Lab 8 – Assembly Programming

Gamage GGTA – 210175C

1. Lab task :-
   Learning assembly programming and interfacing simple input and output devices.
   - Designing a traffic light for two lanes
   - Printing a number using 2 7-segment displays
   - Getting the multiplication of first 5 integers
2. Assembly Codes

   Traffic lights

```
;------ The Main Program ----------------------------------------

Start:
        OUT     01      ;
        MOV     BL,A    ; To get a delay of 10 cycles
        MOV   AL,84   ; RED AND GREEN(10000100)
        OUT   01    ; Send AL to port One
        CALL    36      ; Call the procedure at address [36]

        MOV     BL,1    ; To get a delay of 1 cycle
        MOV     AL,48   ; Yellow and Yellow(01001000)
        OUT     01      ; Send AL to port One
        CALL    36      ; Call the procedure at address [36]

        MOV     BL,5    ; To get a delay of 5 cycle.
        MOV     AL,30   ; Green and Red(00110000)
        OUT     01      ; Send AL to port One
        CALL    36      ; Call the procedure at address [36]

JMP     Start   ; Jump back to the start.

; ----- Time Delay Procedure Stored At Address [36] ------------
        ORG     36      ; Generate machine code from address [36]

        PUSH    BL      ; Save AL on the stack.
        PUSHF           ; Save the CPU flags on the stack.
Rep:
        DEC     BL      ; Subtract one from AL.
        JNZ     REP     ; Jump back to Rep if AL was not Zero.

        POPF            ; Restore the CPU flags from the stack.
        POP     BL      ; Restore AL from the stack.
```

```
        RET                     ; Return from the procedure.
; ---------------------------------------------------------------
        END
; ---------------------------------------------------------------
```

7 Segment Display

```
        Start:
        MOV     AL,8A           ; (210175) 10001010
        OUT     02              ; Send the data in AL to Port 02

        MOV     AL,DD           ; 1101 1101
        OUT     02              ; Send the data in AL to Port 02

        END
```

Multiply all integers from 1 to 5

```
        ; ===== Counting ====================================


        MOV     BL,1            ; Initial value stored in BL
        MOV     CL,5            ; Decrement from 5

Rep:                            ; Jump back to this label
        MUL     BL,CL           ; Multiply in each loop
        DEC     CL              ; Dec ONE from CL
        JNZ     Rep             ; Jump back to Rep

        MOV     AL,8A   ; Store 10001010 in AL
        OUT     02              ; Send AL to port 2
        MOV     AL,FF           ; Store 11111111 in AL
        OUT     02              ; Send AL to port 2
        END                     ; Program Ends

; ===== Program Ends ================================
```

D:\Apps\Lab 8\210175C_08.ASM

File Edit View Examples Help

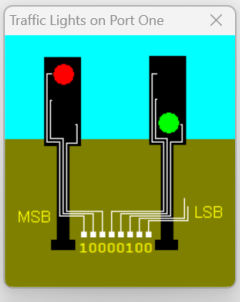
```
AL 10000100 84 -124   IP 00001010 0A +010
BL 00010000 10 +016   SP 10111111 BF -065
CL 00000000 00 +000   SR 00000000 00 +000
DL 00000000 00 +000         ISOZ
```

Assemble | Slower | Continue | Step | Faster | Cpu Reset | Run F9 | STOP | Show Ram

☐ Write Run Log ☐ Log Assembler Activity

Source Code | List File | Configuration | Tokens | Run Log

```
;------ The Main Program -----------------------------------------
Start:
        OUT     01      ; Turn off traffic lights
        MOV     BL,10   ; A short delay.
        MOV     AL,84   ; RED AND GREEN(10000100)
        OUT     01      ; Send AL to port One
        CALL    36      ; Call the procedure at address [36]

        MOV     BL,1    ; A middle sized delay.
        MOV     AL,48   ; Yellow and Yellow(01001000)
        OUT     01      ; Send AL to port One
        CALL    36      ; Call the procedure at address [36]

        MOV     BL,5    ; A Longer delay.
        MOV     AL,30   ; Green and Red(00110000)
        OUT     01      ; Send AL to port One
        CALL    36      ; Call the procedure at address [36]

JMP     Start   ; Jump back to the start.

; ----- Time Delay Procedure Stored At Address [36] -------------
        ORG     36      ; Generate machine code from address [36]

        PUSH    BL      ; Save AL on the stack.
        PUSHF           ; Save the CPU flags on the stack.
Rep:
        DEC     BL      ; Subtract one from AL.
        JNZ     REP     ; Jump back to Rep if AL was not Zero.

        POPF            ; Restore the CPU flags from the stack.
        POP     BL      ; Restore AL from the stack.

        RET             ; Return from the procedure.
; ---------------------------------------------------------------
        END
; ---------------------------------------------------------------
```



