

CSE-3108

Spring, 2016

Interfacing with 7-Segment FND through 8255

Task: Write a Program to display the digits in decimal from 0-7

- **Objective:** Write a program to display the digits in decimal, from 0-7 into 7-segment.
- **7-Segment Display**
- The 7 segment *FND*(Flat Numeric Display) inside the *MDA – 8086* trainer kit can be used to display numbers.
- This requires *PIO-8255* ports which are already connected to the 7 segment internally. You don't have to manually setup a connection between the *FND* and *8255A*
- By executing proper instructions, we can access these ports and provide binary or hex value to switch the required segment on and off.
- The 7-segment display is a common anode display. Hence the segments are active low. In order to turn a segment ON, we need to set logical **0** at the corresponding port.

Introduction to 8255

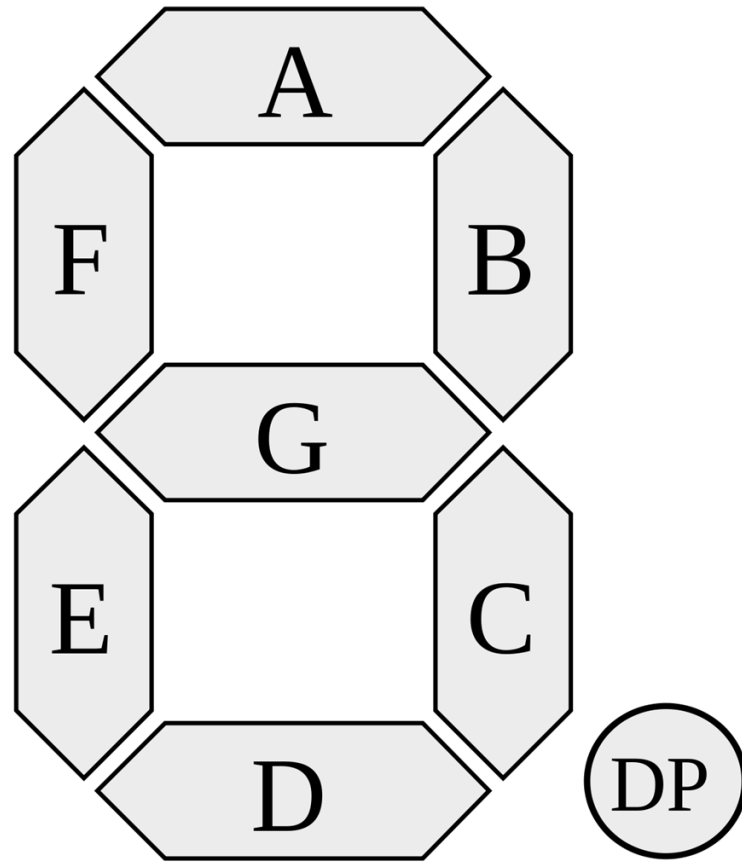
- The Intel 8255 Programmable Peripheral Interface (PPI) chip is a peripheral chip originally developed for the Intel 8085 microprocessor
- 8255 has 40 pins in total, for interfacing with other devices, power supply, chip select etc.
- 24 of these pins are intended towards I/O operations
- These pins are divided into three 8-bit ports[Port A, Port B, Port C]
- Port A, Port B can be used as 8-bit I/O ports
- Port C can either be used as a 8-bit I/O port, or two 4-bit I/O ports.
- Port C can also be used for producing handshaking signals during handshake data transfer

PA3	1		40	PA4
PA2	2		39	PA5
PA1	3		38	PA6
PA0	4		37	PA7
\overline{RD}	5		36	\overline{WR}
\overline{CS}	6		35	RESET
gnd	7		34	D0
A1	8		33	D1
A0	9		32	D2
PC7	10	8255	31	D3
PC6	11	PPI	30	D4
PC5	12		29	D5
PC4	13		28	D6
PC0	14		27	D7
PC1	15		26	Vcc
PC2	16		25	PB7
PC3	17		24	PB6
PB0	18		23	PB5
PB1	19		22	PB4
PB2	20		21	PB3

