Instrumentation II
Chapter 2
BCT III/I

Dr. Raksha Dangol,
Assistant Professor,
Advanced College of Engineering and Management

Chapter – 2 Parallel Interfacing with Microprocessor Based System (4 hours)

- **Chapter Outline**

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

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

2.3 Parallel Interfacing with ISA and PCI bus

Chapter 2 : Introduction

- **Introduction**
  - **Parallel Data Transmission:**
    - N bits of data are handled simultaneously by the bus that, links to the device directly.
    - Achieves faster communication but becomes expensive due to need of multiple wires.
    - Handling of data at higher speed is possible.
    - Information exchanged between a microprocessor and an I/O interface circuit consists of
      - input or output data
      - control information - command by microprocessor to cause I/O device to take some action
    - Status information enable the microprocessor to monitor the device and when it is ready then send or receive data
    - Used for short distance where the speed of information transfer is critical.
    - Found in newer type of computer peripheral equipment with transfer speed of upto one million characters per second
    - Also called Synchronous transmission – Technique used to transfer data between different speed devices and computer / microprocessor

Chapter 2.1 Methods of Parallel Data Transfer

- **Simple I/O**
  - Microprocessor transfers data assuming that I/O is always ready to send/receive data.
  - 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.

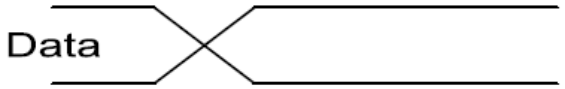


Fig: Simple I /O

  - 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**

- **Handshake I/O**
  - **Single Handshake I/O**
  - **Double Handshake I/O**

Chapter 2.1 Methods of Parallel Data Transfer (Contd…)

- **Simple Strobe I/O**
  - Simple I/O is not suitable for data transmission between slow device and microprocessor. (why?)
  - Slow devices do not send or accept data at the same speed as microprocessor.
  - In Strobe I/O, microprocessor transfers data only when the peripheral is ready.
  - Slow devices generate control signal called *strobe* along with valid data.
  - Strobe signal indicates that valid data is present on the data lines.
  - For example - 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. The sending device outputs parallel data on the data lines, and then outputs $\overline{\text{STB}}$ signal to represent the valid data is present.
  - Microprocessors need to wait until the device is ready for the operation, so this technique is also known as simple wait I/O.

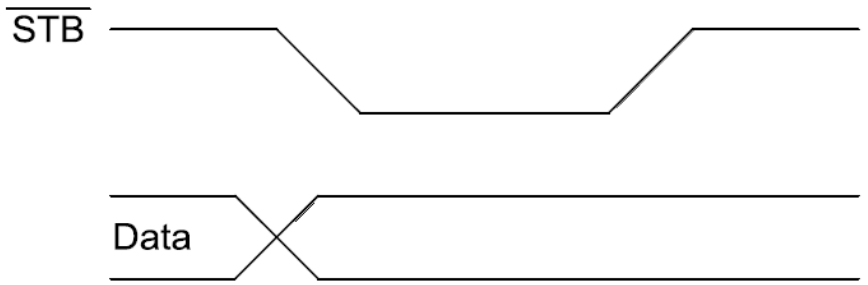


Fig: Simple Strobe I/O

Chapter 2.1 Methods of Parallel Data Transfer (Contd…)

- **Handshake I/O**
  - **Single Handshake I/O**
  - 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 ($\overline{\text{STB}}$) 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.

