Name : NIKHIL P SHESHAGIRI

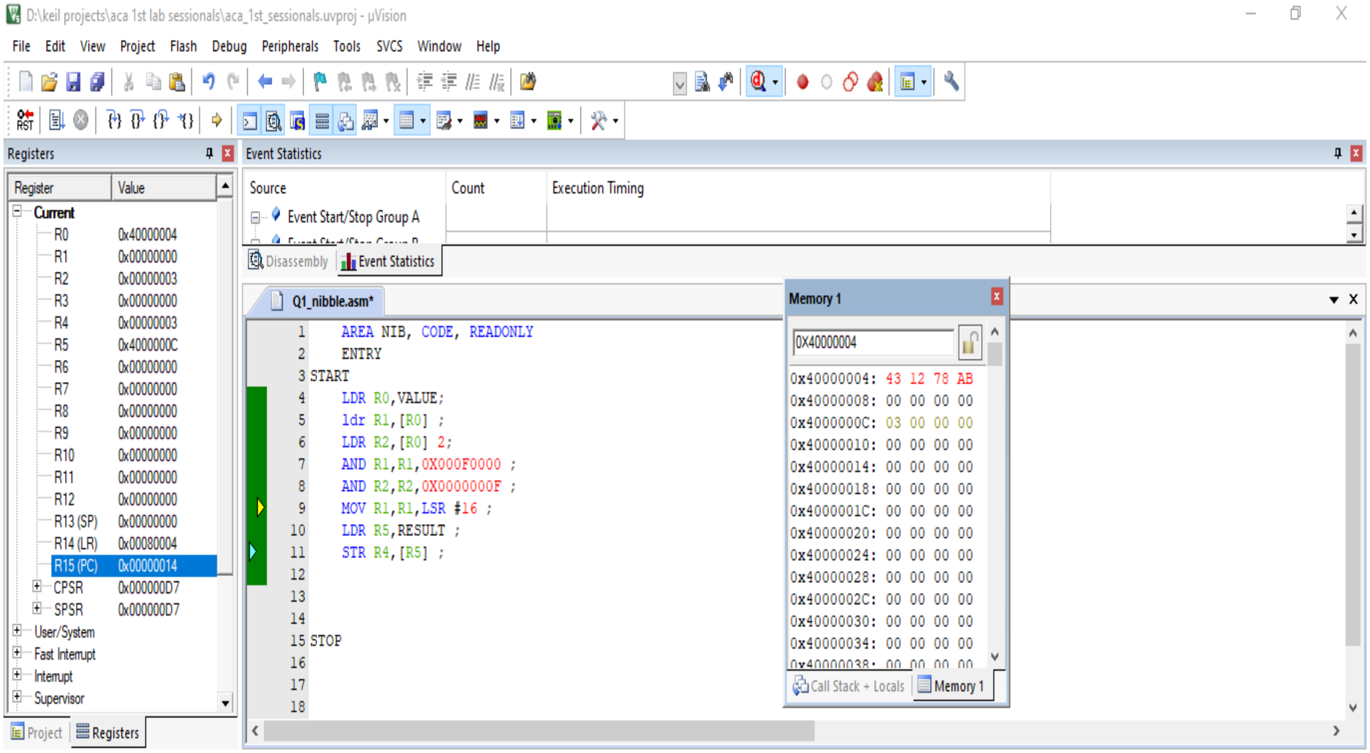
Reg_no : 211039015

;Q1) Assume a 32-bit number in 40000004H. Add nibble4 and nibble0 and store the result in 4000000CH.

Source Code :

```
        AREA NIB, CODE, READONLY
        ENTRY
START
        LDR R0,VALUE ;    loading the value address to reg r0
        ldr R1,[R0] ;           loading the content in ro to r1
        LDR R2,[R0] ;       loading the content in ro to r2
        AND R1,R1,0X000F0000 ; masking by anding and getting the nibble4
        AND R2,R2,0X0000000F ; masking by anding and getting the nibble0
        MOV R1,R1,LSR #16 ; shifting the n4 to lsb , to add with n0
        LDR R5,RESULT ; loading the address where result has to be stored into reg r5
        STR R4,[R5] ;   storing the final result into r5
STOP

VALUE DCD 0X40000004 ; address where we give the input values
RESULT DCD 0X4000000c ; address to check the final result
        END
```

