

# Microprocessor and Computer Architecture Laboratory

UE19CS256

4th Semester, Academic Year 2020-21

Date: 17/2/2021

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

Week# 4 Program Number: 1

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

I. ARM Assembly Code

```
Week4_Program1_PES2UG19CS076 - Notepad
File Edit Format View Help
.text
LDR R0,=A
LDR R1,=B
LDR R2,=C
MOV R6,#5
loop:
LDR R3,[R0],#4
LDR R4,[R1],#4
ADDS R5,R3,R4
STR R5,[R2],#4
SUB R6,R6,#1
CMP R6,#0
BNE loop
SWI 0x11
.data
A:.word 5,10,15,20,25
B:.word 5,10,15,20,25
C:.word 0,0,0,0,0
.end
```

## II. Output Screen Shot

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

**RegistersView:**

- General Purpose:** Floating Point, Hexadecimal, Unsigned Decimal, Signed Decimal.
- Registers:**
  - R0: 00001050
  - R1: 00001064
  - R2: 00001078
  - R3: 00000019
  - R4: 00000019
  - R5: 00000032
  - R6: 00000000
  - R7: 00000000
  - R8: 00000000
  - R9: 00000000
  - R10 (s1): 00000000
  - R11 (fp): 00000000
  - R12 (ip): 00000000
  - R13 (sp): 00000000
  - R14 (lr): 00001030
  - 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
  - CPU Mode: Supervisor

**MemoryView0:**

- Address:** 00001064
- Memory Dump:**

00001064	0000000A	00000014	0000001E	00000028	00000032	81818181	81818181
00001098	81818181	81818181	81818181	81818181	81818181	81818181	81818181

**Assembly Code (Week4\_Program1\_PES2UG19CS076.s):**

```
.text
00001000:E59F0028 LDR R0,=A
00001004:E59F1028 LDR R1,=B
00001008:E59F2028 LDR R2,=C
0000100C:E3A06005 MOV R6,#5
00001010: loop:
00001010:E4903004 LDR R3,[R0],#4
00001014:E4914004 LDR R4,[R1],#4
00001018:E0935004 ADDS R5,R3,R4
0000101C:E4825004 STR R5,[R2],#4
00001020:E2466001 SUB R6,R6,#1
00001024:E3560000 CMP R6,#0
00001028:1AFFFFF8 BNE loop
0000102C:EF000011 SWI 0x11
.data
0000103C: A:.word 5,10,15,20,25
00001050: B:.word 5,10,15,20,25
00001064: C:.word 0,0,0,0,0
.end
```

## III. Output Table for the program

Values of C after execution	
0x0A	10
0x14	20
0x1E	30
0x28	40
0x32	50

# Microprocessor and Computer Architecture Laboratory

UE19CS256

4th Semester, Academic Year 2020-21

Date: 17/2/2021

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

Week# 4 Program Number: 2

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

I. ARM Assembly Code

```
Week4_Program2_PES2UG19CS076 - Notepad
File Edit Format View Help
.text
LDR R0,=A
LDR R1,=B
LDR R2,=C
MOV R6,#5
loop:
CMP R6,#0
BEQ end
SUB R6,R6,#1
LDR R3,[R0],#4
LDR R4,[R1],#4
MUL R5,R3,R4
STR R5,[R2],#4
B loop
end:
SWI 0x11
.data
A:.word 5,10,11,14,15|
B:.word 10,20,30,40,50
C:.word 0,0,0,0,0
.end
```

## II. Output Screen Shot

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

**RegistersView:** The window title is "RegistersView". It has tabs for "General Purpose" and "Floating Point". The "General Purpose" tab is selected, showing a list of registers (R0-R15) and their values in hexadecimal. The values are: R0: 00001054, R1: 00001068, R2: 0000107c, R3: 0000000f, R4: 00000032, R5: 000002ee, R6: 00000000, R7: 00000000, R8: 00000000, R9: 00000000, R10 (s1): 00000000, R11 (fp): 00000000, R12 (ip): 00000000, R13 (sp): 00000000, R14 (lr): 00001034, R15 (pc): 00000008. Below the registers is the **CPSR Register** section, showing: Negative (N): 0, Zero (Z): 1, Carry (C): 1, Overflow (V): 0, IRQ Disable: 1, FIQ Disable: 1, Thumb (T): 0, and CPU Mode: Supervisor.

**MemoryView0:** The window title is "MemoryView0". It shows a list of memory addresses and their contents. The address 00001068 is selected, showing the value 00000032. The address 0000109C is also shown, showing the value 81818181. The window also displays the assembly code for the program, including labels like .text, .loop, .end, and .data.