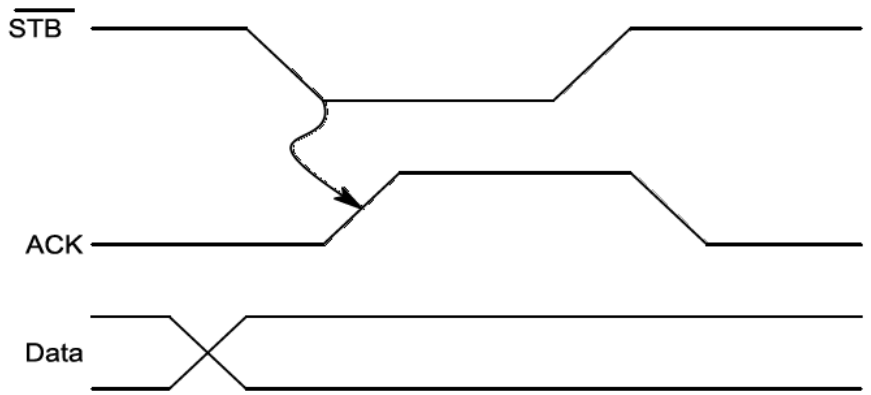


Fig: Single Handshaking

Chapter 2.1 Methods of Parallel Data Transfer (Contd…)

- **Single Handshake I/O (Contd…)**
  - **Input Handshake (Contd..)**
    - The peripheral outputs some data and send $\overline{\text{STB}}$ signal to microprocessor to tell “Here is the data for you”.
    - Microprocessor detects asserted $\overline{\text{STB}}$ 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.

Chapter 2.1 Methods of Parallel Data Transfer (Contd…)

- **Double Handshake I/O**

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 ($\overline{STB}$) 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 ($\overline{STB}$) line high.
- When microprocessor reads data, it drops its acknowledgement (ACK) line low and request sending system to send net byte of data.

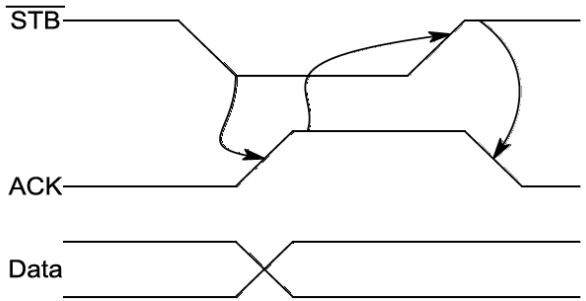


Fig: Double Handshaking

Chapter 2.1 Methods of Parallel Data Transfer (Contd…)

- **Double Handshake I/O : Input Handshake (Contd…):**
  - The peripheral asserts its $\overline{\text{STB}}$ 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 $\overline{\text{STB}}$ 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.”

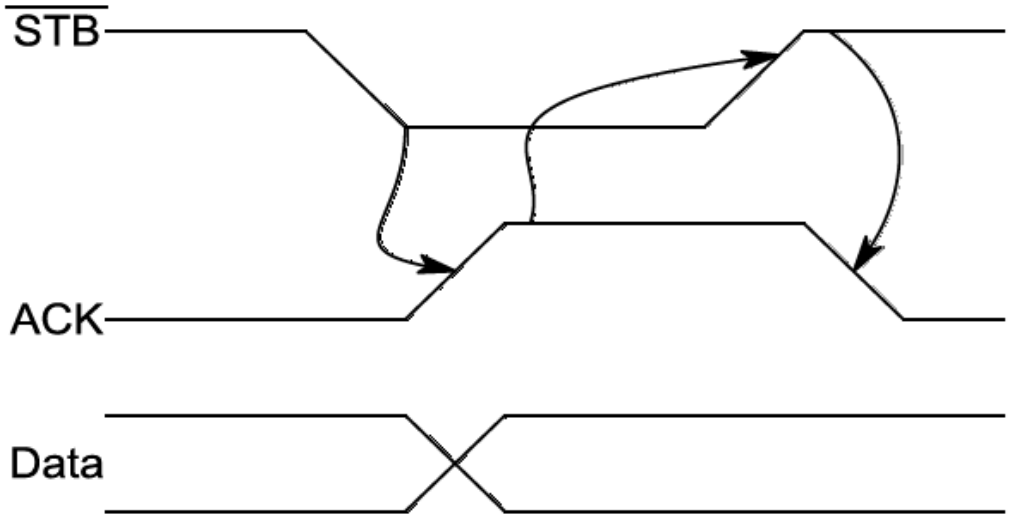


Fig: Double Handshaking

Chapter 2.1 Methods of Parallel Data Transfer (Contd…)

- **5.1 Methods of Parallel Data Transfer (Contd…)**
  - **Double Handshake I/O : Output Handshake (Microprocessor to Peripheral):**
    - Microprocessor sends a strobe ($\overline{\text{STB}}$) signal and data and peripheral sends acknowledgement (ACK) signal.

