**Microprocessor and Computer Architecture**

**UE21CS251B**

**4th Semester, Academic Year 2022-23**

Date:

|  |  |  |
| --- | --- | --- |
| Name: | SRN: | Section |

Week#\_\_\_\_4\_\_\_\_\_\_\_ 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

.data

a: .word 23127437,32287784

b: .word 12345678,87654321

sum: .word 0,0

.text

ldr r0, =a

ldr r1, =b

ldr r2, =sum

ldr r3, [r0],#4

ldr r4, [r1],#4

add r3, r3, r4

str r3, [r2],#4

ldr r3, [r0]

ldr r4, [r1]

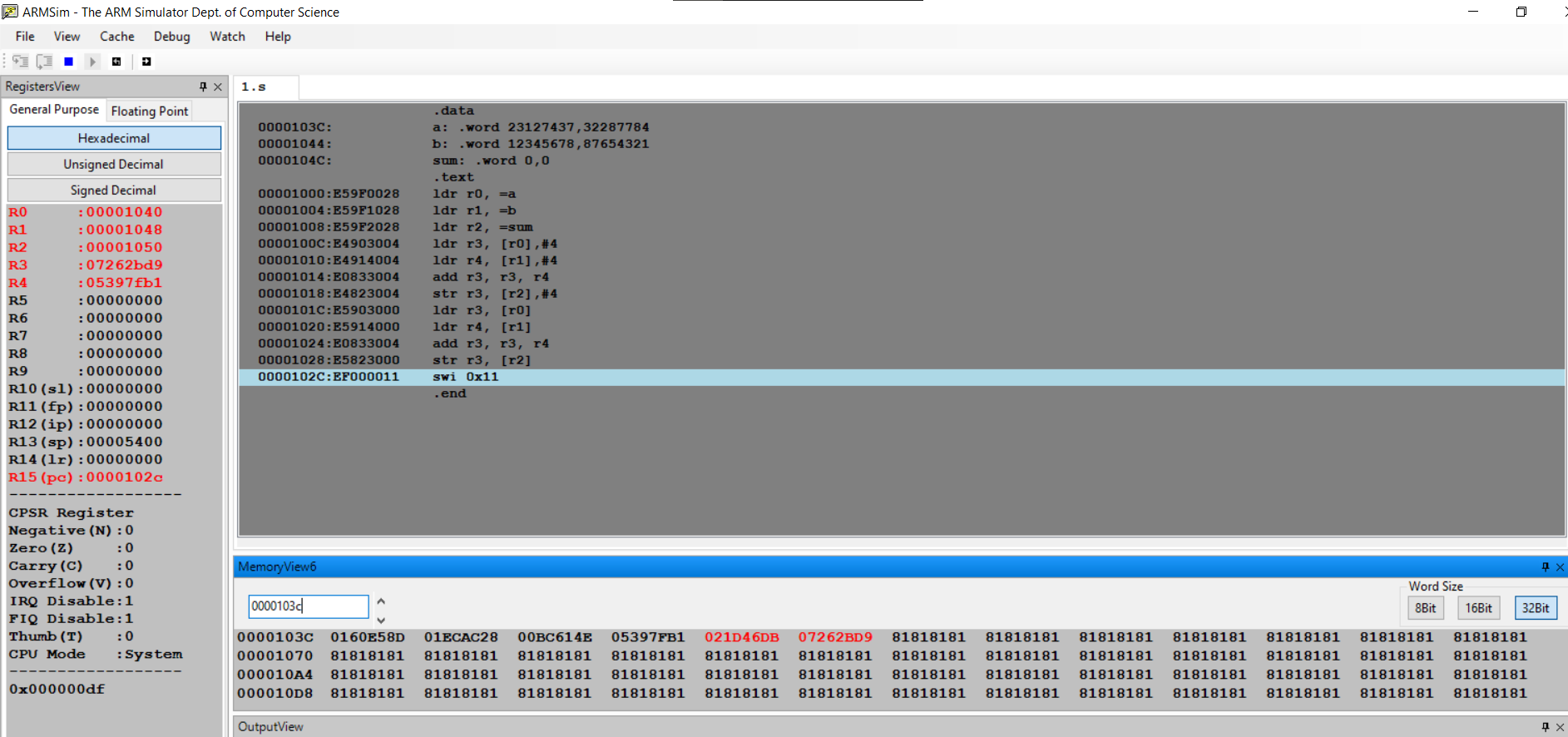
add r3, r3, r4

str r3, [r2]