RAM Source Code View

```
    0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F
00 OUT 01 MOV BL 10 MOV AL 84 OUT 01 CALL36 MOV BL 1  MOV
10 AL  48 OUT 01 CALL36 MOV BL 5  MOV AL 30 OUT 01 CALL36
20 JMP STAREND END END END END END END END END END END END
30 END END END END END END PUSHBL PUSHDEC BL JNZ REP POPFPOP BL
40 RET END END END END END END END END END END END END END
50 END END END END END END END END END END END END END END
60 END END END END END END END END END END END END END END
70 END END END END END END END END END END END END END END
80 END END END END END END END END END END END END END END
90 END END END END END END END END END END END END END END
A0 END END END END END END END END END END END END END END
B0 END END END END END END END END END END END END END END
C0
D0
E0
F0
```

☐ X Hexadecimal ☐ Y ASCII ◉ Z Source

D:\Apps\Lab 8\210175C_08.ASM

File Edit View Examples Help

```
AL 01001000 48 +072   IP 00110110 36 +054
BL 00000001 01 +001   SP 10111110 BE -066
CL 00000000 00 +000   SR 00000000 00 +000
DL 00000000 00 +000         ISOZ
```

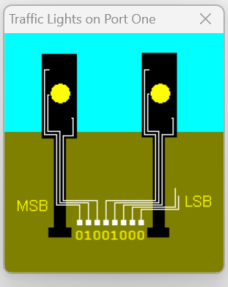
Assemble | Slower | Continue | Step | Faster | Cpu Reset | Run F9 | STOP | Show Ram

PUSH BL

☐ Write Run Log ☐ Log Assembler Activity

Source Code | List File | Configuration | Tokens | Run Log

```
;------ The Main Program -----------------------------------------
Start:
        OUT     01      ; Turn off traffic lights
        MOV     BL,10   ; A short delay.
        MOV     AL,84   ; RED AND GREEN(10000100)
        OUT     01      ; Send AL to port One
        CALL    36      ; Call the procedure at address [36]

        MOV     BL,1    ; A middle sized delay.
        MOV     AL,48   ; Yellow and Yellow(01001000)
        OUT     01      ; Send AL to port One
        CALL    36      ; Call the procedure at address [36]

        MOV     BL,5    ; A Longer delay.
        MOV     AL,30   ; Green and Red(00110000)
        OUT     01      ; Send AL to port One
        CALL    36      ; Call the procedure at address [36]

JMP     Start   ; Jump back to the start.

; ----- Time Delay Procedure Stored At Address [36] -------------
        ORG     36      ; Generate machine code from address [36]

        PUSH    BL      ; Save AL on the stack.
        PUSHF           ; Save the CPU flags on the stack.
Rep:
        DEC     BL      ; Subtract one from AL.
        JNZ     REP     ; Jump back to Rep if AL was not Zero.

        POPF            ; Restore the CPU flags from the stack.
        POP     BL      ; Restore AL from the stack.

        RET             ; Return from the procedure.
; ---------------------------------------------------------------
        END
; ---------------------------------------------------------------
```



RAM Source Code View

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00	OUT	01	MOV	BL	10	MOV	AL	84	OUT	01	CALL	36	MOV	BL	1	MOV
10	AL	48	OUT	01	CALL	36	MOV	BL	5	MOV	AL	30	OUT	01	CALL	36
20	JMP	STAR	END	END	END	END	END	END	END	END	END	END	END	END	END	END
30	END	END	END	END	END	END	PUSH	BL	PUSHDEC	BL	JNZ	REP	POPF	POP	BL	
40	RET	END	END	END	END	END	END	END	END	END	END	END	END	END	END	END
50	END	END	END	END	END	END	END	END	END	END	END	END	END	END	END	END
60	END	END	END	END	END	END	END	END	END	END	END	END	END	END	END	END
70	END	END	END	END	END	END	END	END	END	END	END	END	END	END	END	END
80	END	END	END	END	END	END	END	END	END	END	END	END	END	END	END	END
90	END	END	END	END	END	END	END	END	END	END	END	END	END	END	END	END
A0	END	END	END	END	END	END	END	END	END	END	END	END	END	END	END	END
B0	END	END	END	END	END	END	END	END	END	END	END	END	00	10	16	
C0																
D0																
E0																
F0																