8255A ports

Introduction to 8255

- The three ports are further grouped as follows-
 1. Group A : Port A and upper part of Port C
 2. Group B: Port B and lower part of Port C
- 8255 contains four registers. One for each of Port A, B, C; and another one called 'Control Word Register'(CWR) for storing the current control state of 8255. CWR stores bits which denote information like whether Port A, Port B, Port C upper or Port C lower are in input mode or output mode.
- Eight data lines (D0-D7) are available (with an 8-bit data buffer) to read/write data into the ports or control register
- The address lines A1 and A0 allow to successively access any one of the ports or the control register
- Port A, B, C are connected with external devices, and data lines(D0-D7) and address lines(A0, A1) are connected with the microprocessor



Segment Labels of a 7-segment Display

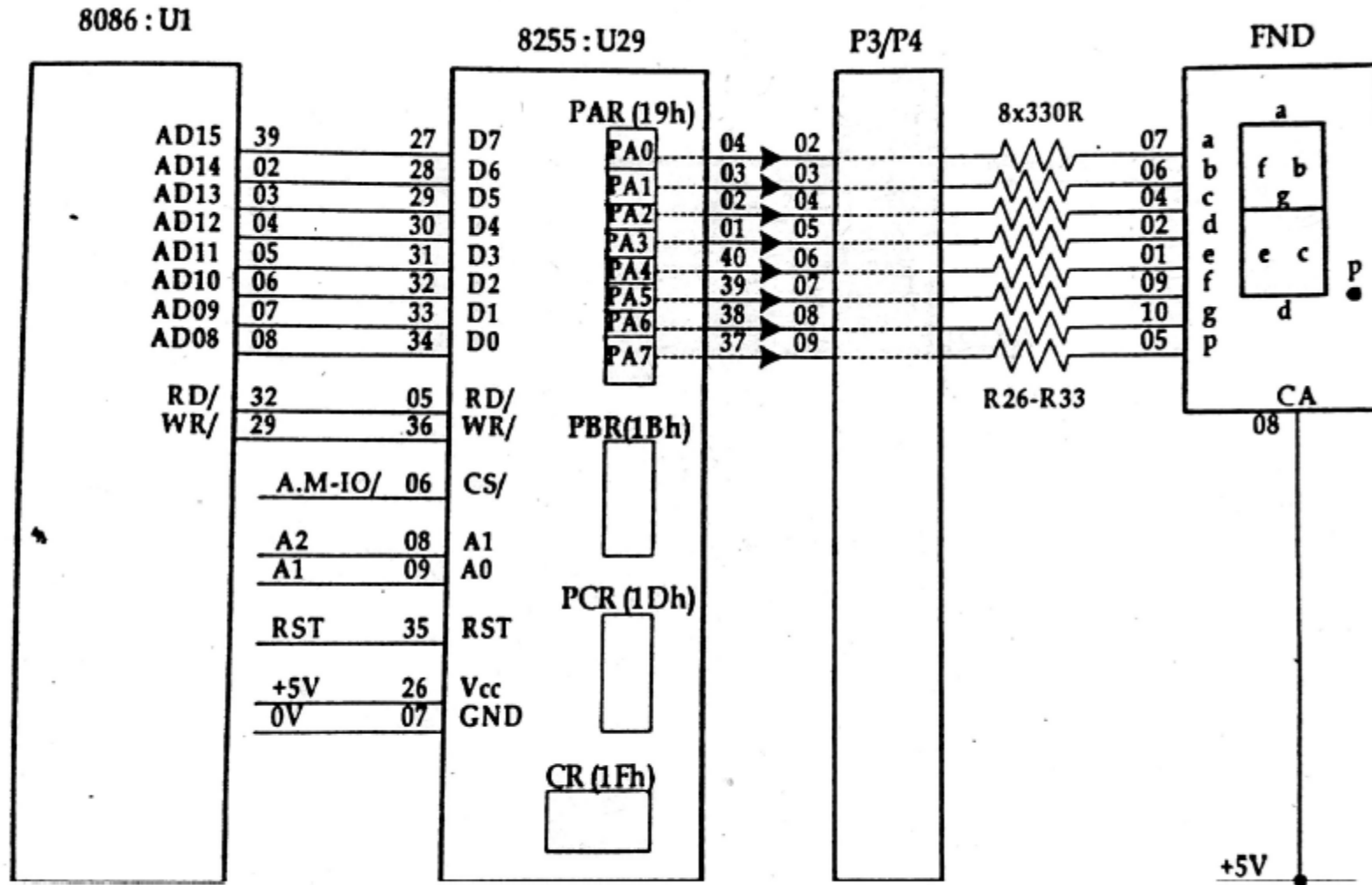


Figure-41.3: Interconnection among 8086, 8255(U29), FND

Write a Program to display the digits in decimal from 0-7

- It's obvious from the last slide that we have to output a specific value to Port A to display a specific number
- Hence, Port A must be in output mode
- Thus, initially, we have to send a control word to the control register to set Port A in output mode
- How do we do that?
- We have to send a proper mode definition control word to the control register. Now Port A is in output mode
- After that, we can output anything to Port A and light the segments of FND correspondingly

