

Microprocessor and Computer Architecture Laboratory

UE19CS256

4th Semester, Academic Year 2020-21

Date: 9/2/2021

Name: B.Pravena	SRN: PES2UG19CS076	Section: B
-----------------	--------------------	------------

Week# 3 Program Number: 1

Write an ALP to add two 64 bit numbers loaded from memory and store the result in memory.

I. ARM Assembly Code for the program.

```
Week3_Program1_PES2UG19CS076 - Notepad
File Edit Format View Help
.text
LDR R0,=a
LDR R1,=b
LDR R2,=c
LDR R4,[R0]
LDR R5,[R1]
ADD R4,R4,R5
STR R4,[R2]
LDR R4,[R0,#4]
LDR R5,[R1,#4]
ADD R6,R4,R5
STR R6,[R2,#4]
SWI 0x11
.data
a:.word 12213443,56657887
b:.word 98764532,45326789
c:.word 0,0
.end
```

II. Output Screen Shot

The screenshot displays a debugger interface with two main windows: **RegistersView** and **MemoryView0**.

RegistersView: This window shows the state of the processor registers. The **General Purpose** tab is selected, and the **Hexadecimal** format is chosen. The registers R0 through R15 are listed, with their current values in hexadecimal. For example, R0 is 0000103c, R1 is 00001044, and R2 is 0000104c. The CPSR Register is also shown, with fields like Negative (N), Zero (Z), Carry (C), Overflow (V), IRQ Disable, FIQ Disable, Thumb (T), and CPU Mode (Supervisor).

MemoryView0: This window shows the contents of memory. The address 0000104c is entered in the search box. The memory view shows the following data:

Address	Value	Comment
0000104c	069D63B7	a: .word 12213443, 56657887
0000104c	061429A4	b: .word 98764532, 45326789
0000104c	00000000	c: .word 0, 0

III. Output Table for the program

	a: .word 12213443, 56657887 b: .word 98764532, 45326789	
	Upper 32 bits	Lower 32 bits
a: .word	56657887 (036087DF)	12213443 (00BA5CC3)
b: .word	45326789 (02B3A1C5)	98764532 (05E306F4)
c: .word	101984676 (061429A4)	110977975 (069D63B7)

Microprocessor and Computer Architecture Laboratory

UE19CS256

4th Semester, Academic Year 2020-21

Date: 9/2/2021

Name: B.Pravena	SRN: PES2UG19CS076	Section: B
-----------------	--------------------	------------