Output :



```
Register      Value
Current
  R0          0x40000004
  R1          0x00000000
  R2          0x00000003
  R3          0x00000000
  R4          0x00000003
  R5          0x4000000C
  R6          0x00000000
  R7          0x00000000
  R8          0x00000000
  R9          0x00000000
  R10         0x00000000
  R11         0x00000000
  R12         0x00000000
  R13 (SP)    0x00000000
  R14 (LR)    0x00080004
  R15 (PC)    0x00000014
  CPSR        0x000000D7
  SPSR        0x000000D7
User/System
Fast Interrupt
Interrupt
Supervisor
```

```
    AREA NIB, CODE, READONLY
    ENTRY
START
    LDR R0,VALUE;
    ldr R1,[R0] ;
    LDR R2,[R0] 2;
    AND R1,R1,0X000F0000 ;
    AND R2,R2,0X0000000F ;
    MOV R1,R1,LSR #16 ;
    LDR R5,RESULT ;
    STR R4,[R5] ;



STOP
```

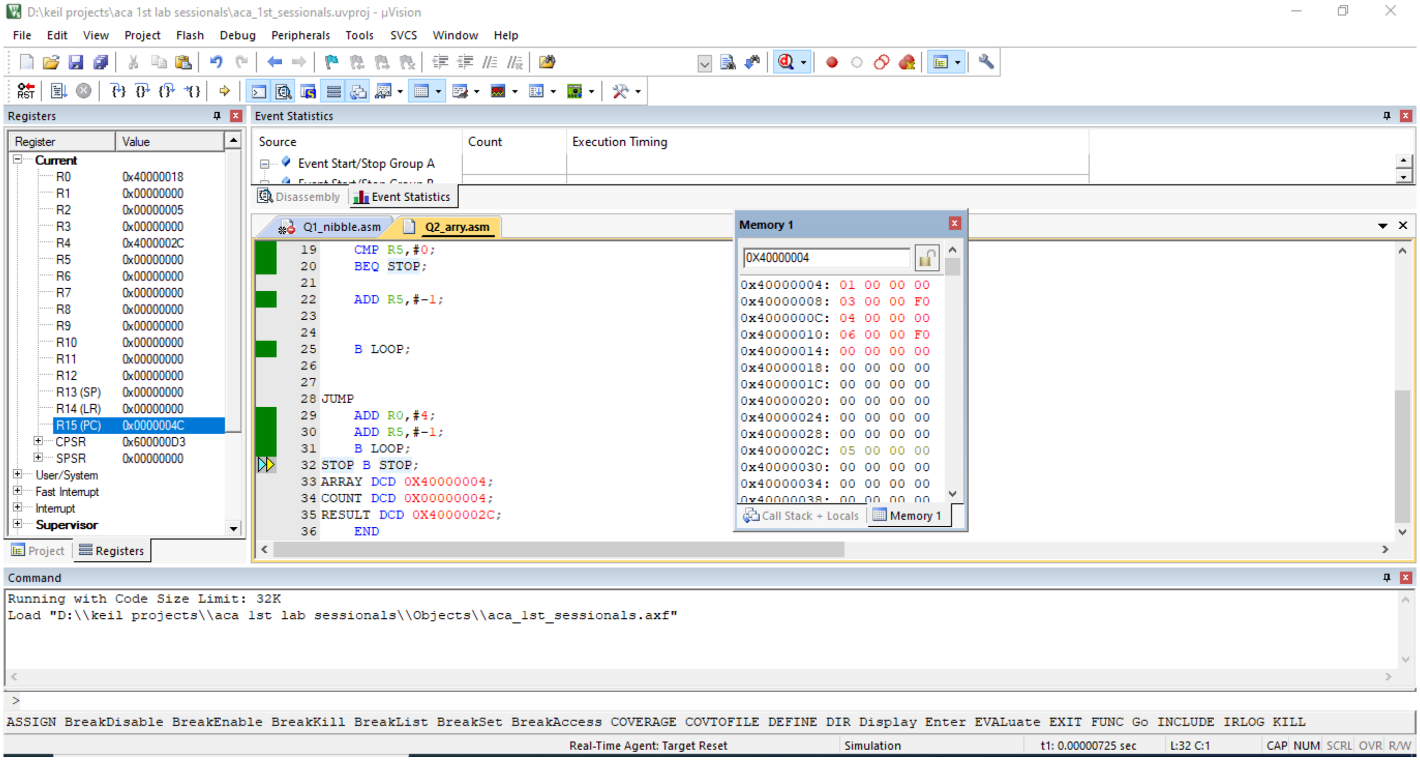
```
0X40000004
0x40000004: 43 12 78 AB
0x40000008: 00 00 00 00
0x4000000C: 03 00 00 00
0x40000010: 00 00 00 00
0x40000014: 00 00 00 00
0x40000018: 00 00 00 00
0x4000001C: 00 00 00 00
0x40000020: 00 00 00 00
0x40000024: 00 00 00 00
0x40000028: 00 00 00 00
0x4000002C: 00 00 00 00
0x40000030: 00 00 00 00
0x40000034: 00 00 00 00
0x40000038: 00 00 00 00
```

