**Microprocessor and Computer Architecture Laboratory**

**UE19CS256**

**4th Semester, Academic Year 2020-21**

Date:

|  |  |  |
| --- | --- | --- |
| Name: A Narendiran | SRN: PES1UG19CS001 | Section  A |

Week#\_\_\_\_\_4\_\_\_\_\_ Program Number: \_\_\_1\_\_\_

ARM Assembly Code for the program

.data

a: .word 10, 20, 30, 40, 50

b: .word 10, 20, 30, 40, 50

c: .word 0,0,0,0,0

.text

MOV r1, #5

LDR r2, =a

LDR r3, =b

LDR r0, =c

BL loop

loop:

LDR r4, [r2]

LDR r5, [r3]

ADD r7, r4, r5

STR r7, [r0]

ADD r2, r2, #4

ADD r3, r3, #4

ADD r0, r0, #4

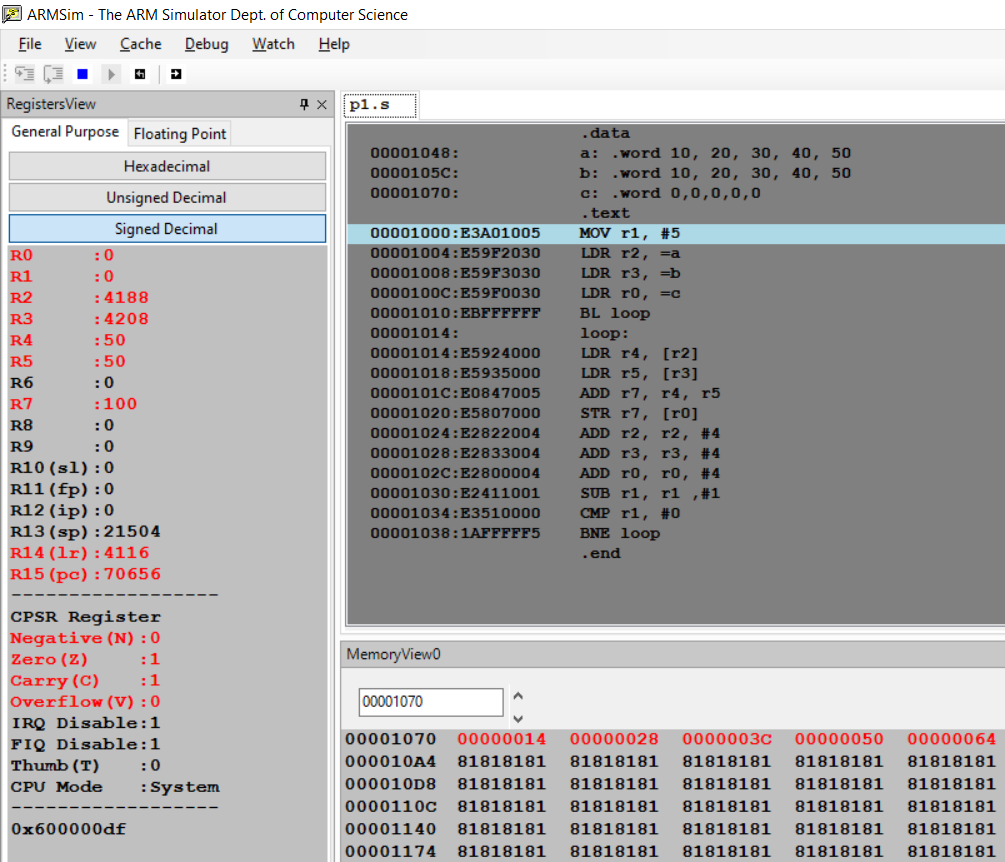
SUB r1, r1 ,#1

CMP r1, #0

BNE loop

.end

Output Screen Shot



Week#\_\_\_\_4\_\_\_\_\_ Program Number: \_\_\_2\_\_\_

ARM Assembly Code for the program

.data

a: .word 10, 20, 30, 40, 50

b: .word 10, 20, 30, 40, 50

c: .word 0,0,0,0,0

.text

MOV r1, #5

LDR r2, =a

LDR r3, =b

LDR r0, =c

BL loop

loop:

LDR r4, [r2]

LDR r5, [r3]

MUL r7, r4, r5

STR r7, [r0]