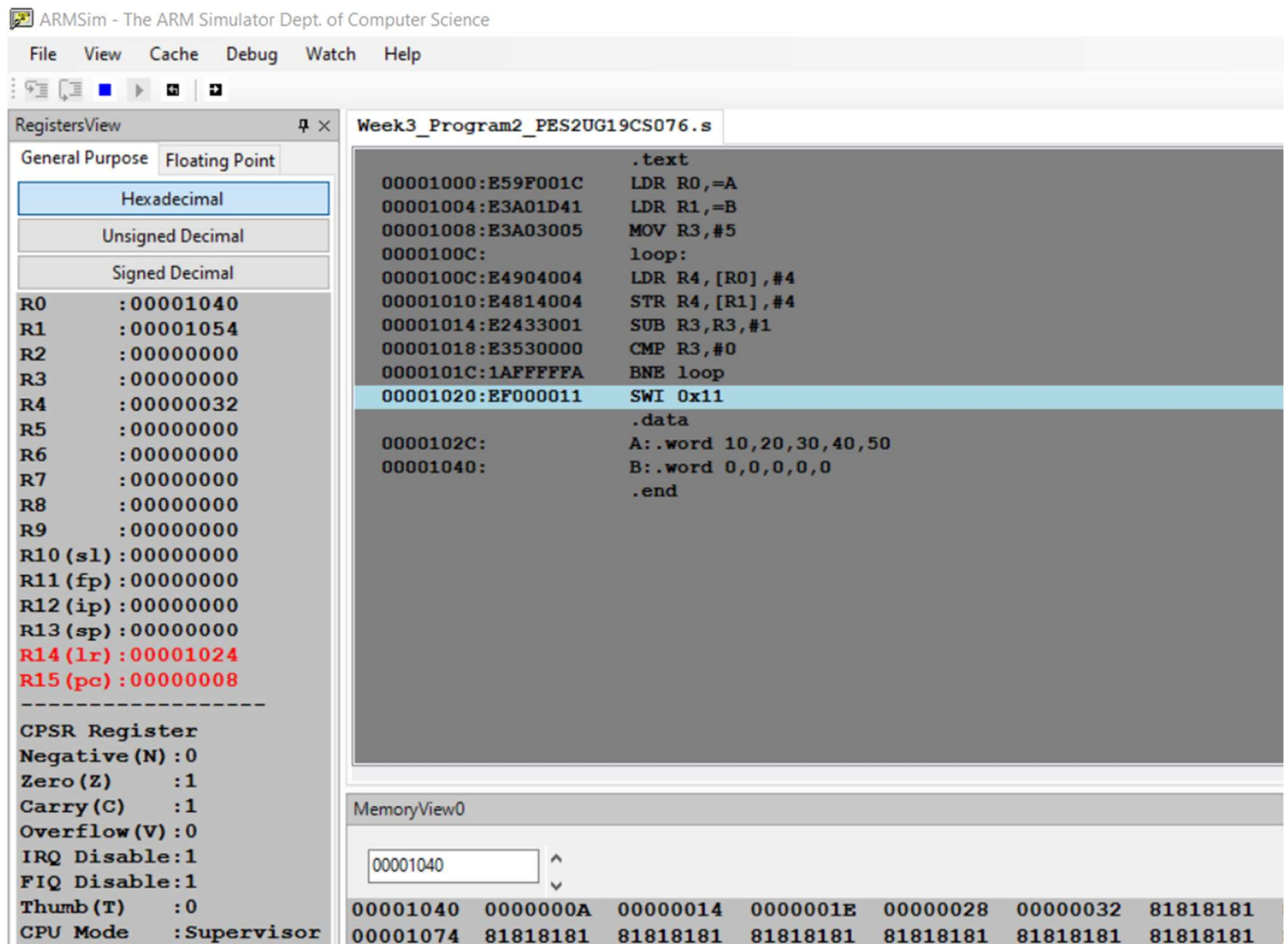
Week# 3 Program Number: 2

Write an ALP to copy n numbers from Memory Location A to Memory Location B

I. ARM Assembly Code for the program.

```
Week3_Program2_PES2UG19CS076 - Notepad
File Edit Format View Help
.text
LDR R0,=A
LDR R1,=B
MOV R3,#5
loop:
LDR R4,[R0],#4
STR R4,[R1],#4
SUB R3,R3,#1
CMP R3,#0
BNE loop
SWI 0x11
.data
A:.word 10,20,30,40,50
B:.word 0,0,0,0,0
.end
```

II. Output Screen Shot



III. Output Table for the program

1 st Iteration	a: .word 0A, 14, 1E, 28,32 b: .word 0A, 0, 0, 0,0
2 nd Iteration	a: .word 0A, 14, 1E, 28,32 b: .word 0A, 14, 0, 0,0
3 rd Iteration	a: .word 0A, 14, 1E, 28,32 b: .word 0A, 14, 1E, 0,0
4 th Iteration	a: .word 0A, 14, 1E, 28,32 b: .word 0A, 14, 1E, 28,0
5 th Iteration	a: .word 0A, 14, 1E, 28,32 b: .word 0A, 14, 1E, 28,32

Microprocessor and Computer Architecture Laboratory

UE19CS256

4th Semester, Academic Year 2020-21

Date: 9/2/2021

Name: B.Pravena	SRN: PES2UG19CS076	Section: B
-----------------	--------------------	------------