## III. Output Table for the program

Contents of C after execution	
0x32	50
0xC8	200
0x014A	330
0x0230	560
0x02EE	750

# Microprocessor and Computer Architecture Laboratory

UE19CS256

4th Semester, Academic Year 2020-21

Date: 17/2/2021

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

Week# \_\_\_\_4\_\_\_\_

Program Number: \_\_\_\_3\_\_

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

## I. ARM Assembly Code

```
Week4_Program3a_PES2UG19CS076 - Notepad
File Edit Format View Help
.text
LDR R0,=A
LDR R1,=B
MOV R2,#4
MOV R5,#0
loop:
LDR R3,[R0],#4
LDR R4,[R1],#4
MUL R6,R3,R4
ADD R5,R5,R6
SUB R2,R2,#1
CMP R2,#0
BNE loop
SWI 0x11

.data
A:.word 10,20,30,40
B:.word 10,20,30,40
.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 various registers. The 'General Purpose' tab is selected, and the 'Hexadecimal' view is chosen. The registers R0 through R15 are listed, with their current values in hexadecimal. R15 (PC) is 00000008. Below the registers, the CPSR Register status is shown: Negative (N) is 0, Zero (Z) is 1, Carry (C) is 1, Overflow (V) is 0, IRQ Disable is 1, FIQ Disable is 1, Thumb (T) is 0, and CPU Mode is Supervisor.

**MemoryView0:** This window shows the memory contents starting at address 00001048. The memory is organized into a grid of 5 columns and 2 rows. The first row contains addresses 00001048, 0000000A, 00000014, 0000001E, and 00000028. The second row contains the corresponding memory values: 81818181, 81818181, 81818181, 81818181, and 81818181.

**Assembly View:** The main window displays the assembly code for the program. The code is as follows:

```
.text
00001000:E59F0028 LDR R0,=A
00001004:E59F1028 LDR R1,=B
00001008:E3A02004 MOV R2,#4
0000100C:E3A05000 MOV R5,#0
00001010:      loop:
00001010:E4903004 LDR R3,[R0],#4
00001014:E4914004 LDR R4,[R1],#4
00001018:E0060493 MUL R6,R3,R4
0000101C:E0855006 ADD R5,R5,R6
00001020:E2422001 SUB R2,R2,#1
00001024:E3520000 CMP R2,#0
00001028:1AFFFFF8 BNE loop
0000102C:EF000011 SWI 0x11

.data
00001038:      A:.word 10,20,30,40
00001048:      B:.word 10,20,30,40
.end
```

## III. Output Table for the program

R5	$(10*10)+(20*20)+(30*30)$ $+(40*40)+(50*50)$ $=3000=00000bb8$
----	---



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

### I. ARM Assembly Code

```
Week4_Program3b_PES2UG19CS076 - Notepad
File Edit Format View Help
.text
LDR R0,=A
LDR R1,=B
MOV R2,#4
MOV R5,#0
loop:
LDR R3,[R0],#4
LDR R4,[R1],#4
MLA R5,R3,R4,R5
SUB R2,R2,#1
CMP R2,#0
BNE loop
SWI 0x11

.data
A:.word 10,20,30,40
B:.word 10,20,30,40
.end
```

### II. Output Screen Shot

The screenshot displays an ARM assembly simulator interface. The top window, titled "RegistersView", shows the state of 16 registers (R0-R15) and the CPSR register. The "General Purpose" tab is selected, and the "Hexadecimal" view is chosen. The registers R0 through R9 are shown with their values in hexadecimal. R10 (s1), R11 (fp), R12 (ip), R13 (sp), R14 (lr), and R15 (pc) are also listed. The CPSR register is shown with its flags: 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.

The bottom window, titled "MemoryView0", shows the memory contents at addresses 00001044 and 00001048. The memory at 00001044 contains the word 0000000A, and the memory at 00001048 contains the word 0000001A. The memory at 0000001E contains the word 00000028.

The assembly code is displayed in the background, showing the following instructions:

```
.text
00001000:E59F0024 LDR R0,=A
00001004:E59F1024 LDR R1,=B
00001008:E3A02004 MOV R2,#4
0000100C:E3A05000 MOV R5,#0
00001010: loop:
00001010:E4903004 LDR R3,[R0],#4
00001014:E4914004 LDR R4,[R1],#4
00001018:E0255493 MLA R5,R3,R4,R5
0000101C:E2422001 SUB R2,R2,#1
00001020:E3520000 CMP R2,#0
00001024:1AFFFFF9 BNE loop
00001028:EF000011 SWI 0x11

