

# MICROPROCESSORS AND COMPUTER ARCHITECTURE LABORATORY

## WEEK 5

**PES1UG20CS224**

**Kumar Abhimanyu**

### Program Title :

Write a program in ARM7TDMI-ISA to add 2 matrices of order 3. i.e.,  
Implement  $c[i][j] = a[i][j] + b[i][j]$

### Program code :

.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 R0, =A

LDR R1, =B

LDR R2, =C

MOV R3, #0

MOV R4, #0

MOV R10, #3

LOOP1:

MLA R11, R3, R10, R4

MOV R11, R11, LSL #2

LDR R5,[R0,R11]

LDR R6,[R1,R11]

ADD R7,R5,R6

STR R7,[R2,R11]

ADD R4,R4,#1

CMP R4,#3

BNE LOOP1

MOV R4,#0

ADD R3,R3,#1

CMP R3,#3

BNE LOOP1

SWI 0X011

## Output Screenshot :

The screenshot displays the ARM5Sim ARM simulator interface. The main window shows assembly code for a program that multiplies two matrices of order 3 using the MLA and MUL instructions. The code is as follows:

```
; Multiply two matrices of order 3 using MLA and MUL
; c[i][j] = c[i][j] + a[i][j]*b[i][j]
; R3 - Row number
; R4 - Column number
; R10 - Number of elements in a row - Here, 3
; R0 - First element of the array
; Logic : R3*R10 + R4

.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
00001000:E59F0048 LDR R0, =A
00001004:E3A01042 LDR R1, =B
00001008:E59F2044 LDR R2, =C
0000100C:          MOV R3, #0
00001010:E3A04000 MOV R4, #0
00001014:E3A0A003 MOV R10, #3
00001018:          LOOP1:
00001018:E02B4A33 MLA R11, R3, R10, R4
0000101C:E1A0B10B MOV R11, R11, LSL, #2
00001020:E790500B LDR R5, [R0, R11]
00001024:E791600B LDR R6, [R1, R11]
00001028:E0B57006 ADD R7, R5, R6
0000102C:E787700B STR R7, [R2, R11]
00001030:E2844001 ADD R4, R4, #1
00001034:E3540003 CMP R4, #3
00001038:1AFF7FF6 BNE LOOP1
0000103C:E3A04000 MOV R4, #0
00001040:E2833001 ADD R3, R3, #1
00001044:E3530003 CMP R3, #3
```

The left pane shows the Register View with the following values:

Register	Value
R0	4180
R1	4224
R2	4260
R3	3
R4	0
R5	9
R6	9
R7	18
R8	0
R9	0
R10	3
R11	32
R12	0
R13	21504
R14	0
R15	4172

The bottom pane shows the Output View with the following text:

```
Execution ending, Instruction Count:100 Elapsed Time:00:00:00.0259121
Instructions per second:3859
```

**Program Title :**

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

**Program code :**

.DATA

MAT: .WORD 2,4,6,8,10,12,14,16,18

SUM: .WORD

.TEXT

LDR R5,=MAT

LDR R6,=SUM

MOV R0, #3

LOOP:

    LDMIA R5!,{R2-R4}

    ADD R2,R2,R3

    ADD R2,R2,R4

    STR R2,[R6],#4

    SUB R0,R0,#1

    TEQ R0,#0

    BNE LOOP

SWI 0X11

**Output Screenshot :**

ARMSim - The ARM Simulator Dept. of Computer Science

FileViewCacheDebugWatchHelp

RegistersView

General PurposeFloating Point

Hexadecimal

Unsigned Decimal

Signed Decimal

R0:0

R1:0

R2:48

R3:16

R4:18

R5:4194

R6:4196

R7:0

R8:0

R9:0

R10(r1):0

R11(fp):0

R12(lp):0

R13(sp):21504

R14(lr):0

R15(pc):4136

CPSR Register

Negative(N):0

Zero(Z):1

Carry(C):0

Overflow(V):0

IRQ Disable:1

FIQ Disable:1

Thumb(T):0

CPU Mode: System

0x40000df

ROWSUM.s

.DATA

00001034: MAT: .WORD 2,4,6,8,10,12,14,16,18

00001058: SUM: .WORD

.TEXT

00001000:E59F5024 LDR R5,=MAT

00001004:E59F6024 LDR R6,=SUM

00001008:E3A00003 MOV R0, #3

0000100C: LOOP:

0000100C:E8B5001C LDMIA R5!,(R2-R4)

00001010:E0822003 ADD R2,R2,R3

00001014:E0822004 ADD R2,R2,R4

00001018:E4862004 STR R2,[R6],#4

0000101C:E2400001 SUB R0,R0,#1

00001020:E3300000 TEQ R0,#0

00001024:1AFF7FF8 BNE LOOP

SWI 0X11

MemoryView0

0000102C

Word Size

8Bit16Bit32Bit

0000102C 00001034 00001058 00000002 00000004 00000006 00000008 0000000A 0000000C 0000000E 00000010 00000012 0000000C 0000001E 00000030 81818181 81818181 81818181

00001070 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181

000010B4 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181

000010F8 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181

OutputView

ConsoleStdin/Stdout/Stderr

Execution ending, Instruction Count:25 Elapsed Time:00:00:00.0029902

Instructions per second:8360

OutputView

WatchView