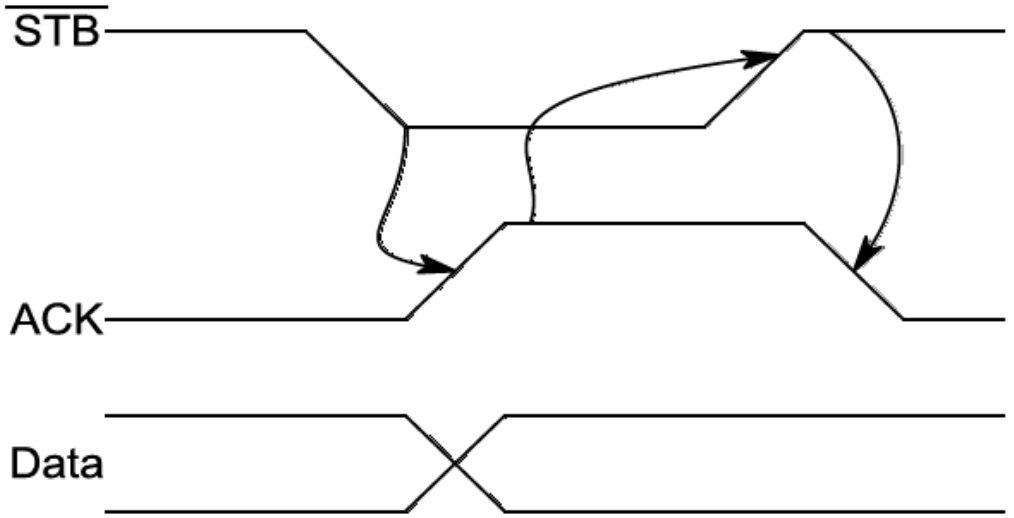


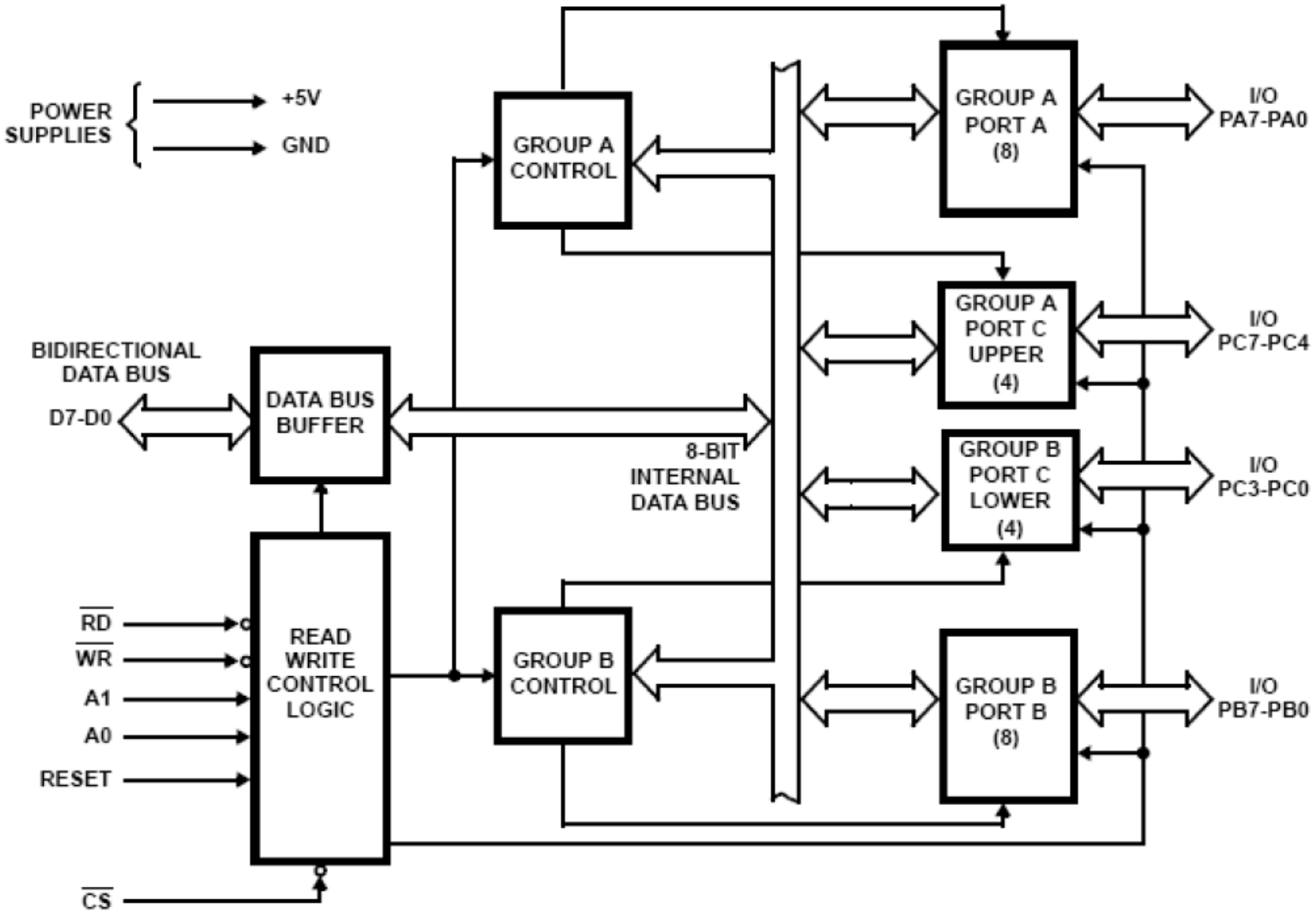
Fig: Double Handshaking

Chapter 2.2 Intel 8255A

- The Intel 8255 A is
  - general purpose programmable I/O device
  - designed for use with Intel microprocessors
  - 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.
- **The 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.

Chapter 2.2 Intel 8255A (Contd…)

Block diagram of 8255:



Chapter 2.2 Intel 8255A (Contd…)

- Main blocks in 8255 are:

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.

Chapter 2.2 Intel 8255A (Contd…)

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 ($C_3 – C_0$)

Chapter 2.2 Intel 8255A (Contd…)

- **8255 Control Word**
- When A0 and A1 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 (D7)** 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 **D7=1**, bits D6-D0 determine **I/O functions** in various modes as shown in figure.
- If bit **D7=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 A1 and A0.
  - Write a control word in control register.
  - Write I/O instructions to communicate with peripherals through Ports A, B and C.

Chapter 2.2 Intel 8255A (Contd...)

