

Microprocessor and Computer Architecture Laboratory

UE19CS256

4th Semester, Academic Year 2020-21

Date: 11/02/2021

Name : Abhishek Aditya BS	SRN: PES1UG19CS019	Section : A
---------------------------	-----------------------	-------------

Week# 3 Program Number: 1

Title of the Program

**Write an ALP to add two 64 bit numbers loaded
from memory and store the result in memory.**

I. ARM Assembly Code for each program

.data

A: .word 13,11

B: .word 14,19

C: .word 0

.text

LDR R0,=A ;address of 1st word is stored in R0

LDR R1,=B ;address of 2nd word is stored in R1

```

LDR R2,[R0] ;Storing 1st word in A in R2
LDR R3,[R0,#4] ;Storing 2nd word in A in R3

LDR R4,[R1] ;Storing 1st word in B in R4
LDR R5,[R1,#4] ;Storing 2nd word in B in R5

ADDS R6,R3,R5 ;Adding the 2nd words in A and B
and updating CPSR flags

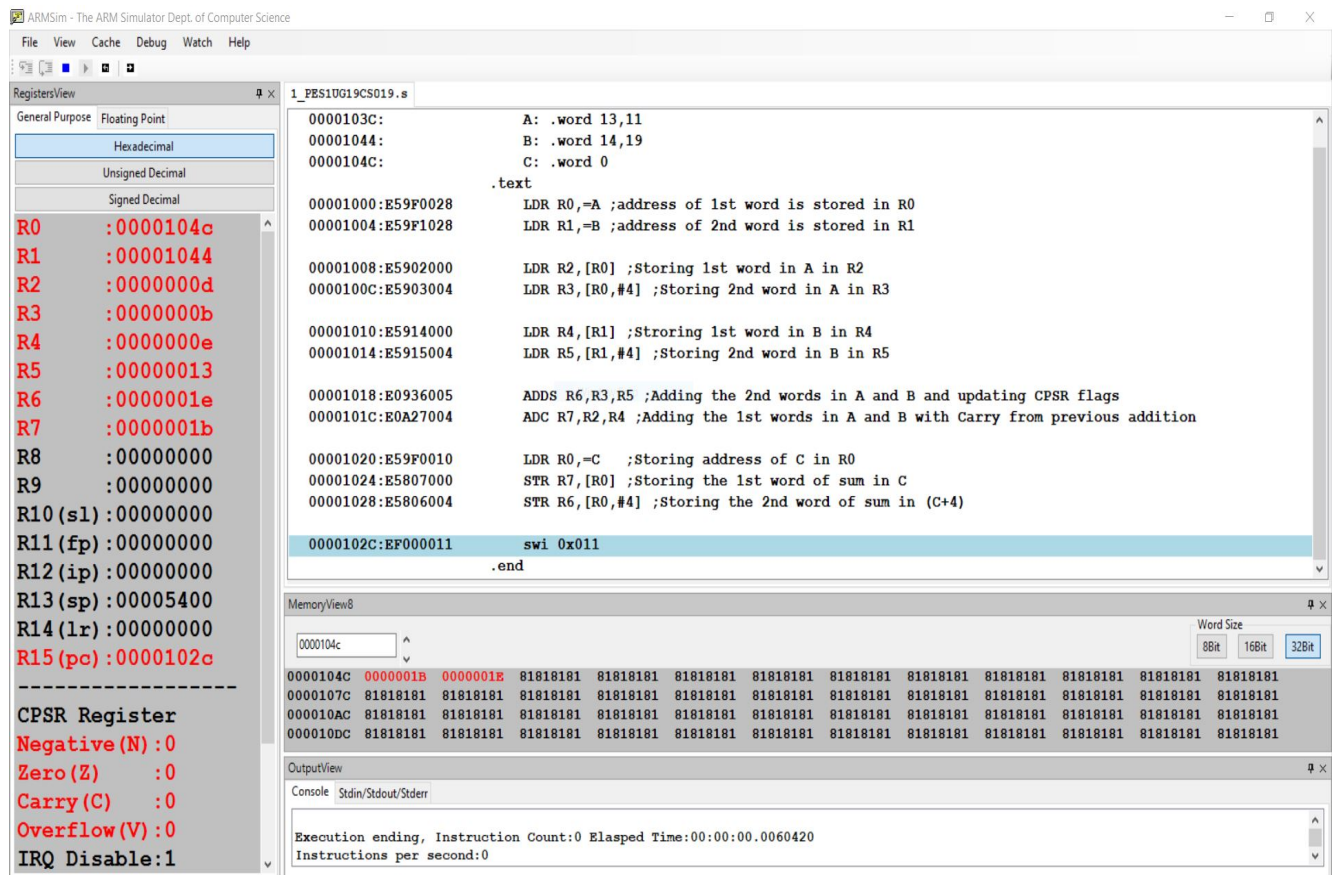
ADC R7,R2,R4 ;Adding the 1st words in A and B
with Carry from previous addition

LDR R0,=C ;Storing address of C in R0
STR R7,[R0] ;Storing the 1st word of sum in C
STR R6,[R0,#4] ;Storing the 2nd word of sum in (C+4)

swi 0x011
.end

