**Experiment 4**

**Aim:**

To write an ARM Assembly Language to

1. Add two 64 bit numbers.
2. Add ten 32 bit numbers.

**Tool Used:**

Keil uVision4

**Theory:**

LDM load multiple register locations with starting address mentioned. ! is used in LDM for updating pointer, else same value will be updated in all registers. STM load the value into consecutive memory locations with starting address mentioned. ADDCS adds the value if the carry flag is set.

**a) Add two 64 bit numbers.**

**Code:**

 AREA PROGRAM,CODE, READONLY

 ENTRY

MAIN

 LDR R0, =0X00000000; SOURCE MEMORY LOCATION

 LDM R0!, {R1-R4}; COPY TO REGISTERS R1 TO R4 FROM MEM LOCATIONS

 ADDS R6,R2,R4; ADDS THE LOWER NIBBLE

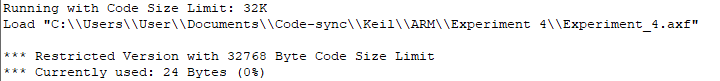
 ADCS R5,R1,R3; ADDS THE UPPER NIBBLE

 LDR R7, =0X00000010; DESTINATION MEMORY LOCATION

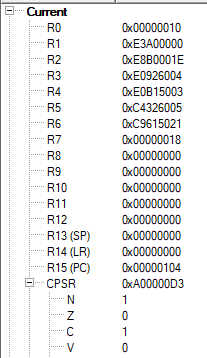
 STM R7!, {R5-R6}; STORE RESULTS IN DESTINATION

 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

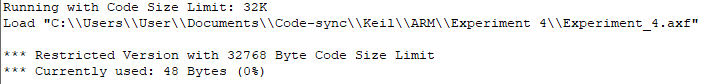
    BNE FOR; IF NOT ZERO REPEAT THE LOOP

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

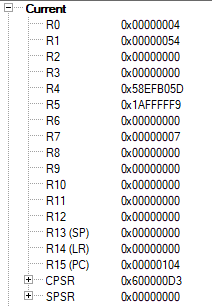
    STR R7, [R1]; STORE NUMBER OF CARRYS IN NEXT LOCATION

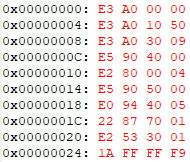
    END

**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.