**Microprocessor and Computer Architecture**

**UE21CS251B**

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

Date:

|  |  |  |
| --- | --- | --- |
| Name: Nihal T M | SRN:PES2UG21CS333 | Section: F |

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.**

1. ARM Assembly Code

Code:

@ ALP program to add two 64-bit numbers

.data

a:.word 12312734,77823438

b:.word 24629982,56436738

c:.word 0,0

.text

ldr r1,=a

ldr r2,=b

ldr r3,=c

ldr r4,[r1],#4

ldr r5,[r2],#4

add r6,r4,r5

str r6,[r3],#4

ldr r4,[r1],#4

ldr r5,[r2],#4

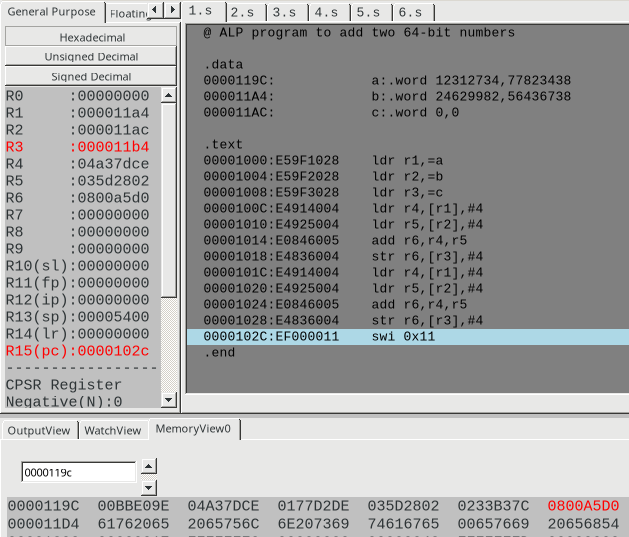
add r6,r4,r5

str r6,[r3],#4

swi 0x11

.end

II. Output Screen Shot (One)



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**

1. ARM Assembly Code

Code:

@ ALP program to find 1's complement and 2's complement of a 32-bit number

.data

num:.word 32

.text

ldr r1,=num

ldr r2,[r1]

@ this is 1's complement code

mvn r3,r2

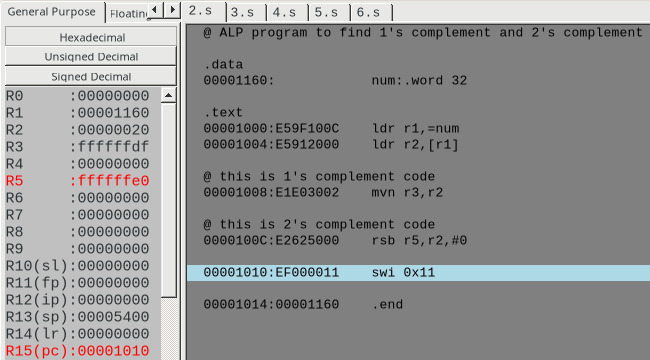
@ this is 2's complement code

rsb r5,r2,#0

swi 0x11

.end

II. Output Screen Shot (One)



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

Title of the Program

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

1. ARM Assembly Code

Code:

@ ALP program to scan a number and find if it is positive or negative

.data

num:.word 0

pos:.asciz "The value is positive"

neg:.asciz "The value is negative"

zer:.asciz "The value is zero"

.text

ldr r1,=num

ldr r2,[r1]

cmp r2,#0

beq zero

bpl positive

bmi negative

positive:

ldr r0,=pos

swi 0x02

b end

negative:

ldr r0,=neg

swi 0x02

b end

zero:

