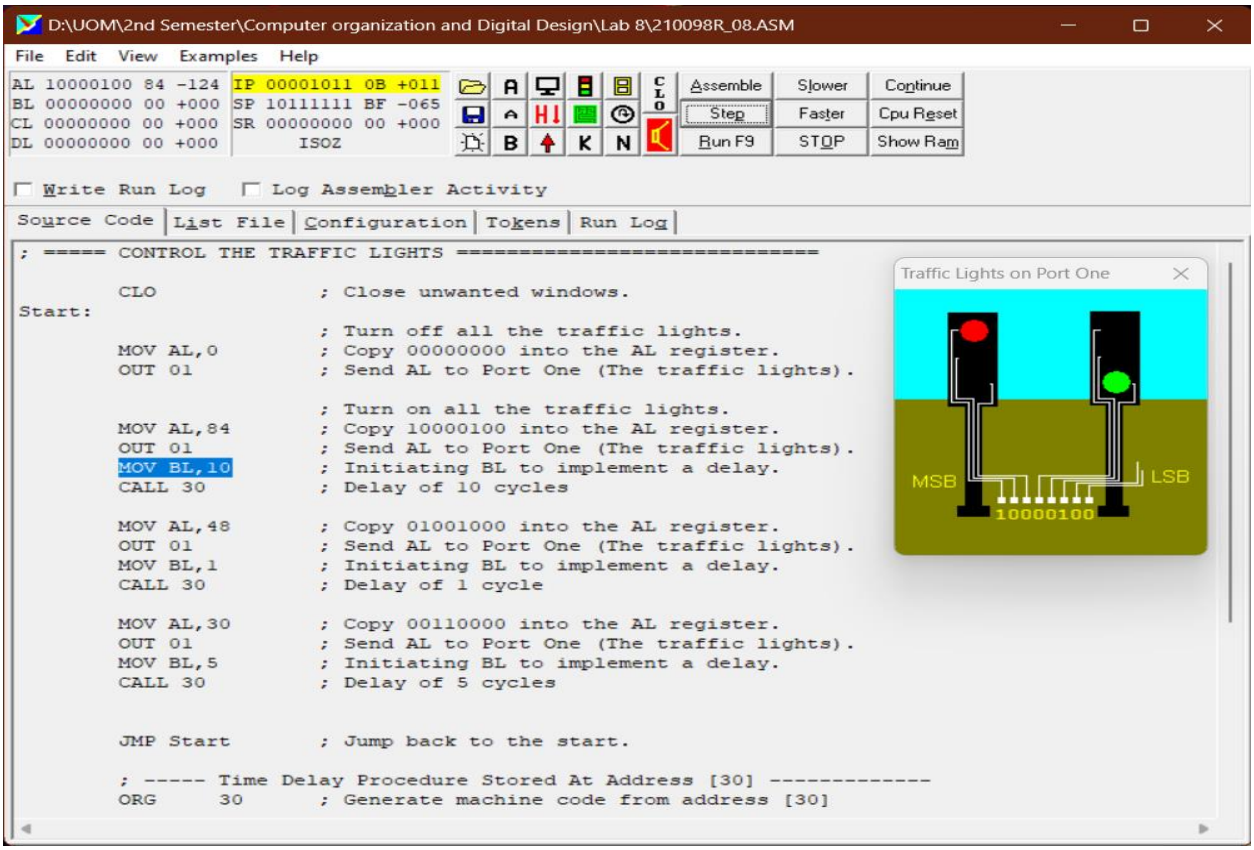
Student Name – De Silva APC

Index Number – 210098R

Assigned Lab tasks –

designing and developing simple Assembly programs to achieve given objectives, interface simple input and output devices to the microprocessor and finally verifyng their functionality via simulation using “smz32” was the core target of the lab. In the process its asked to first to read the documentation and then go through some sample implementations. Then we were been asked to modify some codes given to implement some functionalities such as two traffic lights and a seven segment displays that display last two digits of my index number and afterwards the multiplication result of numbers from 1 to 5.

Screenshots – **(First three – Traffic lights, 4th – Index Number, 5th – 1*2*3*4*5 result display)**



D:\UOM\2nd Semester\Computer organization and Digital Design\Lab 8\210098R_08.ASM

File Edit View Examples Help

```
AL 01001000 48 +072   IP 00010101 15 +021
BL 00010000 10 +016   SP 10111111 BF -065
CL 00000000 00 +000   SR 00000000 00 +000
DL 00000000 00 +000         ISOZ
```

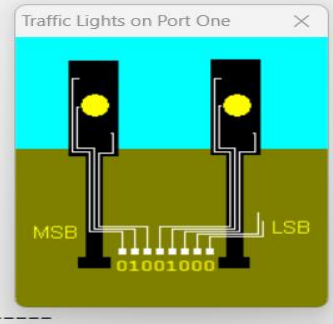
Assemble | Slower | Continue | Step | Faster | Cpu Reset | Run F9 | STOP | Show Ram