```

II. Final Output Screen Shot



Microprocessor and Computer Architecture Laboratory

UE19CS256

4th Semester, Academic Year 2020-21

Date: 11/02/2021

Name : Abhishek Aditya BS	SRN: PES1UG19CS019	Section : A
---------------------------	-----------------------	-------------

Week# 3 Program Number: 2

Title of the Program

Write an ALP to copy n numbers from Memory Location A to Memory Location B

I. ARM Assembly Code for each program

```
.data

    A: .word 5,6,8,7,90

    B: .word 0

.text

    LDR R0,=A ;address of A is stored in R0

    LDR R1,=B ;address of B is stored in R1

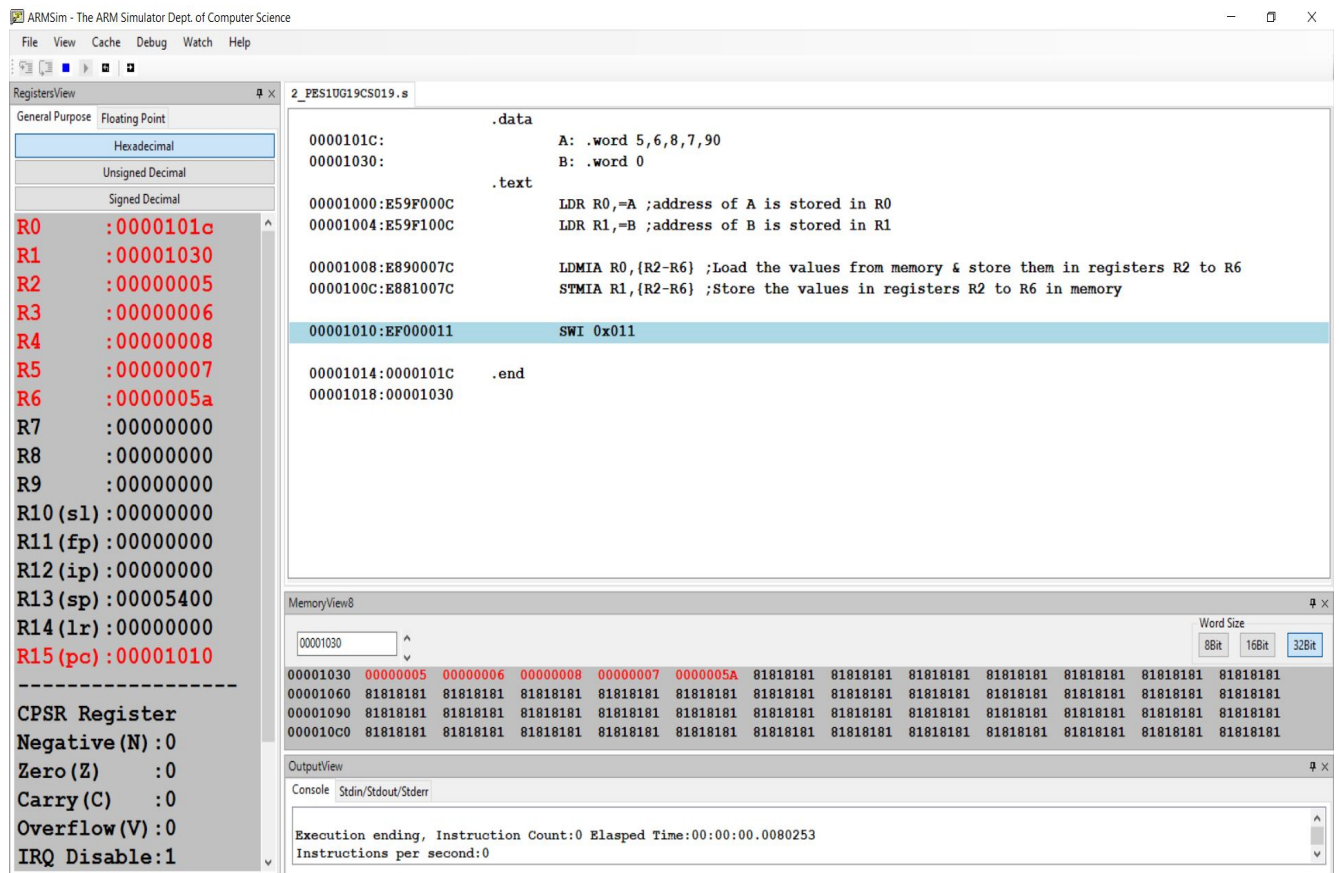

    LDMIA R0,{R2-R6} ;Load the values from
memory & store them in registers R2 to R6

    STMIA R1,{R2-R6} ;Store the values in
registers R2 to R6 in memory


    SWI 0x011

.end
```

II. Final Output Screen Shot



Microprocessor and Computer Architecture Laboratory

UE19CS256

4th Semester, Academic Year 2020-21

Date: 11/02/2021

Name : Abhishek Aditya BS	SRN: PES1UG19CS019	Section : A
---------------------------	-----------------------	-------------

Week# 3 Program Number: 3

Title of the Program

Write an ALP to find smallest number in an array of n - 32 bit numbers

I. ARM Assembly Code for each program

.data

A: .word 144,55,26,19,321

.text

LDR R0,=A ;address of A is stored in R0

**LDR R5,[R0] ;Register R5 is used to store
the smallest number in the array**

**MOV R1,#5 ;Register R1 is used to store the
number of elements in the array**

LOOP: CMP R1,#0 ;Check if R1 is 0

**BEQ END ;If R1 is 0 then all elements
have been processed**

**LDR R2,[R0],#4 ;Load a value from A
in R2 and increment R0 by 4**

**CMP R2,R5 ;Compare the current min
value and element fetched**

```
        BMI SMALL ;If result is negative,  
element is smaller than min value
```

```
        SUB R1,R1,#1 ;Decrementing count of  
elements
```

```
        B LOOP ;Process the next element
```

```
    SMALL: MOV R5,R2 ;Update the new min value
```

