

## EE2016 LAB EXPERIMENT 3 – ARM BASICS

### EE20B056

#### TASK 1 – FACTORIAL:

Compute the factorial of a given number using ARM processor through assembly programming

C:\Users\kathi\Documents\EE2016LAB3\EE20B056\_LAB3.uvproj - uVision

File Edit View Project Flash Debug Peripherals Tools SVCS Window Help

Registers

Register	Value
R0	0x00000000
R1	0x00000078
R2	0x00000001
R3	0x00000000
R4	0x00000000
R5	0x00000000
R6	0x00000000
R7	0x00000000
R8	0x00000000
R9	0x00000000
R10	0x00000000
R11	0x00000000
R12	0x00000000
R13 (SP)	0x00000000
R14 (LR)	0x00000000
R15 (PC)	0x00000028
CPSR	0x600000D3
SPSR	0x00000000
User/System	
Fast Interrupt	
Interrupt	
Supervisor	
Abort	
Undefined	
Internal	
PC \$	0x00000028
Mode	Supervisor

Disassembly

```
13:      MOV     R2,#1
0x00000024 E3A02001 MOV     R2,#0x00000001
14:      b      one
0x00000028 EAF0FFFD B       0x00000024
```

task1.s

```
1      AREA FACTORIAL, CODE, READONLY
2      LDR R0, NUM
3      LDR R1, NUM
4      fact
5      SUBS R0, R0, #0
6      BEQ one
7      SUBS R0, R0, #1
8      BEQ one
9      MUL R2, R1, R0
10     MOV R1, R2
11     B fact
12     one
13     MOV R2, #1
14     b one
15
16     NUM DCW &5
17     END
```

Command

```
Load "C:\\Users\\kathi\\Documents\\EE2016LAB3\\Objects\\EE20B056_LAB3.axf"

*** Restricted Version with 32768 Byte Code Size Limit
*** Currently used: 48 Bytes (0%)
```

