NAME: PALLAVI PATIL

REG_NO: 211039001

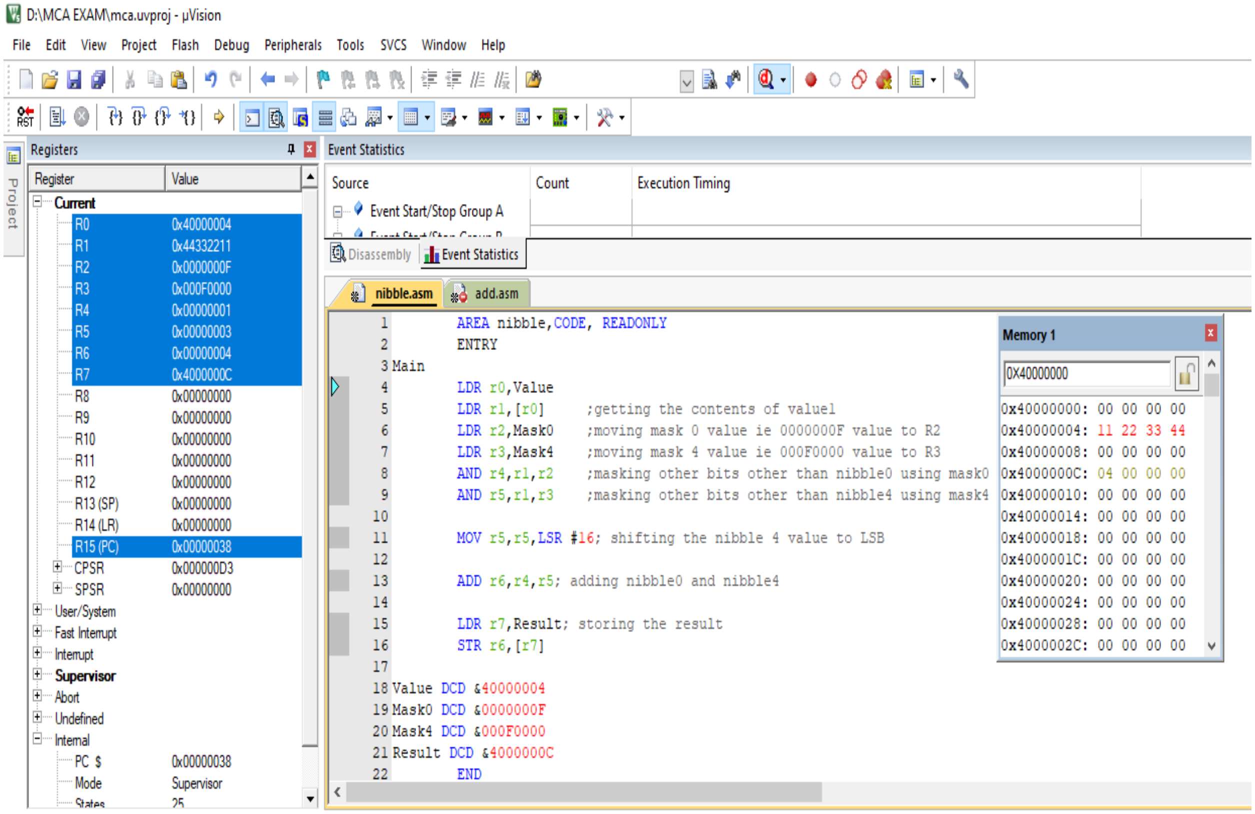
ACA LAB EXAM

1. **Assume a 32-bit number in 40000004H. Add nibble4 and nibble0 and store the result in 4000000CH.**

PROGRAM:

```
AREA nibble, CODE, READONLY
            ENTRY
Main
            LDR r0, Value
            LDR r1, [r0]            ; getting the contents of value1
            LDR r2, Mask0           ; moving mask 0 value i.e. 0000000F value to r2
            LDR r3, Mask4           ; moving mask 4 value i.e. 000F0000 value to r3
            AND r4, r1, r2          ; masking other bits other than nibble0 using mask0
            AND r5, r1, r3          ; masking other bits other than nibble4 using mask4

            MOV r5, r5, LSR #16     ; shifting the nibble 4 value to LSB

            ADD r6, r4, r5          ; adding nibble0 and nibble4

            LDR r7, Result          ; storing the result
            STR r6, [r7]

Value DCD &40000004
Mask0 DCD &0000000F
Mask4 DCD &000F0000
Result DCD &4000000C
            END
```

OUTPUT :



Register	Value
Current	
R0	0x40000004
R1	0x44332211
R2	0x0000000F
R3	0x000F0000
R4	0x00000001
R5	0x00000003
R6	0x00000004
R7	0x4000000C
R8	0x00000000
R9	0x00000000
R10	0x00000000
R11	0x00000000
R12	0x00000000
R13 (SP)	0x00000000
R14 (LR)	0x00000000
R15 (PC)	0x00000038
CPSR	0x000000D3
SPSR	0x00000000
User/System	
Fast Interrupt	
Interrupt	
Supervisor	
Abort	
Undefined	
Internal	
PC $	0x00000038
Mode	Supervisor
States	25

```
        AREA nibble,CODE, READONLY
        ENTRY
Main
        LDR r0,Value
        LDR r1,[r0]     ;getting the contents of value1
        LDR r2,Mask0    ;moving mask 0 value ie 0000000F value to R2
        LDR r3,Mask4    ;moving mask 4 value ie 000F0000 value to R3
        AND r4,r1,r2    ;masking other bits other than nibble0 using mask0
        AND r5,r1,r3    ;masking other bits other than nibble4 using mask4

        MOV r5,r5,LSR #16; shifting the nibble 4 value to LSB

        ADD r6,r4,r5; adding nibble0 and nibble4

        LDR r7,Result; storing the result
        STR r6,[r7]

Value DCD &40000004
Mask0 DCD &0000000F
Mask4 DCD &000F0000
Result DCD &4000000C
        END
```

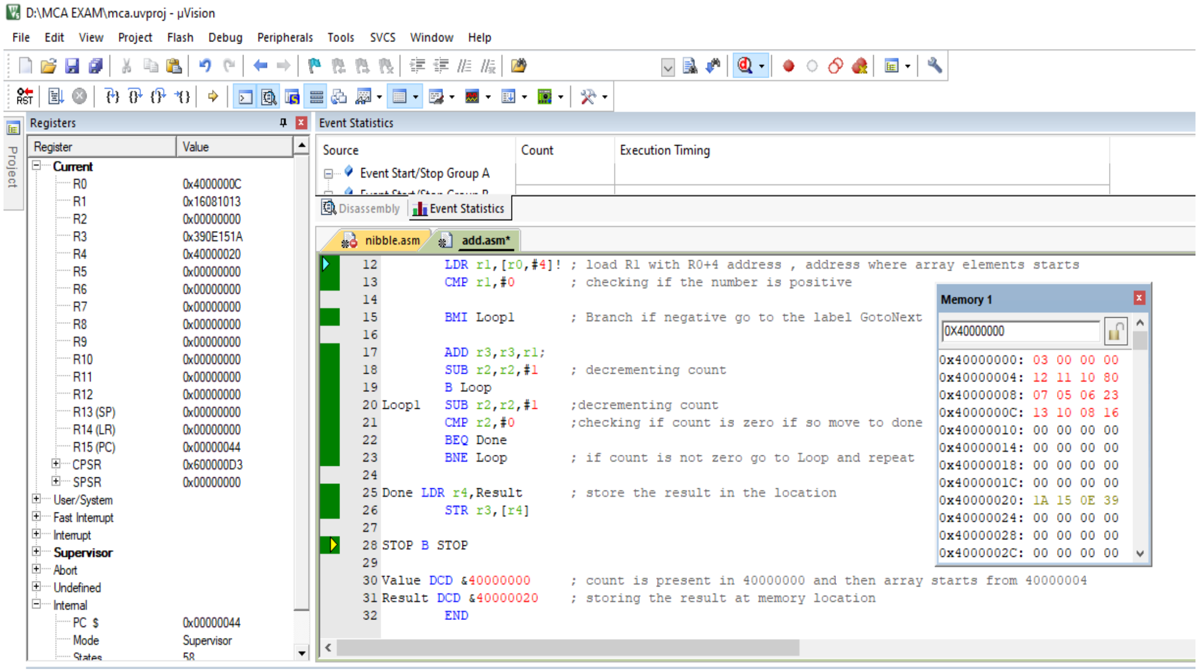
```
0X40000000
0x40000000: 00 00 00 00
0x40000004: 11 22 33 44
0x40000008: 00 00 00 00
0x4000000C: 04 00 00 00
0x40000010: 00 00 00 00
0x40000014: 00 00 00 00
0x40000018: 00 00 00 00
0x4000001C: 00 00 00 00
0x40000020: 00 00 00 00
0x40000024: 00 00 00 00
0x40000028: 00 00 00 00
0x4000002C: 00 00 00 00
```

