

Department of Computer Science & Engineering

Microprocessor & Computer Architecture

MPCA-Laboratory/ Assignment/ Hands-on/ Project UE20CS252

NAME: Y Srinivas
SRN: PES1UG20CS517
WEEK_NO: 05

1. Write a program in ARM7TDMI-ISA to multiply 2 matrices of order 3.

i.e., implement $c[i][j] = c[i][j] + a[i][j] \times b[i][j]$.

a. Use MLA instruction

.TEXT

```
LDR R0,=A
LDR R1,=B
LDR R2,=C
MOV R12,#0
MOV R3,#0
MOV R4,#0
MOV R5,#0
MOV R8,#3
L1:
    MLA R6,R3,R8,R4
    MLA R7,R4,R8,R5
    LDR R10,[R0,R6,LSL #2]
    LDR R11,[R1,R7,LSL #2]
    MLA R12,R10,R11,R12
    ADD R4,R4,#1
    CMP R4,#3
    BNE L1
    MLA R9,R3,R8,R5
    STR R12,[R2,R9,LSL #2]
    MOV R4,#0
    MOV R12,#0
    ADD R5,R5,#1
    CMP R5,#3
    BNE L1
    MOV R5,#0
    ADD R3,R3,#1
    CMP R3,#3
    BNE L1
```

SWI 0X11

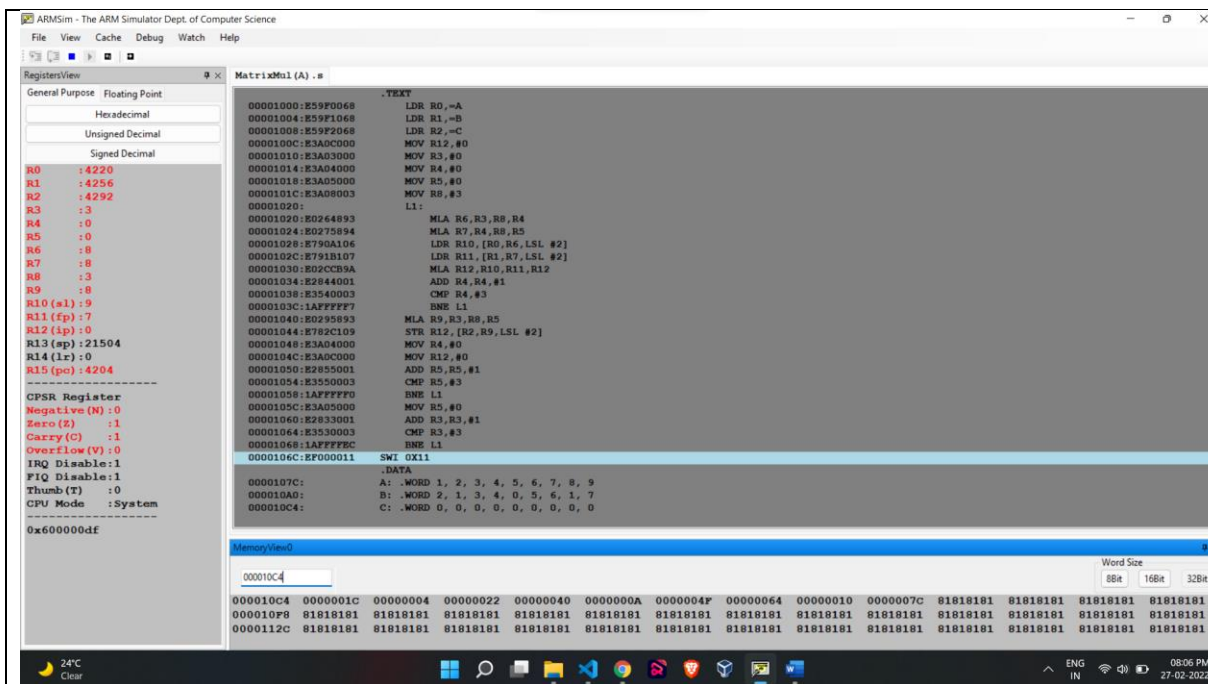
.DATA

A: .WORD 1, 2, 3, 4, 5, 6, 7, 8, 9

B: .WORD 2, 1, 3, 4, 0, 5, 6, 1, 7

C: .WORD 0, 0, 0, 0, 0, 0, 0, 0, 0

OUTPUT SCREENSHOT:



b. Use MUL instruction

.TEXT

LDR R0,=A

LDR R1,=B

LDR R2,=C

MOV R12,#0

MOV R3,#0

MOV R4,#0

MOV R5,#0

MOV R8,#3

L1: MUL R6,R3,R8

ADD R6,R6,R4

MUL R7,R4,R8

ADD R7,R7,R5

LDR R10,[R0,R6,LSL #2]

LDR R11,[R1,R7,LSL #2]

