**Experiment 4**

**Aim:**

To write an ARM Assembly Language to find the number of bytes in a set of 10 locations that match the value 0xAC

**Tool Used:**

Keil uVision4

**Theory:**

LDRB is used to copy just 1 Byte of data to the lower location of the register. CMP compares two operands and if zero sets the zero flag. The EQ condition checks the zero flag for set to let the process happen.

**Code:**

  AREA PROGRAM, CODE, READONLY

  ENTRY

MAIN

        LDR R0, =0X00001000 //starting location

        MOV R2, #10 // counter for 10 locations

LOOP    LDRB R1, [R0], #1 // load the value

        CMP R1, #0XAC // check if same

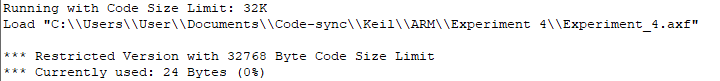
        ADDEQ R3,R3,#1 //if equal label

        SUBS R2,R2,#1 // decrement counter

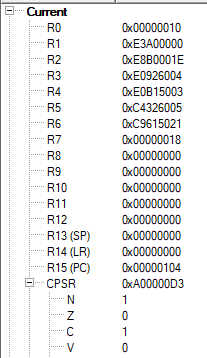
        BNE LOOP // run 10 times

 END

**Output:**

****

Register Contents

****

The memory location input data and the added value.

****

The input starts from 0x…0 to 0x0…08, the output is displayed from 0x00000010

**b) Add ten 32 bit numbers.**

**Code:**

  AREA PROGRAM,CODE, READONLY

 ENTRY

MAIN

    LDR R0, =0x00000000; SOURCE MEMORY LOCATION

    LDR R1, =0x00000050; DESTINATION MEMORY LOCATION

    MOV R3, #9; COUNTER WITH DATA 10

    LDR R4, [R0]; 1ST VALUE

FOR ADD R0, R0, #4; INCREMENTED ADDRESS

    LDR R5, [R0]; FURTHER VALUES

    ADDS R4,R4,R5; ADD CONSECUTIVE VALUES

    ADDCS R7,R7,#1; CARRY COUNT

    SUBS R3,R3,#1; DECREMENT THE COUNTER

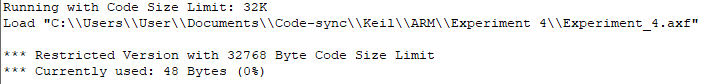
    BNE FOR; IF NOT ZERO REPEAT THE LOOP

    STR R4, [R1], #4; STORE THE ADDED VALUE

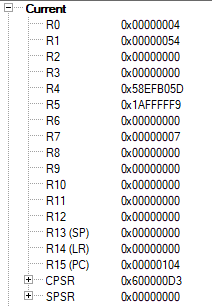
    STR R7, [R1]; STORE NUMBER OF CARRYS IN NEXT LOCATION

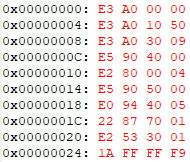
    END

**Output:**



Register Contents



The 10 input words 

The sum and carry



The input words range from 0x0…0 to 0x0…024 and the sum is displayed at 0x0…050 and the carry is in 0x0…054.

**Result:**

The experiments on add operations have been performed and verified to be correct.