Memory 1

Address	Value
0x40000000	00 00 00 00 00 00 00 00
0x40000001	00 00 00 00 00 00 00 00
0x40000002	00 00 00 00 00 00 00 00
0x40000003	00 00 00 00 00 00 00 00
0x40000004	00 00 00 00 00 00 00 00
0x40000005	00 00 00 00 00 00 00 00
0x40000006	00 00 00 00 00 00 00 00
0x40000007	00 00 00 00 00 00 00 00
0x40000008	00 00 00 00 00 00 00 00
0x40000009	00 00 00 00 00 00 00 00
0x4000000A	00 00 00 00 00 00 00 00
0x4000000B	00 00 00 00 00 00 00 00
0x4000000C	00 00 00 00 00 00 00 00
0x4000000D	00 00 00 00 00 00 00 00
0x4000000E	00 00 00 00 00 00 00 00
0x4000000F	00 00 00 00 00 00 00 00
0x40000010	00 00 00 00 00 00 00 00
0x40000011	00 00 00 00 00 00 00 00
0x40000012	00 00 00 00 00 00 00 00
0x40000013	00 00 00 00 00 00 00 00
0x40000014	00 00 00 00 00 00 00 00
0x40000015	00 00 00 00 00 00 00 00
0x40000016	00 00 00 00 00 00 00 00
0x40000017	00 00 00 00 00 00 00 00
0x40000018	00 00 00 00 00 00 00 00
0x40000019	00 00 00 00 00 00 00 00
0x4000001A	00 00 00 00 00 00 00 00
0x4000001B	00 00 00 00 00 00 00 00
0x4000001C	00 00 00 00 00 00 00 00
0x4000001D	00 00 00 00 00 00 00 00
0x4000001E	00 00 00 00 00 00 00 00
0x4000001F	00 00 00 00 00 00 00 00
0x40000020	00 00 00 00 00 00 00 00
0x40000021	00 00 00 00 00 00 00 00
0x40000022	00 00 00 00 00 00 00 00
0x40000023	00 00 00 00 00 00 00 00
0x40000024	00 00 00 00 00 00 00 00
0x40000025	00 00 00 00 00 00 00 00
0x40000026	00 00 00 00 00 00 00 00
0x40000027	00 00 00 00 00 00 00 00
0x40000028	00 00 00 00 00 00 00 00
0x40000029	00 00 00 00 00 00 00 00
0x4000002A	00 00 00 00 00 00 00 00
0x4000002B	00 00 00 00 00 00 00 00
0x4000002C	00 00 00 00 00 00 00 00
0x4000002D	00 00 00 00 00 00 00 00
0x4000002E	00 00 00 00 00 00 00 00
0x4000002F	00 00 00 00 00 00 00 00
0x40000030	00 00 00 00 00 00 00 00
0x40000031	00 00 00 00 00 00 00 00
0x40000032	00 00 00 00 00 00 00 00
0x40000033	00 00 00 00 00 00 00 00
0x40000034	00 00 00 00 00 00 00 00
0x40000035	00 00 00 00 00 00 00 00
0x40000036	00 00 00 00 00 00 00 00
0x40000037	00 00 00 00 00 00 00 00
0x40000038	00 00 00 00 00 00 00 00
0x40000039	00 00 00 00 00 00 00 00
0x4000003A	00 00 00 00 00 00 00 00
0x4000003B	00 00 00 00 00 00 00 00
0x4000003C	00 00 00 00 00 00 00 00
0x4000003D	00 00 00 00 00 00 00 00
0x4000003E	00 00 00 00 00 00 00 00
0x4000003F	00 00 00 00 00 00 00 00
0x40000040	00 00 00 00 00 00 00 00
0x40000041	00 00 00 00 00 00 00 00
0x40000042	00 00 00 00 00 00 00 00
0x40000043	00 00 00 00 00 00 00 00
0x40000044	00 00 00 00 00 00 00 00
0x40000045	00 00 00 00 00 00 00 00
0x40000046	00 00 00 00 00 00 00 00
0x40000047	00 00 00 00 00 00 00 00
0x40000048	00 00 00 00 00 00 00 00
0x40000049	00 00 00 00 00 00 00 00
0x4000004A	00 00 00 00 00 00 00 00
0x4000004B	00 00 00 00 00 00 00 00
0x4000004C	00 00 00 00 00 00 00 00
0x4000004D	00 00 00 00 00 00 00 00
0x4000004E	00 00 00 00 00 00 00 00
0x4000004F	00 00 00 00 00 00 00 00
0x40000050	00 00 00 00 00 00 00 00
0x40000051	00 00 00 00 00 00 00 00
0x40000052	00 00 00 00 00 00 00 00
0x40000053	00 00 00 00 00 00 00 00
0x40000054	00 00 00 00 00 00 00 00
0x40000055	00 00 00 00 00 00 00 00
0x40000056	00 00 00 00 00 00 00 00
0x40000057	00 00 00 00 00 00 00 00
0x40000058	00 00 00 00 00 00 00 00
0x40000059	00 00 00 00 00 00 00 00
0x4000005A	00 00 00 00 00 00 00 00
0x4000005B	00 00 00 00 00 00 00 00
0x4000005C	00 00 00 00 00 00 00 00
0x4000005D	00 00 00 00 00 00 00 00
0x4000005E	00 00 00 00 00 00 00 00
0x4000005F	00 00 00 00 00 00 00 00
0x40000060	00 00 00 00 00 00 00 00
0x40000061	00 00 00 00 00 00 00 00
0x40000062	00 00 00 00 00 00 00 00
0x40000063	00 00 00 00 00 00 00 00
0x40000064	00 00 00 00 00 00 00 00
0x40000065	00 00 00 00 00 00 00 00
0x40000066	00 00 00 00 00 00 00 00
0x40000067	00 00 00 00 00 00 00 00
0x40000068	00 00 00 00 00 00 00 00
0x40000069	00 00 00 00 00 00 00 00
0x4000006A	00 00 00 00 00 00 00 00
0x4000006B	00 00 00 00 00 00 00 00
0x4000006C	00 00 00 00 00 00 00 00
0x4000006D	00 00 00 00 00 00 00 00
0x4000006E	00 00 00 00 00 00 00 00
0x4000006F	00 00 00 00 00 00 00 00
0x40000070	00 00 00 00 00 00 00 00
0x40000071	00 00 00 00 00 00 00 00
0x40000072	00 00 00 00 00 00 00 00
0x40000073	00 00 00 00 00 00 00 00
0x40000074	00 00 00 00 00 00 00 00
