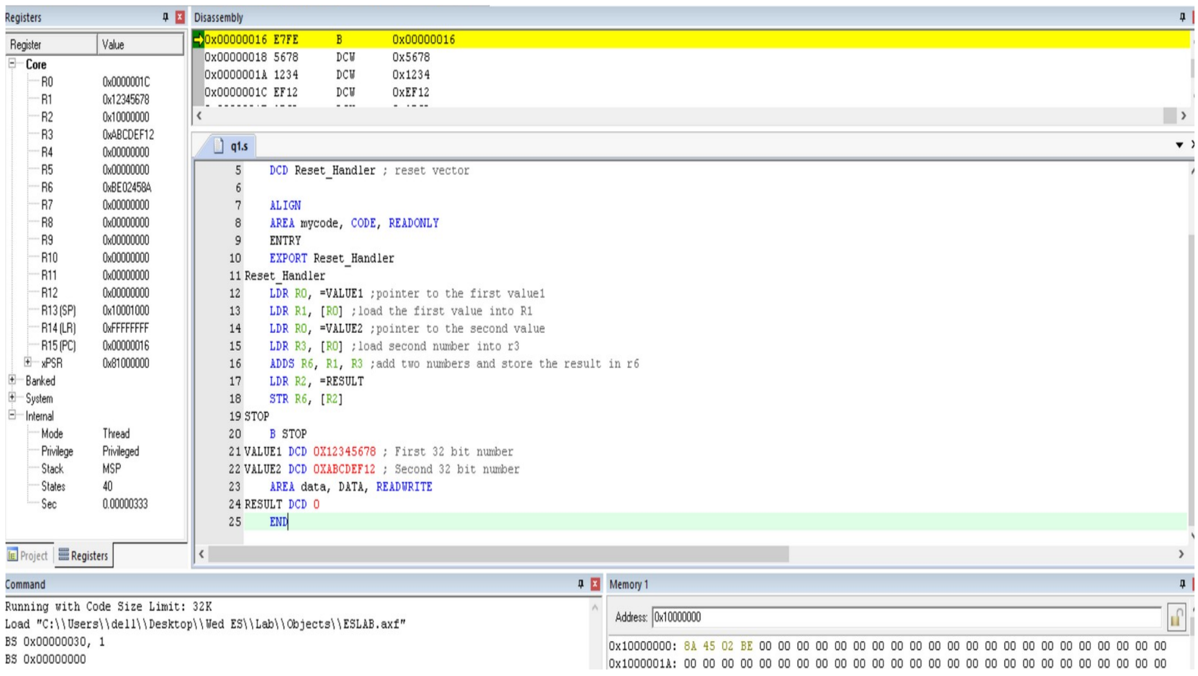
WEEK2 LAB2 :

Solved Exercise :

Write a program to add two 32-bit numbers available in the code memory. Store the result in the data memory.

```
AREA RESET, DATA, READONLY
EXPORT __Vectors
__Vectors
DCD 0x10001000 ; stack pointer value when stack is empty
DCD Reset_Handler ; reset vector
ALIGN
AREA mycode, CODE, READONLY
ENTRY
EXPORT Reset_Handler
Reset_Handler
LDR R0, =VALUE1 ;pointer to the first value1
LDR R1, [R0] ;load the first value into R1
LDR R0, =VALUE2 ;pointer to the second value
LDR R3, [R0] ;load second number into r3
ADDS R6, R1, R3 ;add two numbers and store the result in r6
LDR R2, =RESULT
STR R6, [R2]
STOP
B STOP
VALUE1 DCD 0X12345678 ; First 32 bit number
VALUE2 DCD 0XABCDEF12 ; Second 32 bit number
AREA data, DATA, READWRITE
RESULT DCD 0
END
```