swi 0x11

.end

II. Output Screen Shots (One)



**Microprocessor and Computer Architecture**

**UE21CS251B**

**4th Semester, Academic Year 2022-23**

Date:

|  |  |  |
| --- | --- | --- |
| Name: | SRN: | Section |

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

Title of the Program

**Write an ALP to find 1’s and 2’s complement of a 32-bit number**

I.ARM Assembly Code

.text

ldr r0,=0x12345678

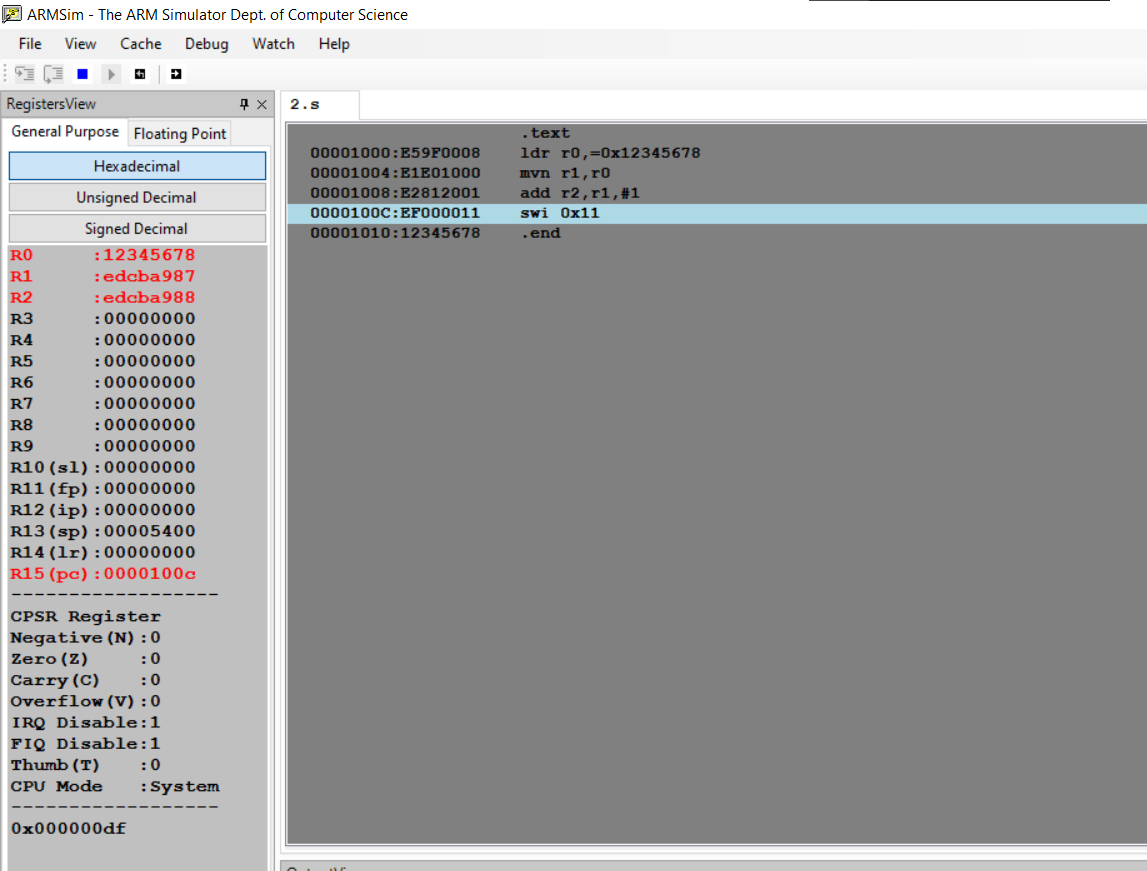
mvn r1,r0

add r2,r1,#1

swi 0x11

.end

II. Output Screen Shots (One)



**Microprocessor and Computer Architecture**

**UE21CS251B**

**4th Semester, Academic Year 2022-23**

Date:

|  |  |  |
| --- | --- | --- |
| Name: | SRN: | Section |

Week#\_\_\_\_4\_\_\_\_\_\_\_ Program Number: \_\_\_\_3\_\_

Title of the Program

**Write an ALP to scan a 32-bit number if it is negative or positive**

I.ARM Assembly Code

.text

mov r0,#-83

cmp r0,#0

bmi neg

b pos

neg:

mov r1,#1

b end

pos:

