Chapter – 2
Parallel Interfacing with Microprocessor Based System

The device which can handle data at higher speed cannot support with serial interface. N bits of data are handled simultaneously by the bus and the links to the device directly. Achieves faster communication but becomes expensive due to need of multiple wires.

2.1 Methods of Parallel Data Transfer: Simple Input and Output, Strobe I/O, Single Handshake I/O, & Double Handshake I/O

Parallel transmission of data is used for short distance where the speed of information transfer is critical. This form of data communication is found in newer type of computer peripheral equipment with transfer speed of to one million characters per second. The equipment includes printers, disk drives and various other forms of peripheral components.

The information exchanged between a microprocessor and an I/O interface circuit consists of input or output data and control information. The status information enable the microprocessor monitor the device and when it is ready then send or receive data. Control information is the command by microprocessor to cause I/O device to take some action. If the device operates at different speeds, then microprocessor can be used to select a particular speed of operation of the device. The techniques used to transfer data between different speed devices and computer is called synchronizing. There are various ways of synchronization techniques which are involved in parallel data transfer such as simple input and output, simple strobe I/O, single handshaking and double handshaking.

Simple I/O

To get digital data from a simple switch into a microprocessor; switch is connected on input port line from which port can be read. The data is always present and ready so that it can be read at any time. Similarly to output data to a simple display device like LED, the input of LED buffer is connected on an output port pin. And output the logic level required turning on the light. The LED is always there and ready so that data can be sent at any time.

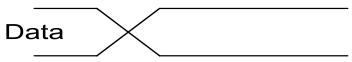


Fig: Simple I /O

This timing waveform illustrates the simple I/O where cross lines represent the time at which a new data byte becomes valid on the output lines of the port. Absences of other waveforms indicate 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 and must be read in at that time. Here a strobe pulse is supplied to indicate the time at which data is being transmitted. For an example, we can discuss the ASCII encoded keyboard. When a key is pressed, circuitry on keyboard sends out ASCII code for 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 eight data lines

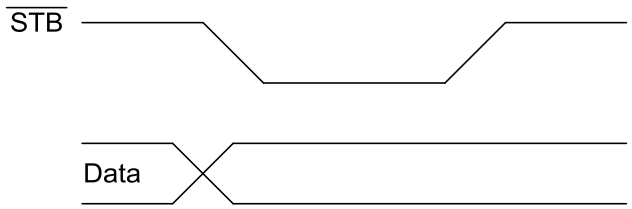


Fig: Simple Strobe I/O

The sending device outputs parallel data on the data lines, and then outputs STB' signal to represent the valid data is present.

In this technique, microprocessors need to wait until the device is ready for the operation and also known as simple wait I/O. Consider a simple keyboard consisting of 8 switches connected to a microprocessor through a parallel interface circuit (Tri-state buffer). The switch is of dip switches. In order to use this keyboard as an input device the microprocessor should be able to detect that a key has been activated. This can be done by observing that all the bits are in required order. The processor should repeatedly read the state of input port until it finds the right order of bits i.e. at least 1 bit of 8 bits should be 0.

Consider the tri-state A/D converter:

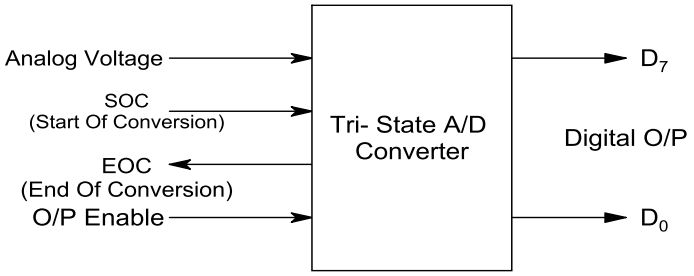


Fig: Tri-State A/D Converter

- Used to convert analog to digital data which can be read by I/O unit of microprocessor.
- When SOC appears 1, I/O unit should ready for reading binary data/digital data.
- When EOC's status is 1, then I/O unit should stop to read data.
- Strobe signal indicates the time at which data is being activated to transmit.

Single Handshaking

Handshaking is the method of synchronizing the actions of slow peripheral devices with that of high speed microprocessor. It can have two transfer schemes.

Input Handshake (Peripheral to Microprocessor):

The peripheral outputs some data and sends some strobe signal to microprocessor. Microprocessor detects asserted strobe signal (STB') and reads the byte of the data. Processor then sends acknowledgement signal (ACK) to peripheral to indicate that the data has been read and can send next byte of data.

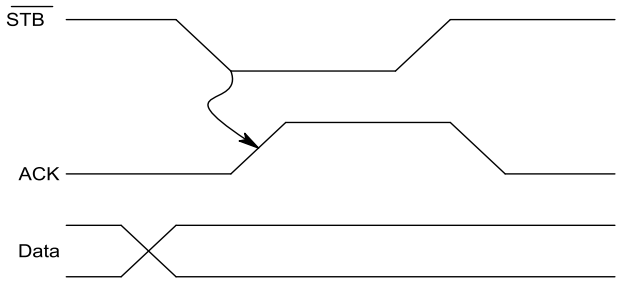


Fig: Single Handshaking

- The peripheral outputs some data and send $\overline{STB}$ signal to microprocessor to tell "Here is the data for you".
- Microprocessor detects asserted $\overline{STB}$ signal, reads the data and sends an acknowledge signal (ACK) to indicate data has been read and peripheral can send next data, "I got that one, send me another".
- Microprocessor sends or receives data when peripheral is ready.

Output Handshake (Peripheral from Microprocessor):

Microprocessor outputs data to peripheral and asserts a strobe (STB') signal. If peripheral is ready it answers back with acknowledgement (ACK) signal to microprocessor.

Double Handshaking

For data transfers where even more coordination is required between the sending system and the receiving system, a double handshake is used. It can have two transfer schemes.

Input Handshake (Peripheral to Microprocessor):

Peripheral asserts strobe (STB') line low to ask receiving device whether it is ready or not for data reception. Receiving system raises its acknowledgement (ACK) line high to indicate it is ready. Peripheral device then sends the byte of data and raises its strobe (STB') line high. When microprocessor reads data, it drops its acknowledgement (ACK) line low and request sending system to send net byte of data.

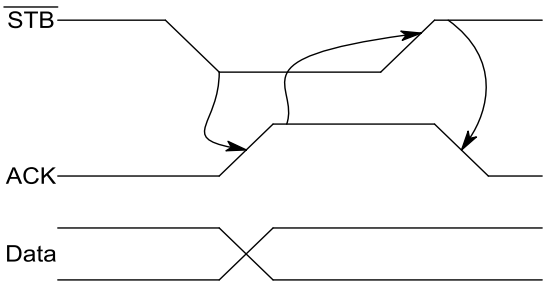


Fig: Double Handshaking

- The peripheral asserts its $\overline{STB}$ line low to ask microprocessor "Are you ready?"
- The microprocessor raises its ACK line high to say "I am ready".
- Peripheral then sends data and raises its $\overline{STB}$ line low to say "Here is some valid data for you."
- Microprocessor then reads the data and drops its ACK line to say, "I have the data, thank you, and I await your request to send the next byte of data."

Output Handshake (Peripheral from Microprocessor):

Microprocessor sends a strobe (STB') signal and data and peripheral sends acknowledgement (ACK) signal.

2.2 8255 as General Purpose Programmable I/O Device and its interfacing examples

The Intel 8255 A is a general purpose programmable I/O device designed for use with Intel microprocessors. It has 24 I/O pins that can be grouped primarily in two 8-bit parallel ports: A and B, with the remaining bits as port C. The 8-bits of port C can be used as individual bits or be grouped in two 4-bits ports: C upper (C_u) and C lower (C_l). The functions of these ports are defined by writing a control word in the control register.

8255 functions in two modes:

- **Bit Set/Reset mode:** The BSR mode is used to set or reset the bits in port C.
- **I/O mode:** The I/O mode is further divided into three modes: mode 0, mode 1 and mode 2. In mode 0, all ports function as simple I/O ports. Mode 1 is a handshake mode whereby ports A and/or B use bits from port C as handshake signals. In the handshake mode, two types of I/O data transfer can be implemented: status check and interrupt. In mode 2, port A can be set up for bidirectional data transfer using handshake signals from port C and port B can be set up either in mode 0 or mode 1.

Block diagram of 8255:

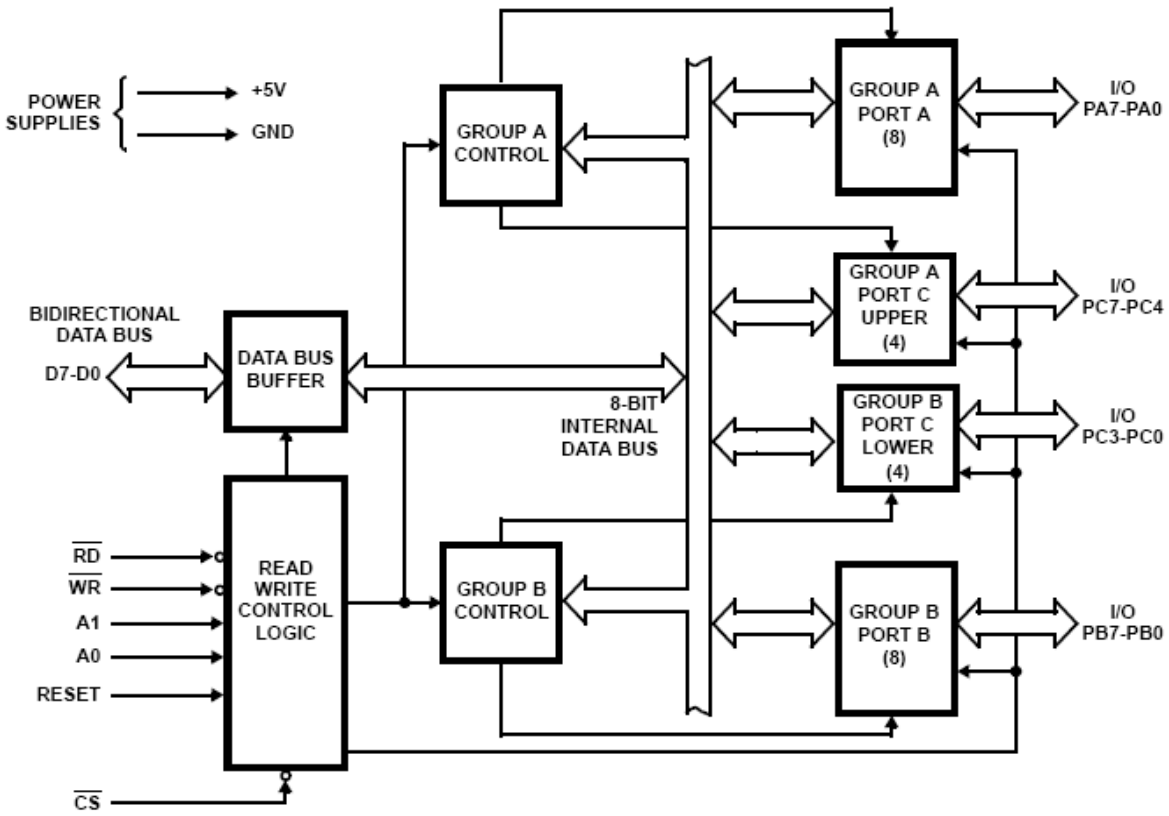


Fig2: Internal Block Diagram of 8255

The pin diagram and block diagram of 8255 is given above. It has the following main blocks.

a. Data Bus Buffer

The 3-state bidirectional 8-bit buffer is used to interface the 8255A 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.

b. Read/Write Control Logic

The function of the 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 buses and in turn, issues commands to both of the control groups.

- Chip Select (CS'): A "low" on this pin enables the communications between the 8255A and the CPU.
- Read (RD'): A "low" on this input enables the 8255A to send the data or status information to the CPU on the data bus. In essence, it allows the CPU to read from the 8255A.
- Write (WR'): A "low" on this input pin enables the CPU to write data or control words into the 8255A.
- Reset (RESET): A "high" to this pin clears the control register and sets all ports (A, B and C) in the input mode.
- A_0 and A_1: These input signals controls the selection of one of the three ports or the control word register. They are connected to the least significant bits of the address bus.

The CS' signal is the master chip select, and A_0 and A_1 specify one of the I/O ports or the control register as given below.

CS'	**A_1**	**A_0**	**Selected**
0	0	0	Port A
0	0	1	Port B
0	1	0	Port C
0	1	1	Control Register
1	X	X	8255A is not selected

c. Group A and Group B controls

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

- Control Group A – Port A and Port C Upper ($C_7 – C_4$)
- Control Group B – Port B and Port C Lower ($C_3 – C_0$)

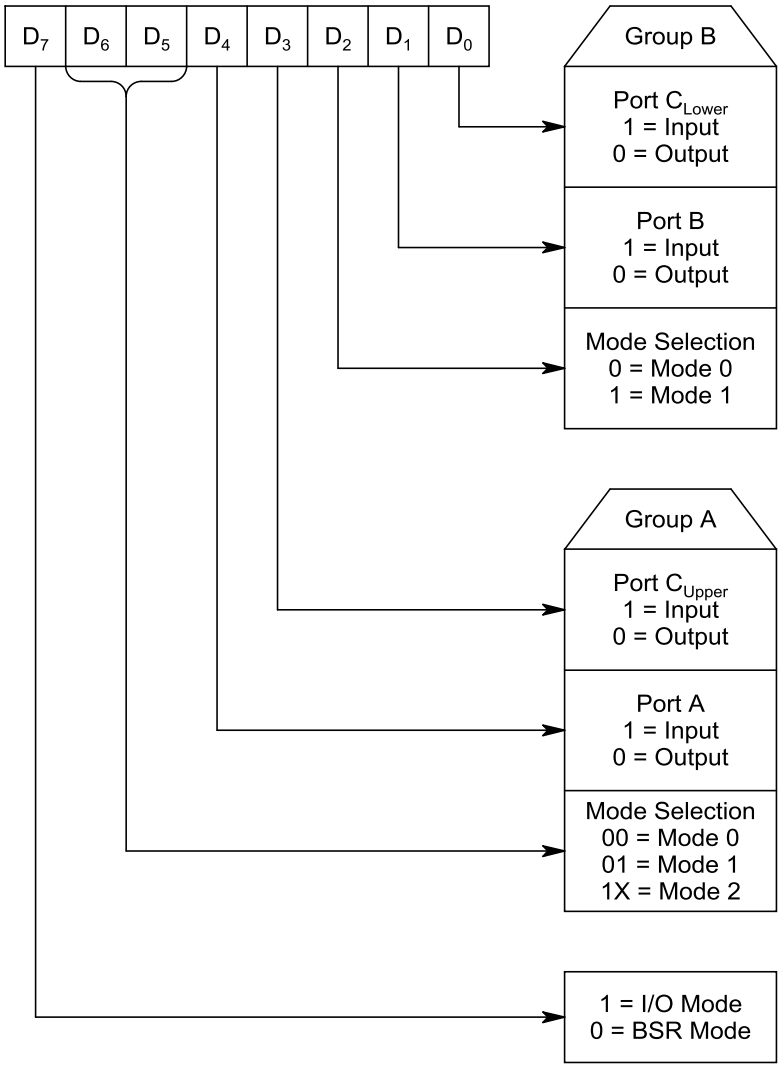
Control Word

When A_0 and A_1 pins have value 1, the mapped address addresses the control register which is the 8-bit register to write the specific content according to the port conditions although it cannot be read. The content of this register is called control word which specifies an I/O function for each port.

The MSB (D_7) of the control word tells which control word we are sending it that is it specifies either the I/O function or the Bit Set/Reset function. If bit D_7=1, bits D_6-D_0 determine I/O functions in various modes as shown in figure. If bit D_7=0, port C operates in the Bit Set/Reset (BSR) mode. The BSR control word does not affect the functions of ports A and B.

To communicate with peripherals through 8255, following are the steps are necessary.

- Determine the Port addresses of Ports A, B and C and of the control register according to Chip Select logic and address lines A_1 and A_0.
- Write a control word in control register.
- Write I/O instructions to communicate with peripherals through Ports A, B and C.



8255A Control Word Format for I/O Mode

I/O Control Word Examples

Q. Determine the Control word for the following configuration of ports of Intel 8255A PPI chip.

a. Port A output, mode of port A mode 1, port B output, mode of port B mode 0, port C lower pins as output and remaining pins of port C upper as output.

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
1	0	1	0	0	0	0	0

= A0H

b. Port A output, mode 0, port B output, mode 0, port C lower output and port C upper input.

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
1	0	0	0	1	0	0	0

= 88H

c. Port A input, mode 1, port B output, mode 1, and remaining pins of port C upper input.

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
1	0	1	1	1	1	0	X

= BCH [Normally don't care (X) = 0]

d. Port A input mode 1, port B output mode 0, port C lower input and port C upper output.

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
1	0	1	1	0	0	0	1

= B1H

e. Port A bidirectional (Mode 2), port B input mode 0, port C lower output.

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
1	1	X	X	X	0	1	0

= C2H [Normally don't care (X) = 0]

Operating Modes

Mode 0 (Basic Input/output)

This functional configuration provides simple input and output operation for each of the three ports. No 'handshaking" is required; data is simply written to or read from a specified port.

Mode 0 basic functional definitions:

- Two 8-bit ports and two 4-bit ports
- Any port can be input or output
- Outputs are latched
- Inputs are not latched
- 16 different input/output configurations are possible in this mode.

BSR Mode (Bit Set/Reset)

BSR mode is concerned only with eight bits of port C, which can be set or reset by writing an appropriate control word in the control register. A control word with bit D_7=0 is recognized as a control word and it does not alter any previously transmitted control word with bit D_7=1; thus the I/O operations of ports A and B are not affected by a BSR control word. In the BSR mode individual bits of port C can be used for applications such as On/Off switch.

BSR Control Word: This control word, when written in control register, sets or resets one bit at a time, as specified in figure.

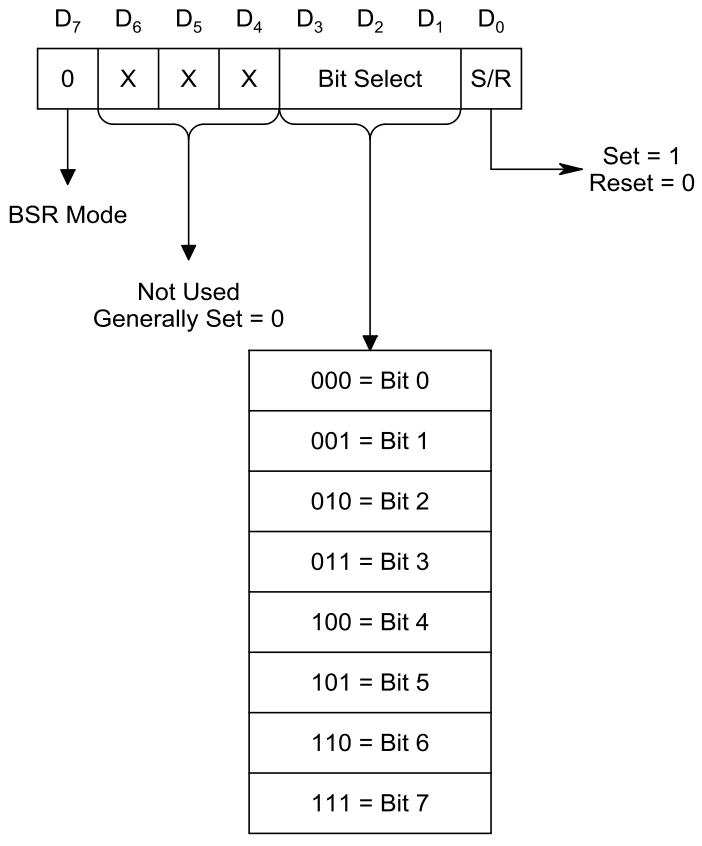


Fig: 8255A Control Word Format for BSR Mode

BSR Control Word Examples

Q. Determine the BSR Control word for the following Port C configurations.

a. Set PC_7

To set PC_7

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
0	X	X	X	1	1	1	1

= 0FH [Normally don't care (X) = 0]

b. Reset PC_3

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
0	X	X	X	0	1	1	0

= 06H [Normally don't care (X) = 0]

Mode 1 (Strobe Input/output)

The functional configuration provides a means for transferring I/O data to or from a specified port in conjunction with strobes or handshaking signals. In mode 1, port A and port B use the lines of port C to generate or accept these handshaking signals.

