Experiment #9

8255 PPI Applications

Birzeit University
Faculty of Engineering and Technology
Department of Electrical and Computer Engineering

Abstract

This experiment aims to use 82C55 Programmable Peripheral Interface (PPI) to interface and test 7-segment display with an 8086 Microprocessor.

The 8086 subsystem will be used to control the display of digits, and characters. The display element is:

- A single 7-segment display

PART I Technical Introduction

1.1 Introduction to 8255A PPI (Programmable Peripheral Interface)

8255A is widely used programmable parallel I/O device. It can be programmed to transfer data under various conditions, from simple I/O to interrupt I/O. It is flexible, versatile and economical (when multiple I/O ports are required), but somewhat complex. It is an important general purpose I/O device that can be used with almost any microprocessor.

The 8255A has 24 input output pins that can be grouped primarily in two 8 bits parallel ports: A and B, with the remaining 8 bits as port C. The 8 bits of port C can be used as individual bits or be grouped in two four bits ports: Cupper (CU) and Clower (CL) as in figure (a). The functions of these ports are defined by writing a control word in the control registers.

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

Group A and Group B Controls
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.

Ports A, B, and C
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 features or "personality" to further enhance the power and flexibility of the 8255.
Port A One 8-bit data output latch/buffer and one 8-bit data input latch. Both "pull-up" and "pull-down" bus-hold devices are present on Port A.
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.

Modes of Operation

The 8255A is primarily operated in two modes: I/O (input-output) mode and the BSR (Bit-Set-Reset) mode. The I/O mode is further grouped into Mode 0 (Simple I/O interfacing), Mode 1 (Interfacing with handshake signals) and Mode 2 (Bidirectional I/O interfacing).

Figure (b) shows all the functions of 8255A, classifying according to two modes: the Bit Set-Reset (BSR) mode and Input Output (I/O) mode. The BSR mode is used to set or reset the bits in port C. 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 ports B use bits from port C as handshake signals. In the handshake mode, two types of data transfer can be implemented: status check and interrupt. In mode 2, port A can be set up for bidirectional data transfer using handshake signal from port C, and port B can be set up either in mode 0 or mode 1.

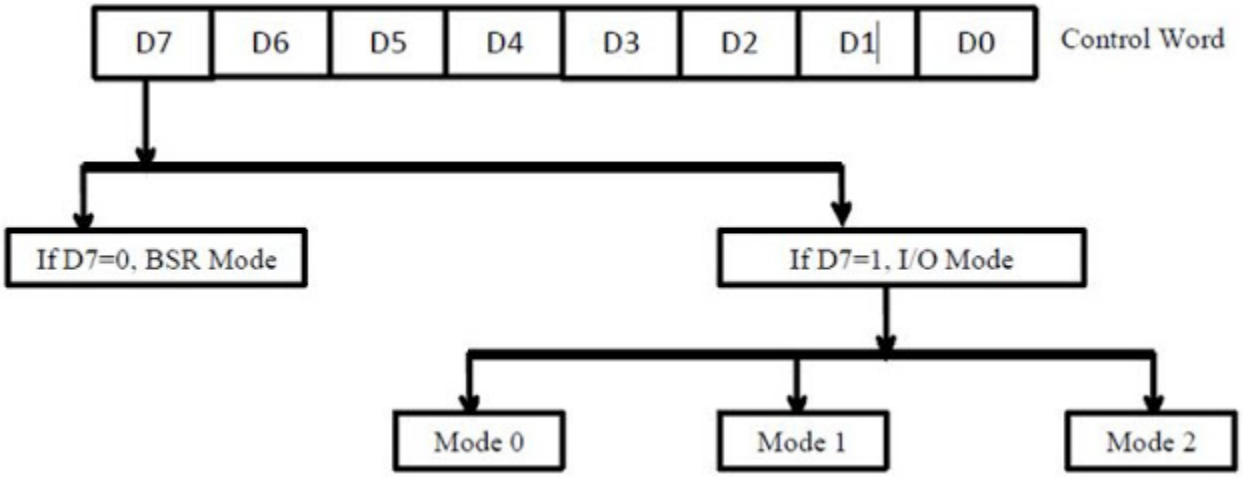

Fig (b): Control word specifying various modes

When D7=0, BSR mode

- For port C
- No effect on I/O mode and functions of port A and B
- Individual bits of port C can be used for applications such as ON/OFF switch