MUL R7,R10,R11

ADD R12,R12,R7

ADD R4,R4,#1

CMP R4,#3

BNE L1

MUL R9,R3,R8

ADD R9,R9,R5

STR R12,[R2,R9,LSL #2]

MOV R4,#0

MOV R12,#0

ADD R5,R5,#1

CMP R5,#3

BNE L1

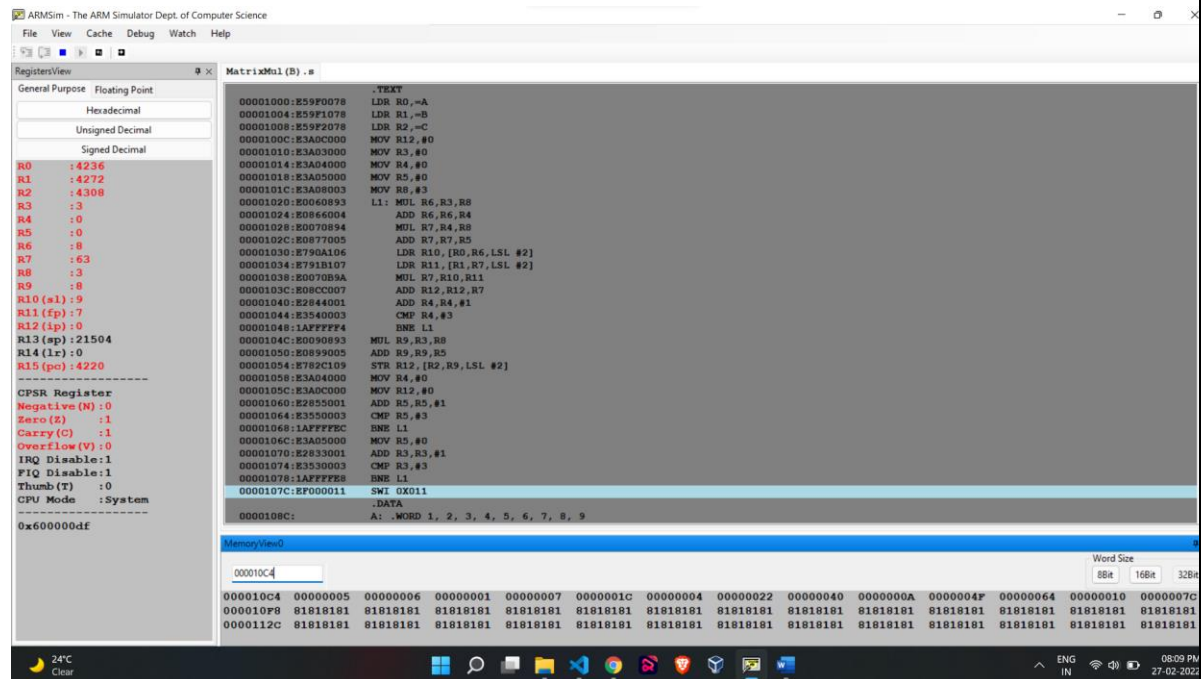
MOV R5,#0

```

ADD R3,R3,#1
CMP R3,#3
BNE L1
SWI 0X011
.DATA
A: .WORD 1, 2, 3, 4, 5, 6, 7, 8, 9
B: .WORD 2, 1, 3, 4, 0, 5, 6, 1, 7
C: .WORD 0, 0, 0, 0, 0, 0, 0, 0, 0

```

OUTPUT SCREENSHOT:



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

Code:

.TEXT

```

LDR R0,=A
LDR R9,=SUM
MOV R1,#0
MOV R2,#0
MOV R3,#3
MOV R4,#0
MOV R8,#0

```

```

LOOP:MLA R5,R3,R1,R2
      MOV R5,R5,LSL #2
      LDR R6,[R0,R5]