Q2) Consider an array of number present from 40000000 H. Add only if the numbers are positive. 40000000 H has the count of the array.

SOURCE CODE:

```
        AREA array, CODE, READONLY
        ENTRY
MAIN
        LDR R0,ARRAY ;  loading the address of 1st element of the array into r0
        LDR R5,COUNT ; intialising the r5 to count i,e  4
        LDR R1,[R0] ; loading the content of r0 to r1
        CMP R1,#0 ; comparing whether the first element is negative or not
        BMI JUMP ; if it's negative then go to jump position
LOOP

        LDR R1,[R0] ; loading the content of r0 to r1

        CMP R1,#0 ; comparing whether the element is negative or not
        BMI JUMP ; if it's negative then go to jump position

        ADD R0,#4 ; incrementing the address to next array location
        ADD R2,R1 ; taking r2 reg and adding the array element to it and using it to store further
addition values
        LDR R4,RESULT; loading the result address into r4
        STR R2,[R4] ; storing the final result into r4
        CMP R5,#0; checking whether the count has become zero
        BEQ STOP ; if its zero then stop the execution

        ADD R5,#-1; decrementing the count if its not zero
        B LOOP ; then go back loop to add the next array element


JUMP
        ADD R0,#4 ; increment to next position if the present array element is a negative
        ADD R5,#-1 ;  decrementing the count if
        B LOOP;
STOP B STOP;
ARRAY DCD 0X40000004 ;   address where we give the input values
COUNT DCD 0X00000004 ; initializing the count to 4,so that we can add 4 array elements
RESULT DCD 0X4000002C ; address to store the result
        END
```

OUTPUT:


D:\keil projects\aca 1st lab sessionals\aca_1st_sessionals.uvproj - µVision
File Edit View Project Flash Debug Peripherals Tools SVCS Window Help
Registers
Event Statistics
Disassembly
Event Statistics
Q1_nibble.asm
Q2_arry.asm
Memory 1
Call Stack + Locals
Memory 1
Project
Registers
Command

Register	Value
Current	
R0	0x40000018
R1	0x00000000
R2	0x00000005
R3	0x00000000
R4	0x4000002C
R5	0x00000000
R6	0x00000000
R7	0x00000000
R8	0x00000000
R9	0x00000000
R10	0x00000000
R11	0x00000000
R12	0x00000000
R13 (SP)	0x00000000
R14 (LR)	0x00000000
R15 (PC)	0x0000004C
CPSR	0x600000D3
SPSR	0x00000000
User/System	
Fast Interrupt	
Interrupt	
Supervisor	

Source	Count	Execution Timing
Event Start/Stop Group A		

```
    CMP R5,#0;
    BEQ STOP;

    ADD R5,#-1;


    B LOOP;


JUMP
    ADD R0,#4;
    ADD R5,#-1;
    B LOOP;
STOP B STOP;
ARRAY DCD 0X40000004;
COUNT DCD 0X00000004;
RESULT DCD 0X4000002C;
    END
```

0X40000004

```
0x40000004: 01 00 00 00
0x40000008: 03 00 00 F0
0x4000000C: 04 00 00 00
0x40000010: 06 00 00 F0
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: 05 00 00 00
0x40000030: 00 00 00 00
0x40000034: 00 00 00 00
0x40000038: 00 00 00 00
```

```
Running with Code Size Limit: 32K
Load "D:\\keil projects\\aca 1st lab sessionals\\Objects\\aca_1st_sessionals.axf"
```

```
>
ASSIGN BreakDisable BreakEnable BreakKill BreakList BreakSet BreakAccess COVERAGE COVTOFILE DEFINE DIR Display Enter EVALuate EXIT FUNC Go INCLUDE IRLOG KILL
```

Real-Time Agent: Target Reset   Simulation   t1: 0.00000725 sec   L:32 C:1   CAP NUM SCRL OVR R/W