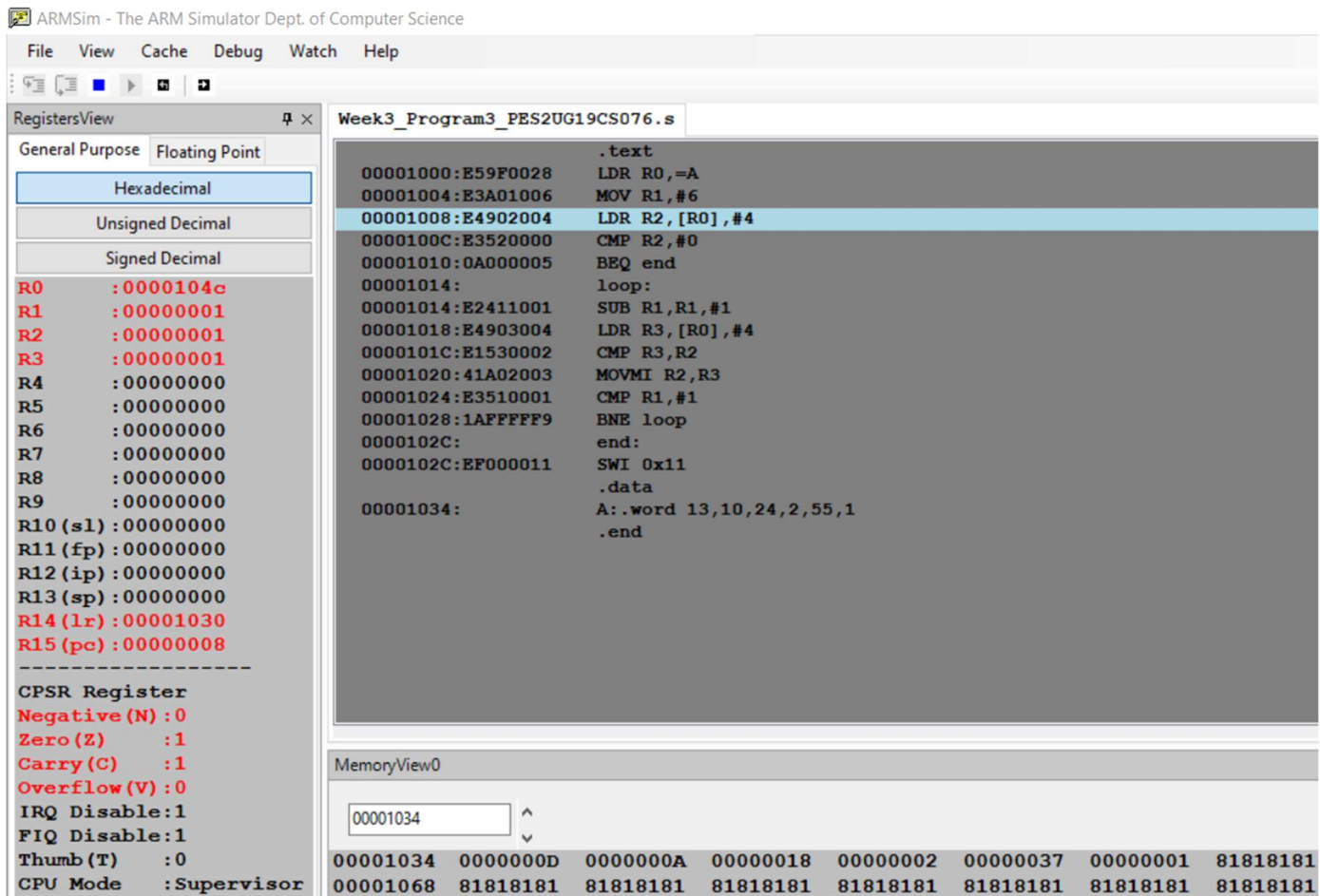
Week# 3 Program Number: 3

Write an ALP to find smallest number in an array of n 32 bit numbers

I. ARM Assembly Code for the program.

```
Week3_Program3_PES2UG19CS076 - Notepad
File Edit Format View Help
.text
LDR R0,=A
MOV R1,#6
LDR R2,[R0],#4
CMP R2,#0
BEQ end
loop:
SUB R1,R1,#1
LDR R3,[R0],#4
CMP R3,R2
MOVMN R2,R3
CMP R1,#1
BNE loop
end:
SWI 0x11
.data
A:.word 13,10,24,2,5,1
.end
```

II. Output Screen Shot



III. Output Table for the program

1 st iteration	R2 = 13, R3 = 10
2 nd iteration	R2 = 10, R3 = 24
3 rd iteration	R2 = 10, R3 = 2
4 th iteration	R2 = 2, R3 = 55
5 th iteration	R2 = 2, R3 = 1
	R2 = 1

Microprocessor and Computer Architecture Laboratory

UE19CS256

4th Semester, Academic Year 2020-21

Date: 9/2/2021

Name: B.Pravena	SRN: PES2UG19CS076	Section: B
-----------------	--------------------	------------

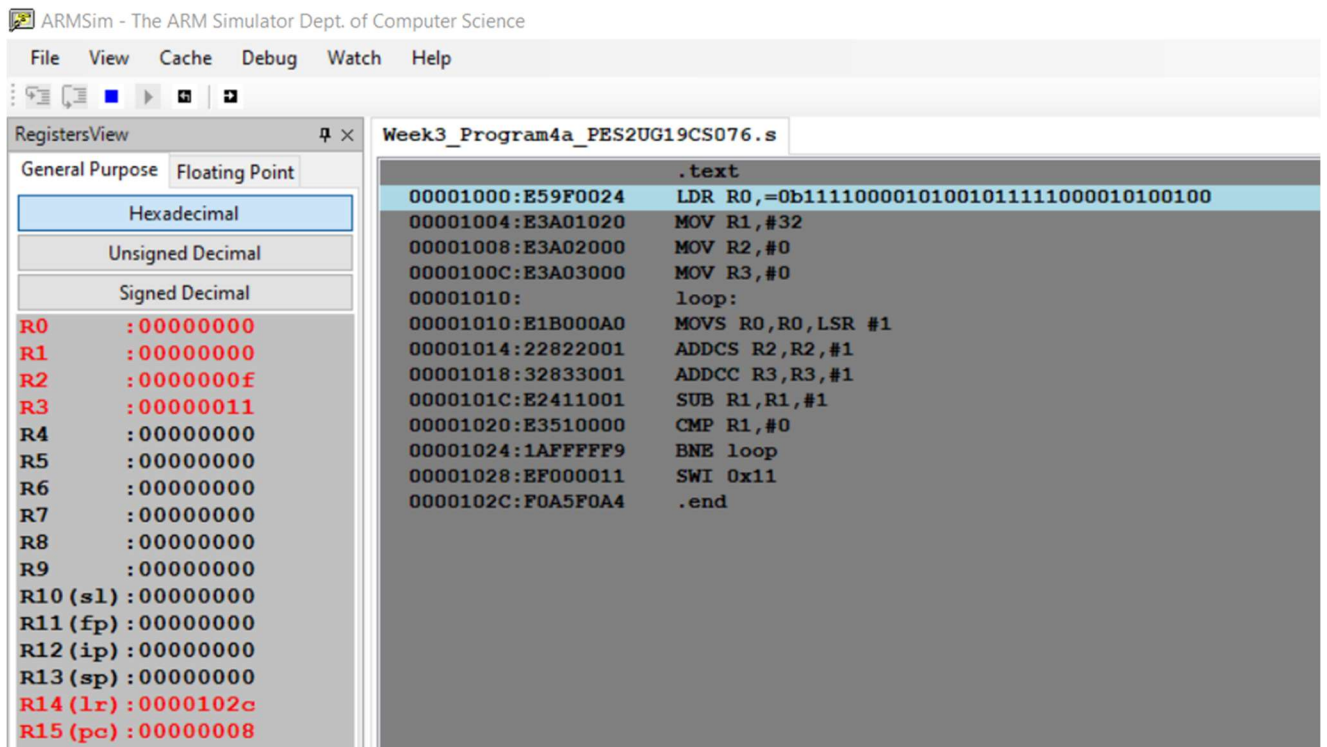
Week# 3 Program Number: 4a

Write an ALP to count the number of 1's and 0's in a given 32 bit number.

I. ARM Assembly Code for the program.

```
Week3_Program4a_PES2UG19CS076 - Notepad
File Edit Format View Help
.text
LDR R0,=0b11110000101001011111000010100100
MOV R1,#32
MOV R2,#0
MOV R3,#0
loop:
MOVS R0,R0,LSR #1
ADDCS R2,R2,#1
ADDCC R3,R3,#1
SUB R1,R1,#1
CMP R1,#0
BNE loop
SWI 0x11
.end
```

II. Output Screen Shot



III. Output Table for the program

r1	32	
r2	After execution	15 (=0F in hex)
r3	After execution	17 (=11 in hex)

Microprocessor and Computer Architecture Laboratory

UE19CS256

4th Semester, Academic Year 2020-21

Date: 9/2/2021

Name: B.Pravena	SRN: PES2UG19CS076	Section: B
-----------------	--------------------	------------

Week# 3

Program Number: 4b

Write an ALP to find the number of zeroes, positive and negative numbers in a given array

I. ARM Assembly Code for the program.

```
Week3_Program4b_PES2UG19CS076 - Notepad
File Edit Format View Help
.text
LDR R0,=A
MOV R1,#9
MOV R5,#0
MOV R4,#0
MOV R3,#0
loop1:
LDR R2,[R0]
CMP R2,#0
BEQ zero
BGT positive
BLT negative
loop2:
SUB R1,R1,#1
ADD R0,R0,#4
CMP R1,#0
BEQ end
B loop1
zero:
ADD R4,R4,#1
B loop2
positive:
ADD R5,R5,#1
B loop2
negative:
ADD R3,R3,#1
B loop2
end:
SWI 0x11
.data
A: .word 20,-10,0,-40,50,0,22,-14,99
.end
```

II. Output Screen Shot

The screenshot shows the ARMSim - The ARM Simulator interface. The title bar indicates it is from the Dept. of Computer Science. The menu bar includes File, View, Cache, Debug, Watch, and Help. The main window is divided into two panes. The left pane, titled 'RegistersView', shows the state of various registers. The right pane displays the assembly code for 'Week3_Program4b_PES2UG19CS076.s'.

RegistersView:

- General Purpose: Floating Point
- Hexadecimal (selected)
- Unsigned Decimal
- Signed Decimal

Register values (R0 to R15):

- R0: 00001080
- R1: 00000000
- R2: 00000063
- R3: 00000003
- R4: 00000002
- R5: 00000004
- R6: 00000000
- R7: 00000000
- R8: 00000000
- R9: 00000000
- R10 (s1): 00000000
- R11 (fp): 00000000
- R12 (ip): 00000000
- R13 (sp): 00000000
- R14 (lr): 00001058
- R15 (pc): 00000008

CPSR Register:

- Negative (N): 0
- Zero (Z): 1
- Carry (C): 1
- Overflow (V): 0
- IRQ Disable: 1
- FIQ Disable: 1
- Thumb (T): 0

Assembly Code (Week3_Program4b_PES2UG19CS076.s):

```
.text
00001000:E59F0050 LDR R0,=A
00001004:E3A01009 MOV R1,#9
00001008:E3A05000 MOV R5,#0
0000100C:E3A04000 MOV R4,#0
00001010:E3A03000 MOV R3,#0
00001014: loop1:
00001014:E5902000 LDR R2,[R0]
00001018:E3520000 CMP R2,#0
0000101C:0A000006 BEQ zero
00001020:CA000007 BGT positive
00001024:BA000008 BLT negative
00001028: loop2:
00001028:E2411001 SUB R1,R1,#1
0000102C:E2800004 ADD R0,R0,#4
00001030:E3510000 CMP R1,#0
00001034:0A000006 BEQ end
00001038:EAF0FFF5 B loop1
0000103C: zero:
0000103C:E2844001 ADD R4,R4,#1
00001040:EAF0FFF8 B loop2
00001044: positive:
00001044:E2855001 ADD R5,R5,#1
00001048:EAF0FFF6 B loop2
0000104C: negative:
0000104C:E2833001 ADD R3,R3,#1
00001050:EAF0FFF4 B loop2
00001054: end:
00001054:EF000011 SWI 0x11
.data
0000105C: A: .word 20,-10,0,-40,50,0,22,-14,99
.end
```

III. Output Table for the program

R3 (no. of -ve numbers)	3
R4 (no. of 0's)	2
R5 (no. of +ve numbers)	4

Microprocessor and Computer Architecture Laboratory

UE19CS256

4th Semester, Academic Year 2020-21

Date: 9/2/2021

Name: B.Pravena	SRN: PES2UG19CS076	Section: B
-----------------	--------------------	------------

Week# ____3____

Program Number: ____5__

Write an ALP to check whether a given number is present in array using Linear Search (Without SWI 0x02), if found move +1 to R6 and key position to R7 else move -1 to R6 (if number not found)

