Microprocessor and Computer Architecture Laboratory

UE19CS256

4th Semester, Academic Year 2020-21

Date: 15/2/21

Name: Adithya M S	SRN: PES1UG19CS027	Section: A

Week#____4_____   Program Number: ___1___

Write an ALP to add corresponding elements of an array.
Ex: for(i=0;i<n;i++)

C[i]=a[i]+b[i]

Code:

.data

a:.word 10,20,30,40,50

b:.word 10,20,30,40,50

c:.word 0,0,0,0,0

.text

ldr r0,=a

ldr r1,=b

```
ldr r2,=c

mov r3,#5

loop:

        ldr r4,[r0],#4

        ldr r5,[r1],#4

        add r6,r4,r5

        str r6,[r2],#4

        sub r3,r3,#1

        cmp r3,#0

        bne loop

swi 0x11
```
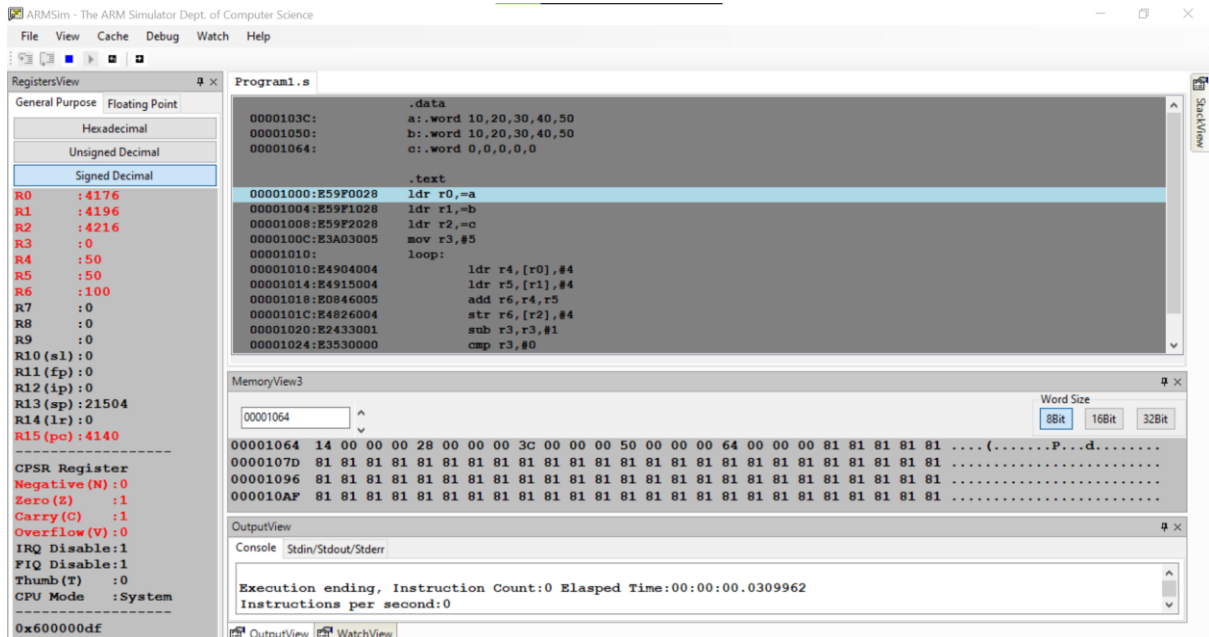
Write an ALP to find the product of corresponding elements of an array.

Ex: for(i=0;i<n;i++)

C[i]=a[i]*b[i]

Code:

```
.data

a:.word 10,20,30,40,50

b:.word 10,20,30,40,50

c:.word 0,0,0,0,0


.text

ldr r0,=a

ldr r1,=b

ldr r2,=c

mov r3,#5

loop:

        ldr r4,[r0],#4

        ldr r5,[r1],#4

        mul r6,r4,r5

        str r6,[r2],#4

        sub r3,r3,#1
```

cmp r3,#0

bne loop

swi 0x11

Write an ALP to perform Convolution operation using MUL instruction (Addition of multiplication of respective numbers of loc A and loc B)

Code:

```
.data

a:.word 10,20,30,40,50

b:.word 10,20,30,40,50

c:.word 0,0,0,0,0


.text

ldr r0,=a

ldr r1,=b

ldr r2,=c

mov r3,#5

mov r7,#0

loop:

        ldr r4,[r0],#4

        ldr r5,[r1],#4

        mul r6,r4,r5

        add r7,r7,r6

        str r7,[r2],#4
```

sub r3,r3,#1

cmp r3,#0

bne loop

swi 0x11

Write an ALP to perform Convolution using MLA instruction (Addition of multiplication of respective numbers of loc A and loc B).

