Microprocessor and Computer Architecture Laboratory

UE19CS256

4th Semester, Academic Year 2020-21

Date:

Name: Nihal T M	SRN: PES2UG21CS333	Section: F

Week#____5_____ Program Number: ____1_

Write an ALP to read from a 2D array such that

B=a[i] [j]

I. ARM Assembly Code (1).

Code:

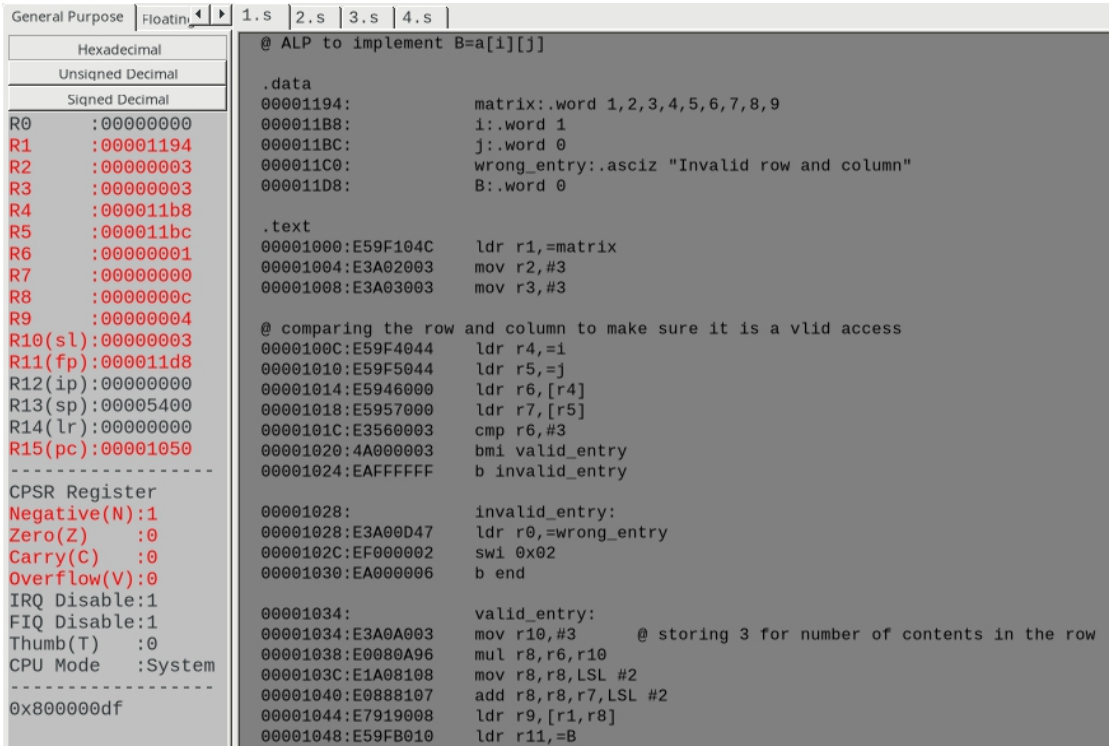
```
@ ALP to implement B=a[i][j]


.data
    matrix:.word 1,2,3,4,5,6,7,8,9
    i:.word 1
    j:.word 0
    wrong_entry:.asciz "Invalid row and column"
    B:.word 0
```

```
.text
	ldr r1,=matrix
	mov r2,#3
	mov r3,#3

	@ comparing the row and column to make sure it is a vlid access
	ldr r4,=i
	ldr r5,=j
	ldr r6,[r4]
	ldr r7,[r5]
	cmp r6,#3
	bmi valid_entry
	b invalid_entry

	invalid_entry:
		ldr r0,=wrong_entry
		swi 0x02
		b end

	valid_entry:
		mov r10,#3   @ storing 3 for number of contents in the row
```

```
        mul r8,r6,r10
        mov r8,r8,LSL #2
        add r8,r8,r7,LSL #2
        ldr r9,[r1,r8]
        ldr r11,=B
        str r9,[r11]
end:
        swi 0x11
    .end
```

II. Output Screen Shot (One Example of your choice)

Here i=1 and j=0; We are trying to access the first element in the second row of the 3x3 matrix. The element is 4 and is present in register r9.