.data
00001034: A:.word 10,20,30,40
00001044: B:.word 10,20,30,40
.end
```

### III. Output Table for the program

R5	$(10*10)+(20*20)+(30*30)$ $+(40*40)+(50*50)$ $=3000=00000bb8$
----	---

## Microprocessor and Computer Architecture Laboratory

UE19CS256, 4th Semester, Academic Year 2020-21

Date: 17/2/2021

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

Week# 4 Program Number: 4

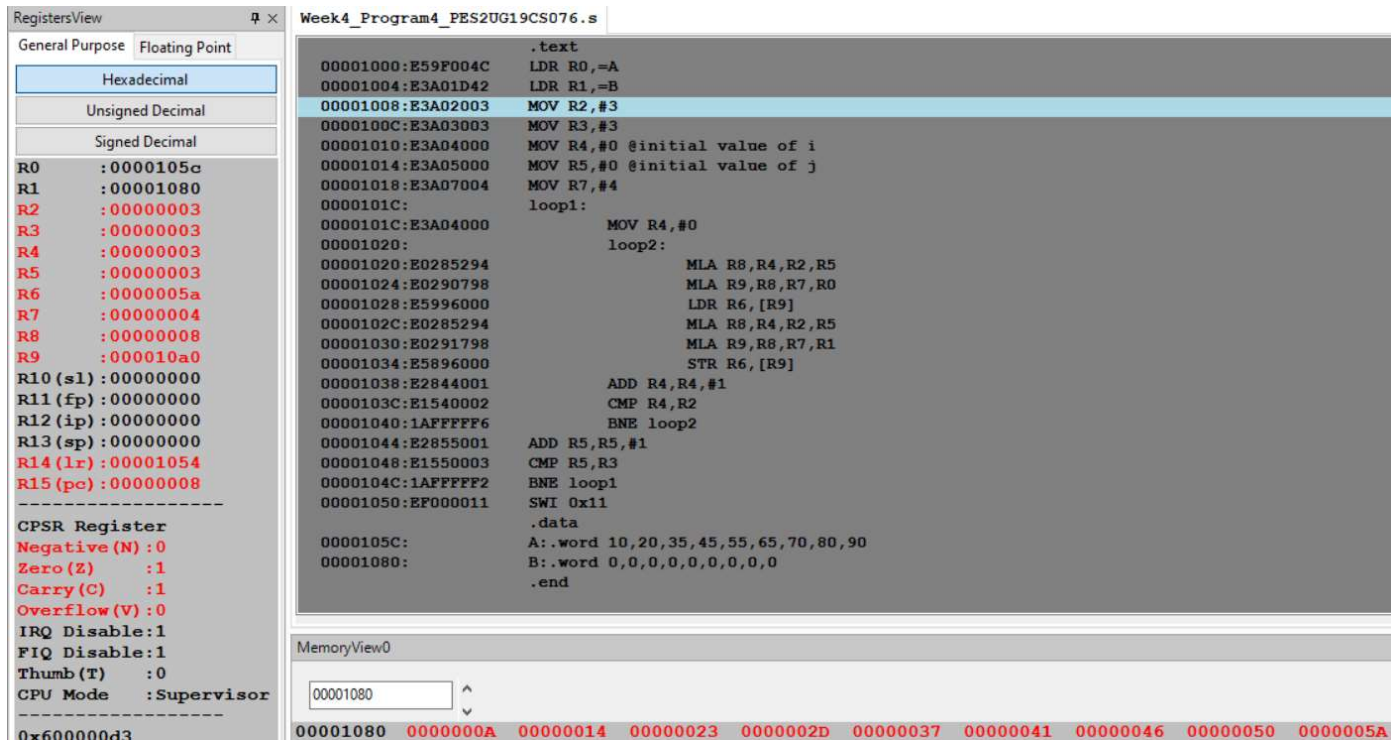
**Write ALP to read from 2D array, B=a[i] [j]**

### I. ARM Assembly Code

