Microprocessor & Embedded System Lab Report

by

Ishiat Al Mahmood

CSE 078 08419

Supervised by: Tarikuzzaman Emon

Assistant Professor

Department of Computer Science and Engineering

STAMFORD UNIVERSITY BANGLADESH

20 May 2024

Experiment - 1

Name of the Experiment: Display of Seven Segment.

Theory:

A seven-segment display (SSD) is a common electronic screen showing numbers using seven parts that light up. These parts are made of small lights or other display materials arranged in a specific way. They make a rectangle shape with two vertical parts on each side and one horizontal part on the top, middle, and bottom. There's also a seventh part that cuts across the rectangle, making different shapes to show numbers. In our training board this display uses the port 19H.

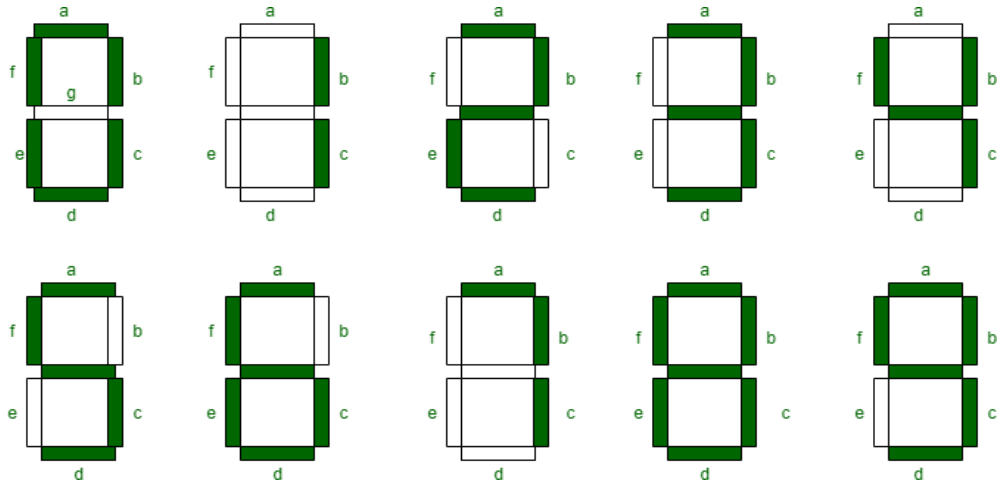

Figure 1: 7 segment display

Hexadecimal encoding to display the digits 0 to 9:

Digits	Hex. Value	H	G	F	E	D	C	B	A
0	0C0h	1	1	0	0	0	0	0	0
1	0F9h	1	1	1	1	1	0	0	1
2	0A4h	1	0	1	0	0	1	0	0
3	0B0h	1	0	1	1	0	0	0	0
4	099h	1	0	0	1	1	0	0	1
5	092h	1	0	0	1	0	0	1	0
6	082h	1	0	0	0	0	0	1	0

7	0F8h	1	1	0	1	1	0	0	0
8	080h	1	0	0	0	0	0	0	0
9	090h	1	0	0	1	0	0	0	0

Requirements:

1. Windows PC
2. Wincom Software
3. MASM Software
4. Notepad++
5. MDA 8086

Code Segment:

```
A SEGMENT PARA PUBLIC 'CODE'
ASSUME CS: A
ORG 1000H

S:
  MOV AL,80H
  OUT 1FH,AL

  MOV AL,082H
  OUT 19H,AL

A ENDS
END S
```

DOS Command:

The following commands are given in the command prompt in order to connect the code segment with the WINCOM.

- Cd \
- Cd MDA
- Cd 8086
- Cd ASM8086
- MASM
- File_name
- File_name
- File_name

- LOD186
- File_name
- File_name

Now we execute the WINCOM software and access the tool kit to initiate a RESET. Upon pressing the RESET button, the PC screen displays machine-generated information. Subsequently, we issue the following commands:

- L
- F3
- File_name
- G

Following the execution of the commands, the desired output becomes visible on the tool kit, indicating successful processing or attainment of the intended result.