Week#____5_____ Program Number: ____2_

Write an ALP to implement C[k]=A[i]+B[j]

I. ARM Assembly Code (1).

Code:

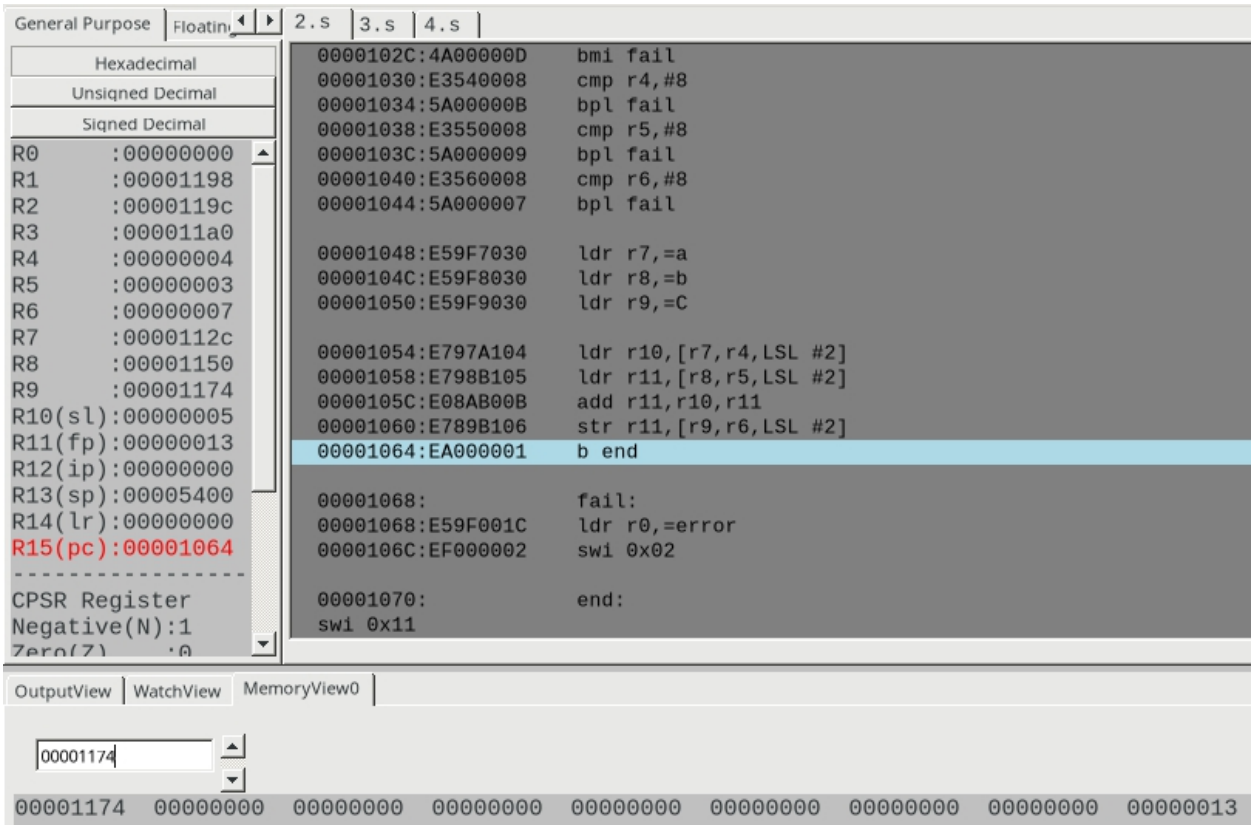
```
@ ALP to implement C[k]=a[i]+b[j]


.data
	a:.word 1,2,3,4,5,6,7,8,9
	b:.word 11,12,13,14,15,16,17,18,19
	C:.word 0,0,0,0,0,0,0,0,0
	i:.word 4
	j:.word 3
	k:.word 7
	error:.asciz "Invalid Indexes!"


.text
	ldr r1,=i
	ldr r2,=j
	ldr r3,=k
	ldr r4,[r1]
	ldr r5,[r2]
	ldr r6,[r3]
```

```
@ making sure we don't access illegal memory
cmp r4,#0
bmi fail
cmp r5,#0
bmi fail
cmp r6,#0
bmi fail
cmp r4,#8
bpl fail
cmp r5,#8
bpl fail
cmp r6,#8
bpl fail

ldr r7,=a
ldr r8,=b
ldr r9,=C

ldr r10,[r7,r4,LSL #2]
ldr r11,[r8,r5,LSL #2]
add r11,r10,r11
str r11,[r9,r6,LSL #2]
```

```
b end


fail:
    ldr r0,=error
    swi 0x02
end:
    swi 0x11
```

II. Output Screen Shot (One Example of your choice)

The output it stored in register r11 and is then stored into memory as can be seen at the bottom right corner.



