

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

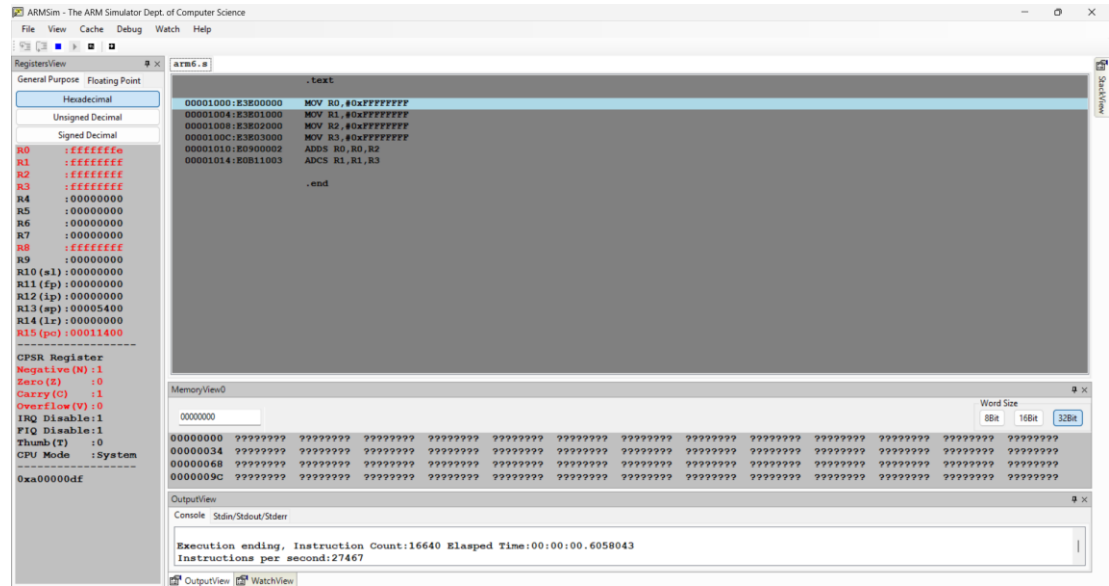
ADCS R2,R2,R6

ADCS R3,R3,R7

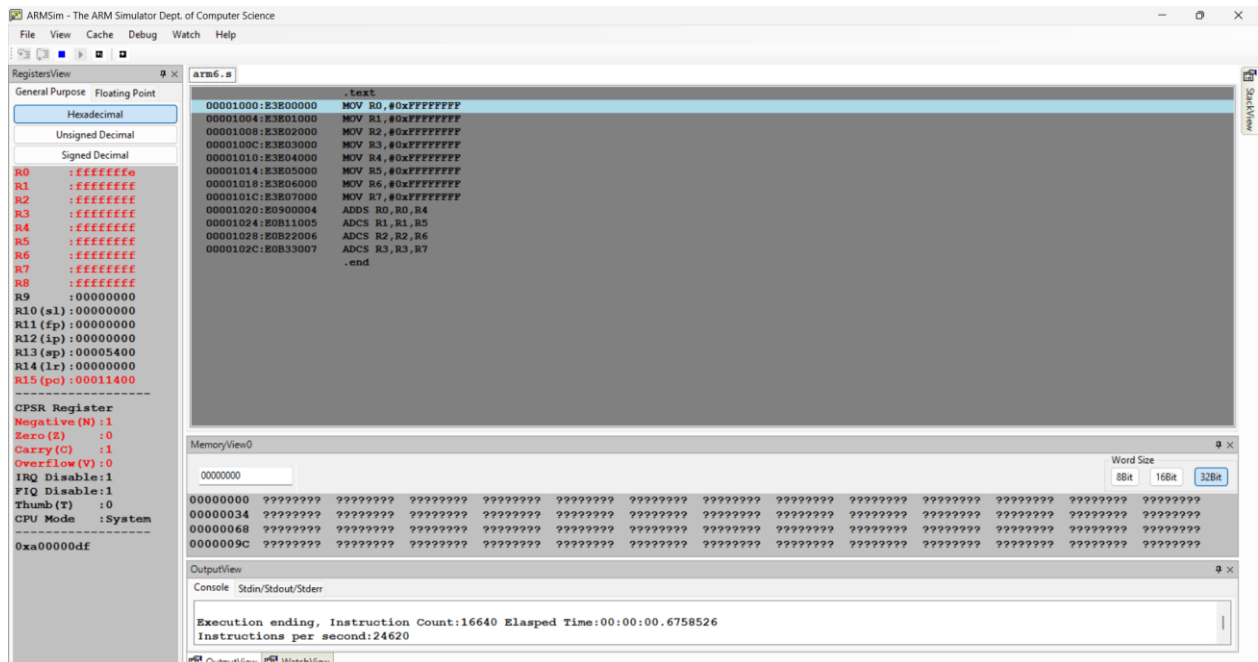
.end

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's complement only using mov and RSB instruction