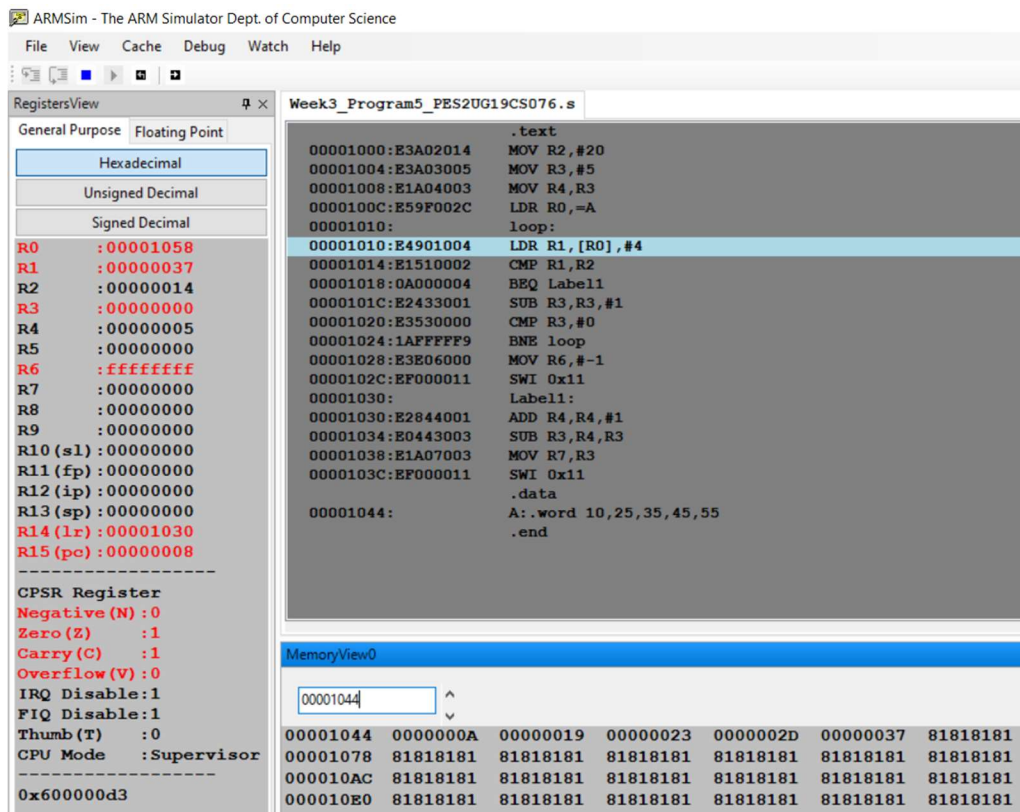
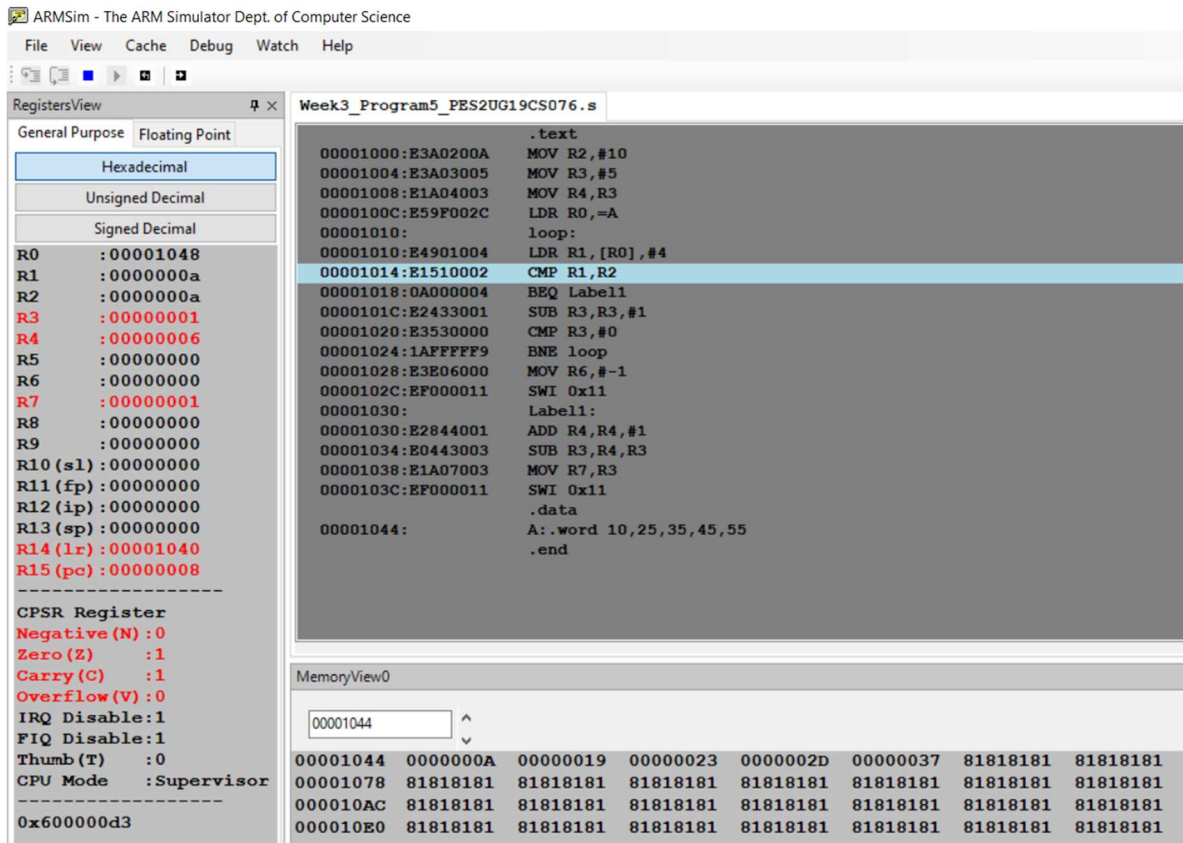
I. ARM Assembly Code for the program.