- **I/O Control Word**

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

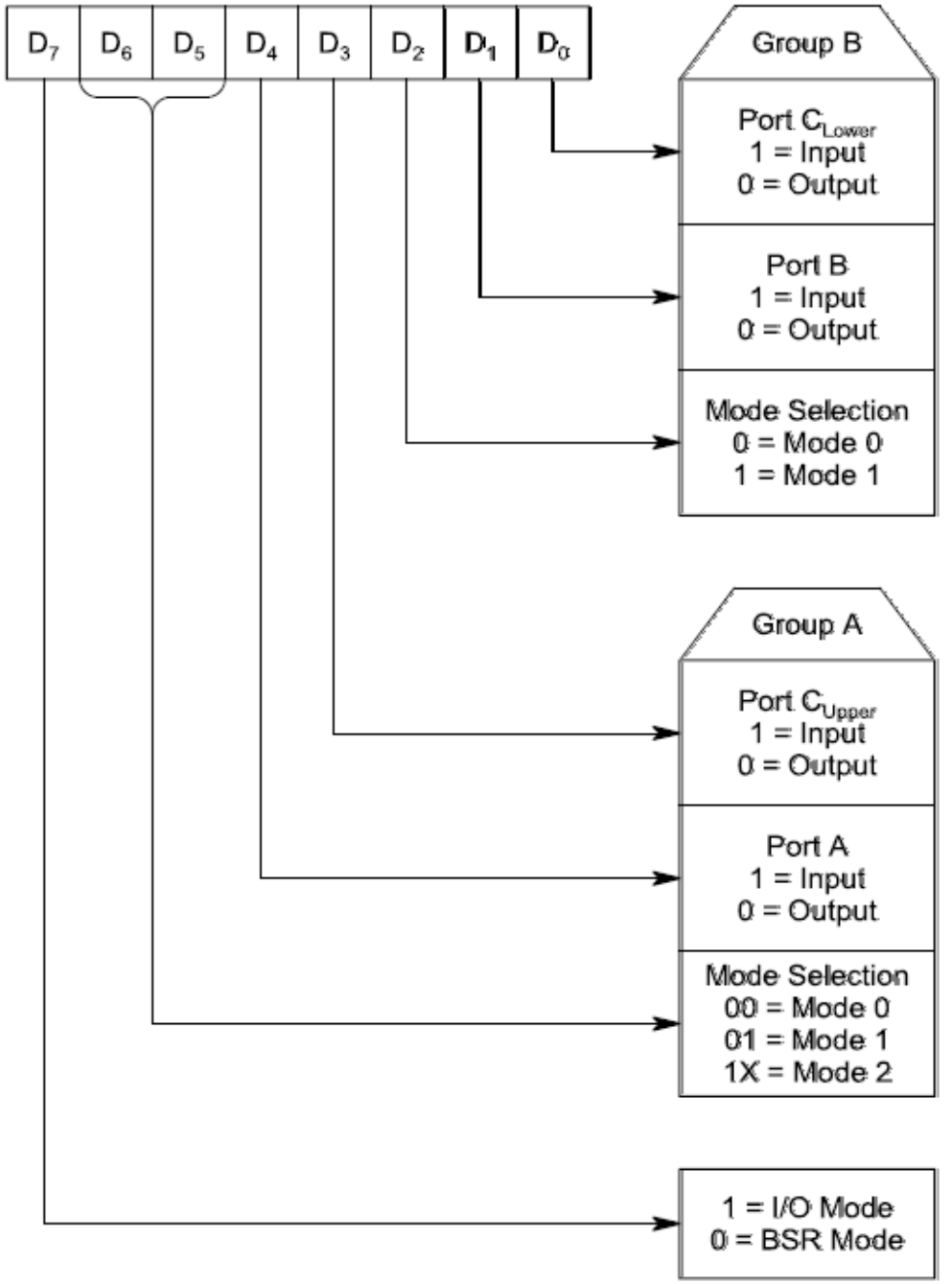
a. Port A output, mode of port A mode 1, port B outpu mode of port B mode 0, port C lower pins as output an remaining pins of port C upper as output.