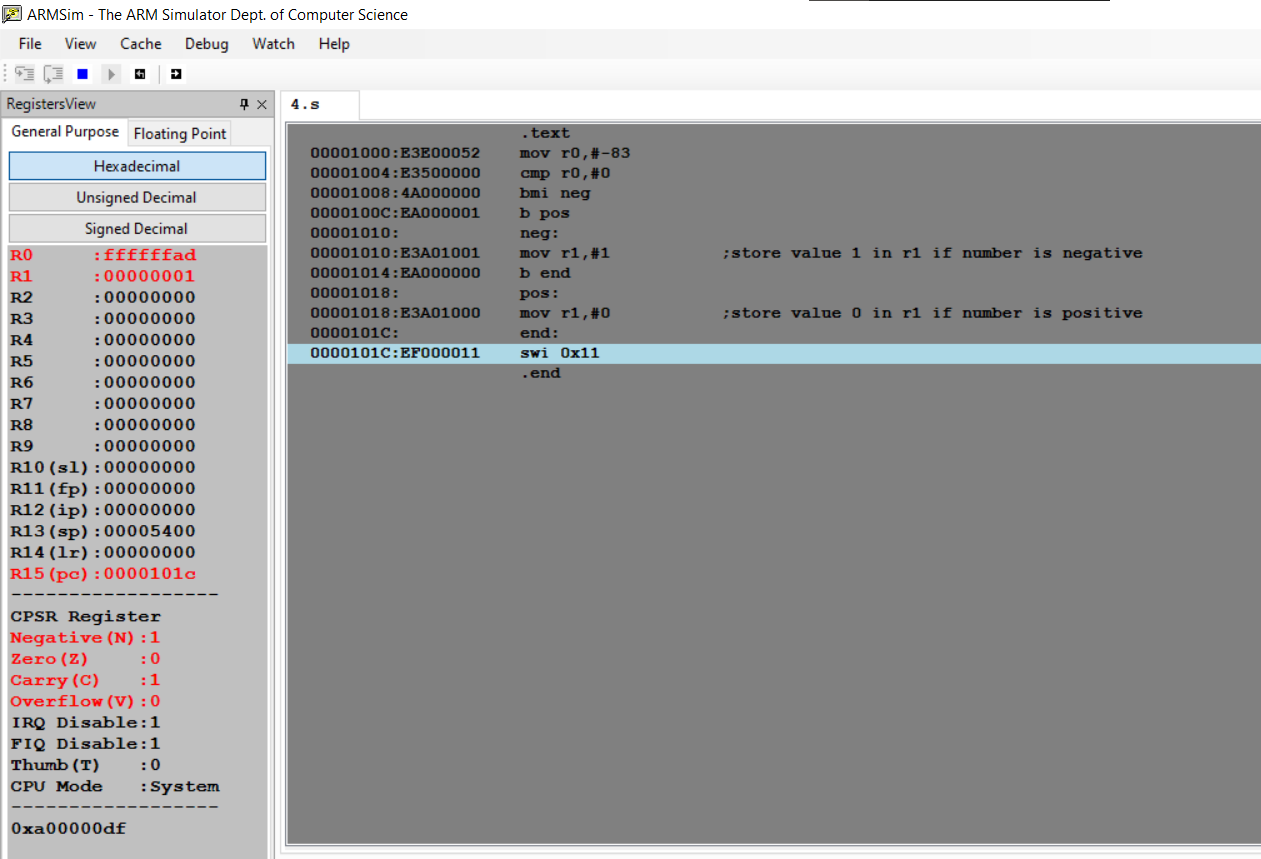
mov r1,#0

end:

swi 0x11

.end

II. Output Screen Shot (One)



**Microprocessor and Computer Architecture**

**UE21CS251B**

**4th Semester, Academic Year 2022-23**

Date:

|  |  |  |
| --- | --- | --- |
| Name: | SRN: | Section |

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

Title of the Program

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

I.ARM Assembly Code

.data

a: .word 42,7,0,-3,0,-54,23,-11,33

.text

ldr r0,=a

mov r1,#0 ;number of zeroes

mov r2,#0 ;number of positive numbers

mov r3,#0 ;number of negative numbers

mov r5,#9

loop:

ldr r4,[r0],#4

cmp r4,#0

beq zero

cmp r4,#0

bgt positive

blt negative

zero:

add r1,r1,#1

b next

positive:

add r2,r2,#1

b next

negative:

add r3,r3,#1

next:

sub r5,r5,#1

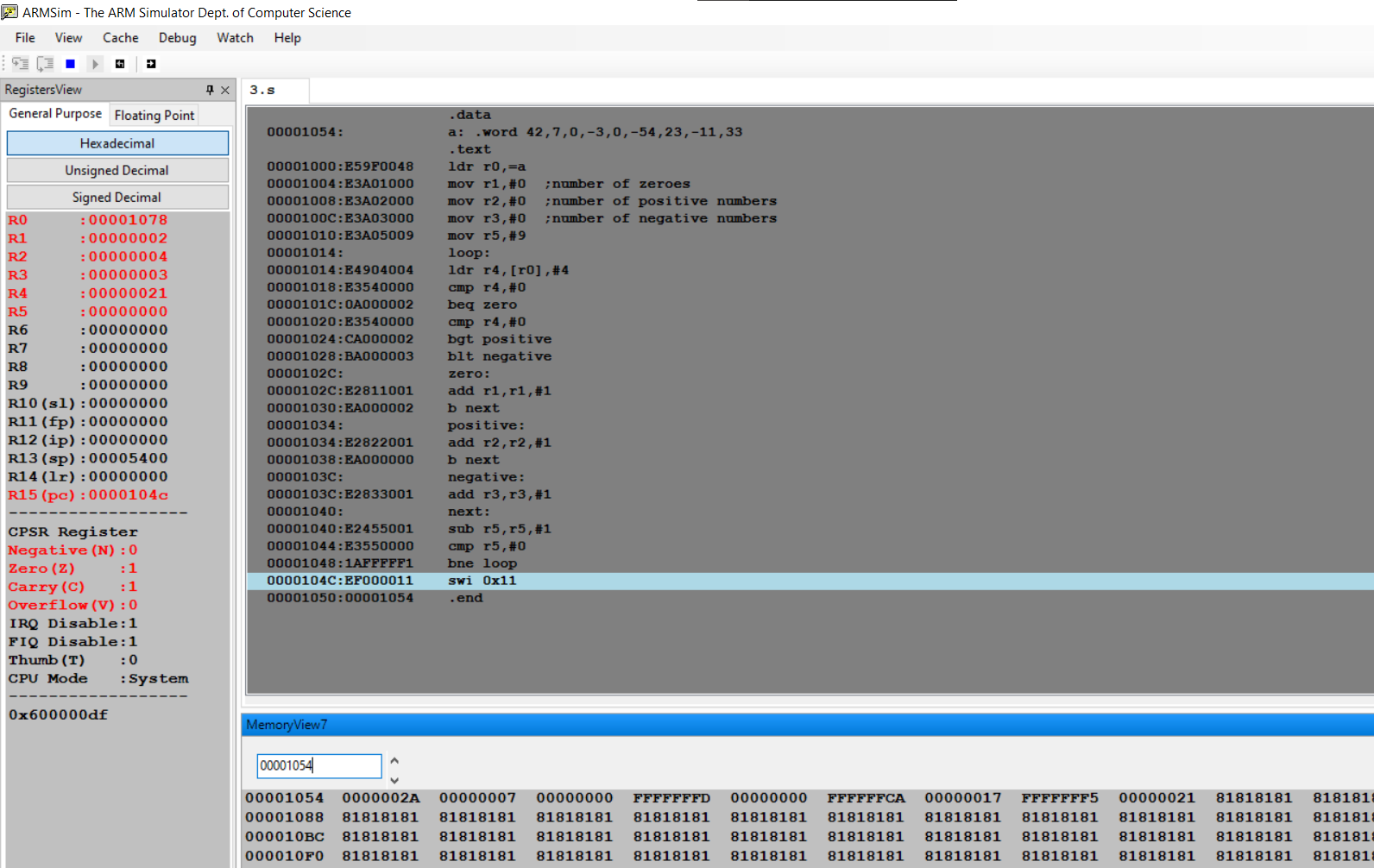
cmp r5,#0

bne loop

swi 0x11

.end

II. Output Screen Shot (One)



**Microprocessor and Computer Architecture**

**UE21CS251B**

**4th Semester, Academic Year 2022-23**

Date:

|  |  |  |
| --- | --- | --- |
| Name: | SRN: | Section |

Week#\_\_\_\_4\_\_\_\_\_\_\_ Program Number: \_\_\_\_5\_\_

Title of the Program

**Write an ALP to count the number of 1’s and 0’s in a given 32-bit number**

I.ARM Assembly Code

.text

ldr r0,=0b11100110010101001101100101101101

mov r1,#32

mov r2,#0 ;count of 1s

mov r3,#0 ;count of 0s

loop:

and r4,r0,#1

cmp r4,#1

beq one

b zero

one:

add r2,r2,#1

mov r0,r0,lsr #1

sub r1,r1,#1

cmp r1,#0

bne loop

swi 0x11

zero:

add r3,r3,#1

mov r0,r0,lsr #1

sub r1,r1,#1

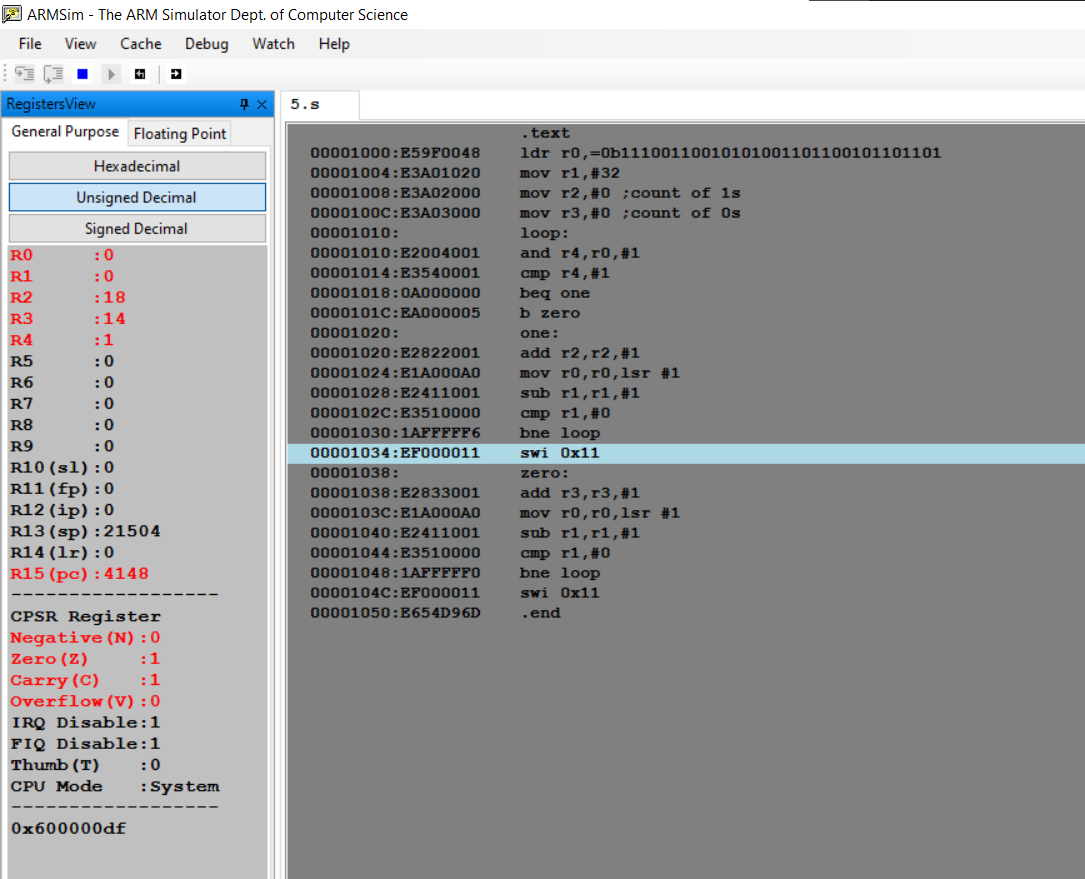
cmp r1,#0

bne loop

swi 0x11

.end

II. Output Screen Shot (One)

****

**Microprocessor and Computer Architecture**

**UE21CS251B**

**4th Semester, Academic Year 2022-23**

Date:

|  |  |  |
| --- | --- | --- |
| Name: | SRN: | Section |

Week#\_\_\_\_4\_\_\_\_\_\_\_ Program Number: \_\_\_\_6\_\_

Title of the Program

**Write an ALP to check the given number has odd or even number of 1’s and display the result. (Even Parity and Odd Parity)**

I.ARM Assembly Code

.data

o: .asciiz "Odd parity"

e: .asciiz "Even parity"

.text

ldr r5,=0b11100110010101001101100101100101

mov r1,#32

mov r2,#0 ;count of 1s

mov r3,#0 ;count of 0s

loop:

and r4,r5,#1

cmp r4,#1

beq one

b zero

one:

add r2,r2,#1

mov r5,r5,lsr #1

sub r1,r1,#1

cmp r1,#0

bne loop

b print

zero:

add r3,r3,#1

mov r5,r5,lsr #1

sub r1,r1,#1

cmp r1,#0

bne loop

print:

and r6,r2,#1

cmp r6,#1

beq odd

b even

odd:

ldr r0,=o

swi 0x02

swi 0x11

even:

ldr r0,=e

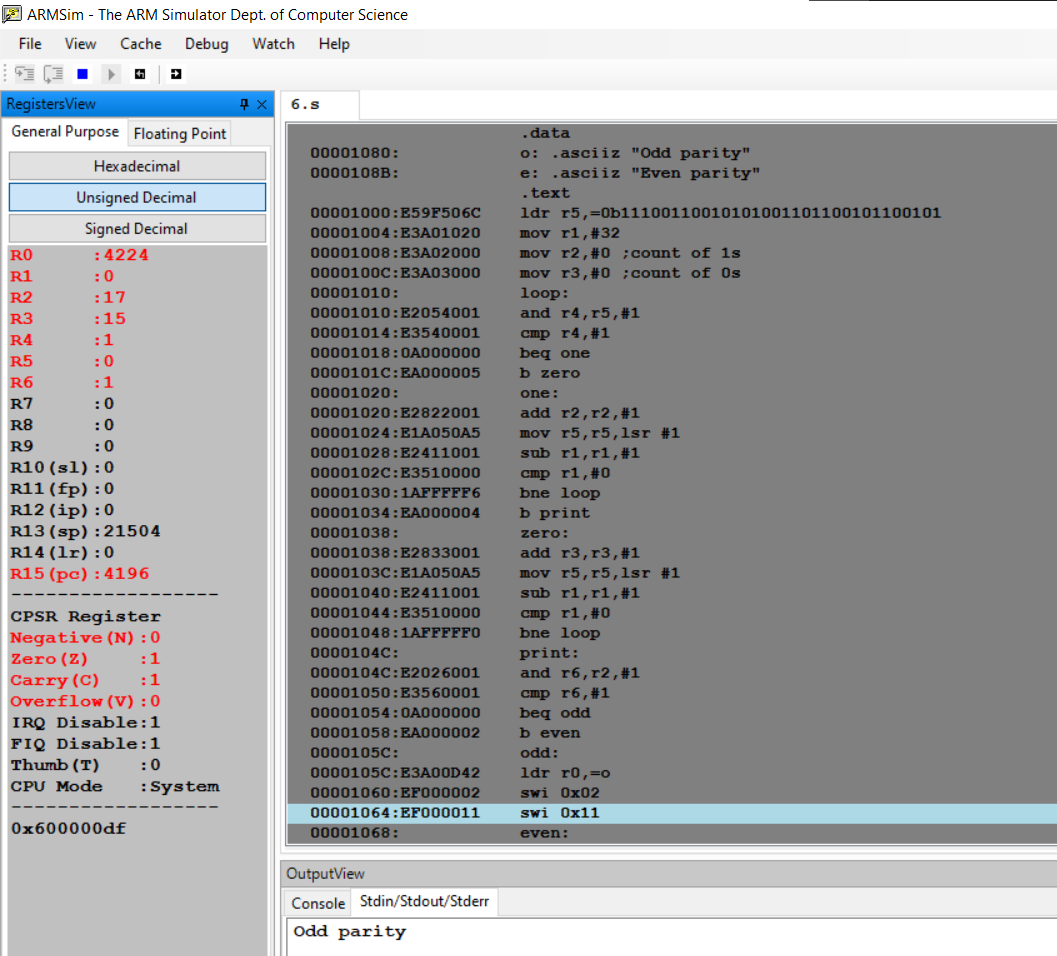
swi 0x02

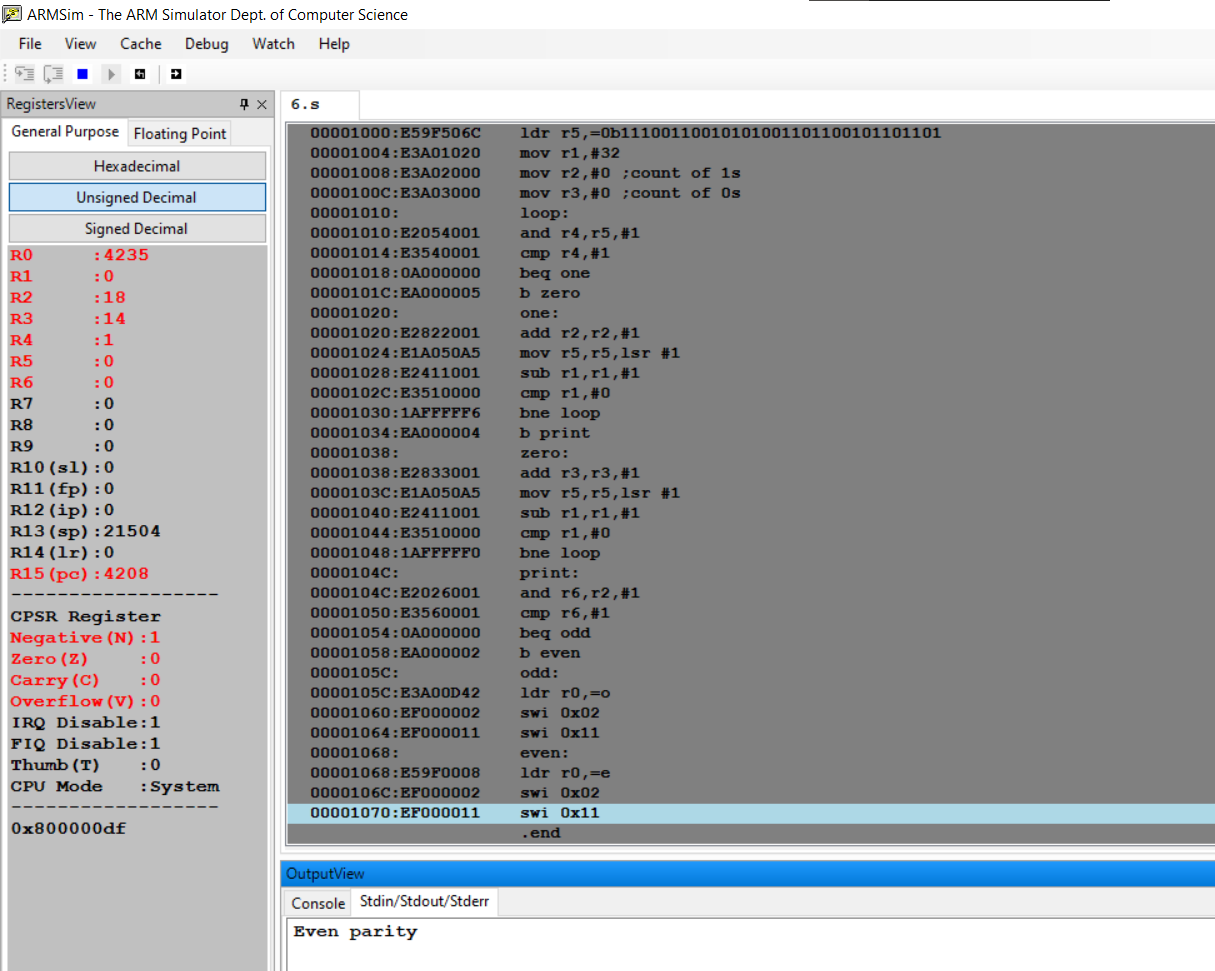
swi 0x11

.end

II. Output Screen Shot

(Two-One for odd Parity, one for even Parity)





