Microprocessor and Computer Architecture Laboratory

Subject Code :UE19CS256

4th Semester, Academic Year 2020-21

Date: 25/01/2021

Name: PRIYA MOHATA	SRN:PES2UG19CS301	Section:E

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

TEST CASE 1

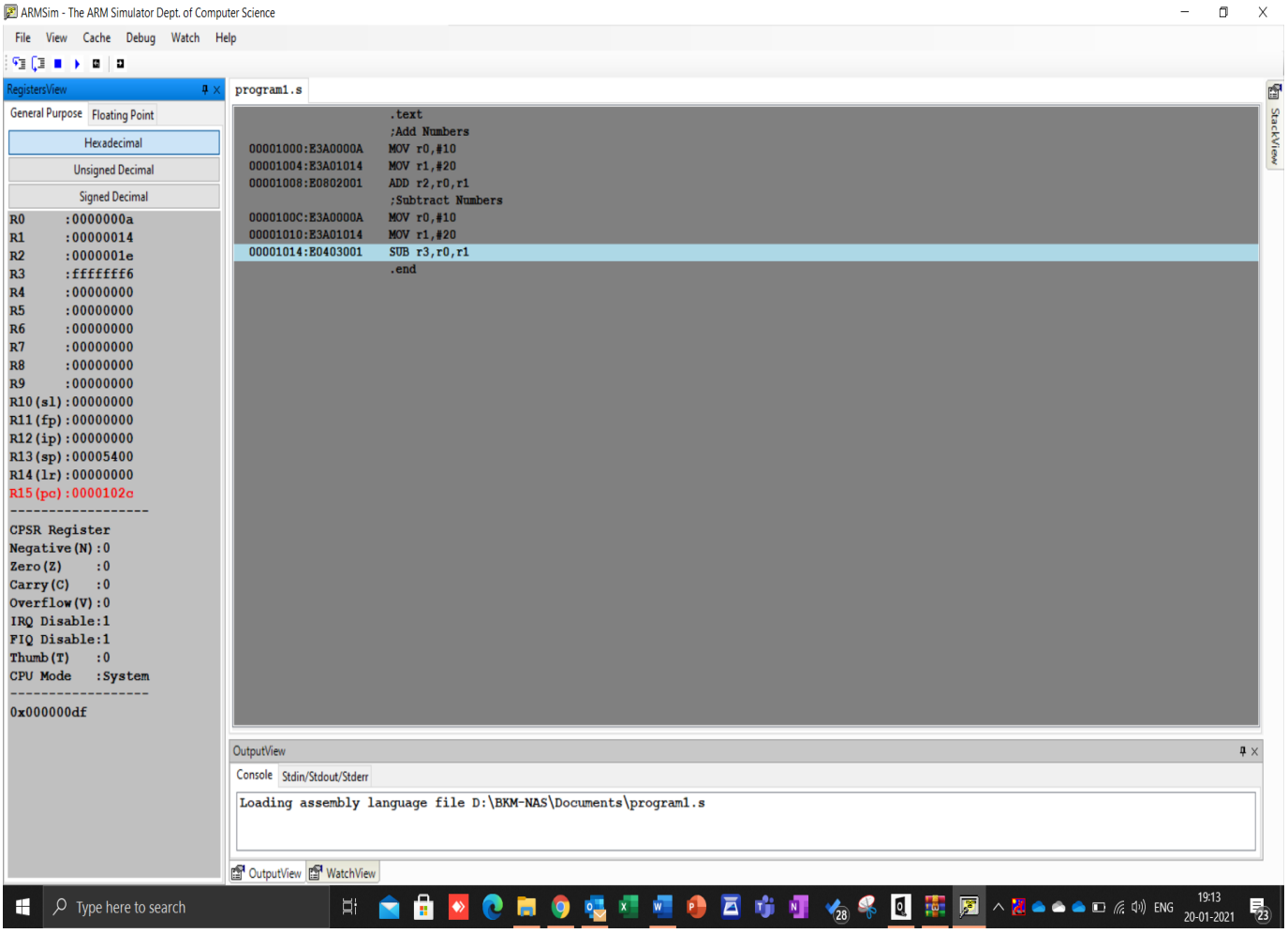
```
.text
;Add Numbers
MOV r0,#10
MOV r1,#20
ADD r2,r0,r1
;Subtract Numbers
MOV r0,#10
MOV r1,#20
SUB r3,r0,r1
.end
```