I. ARM Assembly Code

.text

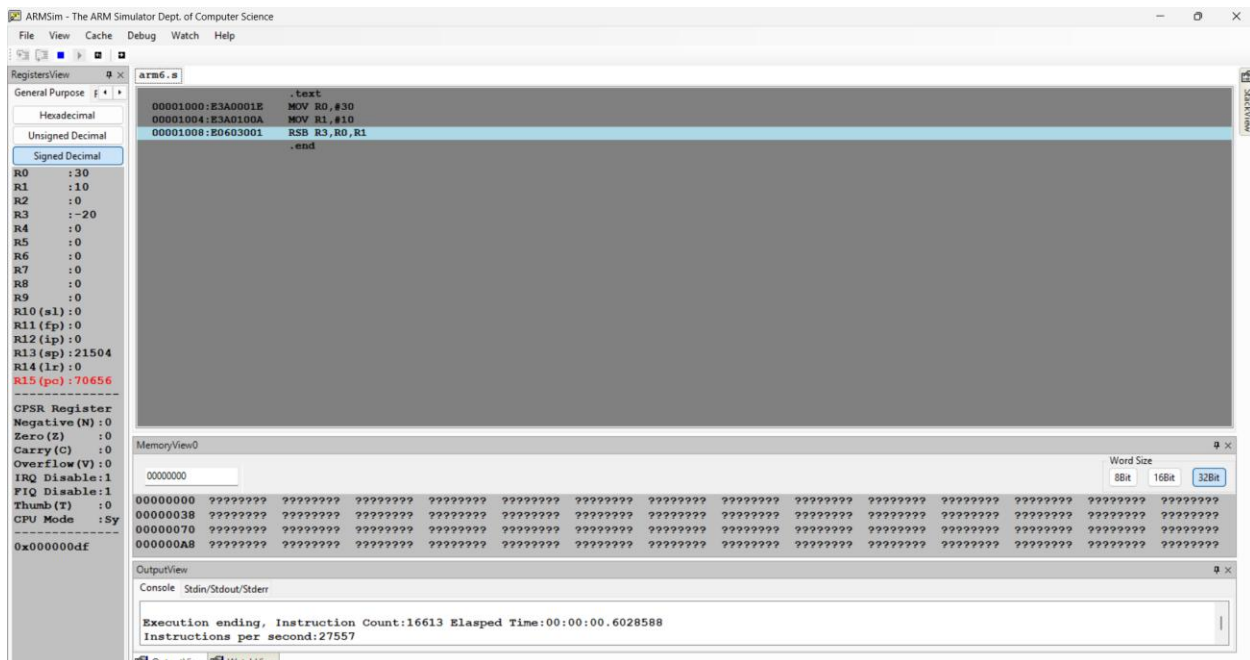
MOV R0,#30