OUTPUT :



```
Register        Value
Core
  R0            0x0000001C
  R1            0x12345678
  R2            0x10000000
  R3            0xABCDEF12
  R4            0x00000000
  R5            0x00000000
  R6            0xBE02458A
  R7            0x00000000
  R8            0x00000000
  R9            0x00000000
  R10           0x00000000
  R11           0x00000000
  R12           0x00000000
  R13 (SP)      0x10001000
  R14 (LR)      0xFFFFFFFF
  R15 (PC)      0x00000016
  xPSR          0x81000000
Banked
System
Internal
  Mode          Thread
  Privilege     Privileged
  Stack         MSP
  States        40
  Sec           0.00000333
```

```
0x00000016 E7FE      B        0x00000016
0x00000018 5678      DCW      0x5678
0x0000001A 1234      DCW      0x1234
0x0000001C EF12      DCW      0xEF12
```

q1.s

```
    DCD Reset_Handler ; reset vector

    ALIGN
    AREA mycode, CODE, READONLY
    ENTRY
    EXPORT Reset_Handler
Reset_Handler
    LDR R0, =VALUE1 ;pointer to the first value1
    LDR R1, [R0] ;load the first value into R1
    LDR R0, =VALUE2 ;pointer to the second value
    LDR R3, [R0] ;load second number into r3
    ADDS R6, R1, R3 ;add two numbers and store the result in r6
    LDR R2, =RESULT
    STR R6, [R2]
STOP
    B STOP
VALUE1 DCD 0X12345678 ; First 32 bit number
VALUE2 DCD 0XABCDEF12 ; Second 32 bit number
    AREA data, DATA, READWRITE
RESULT DCD 0
    END
```

```
Running with Code Size Limit: 32K
Load "C:\\Users\\dell\\Desktop\\Wed ES\\Lab\\Objects\\ESLAB.axf"
BS 0x00000030, 1
BS 0x00000000
```

```
Address: 0x10000000
0x10000000: 8A 45 02 BE 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x1000001A: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
```

