Name: V. Sajeevan.

Index Number: 160544C

Practical Date: 02.11.2017

Submission Date: 09.11.2017

Subject: CS 2052 Computer Architecture

REPORT ON LAB 8
ASSEMBLY PROGRMMING

Lab Task:

- Getting familiar with smz32
- Understand given assembly codes, execute them and try to modify them to get a clear understanding.
- Write a new assembly program to multiply all integers from 1 to 5, assembling them and simulating them.

Given/ modified code samples for Basic Operations:

Addition:

```
; ===== WORK OUT 5 PLUS 4 ================================
    CLO          ; Close unwanted windows.
    MOV AL,5     ; Copy a 5 into the AL register.
    MOV BL,4     ; Copy a 4 into the BL register.
    ADD AL,BL    ; Add AL to BL. Answer goes into AL.
    END          ; Program ends
; ===== Program Ends =====================================
```

```
AL 00001001 09 +009
BL 00000100 04 +004
CL 00000000 00 +000
DL 00000000 00 +000
```

Subtraction:

```
; ===== WORK OUT 7 Subtraction 2 ========================
    CLO          ; Close unwanted windows.
    MOV AL,7     ; Copy a 7 into the AL register.
    MOV BL,2     ; Copy a 2 into the BL register.
    SUB AL,BL    ; Add AL to BL. Answer goes into AL.
    END          ; Program ends
; ===== Program Ends =====================================
```

```
AL 00000101 05 +005
BL 00000010 02 +002
CL 00000000 00 +000
DL 00000000 00 +000
```

Multiplication:

```
; ===== WORK OUT 7 Multiplication 3 ======================
    CLO           ; Close unwanted windows.
    MOV AL,7      ; Copy a 7 into the AL register.
    MOV BL,3      ; Copy a 3 into the BL register.
    MUL AL,BL     ; Add AL to BL. Answer goes into AL.
    END           ; Program ends
; ===== Program Ends ======================================
```

```
AL 00010101 15 +021
BL 00000011 03 +003
CL 00000000 00 +000
DL 00000000 00 +000
```

Division:

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

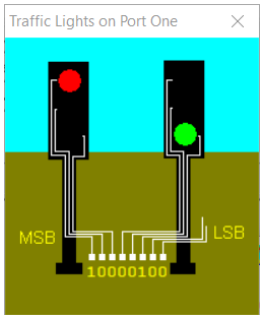
```
AL 00000010 02 +002
BL 00000011 03 +003
CL 00000000 00 +000
DL 00000000 00 +000
```

Signal Light:

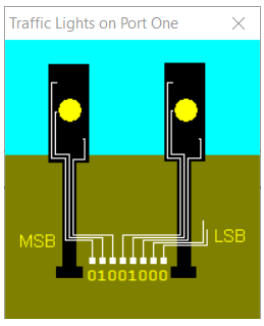
Assembly code for the signal light:

```
; ===== CONTROL THE TRAFFIC LIGHTS ========================================
    CLO           ; Close unwanted windows.
Start:
    MOV AL,84     ; Copy 1000 0100 into the AL register.
    OUT 01        ; Send AL to Port One (The traffic lights).
    MOV BL,A      ; Waiting time 10Cycles (A) Copy A into the BL register.
    CALL 30       ; Call the procedure at address [30]

    MOV AL,48     ; Copy 0100 1000 into the AL register.
    OUT 01        ; Send AL to Port One (The traffic lights).
    MOV BL,1      ; Waiting time 1 - Copy 1 into the BL register.
    CALL 30       ; Call the procedure at address [30]

    MOV AL,30     ; Copy 0011 0000 into the AL register.
    OUT 01        ; Send AL to Port One (The traffic lights).
    MOV BL,5      ; Waiting time 5 - Copy 5 into the BL register.
    CALL 30       ; Call the procedure at address [30]
JMP 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 1 from BL
    JNZ REP       ; Jump back to REP if BL was not 0

    POPF          ; Restore the  CPU flags from the stack
    POP BL        ; Restore BL from the stack
    RET           ; Return
END
; ===== Program Ends ========================================================
```

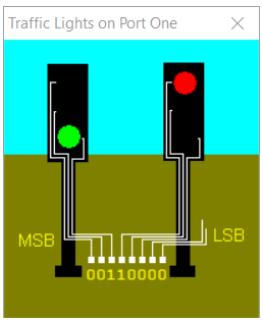
Simulation:



10 Cycles



1 Cycle



5 Cycles

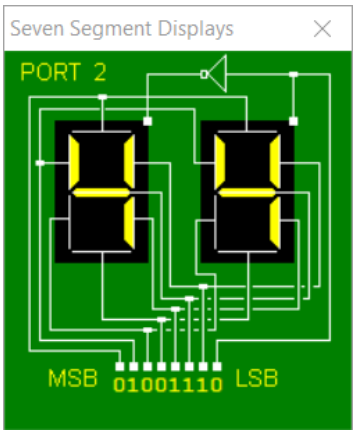
7-Segmented display:

I had to display the last two digits of my index number. Since my index number is 160544C, I had to display 44.