Ex: for(i=0;i<n;i++)

C[i]=C[i]+a[i]*b[i]

Code:

.data

a:.word 10,20,30,40,50

b:.word 10,20,30,40,50

c:.word 0,0,0,0,0

.text

ldr r0,=a

ldr r1,=b

ldr r2,=c

mov r3,#5

mov r6,#0

loop:

     ldr r4,[r0],#4

     ldr r5,[r1],#4

     mla r6,r4,r5,r6

str r6,[r2],#4

sub r3,r3,#1

cmp r3,#0

bne loop

swi 0x11
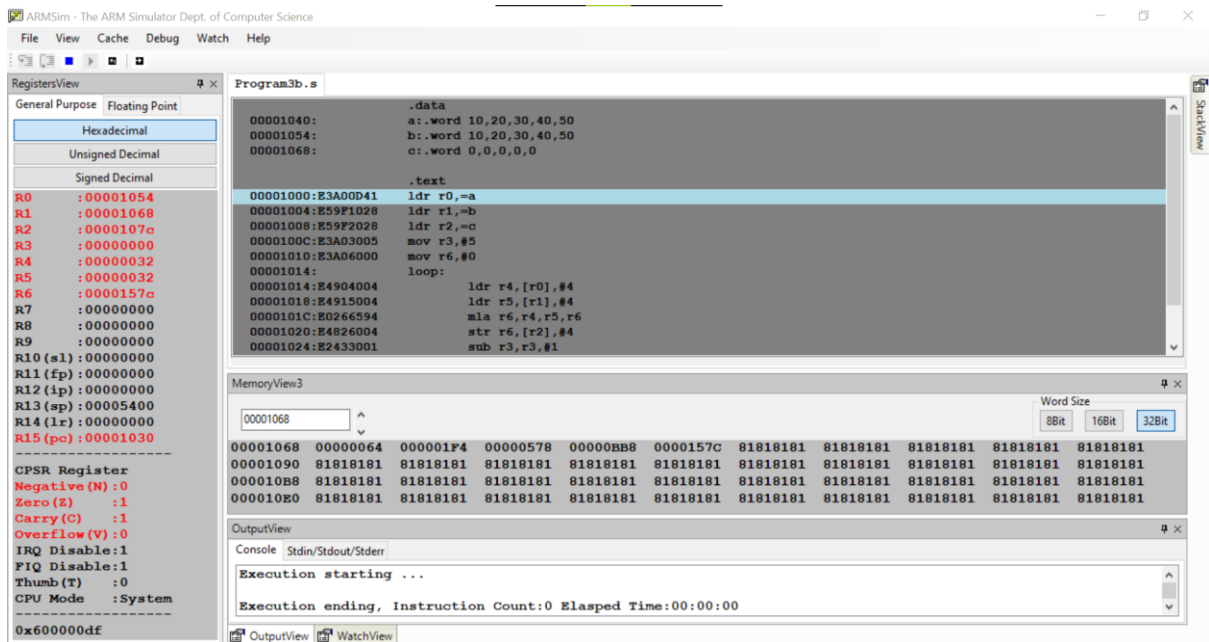
Consider an 2D array. Write an ALP using ARM7TDMI-ISA, to retrieve / access any element from the array.

Code:

.data

a:.word 10,20,30,40,50,60,70,80,90

b:.word 0,0,0

.text

ldr r0,=a

mov r1,#1 ;Row number

mov r2,#3 ; Number of elements in a row

mov r3,#1 ; Column number

mov r4,#4

mla r5,r1,r2,r3 ; Accessing 5th element

mul r5,r4,r5

ldr r6,=b

ldr r7,[r0,r5]

str r7,[r6],#4

mov r1,#2

mov r3,#1

mla r5,r1,r2,r3 ; Accessing 8th element

mul r5,r4,r5

ldr r7,[r0,r5]

str r7,[r6],#4

mov r1,#0

mov r3,#0

mla r5,r1,r2,r3 ; Accessing 1st element

mul r5,r4,r5

ldr r7,[r0,r5]