```

```

      ADD R4,R4,R6

```

```

ADD R2,R2,#1
CMP R2,#3
BNE LOOP
CMP R4,R8
MOVGT R8,R4
ADD R1,R1,#1
MOV R2,#0
MOV R4,#0
CMP R1,#3
BNE LOOP

```

```
STR R8,[R9]
```

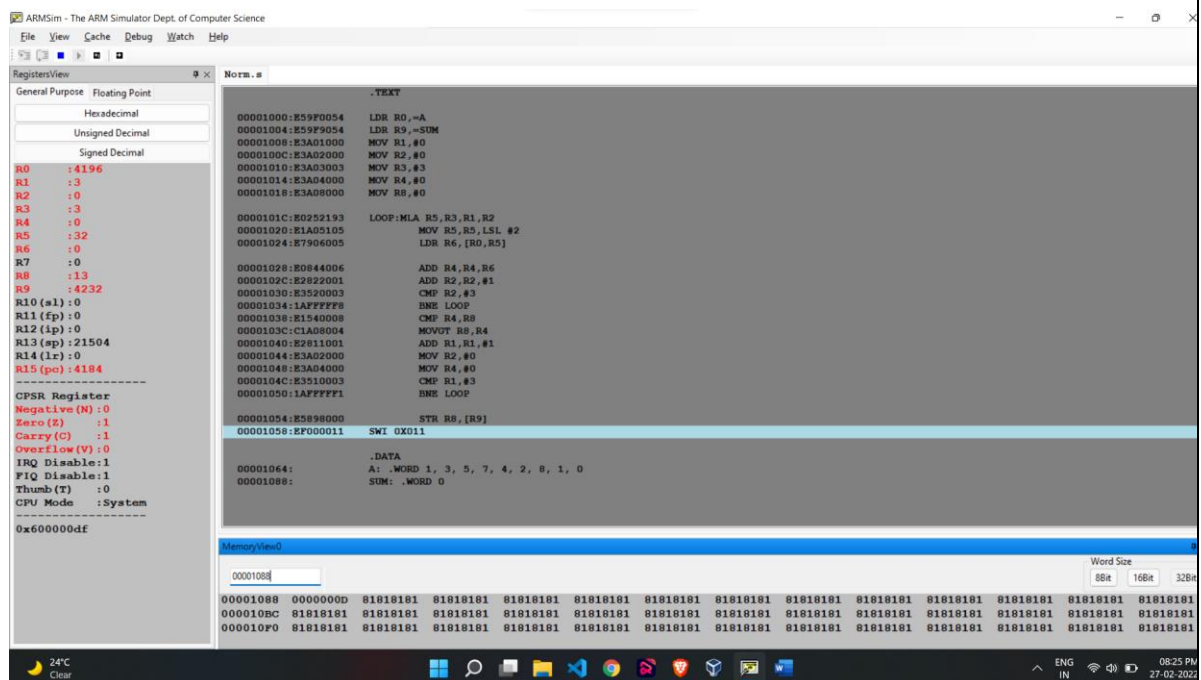
```
SWI 0X011
```

```
.DATA
```

```
A: .WORD 1, 3, 5, 7, 4, 2, 8, 1, 0
```

```
SUM: .WORD 0
```

OUTPUT SCREENSHOT



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

CODE:

```
.TEXT
```

```
LDR R0,=A
```

```
LDR R9,=SUM
```

```
MOV R1,#0
```

MOV R2,#0

MOV R3,#3

MOV R4,#0

LOOP:MLA R5,R3,R1,R2
MOV R5,R5,LSL #2
LDR R6,[R0,R5]

ADD R4,R4,R6
ADD R2,R2,#1
CMP R2,#3
BNE LOOP
STR R4,[R9],#4
ADD R1,R1,#1
MOV R2,#0
MOV R4,#0
CMP R1,#3
BNE LOOP

SWI 0X011

.DATA

A: .WORD 1, 2, 3, 4, 5, 6, 7, 8, 9

SUM: .WORD 0, 0, 0

ARMsim - The ARM Simulator Dept. of Computer Science

File View Cache Debug Watch Help

RegistersView Floating Point

Hexadecimal
Unsigned Decimal
Signed Decimal

R0 : 4184
R1 : 3
R2 : 0
R3 : 3
R4 : 0
R5 : 32
R6 : 9
R7 : 0
R8 : 0
R9 : 4232
R10 (s1) : 0
R11 (fp) : 0
R12 (ip) : 0
R13 (sp) : 21504
R14 (lr) : 0
R15 (pc) : 4172

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

Rowsum.s

.TEXT
00001000:E59F0048 LDR R0,=A
00001004:E59F9048 LDR R9,=SUM
00001008:E3A01000 MOV R1,#0
0000100C:E3A02000 MOV R2,#0
00001010:E3A03003 MOV R3,#3
00001014:E3A04000 MOV R4,#0
00001018:E0252123 LOOP:MLA R5,R3,R1,R2
0000101C:E1A05105 MOV R5,R5,LSL #2
00001020:E7906005 LDR R6,[R0,R5]
00001024:E0844006 ADD R4,R4,R6
00001028:E2822001 ADD R2,R2,#1
0000102C:E3520003 CMP R2,#3
00001030:1AFFFFF8 BNE LOOP
00001034:E4894004 STR R4,[R9],#4
00001038:E2811001 ADD R1,R1,#1
0000103C:E3A02000 MOV R2,#0
00001040:E3A04000 MOV R4,#0
00001044:E3510003 CMP R1,#3
00001048:1AFFFFF2 BNE LOOP
0000104C:EF000011 SWI 0X011
00001058: .DATA
A: .WORD 1, 2, 3, 4, 5, 6, 7, 8, 9
SUM: .WORD 0, 0, 0

MemoryView

0000107C 00000006 0000000F 00000018 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181
00001080 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181
00001084 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181

Word Size
8Bit 16Bit 32Bit

24°C Clear

ENG IN 08:28 PM 27-02-2022