Microprocessor and Computer Architecture Laboratory

UE19CS256

4th Semester, Academic Year 2020-21

Date:

Name: Ramya N Prabhu	SRN:PES1UG19CS380	Section F

Week#____1_____Program Number: ____1___

Title of the Program

Write an ALP using ARM instruction set to add and subtract two 32 bit numbers .Both numbers are in registers.

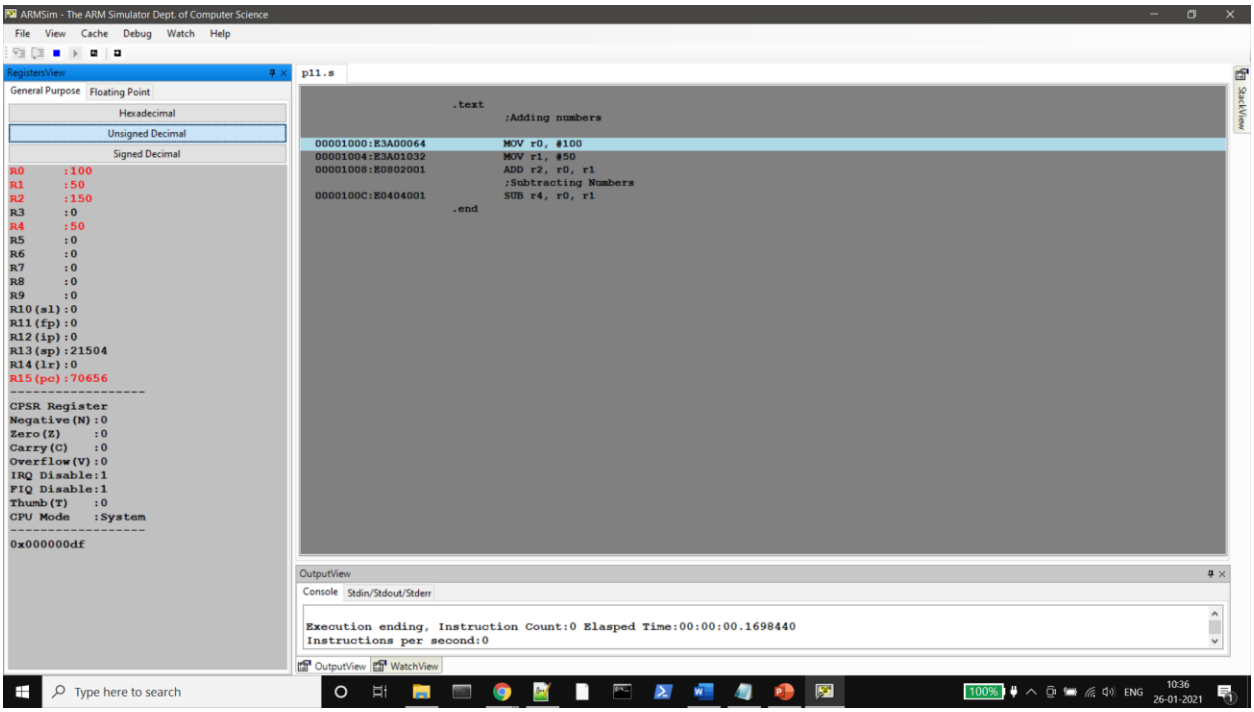
I. ARM Assembly Code for each program:

```
.text
    ;Adding numbers
    MOV r0, #100
    MOV r1, #50
    ADD r2, r0, r1
    ;Subtracting Numbers
```

SUB r4, r0, r1

.end

II. Final Output Screen Shot (Register Window, Output window)



Microprocessor and Computer Architecture Laboratory

UE19CS256

4th Semester, Academic Year 2020-21

Date:

Name: Ramya N Prabhu	SRN:PES1UG19CS380	Section F

Week#____1_____ Program Number: ____2___

Title of the Program

Write an ALP to demonstrate logical operations. All operands are in registers.

