Microprocessor and Computer Architecture

UE23CS251B 4th Semester, Academic Year 2024-25

Date: 18/01/2025

Name: Keerthan P.V.	SRN: PES2UG23CS272	Section: 4E

Week:1 Program Number: 1

Title of the Program

Write an ALP to perform Addition for of two numbers of size

a)64 bit

b) 128 bit

save the result in register (reuse the register to store the result)

I. ARM Assembly Code:

1.64 bit:

```
.text
MOV R0,#0xFFFFFFFF
MOV R1,#0xFFFFFFFF
MOV R2,#0xFFFFFFFF
MOV R3,#0xFFFFFFFF
ADDS R0,R0,R2
ADCS R1,R1,R3
```

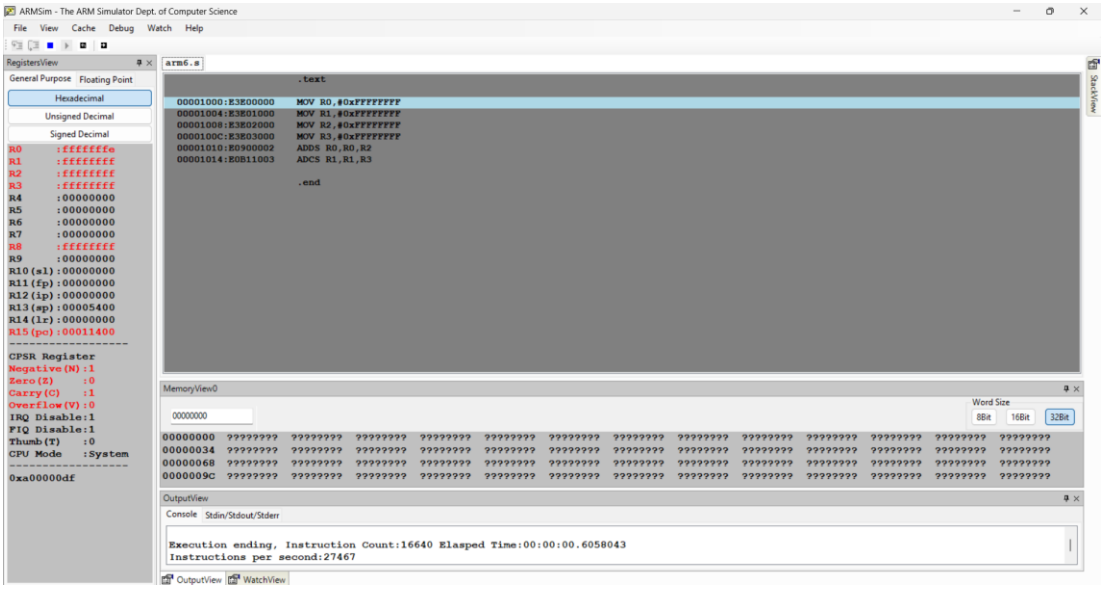
```
.end
```

2.128 bit

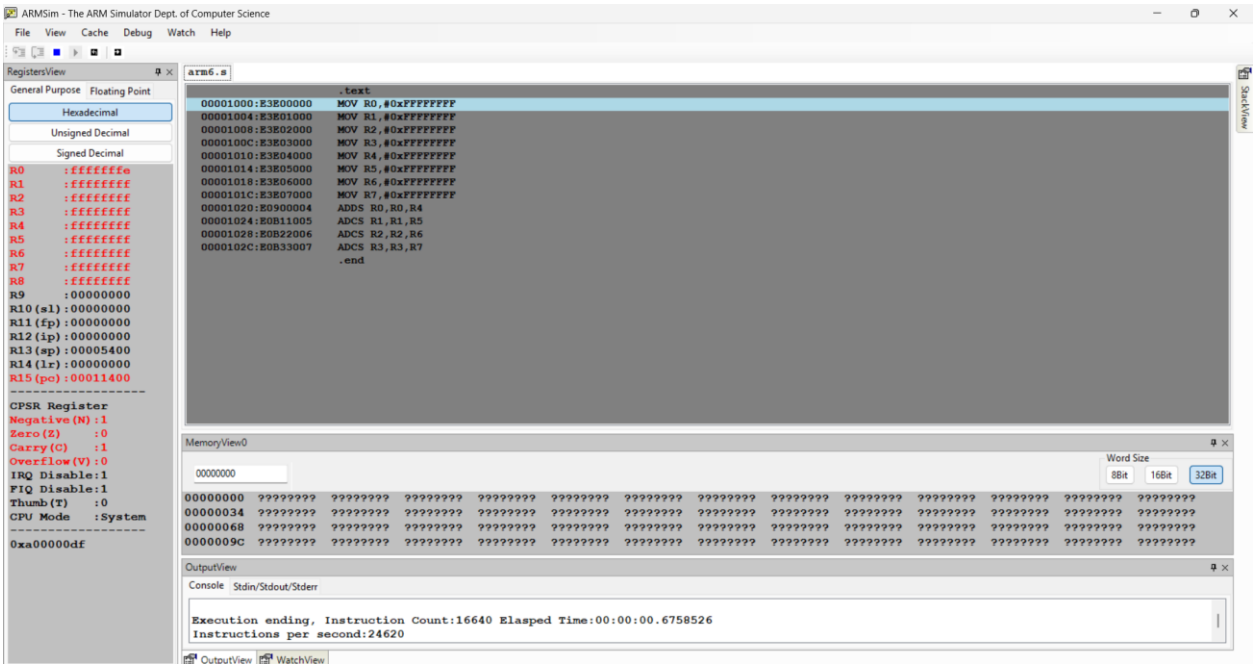
```
.text
MOV R0,#0xFFFFFFFF
MOV R1,#0xFFFFFFFF
MOV R2,#0xFFFFFFFF
MOV R3,#0xFFFFFFFF
MOV R4,#0xFFFFFFFF
MOV R5,#0xFFFFFFFF
MOV R6,#0xFFFFFFFF
MOV R7,#0xFFFFFFFF
ADDS R0,R0,R4
ADCS R1,R1,R5
ADCS R2,R2,R6
ADCS R3,R3,R7
.end
```