str r7,[r6],#4

swi 0x11
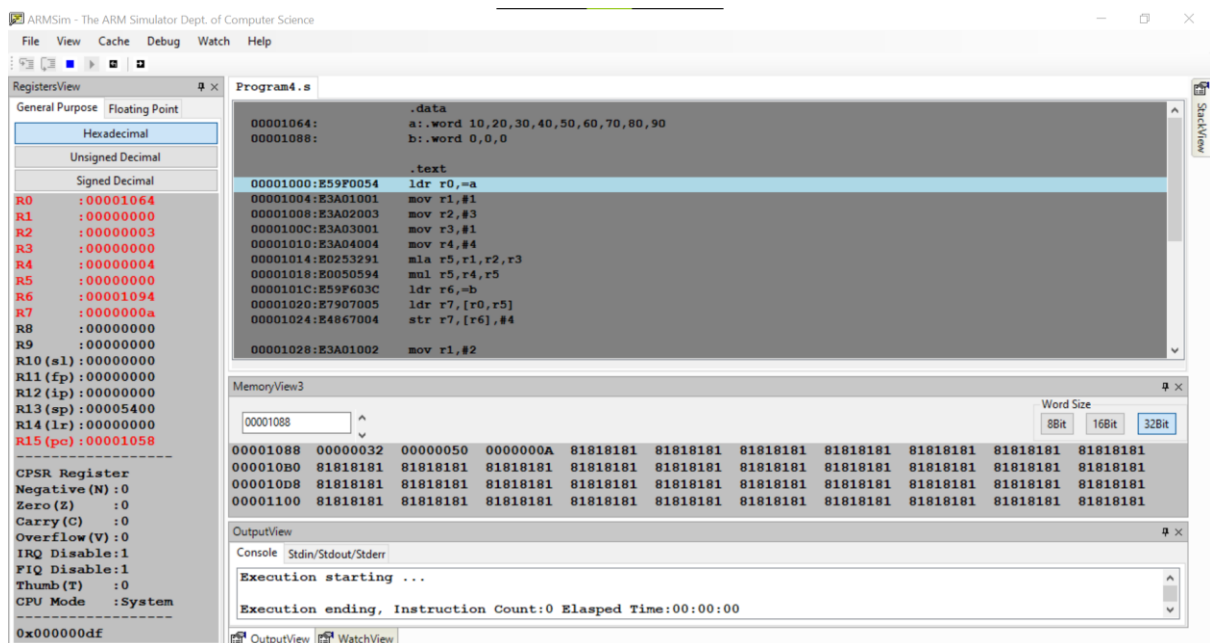
Write an ALP to add corresponding elements of an 2D - array.
Ex: for(i=0;i<n;i++)

     for(j=0;j<n;j++)

        c[i][j]=a[i][j]+b[i][j]

Code:

```
.data

a:.byte 1,2,3,4,2,3,1,2,1

b:.byte 1,1,1,2,2,2,3,3,3

c:.byte 0,0,0,0,0,0,0,0,0


.text

ldr r0,=a

ldr r1,=b

ldr r2,=c

mov r3,#0

mov r4,#3

loop1:

    mov r5,#0

    loop2:

        mla r6,r3,r4,r5
```

```
        ldrb r7,[r0,r6]

        ldrb r8,[r1,r6]

        add r9,r7,r8

        strb r9,[r2,r6]

        cmp r5,#2

        add r5,r5,#1

        bne loop2

    cmp r3,#2

    add r3,r3,#1

    bne loop1


swi 0x11
```


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   :00001054
R1   :0000105d
R2   :00001066
R3   :00000003
R4   :00000003
R5   :00000003
R6   :00000008
R7   :00000001
R8   :00000003
R9   :00000004
R10(sl):00000000
R11(fp):00000000
R12(ip):00000000
R13(sp):00005400
R14(lr):00000000
R15(pc):00001044