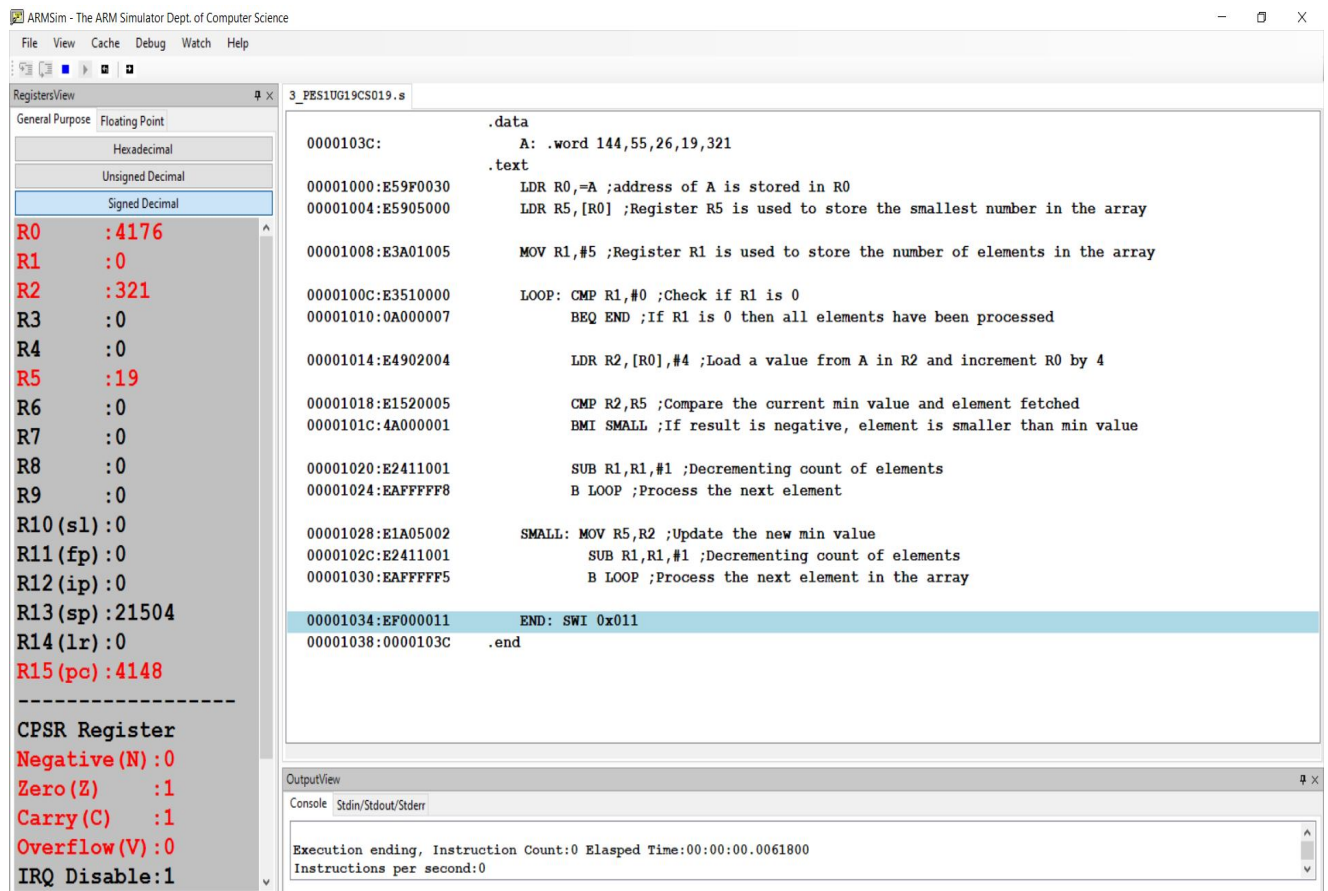
```
        SUB R1,R1,#1 ;Decrementing count of  
elements
```

```
        B LOOP ;Process the next element in  
the array
```

```
    END: SWI 0x011
```

```
.end
```

II. Final Output Screen Shot



Microprocessor and Computer Architecture Laboratory

UE19CS256

4th Semester, Academic Year 2020-21

Date: 11/02/2021

Name : Abhishek Aditya BS	SRN: PES1UG19CS019	Section : A
---------------------------	-----------------------	-------------

Week# 3 Program Number: 4

Title of the Program

a) Write an ALP to count the number of 1's and 0's in a given 32 bit number.

