

# Computer Organization & Digital Design LAB 8

210495G – Nipun Viraj

## (1) LAB TASK

- We should create assembly language programs to do simple calculations by storing the numbers in registers.  
(Calculations like Add, Subtract, Multiply and Divide.
- Then, we should learn how a loop works and how delay works in assembly language and create a simple traffic light simulator program.
- We also should create a program which can display digits of a hexadecimal number in a 7 digit display.
- Using all that knowledge, we should finally create a program from scratch which does a simple multiplication and outputs the result in the 7 digit display.

# 1. Running Examples (Sub, Div & Mul)

```
; ===== 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.
SUB AL,BL          ; Add AL to BL. Answer goes into AL.
END               ; Program ends
; ===== Program Ends =====
```

```
; ===== 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.
DIV AL,BL          ; Add AL to BL. Answer goes into AL.
END               ; Program ends
; ===== Program Ends =====
```

YOUR TASK

=====

Use SUB, DIV and MUL to subtract, divide and multiply.

What happens if you divide by zero?

Make use of CL and DL as well as AL and BL.

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00	CLO	MOV	AL	2	MOV	BL	2	DIV	AL	BL	END	END	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

```
; ===== 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.
MUL AL,BL          ; Add AL to BL. Answer goes into AL.
END               ; Program ends
; ===== Program Ends =====
```

YOUR TASK

=====

Use SUB, DIV and MUL to subtract, divide and multiply.

What happens if you divide by zero?

Make use of CL and DL as well as AL and BL.

