KURUKULASOORIYA KVR 200332X

ASSEMBLY PROGRAMMING

LAB 08 REPORT

TASK ASSIGNED:

- **Run the original and modified code files of 01FIRST.ASM, 03MOVE.ASM, 04INCJMP.ACM to get familiar with microprocessor simulator.**
- **Modify 02LIGHT.ASM as per the given table and reimplement an efficient code using a similar loop in 06PROC.ASM.**
- **Verify the functionality of modified 02LIGHT.ASM via running the simulation.**
- **Modify 99SEVSEG.ASM to display last two digits of the index number. (2003 '32' X)**
- **Create a program to calculate and display hexadecimal value of factorial 5.**

ALL ASSEMBLY FILES:

1.
```
; ===== WORK OUT 2 PLUS 2 ========================================
  CLO           ; Close unwanted windows.
  MOV AL,2      ; Copy a 2 into the AL register.
  MOV BL,2      ; Copy a 2 into the BL register.
  ADD AL,BL     ; Add AL to BL. Answer goes into AL.
  END           ; Program ends
; ===== Program Ends ============================================
```

2.
```
; ===== WORK OUT 2 MINUS 2 =======================================
  CLO           ; Close unwanted windows.
  MOV AL,2      ; Copy a 2 into the AL register.
  MOV BL,2      ; Copy a 2 into the BL register.
  SUB AL,BL     ; Substract BL from AL. Answer goes into AL.
  END           ; Program ends
; ===== Program Ends ============================================
```

3.
```
; ===== WORK OUT MULTIPLY 2 BY 2 ==================================
  CLO           ; Close unwanted windows.
  MOV AL,2      ; Copy a 2 into the AL register.
  MOV BL,2      ; Copy a 2 into the BL register.
  MUL AL,BL     ; Multiply AL by BL. Answer goes into AL.
  END           ; Program ends
; ===== Program Ends ============================================
```

4.
```
; ===== WORK OUT DIVIDE 2 BY 2 ====================================
  CLO           ; Close unwanted windows.
  MOV AL,2      ; Copy a 2 into the AL register.
  MOV BL,2      ; Copy a 2 into the BL register.
  DIV AL,BL     ; Divide AL to BL. Answer goes into AL.
  END           ; Program ends
; ===== Program Ends ============================================
```

5.
```
; ===== Decrementing ==============================================
  MOV   BL,40  ; Initial value stored in BL
  Rep:                ; Jump back to this label
        DEC   BL      ; Substract ONE to BL
        JMP   Rep     ; Jump back to Rep
        END           ; Program Ends
```

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

6.
```
   ; ===== Counting in 3s ==================================
   MOV BL,40  ; Initial value stored in BL
   Rep:                  ; Jump back to this label
      ADD    BL,3   ; Count In 3s
      JMP    Rep    ; Jump back to Rep
      END           ; Program Ends

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

7.
```
   ; ===== Counting in powers of 2 =========================
   MOV BL,1   ; Initial value stored in BL
   Rep:                  ; Jump back to this label
      MUL    BL,2   ; Count in powers of 2
      JMP    Rep    ; Jump back to Rep
      END           ; Program Ends

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

8.
```
   ; ===== Counting in fibonacci sums ======================
   MOV BL,0   ;      Initial value stored in BL
   MOV AL,1   ;      Initial value stored in BL
   MOV [50],AL       ;      Copy a AL value into the [50] location
   MOV [60],BL       ;      Copy a BL value into the [60] location

   REP:
      MOV    AL,[50]       ;Interchange the values between registers and memory locations
      MOV    BL,[60]       ;
      MOV    [60],AL       ;
      ADD    AL,BL  ;      Keep adding to each 2 sums to develop fibonacci sequence

      MOV    [50],AL       ;
      JMP    REP    ;

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

9.
```
   ; ===== CONTROL THE TRAFFIC LIGHTS (without loop) ========

      CLO;          Close unwanted windows.

Start:
      MOV AL,84;    Copy 10000100 into the AL register.
      OUT 01;       Red on left side and Green on right side.
      NOP;          Do nothing for 10 cycles
      NOP;
      NOP;
      NOP;
      NOP;
      NOP;
      NOP;
      NOP;
      NOP;
      NOP;

      MOV AL,48;    Copy 01001000 into the AL register.
      OUT 01;             Yellow on both sides.
      NOP;          Do  nothing for a cycle.

      MOV AL,30;    Copy 00110000 into the AL register.
      OUT 01;             Green on left side and Red on right side.
      NOP;          Do nothing for 5 cycles.
      NOP;
      NOP;
      NOP;
      NOP;

      JMP Start     ; Jump back to the start.
      END           ; Program ends.

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

10.
```
   ; ===== CONTROL THE TRAFFIC LIGHTS-Using a loop ==========

      CLO;          Close unwanted windows.
Start:

      MOV AL,84;    Copy 10000100 into the AL register.
      MOV BL,A;     10 cycles
```

```
       OUT 01;       Red on left side and Green on right side.
       CALL 30;      Call time delay procedure at 30

       MOV AL,48;    Copy 01001000 into the AL register.
       MOV BL,1;     1 cycle
       OUT 01;              Yellow on both sides.
       CALL 30;      Call time delay procedure at 30

       MOV AL,30;    Copy 00110000 into the AL register.
       MOV BL,5;     5 cycles
       OUT 01;              Green on left side and Red on right side.
       CALL 30;      Call time delay procedure


       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 CPU flags on the stack
REP:
       DEC BL;       Substract 1 from BL value
       JNZ REP;      If BL !=0 ,Jump back to REP

       POPF;         Restore the CPU flags from the stack to their original value
       POP BL;              Restore BL from the stack to original

       RET;          Return from the procedure.
;-------------------------------------------------------------------
       END;
; ===== Program Ends ============================================
```

11. `; ===== Seven Segment Displays Port 02 ==========================`

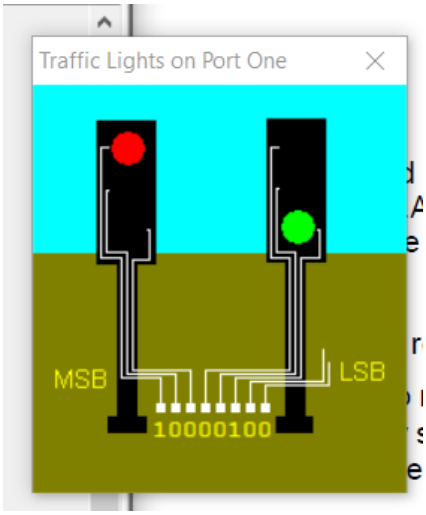
```
;Index number : 2003'32'X
Start:
       MOV    AL,9E  ; 1001 1110
       OUT    02     ; Send the data in AL to Port 02

       MOV    AL,B7  ; 1011 0111
       OUT    02     ; Send the data in AL to Port 02

       JMP    Start

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

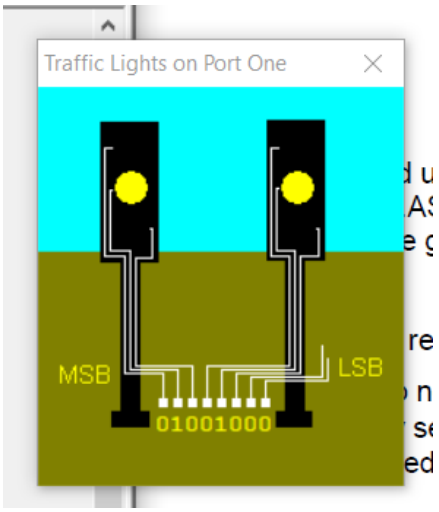
12. `;===========Display Hex value of factorial 5==================`

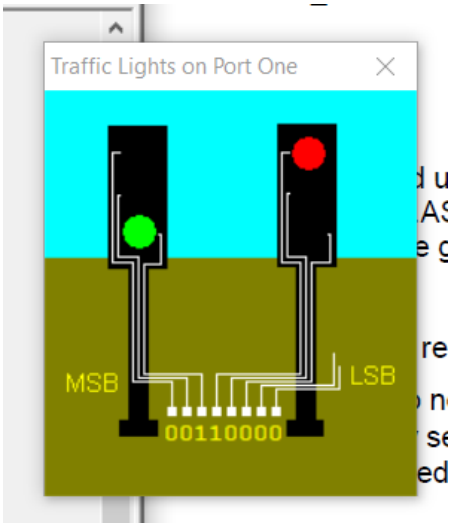
```
MOV    AL,1;  Copy 1 to AL register
MOV    BL,5;  Copy 5 to BL register

REP:
       MUL    AL,BL;        Multiply the values at AL and BL, the answer goes to AL
       DEC    BL;    Decrement BL register by 1
       JNZ    REP;   If BL != 0 then repeat

;Anwer is 120 in decimal and 78 when converted to Hexadecimal
MOV    AL,8A; 1000 1010
OUT    02;    Send the value in Al to port2

MOV    AL,FF; 1111 1111
OUT    02;    Send the value in Al to port2

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

SCREENSHOTS OF SIMULATION:

01.Simulation of traffic lights. *(FROM 200332X_8.ASM)*

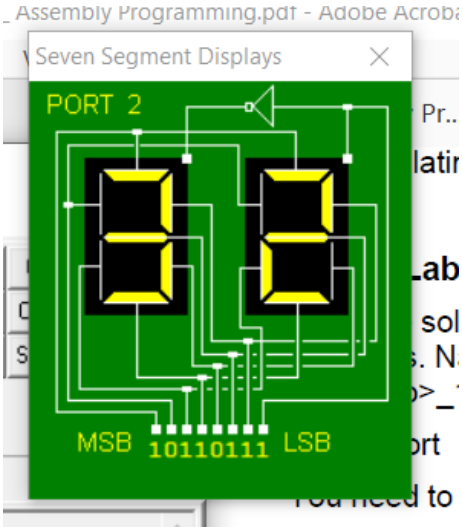






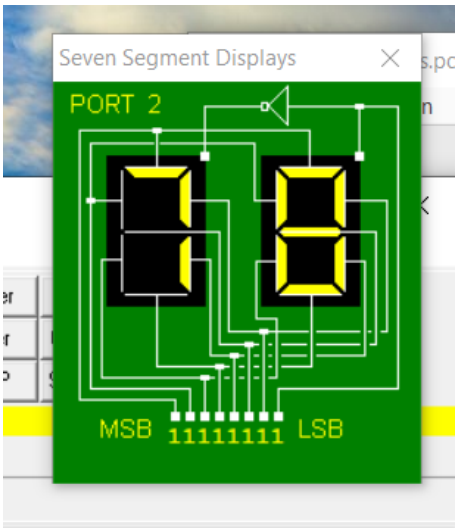
02.Simulation of last 2 digits of index number on the Seven segment.

(FROM 200332X_10.ASM)



03.Simulation of displaying the hexadecimal value of factorial 5 on the Seven segment.

(FROM 200332X_11.ASM)



CONCLUSIONS:

- Assembly language can be used to design and develop simple programs to achieve certain objectives and their functionality can be verified via running a simulation.
- By interfacing input and output devices to the microprocessor, the already developed logics can be put in practice in a useful way.