STAMFORD UNIVERSITY BANGLADESH

Department of Computer Science And Engineering
CT: Tarikuzzaman Emon

CSE 326: Microprocessor
Lab Manual

Experiment No. : 1
Name of the Experiment: Display of Seven Segment

Theory:

A seven-segment display (SSD), or seven-segment indicator, is a form of electronic display device for displaying decimal numerals.
The seven segments are arranged as a rectangle of two vertical segments on each side with one horizontal segment on the top, middle, and bottom. Additionally, the seventh segment bisects the rectangle horizontally. The segments of a 7-segment display are referred to by the letters A to G, where the optional decimal point (an "eighth segment", referred to as DP) is used for the display of non-integer numbers.

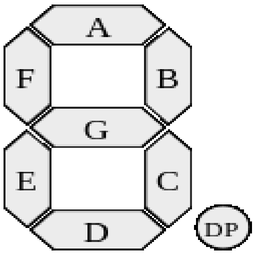

Fig. A Seven Segment Display

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

Note: Active when 0
Inactive when 1

Equipments:

A computer
CMD Command prompt
WINCOM Software
Dash Board

Code Segment:

```
A SEGMENT PARA PUBLIC ‘CODE’  [runs the code publicly]
ASSUME CS: A
ORG 1000H                     [program starts at location 1000h]

S:

MOV AL, 80H                   [ initializes all ports as output]
OUT 1FH, AL                   [ enables display of value in AL register]

MOV AL, 0F8H                  [ gets the hexadeciml value of 7]
OUT 19H, AL                   [displays the value in AL register on seven segment]

A ENDS                        [terminates the assembly]
END S                         [marks the end of segment]
```

DOS Command:

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

```
CD\
CD MDA
CD 8086
CD ASM8086
MASM
(NAME_OF_THE_FILE).ASM
(NAME_OF_THE_FILE).OBJ
(NAME_OF_THE_FILE).LST
LOD186
(NAME_OF_THE_FILE).0BJ
(NAME_OF_THE_FILE).ABS
```

After the commands ar given we select the WINCOM software from PC mode. We run the WINCOM s/w and press RESET from the tool kit. As soon as we press the RESET, we can see some machine written on the PC screen after which we give the following commands:

```
L (ENTER)
F3 (ENTER)
(NAME_OF_THE_FILE).ABS
G (ENTER)
```

After the commands are given, we can see our desired output on the tool kit.

Discussion:

The goal of the given experiment was to display decimal numbers using seven segment displays. We achieved the goal but encountered some difficulty.

Being new to the WINCOM s/w, it took us time to figure out how to connect it with PC. We kept the ASM file on some other folder which led to difficulty as the commands provided was unable to locate the file and showed error.

Later we kept the file in MDA folder followed by 8086 and ASM8086. Thus, we ran the WINCOM s/w after the providing the DOS Commands and got the desired output.

Experiment No. : 2
Name of the Experiment: To turn the Red light on LED

Theory:

This is the segment B(1BH). There are four LED lights available. The process to turn on the lights on is the reverse of seven segments. The code for each LED light are: Red (PB 0), Green (PB 1), Yellow (PB 3), Red (PB 3).

Table to display the values for 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

Note: Active when 1
Inactive when 0

Equipments:

A computer
CMD Command prompt
WINCOM Software
Dash Board

Code Segment:

```
A SEGMENT PARA PUBLIC ‘CODE’       [runs the code publicly]
ASSUME CS: A
ORG 1000H                          [program starts at location 1000h]

S:

MOV AL, 80H                        [ initializes all ports as output]
OUT 1FH, AL                        [ enables display of value in AL register]

MOV AL, 0FFH                       [ stops port A]
OUT 19H, AL

L:

MOV AL, 01H                        [ turn LED on]
OUT 1BH, AL

MOV CX, 0FFFFH                     [ delay for  LED on]
L1: LOOP L1

MOV AL, 00H                        [ turn LED off]
OUT 1BH, AL

MOV CX, 0FFFFH                     [delay the LED off]
L2: LOOP L2

JMP L
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 software.