Week#____5_____ Program Number: ____3_

Write an ALP to implement Sum[i] +=a[i][j]

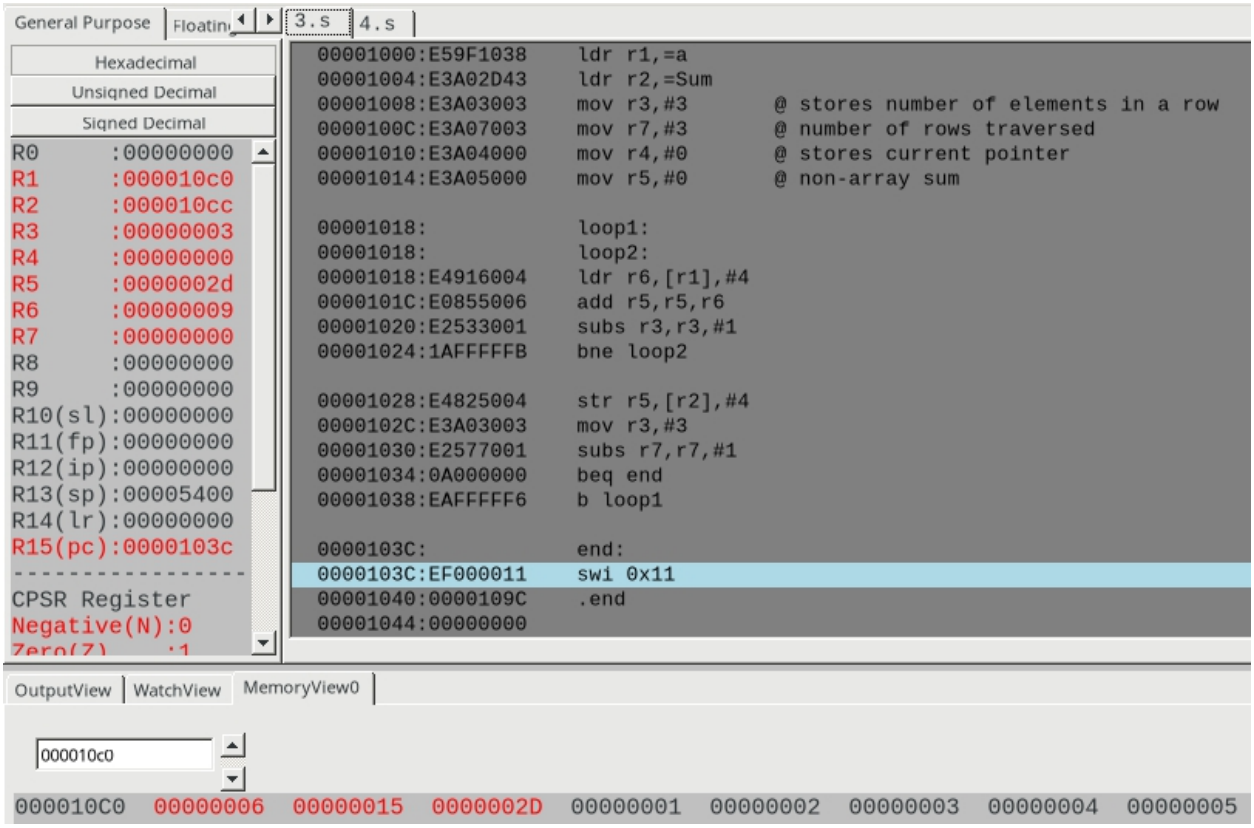
I. ARM Assembly Code (1).