Mode 1 basic functional definitions:

- Two groups (Group A and Group B)
- Each group contains one 8-bit data port and one 4-bit control/data port
- The 8-bit data port can be either input or output. Both inputs and outputs are latched.
- The 4-bit port is used for control and status of the 8-bit data port.

Mode 2 (Strobe Bidirectional Bus I/O)

The functional configuration provides a means for communicating with a peripheral device or a structure on a single 8-bit bus for both transmitting and receiving data (bidirectional bus I/O). "Handshaking Signals" are provided to maintain proper bus flow discipline in a similar manner to Mode 1. Interrupt generation and enable/disable functions are also available.

Mode 2 basic functional definitions:

- Used in Group A only
- One 8-bit bidirectional bus port (Port A) and a 5-bit control port (Port C)
- Both inputs and outputs are latched

- The 5-bit control port (Port C) is used for control and status for the 8-bit, bidirectional bus port (Port A)

8255 Programming and Operation

A high on the RESET pin causes all 24 lines of the three 8-bit ports to be in the input mode. All flip-flops are cleared and the interrupts are reset. This condition is maintained even after the RESET goes low. The ports of the 8255 can then be programmed for any other mode by sending out a single output instruction to the control register. Also, the current mode of operation can be changed by writing a single mode word onto the control register, when required.

Modes for Group A and Group B can be separately defined with Port C taking on responsibilities as dictated by the mode definitions or Ports A and B. If Group A is programmed for Mode 0, and Group B is programmed for Mode 1, Port A and PC_4–PC_7 can be programmed for either input or output, while Port B can be programmed for input or output with PC_0–PC_2 used for handshaking.

The mode definition format and bit set-reset format are discussed in above topics. The control words for both mode definition and Bit Set-Reset are loaded into the same control register, with bit D_7 used for specifying whether the word loaded into the control register is a mode definition word or Bit Set-Reset word. If D_7 is high, the word is taken as a mode definition word, and if it is low, it is taken as a Bit Set-Reset word. The appropriate bits are set of reset depending on the type of operation desired, and loaded into the control register (which is accessed when A_1 and A_0 both are '1'; $\overline{WR}$ and $\overline{CS}$ both are '0'. It is to be noted that Group B does not have provision for operation in Mode 2.

The eight possible combinations of the states of bits D_1 -D_3 (B_2 B_1 B_0) in the Bit Set-Reset format (henceforth referred to as BSR) determine the particular bit in PC_0-PC_7 being set or reset as per the status of bit D_0. A BSR word is to be written for each bit that is to be set or reset. For example, if bit PC_2 is to be set and bit PC_7 is to be reset, the appropriate BSR words that will have to be loaded into the control register will be, 0XXX001 and 0XXX1110, respectively, where X can be either '0' or '1'.

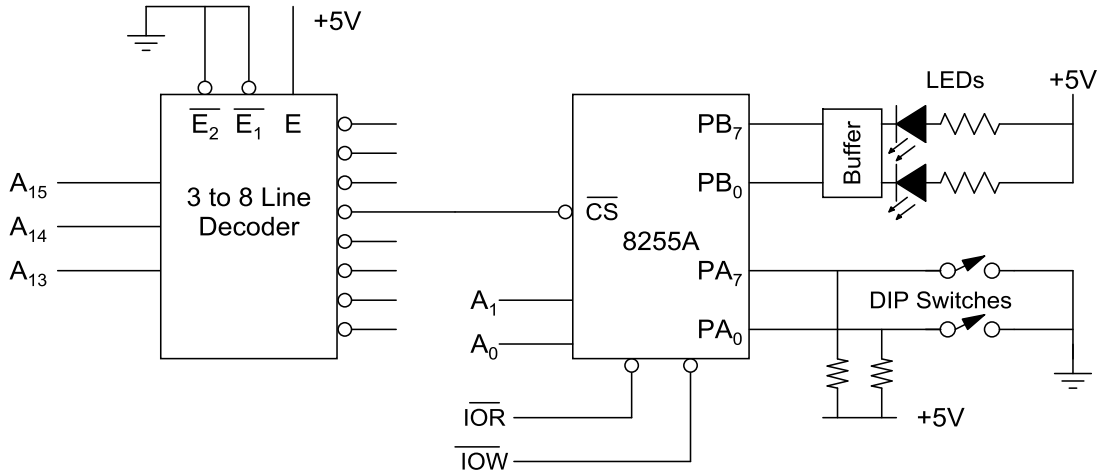
The BSR, word can also be used for enabling or disabling interrupt signals generated by Port C when the 8255 is programmed for Mode 1 or Mode 2 operation. This is done by setting or resetting the associated bits of the interrupts.

Programming in Mode 0 (Basic I/O Mode)

The ports A, B and C can be configured as simple input or output ports by writing the appropriate control word in the control word register. In the control word, D_7 is set to '1' (to define a mode set operation) and D_6, D_5, and D_2 are all set to '0' configure all the ports in Mode 0 operation. The status of bits D_4, D_3, D_1 and D_0 then determine whether the corresponding ports are to be configured as Input or Output.

Example 1

a) Identify the port addresses in given figure.
b) Identify the Mode 0 control word to configure port A as an input port and port B as an output port.
c) Write a program to read the Dip switches and display the reading from port A at port B.



Solution

a) This is I/O mapped I/O; when A_{15} A_{14} A_{13} is 011, then chip select of 8255 is enabled. We also know that during the execution of IN and OUT instruction, A_{15}-A_8 and AD_7-AD_0 carry the same signals. Keeping this in mind, port addresses will be derived. Firstly, port A's port address will be calculated as under:

A_{15}	A_{14}	A_{13}	A_{12}	A_{11}	A_{10}	A_9	A_8		A_7	A_6	A_5	A_4	A_3	A_2	A_1	A_0
0	1	1	X	X	X	X	X	=	X	X	X	X	X	X	0	0

To have equality, 0's and 1's on one side of the equation must appear on other sides. This means that AD_7 AD_6 AD_5 must equal 011 and A_9 and A_8 must equal 00 (port A) to get

0 1 1 X X X 00 = 0 1 1 X X X 00

Since the remaining don't cares can be 0's and 1's, there are many solutions. For instance, if all the don't cares are equal to zero; address of port A becomes 1110 0000 (60H). The port addresses of the given figure are determined as under:

Port A = 60H

Port B = 61H

Port C = 62H

Control Register = 63H

b) The Mode 0 control word to configure port A input and port B output is calculated as under:

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
1	0	0	1	X	0	0	X

= 90H

c) Program subroutine to read DIP switches and display the reading from port A at port B is as under:

```
MVI A, 90H;     Load ACC with the control word
OUT 63H;        Write the control word in control register and initialize the ports
IN 60H;         Reads switches at port A
OUT 61H;        Display  the reading at port B
RET
```

Programming on BSR Mode

Any of the eight bits of port C can be ser or reset using a single output instruction. This feature reduces software requirements in control-based applications. When Port C is being used as Status / Control for Port A or B, these bits can be set or reset by using Bit Set/Reset. Word in the control register when D7 = 0 is recognized as BSR control word and does not affect the I/O operations of Port A and B.

Example 2

Write a BSR control word to set PC_7, PC_6, PC_5, PC_4, PC_3, PC_2, PC_1, and PC_0 and reset each after 1 second.

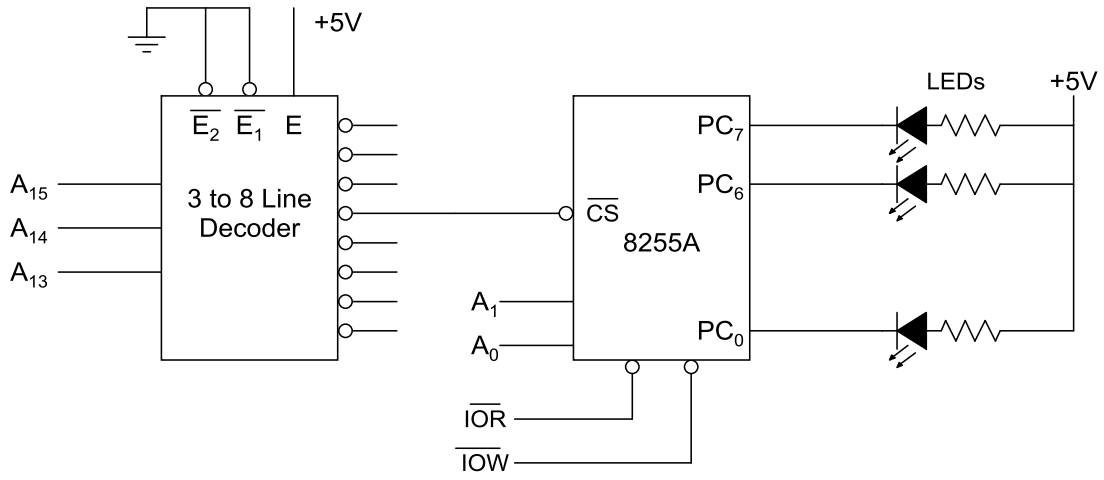


Fig: Example of BSR Mode

Solution

Let us assume Port addresses same as example 1. The control word is calculated with Port C output in this case so it is 10000 0000 (80H). BSR control word for each case is given as under:

Case	**D_7**	**D_6**	**D_5**	**D_4**	**D_3**	**D_2**	**D_1**	**D_0**	**BSR Control Word**
Set PC_7	0	0	0	0	1	1	1	1	0FH
Reset PC_7	0	0	0	0	1	1	1	0	0EH
Set PC_6	0	0	0	0	1	1	0	1	0DH
Reset PC_6	0	0	0	0	1	1	0	0	0CH
Set PC_5	0	0	0	0	1	0	1	1	0BH
Reset PC_5	0	0	0	0	1	0	1	0	0AH
Set PC_4	0	0	0	0	1	0	0	1	09H
Reset PC_4	0	0	0	0	1	0	0	0	08H
Set PC_3	0	0	0	0	0	1	1	1	07H
Reset PC_3	0	0	0	0	0	1	1	0	06H
Set PC_2	0	0	0	0	0	1	0	1	05H
Reset PC_2	0	0	0	0	0	1	0	0	04H
Set PC_1	0	0	0	0	0	0	1	1	03H
Reset PC_1	0	0	0	0	0	0	1	0	02H
Set PC_0	0	0	0	0	0	0	0	1	01H
Reset PC_0	0	0	0	0	0	0	0	0	00H

Program Subroutine

```
        MVI A, 80H
LOOP:   OUT 63H
        MVI A, 0FH
        OUT 63H
        CALL DELAY
        DCR A
        ANI 0FH
        JMP LOOP

DELAY:  MVI C, 0AH
LOOP:   MVI D, 64H
LOOP1:  MVI E, DEH
```

```
LOOP2:  DCR E
        JNZ LOOP2
        DCR D
        JNZ LOOP1
        DCR C
        JNZ LOOP
        RET
```

Programming in Mode 1 (Strobe I/O Mode)

In Mode 1, handshake signals are exchanged between the MPU and peripherals prior to data transfer. Two ports (A and B) function as 8-bit I/O ports. They can be configured either as input or output ports. Each port uses three lines from port C as handshake signals. The remaining two lines of port C can be used for simple I/O functions.

When Port A is to be programmed as an input port, PC_3, PC_4, and PC_5 are used for control, PC_6 and PC_7 can be Input or Output, as programmed by bit D_3 (C_{upper}) of the control word. When Port A is programmed as an output port, PC_3, PC_6, PC_7 are used for control and PC_4 and PC_5 can be Input or Output, as programmed by bit D_3 (C_{upper}) of the control word. When Port B is to be programmed as an input or output port, PC_0, PC_1 and PC_2 are all used for control.

Mode 1 Input

Below figure shows Port A as input port (when it operates in Mode 1) along with the control word and control signals (for handshaking with a peripheral). When the control word is loaded into control register, Group A is configured in Mode 1 with Port A as an input port, Port A can accept parallel data from a peripheral (like a keyboard) and this data can be read by the CPU. The peripheral first loads data into Port A by making the STB_A input low. This latches the data placed by the peripheral on the common data bus into Port A. Port A acknowledges reception of data by making IBF_A (Input Buffer Full) high. IBF_A is set when the STB_A input is made low.

$INTR_A$ is an active output signal which can be used to interrupt the CPU so that the CPU can suspend its current operation and read the data written into Port A by the peripheral. $INTR_A$ can be enabled or disabled by the $INTE_A$ flip-flop which is controlled by BIT Set-Reset operation of PC_4. $INTR_A$ is set (if enabled by setting the $INTE_A$ flip-flop) after the STB_A has gone high again, and if IBF_A is high.

On receipt of the interrupt, the CPU can be made to read Port A. The falling edge of the $\overline{RD}$ input resets IBF_A and it goes low. This can be used to indicate to the peripheral that the input buffer is empty and that data can again be loaded into it.

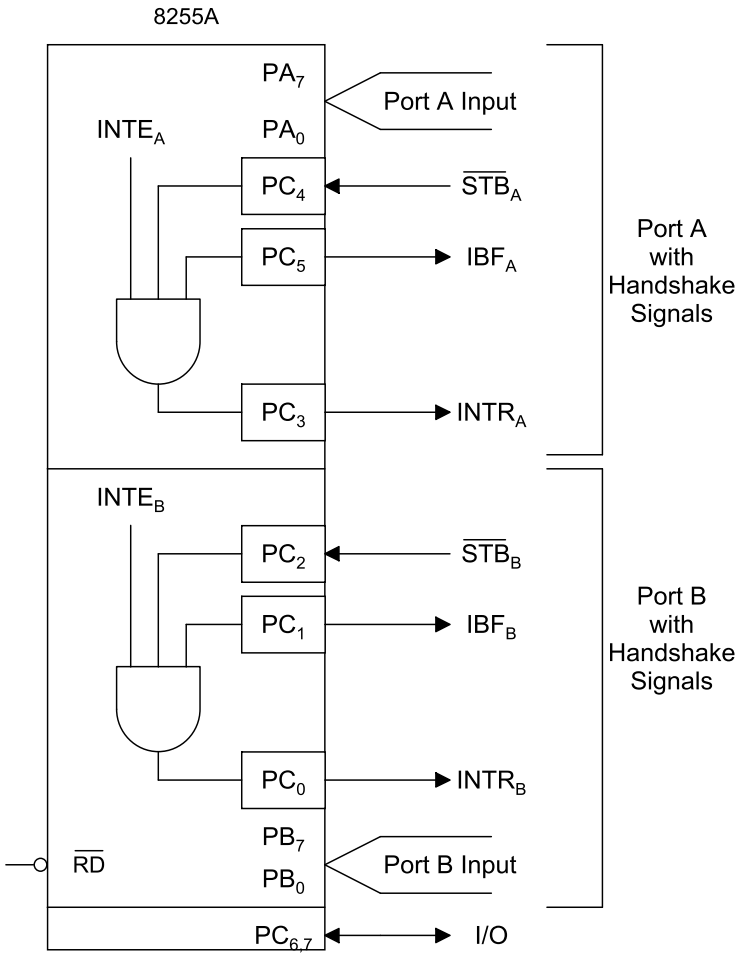


Fig: 8255A Mode 1 Input Configuration

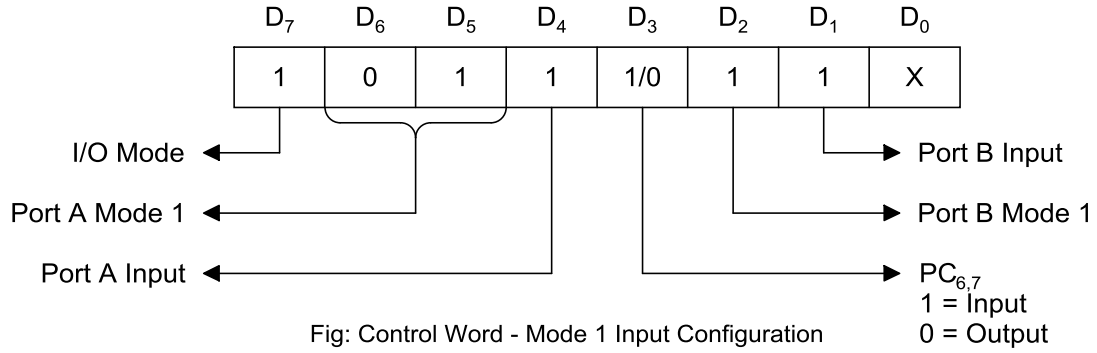


Fig: Control Word - Mode 1 Input Configuration

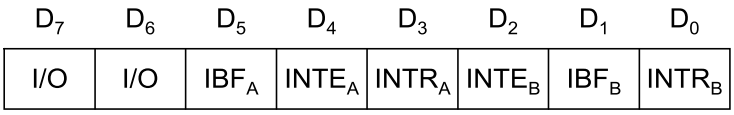


Fig: Status Word - Mode 1 Input Configuration

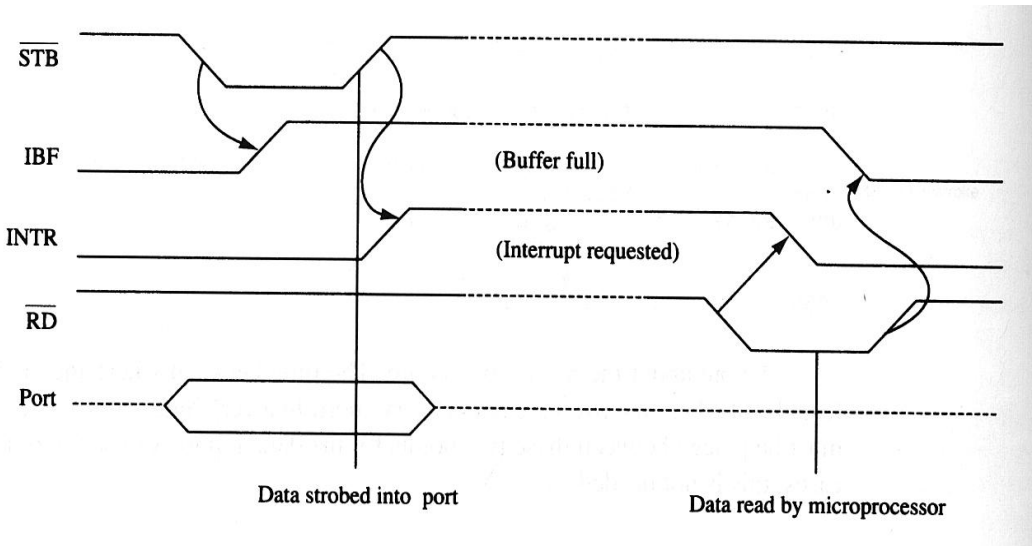


Fig: Timing Waveforms for Strobed Input (With Handshake) – 8255 Mode 1

Above figure shows Port B as an input port (when in Mode 1). The timing diagram and operation of Port B is similar to that of Port A except that it uses different bits of Port C for control. $INTE_B$ is controlled by Bit Set/Reset of PC_2.