```
CD\
CD MDA
CD 8086
CD ASM8086
MASM
(NAME_OF_THE_FILE).ASM
(NAME_OF_THE_FILE).OBJ
(NAME_OF_THE_FILE).LST
LOD186
(NAME_OF_THE_FILE).0BJ
(NAME_OF_THE_FILE).ABS
```

After the commands ar given we select the WINCOM software from PC mode. We run the WINCOM s/w and press RESET from the tool kit. As soon as we press the RESET, we can see some machine written on the PC screen after which we give the following commands:

```
L (ENTER)
F3 (ENTER)
(NAME_OF_THE_FILE).ABS
G (ENTER)
```

After the commands are given, we can see our desired output on the tool kit.

Discussion:

The goal of the given experiment was to turn on RED LED on using LED display. We achieved the goal encountering some difficulty. The PC was shut due to some unavoidable circumstances. We had to rewrite the code and then provided the DOS Commands. Also, we had difficulty turning the RED LED on and off using the JMP (jump) which is a loop.
Thus, we ran the WINCOM s/w after the providing the DOS Commands and got the desired output.

Experiment No. : 3

Name of the Experiment: To turn the LED light clock wise

Theory:

This is the segment B (1BH). There are four LED lights available. The process to turn on the lights on is the reverse of seven segments. The code for each LED light are: Red (PB 0), Green (PB 1), Yellow (PB 3), Red (PB 3).

Table to display the values for 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

Note: Active when 1
Inactive when 0

Equipments:

A computer
CMD Command prompt
WINCOM Software
Dash Board

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

```
CD\
CD MDA
CD 8086
CD ASM8086
MASM
(NAME_OF_THE_FILE).ASM
(NAME_OF_THE_FILE).OBJ
(NAME_OF_THE_FILE).LST
```

```
LOD186
(NAME_OF_THE_FILE).0BJ
(NAME_OF_THE_FILE).ABS
```

After the commands are given we select the WINCOM software from PC mode. We run the WINCOM s/w and press RESET from the tool kit. As soon as we press the RESET, we can see some machine written on the PC screen after which we give the following commands:

```
L (ENTER)
F3 (ENTER)
(NAME_OF_THE_FILE).ABS
G (ENTER)
```

After the commands are given, we can see our desired output on the tool kit.

Discussion:

The goal of the given experiment was to turn on LED on using LED display clock wise. We achieved the goal encountering some difficulty.

The PC was shut due to some unavoidable circumstances. We had to rewrite the code and then provided the DOS Commands. Also, we had difficulty turning the LED clock wise on and off using the JMP (jump) which is a loop.

Thus, we ran the WINCOM s/w after the providing the DOS Commands and got the desired output.

Experiment No. : 4

Name of the Experiment: To turn the LED light Anti clock wise

Theory:

This is the segment B (1BH). There are four LED lights available. The process to turn on the lights on is the reverse of seven segments. The code for each LED light are: Red (PB 0), Green (PB 1), Yellow (PB 3), Red (PB 3).

Table to display the values for 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

Note: Active when 1
Inactive when 0

Equipments:

A computer
CMD Command prompt
WINCOM Software
Dash Board

Code Segment:

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

S:

MOV AL, 80H                 [set control register]
OUT 1FH, AL

MOV AL, 0FFH                [stop port A]
OUT 19H, AL
L:
MOV AL, 04H                 [ turn on Yellow LED]
OUT 1BH, AL

MOV CX, 0FFFFH              [ delay]
L1: LOOP L1

MOV AL, 00H                 [ turn off Yellow LED]
OUT 1BH, AL

MOV AL, 01H                 [turn Red LED on]
OUT 1BH, AL

MOV CX, 0FFFFH              [delay]
L2: LOOP L2

MOV AL, 00H                 [turn off Red LED]
OUT 1BH, AL

MOV AL, 02H                 [turn on Green LED]
OUT 1BH, AL
```

```
MOV CX, 0FFFFH                [delay]
L3: LOOP L3

MOV AL, 00H                   [turn off Green LED]
OUT 1BH, AL

MOV AL, 08H                   [turn on Red LED]
OUT 1BH, AL

MOV CX, 0FFFFH                [delay]
L4: LOOP L4

MOV AL, 00H                   [turn off Red LED]
OUT 1BH, AL

JMP L
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 software.