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

0x600000df

Program5.s

```
                        .data
00001054:           a:.byte 1,2,3,4,2,3,1,2,1
0000105D:           b:.byte 1,1,1,2,2,2,3,3,3
00001066:           c:.byte 0,0,0,0,0,0,0,0,0

                        .text
00001000:E59F0040    ldr r0,=a
00001004:E59F1040    ldr r1,=b
00001008:E59F2040    ldr r2,=c
0000100C:E3A03000    mov r3,#0
00001010:E3A04003    mov r4,#3
00001014:            loop1:
00001014:E3A05000        mov r5,#0
00001018:                loop2:
00001018:E0265493            mla r6,r3,r4,r5
0000101C:E7D07006            ldrb r7,[r0,r6]
00001020:E7D18006            ldrb r8,[r1,r6]
```
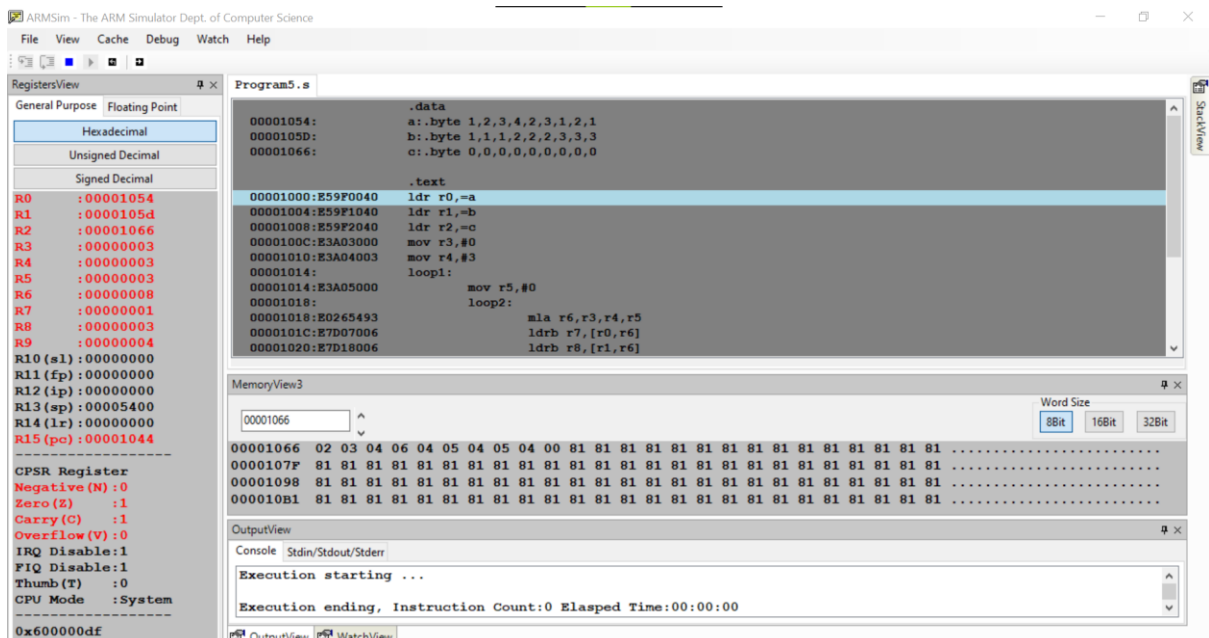
MemoryView3

Word Size | 8Bit | 16Bit | 32Bit |

00001066

```
00001066  02 03 04 06 04 05 04 05 04 00 81 81 81 81 81 81 81 81 81 81 81 81 81 81 81 81 81 ......................
0000107F  81 81 81 81 81 81 81 81 81 81 81 81 81 81 81 81 81 81 81 81 81 81 81 81 81 81 81 ......................
00001098  81 81 81 81 81 81 81 81 81 81 81 81 81 81 81 81 81 81 81 81 81 81 81 81 81 81 81 ......................
000010B1  81 81 81 81 81 81 81 81 81 81 81 81 81 81 81 81 81 81 81 81 81 81 81 81 81 81 81 ......................
```

OutputView

Console Stdin/Stdout/Stderr

Execution starting ...

Execution ending, Instruction Count:0 Elasped Time:00:00:00

OutputView   WatchView

Consider a 2D array. Write an ALP using ARM7TDMI-ISA, to find the sum of all the elements in the array.
Ex: Sum[i]=Sum[i]+a[i][j]

Code:

```
.data

a:.byte 1,2,3,4,2,3,1,2,1

b:.byte 0


.text

ldr r0,=a

ldr r1,=b

mov r2,#0

mov r3,#3

loop1:

      mov r4,#0

      loop2:

            mla r5,r2,r3,r4

            ldrb r6,[r0,r5]

            add r7,r6,r7

            cmp r4,#2

            add r4,r4,#1
```

bne loop2

cmp r2,#2

add r2,r2,#1

bne loop1

strb r7,[r1]

swi 0x11