Assembly code:

```
; ========================================================
; ===== Seven Segment Displays Port 02 ==================
; ===== My Index No. is 160544C I want to  show 44 =====
Start:
                    ;
    MOV AL,4E       ; 01001110
    OUT 02          ; Send the data in AL to Port 02

    MOV AL,4F       ; 01001111
    OUT 02          ; Send the data in AL to Port 02

    JMP Start

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

The simulation on 7-segmented display:



Multiply all integers from 1 to 5 and display:

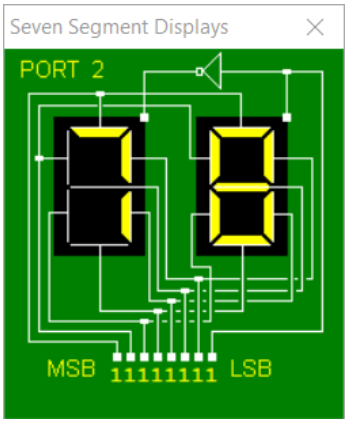
Assembly Code:

```
;An assembly program to multiply all integers from 1 to 5
;=====================================================================
    CLO         ;Close unwanted windows
    MOV AL,1    ;Copy 1 into AL
    MOV BL,5    ;Copy 5 into BL
LOOP:
    MUL AL,BL   ;Multiply AL and BL. Answer goes into AL
    DEC BL      ;Decrease BL by 1
    JNZ LOOP    ;Jump back to LOOP if BL is not zero
;=====================================================================
```

```
;=====================================================================
;To display the vlue on a seven segment
    MOV [C1],AL     ;Copy value in AL to ram location [C1]
    DIV AL,10   ;Divide AL by 10
    MOV [C2],AL ;Copy value in AL to ram location [C2]
    CALL 30     ;Call the procedure at address [30]

    OUT 2       ;Send AL to port 02
    MOV BL,[C2]     ;Copy value in ram location [C2] to BL
    MUL BL,10   ;Multiply BL by 10
    MOV AL,[C1]     ;Copy value in ram location [C1] to AL
    SUB AL,BL   ;Subtract BL from Al
    CALL 30     ;Call the procedure at address [30]

    INC AL      ;Increase AL by 1
    OUT 2       ;Send AL to port 02

    ORG 30      ;Generate machine code from address [30]

    JNZ A       ;If AL is not zero, go to A
    JMP BA      ;As AL is zero, go to BA
```

```
70
71   BA:   MOV AL,FA   ;Copy hexa FA to AL
72     JMP ASD     ;Go to ASD
73   BB:   MOV AL,0A   ;Copy hexa 0A to AL
74     JMP ASD     ;Go to ASD
75   BC:   MOV AL,B6   ;Copy hexa B6 to AL
76     JMP ASD     ;Go to ASD
77   BD:   MOV AL,9E   ;Copy hexa 9E to AL
78     JMP ASD     ;Go to ASD
79   BE:   MOV AL,4E   ;Copy hexa 4E to AL
80     JMP ASD     ;Go to ASD
81   BF:   MOV AL,DC   ;Copy hexa DC to AL
82     JMP ASD     ;Go to ASD
83   BG:   MOV AL,FC   ;Copy hexa FC to AL
84     JMP ASD     ;Go to ASD
85   BH:   MOV AL,8A   ;Copy hexa 8A to AL
86     JMP ASD     ;Go to ASD
87   BI:   MOV AL,FE   ;Copy hexa FE to AL
88     JMP ASD     ;Go to ASD
89   BJ:   MOV AL,DE   ;Copy hexa DE to AL
90     JMP ASD     ;Go to ASD
91
92   ASD:  MOV [C3],AL     ;Move AL to ram location [C3]
93     RET         ; Return from the procedure.
94
95     END
96 ;============Program Ends=============================================
```

```
33
34   A:    DEC AL      ;Decrease AL by 1
35     JNZ B       ;If AL is not zero, go to B
36     JMP BB      ;As AL is zero, go to BB
37
38   B:    DEC AL      ;Decrease AL by 1
39     JNZ C       ;If AL is not zero, go to C
40     JMP BC      ;As AL is zero, go to BC
41
42   C:    DEC AL      ;Decrease AL by 1
43     JNZ D       ;If AL is not zero, go to D
44     JMP BD      ;As AL is zero, go to BD
45
46   D:    DEC AL      ;Decrease AL by 1
47     JNZ E       ;If AL is not zero, go to E
48     JMP BE      ;As AL is zero, go to BE
49
50   E:    DEC AL      ;Decrease AL by 1
51     JNZ F       ;If AL is not zero, go to F
52     JMP BF      ;As AL is zero, go to BF
53
54   F:    DEC AL      ;Decrease AL by 1
55     JNZ G       ;If AL is not zero, go to G
56     JMP BG      ;As AL is zero, go to BG
57
58   G:    DEC AL      ;Decrease AL by 1
59     JNZ H       ;If AL is not zero, go to H
60     JMP BH      ;As AL is zero, go to BH
61
62   H:    DEC AL      ;Decrease AL by 1
63     JNZ I       ;If AL is not zero, go to I
64     JMP BI      ;As AL is zero, go to BI
65
66   I:    DEC AL      ;Decrease AL by 1
67     JNZ A       ;If AL is not zero, go to A
68     JMP BJ      ;As AL is zero, go to BJ
69
```



Output from 7-Segmented Display:
As the integers from 1 to 5 are multiplied, The result = $120_{10} = 78_{16}$

Conclusion:
At the end of this lab, I was able to,

- Get familiar with the common assembly instructions.
- Design and develop simple Assembly programs.
- Interface simple input and output devices.
- Check their functionality using simulation.