ldr r0,=zer

swi 0x02

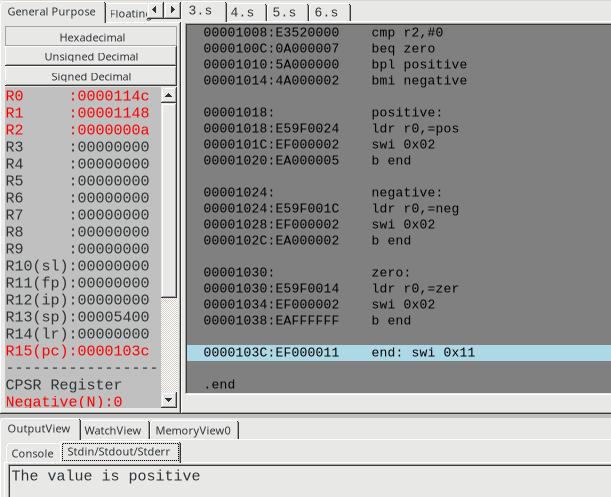
b end

end: swi 0x11

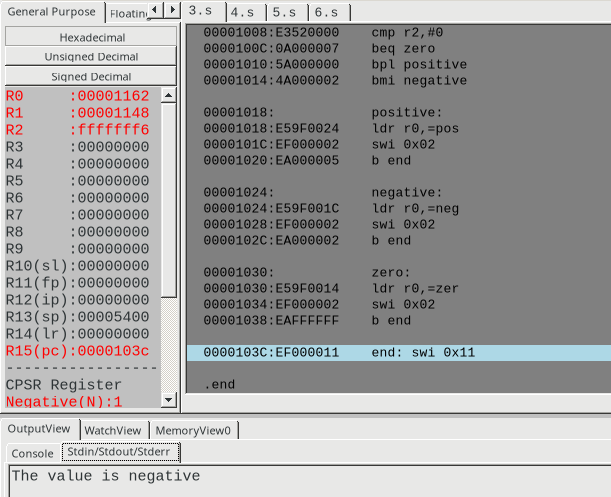
.end

II. Output Screen Shot (One)

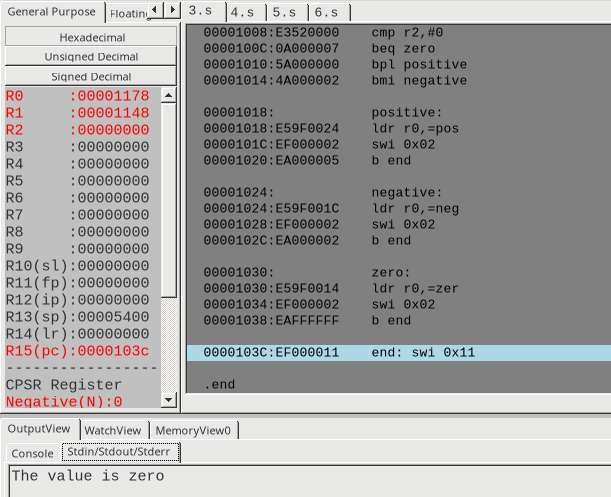
(i) when the value is positive:



(ii) when value is negative:



(iii) when the value is zero:



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**

1. ARM Assembly Code

Code:

@ ALP to find the number of zeroes, positive numbers and negative numbers in the array

.data

arr:.word 0,5,-4,0,23,-4,32,66,-3,0

.text

ldr r0,=arr

mov r1,#11

mov r2,#0 @ holds number of zeroes

mov r3,#0 @ holds number of positive numbers

mov r4,#0 @ holds number of negative numbers

loop:

ldr r5,[r0],#4

subs r1,r1,#1

beq end

cmp r5,#0

beq zero

bpl positive

bmi negative

zero:

add r2,r2,#1

b loop

positive:

add r3,r3,#1

b loop

negative:

add r4,r4,#1

b loop

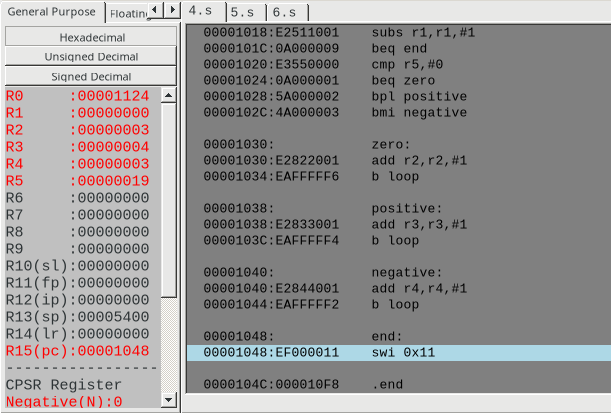
end:

swi 0x11

.end

II. Output Screen Shot (One)

here r2, r3, r4 holds the number of zeroes, positive numbers and negative numbers present in the arrys respectively:



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.**

1. ARM Assembly Code

Code:

@ ALP program to count the number of zeroes and ones in the binary number

.data

num:.word 25

.text

ldr r1,=num

ldr r2,[r1]

mov r3,#0 @ holds the number of zeroes in the no.

mov r4,#0 @ holds the number of ones in the no.