2. Consider an array of number present from 40000000H. Add only if the numbers are positive. 40000000H has the count of the array.

PROGRAM:

```
                AREA add_positive, CODE, READONLY
                ENTRY
Main
                LDR r0, Value
                LDR r2, [r0]            ; count present at r0 is loaded to r2
                EOR r3, r3, r3          ; perform XOR to clearing the r3

Loop            CMP r2, #0              ; count is compared with 0
                BEQ Done                ; if equal loop done and store the result
                LDR r1, [r0, #4]!;load r1 with r0+4 address,address where array elements starts
                CMP r1, #0              ; checking if the number is positive
                BMI Loop1           ; Branch if negative go to the label Loop1
                ADD r3, r3, r1;
                SUB r2, r2, #1      ; decrementing count
                B Loop

Loop1           SUB r2, r2, #1      ; decrementing count
                CMP r2, #0          ; checking if count is zero if so move to done
                BEQ Done
                BNE Loop            ; if count is not zero go to Loop and repeat

Done            LDR r4, Result      ; store the result in the location
                STR r3, [r4]
STOP B STOP

Value DCD &40000000  ; count is present in 40000000 and then array starts from 40000004
Result DCD &4000003C                ; storing the result at memory location
                END
```

OUTPUT :



Register	Value
Current	
R0	0x4000000C
R1	0x16081013
R2	0x00000000
R3	0x390E151A
R4	0x40000020
R5	0x00000000
R6	0x00000000
R7	0x00000000
R8	0x00000000
R9	0x00000000
R10	0x00000000
R11	0x00000000
R12	0x00000000
R13 (SP)	0x00000000
R14 (LR)	0x00000000
R15 (PC)	0x00000044
CPSR	0x600000D3
SPSR	0x00000000
User/System	
Fast Interrupt	
Interrupt	
Supervisor	
Abort	
Undefined	
Internal	
PC $	0x00000044
Mode	Supervisor
States	58

```
        LDR r1,[r0,#4]! ; load R1 with R0+4 address , address where array elements starts
        CMP r1,#0       ; checking if the number is positive

        BMI Loop1       ; Branch if negative go to the label GotoNext

        ADD r3,r3,r1;
        SUB r2,r2,#1    ; decrementing count
        B Loop
Loop1   SUB r2,r2,#1    ;decrementing count
        CMP r2,#0       ;checking if count is zero if so move to done
        BEQ Done
        BNE Loop        ; if count is not zero go to Loop and repeat

Done LDR r4,Result      ; store the result in the location
        STR r3,[r4]

STOP B STOP

Value DCD &40000000     ; count is present in 40000000 and then array starts from 40000004
Result DCD &40000020    ; storing the result at memory location
        END
```

```
0X40000000
0x40000000: 03 00 00 00
0x40000004: 12 11 10 80
0x40000008: 07 05 06 23
0x4000000C: 13 10 08 16
0x40000010: 00 00 00 00
0x40000014: 00 00 00 00
0x40000018: 00 00 00 00
0x4000001C: 00 00 00 00
0x40000020: 1A 15 0E 39
0x40000024: 00 00 00 00
0x40000028: 00 00 00 00
0x4000002C: 00 00 00 00
```