

Experiment 8

Aim:

To write an ARM Assembly Language program for division using repeated subtraction.

Tool Used:

Keil uVision4

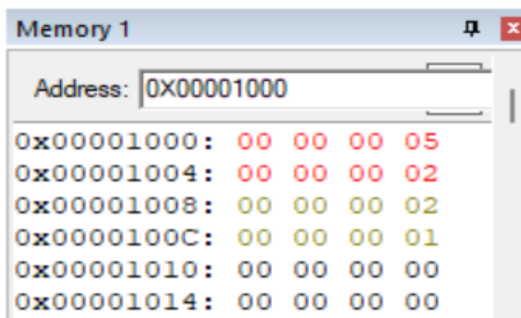
Theory:

LDR loads the register with some value. One register can be used as a counter. The first number can be subtracted by second number. On every loop the counter register is incremented on to the result.

Code:

```
AREA PROGRAM, CODE, READONLY
ENTRY
MAIN
    LDR R0, =0X00001000
    LDR R1, =0X00001004
    LDR R4, =0X00001008
    LDR R6, =0x0000100C
    LDR R2, [R0]
    LDR R3, [R1]
LOOP
    SUB R2, R2, R3
    ADD R5, R5, #1
    CMP R2, R3
    BGE LOOP
    STR R5, [R4]
    STR R2, [R6]
LOOP1 B LOOP1
END
```

Output:



Dividend Location – 0x00001000

Divisor location - 0x00001004

Quotient location – 0x00001008

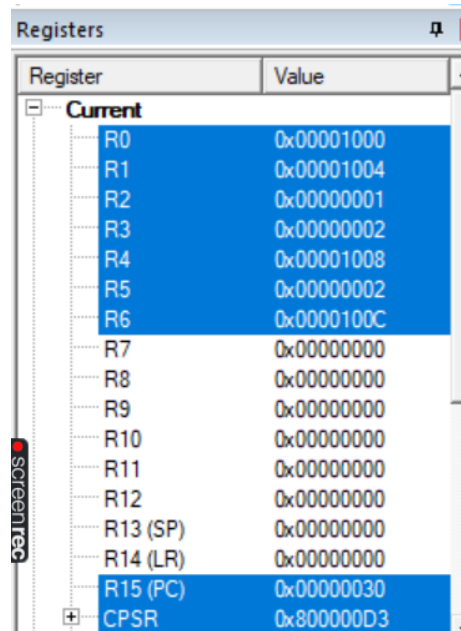
Remainder Location – 0x0000100C

Command

```
Running with Code Size Limit: 32K
Load "C:\\Users\\singh\\Documents\\keil embedded system\\experiment 8\\exp8.axf"

*** Restricted Version with 32768 Byte Code Size Limit
*** Currently used: 64 Bytes (0%)
```

Register Content

A screenshot of the 'Registers' window in a Keil IDE. The window has a title bar 'Registers' with a search icon and a close button. It contains a table with two columns: 'Register' and 'Value'. The table lists registers R0 through R15, plus CPSR. R0, R1, R2, R3, R4, R5, R6, R15 (PC), and CPSR are highlighted in blue. R13 is labeled '(SP)' and R14 is labeled '(LR)'. A vertical 'screenrec' watermark is visible on the left side of the window.

Register	Value
Current	
R0	0x00001000
R1	0x00001004
R2	0x00000001
R3	0x00000002
R4	0x00001008
R5	0x00000002
R6	0x0000100C
R7	0x00000000
R8	0x00000000
R9	0x00000000
R10	0x00000000
R11	0x00000000
R12	0x00000000
R13 (SP)	0x00000000
R14 (LR)	0x00000000
R15 (PC)	0x00000030
CPSR	0x800000D3

Result:

The experiments on Division operation using repeated subtraction have been performed and verified to be correct.