○ X Hexadecimal ○ Y ASCII ◉ Z Source

D:\Apps\Lab 8\210175C_08.ASM

File Edit View Examples Help

```
AL 00110000 30 +048   IP 00111001 39 +057
BL 00000101 05 +005   SP 10111100 BC -068
CL 00000000 00 +000   SR 00000000 00 +000
DL 00000000 00 +000        ISOZ
```

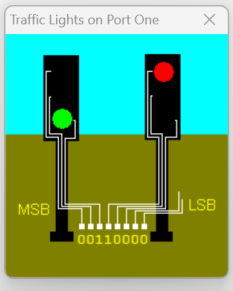
Assemble | Slower | Continue | Step | Faster | Cpu Reset | Run F9 | STOP | Show Ram

DEC BL

☐ Write Run Log ☐ Log Assembler Activity

Source Code | List File | Configuration | Tokens | Run Log

```
;------ The Main Program -----------------------------------------
Start:
        OUT     01      ; Turn off traffic lights
        MOV     BL,10   ; A short delay.
        MOV     AL,84   ; RED AND GREEN(10000100)
        OUT     01      ; Send AL to port One
        CALL    36      ; Call the procedure at address [36]

        MOV     BL,1    ; A middle sized delay.
        MOV     AL,48   ; Yellow and Yellow(01001000)
        OUT     01      ; Send AL to port One
        CALL    36      ; Call the procedure at address [36]

        MOV     BL,5    ; A Longer delay.
        MOV     AL,30   ; Green and Red(00110000)
        OUT     01      ; Send AL to port One
        CALL    36      ; Call the procedure at address [36]

JMP     Start   ; Jump back to the start.

; ----- Time Delay Procedure Stored At Address [36] -------------
        ORG     36      ; Generate machine code from address [36]

        PUSH    BL      ; Save AL on the stack.
        PUSHF           ; Save the CPU flags on the stack.
Rep:
        DEC     BL      ; Subtract one from AL.
        JNZ     REP     ; Jump back to Rep if AL was not Zero.

        POPF            ; Restore the CPU flags from the stack.
        POP     BL      ; Restore AL from the stack.

        RET             ; Return from the procedure.
; ---------------------------------------------------------------
        END
; ---------------------------------------------------------------
```



RAM Source Code View

```
    0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F
00 OUT 01  MOV BL  10  MOV AL  84  OUT 01  CALL36  MOV BL  1   MOV
10 AL  48  OUT 01  CALL36  MOV BL  5   MOV AL  30  OUT 01  CALL36
20 JMP STARTEND END END END END END END END END END END END END
30 END END END END END END PUSHBL  PUSHDEC BL  JNZ REP POPFPOP BL
40 RET END END END END END END END END END END END END END END
50 END END END END END END END END END END END END END END END
60 END END END END END END END END END END END END END END END
70 END END END END END END END END END END END END END END END
80 END END END END END END END END END END END END END END END
90 END END END END END END END END END END END END END END END
A0 END END END END END END END END END END END END END END END
B0 END END END END END END END END END END END END 00  05  20
C0
D0
E0
F0
```

X Hexadecimal Y ASCII Z Source

D:\Apps\Lab 8\210175C_10.ASM

File Edit View Examples Help

```
AL 11011101 DD -035   IP 00001010 0A +010
BL 00000000 00 +000   SP 10111111 BF -065
CL 00000000 00 +000   SR 00000000 00 +000
DL 00000000 00 +000         ISOZ
```

Assemble | Slower | Continue
Step | Faster | Cpu Reset
Run F9 | STOP | Show Ram

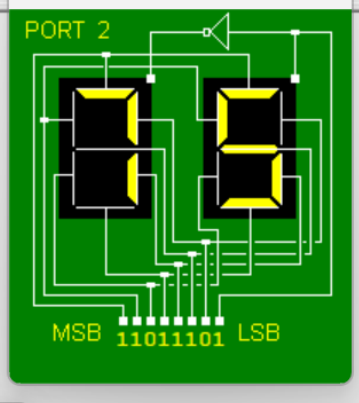
JMP Start

☐ Write Run Log ☐ Log Assembler Activity

Source Code | List File | Configuration | Tokens | Run Log

```
; ====================================================
; ===== 99sevseg.asm ================================
; ===== Seven Segment Displays Port 02 ==============
Start:
        MOV     AL,8A   ; (210175) 10001010
        OUT     02      ; Send the data in AL to Port 02

        MOV     AL,DD   ; 1101 1101
        OUT     02      ; Send the data in AL to Port 02

        JMP     Start

        END
; ====================================================
```