When D7=1, I/O mode

i) Mode 0

- Simple I/O interfacing for port A, B and C

ii) Mode 1

- Interfacing with handshake signals for port A and B
- Port C bits are used for handshake

iii) Mode 2

- Bidirectional I/O interfacing for port A
- Port B: either in mode 0 or mode 1
- Port C bits used for handshake

Control Word

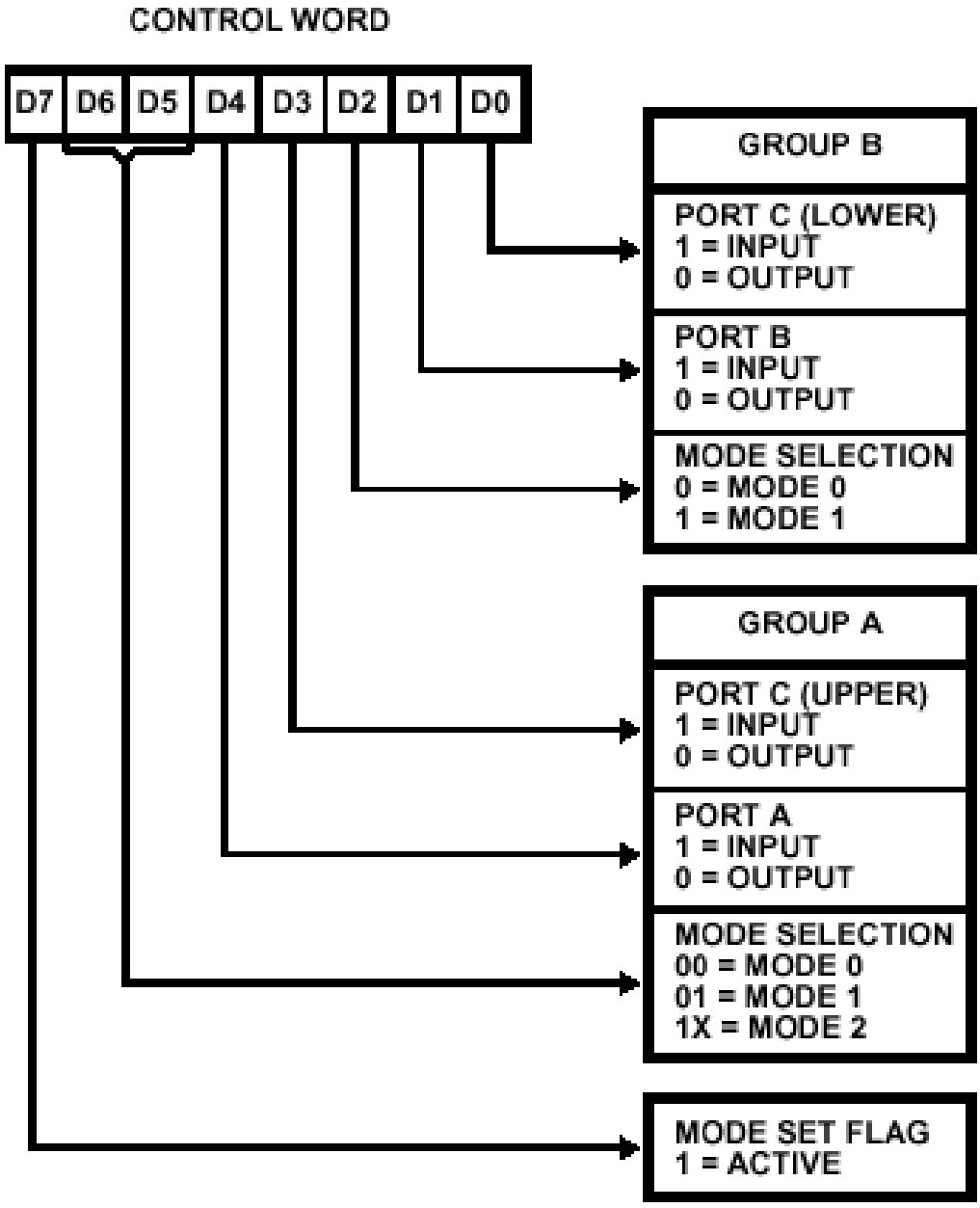

Fig: I/O Mode Definition Control Word Format

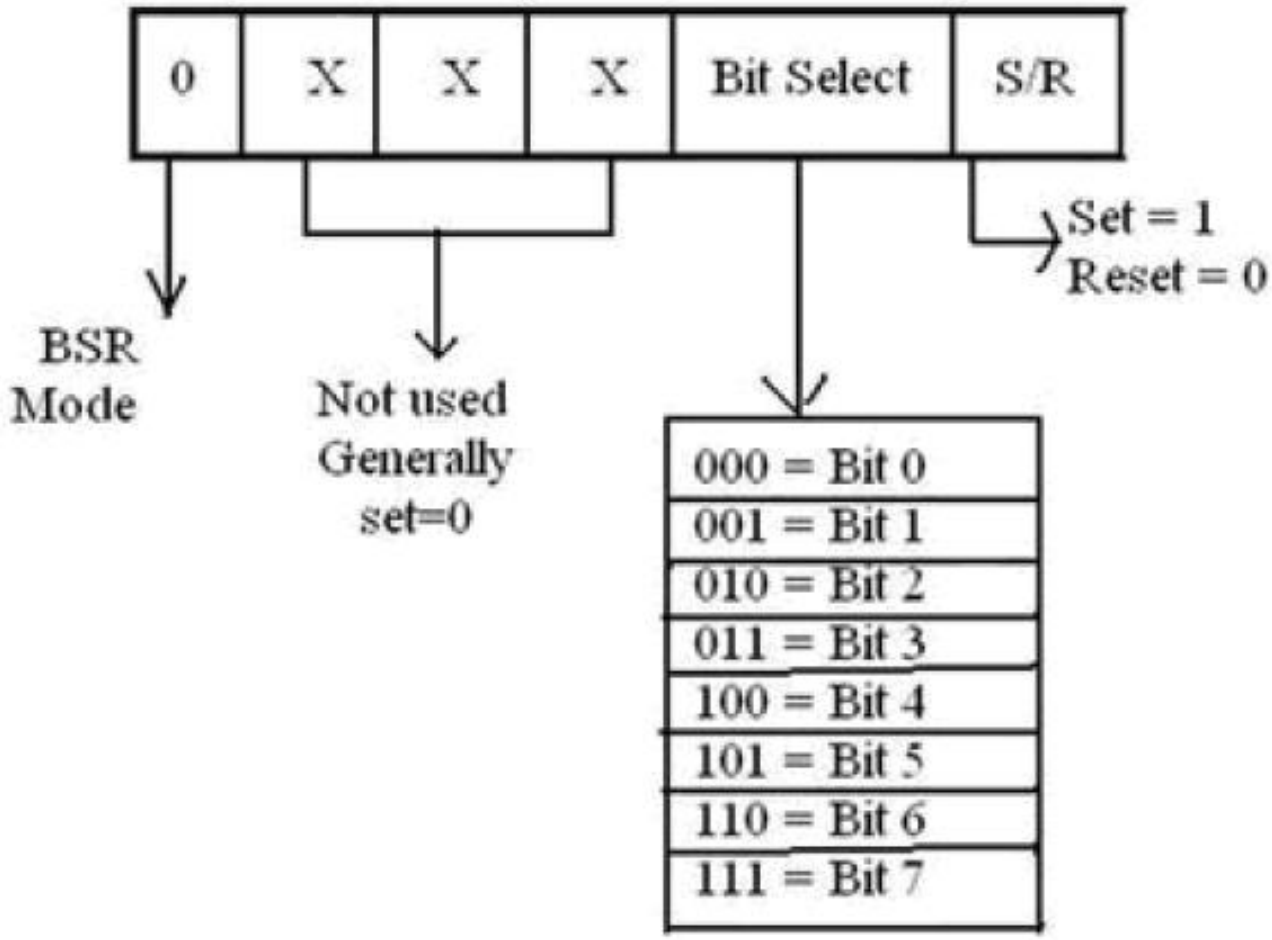

Fig. 6.3 BSR Mode of 8255

Fig: BSR Mode

- The content of control register is called control word specifying an input output functions for each port.
- The register can be accessed to write a control word when A0 and A1 are at logic 1. The register is not accessible for read operation.
- Bit D7 of the control register specifies either I/O functions or Bit Set-Reset function as classified in figure (b).
- If bit D7=1, bit D6-D0 determine I/O function in various modes as shown in figure (b).
- If bit D7=0, port C operates in Bit Set-Reset mode.
- The BSR control word does not affect the function of port A and port B.