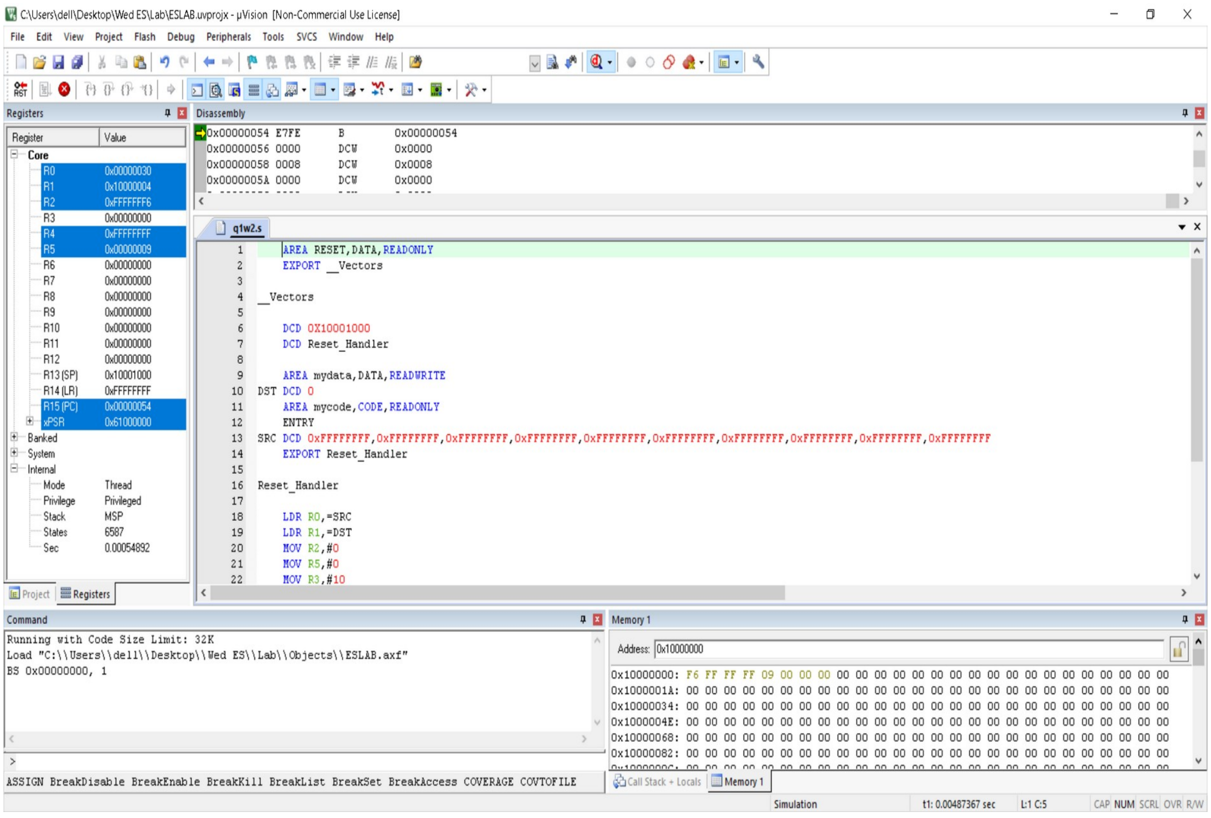
<u>Lab Exercises:</u>

1. Write a program to add ten 32-bit numbers available in code memory and store the result in data memory.

```
AREA RESET,DATA,READONLY
EXPORT __Vectors

__Vectors

DCD 0X10001000
DCD Reset_Handler
AREA mydata,DATA,READWRITE
DST DCD 0
AREA mycode,CODE,READONLY
ENTRY
SRC DCD 0xFFFFFFFF,0xFFFFFFFF,0xFFFFFFFF,0xFFFFFFFF,0xFFFFFFFF,
0xFFFFFFFF, 0xFFFFFFFF,0xFFFFFFFF,0xFFFFFFFF,0xFFFFFFFF
EXPORT Reset_Handler
Reset_Handler

LDR R0,=SRC
LDR R1,=DST
```

```
MOV R2,#0
MOV R5,#0
MOV R3,#10
UP LDR R4,[R0],#4
ADDS R2,R4
ADC R5,#0
SUBS R3,#1
BNE UP
STR R2,[R1],#4
STR R5,[R1]
STOP B STOP
END
```

OUTPUT :



C:\Users\dell\Desktop\Wed ES\Lab\ESLAB.uvprojx - µVision [Non-Commercial Use License]

Registers

Register	Value
Core	
R0	0x00000030
R1	0x10000004
R2	0xFFFFFFF6
R3	0x00000000
R4	0xFFFFFFFF
R5	0x00000009
R6	0x00000000
R7	0x00000000
R8	0x00000000
R9	0x00000000
R10	0x00000000
R11	0x00000000
R12	0x00000000
R13 (SP)	0x10001000
R14 (LR)	0xFFFFFFFF
R15 (PC)	0x00000054
xPSR	0x61000000
Banked	
System	
Internal	
Mode	Thread
Privilege	Privileged
Stack	MSP
States	6587
Sec	0.00054892

Disassembly

```
0x00000054 E7FE      B        0x00000054
0x00000056 0000      DCW      0x0000
0x00000058 0008      DCW      0x0008
0x0000005A 0000      DCW      0x0000
```

q1w2.s

```
    AREA RESET,DATA,READONLY
    EXPORT __Vectors

__Vectors

    DCD 0X10001000
    DCD Reset_Handler

    AREA mydata,DATA,READWRITE
DST DCD 0
    AREA mycode,CODE,READONLY
    ENTRY
SRC DCD 0xFFFFFFFF,0xFFFFFFFF,0xFFFFFFFF,0xFFFFFFFF,0xFFFFFFFF,0xFFFFFFFF,0xFFFFFFFF,0xFFFFFFFF,0xFFFFFFFF,0xFFFFFFFF
    EXPORT Reset_Handler

Reset_Handler

    LDR R0,=SRC
    LDR R1,=DST
    MOV R2,#0
    MOV R5,#0
    MOV R3,#10
```

Command

```
Running with Code Size Limit: 32K
Load "C:\\Users\\dell\\Desktop\\Wed ES\\Lab\\Objects\\ESLAB.axf"
BS 0x00000000, 1
```

```
ASSIGN BreakDisable BreakEnable BreakKill BreakList BreakSet BreakAccess COVERAGE COVTOFILE
```

Memory 1

Address: 0x10000000

```
0x10000000: F6 FF FF FF 09 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x1000001A: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x10000034: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x1000004E: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x10000068: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x10000082: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
```