0x40000075	00 00 00 00 00 00 00 00
0x40000076	00 00 00 00 00 00 00 00
0x40000077	00 00 00 00 00 00 00 00
0x40000078	00 00 00 00 00 00 00 00
0x40000079	00 00 00 00 00 00 00 00
0x4000007A	00 00 00 00 00 00 00 00
0x4000007B	00 00 00 00 00 00 00 00
0x4000007C	00 00 00 00 00 00 00 00
0x4000007D	00 00 00 00 00 00 00 00
0x4000007E	00 00 00 00 00 00 00 00
0x4000007F	00 00 00 00 00 00 00 00
0x40000080	00 00 00 00 00 00 00 00
0x40000081	00 00 00 00 00 00 00 00
0x40000082	00 00 00 00 00 00 00 00
0x40000083	00 00 00 00 00 00 00 00
0x40000084	00 00 00 00 00 00 00 00
0x40000085	00 00 00 00 00 00 00 00
0x40000086	00 00 00 00 00 00 00 00
0x40000087	00 00 00 00 00 00 00 00
0x40000088	00 00 00 00 00 00 00 00
0x40000089	00 00 00 00 00 00 00 00
0x4000008A	00 00 00 00 00 00 00 00
0x4000008B	00 00 00 00 00 00 00 00
0x4000008C	00 00 00 00 00 00 00 00
0x4000008D	00 00 00 00 00 00 00 00
0x4000008E	00 00 00 00 00 00 00 00
0x4000008F	00 00 00 00 00 00 00 00
0x40000090	00 00 00 00 00 00 00 00
0x40000091	00 00 00 00 00 00 00 00
0x40000092	00 00 00 00 00 00 00 00
0x40000093	00 00 00 00 00 00 00 00
0x40000094	00 00 00 00 00 00 00 00
0x40000095	00 00 00 00 00 00 00 00
0x40000096	00 00 00 00 00 00 00 00
0x40000097	00 00 00 00 00 00 00 00
0x40000098	00 00 00 00 00 00 00 00
0x40000099	00 00 00 00 00 00 00 00
0x4000009A	00 00 00 00 00 00 00 00
0x4000009B	00 00 00 00 00 00 00 00
0x4000009C	00 00 00 00 00 00 00 00
0x4000009D	00 00 00 00 00 00 00 00
0x4000009E	00 00 00 00 00 00 00 00
0x4000009F	00 00 00 00 00 00 00 00
0x400000A0	00 00 00 00 00 00 00 00
0x400000A1	00 00 00 00 00 00 00 00
0x400000A2	00 00 00 00 00 00 00 00
0x400000A3	00 00 00 00 00 00 00 00
0x400000A4	00 00 00 00 00 00 00 00
0x400000A5	00 00 00 00 00 00 00 00
0x400000A6	00 00 00 00 00 00 00 00
0x400000A7	00 00 00 00 00 00 00 00
0x400000A8	00 00 00 00 00 00 00 00
0x400000A9	00 00 00 00 00 00 00 00
0x400000AA	00 00 00 00 00 00 00 00
0x400000AB	00 00 00 00 00 00 00 00
0x400000AC	00 00 00 00 00 00 00 00
0x400000AD	00 00 00 00 00 00 00 00
0x400000AE	00 00 00 00 00 00 00 00
0x400000AF	00 00 00 00 00 00 00 00
0x400000B0	00 00 00 00 00 00 00 00
0x400000B1	00 00 00 00 00 00 00 00
0x400000B2	00 00 00 00 00 00 00 00
0x400000B3	00 00 00 00 00 00 00 00
0x400000B4	00 00 00 00 00 00 00 00
0x400000B5	00 00 00 00 00 00 00 00
0x400000B6	00 00 00 00 00 00 00 00
0x400000B7	00 00 00 00 00 00 00 00
0x400000B8	00 00 00 00 00 00 00 00
0x400000B9	00 00 00 00 00 00 00 00
0x400000BA	00 00 00 00 00 00 00 00
0x400000BB	00 00 00 00 00 00 00 00
0x400000BC	00 00 00 00 00 00 00 00
0x400000BD	00 00 00 00 00 00 00 00
0x400000BE	00 00 00 00 00 00 00 00
0x400000BF	00 00 00 00 00 00 00 00
0x400000C0	00 00 00 00 00 00 00 00
0x400000C1	00 00 00 00 00 00 00 00
0x400000C2	00 00 00 00 00 00 00 00
0x400000C3	00 00 00 00 00 00 00 00
0x400000C4	00 00 00 00 00 00 00 00
0x400000C5	00 00 00 00 00 00 00 00
0x400000C6	00 00 00 00 00 00 00 00
0x400000C7	00 00 00 00 00 00 00 00
0x400000C8	00 00 00 00 00 00 00 00
0x400000C9	00 00 00 00 00 00 00 00
0x400000CA	00 00 00 00 00 00 00 00
0x400000CB	00 00 00 00 00 00 00 00
0x400000CC	00 00 00 00 00 00 00 00
0x400000CD	00 00 00 00 00 00 00 00
0x400000CE	00 00 00 00 00 00 00 00
0x400000CF	00 00 00 00 00 00 00 00
0x400000D0	00 00 00 00 00 00 00 00
0x400000D1	00 00 00 00 00 00 00 00
0x400000D2	00 00 00 00 00 00 00 00
0x400000D3	00 00 00 00 00 00 00 00
0x400000D4	00 00 00 00 00 00 00 00
0x400000D5	00 00 00 00 00 00 00 00
0x400000D6	00 00 00 00 00 00 00 00
0x400000D7	00 00 00 00 00 00 00 00
0x400000D8	00 00 00 00 00 00 00 00
0x400000D9	00 00 00 00 00 00 00 00
0x400000DA	00 00 00 00 00 00 00 00
0x400000DB	00 00 00 00 00 00 00 00
0x400000DC	00 00 00 00 00 00 00 00
0x400000DD	00 00 00 00 00 00 00 00
0x400000DE	00 00 00 00 00 00 00 00
0x400000DF	00 00 00 00 00 00 00 00
0x400000E0	00 00 00 00 00 00 00 00
0x400000E1	00 00 00 00 00 00 00 00
0x400000E2	00 00 00 00 00 00 00 00
0x400000E3	00 00 00 00 00 00 00 00
0x400000E4	00 00 00 00 00 00 00 00
0x400000E5	00 00 00 00 00 00 00 00
0x400000E6	00 00 00 00 00 00 00 00
0x400000E7	00 00 00 00 00 00 00 00
0x400000E8	00 00 00 00 00 00 00 00
0x400000E9	00 00 00 00 00 00 00 00
0x400000EA	00 00 0

