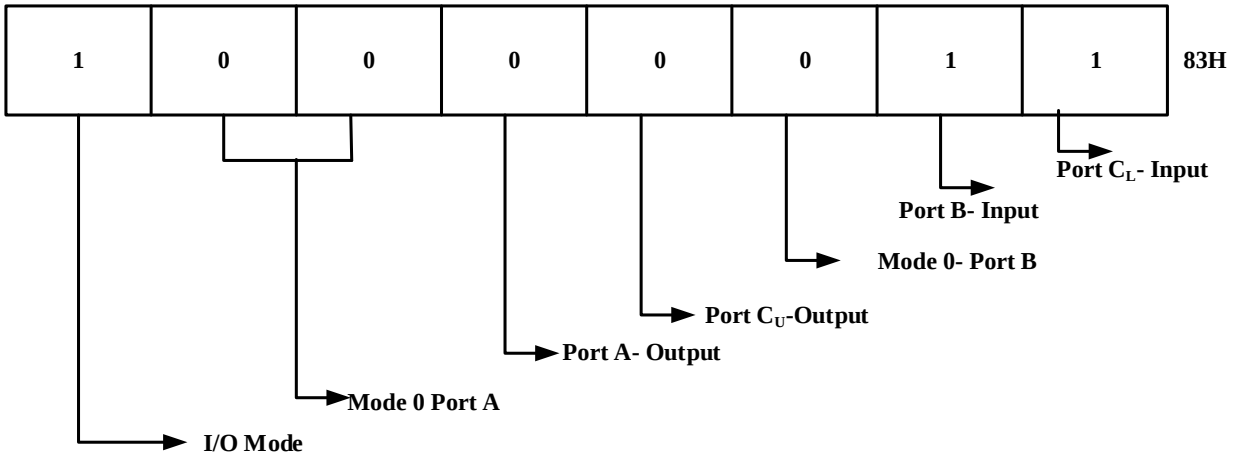
For the memory mapped I/O (16 bit address) below table shows the corresponding address

Memory Mapped I/O

Ports/ Control	Address Lines																Address
Register	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0	
Port A	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	8000H
Port B	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	8001H
Port C	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	8002H
Control Word	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	8003H

Step 2: Determining control word

Control word: 83H



Step 3: Writing Program

```
MVI A, 83H      ; loading the control word on accumulator.
STA 8003H       ; write control word on control register.
LDA 8001H       ; Reading from port B (8-bit data)
STA 8000H       ; Displaying through Port A.
LDA 8002H       ; Reading from port C (Clower - 4 bit data)
ANI 0FH         ; Masking upper 4-bits of port C
RLC             ; Shifting port Clower to port Cupper
RLC
RLC
RLC
STA 8002        ; Displaying through port Cupper
```

Example 2: Write initialization instruction to set up port A and port C_{upper} as an input port, Port B and port C_{lower} as an output port. Use I/O mapped I/O.

Solution:

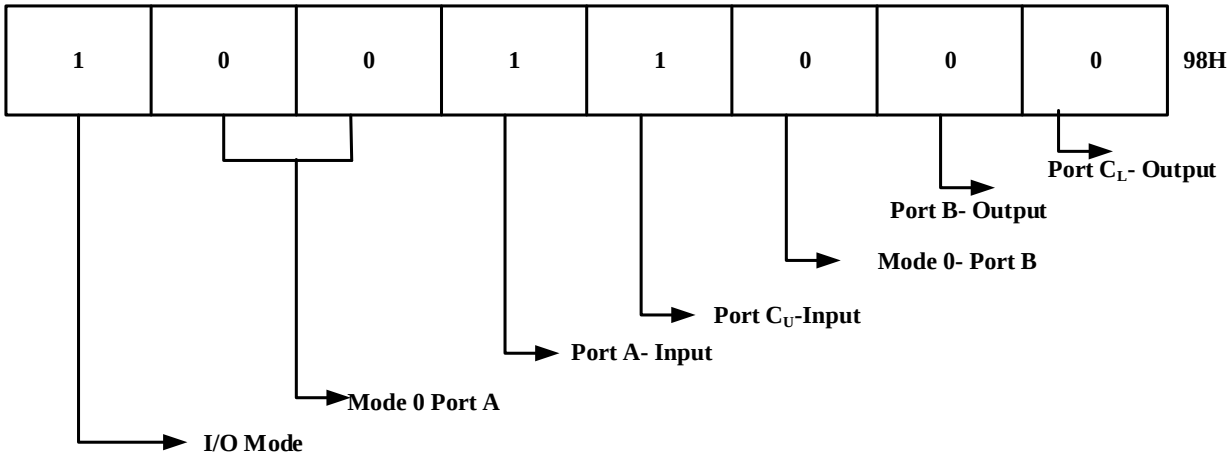
Step 1: Determining address

For the I/O mapped I/O (8 bit address) below table shows the corresponding address

I/O Mapped I/O

Ports/ Control	Address Lines								Address
Register	A7	A6	A5	A4	A3	A2	A1	A0	
Port A	1	0	0	0	0	0	0	0	80H
Port B	1	0	0	0	0	0	0	1	81H
Port C	1	0	0	0	0	0	1	0	82H
Control Word	1	0	0	0	0	0	1	1	83H

Step 2: Determining control word



Step 3: Writing Program

```
MVI A, 98H      ; Loading control word to the accumulator
OUT 83H         ; Writing control word on control register
IN 80H          ; Reading from Port A
OUT 81H         ; Displaying through Port B
IN 82H          ; Reading from Port Cupper
ANI F0H         ; Masking Clower
RAR             ; Shifting Cupper to Clower
RAR
RAR
RAR
OUT 82H         ; Displaying through port Clower
```

2. Mode 1: Input /Output with Handshake

In this mode, handshake signals are exchanged between the MPU and peripherals prior to data transfer. The features of the mode include the following:

1. Two ports (A and B) function as 8-bit I/O ports. They can be configured as either as input or output ports.

2. Each port uses three lines from ort C as handshake signals. The remaining two lines of Port C can be used for simple I/O operations.

3. Input and Output data are latched.

4. Interrupt logic is supported.

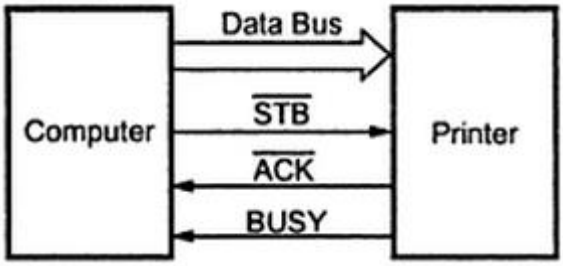


Figure 6-6: Data transfer between computer and printer using handshaking signal

For example, computer can send data to the printer with large speed but printer can't accept data and print data with this rate. So the computer has to send data with the speed with which printer can accept. This type of data transfer is achieved by using handshaking signal along with data signal. Fig. 6-6 shows the data transfer between computer and printer using handshaking signals. These handshaking signals are used to tell computer whether printer is ready to accept the data or not.

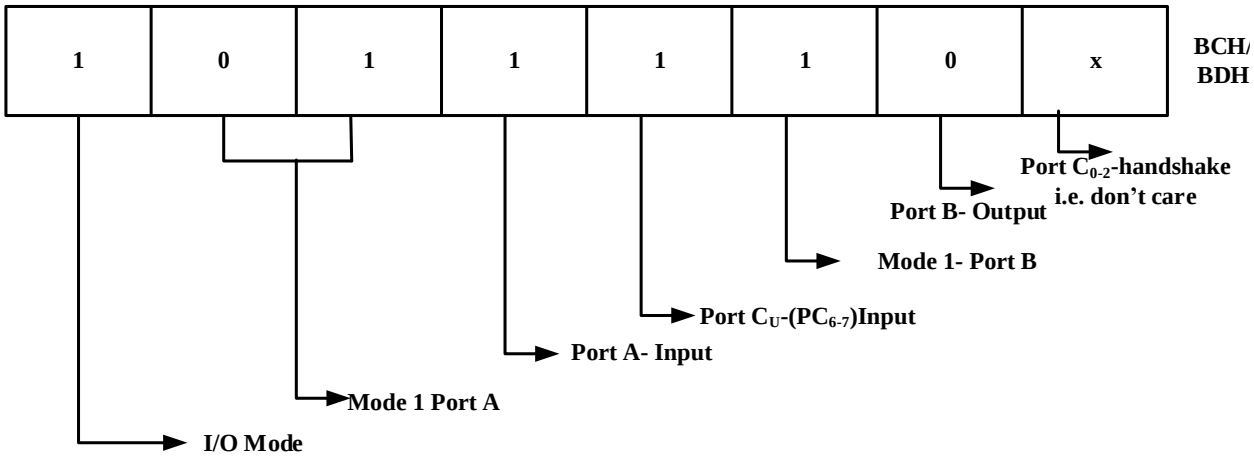
If the printer is ready to accept the data then after sending data on data bus, computer uses another handshaking signal ($\overline{STB}$) to tell printer that valid data is available on the data bus.