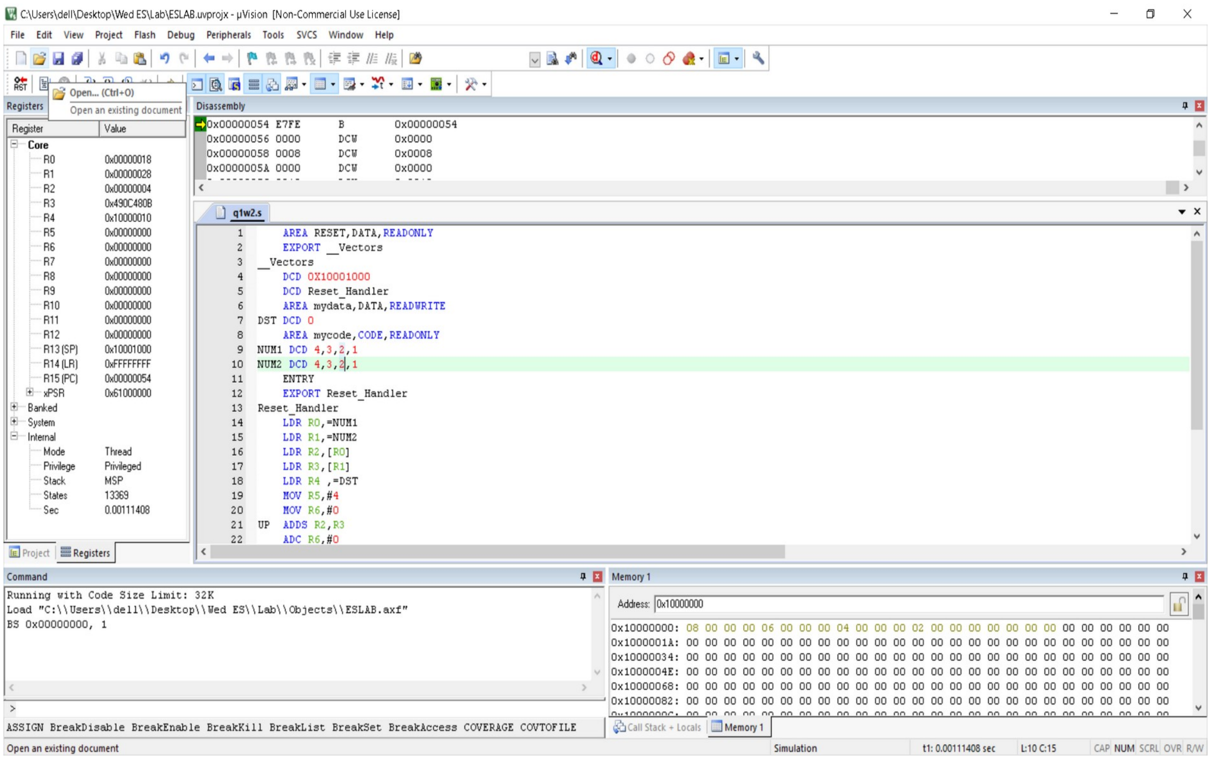
Simulation t1: 0.00487367 sec L:1 C:5

File Edit View Project Flash Debug Peripherals Tools SVCS Window Help
Project
Registers
Call Stack + Locals
Memory 1
CAP NUM SCRL OVR R/W

2. Write a program to add two 128-bit numbers available in code memory and store the result in data memory.
Hint: Use indexed addressing mode.

```
AREA RESET,DATA,READONLY
EXPORT __Vectors
__Vectors
DCD 0X10001000
DCD Reset_Handler
AREA mydata,DATA,READWRITE
DST DCD 0
AREA mycode,CODE,READONLY
NUM1 DCD 4,3,2,1
NUM2 DCD 4,3,2,1
ENTRY
EXPORT Reset_Handler
Reset_Handler
LDR R0,=NUM1
LDR R1,=NUM2
LDR R2,[R0]
LDR R3,[R1]
LDR R4 ,=DST
MOV R5,#4
MOV R6,#0
UP ADDS R2,R3
ADC R6,#0
STR R2,[R4],#4
LDR R2,[R0,#4]!
LDR R3,[R1,#4]!
SUBS R5,#1
BNE UP
STR R6,[R4]
ALIGN
STOP
B STOP
END
```