If the CPU is busy with other system operations, it can read data from the input port when it is interrupted. This is often called Interrupt Controlled I/O. However, if the CPU is otherwise not busy with other jobs, it can continuously poll (read) the status word to check for an IBF_A. This is often called Program Controlled I/O. The status word is accessed by reading Port C (A_1 A_0 must be 10, $\overline{RD}$ and $\overline{CS}$ must be low). The status word format as assumed by the bits of Port C when Ports A and B are input ports in Mode 1, is shown in above figure.

Mode 1 Input Control Signals

STB' (Strobe Input): A low on this input loads data into the input latch. The 8255A, in response to STB', generates IBF and INTR.

IBF (Input Buffer Full): A high on this output indicates that the data bus has been loaded into the input latch; in essence, an acknowledgement, IBF is set by STB input being low and is reset by the rising edge of the RD' input.

INTR (Interrupt Request): This is an output signal that may be used to interrupt the CPU. This signal is generated if STB', IBF and INTE (Internal Flip Flop) are all at logic 1. This is reset by the falling edge of the RD' (Read) signal.

INTE: This is an internal flip-flop used to enable or disable the generation of the INTR signal. The two flip-flops $INTE_A$ and $INTE_B$ are set/reset using the BSR mode through PC_4 and PC_2.

Mode 1 Output

Figure below shows Port A configured as an output port (when in Mode 1) along with the control word and control signals (for handshaking with a peripheral). When the control word is loaded into the control register, Group A is configured in Mode 1 with Port A as an output port. The CPU can send out data to a peripheral (like a display device) through Port A of the 8255.

The $\overline{OBF_A}$ output (Output Buffer Full) goes low on the rising edge of the $\overline{WR}$ signal (when the CPU writes data into the 8255). The $\overline{OBF_A}$ output from 8255 can be used as a strobe input to the peripheral to latch the contents of Port A. The peripheral responds to the receipt of data by making the $\overline{ACK_A}$ input of the 8255 low, thus acknowledging that it has received the data sent out by the CPU through Port A. The $\overline{ACK_A}$ low resets the $\overline{OBF_A}$ signal, which can be polled by the CPU through $\overline{OBF_A}$ of the status word to load the next data when it is high again.

$INTR_A$ is an active high output of the 8255 which is made high (if the associated INTE flip-flop is set) when $\overline{ACK_A}$ is made high again by the peripheral, and when $\overline{OBF_A}$ goes high again (see timing diagram in Figure below). It can be used to interrupt the CPU whenever the output buffer is empty. It is reset by the falling edge of $\overline{WR}$ when the CPU writes data onto Port A. It can be enabled or disabled by writing a '1' or a '0' respectively to PC_6 in the BSR mode.

Figure below shows Port B as an output port when in Mode 1. The operation of Port B is similar to that of Port A. $INTE_B$ is controlled by writing a '1' or '0' to PC_2 in the BSR mode.

The status word is accessed by issuing a Read to Port C. The format of the status word as assumed by the bits of Port C when Ports A and B are Output ports in Mode 1 is shown in Figure below.

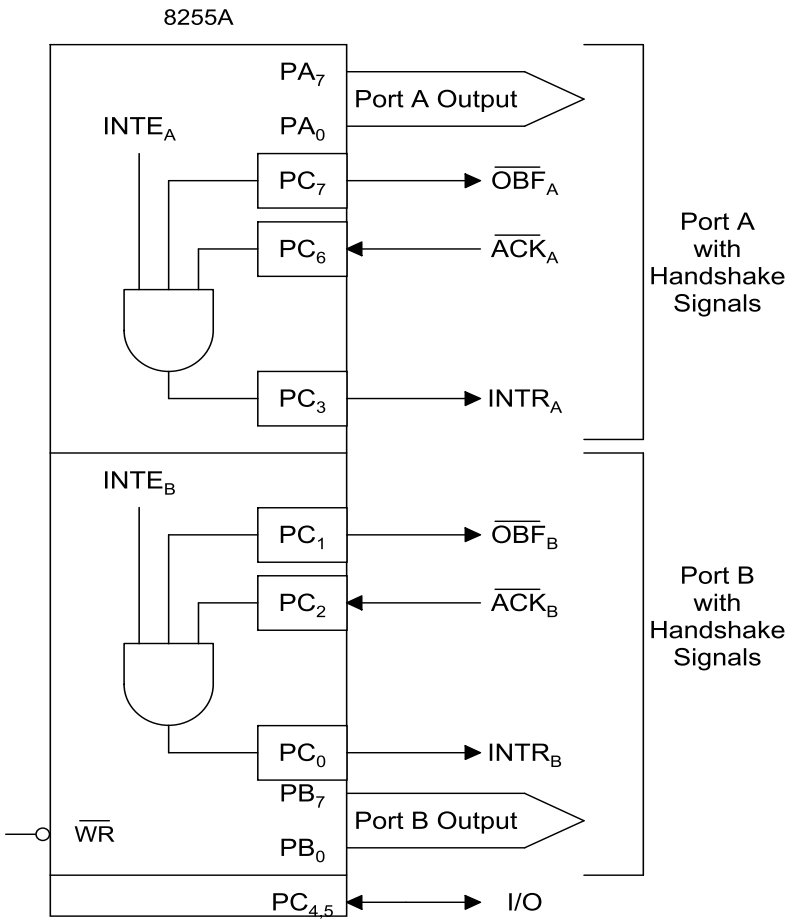


Fig: 8255A Mode 1 Output Configuration

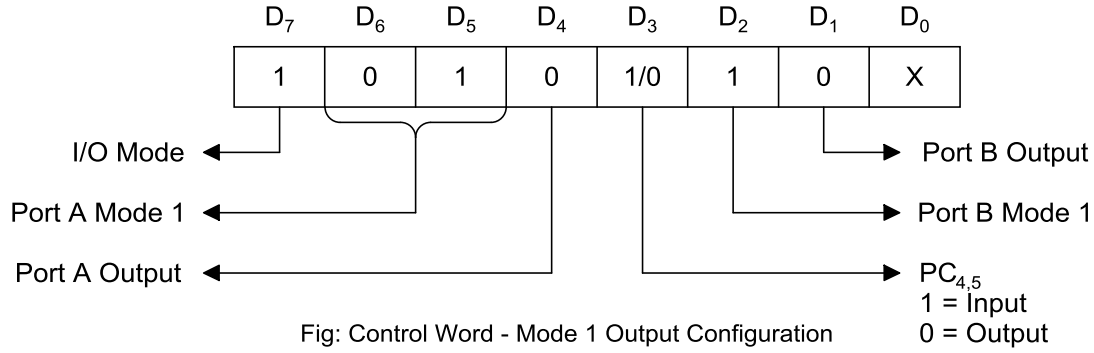


Fig: Control Word - Mode 1 Output Configuration

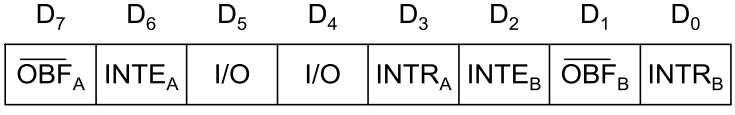


Fig: Status Word - Mode 1 Output Configuration

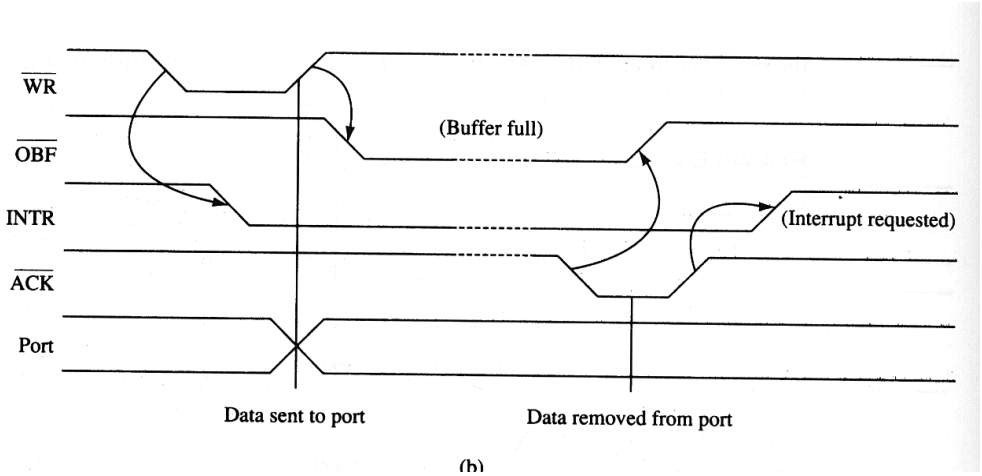


Fig: Timing Waveform for Strobed (With Handshake) Output - 8255 Mode 1

Mode 1 Output Control Signals

OBF' (Output Buffer Full): The OBF' will go low to indicate that the CPU has written data out to the specified port. The OBF' will be set with the rising edge of the WR' input and reset by ACK' input being low.

ACK' (Acknowledgement Input): A low on this input informs the 8255A that the data from port A or port B has been accepted. In essence, a response from the peripheral device indicating that it has received the data output by the CPU.

INTR (Interrupt Request): A high on this output can be used to interrupt the CPU when an output device has accepted data transmitted by the CPU. INTR is set when OBF', ACK' and INTE are all 1 and reset by falling edge of WR'.

INTE: This is an internal flip-flop to a port and needs to be set to generate the INTR signal. The two flip-flops $INTE_A$ and $INTE_B$ are set/reset using the BSR mode through PC_6 and PC_2.

Example 3

Below mentioned figure shows an interfacing circuit using the 8255A in Mode 1. Port A is designated as the input port for a keyboard with interrupt I/O and port B is designated as the output port for a printer with status check I/O.

a) Find port addresses by analyzing the decode logic.
b) Determine the control word to set up port A as input and port B as output in Mode 1.
c) Determine the BSR word to enable $INTE_A$.
d) Determine the masking byte to verify the OBF' line in status check I/O.
e) Write subroutine to accept character from keyboard and send character to printer.