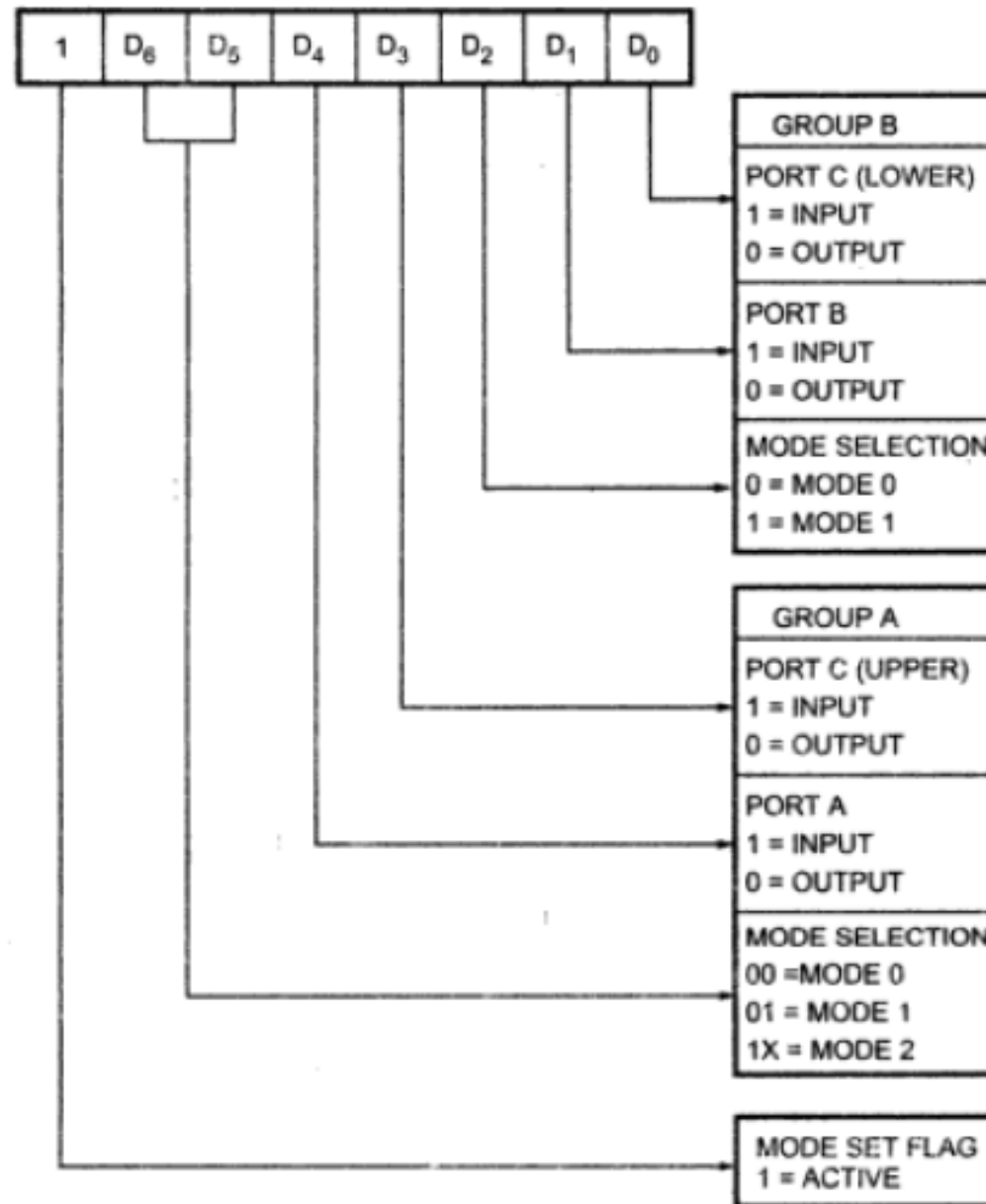