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

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

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

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



8255A Control Word Format for I/O Mode

Chapter 2.2 Intel 8255A (Contd…

- **BSR Control Word**

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

  Q. Determine the BSR control work for following Port C configurations:

  a. Reset PC7

  b. Set PC3

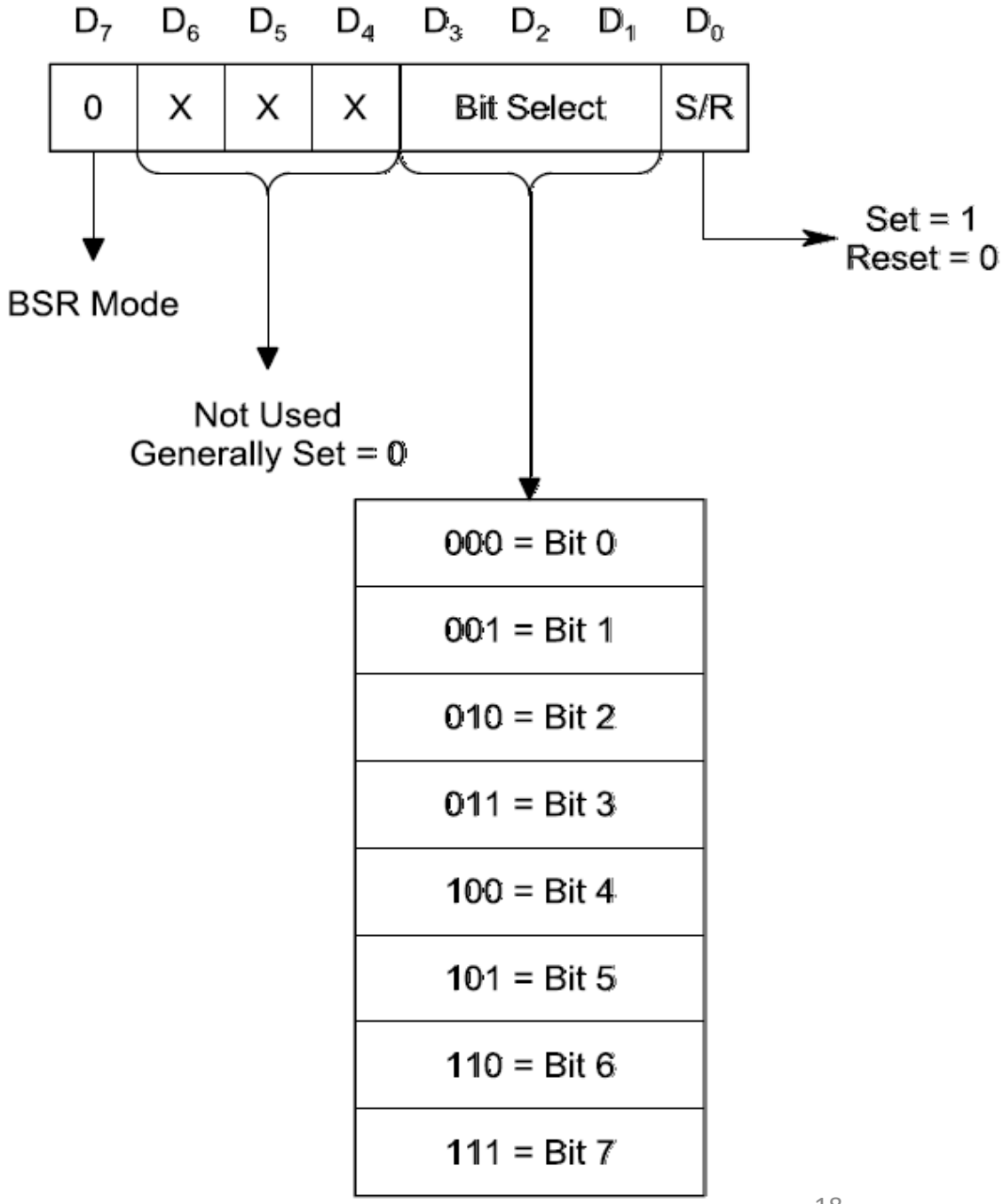


Fig: 8255A Control Word Format for BSR Mode

Chapter 2.2 Intel 8255A (Contd…)

- **Operating Modes of Intel 8255A**

1. 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 D7=0 is recognized as a control word and it does not alter any previously transmitted control word with bit D7=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.

2. 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.

3. 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.

4. 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)

Chapter 2.2 Intel 8255A (Contd…)

- **8255 Programming and Operation**
- **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, D7 is set to '1' (to define a mode set operation) and D6, D5, and D2 are all set to '0' configure all the ports in Mode 0 operation.

➢ The status of bits D4, D3, D1 and D0 then determine whether the corresponding ports are to be configured as Input or Output.

