Lab 8

Assembly Programming

CS1050 Computer Organization and Digital Design

RATHNAYAKE R.M.I.B
220526N
Group 41

Introduction

Task:

Using the SMZ32 microprocessor emulator, we have to study assembly programming in this lab. First, we simulated changing the provided Assembly code of addition(01FIRST.ASM) by inserting the SUB, DIV, and MUL instructions to subtract, divide, and multiply the two values. Next, using loops to create longer delays, we altered a different example to program a color-light system using preset combinations and timings. For this 02TLIGHT.ASM code was changed. Then, we changed another provided assembly code (99SEVSEG.ASM) such that the last two digits of our index number appeared on the 7-segment display. Finally, a new assembly program was written that used loops to multiply the first 5 integers to get the factorial and two seven-segment displays to show the result as a hexadecimal number.

1. Basic Mathematical Operations

- **Addition**

```
; ===== WORK OUT 10 PLUS 2 ======================================
  CLO       ; Close unwanted windows.
  MOV AL,A  ; Copy a 10 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 ===========================================
```

- **Subtraction**

```
; ===== WORK OUT 5 MINUS 2 ======================================
  CLO       ; Close unwanted windows.
  MOV AL,5  ; Copy a 5 into the AL register.
  MOV BL,2  ; Copy a 2 into the BL register.
  SUB AL,BL ; Subtract AL to BL. Answer goes into AL.
  END       ; Program ends
; ===== Program Ends ===========================================
```

- **Multiplication**

```
; ===== WORK OUT 8 INTO 2 ======================================
  CLO       ; Close unwanted windows.
  MOV AL,8  ; Copy a 8 into the AL register.
  MOV BL,2  ; Copy a 2 into the BL register.
  MUL AL,BL ; Multiply AL to BL. Answer goes into AL.
  END       ; Program ends
; ===== Program Ends ===========================================
```

- **Division**

```
; ===== WORK OUT 6 DIVIDED BY 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 ; Divide AL to BL. Answer goes into AL.
  END       ; Program ends
; ===== Program Ends ===========================================
```

2. Traffic Lights

- **explanation of the implementation of delaying.**

Time	Lights on Left	Lights on Right	CPU CYCLES
t	Red	Green	10
t + 10	Yellow	Yellow	1
t + 11	Green	Red	5
t + 16	Red	Green	10
t + 26	Yellow	Yellow	1
…	…	…	…

  - We use a single loop to observe the delays. The required time loads to a register as a hex value. Then decrease its value to 0 using a loop.