MOV BL,1

Write Run Log Log Assembler Activity

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

```
        MOV BL,10       ; Initiating BL to implement a delay.
        CALL 30         ; Delay of 10 cycles

        MOV AL,48       ; Copy 01001000 into the AL register.
        OUT 01          ; Send AL to Port One (The traffic lights).
        MOV BL,1        ; Initiating BL to implement a delay.
        CALL 30         ; Delay of 1 cycle

        MOV AL,30       ; Copy 00110000 into the AL register.
        OUT 01          ; Send AL to Port One (The traffic lights).
        MOV BL,5        ; Initiating BL to implement a delay.
        CALL 30         ; Delay of 5 cycles


        JMP Start       ; Jump back to the start.

        ; ----- Time Delay Procedure Stored At Address [30] --------------
        ORG     30      ; Generate machine code from address [30]

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

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

        RET             ; Return from the procedure.
```



D:\UOM\2nd Semester\Computer organization and Digital Design\Lab 8\210098R_08.ASM

File Edit View Examples Help

```
AL 00110000 30 +048   IP 00011111 1F +031
BL 00000001 01 +001   SP 10111111 BF -065
CL 00000000 00 +000   SR 00000000 00 +000
DL 00000000 00 +000         ISOZ
```

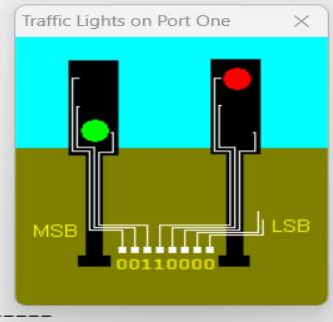
Assemble | Slower | Continue | Step | Faster | Cpu Reset | Run F9 | STOP | Show Ram

MOV BL,5

Write Run Log Log Assembler Activity

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

```
        MOV BL,10       ; Initiating BL to implement a delay.
        CALL 30         ; Delay of 10 cycles

        MOV AL,48       ; Copy 01001000 into the AL register.
        OUT 01          ; Send AL to Port One (The traffic lights).
        MOV BL,1        ; Initiating BL to implement a delay.
        CALL 30         ; Delay of 1 cycle

        MOV AL,30       ; Copy 00110000 into the AL register.
        OUT 01          ; Send AL to Port One (The traffic lights).
        MOV BL,5        ; Initiating BL to implement a delay.
        CALL 30         ; Delay of 5 cycles


        JMP Start       ; Jump back to the start.

        ; ----- Time Delay Procedure Stored At Address [30] --------------
        ORG     30      ; Generate machine code from address [30]

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

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

        RET             ; Return from the procedure.
```



D:\UOM\2nd Semester\Computer organization and Digital Design\Lab 8\210098R_10.ASM

File Edit View Examples Help

```
AL 11111111 FF -001  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
```

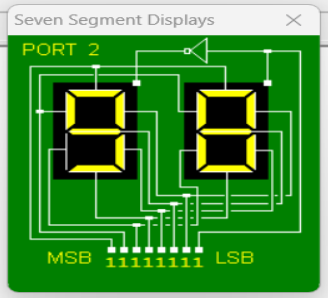
Assemble | Slower | Continue | Step | Faster | Cpu Reset | Run F9 | STOP | Show Ram

END Program has halted.

Write Run Log Log Assembler Activity

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

```
; ========================================================
; ===== 99sevseg.asm ====================================
; ===== Seven Segment Displays Port 02 ==================
Start:
        MOV     AL,DE    ; 1101 1110
        OUT     02       ; Send the data in AL to Port 02

        MOV     AL,FF    ; 1111 1111
        OUT     02       ; Send the data in AL to Port 02
        END
; ========================================================
```



D:\UOM\2nd Semester\Computer organization and Digital Design\Lab 8\210098R_11.ASM

File Edit View Examples Help

```
AL 11111111 FF -001  IP 00011100 1C +028
BL 00000110 06 +006  SP 10111011 BB -069
CL 01111000 78 +120  SR 00001100 0C +012
DL 00000000 00 +000        ISOZ
```

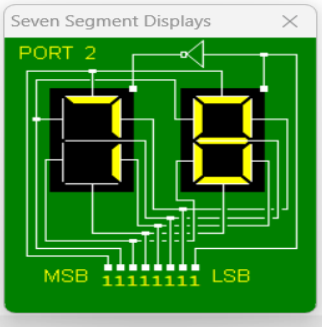
Assemble | Slower | Continue | Step | Faster | Cpu Reset | Run F9 | STOP | Show Ram

END Program has halted.