I. ARM Assembly Code for each program

```
.text
;AND
MOV r0, #9
MOV r1, #5
AND r2, r0, r1
;OR
ORR r3, r0, r1
;NOT
```

MVN r4, r1

;XOR

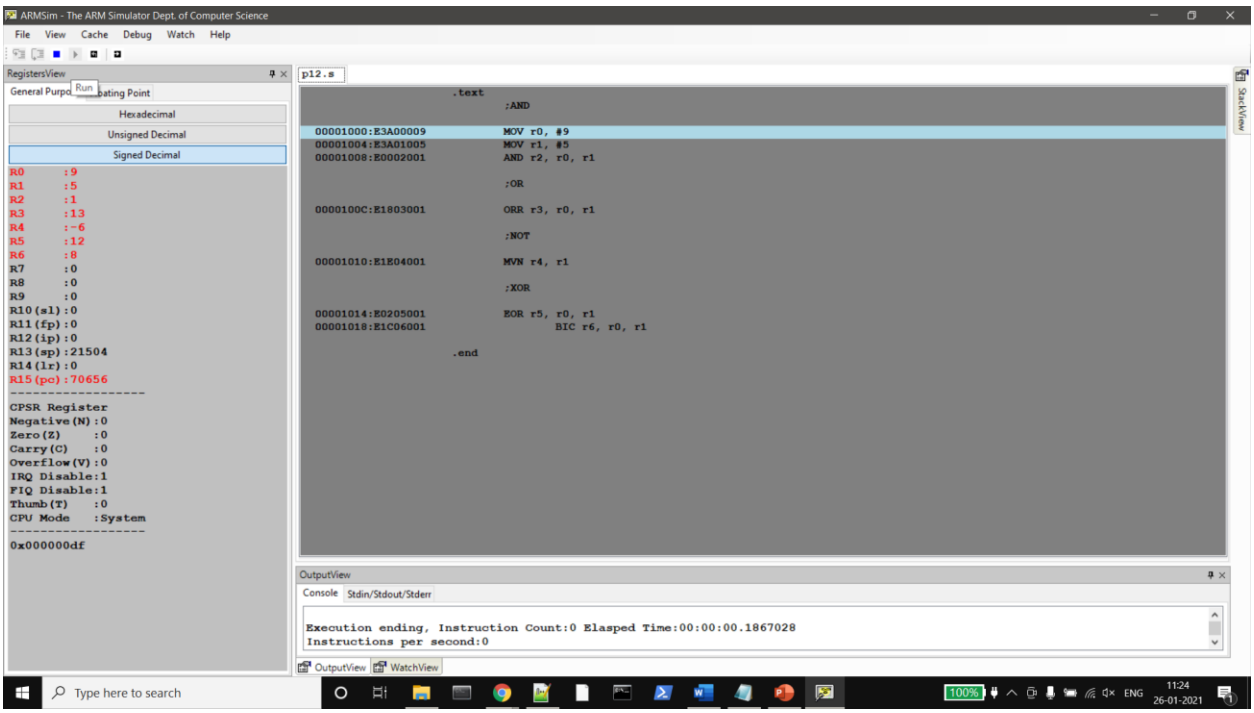
EOR r5, r0, r1

;AND NOT

BIC r6, r0, r1

.end

II. Final Output Screen Shot (Register Window, Output window)



Microprocessor and Computer Architecture Laboratory

UE19CS256

4th Semester, Academic Year 2020-21

Date:

Name: Ramya N Prabhu	SRN:PES1UG19CS380	Section F

Week#____1_____ Program Number: ____3___

Title of the Program

Write an ALP to add 5 numbers where values are present in registers.