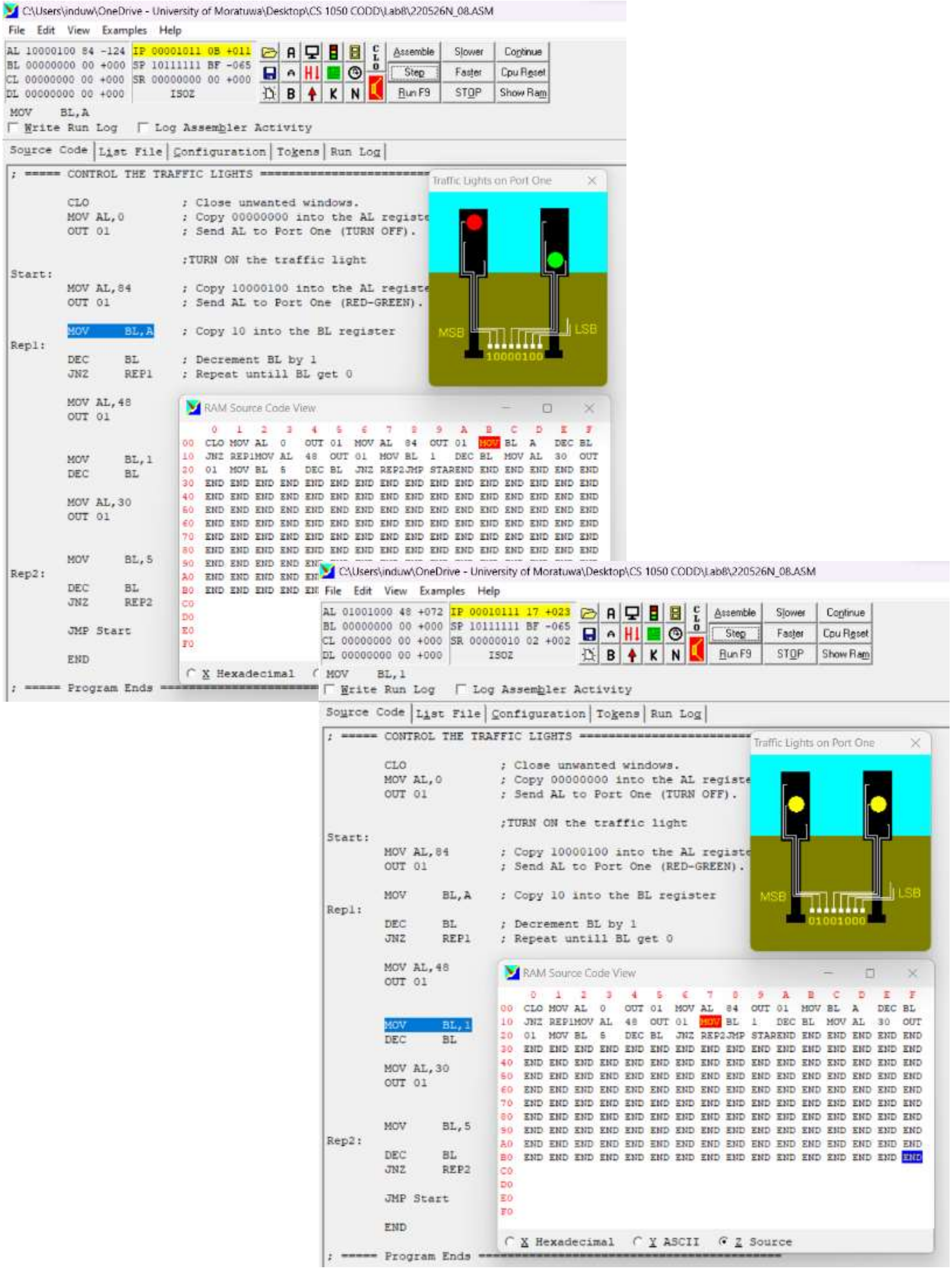
  - i.e. If we need to delay 10 CPU cycles, we can follow these steps:
    1. MOV BL,A : loads the BL register with value 10
    2. DEC BL : decrease the value of the BL register by 1
    3. JNZ REP1 : jump back to the beginning of the loop(Rep1) if not the BL is zero

    This iteration will happen until the BL value is zero.

- **Sample code**

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


   CLO        ; Close unwanted windows.

   MOV AL,0   ; Copy 00000000 into the AL register.

   OUT 01     ; Send AL to Port One (TURN OFF).


             ;TURN ON the traffic light

Start:

   MOV AL,84 ; Copy 10000100 into the AL register.

   OUT 01     ; Send AL to Port One (RED-GREEN).
   MOV  BL,A ; Copy 10 into the BL register
Rep1:
   DEC  BL   ; Decrement BL by 1
   JNZ  REP1 ; Repeat until BL get 0

   MOV AL,48 ; Copy 01001000 into the AL register.
   OUT 01    ; Send AL to Port One (YELLOW-YELLOW).


   MOV  BL,1 ; Copy 1 into the BL register
   DEC  BL   ; Decrement BL by 1

   MOV AL,30 ; Copy 00110000 into the AL register.
   OUT 01    ; Send AL to Port One (GREEN-RED).


   MOV  BL,5 ; Copy 5 into the BL register
Rep2:
   DEC  BL   ; Decrement BL by 1
   JNZ  REP2 ; Repeat until BL get 0

   JMP Start ; Jump back to the start.

   END       ; Program ends.

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

- **Screenshots**



C:\Users\induw\OneDrive - University of Moratuwa\Desktop\CS 1050 CODD\Lab8\220526N_08.ASM

File Edit View Examples Help
Assemble
Slower
Continue
Step
Faster
Cpu Reset
Run F9
STOP
Show Ram
CLO

```
AL 10000100 84 -124   IP 00001011 0B +011
BL 00000000 00 +000   SP 10111111 BF -065
CL 00000000 00 +000   SR 00000000 00 +000
DL 00000000 00 +000          ISOZ
MOV     BL,A
```

□ Write Run Log □ Log Assembler Activity

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

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

        CLO                 ; Close unwanted windows.
        MOV AL,0            ; Copy 00000000 into the AL registe
        OUT 01              ; Send AL to Port One (TURN OFF).

                            ;TURN ON the traffic light
Start:
        MOV AL,84           ; Copy 10000100 into the AL registe
        OUT 01              ; Send AL to Port One (RED-GREEN).