Code:

```
@ ALP to calculate the sum of elements of each row in the matrix

.data
   a:.word 1,2,3,4,5,6,7,8,9
   Sum:.word 0,0,0

.text
   ldr r1,=a
   ldr r2,=Sum
   mov r3,#3    @ stores number of elements in a row
   mov r7,#3    @ number of rows traversed
   mov r4,#0    @ stores current pointer
   mov r5,#0    @ non-array sum

   loop1:
        loop2:
             ldr r6,[r1],#4
             add r5,r5,r6
             subs r3,r3,#1
             bne loop2

        str r5,[r2],#4
```

```
        mov r3,#3
        subs r7,r7,#1
        beq end
        b loop1

    end:
        swi 0x11
.end
```

II. Output Screen Shot (One Example of your choice)



Week#____5_____ Program Number: ____4_

Write an ALP to implement c[k] = a[i] * b[j]

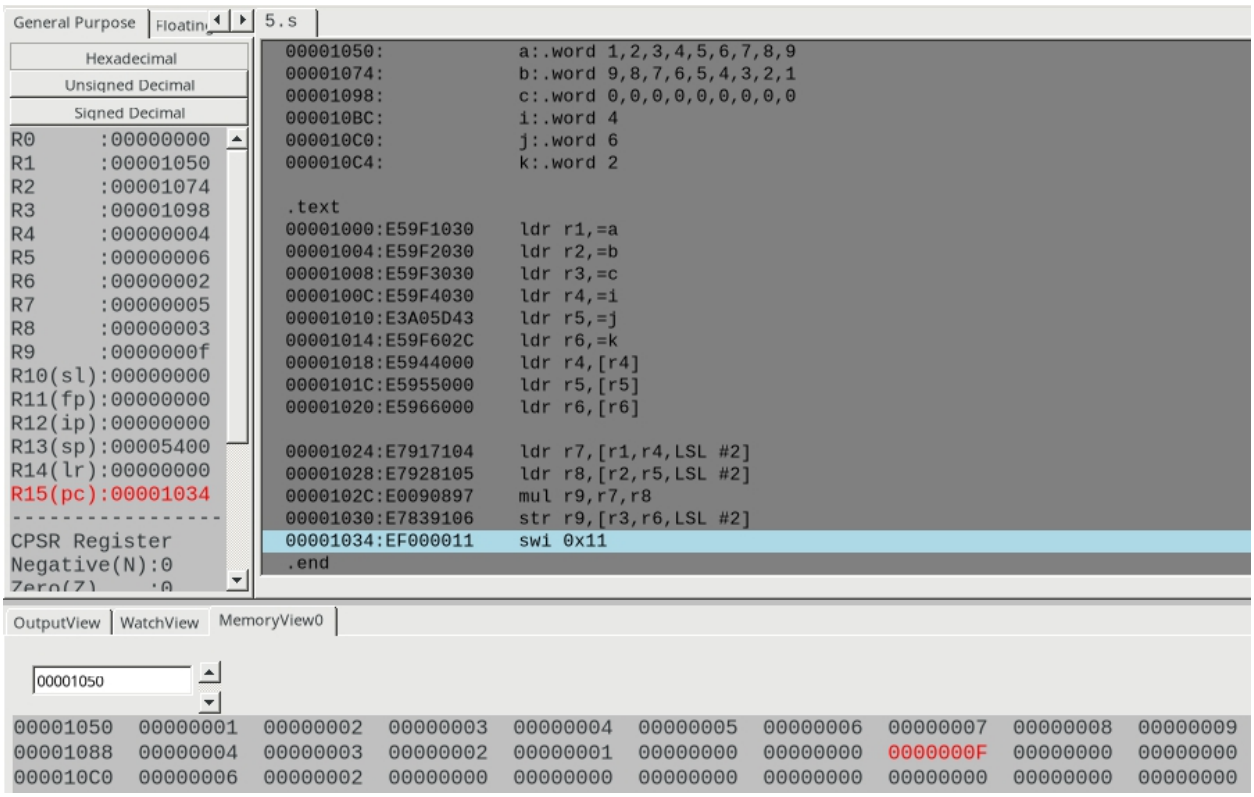
I. ARM Assembly Code (1).

Code:

```
@ ALP to implement c[k]=a[i]*b[j]

.data
   a:.word 1,2,3,4,5,6,7,8,9
   b:.word 9,8,7,6,5,4,3,2,1
   c:.word 0,0,0,0,0,0,0,0,0
   i:.word 4
   j:.word 6
   k:.word 2

.text
   ldr r1,=a
   ldr r2,=b
   ldr r3,=c
   ldr r4,=i
   ldr r5,=j
   ldr r6,=k
   ldr r4,[r4]
   ldr r5,[r5]
   ldr r6,[r6]

   ldr r7,[r1,r4,LSL #2]
   ldr r8,[r2,r5,LSL #2]
```

```
    mul r9,r7,r8
    str r9,[r3,r6,LSL #2]
    swi 0x11
.end
```