Discussion:

This experiment used seven-segment displays to show decimal numbers and accomplished its goal. By keeping files organized and avoiding problems, the process went smoothly. It easily ran the WINCOM software after entering the right commands, giving the expected results without any big problems. One thing to keep in mind how ever was that the value for the “h” bit was always set to one.

Experiment - 2

Name of the Experiment: Turning on the LED.

Theory:

This experiment allows us to turn on the 4 individual LED lights in a 2x2 grid located right under the seven segment display output. The grid features 2 red lights (top left and bottom right) as well as a green and a yellow light (top right and bottom left). The port for the LED display is 1BH.

Hexadecimal encoding to display the LED lights:

LED	Hexadecimal Value	R	Y	G	R
Red	01H	0	0	0	1
Green	02H	0	0	1	0
Yellow	04H	0	1	0	0
red	08H	1	0	0	0

Requirements:

1. Windows PC
2. Wincom Software
3. MASM Software
4. Notepad++
5. MDA 8086

Code Segment:

```
A SEGMENT PARA PUBLIC 'CODE'
ASSUME CS:A
ORG 1000H

S:
    MOV AL,80H
    OUT 1FH,AL
    MOV AL,0FFH
    OUT 19H,AL
```

```
L:
    MOV AL,01H
    OUT 1BH,AL
    JMP L
AENDS
END S
```

DOS Command:

The following commands are given in the command prompt in order to connect the code segment with the WINCOM.

- Cd \
- Cd MDA
- Cd 8086
- Cd ASM8086
- MASM
- File_name
- File_name
- File_name
- LOD186
- File_name
- File_name

Now execute the WINCOM software and access the tool kit to initiate a RESET. Upon pressing the RESET button, the PC screen displays machine-generated information. Subsequently, issue the following commands:

- L
- F3
- File_name
- G

Following the execution of the commands, the desired output becomes visible on the tool kit, indicating successful processing or attainment of the intended result.

Discussion:

The main goal of the experiment was to light up an LED by controlling input/output ports accurately. Port 19H gets a command (0FFH) to turn off the 7-segment display, which suggests a link between the LED and this display. The core of the code is in a loop called 'L,' where continuously sending the value 01H to port 1BH indicates a command to turn on the LED, keeping it lit constantly. The loop keeps going with

the JMP instruction, meaning the LED stays on indefinitely. Although in further testing it was noticed that the seven segment display being off didn't seem to matter in some training boards while in other boards without sending the (0FFH) signal to port 19H the LED display would not function normally.

Experiment - 3

Name of the Experiment: Running the LED clock-wise.

Theory:

This experiment allows us to control the LED lights in such a manner that allows us to create a simple patter that would loop indefinitely. For this specific case, we will explore the methods used to achieve a simple clock-wise motion of the LED lights located at port 1BH.

Hexadecimal encoding to display the LED lights:

LED	Hexadecimal Value	R	Y	G	R
Red	01H	0	0	0	1
Green	02H	0	0	1	0
Yellow	04H	0	1	0	0
red	08H	1	0	0	0

Requirements:

1. Windows PC
2. Wincom Software
3. MASM Software
4. Notepad++
5. MDA 8086

Code Segment:

```
A SEGMENT PARA PUBLIC 'CODE'
ASSUME CS: A
ORG 1000H
 S:
      MOV AL,80H
      OUT 1FH,AL
      MOV AL,0FFH
       OUT 19H,AL
L:
      MOV AL,01H
```

```
        OUT 1BH,AL
        MOV CX,0FFFFH
        L1: LOOP L1
        MOV AL,00H
        OUT 1BH,AL

        MOV AL,02H
        OUT 1BH,AL
        MOV CX,0FFFFH
        L2: LOOP L2
        MOV AL,00H
        OUT 1BH,AL

        MOV AL,08H
        OUT 1BH,AL
        MOV CX,0FFFFH
        L3: LOOP L3
        MOV AL,00H
        OUT 1BH,AL

        MOV AL,04H
        OUT 1BH,AL
        MOV CX,0FFFFH
        L4: LOOP L4
        MOV AL,00H
        OUT 1BH,AL

        JMP L
A ENDS
END S
```

DOS Command:

The DOS command follows the same procedure as previous experiments.