I. ARM Assembly Code for each program

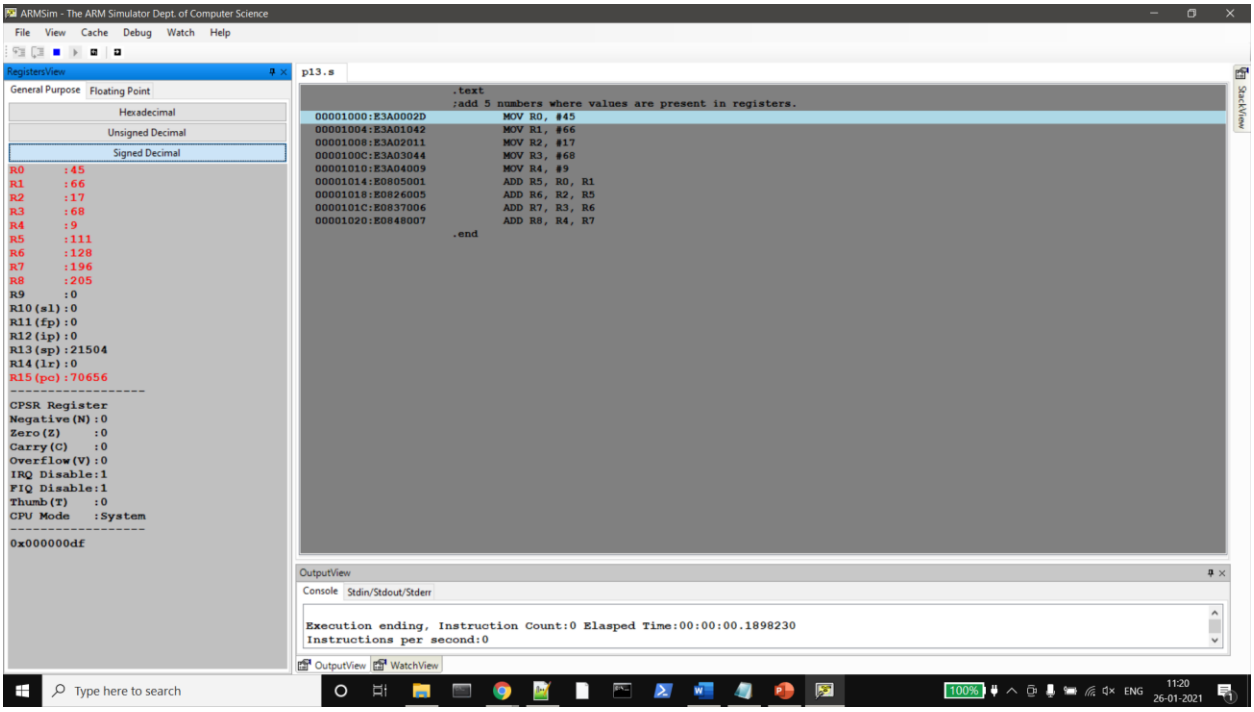
```
.text
;add 5 numbers where values are present in registers.
    MOV R0, #45
    MOV R1, #66
    MOV R2, #17
    MOV R3, #68
    MOV R4, #9
    ADD R5, R0, R1
    ADD R6, R2, R5
```

ADD R7, R3, R6

ADD R8, R4, R7

.end

II. Final Output Screen Shot (Register Window, Output window)



Microprocessor and Computer Architecture Laboratory

UE19CS256

4th Semester, Academic Year 2020-21

Date:

Name: Ramya N Prabhu	SRN:PES1UG19CS380	Section F

Week#____1_____ Program Number: ____4___

Title of the Program

Write an ALP using ARM instruction set to check if a number stored in a register is even or odd. If even, store 00 in R0, else store FF in R0

