PES UNIVERSITY

Department of Computer Science & Engineering

Microprocessor & Computer Architecture

MPCA-Laboratory/Assignment/Hands-on/Project

UE20CS252

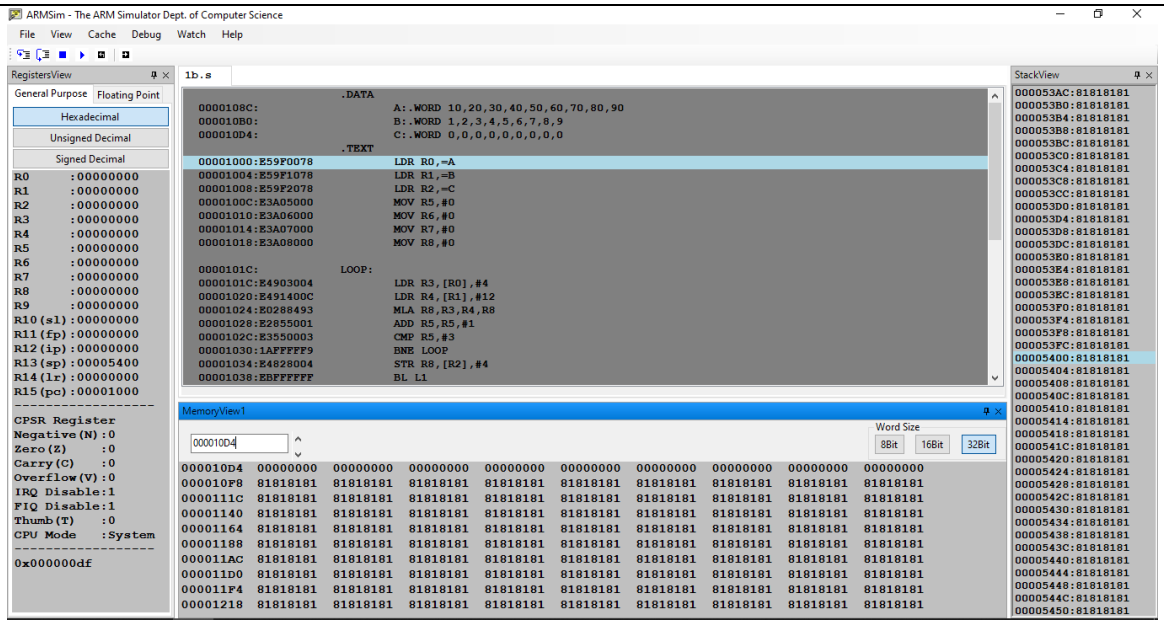
Sl. No	Programs

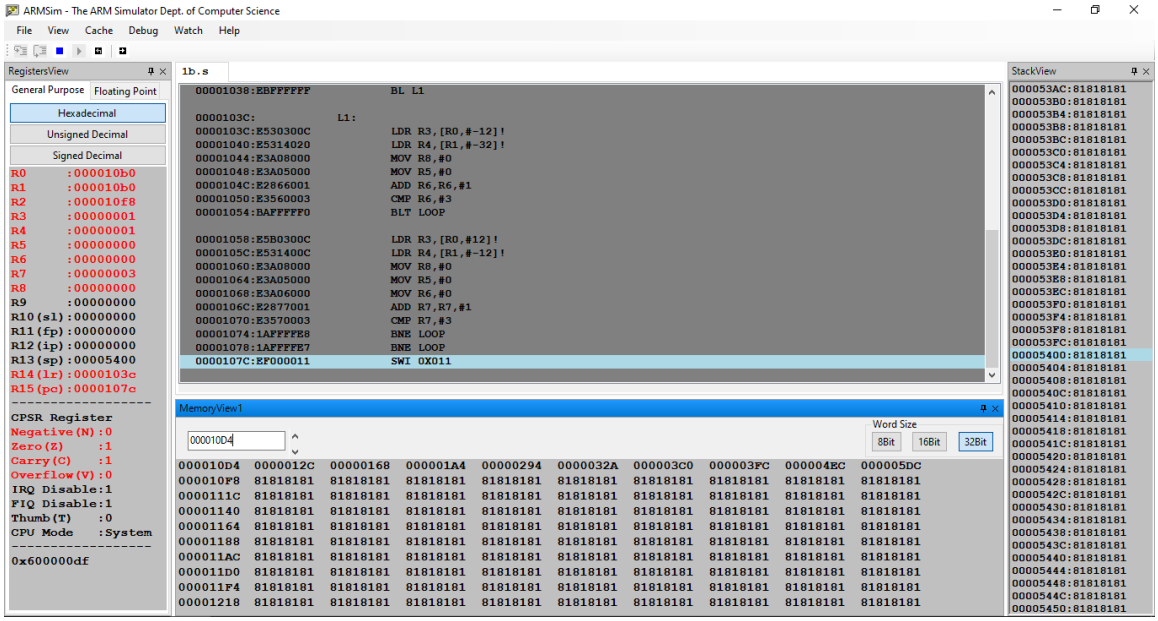
Week No.5	1. Write a program in ARM7TDMI-ISA to generate Fibonacci Series and store them in an array.
	