Discussion:

The experiment is aimed to activate LEDs in a clockwise manner using an LED display. Its goal was to allow us to get comfortable with longer codes that create repeating patterns so we could have a better understanding of the 8086 system.

Experiment - 4

Name of the Experiment: Running the LED anti-clock wise.

Theory:

This experiment allows us to control the LED lights in such a manner that allows us to create a simple patter that would loop indefinitely. For this specific case, we will explore the methods used to achieve a simple anti clock-wise motion of the LED lights located at port 1BH.

Hexadecimal encoding to display the LED lights:

LED	Hexadecimal Value	R	Y	G	R
Red	01H	0	0	0	1
Green	02H	0	0	1	0
Yellow	04H	0	1	0	0
red	08H	1	0	0	0

Requirements:

1. Windows PC
2. Wincom Software
3. MASM Software
4. Notepad++
5. MDA 8086

Code Segment:

```
A SEGMENT PARA PUBLIC 'CODE'
ASSUME CS: A
ORG 1000H
S:
    MOV AL,80H
    OUT 1FH,AL
    MOV AL,0FFH
    OUT 19H,AL

L:
    MOV AL,04H
```

```
    OUT 1BH,AL
    MOV CX,0FFFFH
     L1: LOOP L1
    MOV AL,00H
    OUT 1BH,AL

    MOV AL,08H
     OUT 1BH,AL
    MOV CX,0FFFFH
     L2: LOOP L2
    MOV AL,00H
    OUT 1BH,AL

    MOV AL,02H
     OUT 1BH,AL
    MOV CX,0FFFFH
    L3: LOOP L3
    MOV AL,00H
    OUT 1BH,AL

    MOV AL,01H
    OUT 1BH,AL
    MOV CX,0FFFFH
     L4: LOOP L4
    MOV AL,00H
     OUT 1BH,AL

    JMP L
A ENDS
END S
```

DOS Command:

The DOS command follows the same procedure as previous experiments.

Discussion:

The experiment is aimed to activate LEDs in an anti clockwise manner using an LED display. Its goal was to allow us to get comfortable with longer codes that create repeating patterns so we could have a better understanding of the 8086 system.

Experiment - 5

Name of the Experiment: Traffic Light

Theory:

This experiment explores the idea of repeating patterns further. This time by using both the LED display and the seven segment display we can create a simulated traffic light pattern that goes from red to yellow to green and counts down the time between the switching of the lights through the seven segment display.

Hexadecimal encoding to display the digits 0 to 9:

LED	Hexadecimal Value	R	Y	G	R
Red	01H	0	0	0	1
Green	02H	0	0	1	0
Yellow	04H	0	1	0	0
red	08H	1	0	0	0

Hexadecimal encoding to display the digits 0 to 9:

Digits	Hex. Value	H	G	F	E	D	C	B	A
0	0C0h	1	1	0	0	0	0	0	0
1	0F9h	1	1	1	1	1	0	0	1
2	0A4h	1	0	1	0	0	1	0	0
3	0B0h	1	0	1	1	0	0	0	0
4	099h	1	0	0	1	1	0	0	1
5	092h	1	0	0	1	0	0	1	0
6	082h	1	0	0	0	0	0	1	0
7	0F8h	1	1	0	1	1	0	0	0
8	080h	1	0	0	0	0	0	0	0
9	090h	1	0	0	1	0	0	0	0

Requirements:

1. Windows PC
2. Wincom Software
3. MASM Software
4. Notepad++
5. MDA 8086

Code Segment:

```
A SEGMENT PARA PUBLIC 'CODE'
ASSUME CS: A
ORG 1000H
 S:
MOV AL, 80H
 OUT 1FH, AL
MOV AL, 0FFH
OUT 19H, AL

L:
MOV AL, 01H
OUT 1BH, AL
MOV CX, 0FFFFH
L1: LOOP L1
MOV AL, 090H
 OUT 19H, AL
MOV CX,0FFFFH

LA: LOOP LA
MOV AL, 090H
OUT 19H, AL
 MOV CX,0FFFFH

L2: LOOP L2
MOV AL, 0FFH
OUT 19H, AL
MOV AL, 080H
 OUT 19H, AL
MOV CX, 0FFFFH
 L2A: LOOP L2A
MOV AL, 0FFH
OUT 19H, AL
 MOV AL, 080H
OUT 19H, AL
MOV CX, 0FFFFH

L3: LOOP L3
```

```
MOV AL, 0FFH
 OUT 19H, AL
MOV AL, 0D8H
OUT 19H, AL
MOV CX, 0FFFFH
L3A: LOOP L3A
MOV AL, 0FFH
OUT 19H, AL
 MOV AL, 0D8H
OUT 19H, AL
MOV CX, 0FFFFH

L4: LOOP L4
MOV AL, 0FFH
OUT 19H, AL
MOV AL, 082H
OUT 19H, AL
MOV CX, 0FFFFH
L4A: LOOP L4A
MOV AL, 0FFH
OUT 19H, AL
MOV AL, 082H
OUT 19H, AL
 MOV CX, 0FFFFH
L5: LOOP L5
MOV AL, 0FFH
 OUT 19H, AL
MOV AL, 092H
OUT 19H, AL
MOV CX, 0FFFFH
L5A: LOOP L5A
MOV AL, 0FFH
OUT 19H, AL
MOV AL, 092H
OUT 19H, AL
 MOV CX, 0FFFFH
L6: LOOP L6
MOV AL, 0FFH
 OUT 19H, AL
MOV AL, 099H
```

```
OUT 19H, AL
MOV CX, 0FFFFH
 L6A: LOOP L6A
MOV AL, 0FFH
OUT 19H, AL
MOV AL, 099H
OUT 19H, AL
MOV CX, 0FFFFH
L7: LOOP L7
MOV AL, 0FFH
OUT 19H, AL
MOV AL, 0B0H
 OUT 19H, AL
MOV CX, 0FFFFH
 L7A: LOOP L7A
MOV AL, 0FFH
OUT 19H, AL
 MOV AL, 0B0H
OUT 19H, AL
 MOV CX, 0FFFFH

L8: LOOP L8
MOV AL, 0FFH
OUT 19H, AL
MOV AL, 0A4H
OUT 19H, AL
MOV CX, 0FFFFH
L8A: LOOP L8A
MOV AL, 0FFH
OUT 19H, AL
MOV AL, 0A4H
OUT 19H, AL
MOV CX, 0FFFFH
L9: LOOP L9
MOV AL, 0FFH
 OUT 19H, AL
MOV AL, 0F9H
 OUT 19H, AL
MOV CX, 0FFFFH
L9A: LOOP L9A
```

```
MOV AL, 0FFH
OUT 19H, AL

MOV AL, 0F9H
OUT 19H, AL

MOV CX, 0FFFFH
L10: LOOP L10
MOV AL, 0FFH
 OUT 19H, AL
MOV AL, 0C0H
OUT 19H, AL
MOV CX, 0FFFFH
L10A: LOOP L10A
MOV AL, 0FFH
OUT 19H, AL
MOV AL, 0C0H
OUT 19H, AL
MOV CX, 0FFFFH

L11: LOOP L11
 MOV  AL,  0FFH
 OUT 19H, AL
MOV AL, 00H
 OUT 1BH, AL
MOV CX, 0FFFFH

L12: LOOP L12
MOV  AL,  04H
 OUT 1BH, AL
L12A: LOOP L12A
MOV  AL,  04H