Example: Configure port A as input in mode 1. Port B as output in mode 1. Port C_{6-7} as input port.

Solution:

When port B is to be programmed as an input or output port, PC0, PC1, and PC2 are used for control (handshaking signal).

Determining control word:



Therefore control word may either BCH or BDH.

3. Mode 2: Bidirectional Data Transfer

This mode is used primarily in applications such as data transfer between two computers. In this mode, Port A can be configured as the bidirectional port and Port B either in Mode 0 or Mode 1. Port A uses five signals from Port C as handshake signals for data transfer. The remaining three signals from port C can be used either as simple I/O or as handshake for port B.

The basic features of mode 2:

- Used only for Group A
- Port A is used to work in bidirectional data transfter
- Port C is used to exchange handshaking signals with I/O device
- Handshaking and interrupt logic is supported
- Port B can operate either mode 0 or mode 1

Example: Assume an 8255, whose control word register address is 83H, is such that the port A is to be operated in mode 2 and port B in mode 0 with PB7-PB0 as input and PC2-PC0 as outputs.

Solution: 1. Given, Control word address = 83H so,

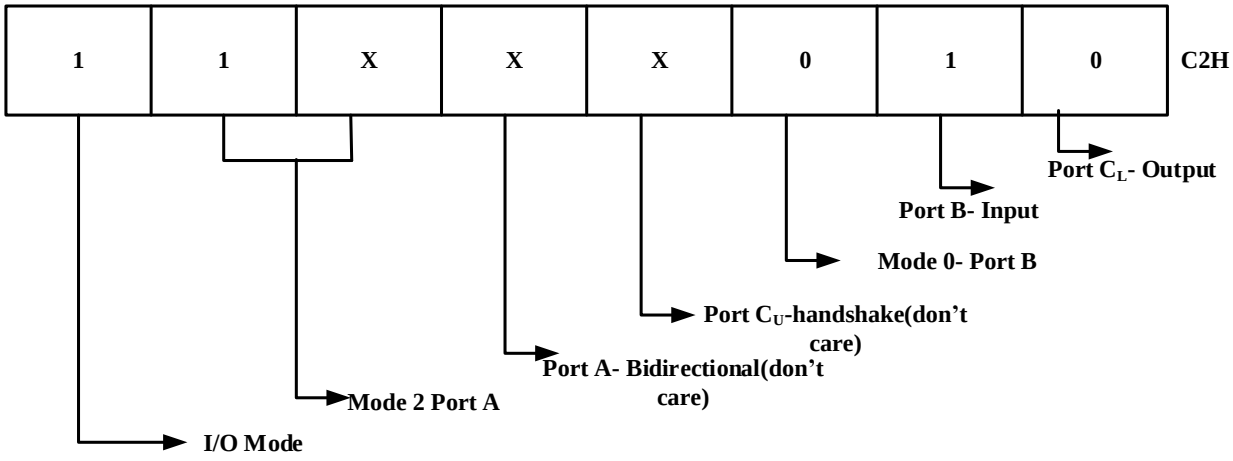
Port A address➔ 80H

Port B address ➔81H

Port C address ➔82H

2. Determination of control word:

Considering the don't care condition to be '0', hence the control word is C2H



3. Writing Program

```
MVI A, C2H
OUT 83H
IN 80H
OUT 80H
IN 81H
OUT 80H
ANI 07H
OUT 82H
HLT
```

7.13. Bus Interface Standards

The micro-computer is a bus oriented system where by subsystem or peripherals are interconnected through the bus. Since, peripherals and computers are manufactured by various manufacturers, there should exist a common understanding among various manufacturers and user that they are compatible. These understanding is