PART II Pre-Lab

(This part should be handed on to the teaching assistant in your Lab)

Study the schematics shown below (Figure 3) for the interface between the 82C55 PPI and the 7-segment display.

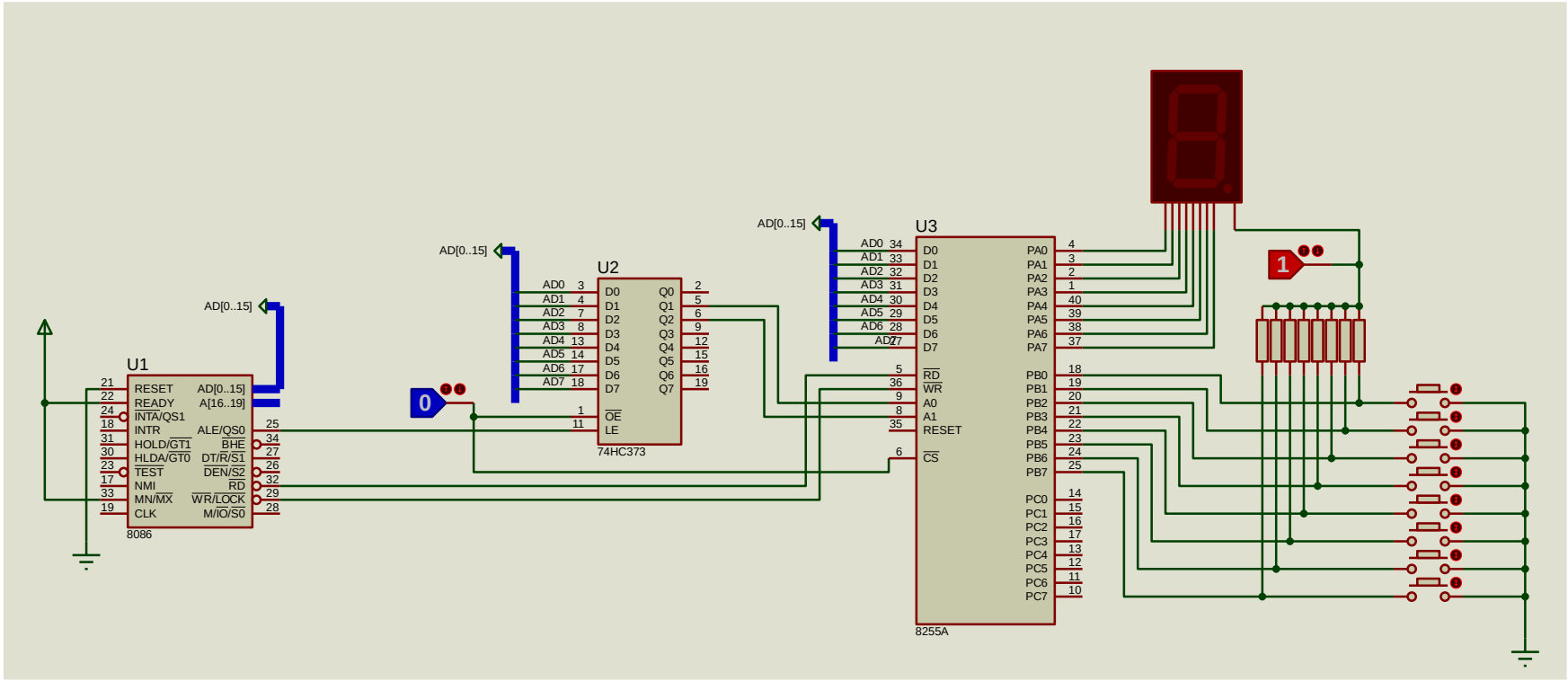

Figure 3 PPI Schematics

PART III Practices

3.1 PRACTIC : SEVEN-SEGMENT DISPLAY

The display consists of seven active low segments (A, B, C, D, E, F, G) and an active low decimal point (P).