loop:and r5,r2,#1

cmp r2,#0

beq end

mov r2,r2,LSR #1

cmp r5,#0

beq zero

b one

zero:

add r3,r3,#1

b loop

one:

add r4,r4,#1

b loop

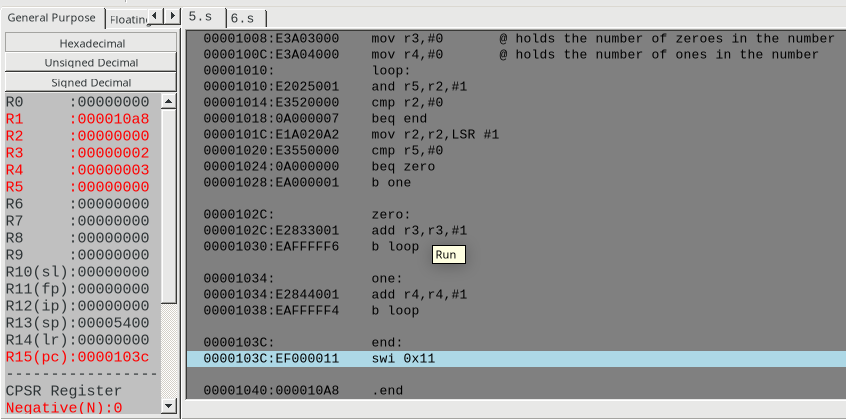
end:

swi 0x11

.end

II. Output Screen Shot (One)

here r3 and r4 hold the number of zeroes and ones present in the binary number respectively.(excluding trailing left zeroes as they are not important)



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)**

1. ARM Assembly Code

Code:

@ ALP program to check if a given number has odd or even number of ones in binary

.data

num:.word 25

even:.asciz "Even Parity"

odd:.asciz "Odd Parity"

.text

ldr r1,=num

ldr r2,[r1]

mov r3,#0 @ number of ones in the binary format of the given number

loop:

and r5,r2,#1

cmp r2,#0

beq print

mov r2,r2,LSR #1

cmp r5,#1

beq ones

b loop

ones:

add r3,r3,#1

b loop

print:

ands r6,r3,#1

beq is\_even

b is\_odd

is\_odd:

ldr r0,=odd

swi 0x02

b end

is\_even:

ldr r0,=even

swi 0x02

swi 0x11

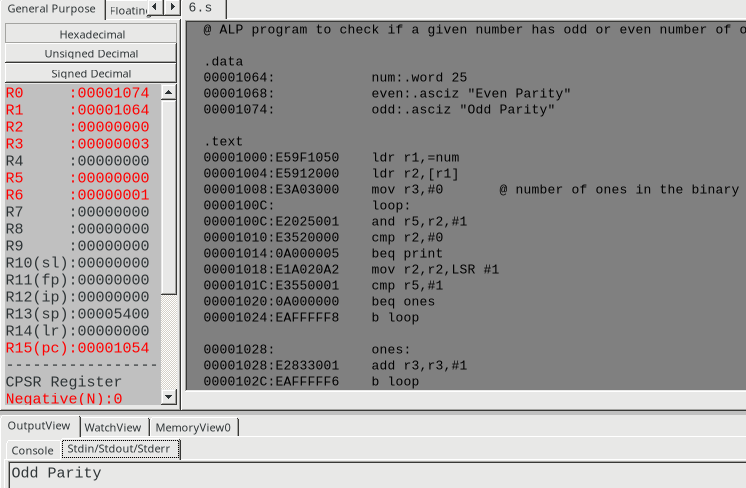
end:

swi 0x11

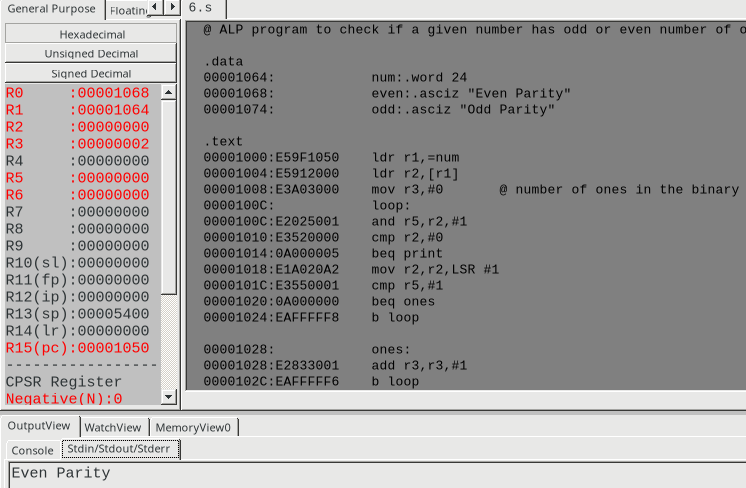
.end

II. Output Screen Shot (One)

(i) when it is odd parity:



(ii) when it is even parity:



**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: Nihal T M

SRN: PES2UG21CS333

Section: F

Date: 11/02/2023