I. ARM Assembly Code for each program

```
.text
```

```
LDR R0, =0b00010111011010101010011000001110
```

```
MOV R2, #32 ;Register R2 is used to store  
Number of digits in the number
```

```
MOV R5, #0 ;Register R5 is used to store  
the no. of 1's in the number
```

```
LOOP:
```

```
SUB R2, R2, #1 ;Decrementing the number  
of digits
```

```
;Performing Logical Right Shift on the number  
and updating the flags
```

```
MOVS R0, R0, LSR #1
```

```
ADDCS R5, R5, #1 ;Check if the digit is  
1 and update R5
```

```
CMP R2, #0 ;Check if all digits are  
processed
```

```
BNE LOOP
```

MOV R2, #32

**SUB R6,R2, R5 ;No. of 0's is stored in
Register R6 = 32 - No. of 1's**

SWI 0x011

.end

II. Final Output Screen Shot

The screenshot displays the ARMSim - The ARM Simulator interface. The main window is titled '4a_PES1UG19CS019.s'. The 'Registers View' panel on the left shows the state of the ARM registers. The 'Output View' panel at the bottom shows the execution log.

Registers View:

Register	Value
R0	:0
R1	:0
R2	:32
R3	:0
R4	:0
R5	:15
R6	:17
R7	:0
R8	:0
R9	:0
R10(s1)	:0
R11(fp)	:0
R12(ip)	:0
R13(sp)	:21504
R14(lr)	:0
R15(pc)	:4136

CPSR Register:

Flag	Value
Negative (N)	:0
Zero (Z)	:1
Carry (C)	:1
Overflow (V)	:0
IRQ Disable	:1

Assembly Code:

```
.text
00001000:E59F0024    LDR R0, =0b00010111010101010011000001110
00001004:E3A02020    MOV R2, #32 ;Register R2 is used to store Number of digits in the number
00001008:E3A05000    MOV R5, #0 ;Register R5 is used to store the no. of 1's in the number

0000100C:          LOOP:
0000100C:E2422001    SUB R2, R2, #1 ;Decrementing the number of digits

                    ;Performing Logical Right Shift on the number and updating the flags
00001010:E1B000A0    MOVS R0, R0, LSR #1

00001014:22855001    ADDCS R5, R5, #1 ;Check if the digit is 1 and update R5

00001018:E3520000    CMP R2, #0 ;Check if all digits are processed
0000101C:1AFFFFFA    BNE LOOP

00001020:E3A02020    MOV R2, #32
00001024:E0426005    SUB R6,R2, R5 ;No. of 0's is stored in Register R6 = 32 - No. of 1's
00001028:EF000011    SWI 0x011
0000102C:176AA60E    .end
```

Output View:

```
Execution ending, Instruction Count:0 Elapsed Time:00:00:00.0070154
Instructions per second:0
```

b) Write an ALP to find the number of zeroes, positive and negative numbers in a given array

I. ARM Assembly Code for each program

```
.data

    A: .word -1,99,22,-230,0

.text

    LDR R0,=A ;Storing address of A in R0

    MOV R2,#5 ;Number of elements in the array

    MOV R3,#0 ;Storing the no. of 0's in array

    MOV R4,#0 ;Storing the no. of positive
elements in the array

    MOV R5,#0 ;Storing the no. of negative
elements in the array


    LOOP:

        CMP R2,#0 ;Check if all elements have
been processed

        BEQ END

        SUB R2,R2,#1 ;Decrement the no. of
elements to process


    LDR R1,[R0] ;Load an element from A to R1
```