II. Output Screen Shots

1.64 bit:



2.128 bit



Week: 1 Program Number: 2

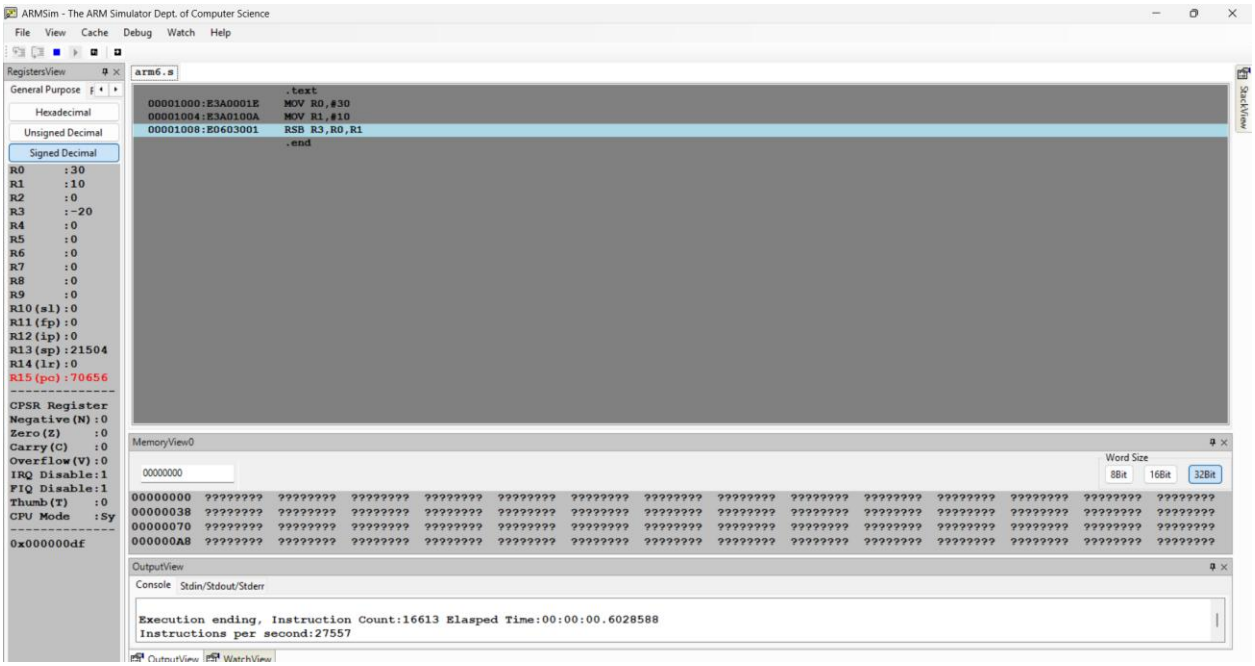
Title of the Program

Write an ALP to perform 2'complement only using mov and RSB instruction

I. ARM Assembly Code

```
.text
MOV R0,#30
MOV R1,#10
RSB R3,R0,R1
.end
```

II. Output Screen Shots (1)



Week: 1 ProgramNumber:3

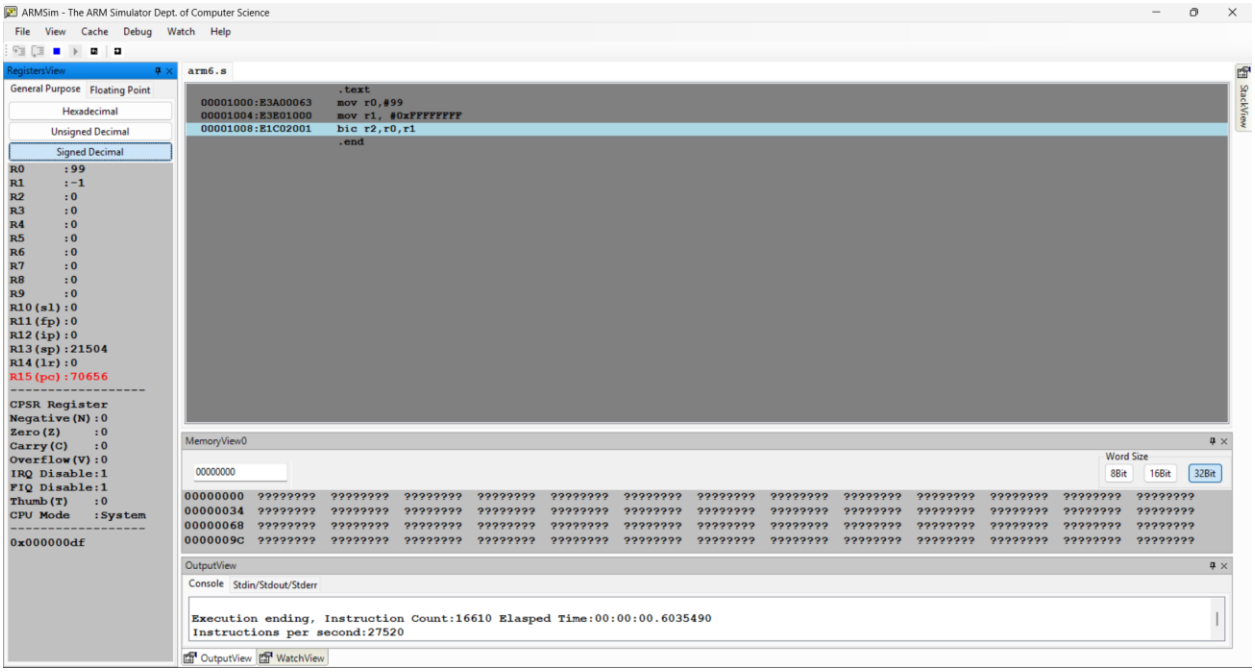
Title of the Program

Write an ALP to perform not operation only using mov and bitwise logical instructions.