```
CD\
CD MDA
CD 8086
CD ASM8086
MASM
(NAME_OF_THE_FILE).ASM
(NAME_OF_THE_FILE).OBJ
(NAME_OF_THE_FILE).LST
LOD186
(NAME_OF_THE_FILE).0BJ
(NAME_OF_THE_FILE).ABS
```

After the commands are given we select the WINCOM software from PC mode. We run the WINCOM s/w and press RESET from the tool kit. As soon as we press the RESET, we can see some machine written on the PC screen after which we give the following commands:

```
L (ENTER)
F3 (ENTER)
(NAME_OF_THE_FILE).ABS
G (ENTER)
```

After the commands are given, we can see our desired output on the tool kit.

Discussion:

The goal of the given experiment was to turn on LED on using LED display clock wise. We achieved the goal encountering some difficulty.

The PC was shut due to some unavoidable circumstances. We had to rewrite the code and then provided the DOS Commands. Also, we had difficulty turning the LED clock wise on and off using the JMP (jump) which is a loop.

Thus, we ran the WINCOM s/w after the providing the DOS Commands and got the desired output.

Experiment No. : 5

Name of the Experiment: The Traffic Signal

Theory:

In general we know how a traffic signal works. At first the red light turns on then the yellow light after which the green light turns on.
Here, we first turn the Red light on and then display 0-9 backwards on the seven segment display. Then the yellow light , green light and then again the red light.

Table to display the values for 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

Note: Active when 1
Inactive when 0

Equipments:

A computer
CMD Command prompt
WINCOM Software
Dash Board

Code Segment:

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

S:

MOV AL, 80H                 [ set control register]
OUT 1FH, AL

MOV AL, 0FFH                [ stop port A]
OUT 19H, AL
L:
MOV AL, 01H                 [ turn Red LED]
OUT 1BH, AL

MOV CX, 0FFFFH              [ delay for on]
L1: LOOP L1

MOV AL, 090H                [ display 9 on seven segment]
OUT 19H, AL

MOV CX, 0FFFFH              [delay]
L2: LOOP L2

MOV AL, 0FFH                [ turn off segment]
OUT 19H, AL

MOV AL, 080H                [ display 8 on seven segment]
OUT 19H, AL

MOV CX, 0FFFFH              [delay]
```

```
L3: LOOP L3


MOV AL, 0FFH                    [ turn off segment]
OUT 19H, AL

MOV AL, 0D8H                    [ display 7 on seven segment]
OUT 19H, AL

MOV CX, 0FFFFH                  [delay]
L4: LOOP L4

MOV AL, 0FFH                    [ turn off segment]
OUT 19H, AL

MOV AL, 082H                    [ display 6 on seven segment]
OUT 19H, AL

MOV CX, 0FFFFH                  [delay]
L5: LOOP L5

MOV AL, 0FFH                    [ turn off segment]
OUT 19H, AL

MOV AL, 092H                    [ display 5 on seven segment]
OUT 19H, AL

MOV CX, 0FFFFH                  [delay]
L6: LOOP L6

MOV AL, 0FFH                    [ turn off segment]
OUT 19H, AL

MOV AL, 099H                    [ display 4 on seven segment]
OUT 19H, AL
```

```
MOV CX, 0FFFFH                [delay]
L7: LOOP L7

MOV AL, 0FFH                  [ turn off segment]
OUT 19H, AL

MOV AL, 0B0H                  [ display 3 on seven segment]
OUT 19H, AL

MOV CX, 0FFFFH                [delay]
L8: LOOP L8

MOV AL, 0FFH                  [ turn off segment]
OUT 19H, AL

MOV AL, 0A4H                  [ display 2 on seven segment]
OUT 19H, AL

MOV CX, 0FFFFH                [delay]
L9: LOOP L9

MOV AL, 0FFH                  [ turn off segment]
OUT 19H, AL