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
```

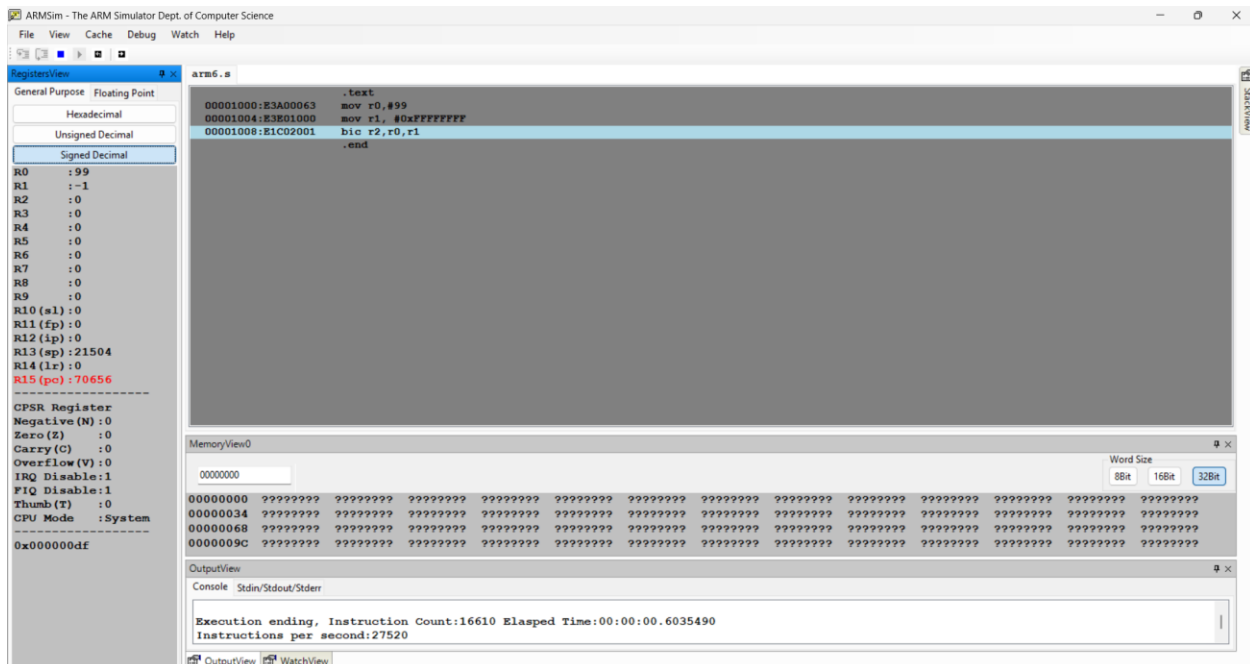
```
mov r0,#99
```