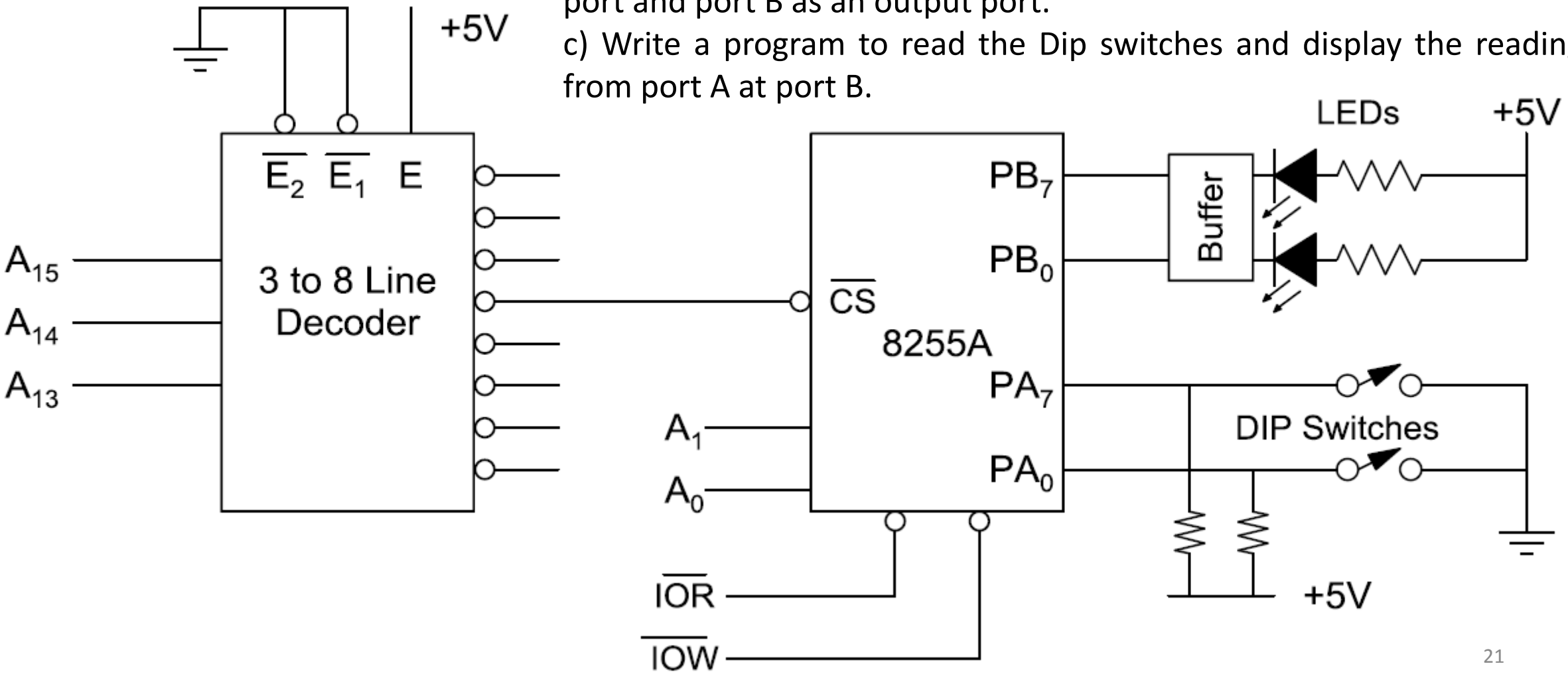
Chapter 2.2 Intel 8255A (Contd…)

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.



Chapter 2.2 Intel 8255A (Contd…)

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

Chapter 2.2 Intel 8255A (Contd…)