I. ARM Assembly Code for each program

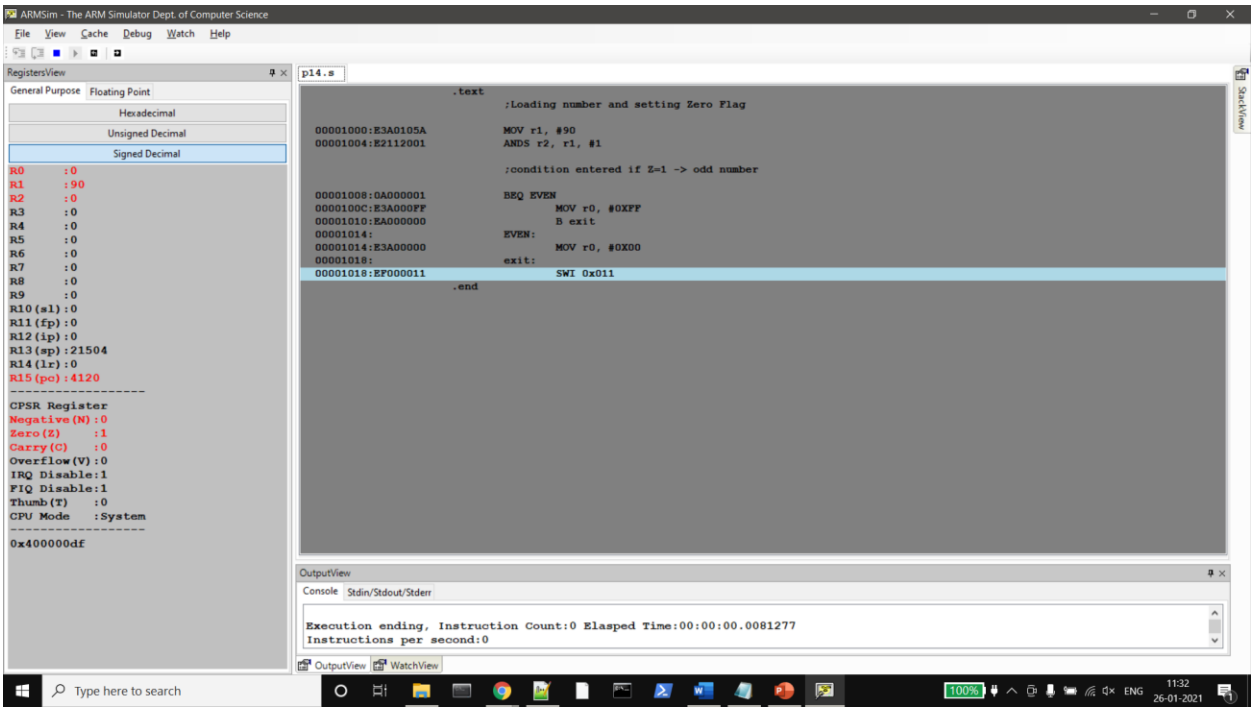
```
;case where num is even
.text
    MOV r1, #90
    ANDS r2, r1, #1
    BEQ EVEN
        MOV r0, #0XFF
        B exit
```

```
    EVEN:
        MOV r0, #0X00
    exit:
        SWI 0x011
.end
```

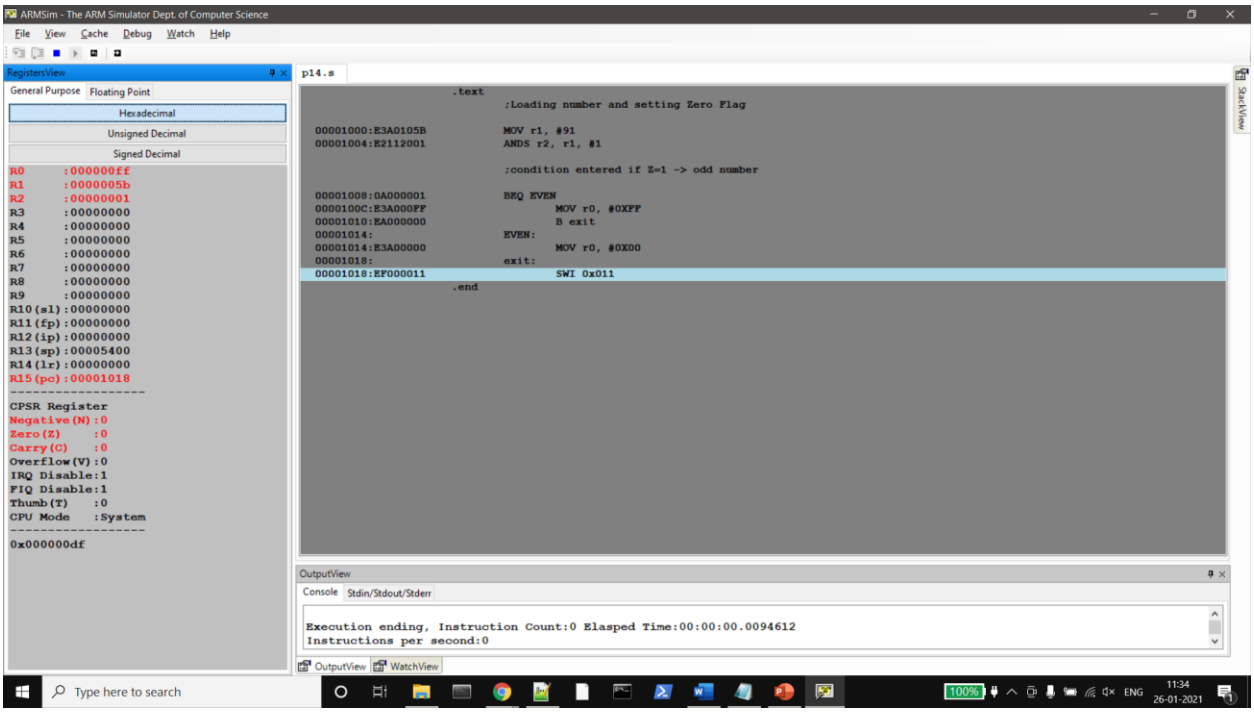
```
;case where the num is odd
.text
    MOV r1, #91
    ANDS r2, r1, #1
    BEQ EVEN
        MOV r0, #0XFF
        B exit
    EVEN:
        MOV r0, #0X00
    exit:
        SWI 0x011
.end
```