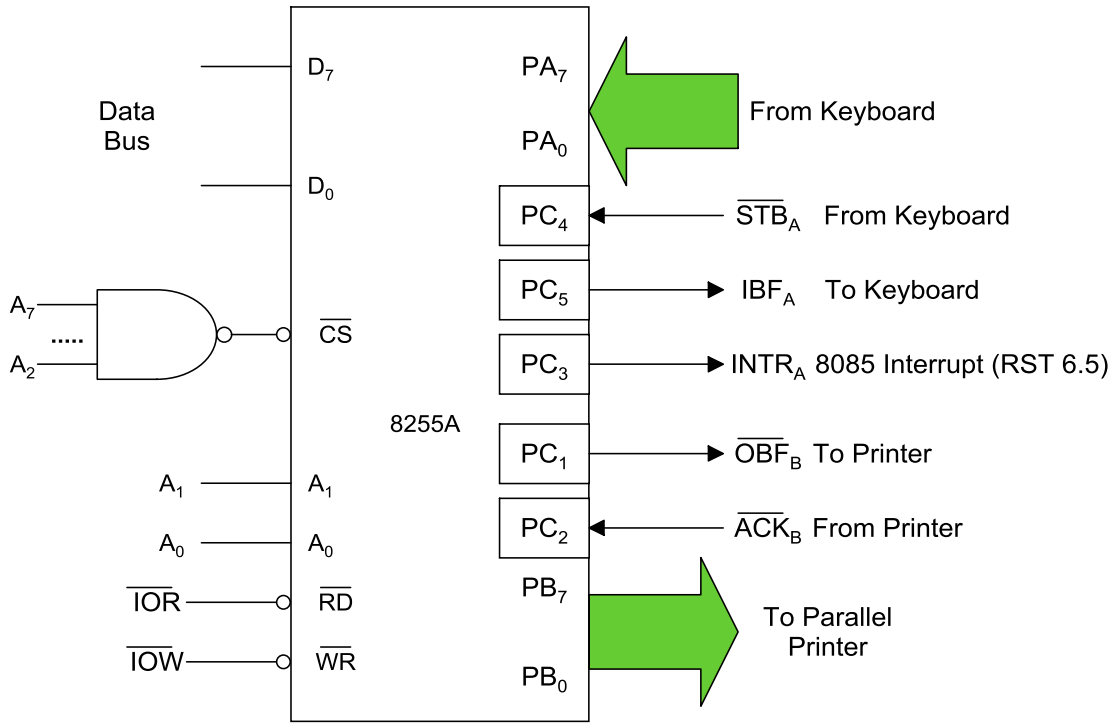


Fig: 8255A Mode 1 Example

Solution

a) The 8255A is connected as I/O mapped I/O. When the address lines A7-A2 are all 1, the output of NAND gate goes low and selects 8255A. The port addresses are calculated as 1111 11XX:

Port A = 1111 1100 (FCH)

Port B = 1111 1101 (FDH)

Port C = 1111 1110 (FEH)

Control Register = 1111 1111 (FFH)

b) Control word to set up port A as input and port B as output Mode 1 is:

D_7	**D_6**	**D_5**	**D_4**	**D_3**	**D_2**	**D_1**	**D_0**
1	0	1	1	X	1	0	X

= B4H

c) BSR word to set $INTE_A$

D_7	**D_6**	**D_5**	**D_4**	**D_3**	**D_2**	**D_1**	**D_0**
0	0	0	0	1	0	0	1

= 09H

d) Status word to check OBF_B'

D_7	**D_6**	**D_5**	**D_4**	**D_3**	**D_2**	**D_1**	**D_0**
X	X	X	X	X	X	OBF_B'	X

Masking Byte = 02H

e) Subroutines to accept character from keyboard and send to printer:

```
        MVI A, B4H   ; Initialize control word
        OUT FFH      ; Using I/O Mode
        MVI A, 09H   ; Set INTE_A (PC4)
        OUT FFH      ; Using BSR Mode
        EI           ; Enable Interrupt
        CALL READ    ; Read Character
        CALL PRINT   ; Display Character
        HLT          ; Terminate Program
READ:               ; Keyboard Read Subroutine
        IN FEH       ; Read Port C
        ANI 20H      ; Check IBF_A (PC5)
        JZ READ
        IN FCH       ; Read ASCII code of character
        MOV C, A     ; Save Character
        RET
PRINT:
        IN FEH       ; Read port C
        ANI 02H      ; Check OBF_B' (D1)
        JZ PRINT
        MOV A, C     ; Get Character
        OUT FDH      ; Send Character to port B
        RET
```

Programming in Mode 2 (Strobe Bidirectional Bus I/O)

When the 8255 is operated in Mode 2 (by loading the appropriate control word); Port A can be used as a bidirectional 8-bit I/O bus using PC_3–PC_7 for handshaking and Port B can be programmed only in Mode 0 (PC_0–PC_2 as Input or Output), or in Mode 1 (with PC_0–PC_2 for handshaking).

Figure below shows the control word that would have to be loaded into the control port to configure 8255 in Mode 2. Figure below shows Port A and associated control signals when 8255 is in Mode 2. Interrupts are generated for both output and input operations on the same $INTR_A$ (PC_3) line.

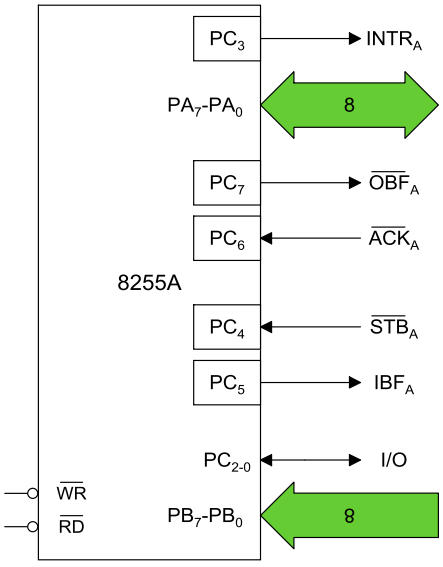


Fig: 8255A Mode 2 and Mode 0 Input Configuration

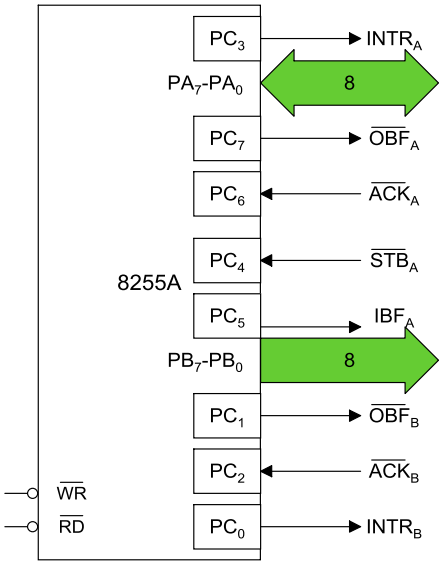


Fig: 8255A Mode 2 and Mode 1 Output Configuration

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
1	1	X	X	X	0	1	1/0

PC_{2-0}
1 = Input
0 = Output

Fig: Control Word - Mode 2 and Mode 0 Input Configuration

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
1	1	X	X	X	1	0	X

Fig: Control Word - Mode 2 and Mode 1 Output Configuration

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
OBF_A	$INTE_A$	IBF_A	$INTE_2$	$INTR_A$	X	X	X

Group A — Defined by Mode 0 and Mode 1 Selection

Fig: Status Word - Mode 2 Configuration

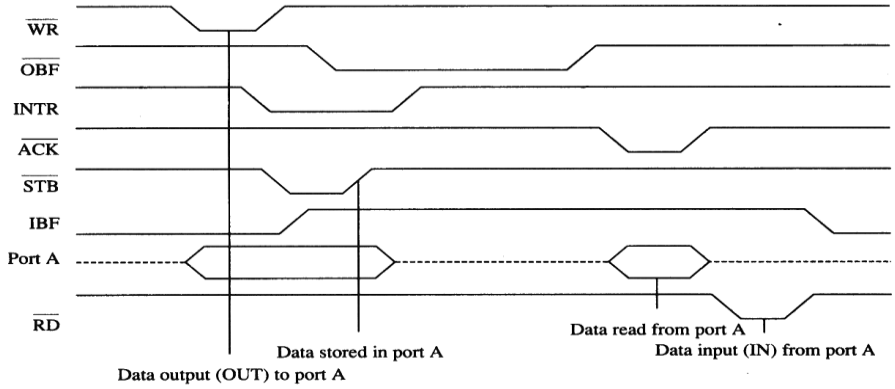


Fig: Timing Waveform for Mode 2 Configuration

The control signal definitions for Mode 2 are:

Output Control Signals

$\overline{OBF}$ ***(Output Buffer Full)***

This is an active low output which indicates that the CPU has written data into Port A.

$\overline{ACK}$ ***(Acknowledge)***

This is an active low input signal (generated by the peripheral) which enables the tri-state output buffer or Port A and makes Port A data available to the peripheral. In Mode 2, Port A outputs are in tri-state until enabled.

INTE 1

This is the flip-flop associated with Output Buffer Full. INTE 1 can be used to enable to disable the interrupt by setting or resetting PC_6 in the BSR Mode.

Input Control Signals

$\overline{STB}$(Strobe Input)

This is an active low input signal which enables Port A to latch the data available as its input.

IBF (Input Buffer Full Flip-Flop)

This is an active high output which indicates that data has been loaded into the input latch of Port A.

INTE 2

This is an Interrupt enable flip-flop associated with Input Buffer Full. It can be controlled by setting or resetting PC_4 in the BSR Mode.