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
```

One example of Mode 0 programming from Gaonkar's book

Chapter 2.2 Intel 8255A (Contd…)

- **Programming in BSR Mode**
- Any of the eight bits of port C can be set 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 $D_7 = 0$ is recognized as BSR control word and does not affect the I/O operations of Port A and B.

Chapter 2.2 Intel 8255A (Contd…)

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 some delay.

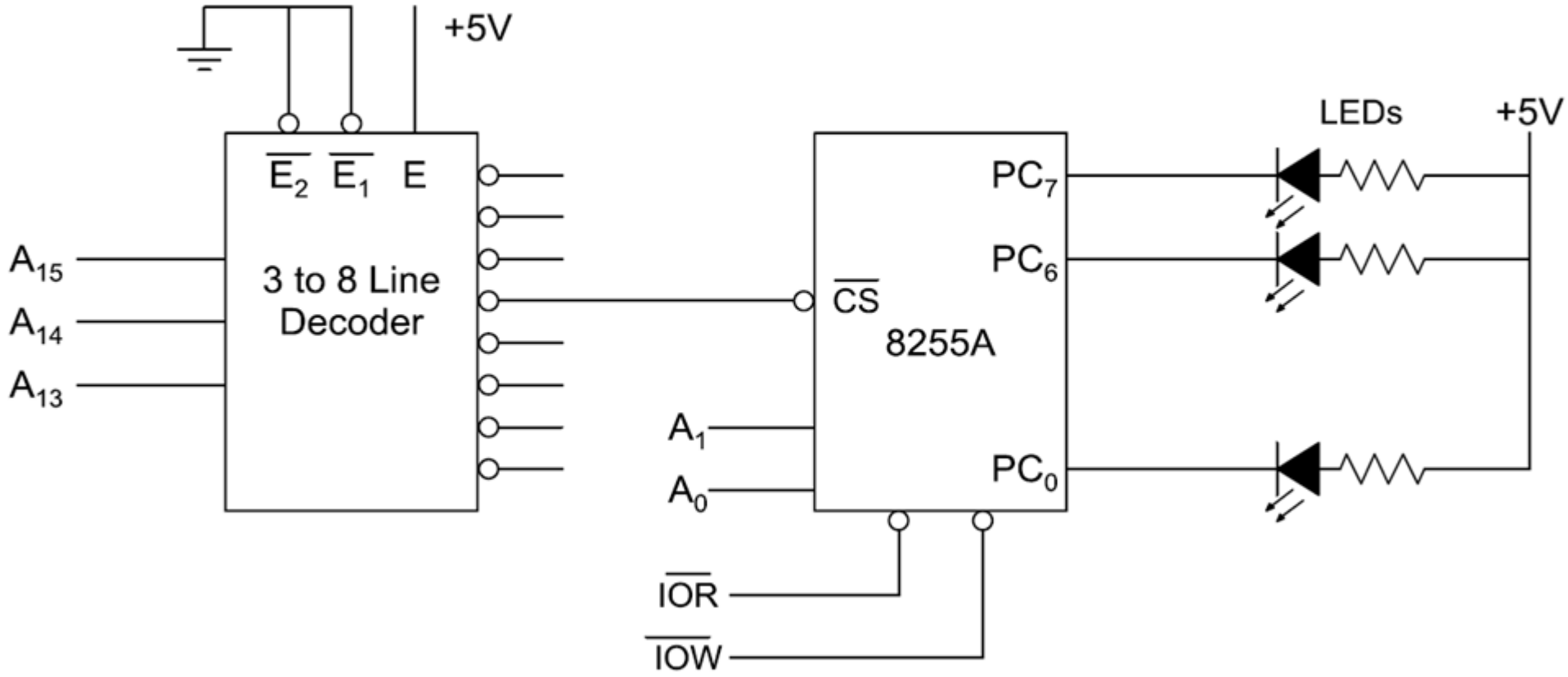


Fig: Example of BSR Mode

Chapter 2.2 Intel 8255A (Contd…)

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

- One example of BSR mode programming from Gaonkar’s book

Chapter 2.2 Intel 8255A (Contd…)

- **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_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 to be programmed as an output port
  - PC_3, PC_6 and PC_7 are used for control
  - PC_4 and PC_5 can be Input or Output, as programmed by bit D_3 (C_{upper}) of the control word.

Chapter 2.2 Intel 8255A (Contd…)

- **Programming in Mode 1 (Strobe I/O Mode)**

- When Port B is programmed as an input (or output) port
  - PC_0 – PC_2 are used for control
  - PC_4 and PC_5 can be Input or Output, as programmed by bit D_3 (C_{upper}) of the control word