called standards. In the field of electronics, these standards are generally defined by professional organizations such as IEEE (Institute of Electrical and Electronics Engineers), EIA (Electronics Industries association). There are many standards for data transfer (e.g. S-100, STD Bus, RS-232, GPIB etc.). Rs-232 is the following sections:

RS-232 Interface

When data are transmitted as voltage, the commonly used standard is known as RS-232. It is defined in reference to Data Terminal Equipment (DTE) and Data Communication Equipment (DCE)—terminal (computer) and modem as shown in fig 7.8; however, its voltage levels are not compatible with TTL logic levels. The rate of data transmission in RS-232 is restricted to a maximum of 20kbaud and the distance is limited to 50 ft. (15m).

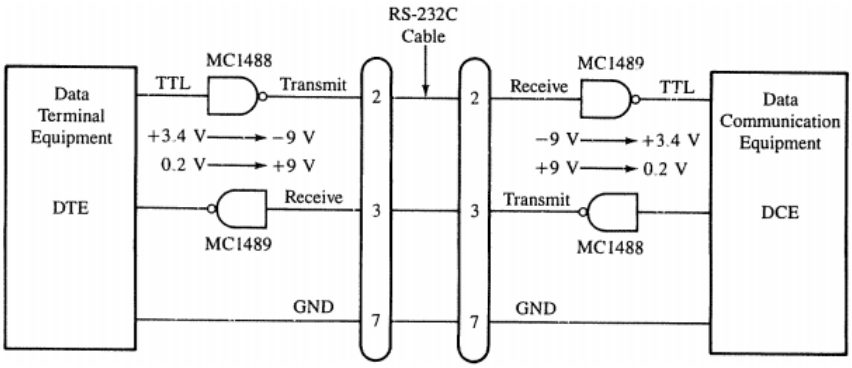


Figure 7-8: Minimum configuration of RS-232 signals and voltage levels

RS -232 works in negative logic. For data lines, the voltage level from +3 V to +15 V is defined as logic 0, and from -3 V to -15 V is defined as logic 1. Because of incompatibility with TTL logic, voltage translators, called line drivers and line receivers, are required to interface TTL logic with the RS-232 signals, as shown in fig. 7-8. The line driver, MC1488, converts logic 1 into approximately -9V and logic 0 into +9 V. Before the signal is received by the DCE, it is again converted by the line receiver, MC1489, into TTL compatible logic.

The minimum interface requires three lines: pin 2, 3, and 7 as shown in fig. 7-8. These lines are defined in relation to the DTE; the terminal transmits on pin 2 and receives on pin 3. On the other hand, the DCE transmits on pin 3 and receives on pin 2. Pin 7 is used for the ground pins.

Fig. 7-9 shows the RS-232 25 pins and associated signals. The signals are divided into four groups: data signals, control signals, timing signals, and grounds.