OUTPUT :



C:\Users\dell\Desktop\Wed ES\Lab\ESLAB.uvprojx - µVision [Non-Commercial Use License]

Register	Value
Core	
R0	0x00000018
R1	0x00000028
R2	0x00000004
R3	0x490C480B
R4	0x10000010
R5	0x00000000
R6	0x00000000
R7	0x00000000
R8	0x00000000
R9	0x00000000
R10	0x00000000
R11	0x00000000
R12	0x00000000
R13 (SP)	0x10001000
R14 (LR)	0xFFFFFFFF
R15 (PC)	0x00000054
xPSR	0x61000000
Banked	
System	
Internal	
Mode	Thread
Privilege	Privileged
Stack	MSP
States	13369
Sec	0.00111408

Disassembly

```
0x00000054 E7FE      B        0x00000054
0x00000056 0000      DCW      0x0000
0x00000058 0008      DCW      0x0008
0x0000005A 0000      DCW      0x0000
```

q1w2.s

```
    AREA RESET,DATA,READONLY
    EXPORT __Vectors
__Vectors
    DCD 0X10001000
    DCD Reset_Handler
    AREA mydata,DATA,READWRITE
DST DCD 0
    AREA mycode,CODE,READONLY
NUM1 DCD 4,3,2,1
NUM2 DCD 4,3,2,1
    ENTRY
    EXPORT Reset_Handler
Reset_Handler
    LDR R0,=NUM1
    LDR R1,=NUM2
    LDR R2,[R0]
    LDR R3,[R1]
    LDR R4 ,=DST
    MOV R5,#4
    MOV R6,#0
UP  ADDS R2,R3
    ADC R6,#0
```

Command

```
Running with Code Size Limit: 32K
Load "C:\\Users\\dell\\Desktop\\Wed ES\\Lab\\Objects\\ESLAB.axf"
BS 0x00000000, 1
```

```
ASSIGN BreakDisable BreakEnable BreakKill BreakList BreakSet BreakAccess COVERAGE COVTOFILE
```

Memory 1

Address: 0x10000000

```
0x10000000: 08 00 00 00 06 00 00 00 04 00 00 00 02 00 00 00 00 00 00 00 00 00 00 00 00 00
0x1000001A: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x10000034: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x1000004E: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x10000068: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x10000082: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
```

