



Department of Computer Science & Engineering
Microprocessor & Computer Architecture Lab

Lab 5 Problem Statements

UE22CS251B

- 1 Write an ALP to multiply 2 matrices. (3X3)

CODE

```
.DATA
MATRIXA: .WORD 1, 1, 1, 1, 1, 1, 1, 1, 1
MATRIXB: .WORD 9, 9, 9, 9, 9, 9, 9, 9, 9
MATRIXC: .WORD 0, 0, 0, 0, 0, 0, 0, 0, 0

.TEXT

MAIN:
    LDR R1, =MATRIXA
    LDR R2, =MATRIXB
    LDR R3, =MATRIXC
    MOV R4, #3

OUTER_LOOP:
    MOV R0, R2
    MOV R9, #3
INNER_LOOP:
    MOV R8, #0
    MOV R7, #3
    MOV R6, R1

INNERMOST_LOOP:
    LDR R10, [R6], #4
    LDR R11, [R0], #4
    MUL R12, R10, R11
    ADD R8, R8, R12
    SUBS R7, R7, #1

    BNE INNERMOST_LOOP
    STR R8, [R3], #4
    SUBS R9, R9, #1
    BNE INNER_LOOP
    ADD R1, R1, #12
    SUBS R4, R4, #1
    BNE OUTER_LOOP
    MOV R0, #0
    MOV R7, #1
    SWI 0x11
.END
```

OUTPUT

FileViewCacheDebugWatchHelp

RegistersView

week 5 3B.s

General PurposeFloating Point

Hexadecimal

Unsigned Decimal

Signed Decimal

R0:0
R1:4240
R2:4240
R3:4312
R4:0
R5:0
R6:4240
R7:1
R8:27
R9:0
R10 (s1):1
R11 (fp):9
R12 (ip):9
R13 (sp):21504
R14 (lr):0
R15 (pc):4188
CPSR Register
Negative (N):0
Zero (Z):1
Overflow (O):1

.DATA
0000106C: MATRIXA: .WORD 1, 1, 1, 1, 1, 1, 1, 1
00001090: MATRIXB: .WORD 9, 9, 9, 9, 9, 9, 9, 9
000010B4: MATRIXC: .WORD 0, 0, 0, 0, 0, 0, 0, 0

.TEXT

00001000: MAIN:
00001000:E59F1058 LDR R1, =MATRIXA
00001004:E59F2058 LDR R2, =MATRIXB
00001008:E59F3058 LDR R3, =MATRIXC
0000100C:E3A04003 MOV R4, #3

00001010: OUTER_LOOP:
00001010:E1A00002 MOV R0, R2
00001014:E3A09003 MOV R9, #3
00001018: INNER_LOOP:
00001018:E3A08000 MOV R8, #0

MemoryView0

0000106C

Word Size
8Bit16Bit32Bit

0000106C 00000001 00000001 00000001 00000001 00000001 00000001 00000001 00000001 00000001 00000001 00000009
00001094 00000009 00000009 00000009 00000009 00000009 00000009 00000009 00000009 00000009 0000001B 0000001B
000010BC 0000001B 0000001B 0000001B 0000001B 0000001B 0000001B 0000001B 81818181 81818181 81818181 81818181
000010E4 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181

OutputView

ConsoleStdin/Stdout/Stderr

Execution ending, Instruction Count:238 Elapsed Time:00:00:00.1405804
Instructions per second:1692

- 2 Write an ALP using conditional ARM instructions to sort an array of numbers using Bubble Sort Algorithm.

CODE

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

.text
MOV R4, #10 ; Main counter n -1
MOV R5, R4

outer_loop:
LDR R0, =a
LDR R1, =a

LDR R3, [R1,#4]! ; i+1 Element

CMP R4, #0
BEQ exit
SUB R4, R4, #1
MOV R5, R4 ; Update secondary counter to main counter()
B loop

loop:
LDR R2,[R0] ; Ith
LDR R3, [R1] ; i+1 Element

CMP R2, R3
STRGT R2,[R1]
STRGT R3,[R0]

ADD R0, R0, #4
ADD R1, R1, #4

CMP R5, #0
BEQ outer_loop
SUB R5, R5, #1

B loop

exit:
SWI 0x11
```

OUTPUT

The screenshot displays the ARM debugger interface with the following components:

- RegistersView:** Shows the state of registers R0 through R15. R0 is 4096, R1 is 4100, and R2 is 0. The CPSR register shows Negative (N) as 0, Zero (Z) as 1, and Program Flag (P) as 1.
- MemoryView:** Shows the memory at address 00001000, which contains the array 'a' with values 9, 8, 4, 3, 2, 6, 7, 1, 5, 0. The word size is set to 32bit.
- OutputView:** Shows the execution ending with an instruction count of 648 and an elapsed time of 00:00:00.0583280. The instructions per second are 11109.

3 Assignment:

i)

Write a program to swap the first and last character of a given string.

Example:

Input: 'dog'

Output: 'god'

CODE