        MOV      BL,A       ; Copy 10 into the BL register
Rep1:
        DEC      BL         ; Decrement BL by 1
        JNZ      REP1       ; Repeat untill BL get 0

        MOV AL,48
        OUT 01


        MOV      BL,1
        DEC      BL

        MOV AL,30
        OUT 01


        MOV      BL,5
Rep2:
        DEC      BL
        JNZ      REP2

        JMP Start

        END

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

RAM Source Code View

```
    0   1   2   3   4   5   6   7   8   9   A   B   C   D   E   F
00  CLO MOV AL  0   OUT 01  MOV AL  84  OUT 01  MOV BL  A   DEC BL
10  JNZ REP1MOV AL  48  OUT 01  MOV BL  1   DEC BL  MOV AL  30  OUT
20  01  MOV BL  5   DEC BL  JNZ REP2JMP STAREND 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 EN
A0  END END END END EN
B0  END END END END EN
C0
D0
E0
F0
```

○ X Hexadecimal ○

C:\Users\induw\OneDrive - University of Moratuwa\Desktop\CS 1050 CODD\Lab8\220526N_08.ASM

```
AL 01001000 48 +072   IP 00010111 17 +023
BL 00000000 00 +000   SP 10111111 BF -065
CL 00000000 00 +000   SR 00000010 02 +002
DL 00000000 00 +000          ISOZ
MOV     BL,1
```

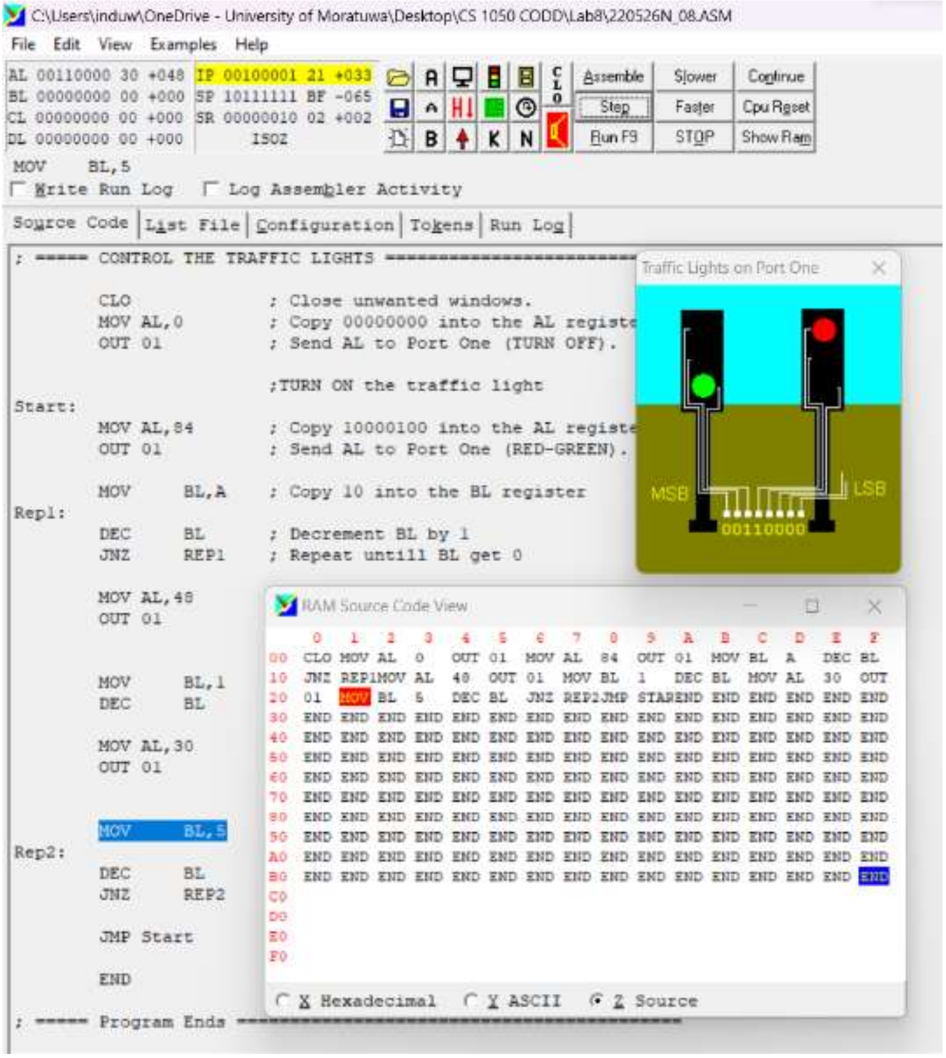
□ Write Run Log □ Log Assembler Activity

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

        CLO                 ; Close unwanted windows.
        MOV AL,0            ; Copy 00000000 into the AL registe
        OUT 01              ; Send AL to Port One (TURN OFF).

                            ;TURN ON the traffic light
Start:
        MOV AL,84           ; Copy 10000100 into the AL registe
        OUT 01              ; Send AL to Port One (RED-GREEN).

        MOV      BL,A       ; Copy 10 into the BL register
Rep1:
        DEC      BL         ; Decrement BL by 1
        JNZ      REP1       ; Repeat untill BL get 0

        MOV AL,48
        OUT 01


        MOV      BL,1
        DEC      BL

        MOV AL,30
        OUT 01


        MOV      BL,5
Rep2:
        DEC      BL
        JNZ      REP2

        JMP Start

        END

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

RAM Source Code View

```
    0   1   2   3   4   5   6   7   8   9   A   B   C   D   E   F
00  CLO MOV AL  0   OUT 01  MOV AL  84  OUT 01  MOV BL  A   DEC BL
10  JNZ REP1MOV AL  48  OUT 01  MOV BL  1   DEC BL  MOV AL  30  OUT
20  01  MOV BL  5   DEC BL  JNZ REP2JMP STAREND 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


C:\Users\induw\OneDrive - University of Moratuwa\Desktop\CS 1050 CODD\Lab8\220526N_08.ASM
File
Edit
View
Examples
Help
Assemble
Slower
Continue
Step
Faster
Cpu Reset
Run F9
STOP
Show Ram
CLO
Traffic Lights on Port One
MSB
LSB
00110000

```
AL 00110000 30 +048   IP 00100001 21 +033
BL 00000000 00 +000   SP 10111111 BF -065
CL 00000000 00 +000   SR 00000010 02 +002
DL 00000000 00 +000         ISOZ
```

```
MOV     BL,5
```

□ Write Run Log   □ Log Assembler Activity

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

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

        CLO             ; Close unwanted windows.
        MOV AL,0        ; Copy 00000000 into the AL registe
        OUT 01          ; Send AL to Port One (TURN OFF).

                        ;TURN ON the traffic light
Start:
        MOV AL,84       ; Copy 10000100 into the AL registe
        OUT 01          ; Send AL to Port One (RED-GREEN).

        MOV     BL,A    ; Copy 10 into the BL register
Rep1:
        DEC     BL      ; Decrement BL by 1
        JNZ     REP1    ; Repeat untill BL get 0

        MOV AL,48
        OUT 01


        MOV     BL,1
        DEC     BL

        MOV AL,30
        OUT 01


        MOV     BL,5
Rep2:
        DEC     BL
        JNZ     REP2

        JMP Start

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

RAM Source Code View

```
    0   1   2   3   4   5   6   7   8   9   A   B   C   D   E   F
00  CLO MOV AL  0   OUT 01  MOV AL  84  OUT 01  MOV BL  A   DEC BL
10  JNZ REP1MOV AL  48  OUT 01  MOV BL  1   DEC BL  MOV AL  30  OUT
20  01  MOV BL  5   DEC BL  JNZ REP2JMP STAREND 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
```

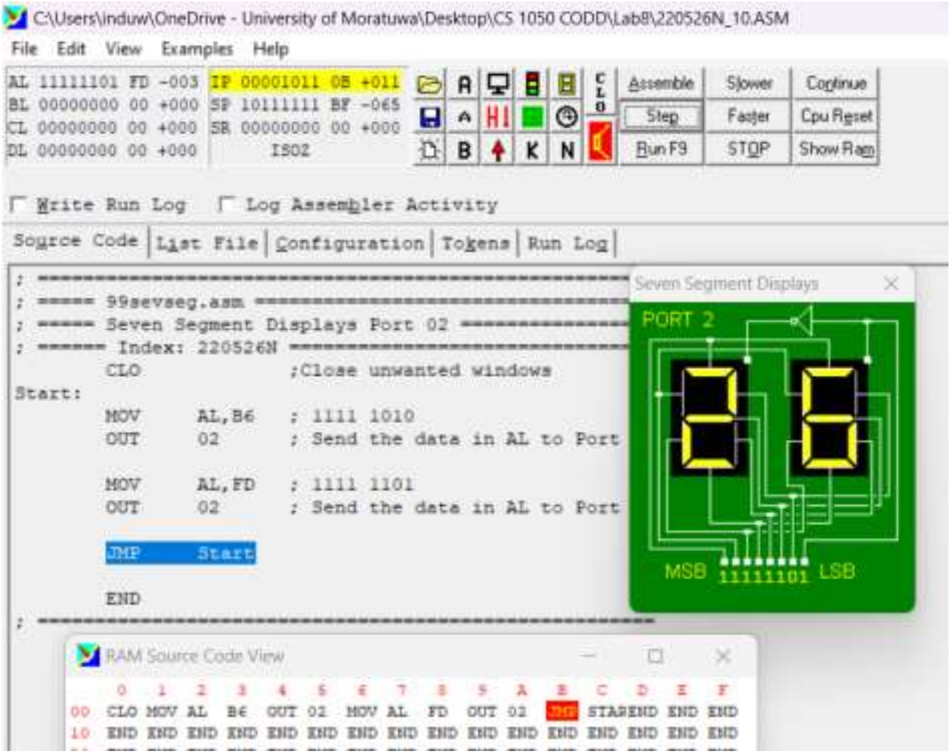
○ X Hexadecimal ○ Y ASCII ● Z Source

3. Seven-Segment Display

- **Sample code**

```
; ==========================================================
; ===== 99sevseg.asm =======================================
; ===== Seven Segment Displays Port 02 =================
; ====== Index: 220526N ==================================
     CLO            ;Close unwanted windows
Start:
     MOV  AL,B6     ; 1111 1010
     OUT  02        ; Send the data in AL to Port 02 (2)

     MOV  AL,FD     ; 1111 1101
     OUT  02        ; Send the data in AL to Port 02 (6)

     JMP  Start
     END
; ==========================================================
```

- **Screenshot**



C:\Users\induw\OneDrive - University of Moratuwa\Desktop\CS 1050 CODD\Lab8\220526N_10.ASM

```
AL 11111101 FD -003   IP 00001011 0B +011
BL 00000000 00 +000   SP 10111111 BF -065
CL 00000000 00 +000   SR 00000000 00 +000
DL 00000000 00 +000          ISO2
```

□ Write Run Log   □ Log Assembler Activity

```
; ====================================================
; ===== 99sevseg.asm =================================
; ===== Seven Segment Displays Port 02 ===============
; ====== Index: 220526N ==============================
        CLO             ;Close unwanted windows
Start:
        MOV     AL,B6   ; 1111 1010
        OUT     02      ; Send the data in AL to Port

        MOV     AL,FD   ; 1111 1101
        OUT     02      ; Send the data in AL to Port