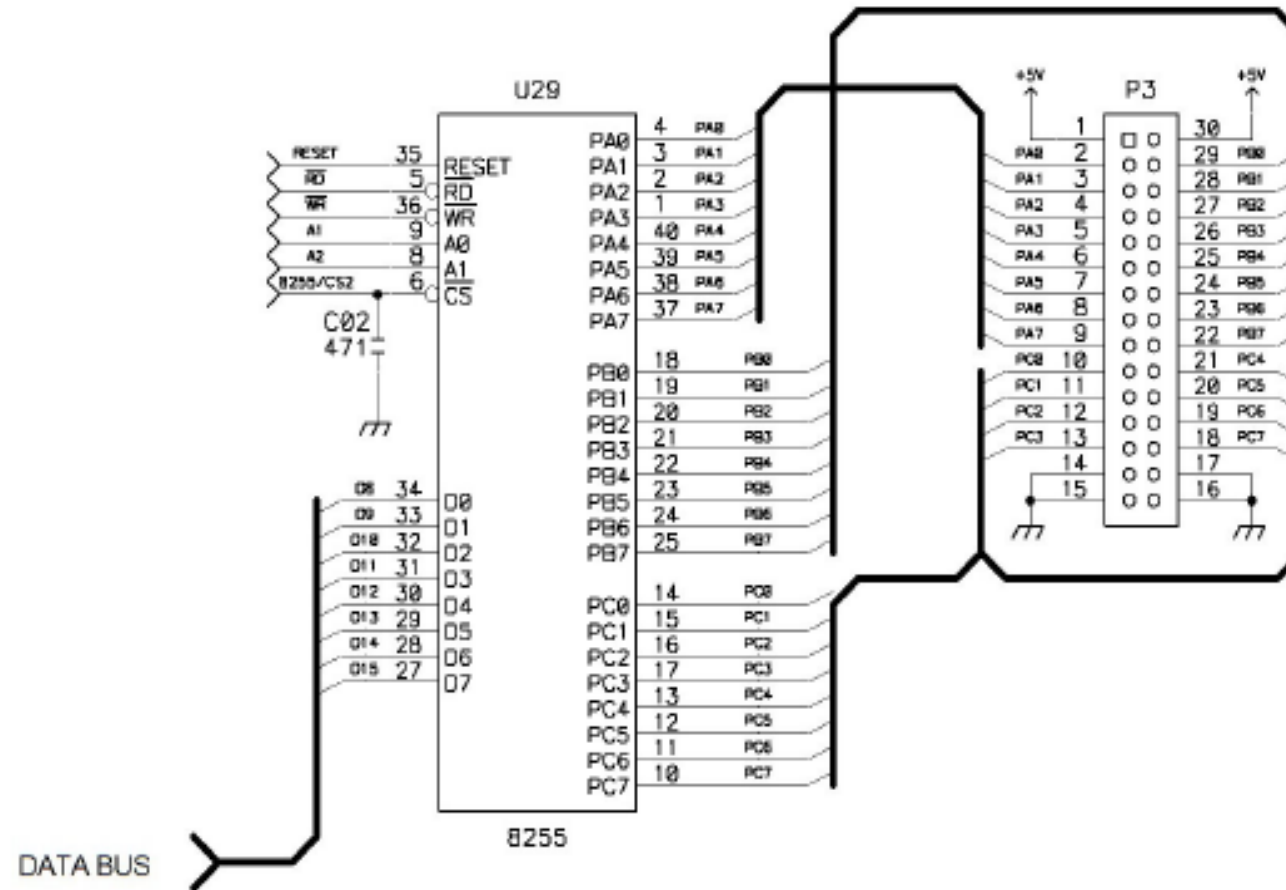


