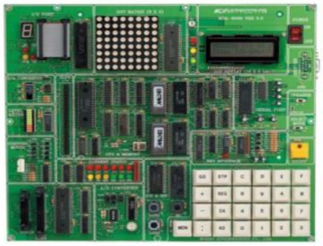
Session 01

Seven Segment Display and LED Display

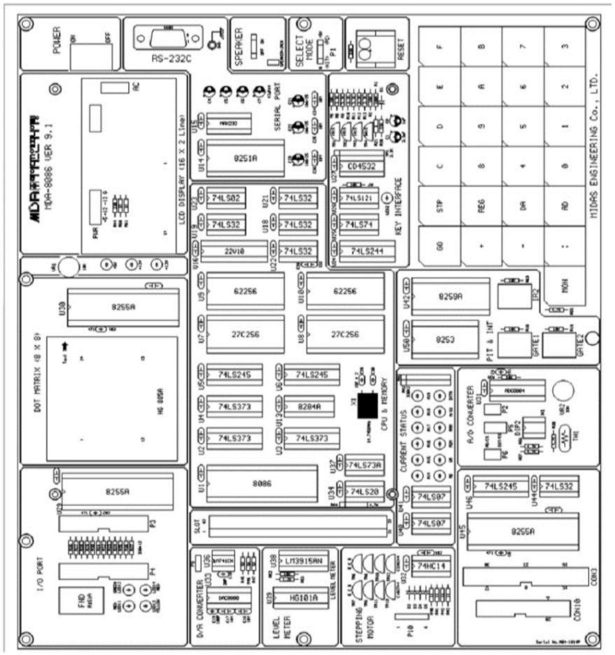
OBJECTIVES:

- Students will become familiar with MDA-8086 kit.
- They will have a brief idea about the types and functions of various keys.
- They can perform some basic operations using 8086 basic instructions.
- They will be able to interface 8255 peripheral devices with MDA-8086 and show output in Seven Segment Display and LED Display.

MDA-8086 Kit Diagram



MDA-8086 System Configuration



The function of IC's at MDA-8086 System Configuration

1. CPU (Central processing unit): Using Intel 8086, using 14.7456MHz.
2. ROM (Read Only Memory): It has program to control user's key input, LCD display, user's program. 64K Byte, it has data communication program. Range of ROM Address is F0000H~FFFFFH.
3. SRAM (Static Random-Access Memory): Input user's program & data. Address of memory is 00000H~0FFFFH, totally 64K Byte.
4. DISPLAY: Text LCD Module, 16(Characters)×2(Lines)
5. KEYBOARD: It is used to input machine language. There are 16 hexadecimal keys and 8 function keys.
6. SPEAKER: Sound test.
7. RS-232C: Serial communication with IBM compatible PC.
8. DOT MATRIX LED: To understand & test the dot matrix structure and principle of display. It is interfaced to 8255A(PPI).
9. A/D CONVERTER: ADC0804 to convert the analog signal to digital signal.
10. D/A CONVERTER: DAC0800 (8-bits D/A converter) to convert the digital signal to the analog signal and to control the level meter.
11. STEPPING MOTOR INTERFACE: Stepping motor driver circuit is designed.
12. POWER: AC 110~220V, DC +5V 3A, +12V 1A, -12V 0.5A SMPS.

MDA-8086 Address Map

1. Memory Map

ADDRESS	MEMORY	DESCRIPTION
00000H~0FFFFH	RAM	PROGRAM & DATA MEMORY
F0000H~FFFFFH	ROM	MONITOR ROM
10000H~EFFFFH	USER'S RANGE	

2. I/O Address Map

ADDRESS	I/O PORT	DESCRIPTION
00H~07H	LCM & KEYBOARD	LCD Display 00H: INSTRUCTION REGISTER 02H: STATUS REGISTER 04H: DATA REGISTER KEYBOARD 01H: KEYBOARD REGISTER (Only read) 01H: KEYBOARD FLAG (Only write)
08H~0FH	8251/8253	8251(Using to data communication) 08H: DATA REGISTER 0AH: INSTRUCTION/STATUS REGISTER 8253 (TIMER/COUNTER) 09H: TIMER 0 REGISTER 0BH: TIMER 1 REGISTER 0DH: TIMER 2 REGISTER 0FH: CONTROL REGISTER
10H~17H	8259/SPEAKER	8259(Interrupt controller) 10H: COMMAND REGISTER 12H: DATA REGISTER SPEAKER 11H: SPEAKER
18H~1FH	8255A-CS1/ 8255A-CS2	8255A-CS1(DOT & ADC INTERFACE) 18H: A PORT DATA REGISTER 1AH: B PORT DATA REGISTER 1CH: C PORT CONTROL REGISTER 8255-CS2(LED & STEPPING MOTOR) 19H: A PORT DATA REGISTER 1BH: B PORT DATA REGISTER 1DH: C PORT CONTROL REGISTER 1FH: CONTROL REGISTER
20H~2FH	I/O EXTEND CONNECTOR	
30H~FFH	USER'S RANGE	

Operation Introduction

MDA-8086 has high performance 64K-byte monitor program. It is designed for easy function. After power is on, the monitor program begins to work. In addition to all the key function the monitor has a memory checking routine.