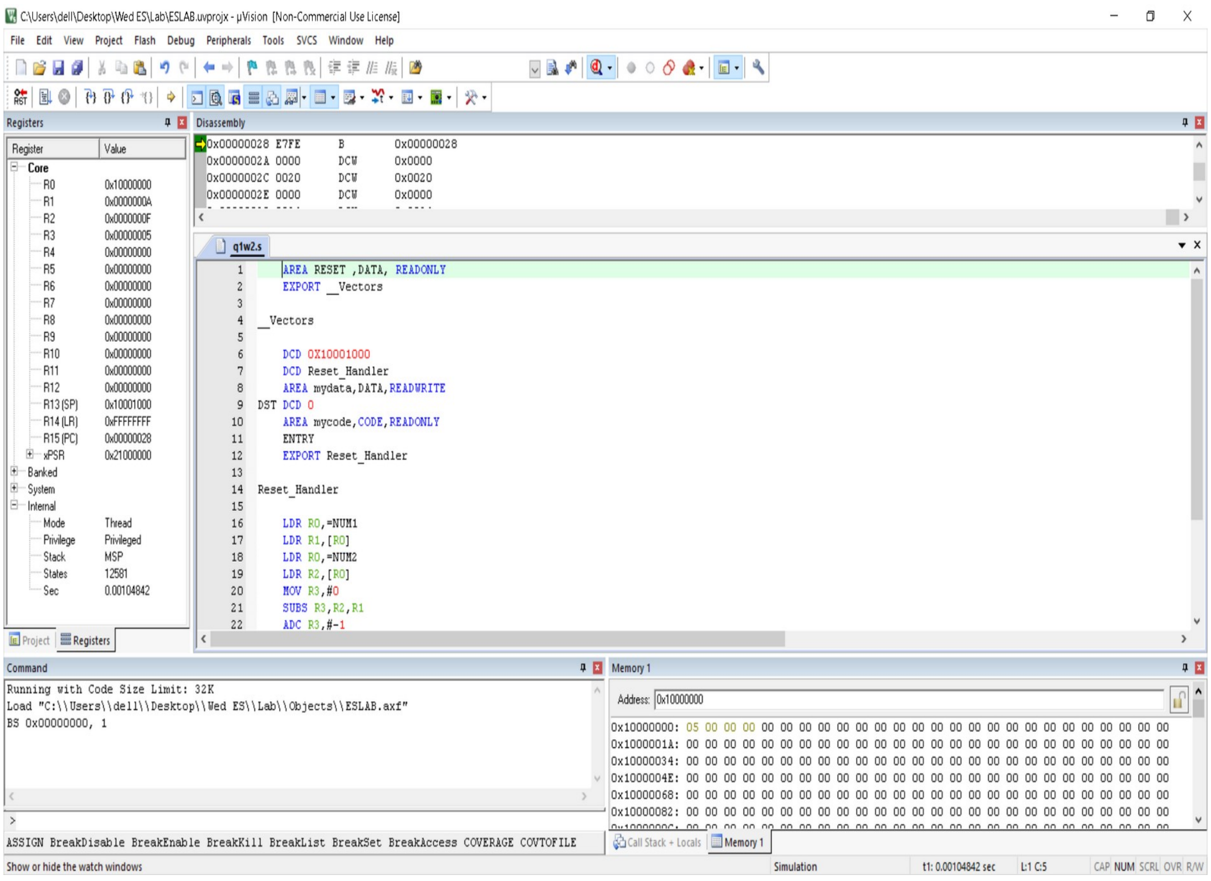
Open an existing document   Simulation   t1: 0.00111408 sec   L:10 C:15

3. Write a program to subtract two 32-bit numbers available in the code memory and store the result in the data memory.

```
AREA RESET ,DATA, READONLY
EXPORT __Vectors

__Vectors
DCD 0X10001000
DCD Reset_Handler
AREA mydata,DATA,READWRITE
DST DCD 0
AREA mycode,CODE,READONLY
ENTRY
EXPORT Reset_Handler

Reset_Handler

LDR R0,=NUM1
LDR R1,[R0]
LDR R0,=NUM2
LDR R2,[R0]
```

```
MOV R3,#0
SUBS R3,R2,R1
ADC R3,#-1
LDR R0,=DST
STR R3,[R0]
ALIGN
NUM1 DCD 10
NUM2 DCD 15
STOP B STOP
END
```

OUTPUT :



C:\Users\dell\Desktop\Wed ES\Lab\ESLAB.uvprojx - µVision [Non-Commercial Use License]

File Edit View Project Flash Debug Peripherals Tools SVCS Window Help

Registers

Register	Value
Core	
R0	0x10000000
R1	0x0000000A
R2	0x0000000F
R3	0x00000005
R4	0x00000000
R5	0x00000000
R6	0x00000000
R7	0x00000000
R8	0x00000000
R9	0x00000000
R10	0x00000000
R11	0x00000000
R12	0x00000000
R13 (SP)	0x10001000
R14 (LR)	0xFFFFFFFF
R15 (PC)	0x00000028
xPSR	0x21000000
Banked	
System	
Internal	
Mode	Thread
Privilege	Privileged
Stack	MSP
States	12581
Sec	0.00104842

Disassembly

```
0x00000028 E7FE      B        0x00000028
0x0000002A 0000      DCW      0x0000
0x0000002C 0020      DCW      0x0020
0x0000002E 0000      DCW      0x0000
```

q1w2.s

```
    AREA RESET ,DATA, READONLY
    EXPORT __Vectors

__Vectors

    DCD 0X10001000
    DCD Reset_Handler
    AREA mydata,DATA,READWRITE
DST DCD 0
    AREA mycode,CODE,READONLY
    ENTRY
    EXPORT Reset_Handler

Reset_Handler

    LDR R0,=NUM1
    LDR R1,[R0]
    LDR R0,=NUM2
    LDR R2,[R0]
    MOV R3,#0
    SUBS R3,R2,R1
    ADC R3,#-1
```