```
.text
LDR R0,=A
LDR R1,=B
MOV R2,#3
MOV R3,#3
MOV R4,#0 @initial value of i
MOV R5,#0 @initial value of j
MOV R7,#4
loop1:
    MOV R4,#0
    loop2:
        MLA R8,R4,R2,R5
        MLA R9,R8,R7,R0
        LDR R6,[R9]
        MLA R8,R4,R2,R5
        MLA R9,R8,R7,R1
        STR R6,[R9]
    ADD R4,R4,#1
    CMP R4,R2
    BNE loop2
ADD R5,R5,#1
CMP R5,R3
BNE loop1
SWI 0x11
.data
A:.word 10,20,35,45,55,65,70,80,90
B:.word 0,0,0,0,0,0,0,0,0
.end
```



## II. Output Screen Shot



## III. Output Table for the program

0x0A	10
0x14	20
0x23	35
0x2D	45
0x37	55
0x41	65
0x46	70
0x50	80
0x5A	90

# Microprocessor and Computer Architecture Laboratory

UE19CS256

4th Semester, Academic Year 2020-21

Date: 17/2/2021

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

Week# \_\_\_\_4\_\_\_\_

Program Number: \_\_\_\_5\_\_

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

## I. ARM Assembly Code

```
Week4_Program5_PES2UG19CS076 - Notepad
File Edit Format View Help
.text
LDR R0,=A
LDR R1,=B
LDR R2,=C
MOV R3,#3
MOV R4,#0 @initial value of i
MOV R5,#0 @initial value of j
MOV R7,#4
loop1:
    MOV R4,#0
    loop2:
        MOV R6,#0 @to store sum
        MLA R8,R4,R3,R5
        MLA R9,R8,R7,R0
        LDR R8,[R9]
        ADD R6,R6,R8
        MLA R8,R4,R3,R5
        MLA R9,R8,R7,R1
        LDR R8,[R9]
        ADD R6,R6,R8
        MLA R8,R4,R3,R5
        MLA R9,R8,R7,R2
        STR R6,[R9]
    ADD R4,R4,#1
    CMP R4,R3
    BNE loop2
ADD R5,R5,#1
CMP R5,R3
BNE loop1
SWI 0x11
.data
A:.word 10,20,35,45,55,65,70,80,90
B:.word 1,2,3,4,5,6,7,8,9
C:.word 0,0,0,0,0,0,0,0,0
.end
```

## II. Output Screen Shot

**RegistersView**

General Purpose Floating Point

Hexadecimal

Unsigned Decimal

Signed Decimal

R0 : 00001078  
R1 : 0000109c  
R2 : 000010c0  
R3 : 00000003  
R4 : 00000003  
R5 : 00000003  
R6 : 00000063  
R7 : 00000004  
R8 : 00000008  
R9 : 000010e0  
R10 (sl) : 00000000  
R11 (fp) : 00000000  
R12 (ip) : 00000000  
R13 (sp) : 00000000  
R14 (lr) : 0000106c  
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  
CPU Mode : Supervisor  
-----  
0x600000d3

**Week4\_Program5\_PES2UG19CS076.s**