FUNCTION KEY		DATA KEY			
				MON	RES
GO	STP	C	D	E	F
+	REG	8	9	A	B
-	DA	4	5	6	7
:	AD	0	1	2	3

- RES→ System reset
- STP→ Execute user's program, a single step
- AD→ Set memory address
- GO→ Go to user's program or execute monitor functions
- DA→ Update segment & Offset and input data to memory
- MON→ Immediately break user's program and Non maskable interrupt.
- : → Offset set
- REG→ Register Display.
- +→ Segment & Offset +1 increment. Register display increment.
- -→ Segment & Offset -1 increment. Register display decrement.

8255 Programmable Peripheral Interface Controller

- It has 24-bit input/output pins
- It consists of three ports: port A, port B and port C- all of which are 8 bits
- It also consists of an 8-bit control register(CR)
- The eight bit of port C can be used as individual bits or be grouped in two 4-bit ports: C_{upper}(CU) and C_{lower}(CL)
- The functions of these ports are defined by writing a control word in the control register

Group A	Group B
Port A	Port B
Port C (Upper 4 bit)	Port C (Lower 4 bit)

8086 Instruction Set Summary

Data Registers	AX (Accumulator Register)	AH	AL
	BX (Base Register)	BH	BL
	CX (Count Register)	CH	CL
	DX (Data Register)	DH	DL
Segment Registers	CS (Code Segment)		
	DS (Data Segment)		
	SS (Stack Segment)		
	ES (Extra Segment)		
Index Registers	SI (Source Index)		
	DI (Destination Index)		
Pointer Registers	SP (Stack Pointer)		
	BP (Base Pointer)		
	IP (Instruction Pointer)		
	FLAGS Registers		

15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
				OF	DF	IF	TF	SF	ZF		AF		PF		CF

Bit	**Name**	**Symbol**	**Status Flags**
0	Carry Flag	CF	
2	Parity Flag	PF	
4	Auxiliary Carry Flag	AF	
6	Zero Flag	ZF	
7	Sign Flag	SF	
11	Overflow Flag	OF	

8	Trap Flag	TF	**Control Flags**
9	Interrupt Flag	IF	
10	Direction Flag	DF	

Data Transfer Instructions

Name	Mnemonic
Load	LD
Store	ST
Move	MOV
Exchange	XCHG
Input	IN
Output	OUT
Push	PUSH
Pop	POP

Arithmetic

Name	Mnemonic
Increment	INC
Decrement	DEC
Add	ADD
Subtract	SUB
Multiply	MUL
Divide	DIV
Add with carry	ADDC
Subtract with borrow	SUBB
Negate	NEG

Logical and Bit Manipulation

Name	Mnemonic
Clear	CLR
Complement	COM
AND	AND
OR	OR
Exclusive-OR	XOR
Clear carry	CLRC
Set carry	SETC
Complement carry	COMC
Disable interrupt	DI

Shift and Rotate

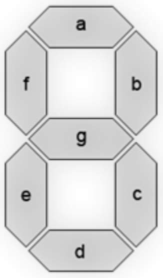
Name	Mnemonic
Logical shift right	SHR
Logical shift left	SHL
Arithmetic shift right	SHRA
Arithmetic shift left	SHLA
Rotate right	ROR
Rotate left	ROL
Rotate right through carry	RORC
Rotate left through carry	ROLC

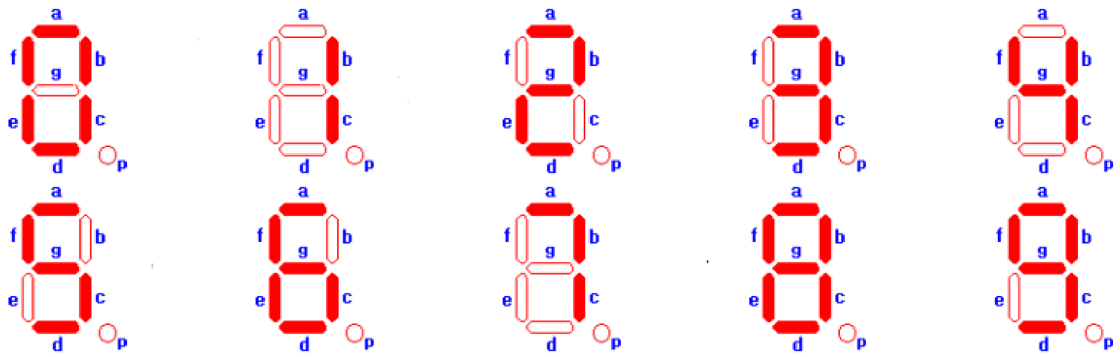
Program Control Instructions

Name	Mnemonic
Branch	BR
Jump	JMP
Skip	SKP
Call	CALL
Return	RET
Compare (Subtract)	CMP
Test (AND)	TST

Experiment No: 01

Experiment Name: Write an assembly code to display 0-9 in Seven Segment Display (SSD).