Write Run Log Log Assembler Activity

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

```
; ===== multiply 1 to 5 ============================

CLO             ; Close unwanted windows.
Start:
        MOV CL,1        ; Copy a 1 into the CL register.
        MOV BL,1        ; Copy a 1 into the BL register.
here:
        INC BL          ; Increment Bl by 1.
        PUSH CL         ; Saving CLs value because its going to end up overflown.
        MUL CL,BL       ; Multiply CL with BL and save to CL.
        JNO here        ; Jump while no overflow
        POP CL          ; After loop restoring the multiplication result.

;====== Below code will implement the displaying the result 78. ===

        MOV AL,8A       ; 1000 1010
        OUT 02          ; Send the data in AL to Port 02

        MOV AL,FF       ; 1111 1111
        OUT 02          ; Send the data in AL to Port 02

        END
;
```



01FIRSTAdd.asm

```
; ===== WORK OUT 2 PLUS 2 ====================================
        CLO             ; Close unwanted windows.
        MOV AL,3        ; Copy a 3 into the AL register.
        MOV BL,6        ; Copy a 6 into the BL register.
        ADD AL,BL       ; Add AL to BL. Answer goes into AL.
        END;
; ===== Program Ends ========================================
```

01FIRSTSub.asm

```
; ===== WORK OUT 2 PLUS 2 ====================================
        CLO             ; Close unwanted windows.
        MOV BL,7        ; Copy a 7 into the BL register.
        MOV CL,3        ; Copy a 3 into the CL register.
        SUB CL,BL       ; Substract BL from CL. Answer goes into CL.
        END;
; ===== Program Ends ========================================
```

01FIRSTMul.asm

```
; ===== WORK OUT 2 PLUS 2 ====================================
        CLO             ; Close unwanted windows.
        MOV BL,7        ; Copy a 7 into the BL register.
        MOV DL,10       ; Copy a 10 into the DL register.
        MUL DL,BL       ; Multiply DL and BL. Answer goes into DL.
        END;
; ===== Program Ends ========================================
```

01FIRSTDiv.asm

```
; ===== WORK OUT 2 PLUS 2 ====================================
        CLO             ; Close unwanted windows.
        MOV DL,13       ; Copy a 6 into the DL register.
        MOV AL,7        ; Copy a 7 into the AL register.
        DIV DL,AL       ; Divide DL by AL. Answer goes into DL.
        END;
; ===== Program Ends ========================================
```

210098R_08.asm

```
; ===== CONTROL THE TRAFFIC LIGHTS ============================

        CLO             ; Close unwanted windows.
Start:
                        ; Turn off all the traffic lights.
        MOV AL,0        ; Copy 00000000 into the AL register.
        OUT 01          ; Send AL to Port One (The traffic lights).

                        ; Turn on all the traffic lights.
        MOV AL,84       ; Copy 10000100 into the AL register.
        OUT 01          ; Send AL to Port One (The traffic lights).
        MOV BL,10       ; Initiating BL to implement a delay.
        CALL 30         ; Delay of 10 cycles

        MOV AL,48       ; Copy 01001000 into the AL register.
        OUT 01          ; Send AL to Port One (The traffic lights).
        MOV BL,1        ; Initiating BL to implement a delay.
```

```
        CALL 30         ; Delay of 1 cycle


        MOV AL,30       ; Copy 00110000 into the AL register.
        OUT 01          ; Send AL to Port One (The traffic lights).
        MOV BL,5        ; Initiating BL to implement a delay.
        CALL 30         ; Delay of 5 cycles



        JMP Start       ; Jump back to the start.

        ; ----- Time Delay Procedure Stored At Address [30] ----------
        ORG     30      ; Generate machine code from address [30]

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

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

        RET             ; Return from the procedure.
        END             ; Program ends.
```

210098R_10.asm

```
; ================================================
; ===== 99sevseg.asm ==============================
```

```
; ===== Seven Segment Displays Port 02 =================
Start:
        MOV     AL,DE   ; 1101 1110
        OUT     02      ; Send the data in AL to Port 02

        MOV     AL,FF   ; 1111 1111
        OUT     02      ; Send the data in AL to Port 02
        END
; ======================================================
```

210098R_11.asm

```
; ===== multiply 1 to 5 ============================

CLO             ; Close unwanted windows.
Start:
        MOV CL,1        ; Copy a 1 into the CL register.
        MOV BL,1        ; Copy a 1 into the BL register.
here:
        INC BL          ; Increment Bl by 1.
        PUSH CL                 ; Saving CLs value because its going to end up overflown.
        MUL CL,BL       ; Multiply CL with BL and save to CL.
        JNO here        ; Jump while no overflow
        POP CL          ; After loop restoring the multiplication result.

;====== Below code will implement the displaying the result 78. ===

        MOV AL,8A       ; 1000 1010
        OUT 02          ; Send the data in AL to Port 02
```

```
        MOV AL,FF       ; 1111 1111
        OUT 02          ; Send the data in AL to Port 02

        END
;
```