ADD r2, r2, #4

ADD r3, r3, #4

ADD r0, r0, #4

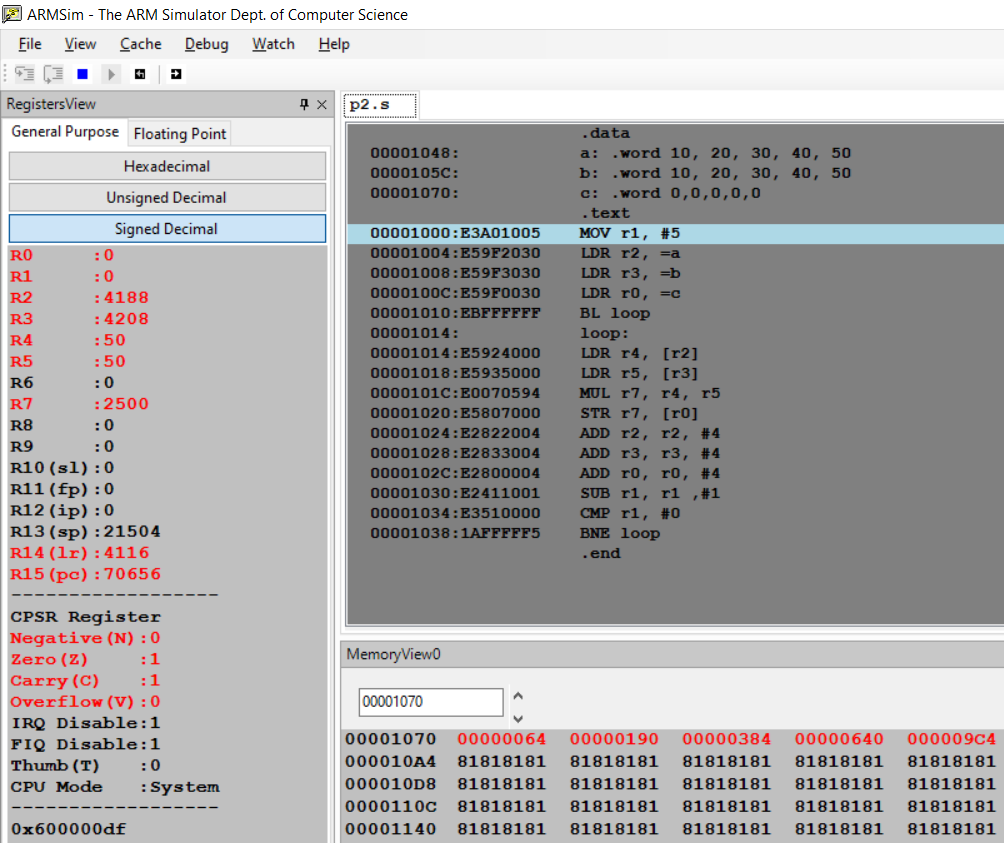
SUB r1, r1 ,#1

CMP r1, #0

BNE loop

.end

Output Screen Shot



Week#\_\_\_\_3\_\_\_\_\_\_ Program Number: \_\_\_3a\_\_\_

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

ARM Assembly Code for the program

.data

a: .word 10, 20, 30, 40, 50

b: .word 10, 20, 30, 40, 50

c: .word 0,0,0,0,0

.text

MOV r1, #5

LDR r2, =a

LDR r3, =b

LDR r0, =c

BL loop

loop:

LDR r4, [r2]

LDR r5, [r3]

MUL r8, r4, r5

ADD r7, r7, r8

STR r7, [r0]