## 2. Modifying Existing Example

- 1. TRAFFIC LIGHT

```
;----- The Main Program -----
Start:
MOV     AL,0      ;Initially switch off all lights.
OUT     01        ;And display it.
MOV     BL,9      ;BL register is used for the 10s delay.
MOV     CL,4      ;CL register is used for the 4s delay.

MOV     AL,84     ; Red on in Left light, Green on in right light.
OUT     01        ; Send data to port 01.
CALL    30        ; Call the procedure at address [30] (10 second delay)

MOV     AL,48     ; Yellow on in Left light, Yellow on in right light.
OUT     01        ; Send data to port 01. No need to call a delay because it should change with 1 CPU cycle.

MOV     AL,30     ; Green on in Left light, Red on in right light.
OUT     01        ; Send data to port 01.
CALL    40        ; Call the procedure at address [40] (5 second delay)

JMP     Start     ; Jump back to the start.The traffic lights work in a loop and with the same procedure.

; ----- Time Delay Procedure Stored At Address [30] (10 second delay)-----
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.

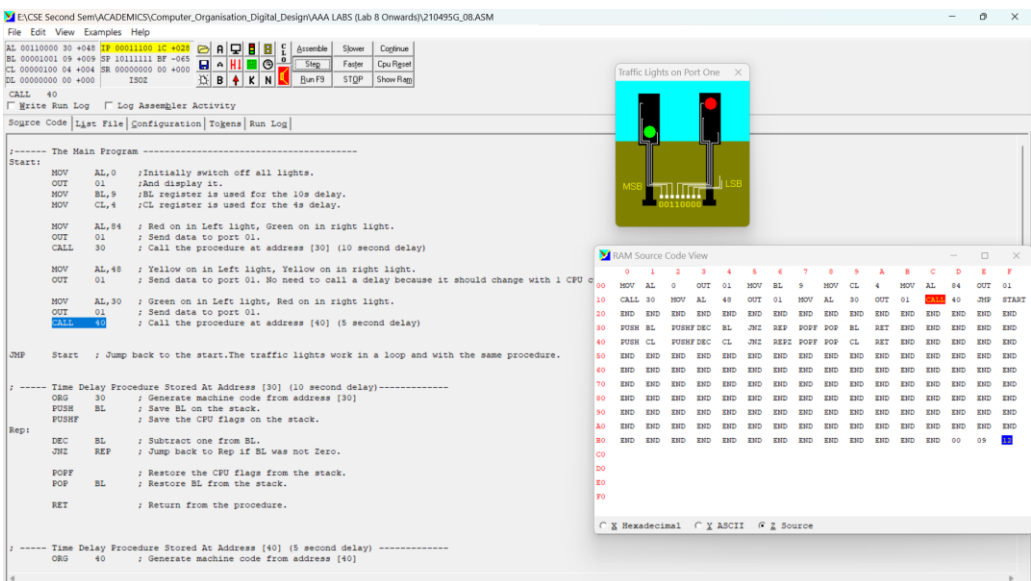
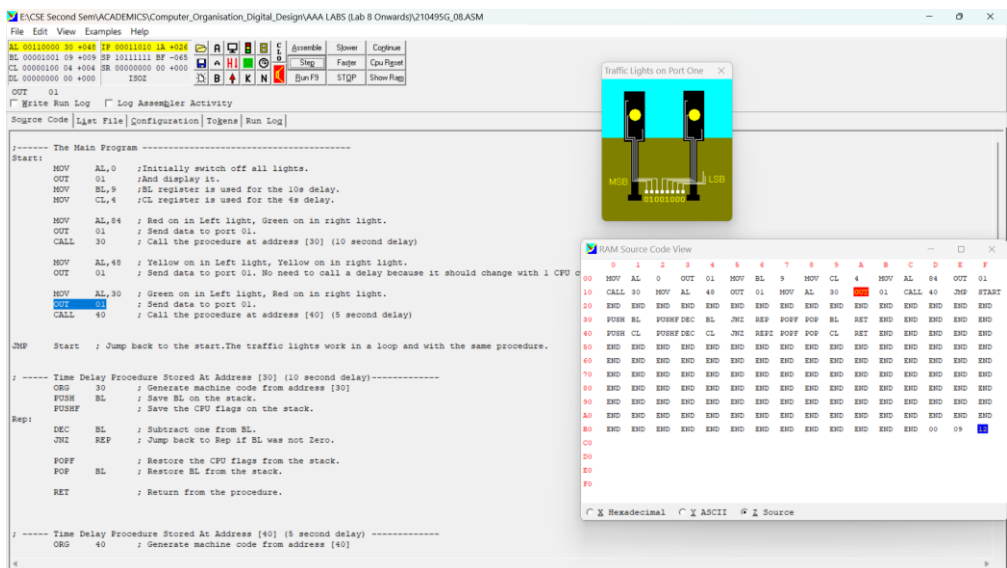
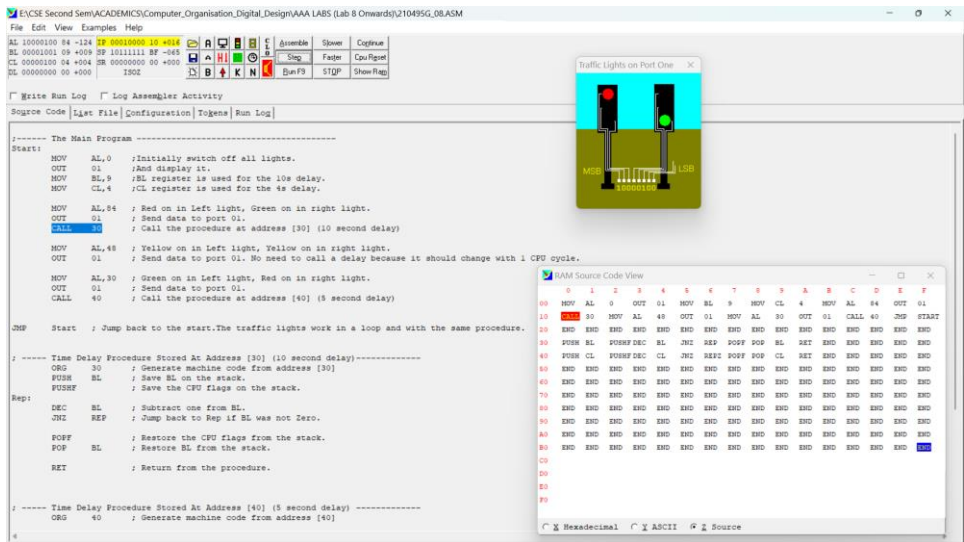
RET     ; Return from the procedure.

; ----- Time Delay Procedure Stored At Address [40] (5 second delay) -----
ORG     40        ; Generate machine code from address [40]
PUSH     CL       ; Save CL on the stack.
PUSHF    ; Save the CPU flags on the stack.
Repz:
DEC     CL        ; Subtract one from CL.
JNZ     REPZ      ; Jump back to Rep if CL was not Zero.

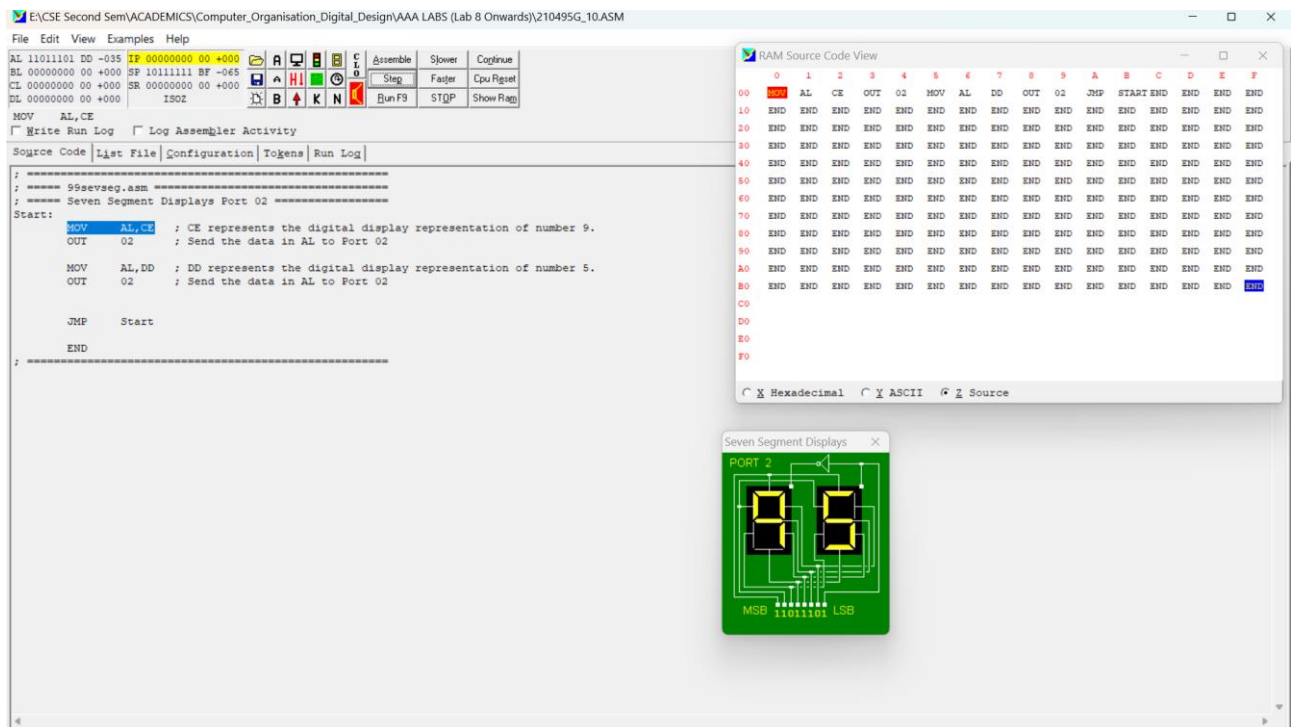
POPF    ; Restore the CPU flags from the stack.
POP     CL        ; Restore CL from the stack.

RET     ; Return from the procedure

END
```



## • 2. SHOWING LAST 2 DIGITS OF INDEX NUMBER



## 3. Creating a New Program