```
.DATA
a:.ascii "Pot"

.TEXT
LDR R0,=a
MOV R5,#3

LDRB R2,[R0]
LDRB R3,[R0,#2]
STRB R3,[R0],#2
STRB R2,[R0]
LDR R0,=a
SWI 0x02
```

OUTPUT

The screenshot displays a debugger window with the following components:

- RegistersView:** Shows the state of registers R0 through R15. R0 is highlighted with a value of 00001028. R15 (pc) is highlighted with a value of 00011400.
- MemoryView:** Displays the memory contents starting from address 00001028. It shows the assembly code being executed, including the .DATA section for 'a' and the .TEXT section with instructions like LDR R0,=a, MOV R5,#3, LDRB R2,[R0], LDRB R3,[R0,#2], STRB R3,[R0],#2, STRB R2,[R0], LDR R0,=a, and SWI 0x02.
- OutputView:** Shows the output of the program, which is 'toP'.

ii)

Given a c Code convert it in its equivalent Arm Code.

a)x = (a + b) - c;

CODE

```
.DATA
A: .WORD 8
B: .WORD 4
C: .WORD 2
result: .WORD 0
```

```
.TEXT
LDR R0,=A
LDR R4,[R0]
LDR R0,=B
LDR R5,[R0]
LDR R0,=C
LDR R6,[R0]
ADD R5,R4,R5
SUB R7,R5,R6
```

OUTPUT

The screenshot displays the Keil uVision IDE interface. The main window shows the assembly code for the file 'lab 5 3(2).s'. The code is divided into two sections: .DATA and .TEXT. The .DATA section defines variables A, B, C, and result. The .TEXT section contains the assembly instructions for the program. The left pane shows the Register View, displaying the values of the registers R0 through R15. The right pane shows the Memory View, displaying the memory contents. The bottom pane shows the Output View, displaying the console output.

RegistersView: lab 5 3(2).s

General Purpose Floating Point

Hexadecimal

Unsigned Decimal

Signed Decimal

R0 : 4148

R1 : 0

R2 : 0

R3 : 0

R4 : 8

R5 : 12

R6 : 2

R7 : 10

R8 : 0

R9 : 0

R10 (sl) : 0

R11 (fp) : 0

R12 (ip) : 0

R13 (sp) : 21504

R14 (lr) : 0

R15 (pc) : 4128

CPSR Register

Negative (N) : 0

Zero (Z) : 0

Overflow (O) : 0

Memory View: 00001028

Word Size: 8Bit 16Bit 32Bit

Output View: Console Stdin/Stdout/Stderr

Loading assembly language file C:\Users\Admin\OneDrive\Desktop\college\SEM 4\MPCA\lab 5 3(2).s

b)

$z = (a \ll 2) \mid (b \& 15);$

CODE

```
.DATA
A: .WORD 10
B: .WORD 5
Z: .WORD 0

.TEXT
LDR R0, =A
LDR R1, [R0]
MOV R2, R1, LSL #2
LDR R0, =B
LDR R3, [R0]
AND R3, R3, #15
ORR R2, R2, R3
LDR R0, =Z
STR R2, [R0]
```

OUTPUT

RegistersView

General Purpose Floating Point

Hexadecimal

Unsigned Decimal

Signed Decimal

R0 : 4152

R1 : 10

R2 : 45

R3 : 5

R4 : 0

R5 : 0

R6 : 0

R7 : 0

R8 : 0

R9 : 0

R10 (s1) : 0

R11 (fp) : 0

R12 (ip) : 0

R13 (sp) : 21504

R14 (lr) : 0

R15 (pc) : 70656

CPSR Register

Negative (N) : 0

Zero (Z) : 0

Overflow (O) : 0

MemoryView0

Word Size

8Bt 16Bt 32Bt

00001000

00001000 E59F001C E5901000 E1A02101 E59F0014 E5903000 E203300F E1822003 E59F0008 E5802000 00001030

00001028 00001034 00001038 0000000A 00000005 0000000D 81818181 81818181 81818181 81818181

00001050 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181

00001078 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181

OutputView

Console Stdin/Stdout/Stderr

Execution ending, Instruction Count:16640 Elapsed Time:00:00:00.5991195

Instructions per second:27774