**CODE:**

```
        AREA FACTORIAL, CODE, READONLY

        LDR    R0, NUM
        LDR    R1, NUM

fact
        SUBS   R0, R0, #0
        BEQ    one
        SUBS   R0, R0, #1
        BEQ    one
        MUL    R2, R1, R0
        MOV    R1, R2
        B fact

one
        MOV    R2, #1
b       one

NUM     DCW    &5

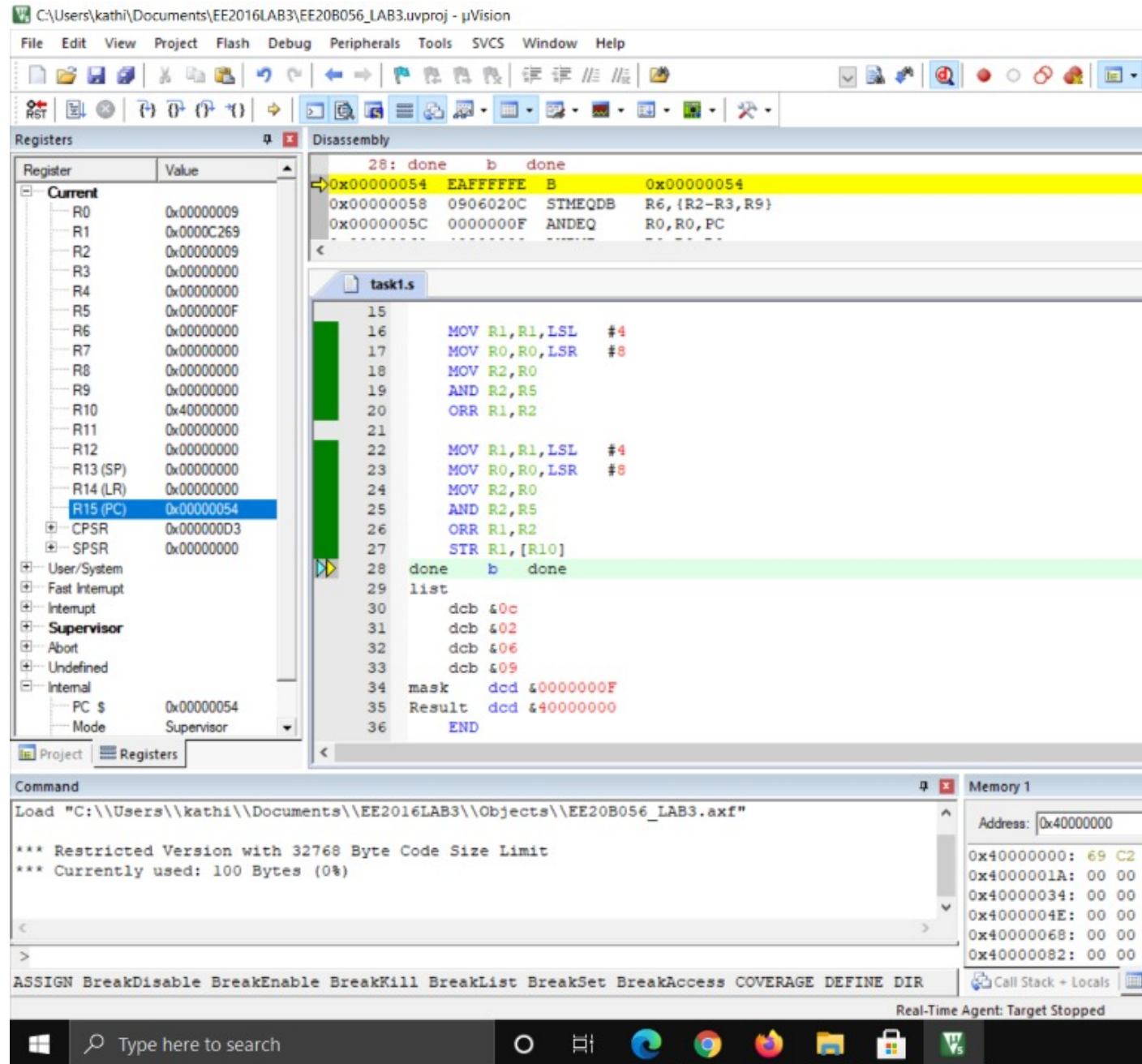
        END
```