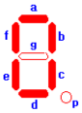
- For seven segments display we use 0 for ON and 1 for OFF.
- Control register values will be the column headings of the following table:

D7	D6	D5	D4	D3	D2	D1	D0
1	0	0	0	0	0	0	0
Control Register 0- BSR mode 1- I/O mode	Mode selection for group A 00- I/O 01- Handshaking		Port A 0- Output 1- Input	Upper 4 bit of port C	Mode selection for group B 0- I/O 1- Handshaking	For port B	For lower 4 bit of port C

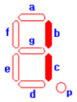
Assembly Code:

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

START:
  ;control register turn on
   MOV AL,80H
   OUT 1FH,AL
SSD:
    ;display 0
    MOV AL,0C0H
    OUT 19H,AL
    ;for delay
    MOV CX,0FFFFH
    L0:LOOP L0
    ;display 1
    MOV AL,0F9H
    OUT 19H,AL
    ;for delay
    MOV CX,0FFFFH
    L1:LOOP L1
    ;display 2
    MOV AL,0A4H
    OUT 19H,AL
    ;for delay
    MOV CX,0FFFFH
    L2:LOOP L2
    ;display 3
    MOV AL,0B0H
    OUT 19H,AL
    ;for delay
    MOV CX,0FFFFH
    L3:LOOP L3
    ;display 4
    MOV AL,099H
    OUT 19H,AL
    ;for delay
    MOV CX,0FFFFH
    L4:LOOP L4
    ;display 5
    MOV AL,092H
    OUT 19H,AL
    ;for delay
    MOV CX,0FFFFH
```

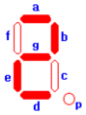
	g	f	e	d	c	b	a
1	1	0	0	0	0	0	0



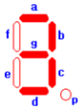
	g	f	e	d	c	b	a
1	1	1	1	1	0	0	1



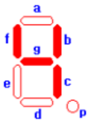
	g	f	e	d	c	b	a
1	0	1	0	0	1	0	0



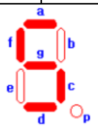
	g	f	e	d	c	b	a
1	0	1	1	0	0	0	0



	g	f	e	d	c	b	a
1	0	0	1	1	0	0	1

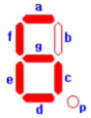


	g	f	e	d	c	b	a
1	0	0	1	0	0	1	0

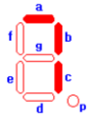