MOVS R1,R1 ;Update the flags for the
element

BEQ ZERO ;If element is 0

BMI NEGATIVE ;If element is negative

ADD R4,R4,#1;If element is
positive,increment the count(R4)

ADD R0,R0,#4 ;Update R0 to next element
in array

B LOOP ;Process the next element

ZERO:

ADD R3,R3,#1 ;Increment the count(R3)

ADD R0,R0,#4 ;Update R0 to next element
in array

B LOOP ;Process the next element

NEGATIVE:

ADD R5,R5,#1 ;Increment the count(R5)

ADD R0,R0,#4 ;Update R0 to next element
in array

B LOOP ;Process the next element

END: SWI 0x011

.end

II. Final Output Screen Shot

The screenshot displays the ARMSim - The ARM Simulator interface. The main window shows assembly code for a file named 4b_PES1UG19CS019.s. The code includes a data section with a word array A, and a text section with instructions for loading, moving, and processing array elements. The RegistersView panel on the left shows the current state of registers R0 through R15, with R15 (pc) at 00001054. The CPSR Register panel shows flags: Negative (N) is 0, Zero (Z) is 1, Carry (C) is 1, Overflow (V) is 0, and IRQ Disable is 1. The OutputView panel at the bottom shows the execution ending with an instruction count of 0 and an elapsed time of 00:00:00.3078388.

```
.data
0000105C:          A: .word -1,99,22,-230,0
.text
00001000:E59F0050      LDR R0,=A ;Storing address of A in R0
00001004:E3A02005      MOV R2,#5 ;Number of elements in the array
00001008:E3A03000      MOV R3,#0 ;Storing the no. of 0's in array
0000100C:E3A04000      MOV R4,#0 ;Storing the no. of positive elements in the array
00001010:E3A05000      MOV R5,#0 ;Storing the no. of negative elements in the array
00001014:
00001014:E3520000      LOOP:
00001018:0A00000D      CMP R2,#0 ;Check if all elements have been processed
0000101C:E2422001      BEQ END
0000101C:E2422001      SUB R2,R2,#1 ;Decrement the no. of elements to process
00001020:E5901000      LDR R1,[R0] ;Load an element from A to R1
00001024:E1B01001      MOVS R1,R1 ;Update the flags for the element
00001028:0A000003      BEQ ZERO ;If element is 0
0000102C:4A000005      BMI NEGATIVE ;If element is negative
00001030:E2844001      ADD R4,R4,#1 ;If element is positive,increment the count(R4)
00001034:E2800004      ADD R0,R0,#4 ;Update R0 to next element in array
00001038:EAF0FF5      B LOOP ;Process the next element
0000103C:
0000103C:E2833001      ZERO:
00001040:E2800004      ADD R3,R3,#1 ;Increment the count(R3)
00001040:E2800004      ADD R0,R0,#4 ;Update R0 to next element in array
```

RegistersView

General Purpose	Floating Point
Hexadecimal	
Unsigned Decimal	
Signed Decimal	

R0 : 00001070
R1 : 00000000
R2 : 00000000
R3 : 00000001
R4 : 00000002
R5 : 00000002
R6 : 00000000
R7 : 00000000
R8 : 00000000
R9 : 00000000
R10 (s1) : 00000000
R11 (fp) : 00000000
R12 (ip) : 00000000
R13 (sp) : 00005400
R14 (lr) : 00000000
R15 (pc) : 00001054

CPSR Register

Negative (N) : 0
Zero (Z) : 1
Carry (C) : 1
Overflow (V) : 0
IRQ Disable : 1

OutputView

Console Stdin/Stdout/Stderr

Execution ending, Instruction Count:0 Elapsed Time:00:00:00.3078388
Instructions per second:0

Microprocessor and Computer Architecture Laboratory

UE19CS256

4th Semester, Academic Year 2020-21

Date: 11/02/2021

Name : Abhishek Aditya BS	SRN: PES1UG19CS019	Section : A
---------------------------	-----------------------	-------------

Week#____3____Program Number: ____5____

Title of the Program

Write an ALP to check whether a given number is present in array using Linear Search (Without SWI 0x02), if found move +1 to R6 and key position to R7 else move -1 to R6 (if number not found).