```
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

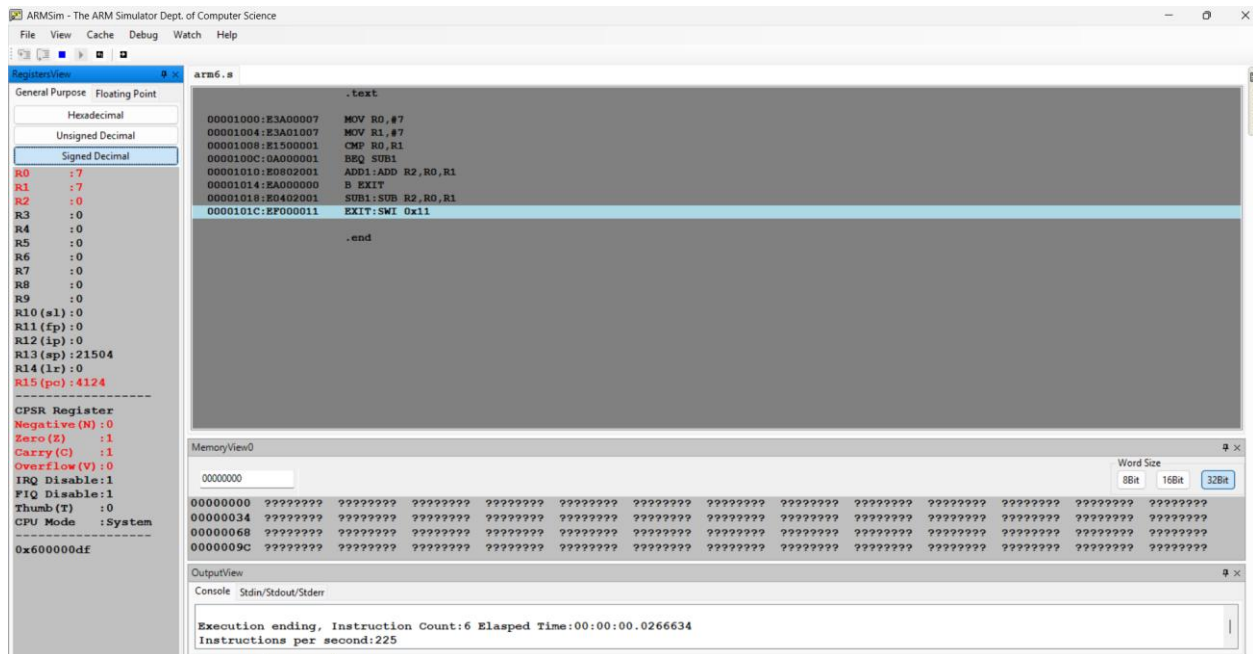
SUB1:SUB R2,R0,R1

EXIT:SWI 0x11

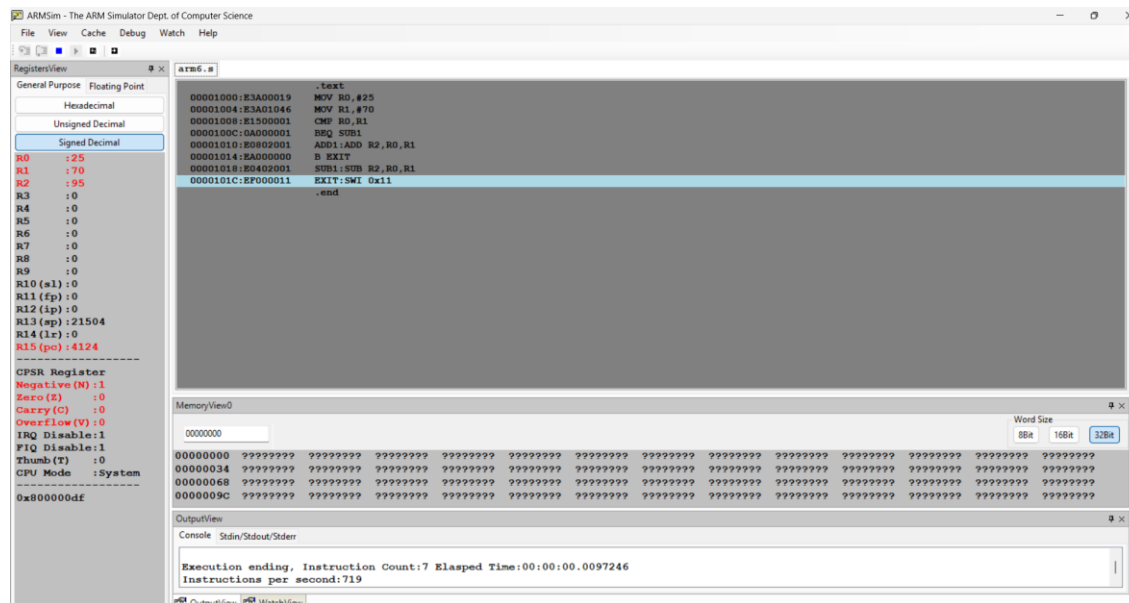
.end

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

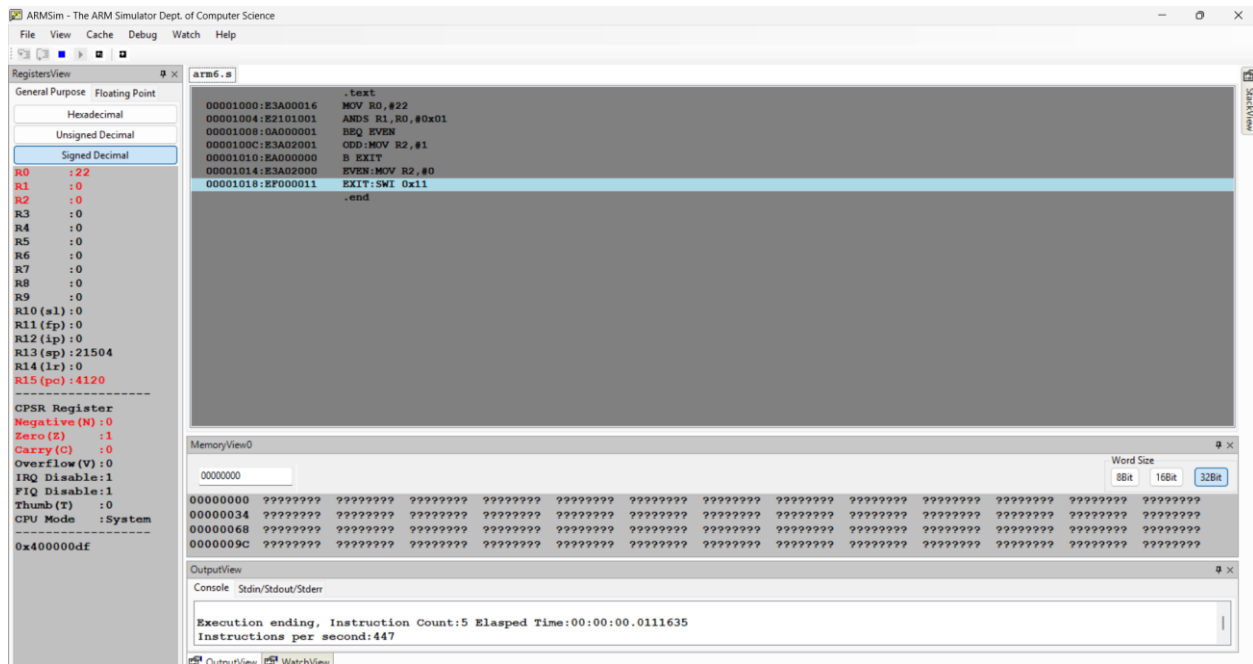
EVEN:MOV R2,#0

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

FileViewCacheDebugWatchHelp

RegistersView

General PurposeFloating Point

HexadecimalUnsigned DecimalSigned Decimal

R0:21

R1:1

R2:1

R3:0

R4:0

R5:0

R6:0

R7:0

R8:0

R9:0

R10(s1):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:E3A00015MOV R0,#21

00001004:E2101001ANDS R1,R0,#0x01

00001008:0A000001BEQ EVEN

0000100C:E3A02001ODD-MOV R2,#1

00001010:EA000000B EXIT

00001014:E3A02000EVEN-MOV R2,#0

00001018:EF000011EXIT:SWI 0x11

.end

MemoryView0

Word Size8Bit16Bit32Bit

00000000

00000000??

00000034??

00000068??

0000009C??

OutputView

ConsoleStdin/Stdout/Stderr

Execution ending, Instruction Count:6 Elapsed 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