ADD r2, r2, #4

ADD r3, r3, #4

ADD r0, r0, #4

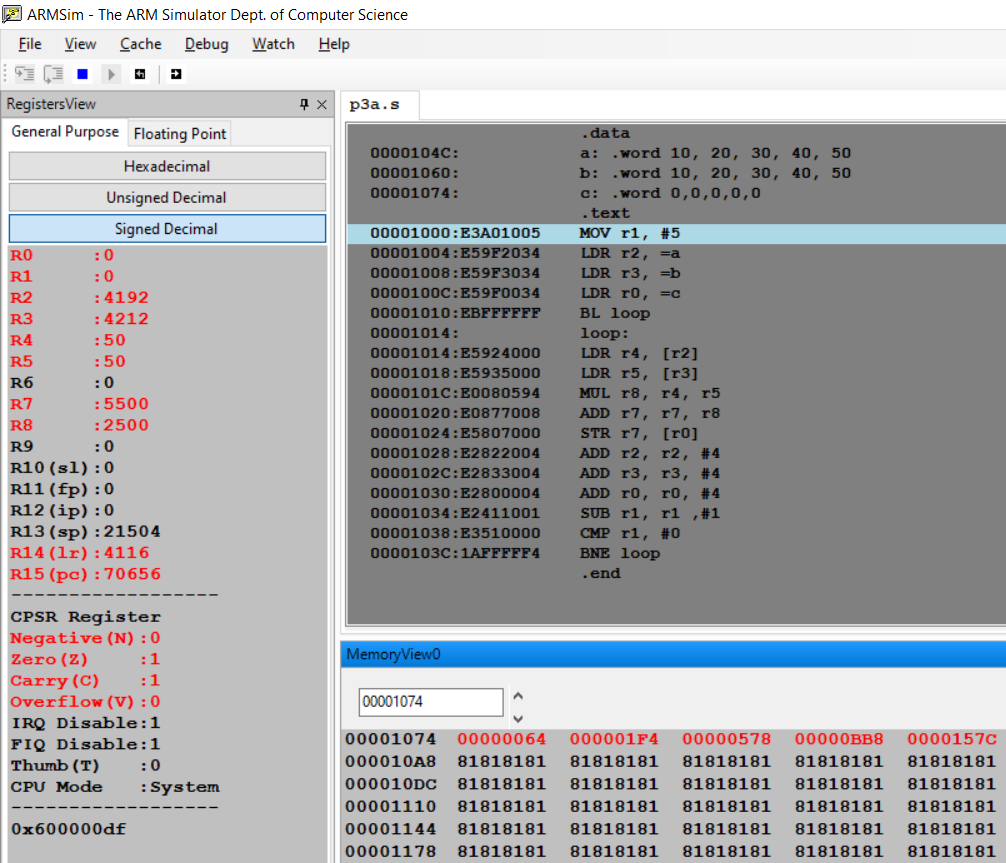
SUB r1, r1 ,#1

CMP r1, #0

BNE loop

.end

Output Screen Shot



Week#\_\_\_\_3\_\_\_\_\_\_ Program Number: \_\_3b\_\_

ARM Assembly Code for the program

.data

a: .word 10, 20, 30, 40, 50

b: .word 10, 20, 30, 40, 50

c: .word 0,0,0,0,0

.text

MOV r1, #5

LDR r2, =a

LDR r3, =b

LDR r0, =c

BL loop

loop:

LDR r4, [r2]

LDR r5, [r3]

MLA r7, r4, r5, r7

STR r7, [r0]