I. ARM Assembly Code for each program

.data

A: .word 11,1,56,88,155

.text

LDR R0,=A ;Storing the address of A in R0

MOV R2,#88 ;Key element to be searched is stored in R2

MOV R3,#5 ;No. of elements in the array is stored in R3

L2: LDR R1,[R0],#4 ;Load a value from A to R1 and increment R0 to next element

CMP R1,R2 ;Compare element and key

BEQ FOUND ;If element = key, search successful

```
        SUBS R3,R3,#1 ;Decrement the count of
elements & update flags
```

```
        BNE L2 ;Process the next element
```

```
        MOV R6,#-1 ;If all elements processed ,
search unsuccessful
```

```
        B END ;End execution
```

FOUND:

```
        RSB R7,R3,#6 ;Store position of the
element found in R7
```

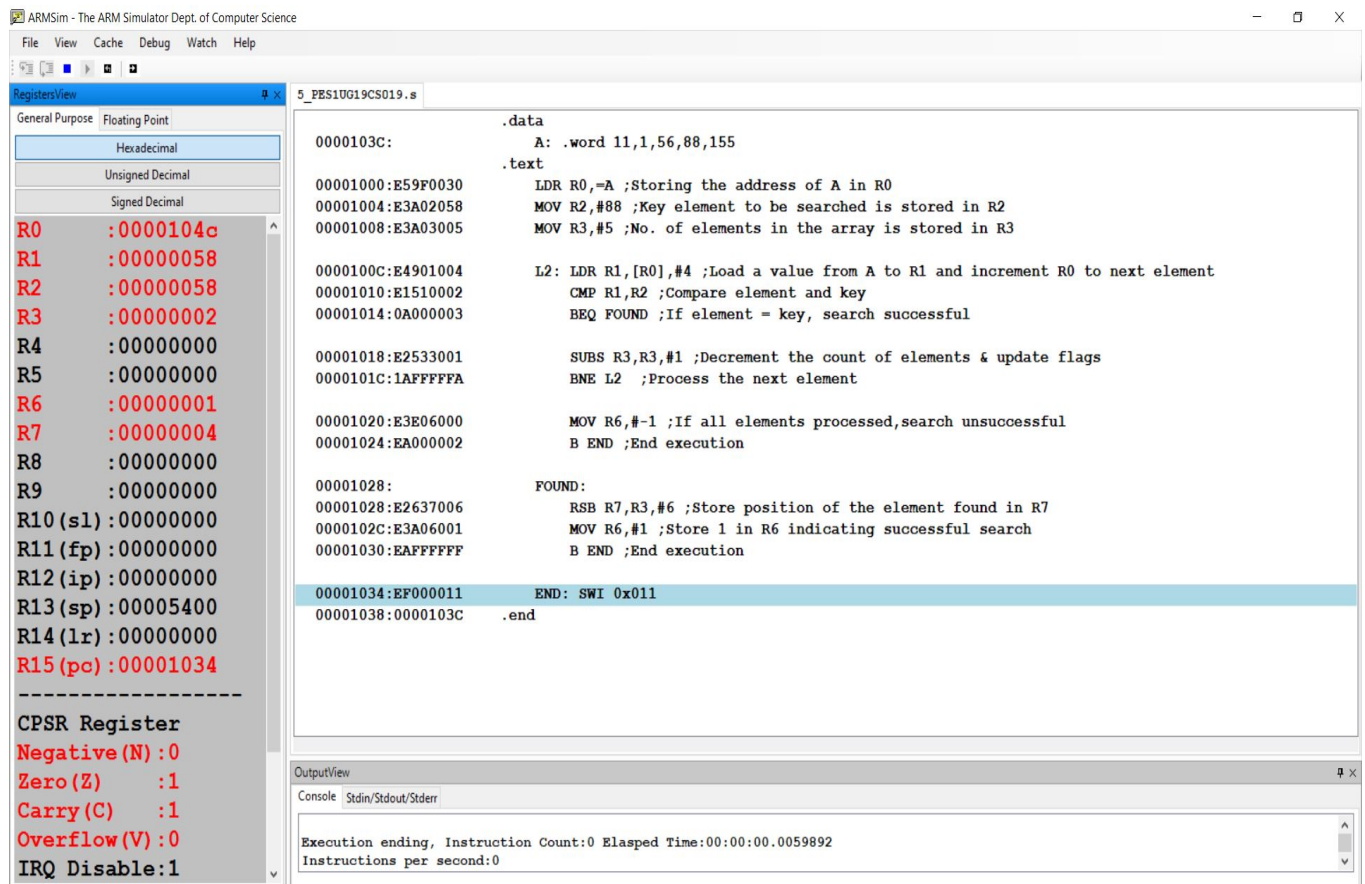
```
        MOV R6,#1 ;Store 1 in R6 indicating
successful search
```

```
        B END ;End execution
```

```
END: SWI 0x011
```

```
.end
```