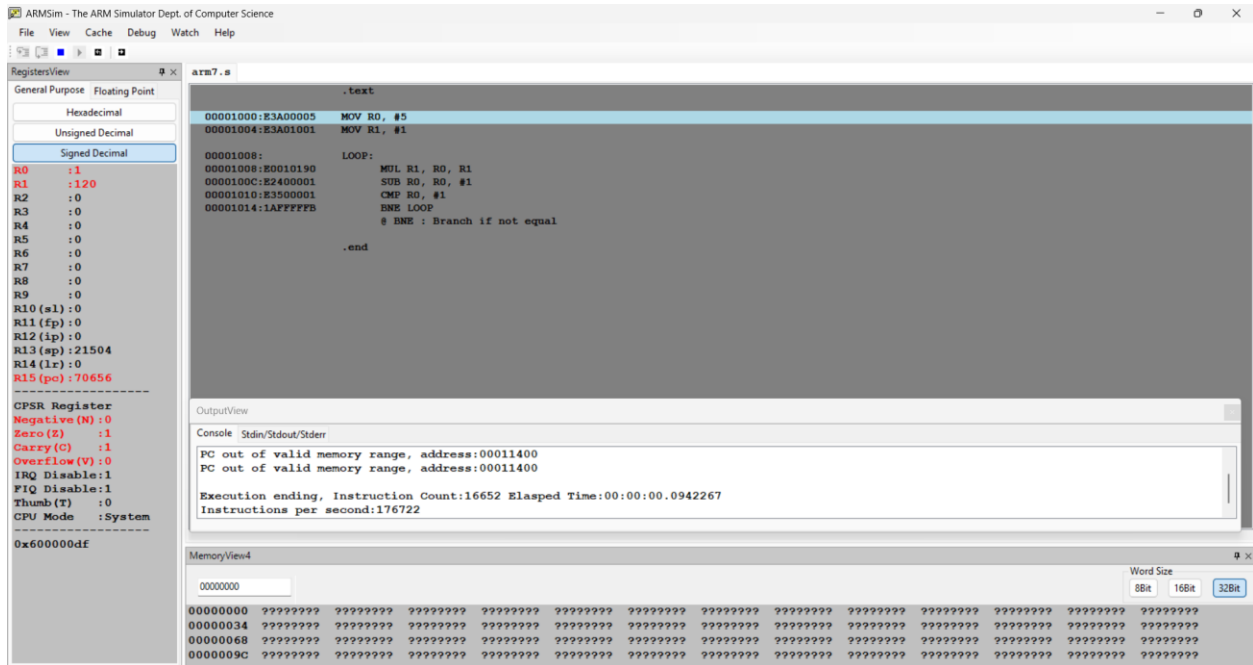
CMP R0, #1

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

II. Output Screen Shots (3)

1.r0=r1

The screenshot shows the ARM Simulator interface. The main window displays assembly code for a file named `arm6.s`. The code includes instructions such as `MOV R0, #30`, `MOV R1, #30`, `OCB:`, `CMP R0, R1`, `BEQ EXIT`, `BGT SUB1`, `BLT SUB2`, `SUB1: SUB R0, R0, R1`, `B OCB`, `SUB2: SUB R1, R1, R0`, `B OCB`, `EXIT: SWI 0x11`, and `.end`.

The left sidebar shows the **RegistersView** with a list of registers (R0-R15) and their values. The **CPSR Register** section shows status flags: **Negative (N)**: 0, **Zero (Z)**: 1, **Carry (C)**: 1, **Overflow (V)**: 1, **IRQ Disable**: 1, **FIQ Disable**: 1, **Thumb (T)**: 0, and **CPU Mode**: System.

The bottom section shows the **Console** output, indicating execution starting and ending, and the **MemoryView** at the bottom.

[illegible]

3.r0<r1:

The screenshot displays the ARMSim ARM Simulator interface. The main window shows assembly code for a program labeled 'Lab1_7.s'. The code includes instructions for moving values into registers, comparing, branching, and exiting. The left sidebar shows the 'RegistersView' with a list of registers (R0-R15) and their current values. The bottom status bar indicates the execution is ending with an instruction count of 22 and an elapsed time of 00:00:00.0072496.

```
.text
00001000:E3A0000F  MOV R0,#15
00001004:E3A01019  MOV R1,#25

00001008:          GCD:
00001008:E1500001  CMP R0,R1
0000100C:0A000005  BEQ EXIT
00001010:CA000000  BGT SUB1
00001014:BA000001  BLT SUB2

00001018:E0400001  SUB1: SUB R0,R0,R1
0000101C:EAF00007  B GCD

00001020:E0411000  SUB2: SUB R1,R1,R0
00001024:EAF00007  B GCD

00001028:EF000011  EXIT: SWI 0x11

.end
```

RegistersView:

Register	Value
R0	5
R1	5
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)	4136

CPSR Register:

Field	Value
Negative (N)	0
Zero (Z)	1
Carry (C)	1
Overflow (V)	0
IRQ Disable	1
FIQ Disable	1
Thumb (T)	0
CPU Mode	System

MemoryView0:

Address	Value
00000000	00000000
00000004	00000000
00000008	00000000
0000000C	00000000
00000010	00000000
00000014	00000000
00000018	00000000
0000001C	00000000
00000020	00000000
00000024	00000000
00000028	00000000
0000002C	00000000
00000030	00000000
00000034	00000000
00000038	00000000
0000003C	00000000
00000040	00000000
00000044	00000000
00000048	00000000
0000004C	00000000
00000050	00000000
00000054	00000000
00000058	00000000
0000005C	00000000
00000060	00000000
00000064	00000000
00000068	00000000
0000006C	00000000
00000070	00000000
00000074	00000000
00000078	00000000
0000007C	00000000
00000080	00000000
00000084	00000000
00000088	00000000
0000008C	00000000
00000090	00000000
00000094	00000000
00000098	00000000
0000009C	00000000
000000A0	00000000
000000A4	00000000
000000A8	00000000
000000AC	00000000
000000B0	00000000
000000B4	00000000
000000B8	00000000
000000BC	00000000
000000C0	00000000
000000C4	00000000
000000C8	00000000
000000CC	00000000
000000D0	00000000
000000D4	00000000
000000D8	00000000
000000DC	00000000
000000E0	00000000
000000E4	00000000
000000E8	00000000
000000EC	00000000
000000F0	00000000
000000F4	00000000
000000F8	00000000
000000FC	00000000
00000100	00000000
00000104	00000000
00000108	00000000
0000010C	00000000
00000110	00000000
00000114	00000000
00000118	00000000
0000011C	00000000
00000120	00000000
00000124	00000000
00000128	00000000
0000012C	00000000
00000130	00000000
00000134	00000000
00000138	00000000
0000013C	00000000
00000140	00000000
00000144	00000000
00000148	00000000
0000014C	00000000
00000150	00000000
00000154	00000000
00000158	00000000
0000015C	00000000
00000160	00000000
00000164	00000000
00000168	00000000
0000016C	00000000
00000170	00000000
00000174	00000000
00000178	00000000
0000017C	00000000
00000180	00000000
00000184	00000000
00000188	00000000
0000018C	00000000
00000190	00000000
00000194	00000000
00000198	00000000
0000019C	00000000
000001A0	00000000
000001A4	00000000
000001A8	00000000
000001AC	00000000
000001B0	00000000
000001B4	00000000
000001B8	00000000
000001BC	00000000
000001C0	00000000
000001C4	00000000
000001C8	00000000
000001CC	00000000
000001D0	00000000
000001D4	00000000
000001D8	00000000
000001DC	00000000
000001E0	00000000
000001E4	00000000
000001E8	00000000
000001EC	00000000
000001F0	00000000
000001F4	00000000
000001F8	00000000
000001FC	00000000
00000200	00000000
00000204	00000000
00000208	00000000
0000020C	00000000
00000210	00000000
00000214	00000000
00000218	00000000
0000021C	00000000
00000220	00000000
00000224	00000000
00000228	00000000
0000022C	00000000
00000230	00000000
00000234	00000000
00000238	00000000
0000023C	00000000
00000240	00000000
00000244	00000000
00000248	00000000
0000024C	00000000
00000250	00000000
00000254	00000000
00000258	00000000
0000025C	00000000
00000260	00000000
00000264	00000000
00000268	00000000
0000026C	00000000
00000270	00000000
00000274	00000000
00000278	00000000
0000027C	00000000
00000280	00000000
00000284	00000000
00000288	00000000
0000028C	00000000
00000290	00000000
00000294	00000000
00000298	00000000
0000029C	00000000
000002A0	00000000
000002A4	00000000
000002A8	00000000
000002AC	00000000
000002B0	00000000
000002B4	00000000
000002B8	00000000
000002BC	00000000
000002C0	00000000
000002C4	00000000
000002C8	00000000
000002CC	00000000
000002D0	00000000
000002D4	00000000
000002D8	00000000
000002DC	00000000
000002E0	00000000
000002E4	00000000
000002E8	00000000
000002EC	00000000
000002F0	00000000
000002F4	00000000