Week3_Program5_PES2UG19CS076 - Notepad

File Edit Format View Help

```
.text
MOV R2,#10
MOV R3,#5
MOV R4,R3
LDR R0,=A
loop:|
LDR R1,[R0],#4
CMP R1,R2
BEQ Label1
SUB R3,R3,#1
CMP R3,#0
BNE loop
MOV R6,#-1
SWI 0x11
Label1:
ADD R4,R4,#1
SUB R3,R4,R3
MOV R7,R3
SWI 0x11
.data
A:.word 10,25,35,45,55
.end
```

II. Output Screen Shot



III. Output Table for the program

Registers		Hexadecimal Value
When key element is found		
R0		0x1044
R2	10	0x0a
R7	1	0x01
When key element is not found		
R2	20	0x14
R6	-1	0xffffffff

Microprocessor and Computer Architecture Laboratory

UE19CS256

4th Semester, Academic Year 2020-21

Date: 9/2/2021

Name: B.Pravena	SRN: PES2UG19CS076	Section: B
-----------------	--------------------	------------

Week# 3

Program Number: 6

Write an ALP to generate Fibonacci Series and store them in an array

I. ARM Assembly Code for the program.

```
Week3_Program6_PES2UG19CS076 - Notepad
File Edit Format View Help
.text
MOV R0,#6
LDR R1,=A
MOV R2,#0
MOV R3,#1
STR R2,[R1]
ADD R1,R1,#4
STR R3,[R1],#4
loop:
ADD R4,R2,R3
STR R4,[R1],#4
MOV R2,R3
MOV R3,R4
SUB R0,R0,#1
CMP R0,#0
BNE loop
SWI 0x11
.data
A:.word 0,0,0,0,0,0
.end
```

II. Output Screen Shot

The screenshot displays the execution output of the ARM assembly program. It is divided into three main sections: RegistersView, CPSR Register, and MemoryView.

RegistersView: Shows the state of 16 general-purpose registers (R0-R15). R0 is 00000000, R1 is 00001060, R2 is 00000008, R3 is 0000000d, R4 is 0000000d, R5 is 00000000, R6 is 00000000, R7 is 00000000, R8 is 00000000, R9 is 00000000, R10 (s1) is 00000000, R11 (fp) is 00000000, R12 (ip) is 00000000, R13 (sp) is 00000000, R14 (lr) is 0000103c, and R15 (pc) is 00000008.

CPSR Register: Shows the Current Program Status Register. Negative (N) is 0, Zero (Z) is 1, Carry (C) is 1, Overflow (V) is 0, IRQ Disable is 1, FIQ Disable is 1, and Thumb (T) is 0.

MemoryView: Shows the memory dump starting at address 00001040. The memory contains the following data: 00000000, 00000001, 00000001, 00000002, 00000003, 00000005, 00000008, 0000000D, and 81818181.

Assembly Code: The assembly code is displayed in the background, showing the instructions executed: MOV R0,#6, LDR R1,=A, MOV R2,#0, MOV R3,#1, STR R2,[R1], ADD R1,R1,#4, STR R3,[R1],#4, loop:, ADD R4,R2,R3, STR R4,[R1],#4, MOV R2,R3, MOV R3,R4, SUB R0,R0,#1, CMP R0,#0, BNE loop, SWI 0x11, .data, A:.word 0,0,0,0,0,0, .end.

III. Output Table for the program

R0	Fibonacci Count	6
R1	Address of A	
R2	Initially 0	
R3	Initially 1	
R4	1 st Iteration	$0+1=1$
R4	2 nd Iteration	$1+1=2$
R4	3 rd Iteration	$2+1=3$
R4	4 th Iteration	$3+2=5$
R4	5 th Iteration	$5+3=8$
R4	6 th Iteration	$8+5=13 = 0D$