MOV AL, 0F9H                  [ display 1 on seven segment]
OUT 19H, AL

MOV CX, 0FFFFH                [delay]
L10: LOOP L10

MOV AL, 0FFH                  [ turn off segment]
OUT 19H, AL

MOV AL, 0C0H                  [ display 0 on seven segment]
```

```
OUT 19H, AL

MOV CX, 0FFFFH                  [delay]
L11: LOOP L11

MOV AL, 0FFH                    [ turn off segment]
OUT 19H, AL

MOV AL, 00H                     [ turn of Red LED]
OUT 1BH, AL

MOV CX, 0FFFFH                  [ delay]
L12: LOOP L12

MOV AL, 04H                     [ turn on Yellow LED]
OUT 1BH, AL

MOV CX, 0FFFFH                  [ delay]
L13: LOOP L13

MOV AL, 00H                     [ turn off Yellow LED]
OUT 1BH, AL

MOV CX, 0FFFFH                  [ delay]
L14: LOOP L14

MOV AL, 02H                     [ turn on Green LED]
OUT 1BH, AL

MOV CX, 0FFFFH                  [ delay]
L15: LOOP L15

MOV AL, 090H                    [ display 9 on seven segment]
OUT 19H, AL
```

```
MOV CX, 0FFFFH                [delay]
L16: LOOP 16

MOV AL, 0FFH                  [ turn off segment]
OUT 19H, AL

MOV AL, 080H                  [ display 8 on seven segment]
OUT 19H, AL

MOV CX, 0FFFFH                [delay]
L17: LOOP L17

MOV AL, 0FFH                  [ turn off segment]
OUT 19H, AL

MOV AL, 0D8H                  [ display 7 on seven segment]
OUT 19H, AL

MOV CX, 0FFFFH                [delay]
L18: LOOP L18

MOV AL, 0FFH                  [ turn off segment]
OUT 19H, AL

MOV AL, 082H                  [ display 6 on seven segment]
OUT 19H, AL

MOV CX, 0FFFFH                [delay]
L19: LOOP L19

MOV AL, 0FFH                  [ turn off segment]
OUT 19H, AL

MOV AL, 092H                  [ display 5 on seven segment]
OUT 19H, AL
```

```
MOV CX, 0FFFFH                [delay]
L20: LOOP L20

MOV AL, 0FFH                  [ turn off segment]
OUT 19H, AL

MOV AL, 099H                  [ display 4 on seven segment]
OUT 19H, AL

MOV CX, 0FFFFH                [delay]
L21: LOOP L21

MOV AL, 0FFH                  [ turn off segment]
OUT 19H, AL

MOV AL, 0B0H                  [ display 3 on seven segment]
OUT 19H, AL

MOV CX, 0FFFFH                [delay]
L22: LOOP L22

MOV AL, 0FFH                  [ turn off segment]
OUT 19H, AL

MOV AL, 0A4H                  [ display 2 on seven segment]
OUT 19H, AL

MOV CX, 0FFFFH                [delay]
L23: LOOP L23

MOV AL, 0FFH                  [ turn off segment]
OUT 19H, AL

MOV AL, 0F9H                  [ display 1 on seven segment]
```

```
OUT 19H, AL

MOV CX, 0FFFFH              [delay]
L24: LOOP L24



MOV AL, 0FFH                [ turn off segment]
OUT 19H, AL

MOV AL, 0C0H                [ display 0 on seven segment]
OUT 19H, AL

MOV CX, 0FFFFH              [delay]
L25: LOOP L25

MOV AL, 0FFH                [ turn off segment]
OUT 19H, AL

MOV AL, 090H                [ display 9 on seven segment]
OUT 19H, AL

MOV CX, 0FFFFH              [delay]
L26: LOOP L26

MOV AL, 0FFH                [ turn off segment]
OUT 19H, AL

MOV AL, 080H                [ display 8 on seven segment]
OUT 19H, AL

MOV CX, 0FFFFH              [delay]
L27: LOOP L27
```

```
MOV AL, 0FFH                    [ turn off segment]
OUT 19H, AL

MOV AL, 0D8H                    [ display 7 on seven segment]
OUT 19H, AL



MOV CX, 0FFFFH                  [delay]
L28: LOOP L28

MOV AL, 0FFH                    [ turn off segment]
OUT 19H, AL

MOV AL, 082H                    [ display 6 on seven segment]
OUT 19H, AL

MOV CX, 0FFFFH                  [delay]