000002F8	00000000
000002FC	00000000
00000300	00000000
00000304	00000000
00000308	00000000
0000030C	00000000
00000310	00000000
00000314	00000000
00000318	00000000
0000031C	00000000
00000320	00000000
00000324	00000000
00000328	00000000
0000032C	00000000
00000330	00000000
00000334	00000000
00000338	00000000
0000033C	00000000
00000340	00000000
00000344	00000000
00000348	00000000
0000034C	00000000
00000350	00000000
00000354	00000000
00000358	00000000
0000035C	00000000
00000360	00000000
00000364	00000000
00000368	00000000
0000036C	00000000
00000370	00000000
00000374	00000000
00000378	00000000
0000037C	00000000
00000380	00000000
00000384	00000000
00000388	00000000
0000038C	00000000
00000390	00000000
00000394	00000000
00000398	00000000
0000039C	00000000
000003A0	00000000
000003A4	00000000
000003A8	00000000
000003AC	00000000
000003B0	00000000
000003B4	00000000
000003B8	00000000
000003BC	00000000
000003C0	00000000
000003C4	00000000
000003C8	00000000
000003CC	00000000
000003D0	00000000
000003D4	00000000
000003D8	00000000
000003DC	00000000
000003E0	00000000
000003E4	00000000
000003E8	00000000
000003EC	00000000
000003F0	00000000
000003F4	00000000
000003F8	00000000
000003FC	00000000
00000400	00000000
00000404	00000000
00000408	00000000
0000040C	00000000
00000410	00000000
00000414	00000000
00000418	00000000
0000041C	00000000
00000420	00000000
00000424	00000000
00000428	00000000
0000042C	00000000
00000430	00000000
00000434	00000000
00000438	00000000
0000043C	00000000
00000440	00000000
00000444	00000000
00000448	00000000
0000044C	00000000
00000450	00000000
00000454	00000000
00000458	00000000
0000045C	00000000
00000460	00000000
00000464	00000000
00000468	00000000
0000046C	00000000
00000470	00000000
00000474	00000000
00000478	00000000
0000047C	00000000
00000480	00000000
00000484	00000000
00000488	00000000
0000048C	00000000
00000490	00000000
00000494	00000000
00000498	00000000
0000049C	00000000
000004A0	00000000
000004A4	00000000
000004A8	00000000
000004AC	00000000
000004B0	00000000
000004B4	00000000
000004B8	00000000
000004BC	00000000
000004C0	00000000
000004C4	00000000
000004C8	00000000
000004CC	00000000
000004D0	00000000
000004D4	00000000
000004D8	00000000
000004DC	00000000
000004E0	00000000
000004E4	00000000
000004E8	00000000
000004EC	00000000
000004F0	00000000
000004F4	00000000
000004F8	00000000
000004FC	00000000
00000500	00000000
00000504	00000000
00000508	00000000
0000050C	00000000
00000510	00000000
00000514	00000000
00000518	00000000
0000051C	00000000
00000520	00000000
00000524	00000000
00000528	00000000
0000052C	00000000
00000530	00000000
00000534	00000000
00000538	00000000
0000053C	00000000
00000540	00000000
00000544	00000000
00000548	00000000
0000054C	00000000
00000550	00000000
00000554	00000000
00000558	00000000
0000055C	00000000
00000560	00000000
00000564	00000000
00000568	00000000
0000056C	00000000
00000570	00000000
00000574	00000000
00000578	00000000
0000057C	00000000
00000580	00000000
00000584	00000000
00000588	00000000
0000058C	00000000
00000590	00000000
00000594	00000000
00000598	00000000
0000059C	00000000
000005A0	00000000
000005A4	00000000
000005A8	00000000
000005AC	00000000
000005B0	00000000
000005B4	00000000
000005B8	00000000
000005BC	00000000
000005C0	00000000
000005C4	00000000
000005C8	00000000
000005CC	00000000
000005D0	00000000
000005D4	00000000
000005D8	00000000
000005DC	00000000
000005E0	00000000
000005E4	00000000
000005E8	00000000
000005EC	00000000

Section:E

Date:

18/01/2025