TEST CASE 2

```
.text
;Add Numbers
MOV r0,#45
MOV r1,#20
ADD r2,r0,r1
;Subtract Numbers
MOV r0,#45
MOV r1,#20
SUB r3,r0,r1
.end
```

2.Output screenshot(Register window,Output window)

Test Case 1: (given)



Test case 2

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      :0000002d
R1      :00000014
R2      :00000041
R3      :00000019
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
```

programla.s

```
                    .text
                    ;Add Numbers
00001000:E3A0002D   MOV r0,#45
00001004:E3A01014   MOV r1,#20
00001008:E0802001   ADD r2,r0,r1
                    ;Subtract Numbers
0000100C:E3A0002D   MOV r0,#45
00001010:E3A01014   MOV r1,#20
00001014:E0403001   SUB r3,r0,r1
                    .end
```

OutputView

Console | Stdin/Stdout/Stderr

```
Loading assembly language file D:\BKM-NAS\Documents\programla.s
```

OutputView | WatchView

Type here to search

ENG 19:20 01-2021

III.Output Table for each program

TEST CASE 1 **R0=10=Hex 0A. R1=20=Hex 14** **After Addition R2=30=Hex 1E** **After Subtraction R2 = 10 = Hex 0xfffffff6 or – 0x0A**			
R0	R1	Arithmetic Operation	Result
0x0A	0x14	ADD	R0 =0x1E
0x0A	0x14	SUBTRACT	R0=0xfffffff6 (**– 0x0A)**

TEST CASE 2 **R0=45=Hex 2d. R1=20=Hex 14** **After Addition R2=65=Hex 41** **After Subtraction R2 = 25= Hex 19**			
R0	R1	Arithmetic Operation	Result
0x2d	0x14	ADD	R0 =0x41

0x2d	0x14	SUBTRACT	R0=0x19

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

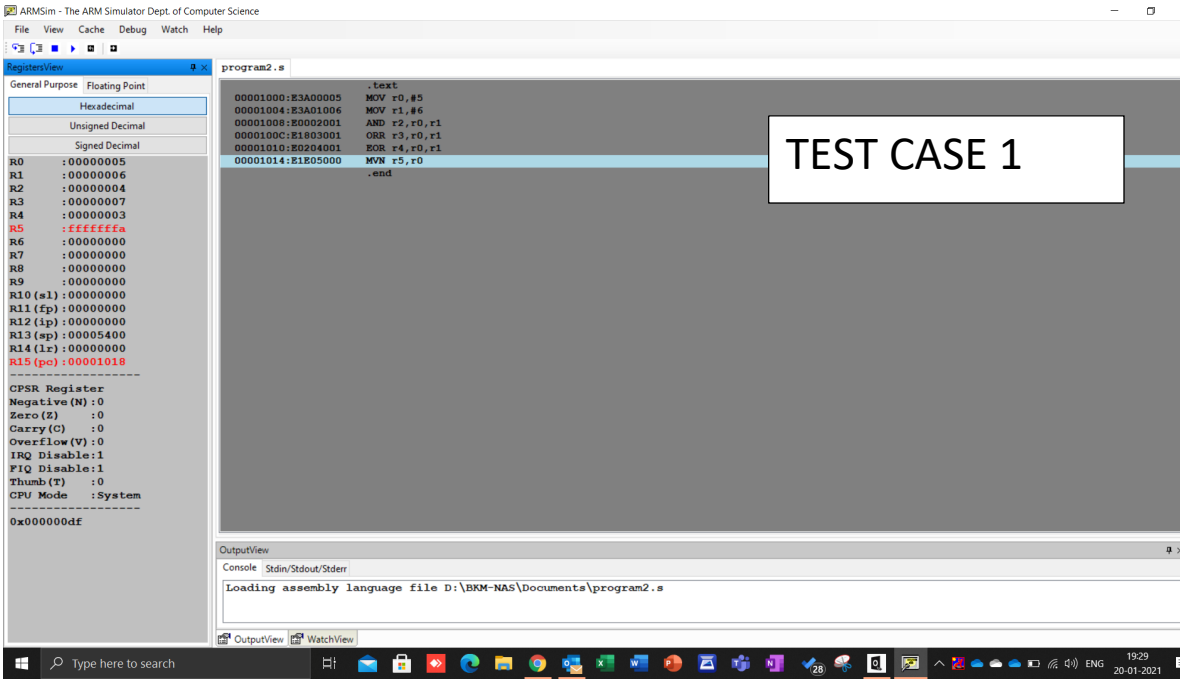
TEST CASE 1

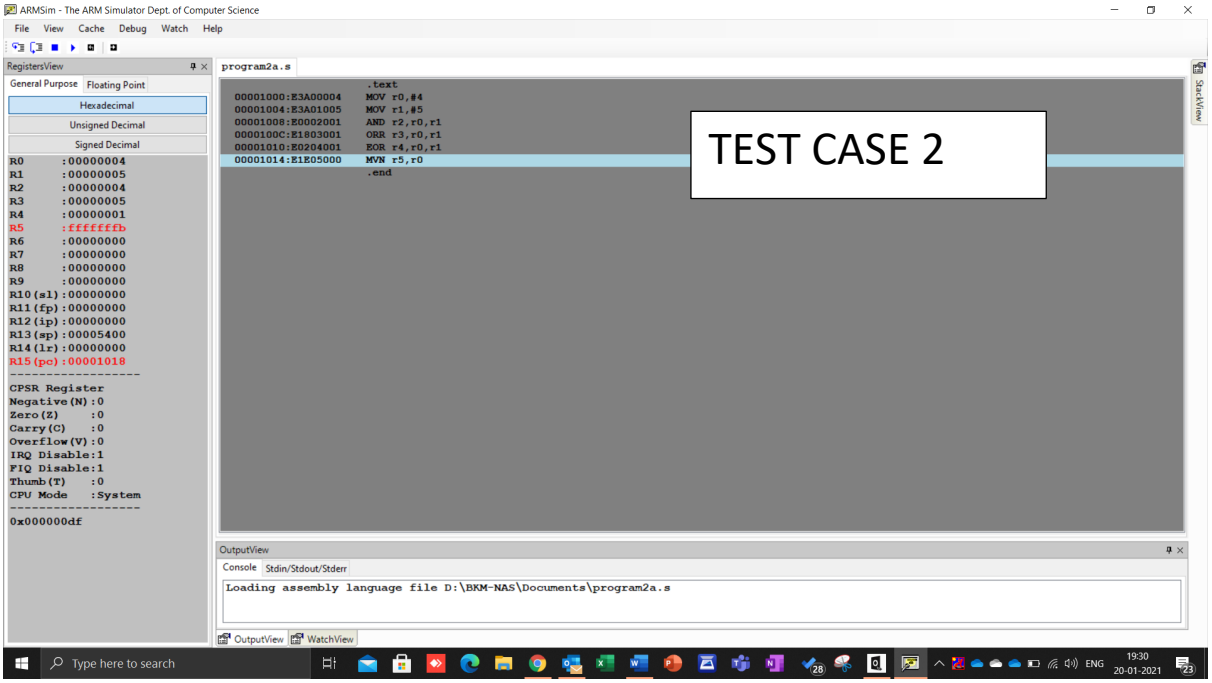
```
.text
MOV r0,#5
MOV r1,#6
AND r2,r0,r1
ORR r3,r0,r1
EOR r4,r0,r1
MVN r5,r0
.end
```