```

00001014:E3A05000    MOV R5,#0 @initial value of j
00001018:E3A07004    MOV R7,#4
0000101C:              loop1:
0000101C:E3A04000    MOV R4,#0
00001020:              loop2:
00001020:E3A06000    MOV R6,#0 @to store sum
00001024:E0285394    MLA R8,R4,R3,R5
00001028:E0290798    MLA R9,R8,R7,R0
0000102C:E5998000    LDR R8,[R9]
00001030:E0866008    ADD R6,R6,R8
00001034:E0285394    MLA R8,R4,R3,R5
00001038:E0291798    MLA R9,R8,R7,R1
0000103C:E5998000    LDR R8,[R9]
00001040:E0866008    ADD R6,R6,R8
00001044:E0285394    MLA R8,R4,R3,R5
00001048:E0292798    MLA R9,R8,R7,R2
0000104C:E5896000    STR R6,[R9]
00001050:E2844001    ADD R4,R4,#1
00001054:E1540003    CMP R4,R3
00001058:1AFFFFF0    BNE loop2
0000105C:E2855001    ADD R5,R5,#1
00001060:E1550003    CMP R5,R3
00001064:1AFFFPEC    BNE loop1
00001068:EF000011    SWI 0x11
                .data
00001078:      A: .word 10,20,35,45,55,65,70,80,90
0000109C:      B: .word 1,2,3,4,5,6,7,8,9
000010C0:      C: .word 0,0,0,0,0,0,0,0
                .end

```

**MemoryView0**

000010C0

000010C0 0000000B 00000016 00000026 00000031 0000003C 00000047 0000004D 00000058 00000063

## III. Output Table for the program

A	B	C (hexadecimal)	C (in decimal)
10	1	0x0B	11
20	2	0x16	22
35	3	0x26	38
45	4	0x31	49
55	5	0x3C	60
65	6	0x47	71
70	7	0x4D	77
80	8	0x58	88
90	9	0x63	99

# Microprocessor and Computer Architecture Laboratory

UE19CS256

4th Semester, Academic Year 2020-21

Date: 17/2/2021

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

Week# 4 Program Number: 6

**Write an ALP to implement  $\text{Sum}[i] += a[i][j]$**

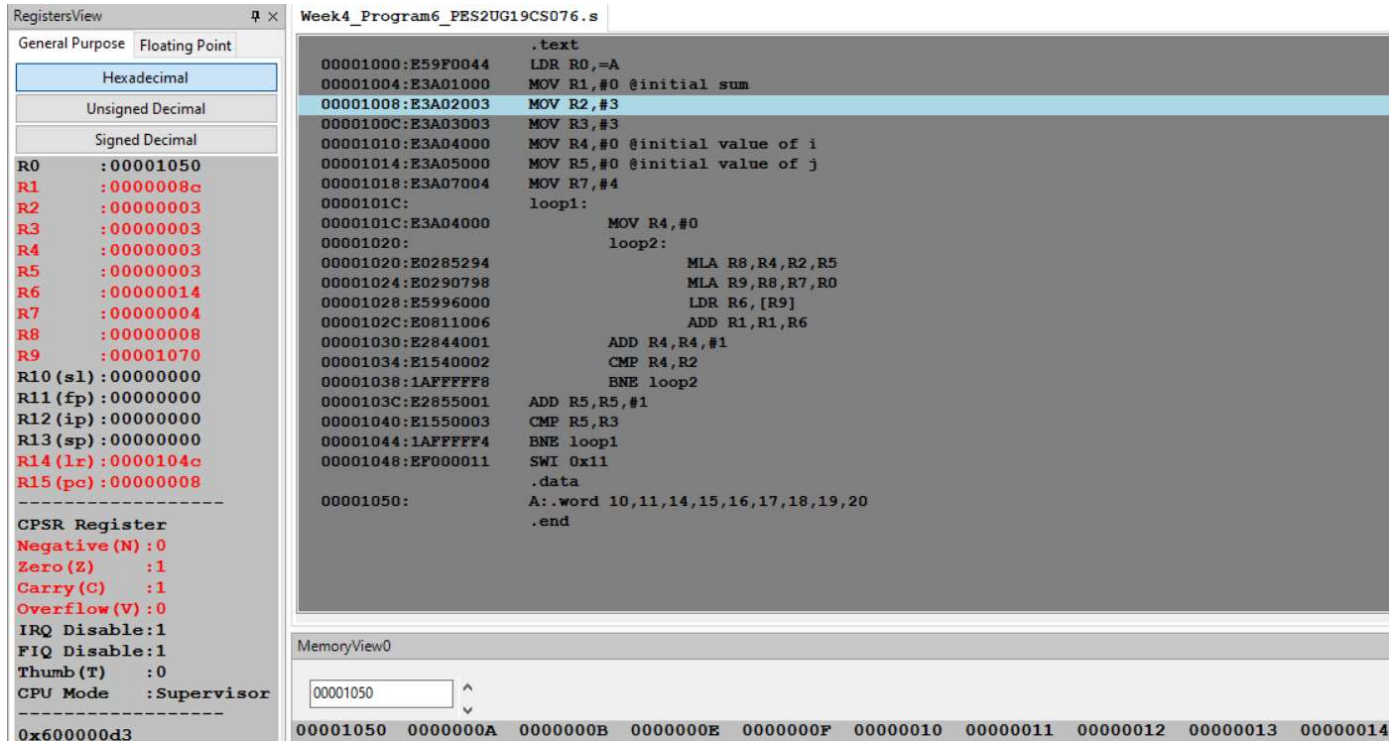
## I. ARM Assembly Code

Week4\_Program6\_PES2UG19CS076 - Notepad

File Edit Format View Help

```
.text
LDR R0,=A
MOV R1,#0 @initial sum
MOV R2,#3
MOV R3,#3
MOV R4,#0 @initial value of i
MOV R5,#0 @initial value of j
MOV R7,#4
loop1:
    MOV R4,#0
    loop2:
        MLA R8,R4,R2,R5
        MLA R9,R8,R7,R0
        LDR R6,[R9]
        ADD R1,R1,R6
        ADD R4,R4,#1
        CMP R4,R2
        BNE loop2
    ADD R5,R5,#1
    CMP R5,R3
    BNE loop1
SWI 0x11
.data
A:.word 10,11,14,15,16,17,18,19,20
.end
```

## II. Output Screen Shot



## III. Output Table for the program

<b>Addition of elements in A (stored in R1)</b>	<b>0x8C</b>	<b>140</b>
---	-------------	------------