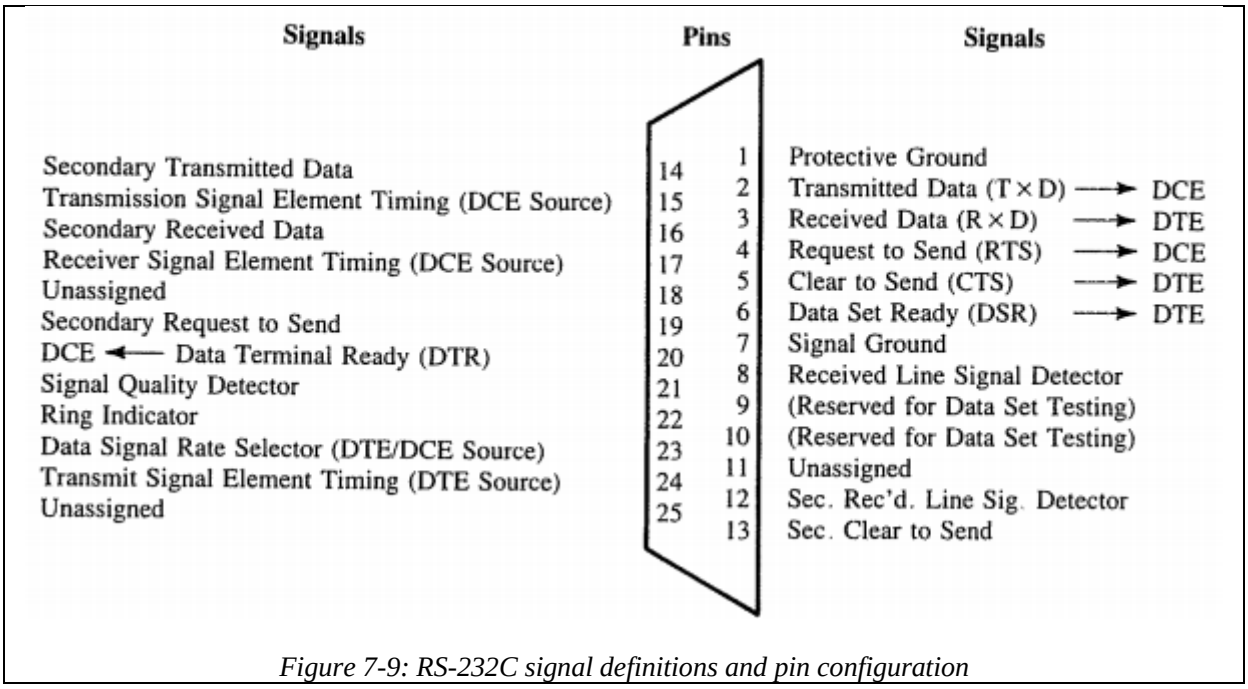


Figure 7-9: RS-232C signal definitions and pin configuration

7.14. Keyboard and Display Controller: Introduction to 8279

The INTEL 8279 is specially developed for interfacing keyboard and display devices to 8085/8086/8088 microprocessor based system. The important features of 8279 are,

- ✓ Simultaneous keyboard and display operations.
- ✓ Scanned keyboard mode.
- ✓ Scanned sensor mode.
- ✓ 8-character keyboard FIFO.
- ✓ 1 6-character display.
- ✓ Right or left entry 1 6-byte display RAM.
- ✓ Programmable scan timing.

Block diagram of 8279

The functional block diagram of 8279 is shown.