```
.text
MOV r0,#4
MOV r1,#5
AND r2,r0,r1
ORR r3,r0,r1
EOR r4,r0,r1
MVN r5,r0
.end
```

TEST CASE 2

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





II. Output table for each program

TEST CASE 1:

R0	R1	Logical Operation	Instruction	Result
0x05	0x06	AND	AND	R2=0x04
0x05	0x06	OR	ORR	R3=0x07
0x05	0x06	EX-OR	EOR	R4=0x03
0x05		NOT	MVN	R5=0xfffffffa

TEST CASE 2:

R0	R1	Logical Operation	Instruction	Result
0x04	0x05	AND	AND	R2=0x04
0x04	0x05	OR	ORR	R3=0x05
0x04	0x05	EX-OR	EOR	R4=0x01
0x05		NOT	MVN	R5=0xfffffffb

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

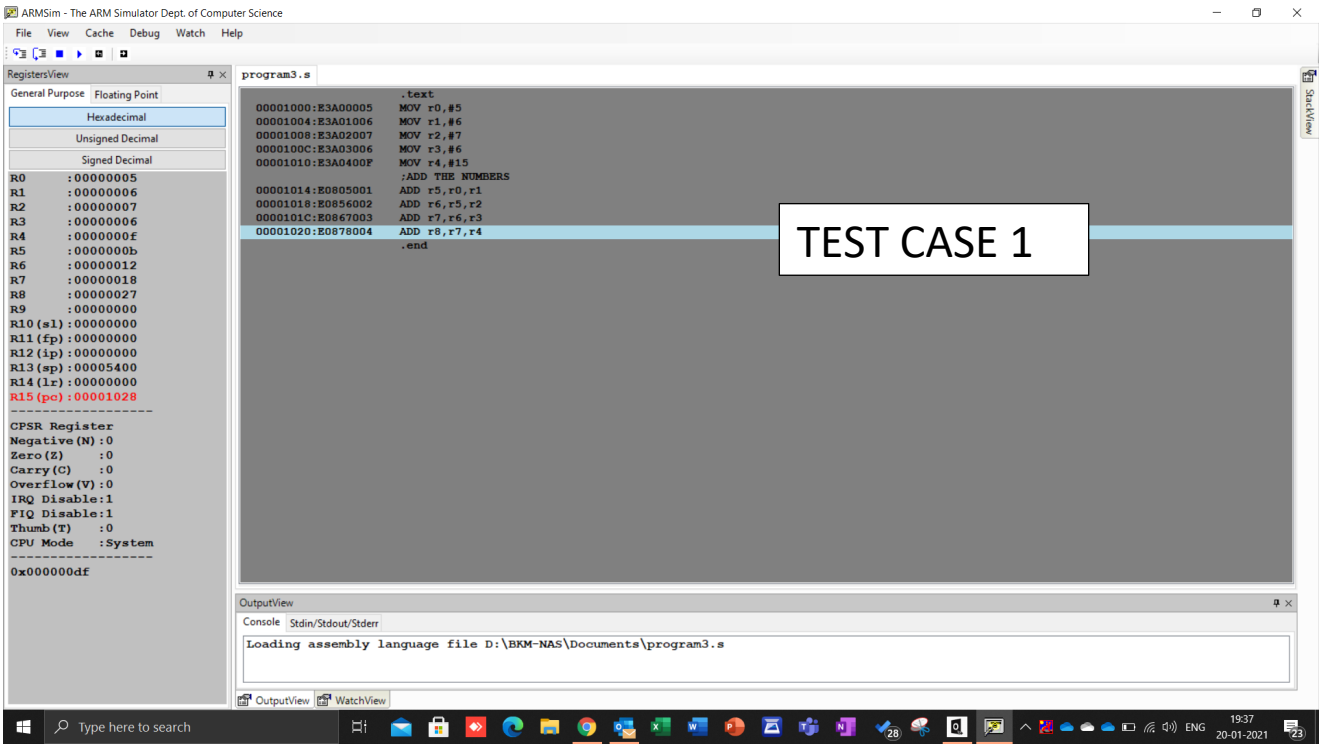
```
.text
MOV r0,#5
MOV r1,#6
MOV r2,#7
MOV r3,#6
MOV r4,#15
;ADD THE NUMBERS
ADD r5,r0,r1
ADD r6,r5,r2
ADD r7,r6,r3
ADD r8,r7,r4
.end
```

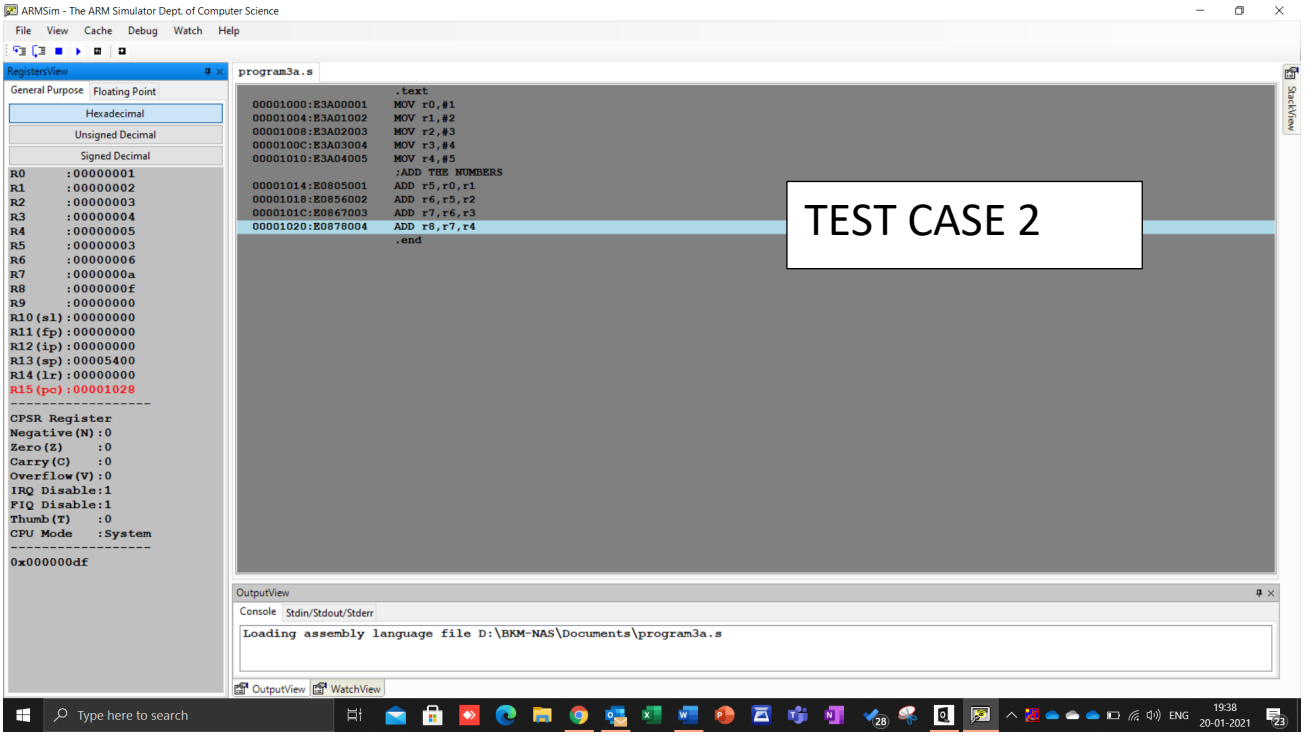
TEST CASE 1

```
.text
MOV r0,#1
MOV r1,#2
MOV r2,#3
MOV r3,#4
MOV r4,#5
;ADD THE NUMBERS
ADD r5,r0,r1
ADD r6,r5,r2
ADD r7,r6,r3
ADD r8,r7,r4
.end
```

TEST CASE 2

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





III. Output table for each program

REGISTERS		VALUE
R0		0x05
R1		0x06
R2		0x07
R3		0x06
R4		0x0f
R5	R0+R1	0x0b
R6	R5+R2	0x12

R7	R6+R3	0x18
R8	R7+R4	0x27

REGISTERS		VALUE
R0		0x01
R1		0x02
R2		0x03
R3		0x04
R4		0x05
R5	R0+R1	0x03
R6	R5+R2	0x06
R7	R6+R3	0x0a
R8	R7+R4	0x0f

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

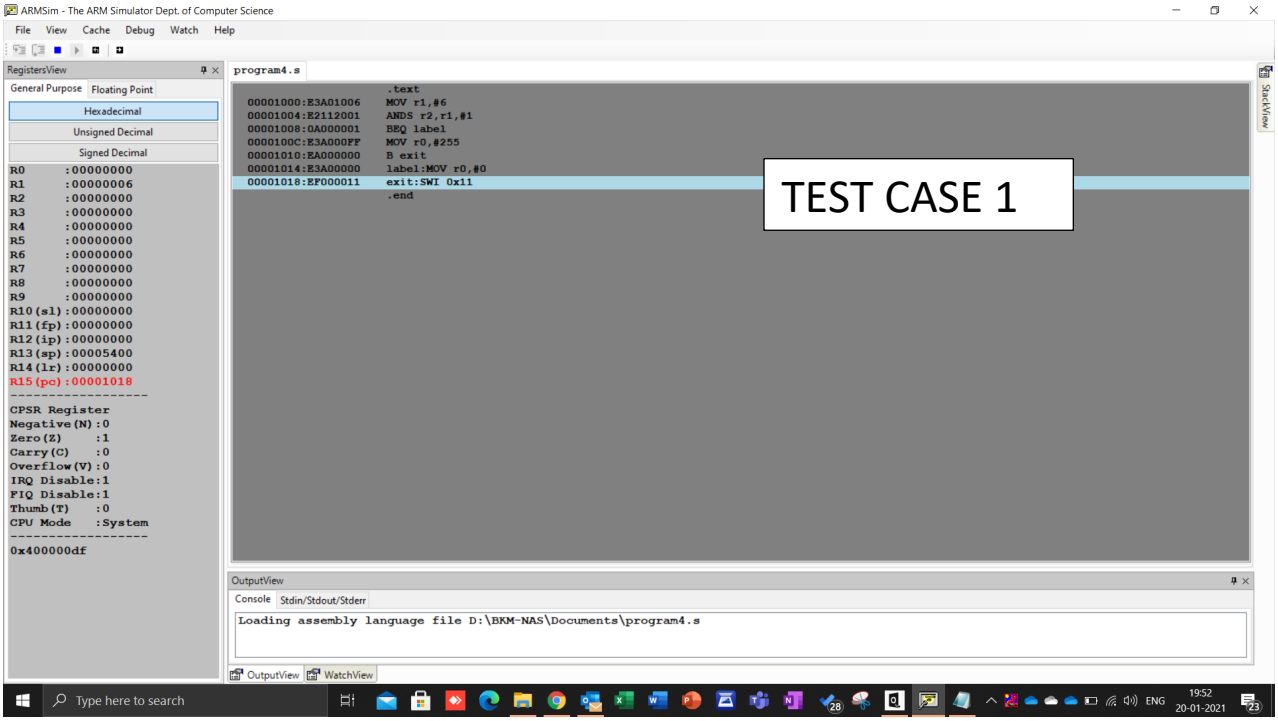
TEST CASE 1

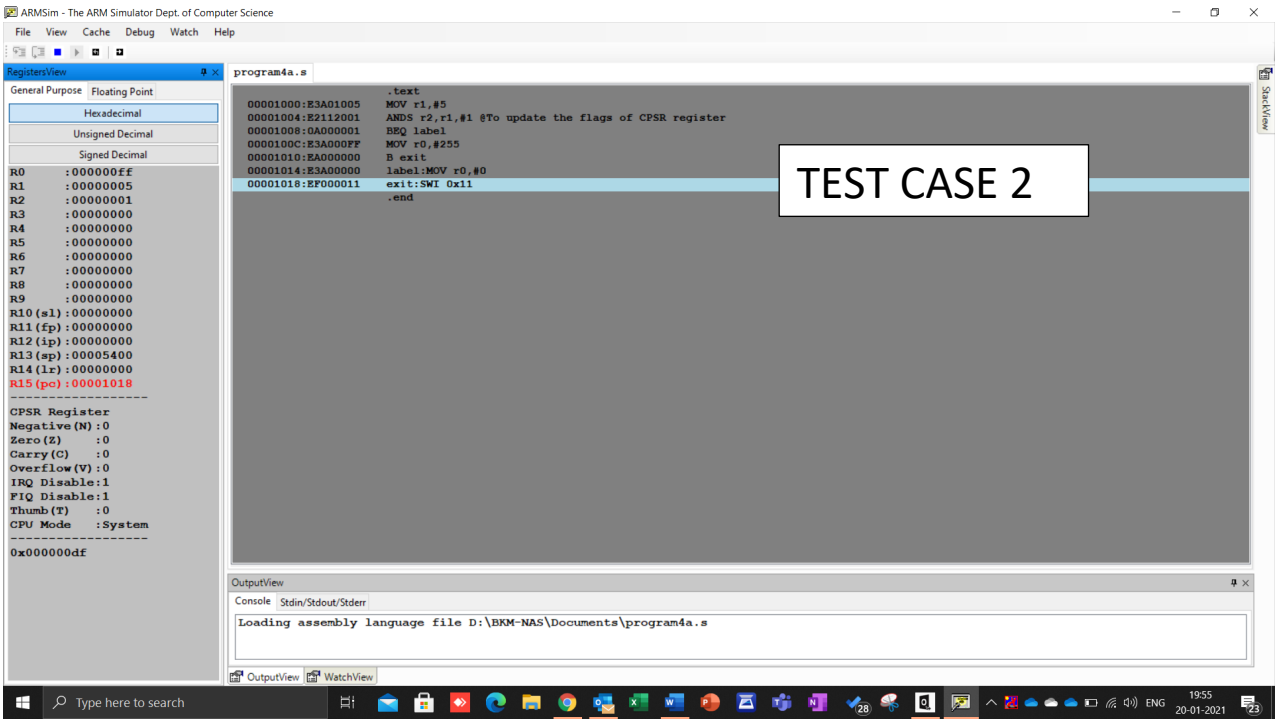
```
.text
MOV r1,#6
ANDS r2,r1,#1
BEQ label
MOV r0,#255
B exit
label:MOV r0,#0
exit:SWI 0x11
.end
```

TEST CASE 2

```
.text
MOV r1,#5
ANDS r2,r1,#1 @To update the flags of CPSR register
BEQ label
MOV r0,#255
B exit
label:MOV r0,#0
exit:SWI 0x11
.end
```

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





III. Output table for each program

CASES	REGISTERS		OUTPUT
CASE 1	R1		0x06
	R2	After AND operation	0x00
	R0	(EVEN)	0x00
CASE 2	R1		0x05
	R2	After AND operation	0x01
	R0	(ODD)	0xFF

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: Priya Mohata
Name: Priya Mohata
SRN: PES2UG19CS301
Section: E
Date: 25/01/2021