	2. Write a program in ARM7TDMI-ISA to find smallest number in an array of n 32 bit numbers. Display the element if found.
	
	3. Write a program in ARM7TDMI-ISA to add 2 matrices of order3.
	i.e., Implement c[i][j]= a[i][j] + b[i][j].
	
	4. Write a program in ARM7TDMI-ISA to transfer a block of 256 words stored at memory location X to memory location Y using Load Multiple and Store Multiple instructions. The rate of transfer is 32 bytes.
	
	Student exercises:
	1. Write a program in ARM7TDMI-ISA to multiply 2 matrices of order3.
	i.e., implement c[i][j]=c[i][j] + a[i][j] x b[i][j].
	a. Use MLA instruction
	
	.DATA
	A:.WORD 10,20,30,40,50,60,70,80,90
	B:.WORD 1,2,3,4,5,6,7,8,9
	C:.WORD 0,0,0,0,0,0,0,0,0
	.TEXT
	LDR R0,=A
	LDR R1,=B
	LDR R2,=C
	MOV R5,#0
	MOV R6,#0
	MOV R7,#0
	MOV R8,#0

```
LOOP:
    LDR R3,[R0],#4
    LDR R4,[R1],#12
    MLA R8,R3,R4,R8
    ADD R5,R5,#1
    CMP R5,#3
    BNE LOOP
    STR R8,[R2],#4
    BL L1

L1:
    LDR R3,[R0,#-12]!
    LDR R4,[R1,#-32]!
    MOV R8,#0
    MOV R5,#0
    ADD R6,R6,#1
    CMP R6,#3
    BLT LOOP

    LDR R3,[R0,#12]!
    LDR R4,[R1,#-12]!
    MOV R8,#0
    MOV R5,#0
    MOV R6,#0
    ADD R7,R7,#1
    CMP R7,#3
    BNE LOOP
    BNE LOOP
    SWI 0X011
```





b. Use MUL instruction

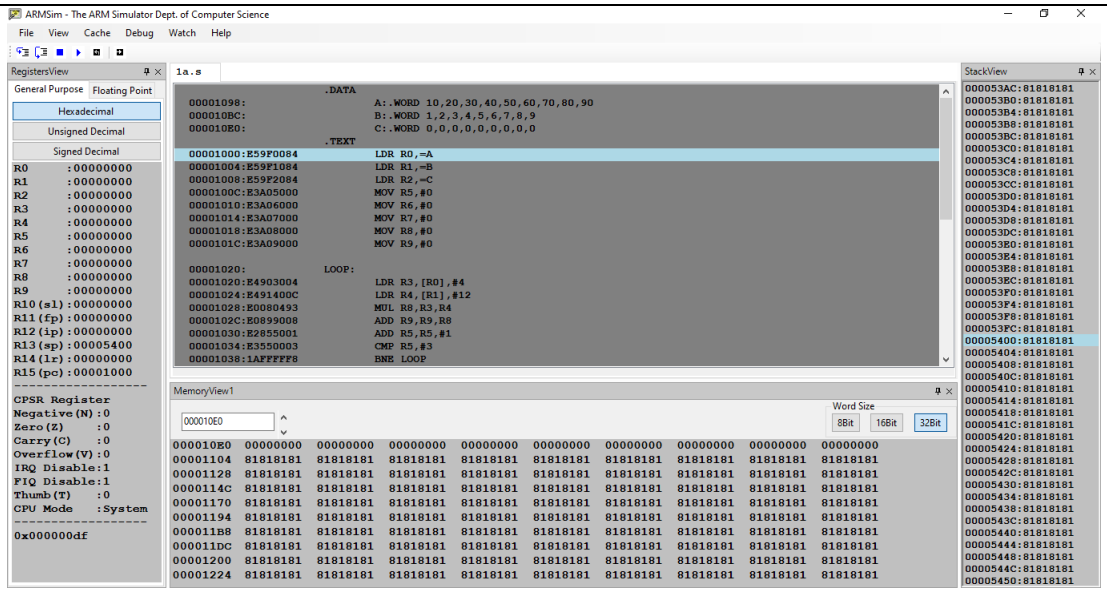
```
.DATA
        A:.WORD 10,20,30,40,50,60,70,80,90
        B:.WORD 1,2,3,4,5,6,7,8,9
        C:.WORD 0,0,0,0,0,0,0,0,0
.TEXT
        LDR R0,=A
        LDR R1,=B
        LDR R2,=C
        MOV R5,#0
        MOV R6,#0
        MOV R7,#0
```

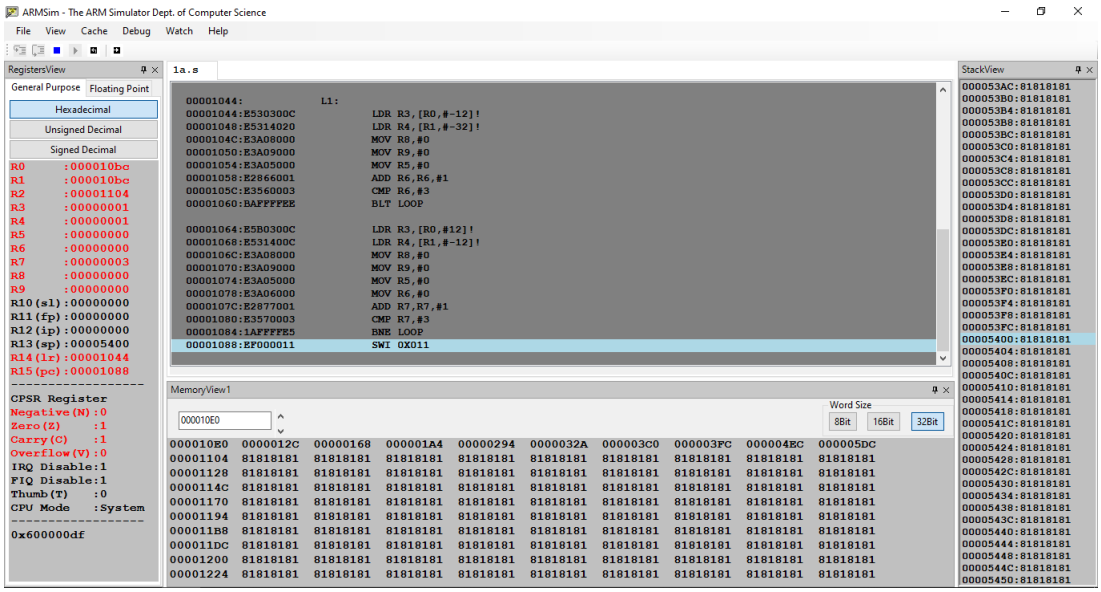
```
        MOV R8,#0
        MOV R9,#0

LOOP:
        LDR R3,[R0],#4
        LDR R4,[R1],#12
        MUL R8,R3,R4
        ADD R9,R9,R8
        ADD R5,R5,#1
        CMP R5,#3
        BNE LOOP
        STR R9,[R2],#4
        BL L1

L1:
        LDR R3,[R0,#-12]!
        LDR R4,[R1,#-32]!
        MOV R8,#0
        MOV R9,#0
        MOV R5,#0
        ADD R6,R6,#1
        CMP R6,#3
        BLT LOOP

        LDR R3,[R0,#12]!
        LDR R4,[R1,#-12]!
        MOV R8,#0
        MOV R9,#0
        MOV R5,#0
        MOV R6,#0
        ADD R7,R7,#1
        CMP R7,#3
        BNE LOOP
        SWI 0X011
```





2. Write a program in ARM7TDMI-ISA to find the NORM of a square matrix of order n.

3. Write a program in ARM7TDMI-ISA to find the ROWSUM of a matrix.

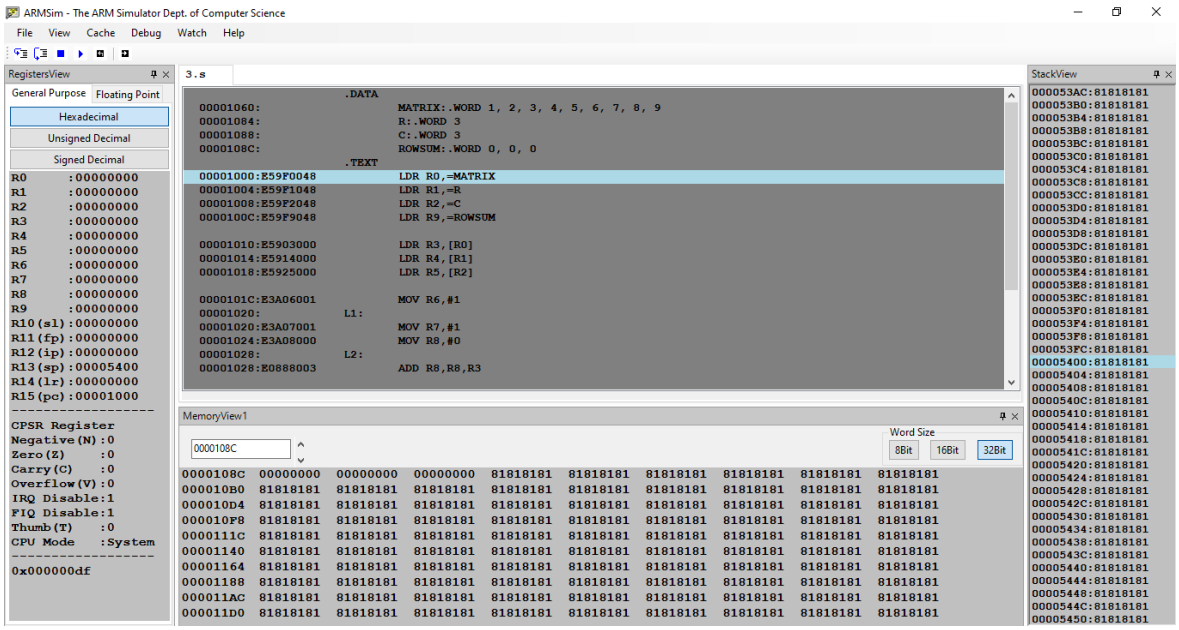
```
.DATA
    MATRIX:.WORD 1, 2, 3, 4, 5, 6, 7, 8, 9
    R:.WORD 3
    C:.WORD 3
    ROWSUM:.WORD 0, 0, 0
.TEXT
    LDR R0,=MATRIX
    LDR R1,=R
    LDR R2,=C
```

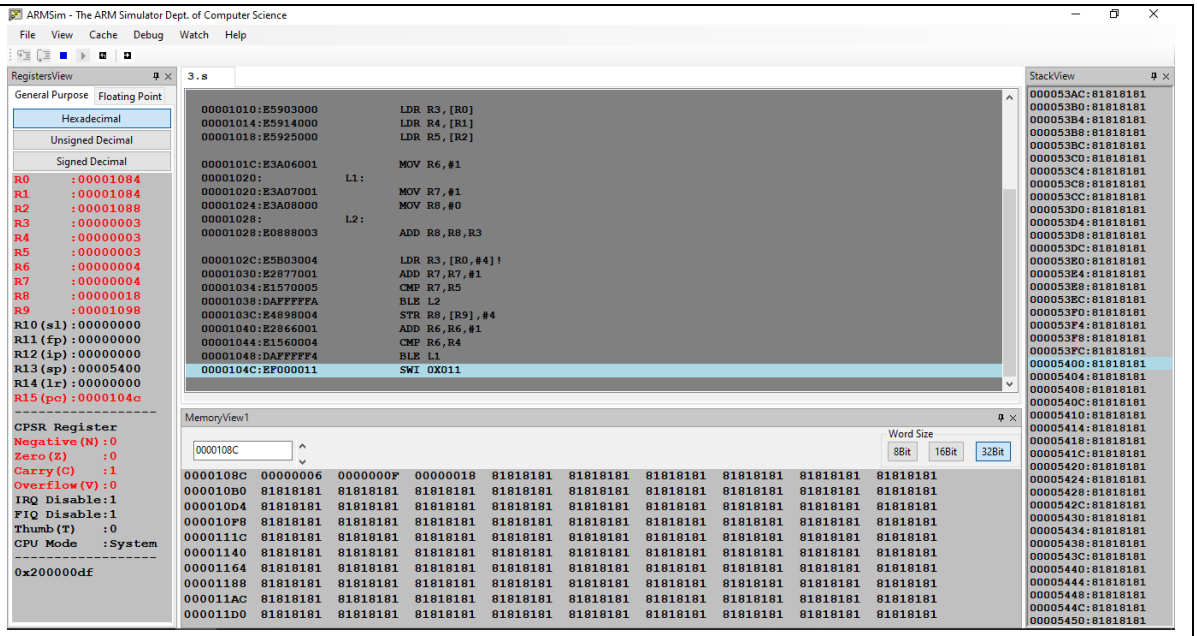
```
    LDR R9,=ROWSUM

    LDR R3,[R0]
    LDR R4,[R1]
    LDR R5,[R2]

    MOV R6,#1
L1:
    MOV R7,#1
    MOV R8,#0
L2:
    ADD R8,R8,R3

    LDR R3,[R0,#4]!
    ADD R7,R7,#1
    CMP R7,R5
    BLE L2
    STR R8,[R9],#4
    ADD R6,R6,#1
    CMP R6,R4
    BLE L1
    SWI 0X011
```




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 :00001084
R1 :00001084
R2 :00001088
R3 :00000003
R4 :00000003
R5 :00000003
R6 :00000004
R7 :00000004
R8 :00000018
R9 :00001098
R10(sl):00000000
R11(fp):00000000
R12(ip):00000000
R13(sp):00005400
R14(lr):00000000
R15(pc):0000104c
CPSR Register
Negative(N):0
Zero(Z) :0
Carry(C) :1
Overflow(V):0
IRQ Disable:1
FIQ Disable:1
Thumb(T) :0
CPU Mode :System
0x200000df
3.s
00001010:E5903000 LDR R3,[R0]
00001014:E5914000 LDR R4,[R1]
00001018:E5925000 LDR R5,[R2]
0000101C:E3A06001 MOV R6,#1
00001020: L1:
00001020:E3A07001 MOV R7,#1
00001024:E3A08000 MOV R8,#0
00001028: L2:
00001028:E0888003 ADD R8,R8,R3
0000102C:E5B03004 LDR R3,[R0,#4]!
00001030:E2877001 ADD R7,R7,#1
00001034:E1570005 CMP R7,R5
00001038:DAFFFFFA BLE L2
0000103C:E4898004 STR R8,[R9],#4
00001040:E2866001 ADD R6,R6,#1
00001044:E1560004 CMP R6,R4
00001048:DAFFFFF4 BLE L1
0000104C:EF000011 SWI 0X011
MemoryView1
0000108C
Word Size
8Bit 16Bit 32Bit
0000108C 00000006 0000000F 00000018 81818181 81818181 81818181 81818181 81818181 81818181
000010B0 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181
000010D4 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181
000010F8 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181
0000111C 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181
00001140 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181
00001164 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181
00001188 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181
000011AC 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181
000011D0 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181
StackView
000053AC:81818181
000053B0:81818181
000053B4:81818181
000053B8:81818181
000053BC:81818181
000053C0:81818181
000053C4:81818181
000053C8:81818181
000053CC:81818181
000053D0:81818181
000053D4:81818181
000053D8:81818181
000053DC:81818181
000053E0:81818181
000053E4:81818181
000053E8:81818181
000053EC:81818181
000053F0:81818181
000053F4:81818181
000053F8:81818181
000053FC:81818181
00005400:81818181
00005404:81818181
00005408:81818181
0000540C:81818181
00005410:81818181
00005414:81818181
00005418:81818181
0000541C:81818181
00005420:81818181
00005424:81818181
00005428:81818181
0000542C:81818181
00005430:81818181
00005434:81818181
00005438:81818181
0000543C:81818181
00005440:81818181
00005444:81818181
00005448:81818181
0000544C:81818181
00005450:81818181