ADD r2, r2, #4

ADD r3, r3, #4

ADD r0, r0, #4

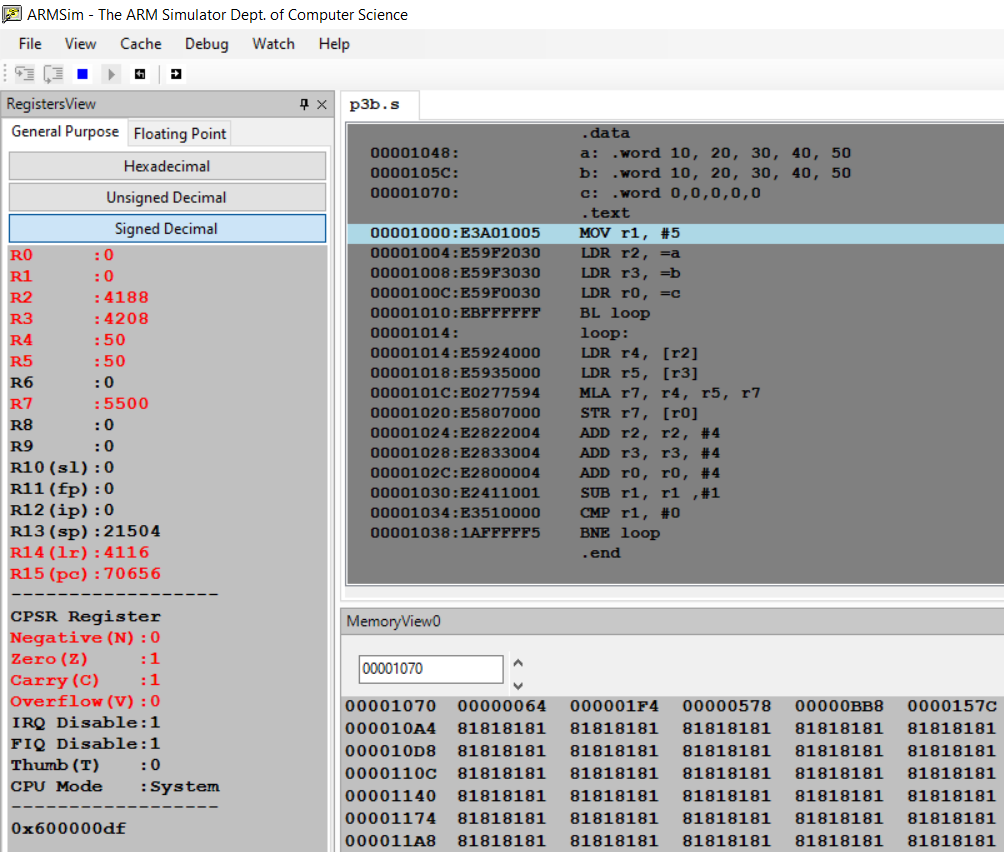
SUB r1, r1 ,#1

CMP r1, #0

BNE loop

.end

Output Screen Shot



Week#\_\_\_\_3\_\_\_\_\_\_\_ Program Number: \_\_\_4b\_\_\_

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

ARM Assembly Code for the program

Output Screen Shot

Week#\_\_\_\_3\_\_\_\_\_\_\_ Program Number: \_\_5\_\_

5. 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)

ARM Assembly Code for the program

.data

a: .WORD 3,9,10,12,16,38,40,58,65,79

.text

LDR r0, =a

MOV r1, #10

MOV r4,#0 *;to count the position*

Loop:

LDR r2, [r0]

CMP r2, r1

BEQ l1

ADD r4, r4, #1

ADD r0, r0, #4

SUBS r1, r1, #1

BNE Loop

MOV R3, #-1 *;if not found*

SWI 0x011

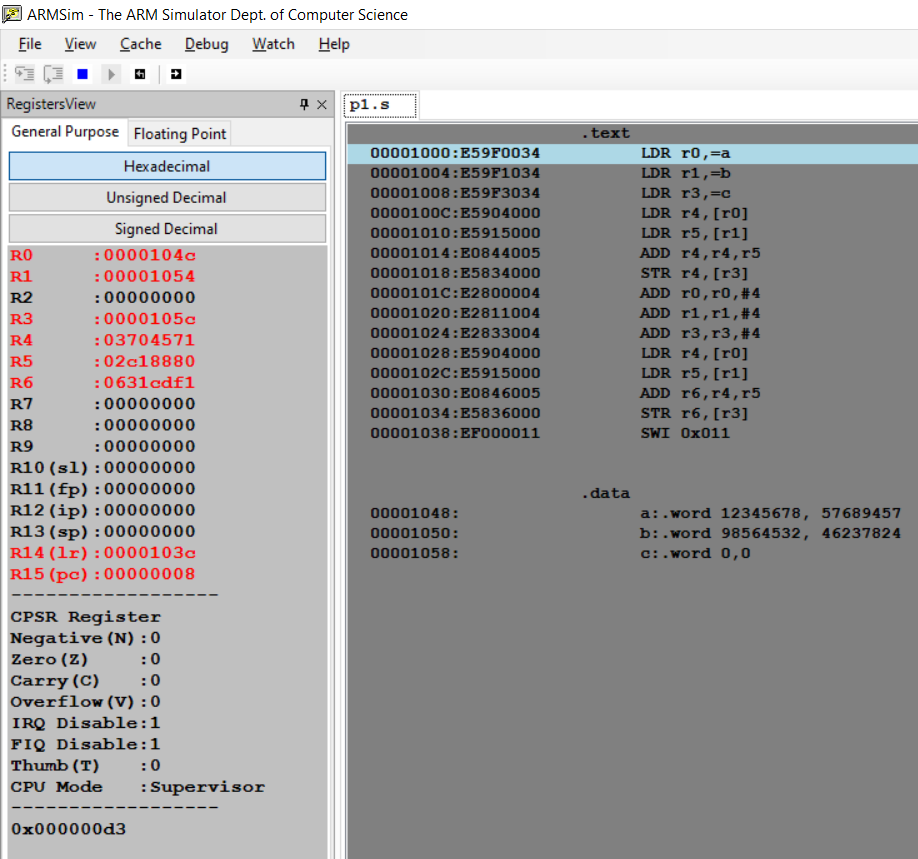
l1:

MOV R3, #1 *;if found*

SWI 0x011

.end

Output Screen Shot



Week#\_\_\_\_3\_\_\_\_\_ Program Number: \_\_6\_\_

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

ARM Assembly Code for the program

.text

MOV r0, #10

MOV r1, #0

MOV r2, #1

LDR r3, =a

STR r1, [r3], #4

STR r2 ,[r3], #4

CMP r0, #1

BNE Loop

Loop:

ADD r5, r1, r2

MOV r1, r2

MOV r2, r5

STR r5, [r3], #4

SUBS r0, r0, #1

CMP r0, #1

BEQ exit

BL Loop

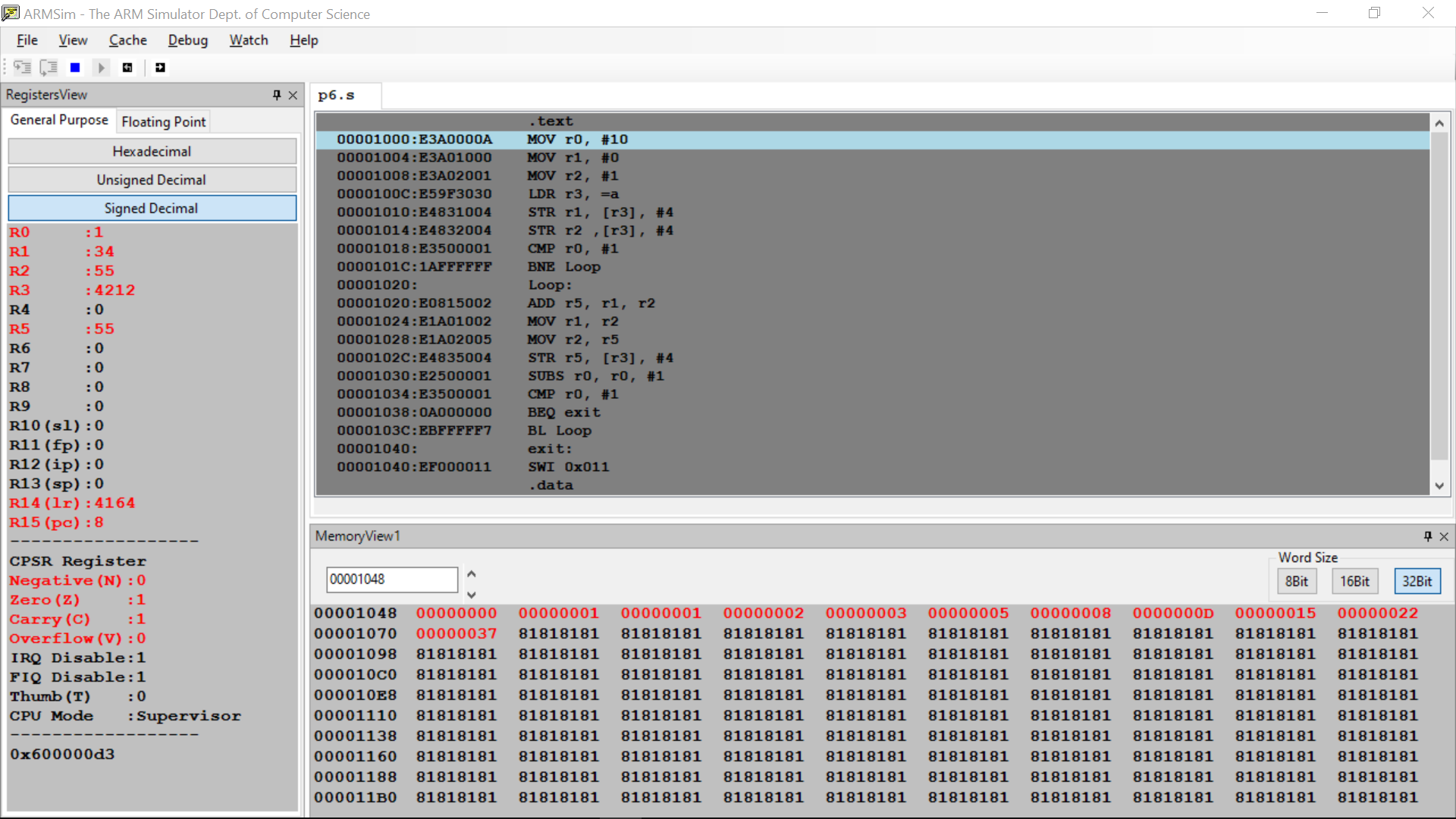
exit:

SWI 0x011

.data

a: .WORD

Output Screen Shot

****