        JMP     Start

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

Seven Segment Displays

PORT 2

MSB 11111101 LSB

RAM Source Code View

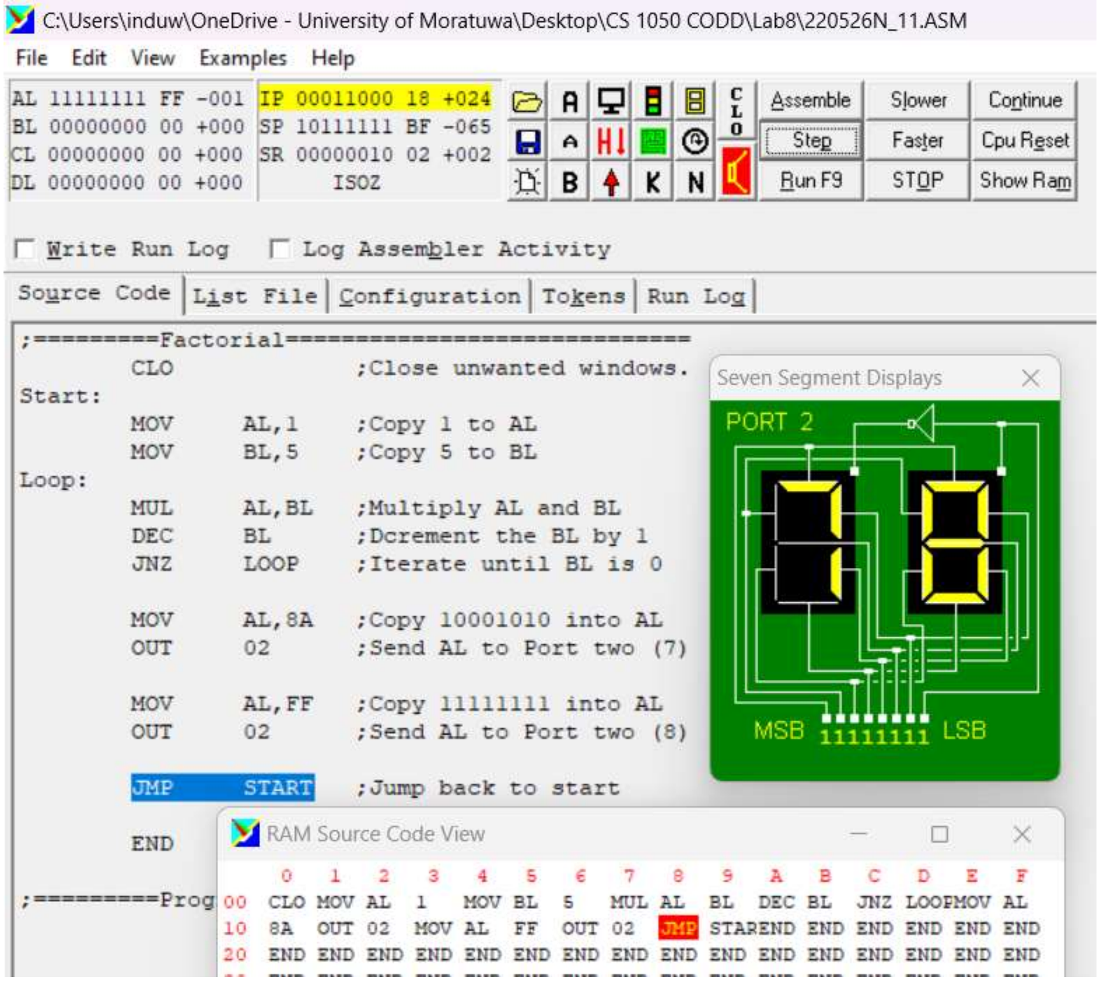
```
     0   1   2   3   4   5   6   7   8   9   A   B   C   D   E   F
00  CLO MOV AL  B6  OUT 02  MOV AL  FD  OUT 02  JMP STAREND END END
10  END END END END END END END END END END END END END END END END
```

4. Factorial

- **Sample code**

```
 ;=========Factorial=============================
     CLO                 ;Close unwanted windows.
Start:
     MOV  AL,1           ;Copy 1 to AL
     MOV  BL,5           ;Copy 5 to BL
Loop:
     MUL  AL,BL          ;Multiply AL and BL
     DEC  BL             ;Decrement the BL by 1
     JNZ  LOOP           ;Iterate until BL is 0

     MOV  AL,8A          ;Copy 10001010 into AL
     OUT  02             ;Send AL to Port two (7)

     MOV  AL,FF          ;Copy 11111111 into AL
     OUT  02             ;Send AL to Port two (8)

     JMP  START          ;Jump back to start

     END                 ;Program ends

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

- **Screenshot**



C:\Users\induw\OneDrive - University of Moratuwa\Desktop\CS 1050 CODD\Lab8\220526N_11.ASM

```
AL 11111111 FF -001   IP 00011000 18 +024
BL 00000000 00 +000   SP 10111111 BF -065
CL 00000000 00 +000   SR 00000010 02 +002
DL 00000000 00 +000          ISOZ
```

□ Write Run Log   □ Log Assembler Activity

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

```
;=========Factorial=============================
        CLO                 ;Close unwanted windows.
Start:
        MOV     AL,1        ;Copy 1 to AL
        MOV     BL,5        ;Copy 5 to BL
Loop:
        MUL     AL,BL       ;Multiply AL and BL
        DEC     BL          ;Dcrement the BL by 1
        JNZ     LOOP        ;Iterate until BL is 0

        MOV     AL,8A       ;Copy 10001010 into AL
        OUT     02          ;Send AL to Port two (7)

        MOV     AL,FF       ;Copy 11111111 into AL
        OUT     02          ;Send AL to Port two (8)

        JMP     START       ;Jump back to start

        END

;=========Prog
```

```
     0   1   2   3   4   5   6   7   8   9   A   B   C   D   E   F
00  CLO MOV AL  1   MOV BL  5   MUL AL  BL  DEC BL  JNZ LOOPMOV AL
10  8A  OUT 02  MOV AL  FF  OUT 02  JMP STAREND END END END END END
20  END END END END END END END END END END END END END END END END
```

Conclusion

Assembly programming is essential for understanding microprocessor operations and producing efficient code. By using a loop to add longer delays, you can introduce delays to a code in a more effective manner.