Command

```
Running with Code Size Limit: 32K
Load "C:\\Users\\dell\\Desktop\\Wed ES\\Lab\\Objects\\ESLAB.axf"
BS 0x00000000, 1
```

```
ASSIGN BreakDisable BreakEnable BreakKill BreakList BreakSet BreakAccess COVERAGE COVTOFILE
```

Memory 1

Address: 0x10000000

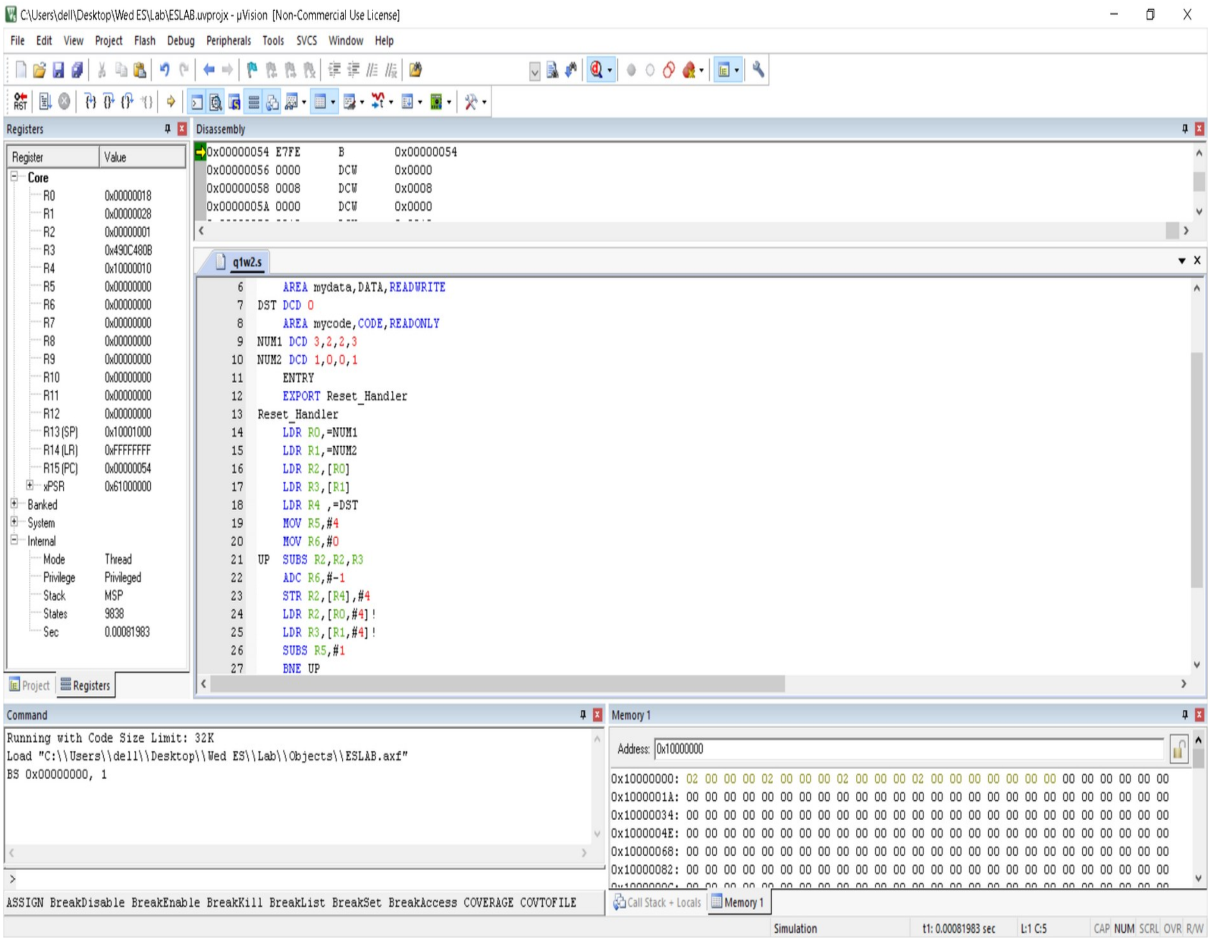
```
0x10000000: 05 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x1000001A: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x10000034: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x1000004E: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x10000068: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x10000082: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
```