II. Final Output Screen Shot (Register Window, Output window)

Case even:


ARMSim - The ARM Simulator Dept. of Computer Science
File View Cache Debug Watch Help
RegistersView
General Purpose Floating Point
Hexadecimal
Unsigned Decimal
Signed Decimal
R0 :0
R1 :90
R2 :0
R3 :0
R4 :0
R5 :0
R6 :0
R7 :0
R8 :0
R9 :0
R10(sl):0
R11(fp):0
R12(ip):0
R13(sp):21504
R14(lr):0
R15(pc):4120
CPSR Register
Negative(N):0
Zero(Z) :1
Carry(C) :0
Overflow(V):0
IRQ Disable:1
FIQ Disable:1
Thumb(T) :0
CPU Mode :System
0x400000df
p14.s
.text
;Loading number and setting Zero Flag
00001000:E3A0105A MOV r1, #90
00001004:E2112001 ANDS r2, r1, #1
;condition entered if Z=1 -> odd number
00001008:0A000001 BEQ EVEN
0000100C:E3A000FF MOV r0, #0XFF
00001010:EA000000 B exit
00001014: EVEN:
00001014:E3A00000 MOV r0, #0X00
00001018: exit:
00001018:EF000011 SWI 0x011
.end
OutputView
Console Stdin/Stdout/Stderr
Execution ending, Instruction Count:0 Elasped Time:00:00:00.0081277
Instructions per second:0
OutputView WatchView
Type here to search
100%
ENG
11:32
26-01-2021

Case odd:


ARMSim - The ARM Simulator Dept. of Computer Science
File View Cache Debug Watch Help
RegistersView
General Purpose Floating Point
Hexadecimal
Unsigned Decimal
Signed Decimal
R0 :000000ff
R1 :0000005b
R2 :00000001
R3 :00000000
R4 :00000000
R5 :00000000
R6 :00000000
R7 :00000000
R8 :00000000
R9 :00000000
R10(sl):00000000
R11(fp):00000000
R12(ip):00000000
R13(sp):00005400
R14(lr):00000000
R15(pc):00001018
CPSR Register
Negative(N):0
Zero(Z) :0
Carry(C) :0
Overflow(V):0
IRQ Disable:1
FIQ Disable:1
Thumb(T) :0
CPU Mode :System
0x000000df
p14.s
.text
;Loading number and setting Zero Flag
00001000:E3A0105B MOV r1, #91
00001004:E2112001 ANDS r2, r1, #1
;condition entered if Z=1 -> odd number
00001008:0A000001 BEQ EVEN
0000100C:E3A000FF MOV r0, #0XFF
00001010:EA000000 B exit
00001014: EVEN:
00001014:E3A00000 MOV r0, #0X00
00001018: exit:
00001018:EF000011 SWI 0x011
.end
OutputView
Console Stdin/Stdout/Stderr
Execution ending, Instruction Count:0 Elasped Time:00:00:00.0094612
Instructions per second:0
OutputView WatchView
Type here to search
100%
ENG
11:34
26-01-2021

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:[Ramya]
Name: Ramya N Prabhu
SRN:PES1UG19CS380
Section: F
Date:26/1/2021