OUT 1BH, AL
L12B: LOOP L12B
MOV  AL,  04H
OUT 1BH, AL
 L12C: LOOP L12C
MOV  AL,  04H
OUT 1BH, AL
MOV CX, 0FFFFH
L13: LOOP L13
```

```
MOV AL, 00H
 OUT 1BH, AL
MOV CX, 0FFFFH
L14: LOOP L14
MOV AL, 02H
 OUT 1BH, AL
MOV CX, 0FFFFH

L15: LOOP L15
MOV AL, 090H
 OUT 19H, AL
MOV CX, 0FFFFH
L15A: LOOP L15A
MOV AL, 090H
OUT 19H, AL
 MOV CX, 0FFFFH

L16: LOOP L16
MOV AL, 0FFH
OUT 19H, AL
MOV AL, 080H
OUT 19H, AL
MOV CX, 0FFFFH
L16A: LOOP L16A
MOV AL, 0FFH
OUT 19H, AL
 MOV AL, 080H
OUT 19H, AL
MOV CX, 0FFFFH

L17: LOOP L17
MOV AL, 0FFH
OUT 19H, AL
MOV AL, 0D8H
OUT 19H, AL
MOV CX, 0FFFFH
 L17A: LOOP L17A
MOV AL, 0FFH
OUT 19H, AL
MOV AL, 0D8H
OUT 19H, AL
```

```
MOV CX, 0FFFFH

L18: LOOP L18
MOV AL, 0FFH
OUT 19H, AL
MOV AL, 082H
OUT 19H, AL
MOV CX, 0FFFFH
L18A: LOOP L18A
MOV AL, 0FFH
OUT 19H, AL
 MOV AL, 082H
OUT 19H, AL
MOV CX, 0FFFFH
L19: LOOP L19
MOV AL, 0FFH
OUT 19H, AL
MOV AL, 092H
OUT 19H, AL
MOV CX, 0FFFFH
 L19A: LOOP L19A
MOV AL, 0FFH
OUT 19H, AL
MOV AL, 092H
OUT 19H, AL
MOV CX, 0FFFFH
L20: LOOP L20
MOV AL, 0FFH
OUT 19H, AL
MOV AL, 099H
OUT 19H, AL
MOV CX, 0FFFFH
 L20A: LOOP L20A
MOV AL, 0FFH
OUT 19H, AL
MOV AL, 099H
OUT 19H, AL
MOV CX, 0FFFFH
```

```
L21: LOOP L21
MOV AL, 0FFH
 OUT 19H, AL
MOV AL, 0B0H
OUT 19H, AL
MOV CX, 0FFFFH
 L21A: LOOP L21A
MOV AL, 0FFH
OUT 19H, AL
 MOV AL, 0B0H
OUT 19H, AL
MOV CX, 0FFFFH
L22: LOOP L22
MOV AL, 0FFH
 OUT 19H, AL
MOV AL, 0A4H
OUT 19H, AL
MOV CX, 0FFFFH
 L22A: LOOP L22A
MOV AL, 0FFH
OUT 19H, AL
 MOV AL, 0A4H
OUT 19H, AL
MOV CX, 0FFFFH

L23: LOOP L23
MOV AL, 0FFH
OUT 19H, AL
MOV AL, 0F9H
OUT 19H, AL
MOV CX, 0FFFFH
L23A: LOOP L23A
MOV AL, 0FFH
OUT 19H, AL
MOV AL, 0F9H
OUT 19H, AL
MOV CX, 0FFFFH
L24: LOOP L24
MOV AL, 0FFH
 OUT 19H, AL
```

```
MOV AL, 0C0H

OUT 19H, AL
MOV CX, 0FFFFH
L24A: LOOP L24A
MOV AL, 0FFH
OUT 19H, AL
MOV AL, 0C0H
OUT 19H, AL
MOV CX, 0FFFFH

L45: LOOP L45
MOV AL, 0FFH
OUT 19H, AL
MOV AL, 00H
OUT 1BH, AL
MOV AL, 01H
OUT 1BH, AL
MOV CX, 0FFFFH

L46: LOOP L46
JMP L
 A ENDS
END S
```