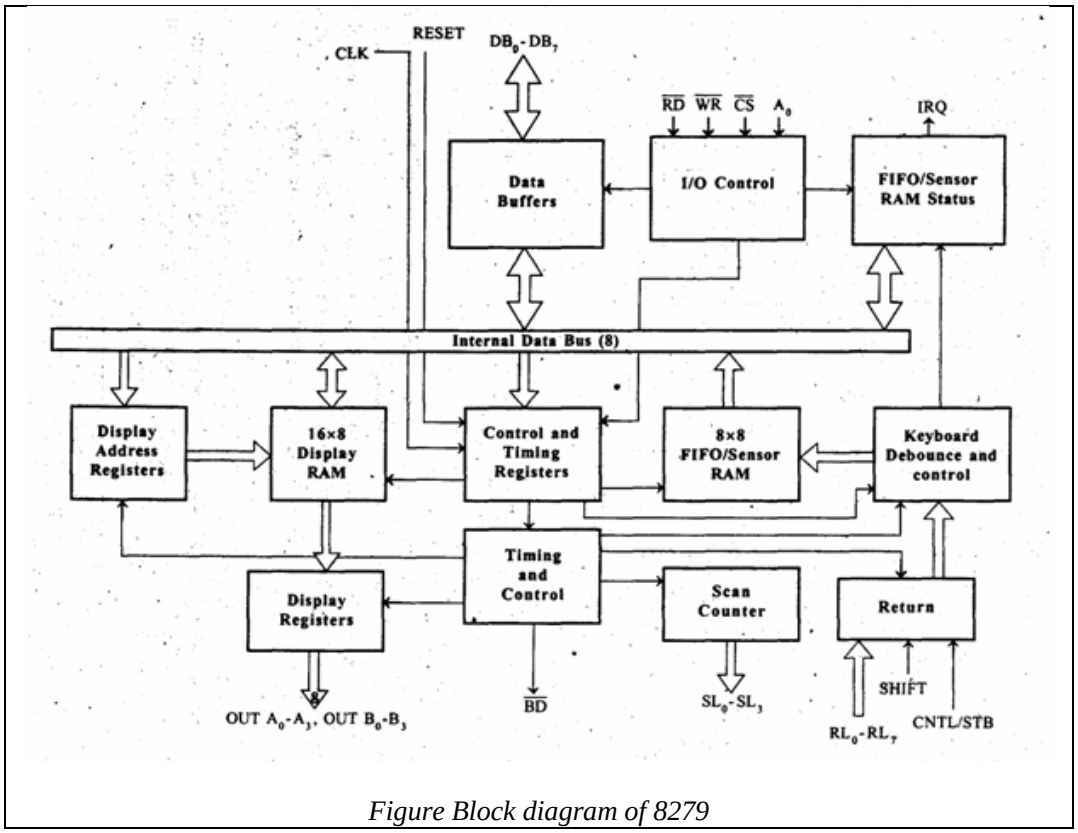


Figure Block diagram of 8279

The four major sections of 8279 are keyboard, scan, display and CPU interface.

Keyboard section:

- The keyboard section consists of eight return lines RL0 - RL7 that can be used to form the columns of a keyboard matrix.
- It has two additional input: shift and control/strobe. The keys are automatically denounced.
- The two operating modes of keyboard section are 2-key lockout and N-key rollover.
- In the 2-key lockout mode, if two keys are pressed simultaneously, only the first key is recognized.
- In the N-key rollover mode simultaneous keys are recognized and their codes are stored in FIFO.
- The keyboard section also have an 8 x 8 FIFO (First In First Out) RAM.
- The FIFO can store eight key codes in the scan keyboard mode. The status of the shift key and control key are also stored along with key code. The 8279 generate an interrupt signal when there is an entry in FIFO. The format of key code entry in FIFO for scan keyboard mode is,

Display section:

- The display section has eight output lines divided into two groups A0-A3 and B0-B3.
- The output lines can be used either as a single group of eight lines or as two groups of four lines, in conjunction with the scan lines for a multiplexed display.
- The output lines are connected to the anodes through driver transistor in case of common cathode 7-segment LEDs.
- The cathodes are connected to scan lines through driver transistors.
- The display can be blanked by BD (low) line.
- The display section consists of 16 x 8 display RAM. The CPU can read from or write into any location of the display RAM.

Scan section:

- The scan section has a scan counter and four scan lines, SL0 to SL3.
- In decoded scan mode, the output of scan lines will be similar to a 2-to-4 decoder.
- In encoded scan mode, the output of scan lines will be binary count, and so an external decoder should be used to convert the binary count to decoded output.
- The scan lines are common for keyboard and display.
- The scan lines are used to form the rows of a matrix keyboard and also connected to digit drivers of a multiplexed display, to turn ON/OFF.

CPU interface section:

- The CPU interface section takes care of data transfer between 8279 and the processor.
- This section has eight bidirectional data lines DB0 to DB7 for data transfer between 8279 and CPU.
- It requires two internal address A =0 for selecting data buffer and A = 1 for selecting control register of8279.
- The control signals WR (low), RD (low), CS (low) and A0 are used for read/write to 8279.
- It has an interrupt request line IRQ, for interrupt driven data transfer with processor.

- The 8279 require an internal clock frequency of 100 kHz. This can be obtained by dividing the input clock by an internal prescaler.
- The RESET signal sets the 8279 in 16-character display with two -key lockout keyboard modes.