Status Word in Mode 2

The status word for Mode 2 (accessed by reading Port C) is shown in above figure. D_7–D_3 of the status word carry information about $\overline{OBF_A}$, $INTE_1$, IBF_A, $INTE_2$, and $INTR_A$. The status of the bits $D_2 – D_0$ depend on the mode setting of Group B. If B is programmed in Mode 0, D_2–D_0 carry information about the control signals for B, depending upon whether B is an Input port or Output port respectively.

Assignment 1:

- Interfacing keyboard and seven segment display
- Interfacing a microprocessor to a tape reader and lathe
- Interfacing to parallel printer

2.3 Parallel Interfacing with ISA and PCI bus

I/O buses are used to connect the system bus (address, data, and control buses) for example ISA (8 or 16 bit), EISA (Extended ISA - 32 bit), VESA (Video Electronics Standards Association) local bus (VL Bus), PCI (32 or 64 bit), Accelerated graphics port (AGP), PCI-X (64 bit, 133MHZ), PCI-Express etc.

ISA Bus (Industry Standard Architecture)

- First introduced in 1982 with the first PC (IBM/PC) – [Intel 8088 8 bit microprocessor].
- Originally ISA bus was with 8-bit bus which runs at 4.77 MHz.
- 16 bit version of ISA was introduced in 1984 used with Intl 80286 (16-bit microprocessor).
- Peripheral devices such as sound cards, disk drives, network cards etc. are connected via ISA slots.
- ISA bus is mostly obsolete for PC nowadays, but is still used in many industrial applications due to their low costs and existing cards.

8-bit ISA bus Architecture

- Has data bus width of 8 bits and address bus width of 20 bits.
- Number of pins in ISA slots/cards are 62.
- Clock frequency of 4.77 MHz.
- ISA bus connector contains:
    - 20 bit address bus (A_{19}-A_0)
    - 8 bit data bus
    - MEMR', MEMW'. IOR', IOW' control signal for controlling I/O or memory on the ISA card.
    - Interrupt request lines IRQ_2-IRQ_7
    - DMA request inputs DRQ_1-DRQ_3
    - DMA acknowledgement O/Ps $DACK_0$'-$DACK_3$'
    - Clock signals
    - Power lines and Reset

Back of Computer

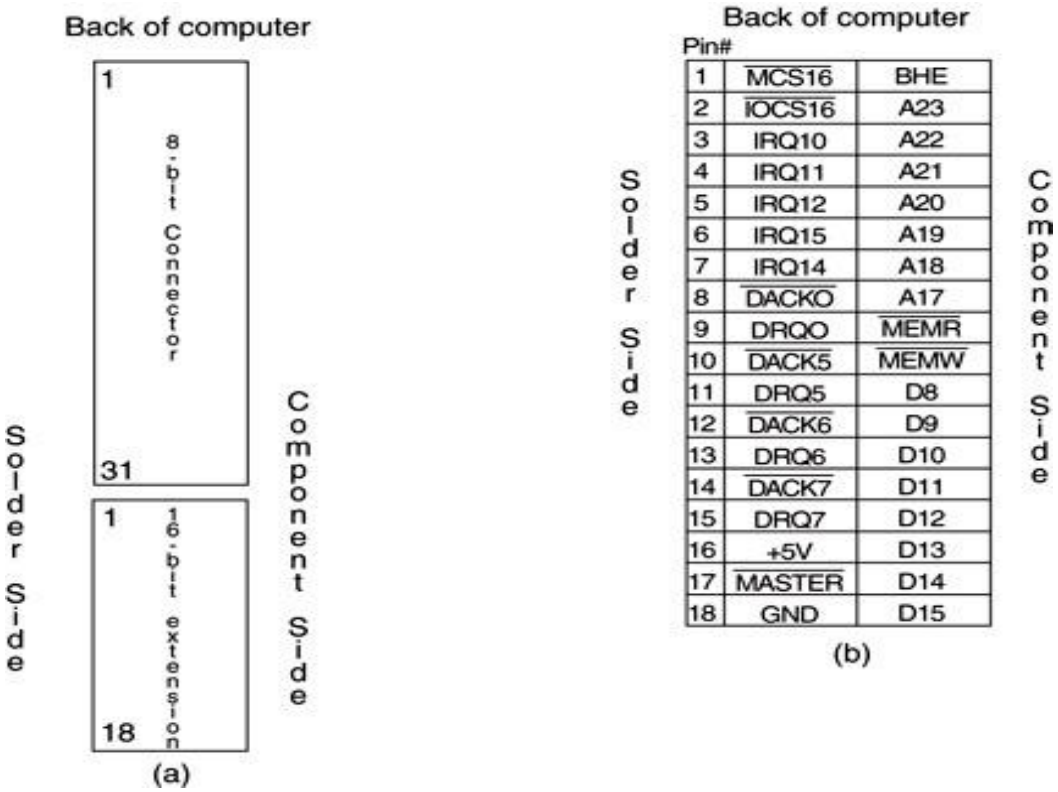
Pin #

	Solder Side	Component Side
1	GND	$\overline{\text{IO CHK}}$
2	RESET	D7
3	+5V	D6
4	IRQ9	D5
5	–5V	D4
6	DRQ2	D3
7	–12V	D2
8	OWS	D1
9	+12V	D0
10	GND	IO RDY
11	$\overline{\text{MEMW}}$	AEN
12	$\overline{\text{MEMR}}$	A19
13	$\overline{\text{IOW}}$	A18
14	$\overline{\text{IOR}}$	A17
15	$\overline{\text{DACK3}}$	A16
16	DRQ3	A15
17	$\overline{\text{DACK1}}$	A14
18	DRQ1	A13
19	$\overline{\text{DACK0}}$	A12
20	CLOCK	A11
21	IRQ7	A10
22	IRQ6	A9
23	IRQ5	A8
24	IRQ4	A7
25	IRQ3	A6
26	$\overline{\text{DACK2}}$	A5
27	T/C	A4
28	ALE	A3
29	+5V	A2
30	OSC	A1
31	GND	A0

Fig: 8-bit ISA Bus

16-bit ISA bus Architecture

- Data bus width of 16 bit and address bus width of 24 bits.
- Number of pins in ISA card/slot are 98
- Clock frequency of 8.33 MHz
- Consists of an extra connector with 36 pins behind the 8-bit connector.
- Compatible with both 8-bit and 16-bit ISA cards.
- 16-bit card consists of two edge connectors
    - One plugs into the original 8-bit connector
    - Other plugs into the new 16-bit connector
- Extra connector consists of
    - 4 additional address lines – 24 lines in total
    - 8 additional data lines – 16 lines in total
    - 4-bit DMA channel request and acknowledgement lines
    - Additional Interrupt lines
    - Control lines to select 8 or 16 bit transfer



Pin#		
1	$\overline{\text{MCS16}}$	BHE
2	$\overline{\text{IOCS16}}$	A23
3	IRQ10	A22
4	IRQ11	A21
5	IRQ12	A20
6	IRQ15	A19
7	IRQ14	A18
8	$\overline{\text{DACKO}}$	A17
9	DRQO	$\overline{\text{MEMR}}$
10	$\overline{\text{DACK5}}$	$\overline{\text{MEMW}}$
11	DRQ5	D8
12	$\overline{\text{DACK6}}$	D9
13	DRQ6	D10
14	$\overline{\text{DACK7}}$	D11
15	DRQ7	D12
16	+5V	D13
17	$\overline{\text{MASTER}}$	D14
18	GND	D15

Fig: The 16-bit ISA bus. (a) Both 8- and 16-bit connectors and (b) the pinout of the 16-bit connector.

Reasons for elimination of ISA Bus

- ISA bus is slow, hard to use and bulky.
- Once each ISA slot/card uses dedicated interrupt lines, only limited number of cards can be used.
- Since address lines of 24 bits, a maximum of 2^{24}=16 MB of RAM can only be accessed for DMA.
- Since data bus size is 16 bits only, higher bits data (32-bits) communication would reduce system performance.
- ISA cards do not have plug and play (PnP) technology i.e. they can't be configures automatically by BIOS or operating system.
- ISA cards must be controlled manually by setting the I/O addresses, interrupts and clock speed using jumpers and switches on the card itself.

Improvements in ISA bus

- EISA (Extended ISA) of 32-bits, 8 MHz; now obsolete
- ISA PnP for plug and play; now obsolete
- VL-Bus of 32-bits operated at the speed of local bus (CPU)
    - Used only for graphics cards
    - Possibility of interference with the performance of the CPU

PCI Bus (Peripheral Component Interconnect)

- Introduced in 1990 by Intel
- Provides direct access to the CPU and system memory but uses a bridge to connect to the system bus to eliminate the potential for interference with CPU.
- PCI bus is independent of processor type or speed
- Originally operated at 33 MHz using 32-bit data lines
- Revised standard at 66 MHz using 64-bit data lines
- The 32-bit PCI connector has 124 pins and 64-bit PCI connector has 188 pins
- The PCI bus is able to work with so few pins because of hardware multiplexing i.e. the device sends more than one signal over a single pin
- Also, PCI supports devices that use 5v signalling voltage levels
- PCI card support plug and play (PnP) feature i.e. PCI devices are automatically recognized and configured to work in system.