II. Output Screen Shot (One Example of your choice)



Week#____5_____ Program Number: ____5_

Write an ALP to implement C[i][j]=a[i][j]+b[i][j]

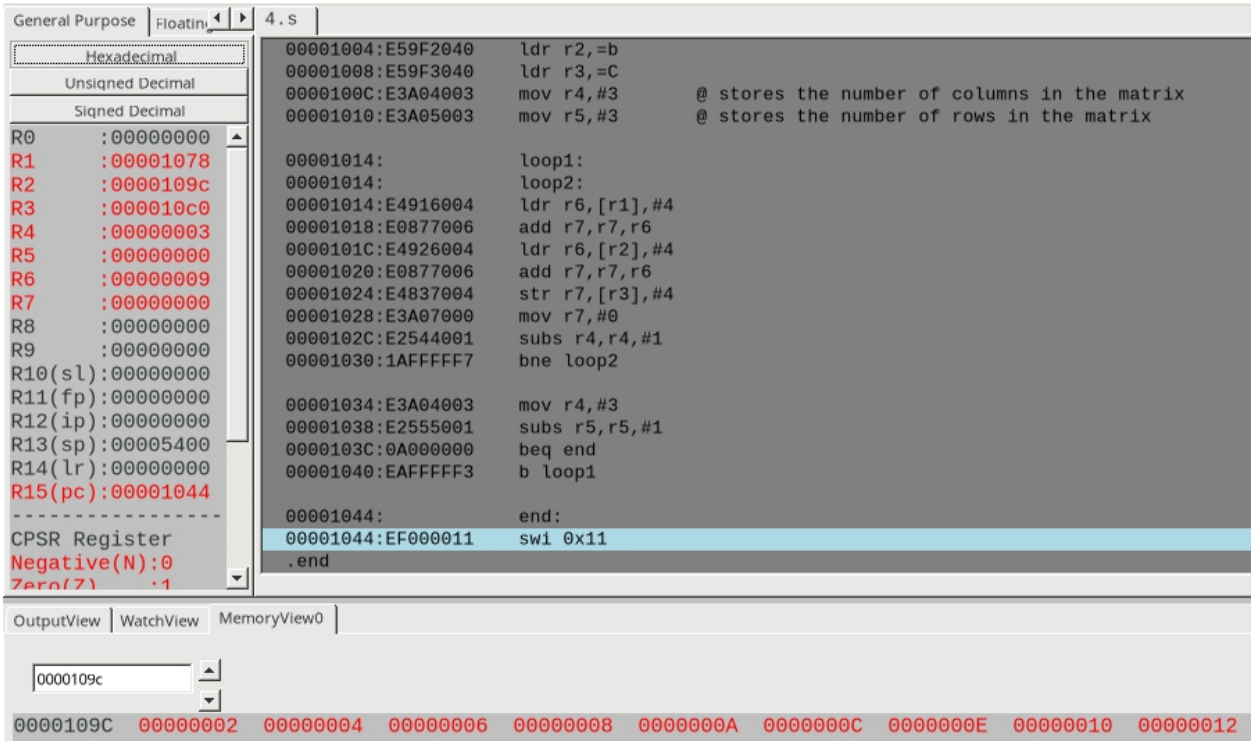
1. ARM Assembly Code (1).

Code:

```
@ ALP to implement C[i][j]=a[i][j]+b[i][j] (basically matrix addition)

.data
   a:.word 1,2,3,4,5,6,7,8,9
   b:.word 1,2,3,4,5,6,7,8,9
   C:.word 0,0,0,0,0,0,0,0,0

.text
   ldr r1,=a
   ldr r2,=b
   ldr r3,=C
   mov r4,#3    @ stores the number of columns in the matrix
   mov r5,#3    @ stores the number of rows in the matrix

   loop1:
        loop2:
             ldr r6,[r1],#4
             add r7,r7,r6
             ldr r6,[r2],#4
             add r7,r7,r6
             str r7,[r3],#4
```

```
        mov r7,#0
        subs r4,r4,#1
        bne loop2

    mov r4,#3
    subs r5,r5,#1
    beq end
    b loop1

end:
    swi 0x11
.end
```

2. Output Screen Shot (One Example of your choice)



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:
Name: Nihal T M
SRN: PES2UG21CS333
Section: F
Date: 18/03/2023