Show or hide the watch windows   Simulation   t1: 0.00104842 sec   L:1 C:5   CAP NUM SCRL OVR R/W

4. Write a program to subtract two 128-bit numbers available in the code memory and store the result in the data memory.

```
AREA RESET,DATA,READONLY
EXPORT __Vectors
__Vectors
DCD 0X10001000
DCD Reset_Handler
AREA mydata,DATA,READWRITE
DST DCD 0
AREA mycode,CODE,READONLY
NUM1 DCD 3,2,2,3
NUM2 DCD 1,0,0,1
ENTRY
EXPORT Reset_Handler
Reset_Handler
LDR R0,=NUM1
LDR R1,=NUM2
LDR R2,[R0]
LDR R3,[R1]
LDR R4 ,=DST
MOV R5,#4
MOV R6,#0
UP SUBS R2,R2,R3
ADC R6,#-1
STR R2,[R4],#4
LDR R2,[R0,#4]!
LDR R3,[R1,#4]!
SUBS R5,#1
BNE UP
STR R6,[R4]
ALIGN
STOP
B STOP
END
```

OUTPUT :


C:\Users\dell\Desktop\Wed ES\Lab\ESLAB.uvprojx - µVision [Non-Commercial Use License]
File Edit View Project Flash Debug Peripherals Tools SVCS Window Help
Registers
Disassembly
q1w2.s
Command
Memory 1
Project
Registers
Call Stack + Locals
Memory 1
Simulation
t1: 0.00081983 sec
L:1 C:5
CAP NUM SCRL OVR R/W

Register	Value
Core	
R0	0x00000018
R1	0x00000028
R2	0x00000001
R3	0x490C480B
R4	0x10000010
R5	0x00000000
R6	0x00000000
R7	0x00000000
R8	0x00000000
R9	0x00000000
R10	0x00000000
R11	0x00000000
R12	0x00000000
R13 (SP)	0x10001000
R14 (LR)	0xFFFFFFFF
R15 (PC)	0x00000054
xPSR	0x61000000
Banked	
System	
Internal	
Mode	Thread
Privilege	Privileged
Stack	MSP
States	9838
Sec	0.00081983

```
0x00000054 E7FE      B         0x00000054
0x00000056 0000      DCW       0x0000
0x00000058 0008      DCW       0x0008
0x0000005A 0000      DCW       0x0000
```

```
    AREA mydata,DATA,READWRITE
DST DCD 0
    AREA mycode,CODE,READONLY
NUM1 DCD 3,2,2,3
NUM2 DCD 1,0,0,1
    ENTRY
    EXPORT Reset_Handler
Reset_Handler
    LDR R0,=NUM1
    LDR R1,=NUM2
    LDR R2,[R0]
    LDR R3,[R1]
    LDR R4 ,=DST
    MOV R5,#4
    MOV R6,#0
UP  SUBS R2,R2,R3
    ADC R6,#-1
    STR R2,[R4],#4
    LDR R2,[R0,#4]!
    LDR R3,[R1,#4]!
    SUBS R5,#1
    BNE UP
```

```
Running with Code Size Limit: 32K
Load "C:\\Users\\dell\\Desktop\\Wed ES\\Lab\\Objects\\ESLAB.axf"
BS 0x00000000, 1
```

```
ASSIGN BreakDisable BreakEnable BreakKill BreakList BreakSet BreakAccess COVERAGE COVTOFILE
```

Address: 0x10000000

```
0x10000000: 02 00 00 00 02 00 00 00 02 00 00 00 02 00 00 00 00 00 00 00 00 00 00 00 00 00
0x1000001A: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x10000034: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x1000004E: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x10000068: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x10000082: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
```