The segments and P are driven by the 82C55 Port-A signals as follow:

Segment	82C55 Port-A
A	PA0
B	PA1
C	PA2
D	PA3
E	PA4
F	PA5
G	PA6
P	PA7

The schematic above (Figure 3) shows the interface circuitry between the 82C55 and the 7-segment display.

Code:

```
DATA SEGMENT
 PORTA EQU 00H
 PORTB EQU 02H
 PORTC EQU 04H
 PORT_CON EQU 06H
DATA ENDS
CODE SEGMENT
 MOV AX,DATA
 MOV DS, AX
```

```
ORG 0000H
START:

 MOV DX, PORT_CON
 MOV AL, 10000010B; port C (output), port A (output) in mode 0 and PORT B (INPUT) in mode 0
 OUT DX, AL

 MOV AL, 11000000B
 MOV DX, PORTA
 OUT DX,AL
 JMP XX
XX:
 MOV DX, PORTB
 IN AL, DX
 CMP AL, 11111110B
 JZ S0
 IN AL, DX
 CMP AL, 11111101B
 JZ S1
 IN AL, DX
 CMP AL, 11111011B
 JZ S2
 IN AL, DX
 CMP AL, 11110111B
 JZ S3
 IN AL, DX
 CMP AL, 11101111B
 JZ S4
 IN AL, DX
 CMP AL, 11011111B
 JZ S5
 IN AL, DX
 CMP AL, 10111111B
 JZ S6
 IN AL, DX
 CMP AL, 01111111B
 JZ S7
 JMP XX


S0:
 MOV AL, 11000000B
```

```
 MOV DX, PORTA
 OUT DX,AL
 JMP XX
S1:
 MOV AL, 11111001B
 MOV DX, PORTA
 OUT DX,AL
 JMP XX
S2:
 MOV AL, 10100100B
 MOV DX, PORTA
 OUT DX,AL
 JMP XX
S3:
 MOV AL, 10110000B
 MOV DX, PORTA
 OUT DX,AL
 JMP XX
S4:
 MOV AL, 10011001B
 MOV DX, PORTA
 OUT DX,AL
 JMP XX
S5:
 MOV AL, 10010010B
 MOV DX, PORTA
 OUT DX,AL
 JMP XX
S6:
 MOV AL, 10000010B
 MOV DX, PORTA
 OUT DX,AL
 JMP XX
S7:
 MOV AL, 11111000B
 MOV DX, PORTA
 OUT DX,AL
 JMP XX
JMP START
CODE ENDS
END
```

TO DO: Write an assembly code that will count from 0 to 9 and wraps back to 0. Use the 7-segment display to demo your code. It should look something like (Figure 6):

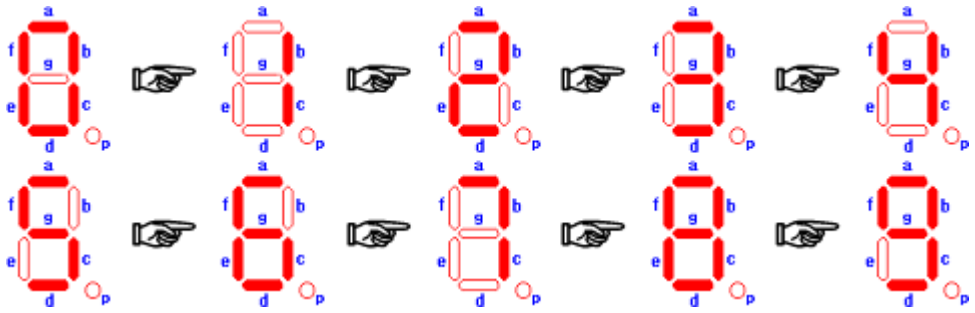

Figure 6 Output of TODO 3

Hint: use the following partial set of data to turn on/off certain segments for every displayed digit:

```
DATA:  DB      11000000B ; “P G F E D C B A” values for number 0
       DB      11111001B
       DB      10100100B
       DB      10110000B
       DB      10011001B
       DB      10010010B
       DB      10000010B
       DB      11111000B
       DB      10000000B
       DB      10010000B
       DB      00H
```