L29: LOOP L29

MOV AL, 0FFH                    [ turn off segment]
OUT 19H, AL

MOV AL, 092H                    [ display 5 on seven segment]
OUT 19H, AL

MOV CX, 0FFFFH                  [delay]
L30: LOOP L30

MOV AL, 0FFH                    [ turn off segment]
OUT 19H, AL

MOV AL, 099H                    [ display 4 on seven segment]
OUT 19H, AL
```

```
MOV CX, 0FFFFH                [delay]
L31: LOOP L31

MOV AL, 0FFH                  [ turn off segment]
OUT 19H, AL

MOV AL, 0B0H                  [ display 3 on seven segment]
OUT 19H, AL

MOV CX, 0FFFFH                [delay]
L32: LOOP L32

MOV AL, 0FFH                  [ turn off segment]
OUT 19H, AL

MOV AL, 0A4H                  [ display 2 on seven segment]
OUT 19H, AL

MOV CX, 0FFFFH                [delay]
L33: LOOP L33

MOV AL, 0FFH                  [ turn off segment]
OUT 19H, AL

MOV AL, 0F9H                  [ display 1 on seven segment]
OUT 19H, AL

MOV CX, 0FFFFH                [delay]
L34: LOOP L34

MOV AL, 0FFH                  [ turn off segment]
OUT 19H, AL

MOV AL, 0C0H                  [ display 0 on seven segment]
```

```
OUT 19H, AL

MOV CX, 0FFFFH                  [delay]
L35: LOOP L35

MOV AL, 0FFH                    [ turn off segment]
OUT 19H, AL

MOV AL, 090H                    [ display 9 on seven segment]
OUT 19H, AL

MOV CX, 0FFFFH                  [delay]
L36: LOOP L36

MOV AL, 0FFH                    [ turn off segment]
OUT 19H, AL

MOV AL, 080H                    [ display 8 on seven segment]
OUT 19H, AL

MOV CX, 0FFFFH                  [delay]
L37: LOOP L37

MOV AL, 0FFH                    [ turn off segment]
OUT 19H, AL

MOV AL, 0D8H                    [ display 7 on seven segment]
OUT 19H, AL

MOV CX, 0FFFFH                  [delay]
L38: LOOP L38

MOV AL, 0FFH                    [ turn off segment]
OUT 19H, AL
```

```
MOV AL, 082H                    [ display 6 on seven segment]
OUT 19H, AL

MOV CX, 0FFFFH                  [delay]
L39: LOOP L39

MOV AL, 0FFH                    [ turn off segment]
OUT 19H, AL

MOV AL, 092H                    [ display 5 on seven segment]
OUT 19H, AL

MOV CX, 0FFFFH                  [delay]
L40: LOOP L40

MOV AL, 0FFH                    [ turn off segment]
OUT 19H, AL

MOV AL, 099H                    [ display 4 on seven segment]
OUT 19H, AL

MOV CX, 0FFFFH                  [delay]
L41: LOOP L41

MOV AL, 0FFH                    [ turn off segment]
OUT 19H, AL

MOV AL, 0B0H                    [ display 3 on seven segment]
OUT 19H, AL

MOV CX, 0FFFFH                  [delay]
L42: LOOP L42

MOV AL, 0FFH                    [ turn off segment]
OUT 19H, AL
```

```
MOV AL, 0A4H                [ display 2 on seven segment]
OUT 19H, AL

MOV CX, 0FFFFH              [delay]
L43: LOOP L43

MOV AL, 0FFH                [ turn off segment]
OUT 19H, AL

MOV AL, 0F9H                [ display 1 on seven segment]
OUT 19H, AL

MOV CX, 0FFFFH              [delay]
L44: LOOP L44

MOV AL, 0FFH                [ turn off segment]
OUT 19H, AL

MOV AL, 0C0H                [ display 0 on seven segment]
OUT 19H, AL

MOV CX, 0FFFFH              [delay]
L45: LOOP L45

MOV AL, 0FFH                [ turn off segment]
OUT 19H, AL

MOV AL, 00H                 [ turn off Green LED]
OUT 1BH, AL

MOV AL, 01H                 [ turn Red LED on]
OUT 1BH, AL

MOV CX, 0FFFFH              [ delay to turn on]
```

```
L46: LOOP L46

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 software.