**LOGIC:** Given number is put in R0 and R1, firstly we check if given number is 1 if true, we jump to 'one' label meaning the factorial is 1, if given number is not 1, we subtract the value in R0 by 1 and multiply with the value in R1 and put it in R2, and we copy this value back to R1 and proceed the same way. For a given value the process is shown below

[R0,R1,R2] : [5,-,-] → [5,5,-] → [4,5,-] → [4,5,20] → [4,20,20] → [3,20,20] → [3,20,60] → [3,60,60] → [2,60,60] → [2,60,120] → [2,120,120] → [1,120,120] → [1,120,1]

## TASK 2 – 16-bit HALFWORD:

Combine the low four bits of each of the four consecutive bytes beginning at LIST into one 16-bit halfword. The value at LIST goes into the most significant nibble of the result. Store the result in the 32-bit variable RESULT.



## CODE:

```
AREA Program ,CODE,READONLY
```

```
LDR R0, list ; four consecutive bytes as defined below
```

```
LDR R5, mask ; 4 LSB bits alone are kept as 1s to be used as mask
```

LDR R10, Result ; the address being labelled as result is copied to this register

MOV R1,R0

AND R1,R5 ; the first 4 lsb bits alone are kept

MOV R1,R1,LSL #4 ; the four bits are shifted to left,

MOV R0,R0,LSR #8 ; eight bits here are shifted to the right so that the MSB half byte is removed

MOV R2,R0

AND R2,R5 – mask the other bits than the lower half byte of the next byte alone

ORR R1,R2

MOV R1,R1,LSL #4

MOV R0,R0,LSR #8

MOV R2,R0

AND R2,R5

ORR R1,R2

MOV R1,R1,LSL #4

MOV R0,R0,LSR #8

MOV R2,R0

AND R2,R5

ORR R1,R2

STR R1,[R10]