II. Final Output Screen Shot



Microprocessor and Computer Architecture Laboratory

UE19CS256

4th Semester, Academic Year 2020-21

Date: 11/02/2021

Name : Abhishek Aditya BS	SRN: PES1UG19CS019	Section : A
---------------------------	-----------------------	-------------

Week# 3 Program Number: 6

Title of the Program

Write an ALP to generate Fibonacci Series and store them in an array

I. ARM Assembly Code for each program

.data

A: .word 0

.text

MOV R0,#10 ;Number of elements in the series to be generated after 0 and 1

LDR R1,=A ;Store the address of A in R1

MOV R2,#0 ;Storing first no. in the fib series

MOV R3,#1 ;Storing second no. in the fib series

STR R2,[R1],#4 ;Store 0 in array and increment R1

STR R3,[R1],#4 ;Store 1 in array and increment R1

LOOP: ADD R4,R2,R3 ;Calculate the next fib number

STR R4,[R1],#4 ;Store the calculated fib no. in the array

MOV R2,R3 ;Store the next fib no. R3
in R2

MOV R3,R4 ;Store the new fib no. R4
in R3

SUBS R0,R0,#1 ;Decrement the count of
elements & updating CPSR flags

BNE LOOP ;Generate the next fib no.

SWI 0x011 ;If all elements generated,end
the execution

.end

II. Final Output Screen Shot

The screenshot displays the ARMSim interface with the following components:

- Registers View:** Shows the state of 16 registers. R0 through R15 are listed with their hexadecimal values. R15 (PC) is 00001030.
- CPSR Register:** Shows the status of various flags: Negative (N) is 0, Zero (Z) is 1, Carry (C) is 1, Overflow (V) is 0, and IRQ Disable is 1.
- Assembly Code:** The main window displays the assembly code for file 6_PES1UG19CS019.s. The code includes comments for each instruction, such as "MOV R2,R3 ;Store the next fib no. R3 in R2".
- Memory View:** Shows a memory dump starting at address 00001038, displaying hexadecimal data in columns.
- Output View:** The console shows the message "Execution ending, Instruction Count:0 Elapsed Time:00:00:00.0060797" and "Instructions per second:0".

Disclaimer:

- The programs and output submitted is duly written, verified and executed by me.
- I have not copied from any of my peers nor from external resources such as the internet.
- If found plagiarized, I will abide with the disciplinary action of the University.

Signature : Abhishek Aditya BS

Date : 11/02/2021

Name : Abhishek Aditya BS

SRN : PES1UG19CS019

Section : A