```
CD\
CD MDA
CD 8086
CD ASM8086
MASM
(NAME_OF_THE_FILE).ASM
(NAME_OF_THE_FILE).OBJ
(NAME_OF_THE_FILE).LST
LOD186
(NAME_OF_THE_FILE).OBJ
(NAME_OF_THE_FILE).ABS
```

After the commands are given we select the WINCOM software from PC mode. We run the WINCOM s/w and press RESET from the tool kit. As soon as we press the RESET, we can see some machine written on the PC screen after which we give the following commands:

```
L (ENTER)
F3 (ENTER)
(NAME_OF_THE_FILE).ABS
G (ENTER)
```

After the commands are given, we can see our desired output on the tool kit.

Discussion:

The goal of the given experiment was to display the Traffic Signal. We achieved the goal encountering some difficulty.
The PC was shut due to some unavoidable circumstances. We had to rewrite the code and then provided the DOS Commands.
Moreover, the code for the experiment was too lengthy we experiment which is why there were several mistakes and the outcome of the experiment was achieved lately. After several checking of the code we achieved the goal.
Thus, we ran the WINCOM s/w after the providing the DOS Commands and got the desired output.

Experiment No. : 6
Name of the Experiment: Addition

Theory:

We add to numbers eg, 9 and 5 which is displayed on the seven segment and their result is also displayed in the seven segment. The result of the addition shall only be shown after the LED lights turn on in the order as PB0, PB1, PB2 and PB3 sequentially. After the LED lights are done then the seven segment shall display 1 and 4 one after the other which is the required result.

Equipments:

A computer
CMD Command prompt
WINCOM Software
Dash Board

Code Segment:

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

S:

MOV AL, 80H
OUT 1FH, AL

MOV AL, 090H
OUT 19H, AL

MOV CX, 0FFFFH                    [delay]
L: LOOP L
```

```
MOV AL, 092H                [ display 5 on seven segment]
OUT 19H, AL

MOV CX, 0FFFFH              [delay]
L1: LOOP L1

MOV AL, 0FFH                [ turn off segment]
OUT 19H, AL

MOV AL, 01H                 [turn on PB0 LED]
OUT 1BH, AL

MOV CX, 0FFFFH              [delay]
L2: LOOP L2

MOV AL, 00H                 [ turn off PB0 LED]
OUT 1BH, AL

MOV CX, 0FFFFH              [delay]
L3: LOOP L3

MOV AL, 02H                 [ turn on PB1 LED]
OUT 19H, AL

MOV CX, 0FFFFH              [delay]
L4: LOOP L4

MOV AL, 00H                 [ turn off PB1 LED]
OUT 1BH, AL

MOV CX, 0FFFFH              [delay]
L5: LOOP L5

MOV AL, 04H                 [ turn on PB2 LED]
```

```
OUT 19H, AL

MOV CX, 0FFFFH              [delay]
L6: LOOP L6

MOV AL, 00H                 [ turn off PB2 LED]
OUT 1BH, AL

MOV CX, 0FFFFH              [delay]
L7: LOOP L7

MOV AL, 08H                 [ turn on PB3 LED]
OUT 19H, AL

MOV CX, 0FFFFH              [delay]
L8: LOOP L8

MOV AL, 00H                 [ turn off PB3 LED]
OUT 1BH, AL

MOV CX, 0FFFFH              [delay]
L9: LOOP L9

MOV AL, 0F9H                [ display 1 on seven segment]
OUT 19H, AL

MOV CX, 0FFFFH              [delay]
L10: LOOP L10

MOV AL, 099H                [ display 4 on seven segment]
OUT 19H, AL

MOV CX, 0FFFFH              [delay]
L11: LOOP L11
```

```
MOV AL, 0FFH                    [ stop segment]
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 software.

```
CD\
CD MDA
CD 8086
CD ASM8086
MASM
(NAME_OF_THE_FILE).ASM
(NAME_OF_THE_FILE).OBJ
(NAME_OF_THE_FILE).LST
LOD186
(NAME_OF_THE_FILE).OBJ
(NAME_OF_THE_FILE).ABS
```

After the commands are given we select the WINCOM software from PC mode. We run the WINCOM s/w and press RESET from the tool kit. As soon as we press the RESET, we can see some machine written on the PC screen after which we give the following commands:

```
L (ENTER)
F3 (ENTER)
(NAME_OF_THE_FILE).ABS
G (ENTER)
```

After the commands are given, we can see our desired output on the tool kit.

Discussion:

The goal of the given experiment was to display the addition result using the seven segment. We achieved the goal encountering some difficulty.
The PC was shut due to some unavoidable circumstances. We had to rewrite the code and then provided the DOS Commands.
Moreover, the code for the experiment was too lengthy we experiment which is why there were several mistakes and the outcome of the experiment was achieved lately. After several checking of the code we achieved the goal.
Thus, we ran the WINCOM s/w after the providing the DOS Commands and got the desired output.