Seven Segment Displays



RAM Source Code View

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00	MOV	AL	8A	OUT	02	MOV	AL	DD	OUT	02	JMP	STAR	END	END	END	END
10	END	END	END	END	END	END	END	END	END	END	END	END	END	END	END	END
20	END	END	END	END	END	END	END	END	END	END	END	END	END	END	END	END
30	END	END	END	END	END	END	END	END	END	END	END	END	END	END	END	END
40	END	END	END	END	END	END	END	END	END	END	END	END	END	END	END	END
50	END	END	END	END	END	END	END	END	END	END	END	END	END	END	END	END
60	END	END	END	END	END	END	END	END	END	END	END	END	END	END	END	END
70	END	END	END	END	END	END	END	END	END	END	END	END	END	END	END	END
80	END	END	END	END	END	END	END	END	END	END	END	END	END	END	END	END
90	END	END	END	END	END	END	END	END	END	END	END	END	END	END	END	END
A0	END	END	END	END	END	END	END	END	END	END	END	END	END	END	END	END
B0	END	END	END	END	END	END	END	END	END	END	END	END	END	END	END	END
C0																
D0																
E0																
F0																

◯ X Hexadecimal ◯ Y ASCII ◉ Z Source

D:\Apps\Lab 8\210175C_11.ASM

File Edit View Examples Help

```
AL 11111111 FF -001   IP 00010111 17 +023
BL 01111000 78 +120   SP 10111111 BF -065
CL 00000000 00 +000   SR 00000010 02 +002
DL 00000000 00 +000         ISOZ
```

Assemble | Slower | Continue | Step | Faster | Cpu Reset | Run F9 | STOP | Show Ram

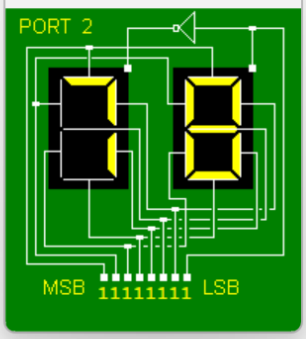
END Program has halted.

☐ Write Run Log ☐ Log Assembler Activity

Source Code | List File | Configuration | Tokens | Run Log

```
; ===== Counting ================================


        MOV     BL,1    ; Initial value stored in BL
        MOV     CL,5    ; Decrement from 5

Rep:                    ; Jump back to this label
        MUL     BL,CL   ; Multiply in each loop
        DEC     CL      ; Dec ONE from CL
        JNZ     Rep     ; Jump back to Rep

        MOV     AL,8A   ; Store 10001010 in AL
        OUT     02      ; Send AL to port 2
        MOV     AL,FF   ; Store 11111111 in AL
        OUT     02      ; Send AL to port 2
        END             ; Program Ends

; ===== Program Ends ============================
```

Seven Segment Displays



RAM Source Code View

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00	MOV	BL	1	MOV	CL	5	MUL	BL	CL	DEC	CL	JNZ	REP	MOV	AL	8A
10	OUT	02	MOV	AL	FF	OUT	02	END	END	END	END	END	END	END	END	END
20	END	END	END	END	END	END	END	END	END	END	END	END	END	END	END	END
30	END	END	END	END	END	END	END	END	END	END	END	END	END	END	END	END
40	END	END	END	END	END	END	END	END	END	END	END	END	END	END	END	END
50	END	END	END	END	END	END	END	END	END	END	END	END	END	END	END	END
60	END	END	END	END	END	END	END	END	END	END	END	END	END	END	END	END
70	END	END	END	END	END	END	END	END	END	END	END	END	END	END	END	END
80	END	END	END	END	END	END	END	END	END	END	END	END	END	END	END	END
90	END	END	END	END	END	END	END	END	END	END	END	END	END	END	END	END
A0	END	END	END	END	END	END	END	END	END	END	END	END	END	END	END	END
B0	END	END	END	END	END	END	END	END	END	END	END	END	END	END	END	END
C0																
D0																
E0																
F0																

◯ X Hexadecimal ◯ Y ASCII ◉ Z Source