Fig. 7.5 8255 Mode definition format

Experiment 1. 8255A Interface

1-1. LED & 7-Segment



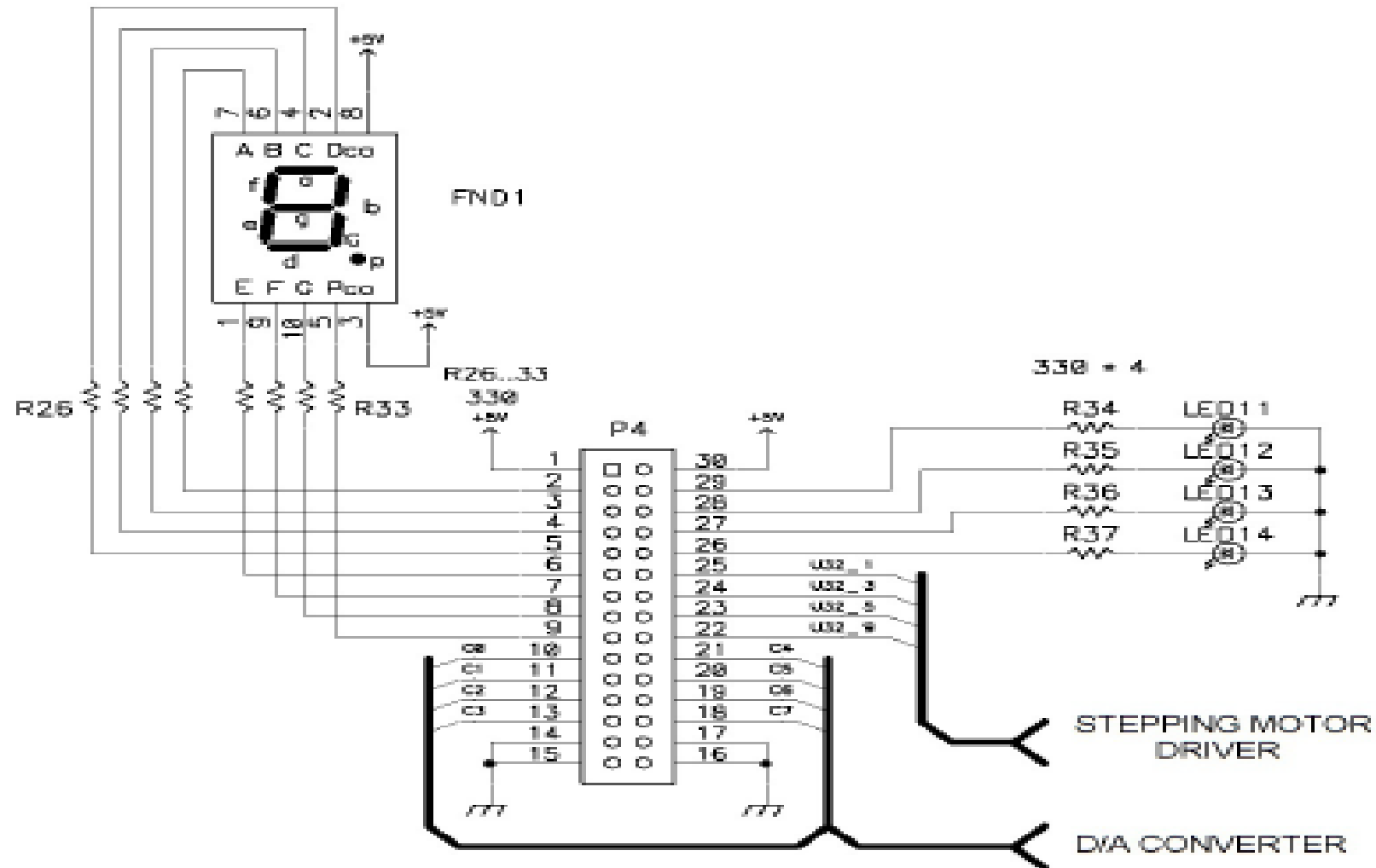
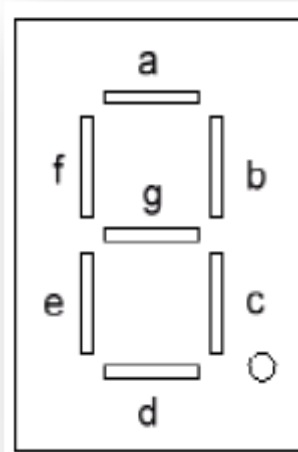