```
    L5:LOOP L5
    ;display 6
    MOV AL,082H
    OUT 19H,AL
    ;for delay
    MOV CX,0FFFFH
    L6:LOOP L6
    ;display 7
    MOV AL,0F8H
    OUT 19H,AL
    ;for delay
    MOV CX,0FFFFH
    L7:LOOP L7
    ;display 8
    MOV AL,080H
    OUT 19H,AL
    ;for delay
    MOV CX,0FFFFH
    L8:LOOP L8
    ;display 9
    MOV AL,090H
    OUT 19H,AL
    ;for delay
    MOV CX,0FFFFH
    L9:LOOP L9
    JMP SSD
S ENDS
END START
```

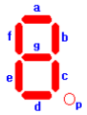
	g	f	e	d	c	b	a
1	0	0	0	0	0	1	0



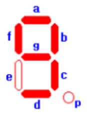
	g	f	e	d	c	b	a
1	1	1	1	1	0	0	0



	g	f	e	d	c	b	a
1	0	0	0	0	0	0	0



	g	f	e	d	c	b	a
1	0	0	1	0	0	0	0



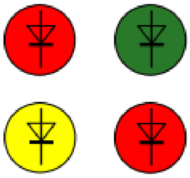
Steps to Run code in MDA-8086 through PC:

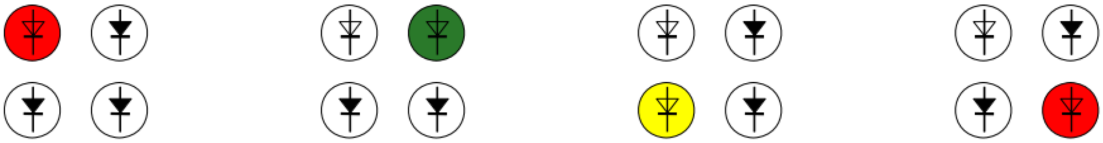
- At first copy paste the .ASM file in the mda folder of computer
- Then open cmd and write cd\ and press enter
- Then type cd mda and press enter
- Then type MASM and press enter
- Then write the file_name.ASM and press enter. For our example we will write S.ASM
- Then write the file_name.OBJ and press enter. For our example we will write S.OBJ
- Then write the file_name.LST and press enter. This step is used for error checking. For our example we will write S.LST
- Then when it wants .CRF file simply press enter
- If there is any error in the file, then after this line we can see the number of errors.
- If any error is found, then type EDIT file_name.LST and press enter.
- If no error is found, then type LOD186 and press enter
- Then type file_name.OBJ and press enter. For our example we will write S.OBJ

- Then type file_name.ABS and press enter. For our example we will write S.ABS
- Then type COMM and press enter.
- Then a blue window will occur
- We will now turn on the kit and we will select PC mode from kit mode
- Then press RESET
- If your kit is ok, then it will show up in the blue screen
- Then type L from keyboard and press enter
- If L does not show up, then it means your PC is not connected and you have to try in different PC
- Otherwise press F3 and in the pop-up screen write filename.ABS and press enter. For our example we will write S.ABS
- Then in the kit select kit mode from PC mode
- Then press RESET
- After that press AD
- Then Press GO
- Then you can see the output in the seven segments display

Experiment No: 02

Experiment Name: Write an assembly code to glow R1, G, Y and R2 in LED Display respectively.





- For LED display we use 1 for ON and 0 for OFF
- Control register value will be the column headings of the following table:

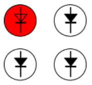
D7	D6	D5	D4	D3	D2	D1	D0
1	0	0	0	0	0	0	0
Control Register 0- BSR mode 1- I/O mode	Mode selection for group A 00- I/O 01- Handshaking		Port A 0- Output 1- Input	Upper 4 bit of port C	Mode selection for group B 0- I/O 1- Handshaking	For port B	For lower 4 bit of port C

Assembly Code:

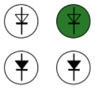
```
L SEGMENT PARA PUBLIC 'CODE'
ASSUME CS: L
ORG 1000H

START:
  ;control register turn on
   MOV AL,80H
   OUT 1FH,AL
   ;segment address forcefully off
   MOV AL,0FFH
   OUT 19H,AL
```

```
LED:
   ;R1 LED turn on
   MOV AL,01H
   OUT 1BH,AL
   ;for delay
   MOV CX,0FFFFH
   LR1:LOOP LR1
   ;G LED turn on
   MOV AL,02H
   OUT 1BH,AL
   ;for delay
   MOV CX,0FFFFH
   LG:LOOP LG
   ;Y LED turn on
   MOV AL,04H
   OUT 1BH,AL
   ;for delay
   MOV CX,0FFFFH
   LY:LOOP LY
   ;R2 LED turn on
   MOV AL,08H
   OUT 1BH,AL
   ;for delay
   MOV CX,0FFFFH
   LR2:LOOP LR2
   JMP LED
L ENDS
END START
```

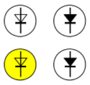
				R2	Y	G	R1
0	0	0	0	0	0	0	1



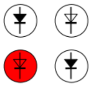
				R2	Y	G	R1
0	0	0	0	0	0	1	0



				R2	Y	G	R1
0	0	0	0	0	1	0	0



				R2	Y	G	R1
0	0	0	0	1	0	0	0



Steps to run code in MDA-8086 through PC:

- At first copy paste the .ASM file in the mda folder of computer
- Then open cmd and write cd\ and press enter
- Then type cd mda and press enter
- Then type MASM and press enter
- Then write the file_name.ASM and press enter. For our example we will write L.ASM
- Then write the file_name.OBJ and press enter. For our example we will write L.OBJ
- Then write the file_name.LST and press enter. This step is used for error checking. For our example we will write L.LST
- Then when it wants .CRF file simply press enter
- If there is any error in the file, then after this line we can see the number of errors.
- If any error is found, then type EDIT file_name.LST and press enter.
- If no error is found, then type LOD186 and press enter

- Then type file_name.OBJ and press enter. For our example we will write L.OBJ
- Then type file_name.ABS and press enter. For our example we will write L.ABS
- Then type COMM and press enter.
- Then a blue window will occur
- We will now turn on the kit and we will select PC mode from kit mode
- Then press RESET
- If your kit is ok, then it will show up in the blue screen
- Then type L from keyboard and press enter
- If L does not show up, then it means your PC is not connected and you have to try in different PC
- Otherwise press F3 and in the pop-up screen write filename.ABS and press enter. For our example we will write L.ABS
- Then in the kit select kit mode from PC mode
- Then press RESET
- After that press AD
- Then Press GO
- Then you can see the output in the LED display