DOS Command:

The DOS command follows the same procedure as previous experiments.

Discussion:

The code for this experiment was the longest by far due to the count down happening twice for red and green lights. Fortunately, when the experiment was conducted the code was written slowly and carefully to avoid any miniscule errors that might cascade into something big. As a result the code ran flawlessly the first time around.

Experiment – 6

Name of the Experiment: Display Wedding Lights.

Theory:

This experiment follows similar trends with the previous experiments while escalating the complexity of the code by introducing more complicated patterns which would require long segments of code and multiple tests to achieve the intended results. In this case we are to create a wedding light pattern which is a combination of different patterns of the LED display located at port 1BH.

Hexadecimal encoding to display the LED lights:

LED	Hexadecimal Value	R	Y	G	R
Red	01H	0	0	0	1
Green	02H	0	0	1	0
Yellow	04H	0	1	0	0
red	08H	1	0	0	0

Requirements:

1. Windows PC
2. Wincom Software
3. MASM Software
4. Notepad++
5. MDA 8086

Code Segment:

```
A SEGMENT PARA PUBLIC 'CODE'
ASS A SEGMENT PARA PUBLIC 'CODE'
ASSUME CS:A
ORG 1000H
S:
      MOV AL,80H
      OUT 1FH,AL
      MOV AL,0FFH
      OUT 19H,AL

L:
      MOV AL,01H
      OUT 1BH,AL
      MOV CX,0FFFFH
      L1: LOOP L1
      MOV AL,00H
      OUT 1BH,AL

      MOV AL,02H
      OUT 1BH,AL
      MOV CX,0FFFFH
      L2: LOOP L2
      MOV AL,00H
      OUT 1BH,AL

      MOV AL,08H
      OUT 1BH,AL
      MOV CX,0FFFFH
      L3: LOOP L3
      MOV AL,00H
      OUT 1BH,AL

      MOV AL,04H
      OUT 1BH,AL
      MOV CX,0FFFFH
      L4: LOOP L4
      MOV AL,00H
      OUT 1BH,AL

      MOV AL,01H
      OUT 1BH,AL
      MOV CX,0FFFFH
      L1R: LOOP L1R
      MOV AL,00H
      OUT 1BH,AL

    MOV AL,08H
      OUT 1BH,AL
      MOV CX,0FFFFH
      L2R: LOOP L2R
      MOV AL,00H
      OUT 1BH,AL
```

```
      MOV AL,04H
      OUT 1BH,AL
      MOV CX,0FFFFH
      L3R: LOOP L3R
      MOV AL,00H
      OUT 1BH,AL

      MOV AL,02H
      OUT 1BH,AL
      MOV CX,0FFFFH
      L4R: LOOP L4R
      MOV AL,00H
      OUT 1BH,AL


    MOV AL,11111111B
      OUT 1BH,AL
      MOV CX,0FFFFH
      L1A: LOOP L1A
      MOV AL,00H
      OUT 1BH,AL

   MOV AL,00H
      OUT 1BH,AL
      MOV CX,0FFFFH
      L1B: LOOP L1B
      MOV AL,00H
      OUT 1BH,AL

    MOV AL,11111111B
      OUT 1BH,AL
      MOV CX,0FFFFH
      L2A: LOOP L2A
      MOV AL,00H
      OUT 1BH,AL

    MOV AL,00H
      OUT 1BH,AL
      MOV CX,0FFFFH
      L1C: LOOP L1C
      MOV AL,00H
      OUT 1BH,AL

    MOV AL,11111111B
      OUT 1BH,AL
      MOV CX,0FFFFH
      L3A: LOOP L3A
      MOV AL,00H
      OUT 1BH,AL
    JMP L
A ENDS
END s
```

DOS Command:

The DOS command follows the same procedure as previous experiments.

Discussion:

This experiment was similar to the previous ones with the patterns of the LED display. In this specific experiment multiple patterns were attached together to create a long repeating pattern that mimics that of the standard wedding lights used in LED stips.