FIGURE 1.1 8255A INTERFACE



- Any number from 0 – 9 can be display on the 7 segment by providing the actual hex or binary value which turns those segments ON to display the digit.

8	4	2	1	8	4	2	1		
dp	g	f	e	d	c	b	a	DECIMAL VALUE	HEX VALUE
0	0	0	0	0	0	0	0	0	00
1	1	1	1	1	0	0	1	1	F9
1	0	1	0	0	1	0	0	2	A4
1	0	1	1	0	0	0	0	3	B0
1	0	0	1	1	0	0	1	4	99
1	0	0	1	0	0	1	0	5	92
1	0	0	0	0	0	1	0	6	82
1	1	1	1	1	0	0	0	7	F8
1	0	0	0	0	0	0	0	8	80
1	0	0	1	0	0	0	0	9	90

Task 1 - Display the digit 5 on the FND

- Try out the following code

```

CODE SEGMENT
    ASSUME CS:CODE
    PPI_Control EQU 1FH ; 'EQU' is similar to macro in C
    PPIA EQU 19H

    ORG 1000H

    MOV AL, 10000000B
    OUT PPI_Control, AL ; Now PortA is in output mode

    MOV AL, 92H ; The bitmask for the digit 5
    OUT PPIA, AL ; Display the digit!
    INT 3

CODE ENDS
END

```

Task 2 – Display the digits 0 through 9 in a loop

- Store the bitmasks for displaying the numbers from 0 to 9 starting from address *1000:0000* through *1000:0009*
- Bitmasks : *C0, F9, A4, B0, 99, 92, 82, F8, 80, 90*
- Then write a program that displays the digits 0 through 9 in the 7 segment FND using the bitmasks currently stored in the RAM

CODE SEGMENT

ASSUME CS:CODE

PPI_Control EQU 1FH

PPIA EQU 19H

ORG 1000H

MOV AL, 10000000B

OUT PPI_Control, AL

MOV AX, 1000H

MOV DS, AX

L2: MOV SI, 0H ; Now, DS:SI = 10000:0000

MOV CX, 0AH

L1: MOV AL, BYTE PTR DS:[SI]

CMP CX, 00H

JE L2

OUT PPIA, AL

CALL TIMER

INC SI

DEC CX

JMP L1

INT 3

; Write the TIMER procedure now

TIMER: PUSH CX

MOV CX, 1000H

TIMER1: NOP

LOOP TIMER1

POP CX

RET

CODE ENDS

END

Result

Lab Objectives:

Task#1: Write a program to display all digits from 0-9 as shown in the following diagram.



Practice

- Display the numbers in a reverse manner, from 9 down-to 0, then 9 again.