done b done

list

dcb &0c

dcb &02

dcb &06

dcb &09

mask dcd &0000000F

```
Result dcd    &40000000
```

```
END
```

**LOGIC:**

R0 – LIST – 4 bytes sequenced as a single 32 bit

R1 – output - LSBs of the given 4 bytes in list

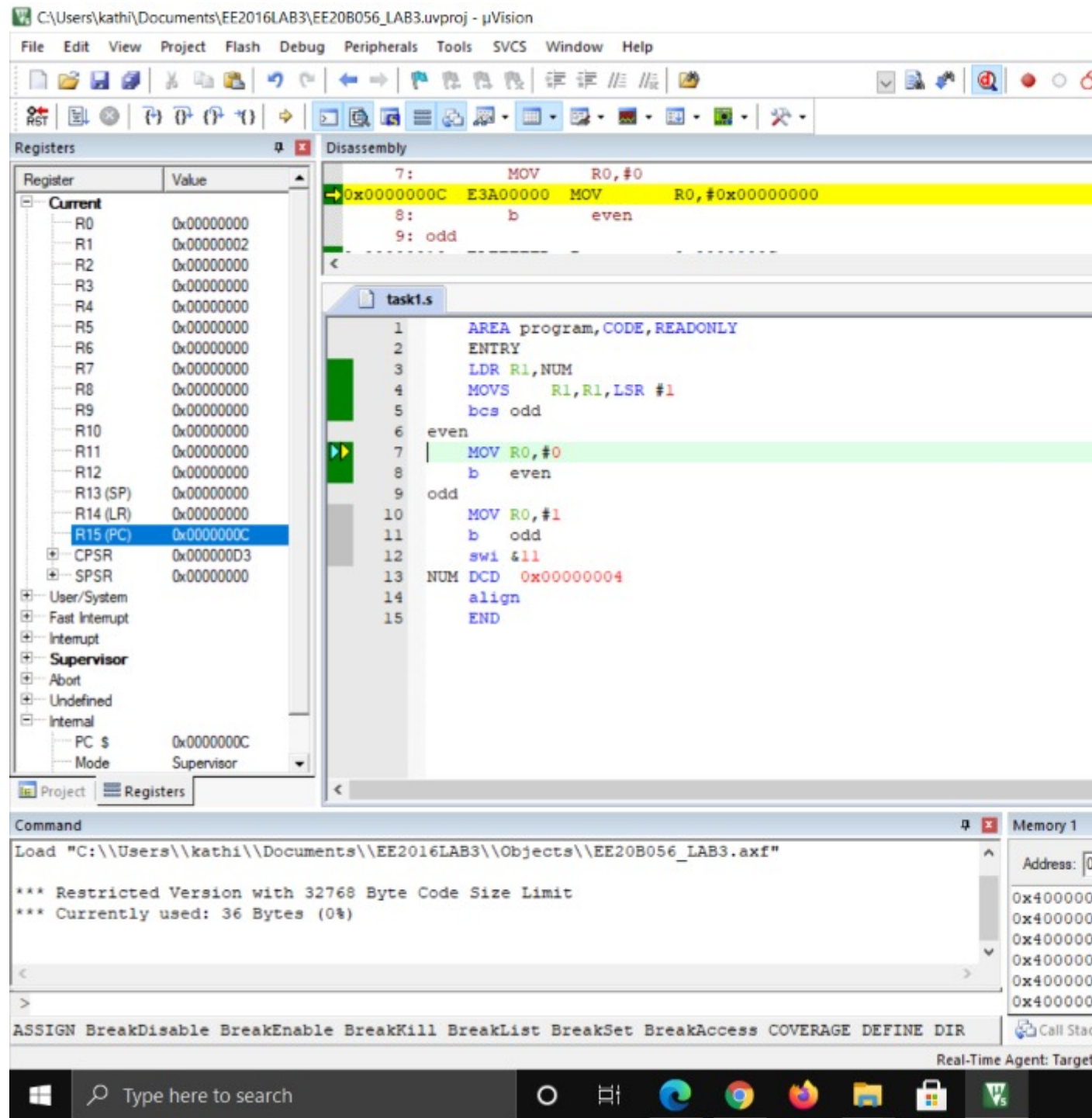
R2 – used to take the lower byte one by one by masking with R0 with R5

R5 – 0x0000000F – mask, when masked with this produces 4 bit LSBs alone

The above registers are used as mentioned in the comments in the code

### TASK 3 – ODD OR EVEN:

Given a 32-bit number, identify whether it is an even or odd. (Your implementation should not involve division).



### CODE:

```
AREA program, CODE, READONLY
```

```
ENTRY
```

```
LDR R1, NUM
```

```

        MOVS  R1,R1,LSR #1
        bcs  odd
even:
        MOV   R0,#0
        b     even
odd:
        MOV   R0,#1
        b     odd
        swi   &11
NUM DCD 0x00000004
        align
        END

```

**LOGIC:**

Given number is taken in R1

R1 is logically shifted to right , if it is odd carry is set, if it is even carry is not set – in both cases it jumps to either of the loops corresponding to the case.