I. ARM Assembly Code

```
.text
mov r0,#99
mov r1, #0xFFFFFFFF
bic r2,r0,r1
.end
```

II. Output Screen Shots (1)



Week: 1 Program Number: 4

Title of the Program

Write an ALP to subtract if the numbers are equal, otherwise add them.

I. ARM Assembly Code

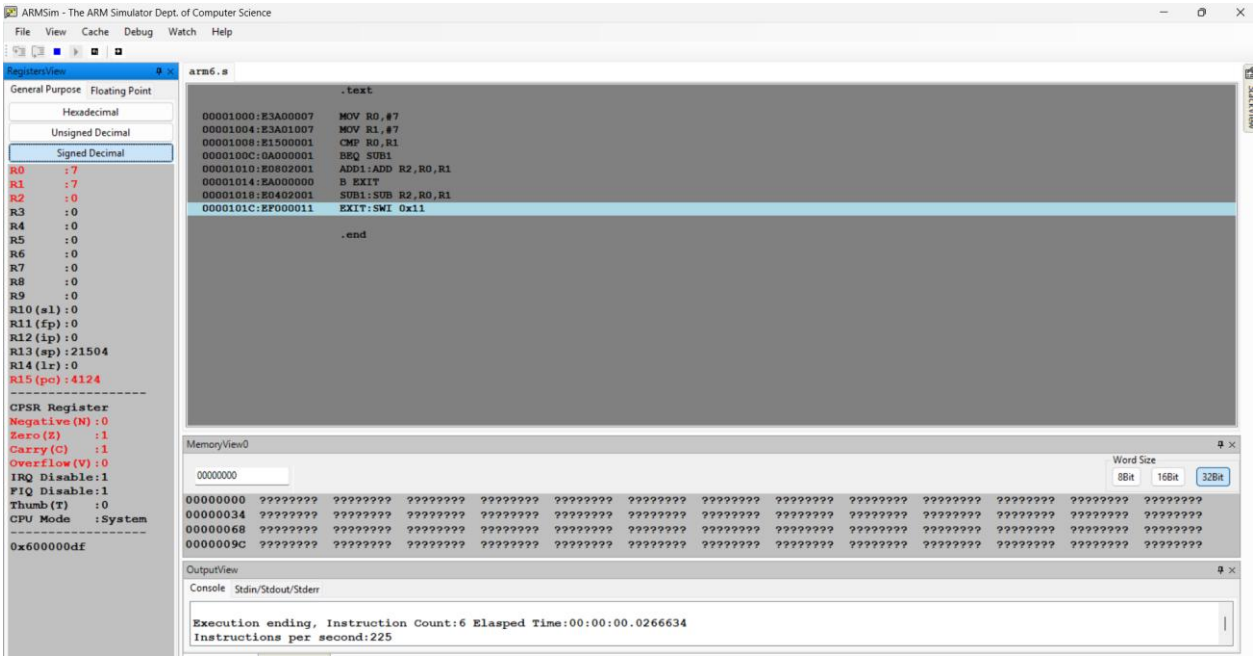
```
1.equal_numbers:
.text

MOV R0,#7
MOV R1,#7
CMP R0,R1
BEQ SUB1
ADD1:ADD R2,R0,R1
B EXIT
SUB1:SUB R2,R0,R1
EXIT:SWI 0x11

.end
```

2.unequal_numbers:

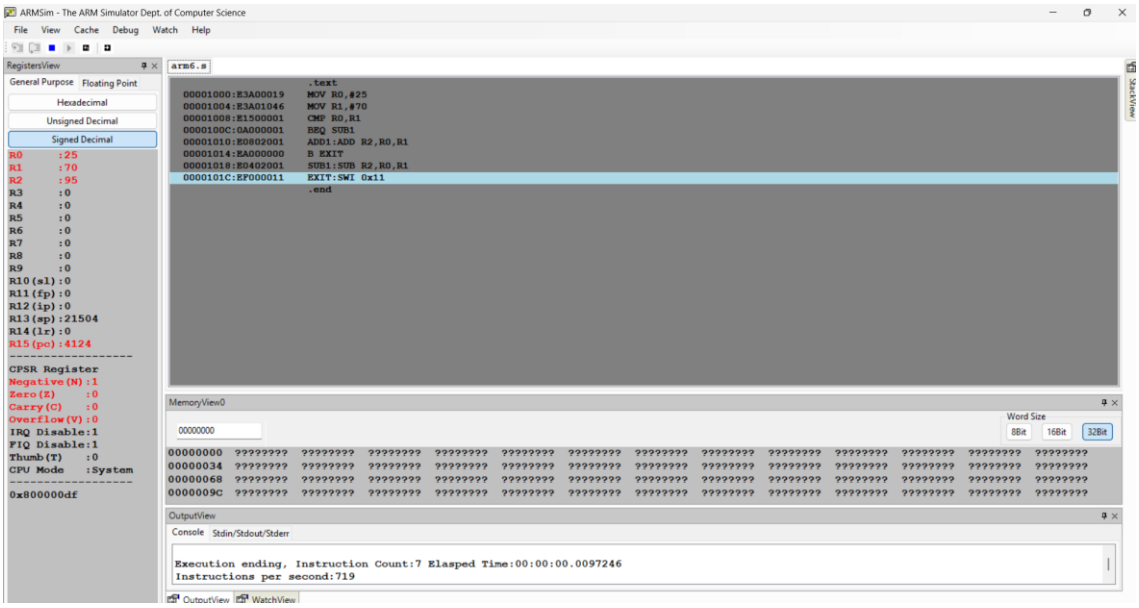
```
.text
MOV R0,#25
MOV R1,#70
CMP R0,R1
BEQ SUB1
ADD1:ADD R2,R0,R1
B EXIT
SUB1:SUB R2,R0,R1
EXIT:SWI 0x11
.end
```

II. Output Screen Shots (2)

1.equal numbers:



2.unequal numbers:



Week: 1 Program Number: 5

Title of the Program

Write an ALP using ARM instruction set to check if a number stored in a register is even or odd.

Note: at the end of the program execution R2 contains 0 if number is even, otherwise R2 contains 1.

I. ARM Assembly Code

1.for_even_number:

```
.text
MOV R0,#22
ANDS R1,R0,#0x01
BEQ EVEN
ODD:MOV R2,#1
B EXIT
EVEN:MOV R2,#0
EXIT:SWI 0x11
.end
```

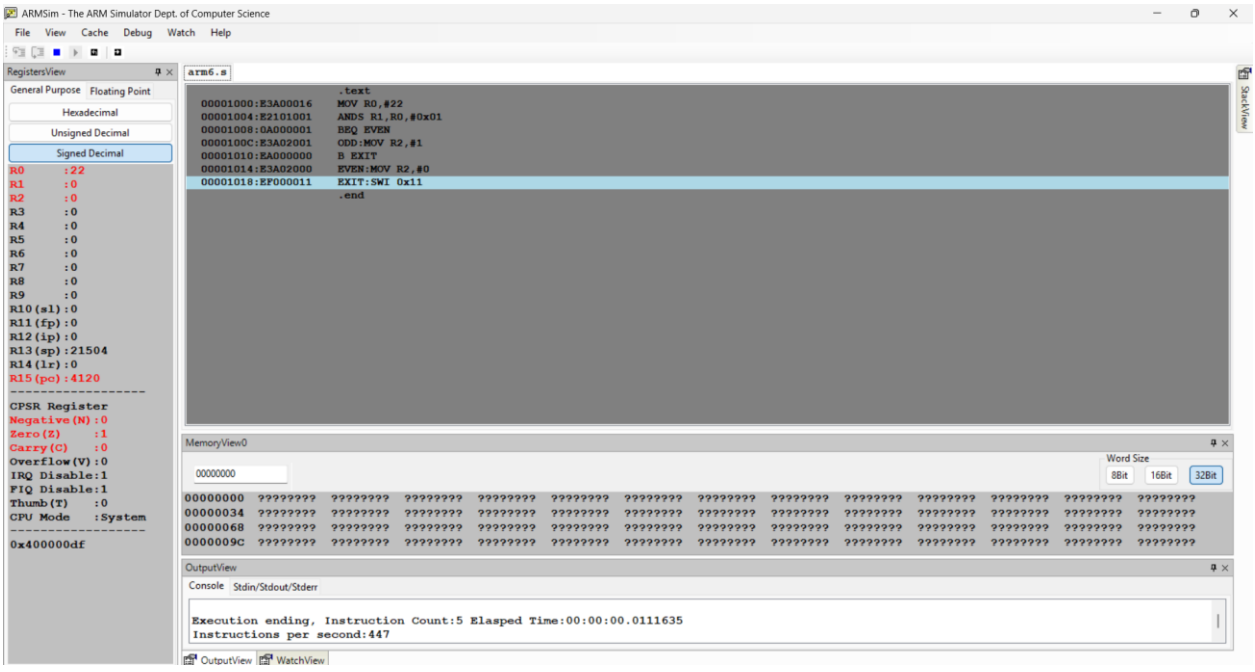
2.for_odd_number:

```
.text

MOV R0,#21
```

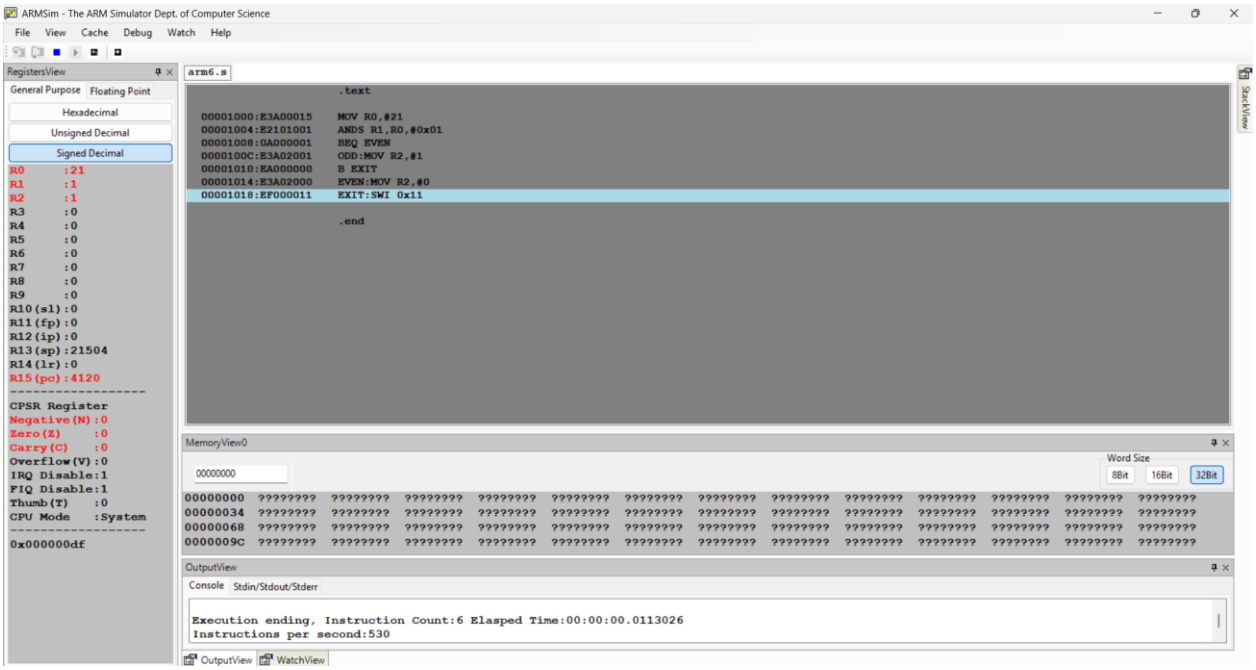
```
ANDS R1,R0,#0x01
BEQ EVEN
ODD:MOV R2,#1
B EXIT
EVEN:MOV R2,#0
EXIT:SWI 0x11

.end
```

II. Output Screen Shot (2)

1.for_even_number:



2.for_odd_number:


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 :21
R1 :1
R2 :1
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) :0
Carry(C) :0
Overflow(V):0
IRQ Disable:1
FIQ Disable:1
Thumb(T) :0
CPU Mode :System
0x000000df
arm6.s
.text
00001000:E3A00015 MOV R0,#21
00001004:E2101001 ANDS R1,R0,#0x01
00001008:0A000001 BEQ EVEN
0000100C:E3A02001 ODD:MOV R2,#1
00001010:EA000000 B EXIT
00001014:E3A02000 EVEN:MOV R2,#0
00001018:EF000011 EXIT:SWI 0x11
.end
StackView
MemoryView0
00000000
Word Size
8Bit 16Bit 32Bit
00000000 ???????? ???????? ???????? ???????? ???????? ???????? ???????? ???????? ???????? ???????? ???????? ???????? ????????
00000034 ???????? ???????? ???????? ???????? ???????? ???????? ???????? ???????? ???????? ???????? ???????? ???????? ????????
00000068 ???????? ???????? ???????? ???????? ???????? ???????? ???????? ???????? ???????? ???????? ???????? ???????? ????????
0000009C ???????? ???????? ???????? ???????? ???????? ???????? ???????? ???????? ???????? ???????? ???????? ???????? ????????
OutputView
Console Stdin/Stdout/Stderr
Execution ending, Instruction Count:6 Elasped Time:00:00:00.0113026
Instructions per second:530
OutputView WatchView

Week: 1 Program Number: 6

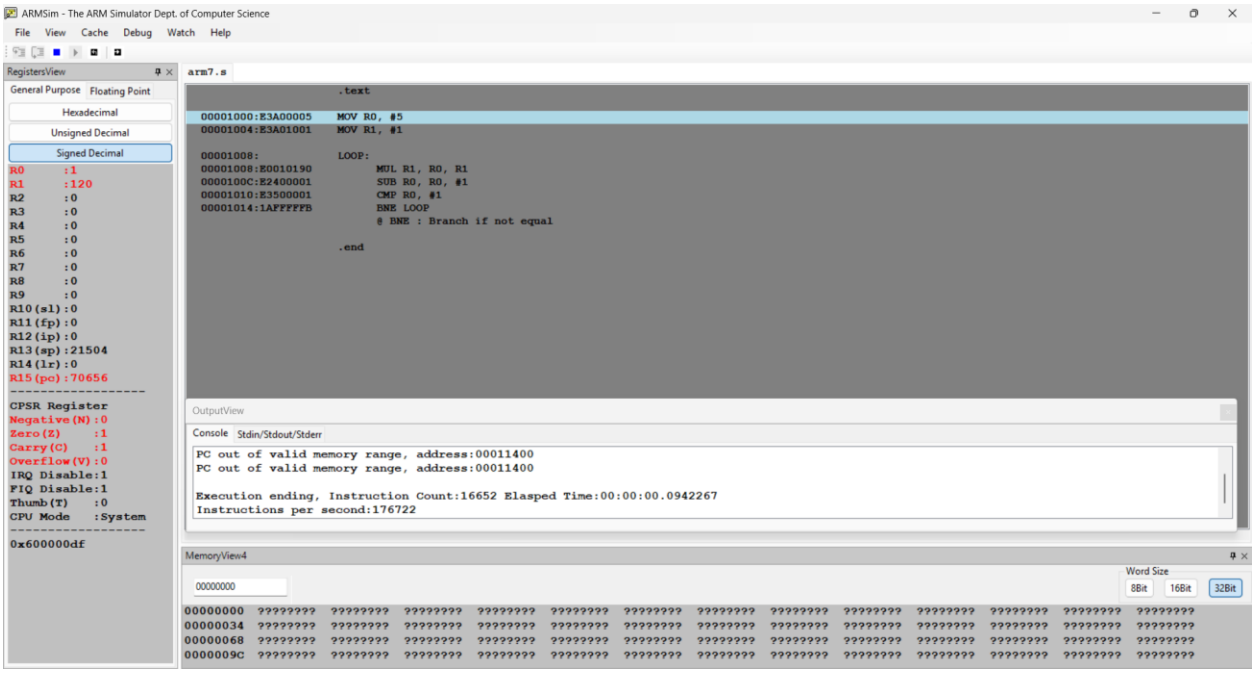
Title of the Program

Write an ALP using ARM instruction set to find the factorial of a given number.

I. ARM Assembly Code

```
.text

MOV R0, #5
MOV R1, #1

LOOP:
    MUL R1, R0, R1
    SUB R0, R0, #1
    CMP R0, #1
    BNE LOOP
    @ BNE : Branch if not equal

.end
```

II. Output Screen Shot



Week: 1 Program Number: 7

Title of the Program

Write an ALP using ARM instruction set to find the GCD of two numbers A and B.

I. ARM Assembly Code

```
1.r0=r1
.text
MOV R0,#30
MOV R1,#30
GCD:
CMP R0,R1
BEQ EXIT
```

```
BGT SUB1
BLT SUB2
SUB1:SUB R0,R0,R1
B GCD
SUB2:SUB R1,R1,R0
B GCD
EXIT:SWI 0x11
.end
```

2.r0<r1:

```
.text
MOV R0,#15
MOV R1,#25
GCD:
CMP R0,R1
BEQ EXIT
BGT SUB1
BLT SUB2
SUB1:SUB R0,R0,R1
B GCD
SUB2:SUB R1,R1,R0
B GCD
EXIT:SWI 0x11
```

```
.end
```

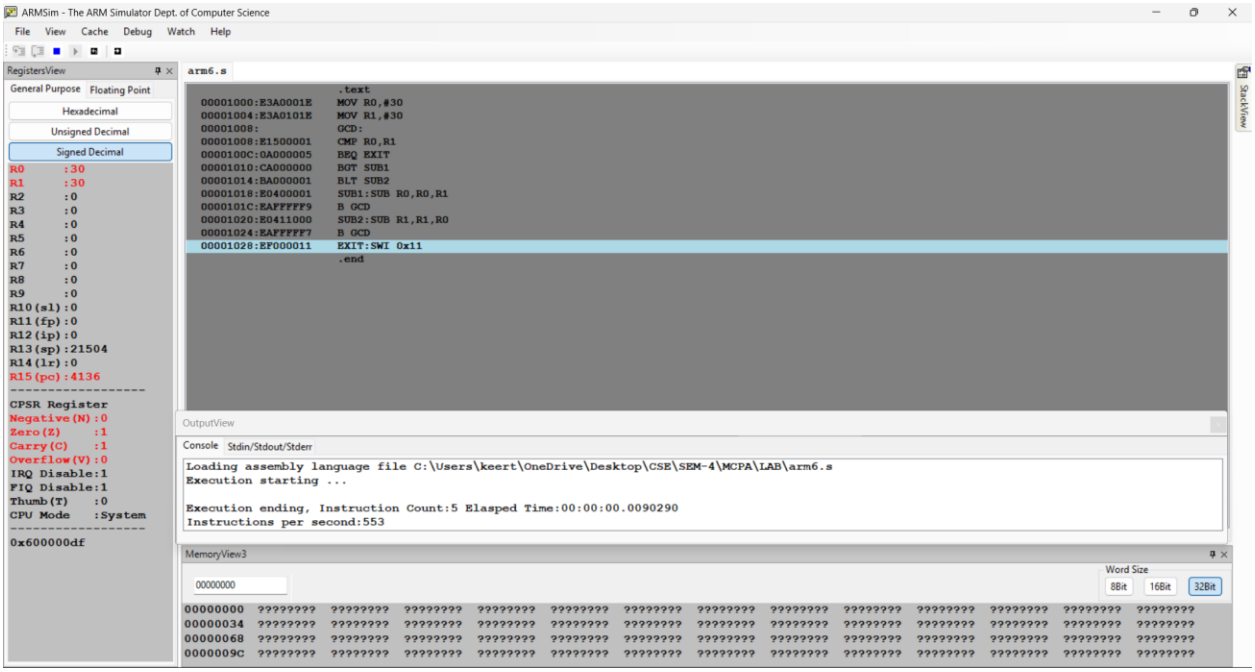
3.r0>r1:

```
.text

MOV R0,#25
MOV R1,#10

GCD:
CMP R0,R1
BEQ EXIT
BGT SUB1
BLT SUB2

SUB1:SUB R0,R0,R1
B GCD

SUB2:SUB R1,R1,R0
B GCD

EXIT:SWI 0x11
```

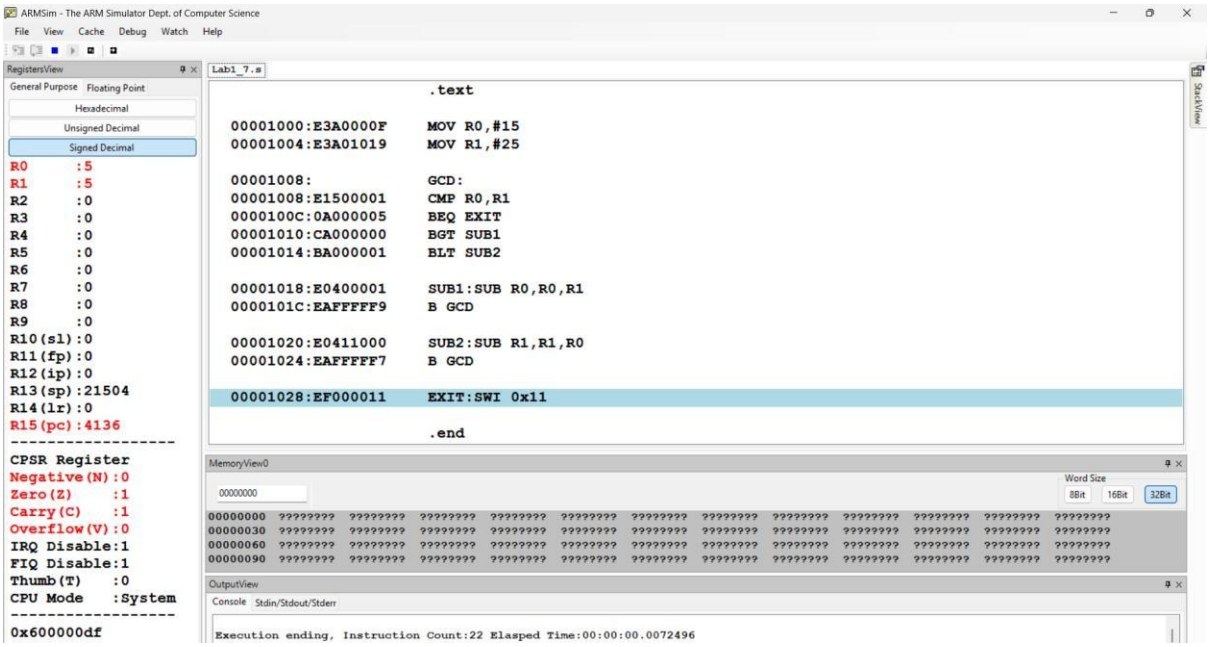
.end

II. Output Screen Shots (3)

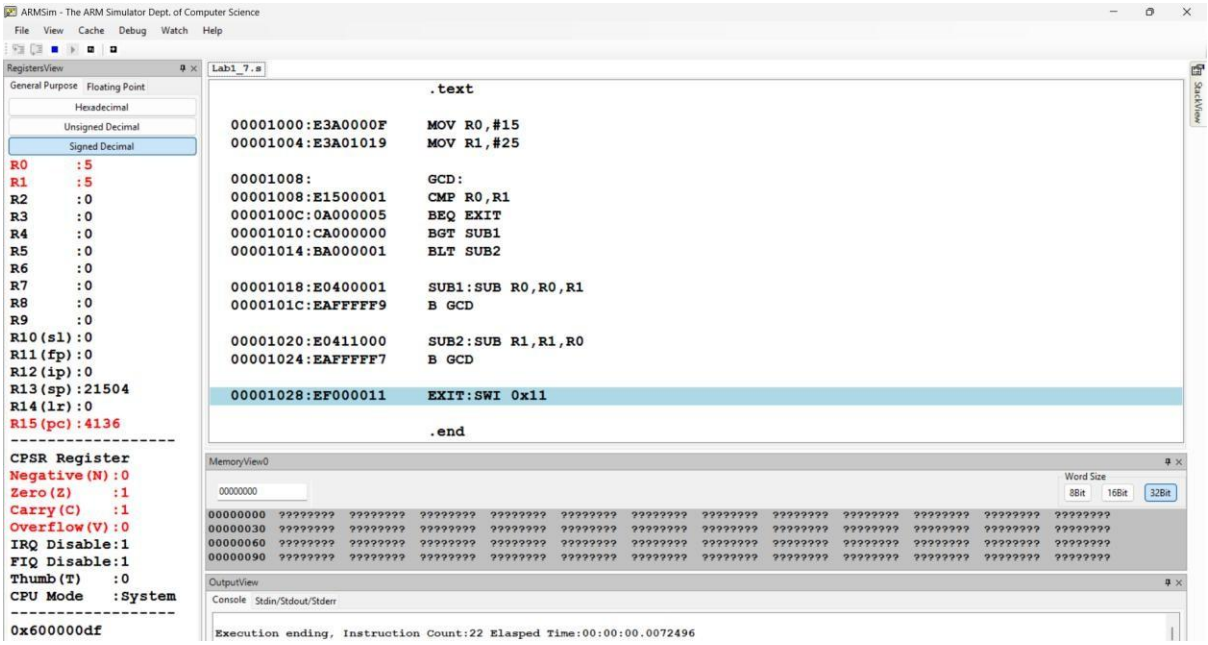
1.r0=r1



2.r0>r1



3.r0<r1:



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: Keerthan pv

Name: Keerthan pv

SRN: pes2ug23cs272

Section:E

Date:

18/01/2025