Advancement in PCI bus

- PCI-X (PCI extended): runs at 133 MHz, 32-bit and 1.06 GBps data rate
- PCI-E (PCI express): replaced PCI, PCI-X & AGP standards

Back of computer

Pin #	Solder Side	Component Side	Pin #	Solder Side	Component Side
1	−12V	$\overline{\text{TRST}}$	41	+3 3V	$\overline{\text{SBO}}$
2	TCK	+12V	42	$\overline{\text{SERR}}$	GND
3	GND	TMS	43	+3 3V	PAR
4	TDO	TD1	44	C/$\overline{\text{BE1}}$	AD15
5	+5V	+5V	45	AD14	+3.3V
6	+5V	$\overline{\text{INTA}}$	46	GND	AD13
7	$\overline{\text{INTB}}$	$\overline{\text{INTC}}$	47	AD12	AD11
8	$\overline{\text{INTD}}$	+5V	48	AD10	GND
9	$\overline{\text{PRSNT 1}}$		49	GND	AD9
10		+VI/O	50	KEY	KEY
11	$\overline{\text{PRSNT 2}}$		51	KEY	KEY
12	KEY	KEY	52	AD8	C/$\overline{\text{BE0}}$
13	KEY	KEY	53	AD7	+3.3V
14			54	+3 3V	AD6
15	GND	$\overline{\text{RST}}$	55	AD5	AD4
16	CLK	VI/O	56	AD3	GND
17	GND	$\overline{\text{VNT}}$	57	GND	AD2
18	$\overline{\text{REQ}}$	GND	58	AD1	AD0
19	+V IO		59	+V IO	+V IO
20	AD31	AD30	60	$\overline{\text{ACK64F}}$	$\overline{\text{REQ64}}$
21	AD29	+3.3V	61	+5V	+5V
22	GND	AD28	62	+5V	+5V
23	AD27	AD26	63		GND
24	AD25	GND	64	GND	C/$\overline{\text{BE7}}$
25	+3.3V	AD24	65	C/$\overline{\text{BE6}}$	C/$\overline{\text{BE5}}$
26	C/$\overline{\text{BE3}}$	IDSEL	66	C/$\overline{\text{BE4}}$	+V IO
27	AD23	+3.3V	67	GND	PAR64
28	GND	AD22	68	AD63	AD62
29	AD21	AD20	69	AD61	GND
30	AD19	GND	70	+V IO	AD60
31	+3 3V	AD18	71	AD59	AD58
32	AD17	AD16	72	AD57	GND
33	C/$\overline{\text{BE2}}$	+3 3V	73	GND	AD56
34	GND	$\overline{\text{FRAME}}$	74	AD55	AD54
35	$\overline{\text{IRDY}}$	GND	75	AD53	+V IO
36	+3 3V	$\overline{\text{TRDY}}$	76	GND	AD52
37	$\overline{\text{DEVSEL}}$	GND	77	AD51	AD50
38	GND	$\overline{\text{STOP}}$	78	AD49	GND
39	$\overline{\text{LOCK}}$	+3 3V	79	+V IO	AD48
40	$\overline{\text{PERR}}$	SDONE	80	AD47	AD46
			81	AD45	GND
			82	GND	AD44
			83	AD43	AD42
			84	AD41	+VI/O
			85	GND	AD40
			86	AD39	AD38
			87	AD37	GND
			88	+VI/O	AD36
			89	AD35	AD34
			90	AD33	GND
			91	GND	AD32
			92		
			93		GND
			94	GND	

Notes: (1) pins 63–94 exist only on the 64-bit PCI card
(2) + VI/O is 3.3V on a 3.3V board and +5V on a 5V board
(3) blank pins are reserved

Fig: The pin out of the PCI bus

Tutorials:

1. Assume that your group has decided to make a PC based control system for a wine company. After studying the system, your group found out that the following to be implemented for controlling purpose:
   - Pressure measurement (6 points)
   - Temperature measurement (5 points)
   - Weight measurement (1 point)
   - Volume measurement for filling (5 points)

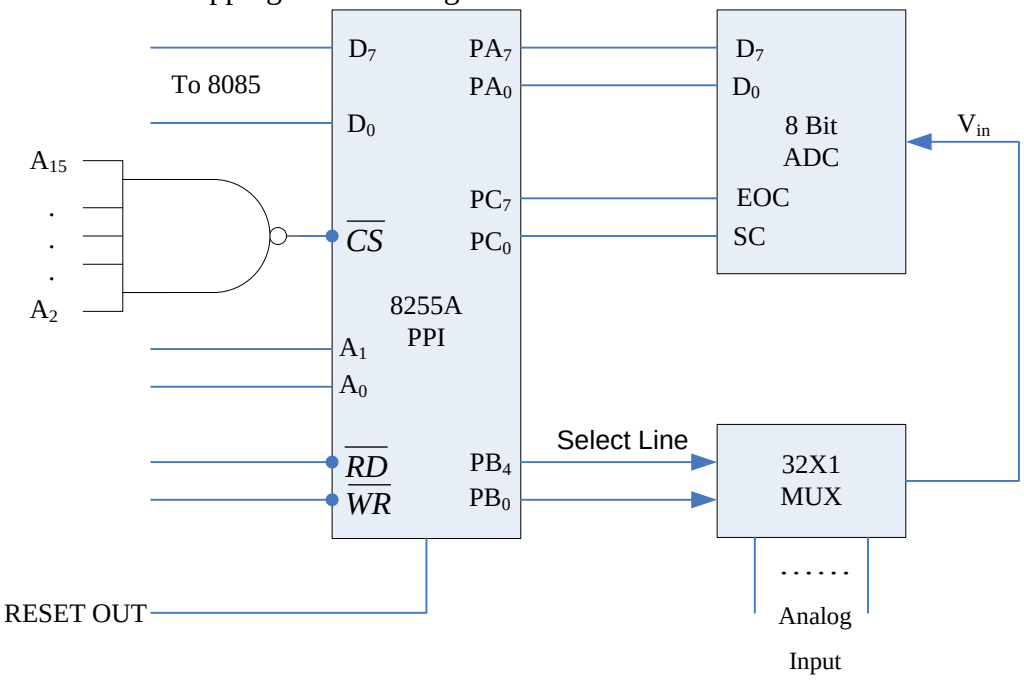
   Your group also decided to use 8255A PPI card at base address 0550H.
   a) List out collected documents and components
   b) List out different signals you need to derive and or can be directly connected to your interfacing circuit.
   c) Draw minimum mapping circuit for above system
   d) What are the address captured by card
   e) Generate necessary control word
   f) Write a program module for measuring the pressure of all the points and control if the pressure is not in a range, Assume suitable data if necessary.

Solution:

a) Components: 8255A card, ADC, MUX, Memory, Processor, connecting wires, power supplies (+5V, GND), gates etc.
   Documents: Data sheets and technical documentation of above components

b) Signals needed to be derived on directly connected to circuit
   - A1, A2, Chip Select ($\overline{CS}$) for Port selection of of 8255A, RESET signal
   - Read ($\overline{RD}$) and Write ($\overline{WR}$) signals
   - Start Conversion (SC) and End of Conversion (EOC)

c) The minimum mapping circuit is as given below:



d) The base address of card is 0550H, following are address captured by card.

Port	Address	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
A	0550H	0	0	0	0	0	1	0	1	0	1	0	1	0	0	**0**	**0**
B	0551H	0	0	0	0	0	1	0	1	0	1	0	1	0	0	**0**	**1**
C	0552H	0	0	0	0	0	1	0	1	0	1	0	1	0	0	**1**	**0**
CR	0553H	0	0	0	0	0	1	0	1	0	1	0	1	0	0	**1**	**1**

The total numbers of monitoring points are 17. If we use 1 ADC for all of them, we need to select any one at given time. So, we can use 32X1 MUX which would then have 2^5=32 i.e. 5 selection lines (B_0 to B_4). These lines can have defined for any of the 17 lines.
In the above circuit,
Port A → Input port to read data from ADC in mode 0
Port B → Output port to select any one of 17 lines from MUX in mode 0
Port C → Output port (PC_0 as SC) and Input port (PC_7 as EOC)

e) Control word and BSR words:
Control word to set up port ports in above configuration:

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
1	0	0	1	1	0	0	0

= 98H

BSR word to set PC_0

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
0	0	0	0	0	0	0	1

= 01H

BSR word to reset PC_0

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
0	0	0	0	0	0	0	0

= 00H

Assuming that ADC starts the conversion process only when it receives SC signal and after conversion indicates via EOC line i.e. it has finished conversion and so ADC port data in its data lines which can be now be read through port A.

f) Program Module:

```
        LXI H, MEMORY
        MVI A, 98H
        STA 0553H;   write control word in CR
        MVI C, 06H;  set counter to read 6 pressure points
        MVI B, 00H;  selection of points for MUX
NEXT:   MOV A, B
        STA 0551H;   select first pressure point
        MVI A, 01H;  load A with BSR word to set PC0
```

```
        STA 0553H;   set SC line
        CALL DELAY
        MVI A, 00H;  load A with BSR word to reset PC0
        STA 0553H;   reset SC line
READ:   LDA 0552H;   read port C
        RAL
        JNC READ;    check for PC7
        LDA 0550H;   read data from port A
        MOV M, A;    store value in memory
        CPI MAX_VALUE;   compare with maximum value
        JNC CONTROL;     control value
        CPI MIN_VALUE;   compare with minimum value
        JC CONTROL;      control value
        INR B
        INX H
        DCR C
        JNZ NEXT
```

2. Interfacing keyboard and seven-segment display.
   (Refer Gaonkar → 15.2 → pages 480-487)
3. Interfacing Lathe machine and tape reader.
   (Refer Hall)
4. Interfacing parallel printer.
   (Refer Hall)
5. Interface a temperature sensor using an A/D converter and port A of the 8255. Interface a fan and a heater using optocouplers and triacs to drive the I/O devices. Write instructions to read the temperature; if the temperature is less than 10^oC, turn on the heater; and if the temperature is higher than 35^oC, turn on the fan.

   Load temperature from temperature sensor LM135 and control fan and heater.
   If temperature > 35^o → Fan ON
   If temperature < 10^o → Heater ON
   (Refer Gaonkar → 15.1.4 → pages 468-472)

6. You are required to monitor the operation of pump as well as status of upper and lower tank in the household. Apart from that you need to control 3 lights that are to turn ON in the evening and turn OFF in the morning time. Additionally, you also need to check the status of smoke sensors in Room1, Room2 & Room3, and heat sensor in kitchen and ring alarm when necessary.
   Your group also decided to use 8255 PPI card at base address 3000H in memory mapped I/O for controlling purpose. Make complete circuitry including relays and relay driving transistor.
   Write a program module to read status of heat sensor and generate alarm when the limit exceeds.