**Microprocessor and Computer Architecture**

**UE21CS251B**

**4th Semester, Academic Year 2022-23**

Date:

|  |  |  |
| --- | --- | --- |
| Name: | SRN: | Section |

Week#\_\_\_\_2\_\_\_\_\_\_\_ Program Number: \_\_\_\_7\_\_

Title of the Program

**Write an ALP to perform multiplication using shift method (without using MUL)**

I.ARM Assembly Code

Code to multiply a number by 25

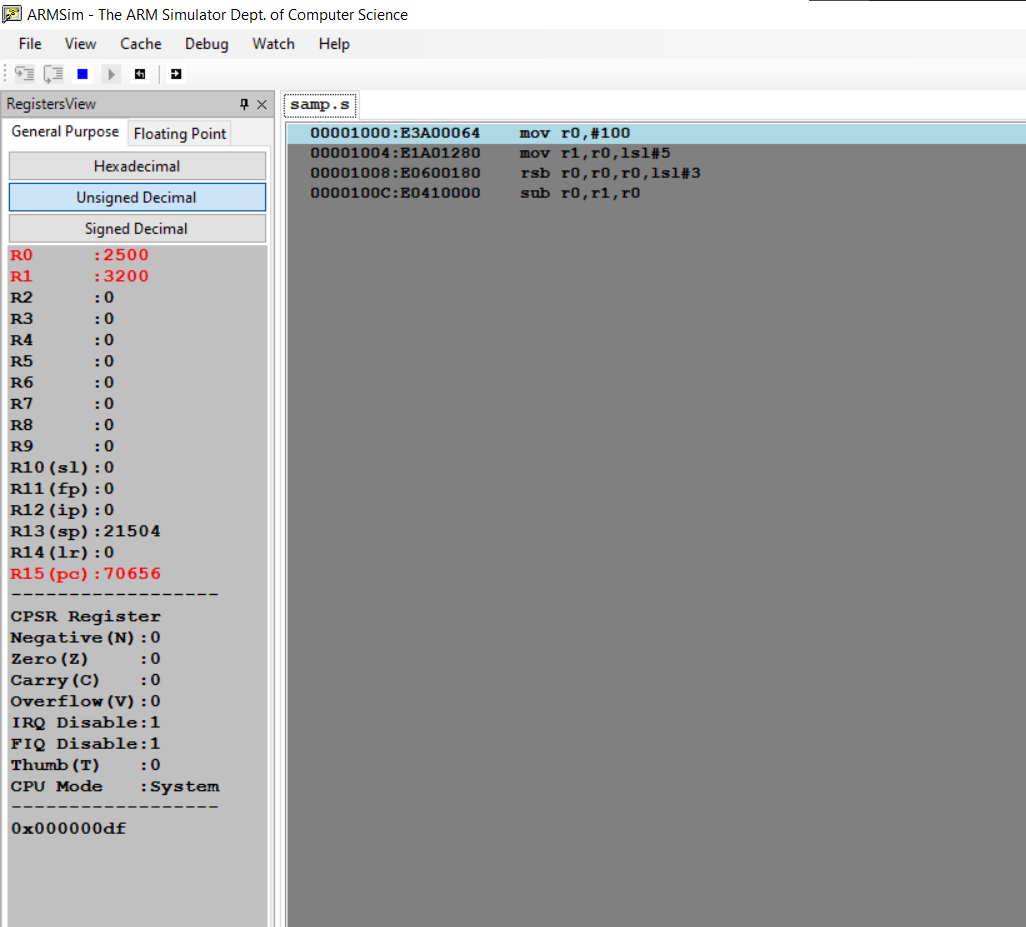
mov r0,#100

mov r1,r0,lsl#5

rsb r0,r0,r0,lsl#3

sub r0,r1,r0

II. Output Screen Shot (One)

****

**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 the external resource such as internet.
* If found plagiarized, I will abide with the disciplinary action of the University.

Signature:

Name:

SRN:

Section:

Date: