

Microprocessor and Computer Architecture

UE24CS251B

4th Semester, Academic Year 2025-26

Date:

Name: Harshit Chandak	SRN: PES2UG24CS185	Section C
-----------------------	-----------------------	--------------

LAB # 5 Program Number: 1

Title of the Program

Write an ARM assembly language program to compute the sum of all elements in a 3×3 matrix stored in memory.

The matrix elements are stored in row-major order as a one-dimensional array. The program should:

- Use nested loops to iterate over rows and columns.
- Compute the effective memory address of each matrix element using a separate subroutine.
- Load each matrix element from memory and accumulate the total sum in a register.
- Store or maintain the final sum in a designated register before program termination.

The program must use:

- A subroutine to calculate the address of $a[i][j]$
 - Stack operations (STMFD, LDMFD) to preserve register values
 - Multiply-Accumulate (MLA) for address computation
-
- I. Typed ARM Assembly Code
 - II. Output Screen Shot
(Screenshot including Register Window, Output Window and Code Window)

.DATA

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

SUM: .WORD 0

.TEXT

LDR R8,=A

MOV R0,#0

MOV R7,#0

MOV R3,#3

LDR R9,=SUM

ROW:

CMP R0,#3

BEQ EXIT

MOV R1,#0

COLUMN:

CMP R1,#3

BEQ NEXT_ROW

BL ADDRESS

LDR R6,[R5]

ADD R7,R7,R6

ADD R1,R1,#1

B COLUMN

NEXT_ROW:

ADD R0,R0,#1

B ROW

ADDRESS:

STMFD R13!,{R3,R4,LR}

MLA R4,R0,R3,R1

MOV R4, R4, LSL #2

ADD R5,R4,R8

LDMFD R13!,{R3,R4,LR}

MOV PC,LR

EXIT:

STR R7,[R9]

SWI 0x11

.END

The screenshot shows the ARMSim interface with the following details:

- Registers View:** Shows general purpose registers R0-R15, CPSR register, and memory pointers R10-R15.
- Memory View:** Displays memory contents at address 0x00001090, showing a repeating pattern of bytes: 00 00 00 2D, followed by 81 81 81 81.
- Output View:** Console output showing "Execution ending, Instruction Count:147 Elapsed Time:00:00:00.4143570 Instructions per second:354".

```

RegistersView
File View Cache Debug Watch Help
RegistersView 1.s StackView
General Purpose F4
    Hexadecimal
    Unsigned Decimal
    Signed Decimal
R0 : 3
R1 : 3
R2 : 0
R3 : 3
R4 : 0
R5 : 4236
R6 : 9
R7 : 45
R8 : 4204
R9 : 4240
R10(=1) : 0
R11(FP) : 0
R12(IP) : 0
R13(MP) : 21504
R14(LR) : 4340
R15(PC) : 4192
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 :Sy
0x600000df

1.s
0000106C:     .DATA
00001090:     A: .WORD 1,2,3,4,5,6,7,8,9
SUM: .WORD 0
00001090:     .TEXT
00001000:E59F805C LDR R9,=A
00001004:E3A00000 MOV R0,#0
00001008:E3A07000 MOV R7,#0
0000100C:E3A03003 MOV R3,#3
00001010:E59F9050 LDR R9,=SUM
00001014:        ROW:
00001014:E3500003 CMP R0,#3
00001018:0A00000F BEQ EXIT
0000101C:E3A01000 MOV R1,#0
00001020:        COLUMN:
00001020:E3510003 CMP R1,#3
00001024:0A000003 BEQ NEXT_ROW
00001028:0A000003 BL ADDRESS
0000102C:E5996000 LDR R6,[R5]
00001030:E0877006 ADD R7,R7,R6
00001034:E2811001 ADD R1,R1,#1
00001038:EAEFFFFE B COLUMN
0000103C:        NEXT_ROW:
0000103C:E2800001 ADD R0,R0,#1
00001040:EAEFFFF3 B ROW
00001090

MemoryView
Word Size
    8Bit 16Bit 32Bit
00001090 0000002D 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181
000010C8 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181
00001100 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181
00001138 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181

OutputView
Console Stdin/Stdout/Stderr
Execution ending, Instruction Count:147 Elapsed Time:00:00:00.4143570
Instructions per second:354

OutputView WatchView

```

Microprocessor and Computer Architecture

UE24CS251B

4th Semester, Academic Year 2025-26

Date:

Name: Harshit Chandak	SRN: PES2UG24CS185	Section C
-----------------------	-----------------------	--------------

LAB # 5

Program Number: 2

Title of the Program

Write an ARM assembly program to transfer a 3×3 matrix stored in memory location A to memory location C. Use a subroutine to compute element addresses and stack operations to save and restore registers.

The source matrix is stored as a one-dimensional array in row-major order. The program should:

Use nested loops to traverse rows and columns of the matrix.

Compute the effective address of each matrix element using a separate subroutine.

Load each element from the source matrix and store it into the destination matrix.

Use stack operations to preserve register values during subroutine calls.

Ensure that all matrix elements are copied correctly before program termination.

- I. Typed ARM Assembly Code
- II. Output Screen Shot

(One Screenshot including Register Window, Output Window and Code Window, Memory Window)

.DATA

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

B: .WORD 0,0,0,0,0,0,0,0,0

.TEXT

MOV R0,#0

MOV R2,#3

LDR R3,=A

LDR R4,=B

ROW:

CMP R0,#3

BEQ EXIT

MOV R1,#0

COLUMN:

CMP R1,#3

BEQ NEXT_ROW

BL ADDRESS

LDR R7,[R6]
STR R7,[R4]
ADD R4,R4,#4
ADD R1,R1,#1
B COLUMN

NEXT_ROW:

ADD R0,R0,#1
B ROW

ADDRESS:

STMFD R13!,{R2,R1,LR}
MLA R5,R0,R2,R1
MOV R5, R5, LSL #2
ADD R6,R3,R5
LDMFD R13!,{R2,R1,LR}
MOV PC,LR

EXIT:

SWI 0x11

.END

The figure shows the ARMSim interface with three main panes: RegistersView, StackView, and MemoryView.

- RegistersView:** Shows the ARM register state. The R0-R7 registers contain values 0x3, 0x3, 0x3, 0x4200, 0x4272, 0x32, 0x4232, and 0x9 respectively. The CPSR register shows Negative(0), Zero(Z) set, Carry(C) set, and Overflow(V) set. The IRQ, FIQ, and Thumb bits are disabled, while CPU mode is set to System.
- StackView:** Displays the stack contents starting at address 0x0000108C, which are all zeros.
- MemoryView:** Shows memory starting at address 0x00001000. It displays assembly code for a program that initializes R0-R7, sets CPSR flags, and performs various arithmetic operations like ADD, SUB, CMP, BEQ, SBC, and ADD. The code includes labels .DATA, .TEXT, and .CODE, and uses immediate values like #3, #4, #0, and #1. The memory dump also shows the CPSR register value at address 0x0000103C.

Microprocessor and Computer Architecture

UE24CS251B

4th Semester, Academic Year 2025-26

Date:

Name: Harshit Chandak	SRN: PES2UG24CS185	Section C
-----------------------	-----------------------	--------------

LAB # 5

Program Number: 3

Title of the Program

Generate the first N Fibonacci numbers and store them in memory while using stack operations for register preservation.

- I. ARM Assembly Code
- II. Output Screen Shot
(One Screenshot including Register Window, Memory Window and Code Window)

.DATA

FIB: .WORD 0,0,0,0,0

.TEXT

LDR R4, =FIB

MOV R2, #10

MOV R0, #0

MOV R1, #1

STMFD SP!, {R0,R1}

STR R0, [R4], #4

STR R1, [R4], #4

SUB R2, R2, #2

LOOP:

ADD R3, R0, R1

MOV R0, R1

MOV R1, R3

STMFD SP!, {R3}

STR R3, [R4], #4

SUBS R2, R2, #1

BNE LOOP

SWI 0x11

.END

```

RegistersView 3A.s StackView
File View Cache Debug Watch Help
RegistersView 3A.s StackView
General Purpose F4
    Hexadecimal
    Unsigned Decimal
    Signed Decimal
R0 : 21
R1 : 34
R2 : 0
R3 : 34
R4 : 4204
R5 : 0
R6 : 0
R7 : 0
R8 : 0
R9 : 0
R10 (s1) : 0
R11 (fp) : 0
R12 (ip) : 0
R13 (sp) : 21464
R14 (lr) : 0
R15 (pc) : 4156
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 : Sys
0x6000000df

MemoryView0
00001044
00000000 00000001 00000000 00000002 00000003 00000005 00000008 0000000D 00000015 00000022 81818181 81818181 81818181
0000107C 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
000010EC 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181

OutputView
Console Stdin/Stdout/Stder
Execution starting ...
Execution ending, Instruction Count:65 Elapsed Time:00:00:00
OutputView WatchView

```

Microprocessor and Computer Architecture

UE22CS251B

4th Semester, Academic Year 2023-24

Date:

Name:Harshit Chandak	SRN: PES2UG24CS185	Section C
----------------------	-----------------------	--------------

LAB # 5

Program Number: 4

Title of the Program

Write an ARM ALP to count the number of words in a string.

- I. ARM Assembly Code
- II. Output Screen Shot
(One Screenshot including Register Window, Memory Window ,Output Window, Code Window)

.DATA

MSG: .ASCIZ "ARM MPCA PES"

COUNT: .WORD 0

.TEXT

LDR R0,=MSG

MOV R1,#0

MOV R2,#1

LOOP:

LDRB R3,[R0],#1

CMP R3,#0

BEQ EXIT

CMP R3,#' '

BEQ SPACE

CMP R2,#1

ADDEQ R1,R1,#1

MOVEQ R2,#0

B LOOP

SPACE:

MOV R2,#1

B LOOP

EXIT:

LDR R3,=COUNT

STR R1,[R3]

SWI 0x11

.END

```

RegistersView
File View Cache Debug Watch Help
RegistersView
General Purpose F4 x
Hexadecimal
Unsigned Decimal
Signed Decimal
R0 : 4185
R1 : 3
R2 : 0
R3 : 4188
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) : 4160

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 :Sy
0x600000df

4.s
0000104C: .DATA
0000104C: MSG: .ASCIZ "ARM MPCA PES"
0000105C: COUNT: .WORD 0
00001000:259F003C LDR R0,=MSG
00001004:E3A01000 MOV R1,#0
00001008:E3A02001 MOV R2,#1
0000100C: LOOP:
0000100C:E4D03001 LDRB R3,[R0],#1
00001010:E3530000 CMP R3,#0
00001014:0A000007 BEQ EXIT
00001018:E3530020 CMP R3,#1
0000101C:0A000003 BEQ SPACE
00001020:E3520001 CMP R2,#1
00001024:02811001 ADDEQ R1,R1,#1
00001028:03A02000 MOVEQ R2,#0
0000102C:EAFFFFF6 B LOOP
00001030: SPACE:
00001030:E3A02001 MOV R2,#1
00001034:EAFFFFF4 B LOOP
00001038: EXIT:
00001038:E59F3008 LDR R3,=COUNT
0000103C:E5831000 STR R1,[R3]
00001040:EF000011 SWI 0x11

MemoryView
Word Size
8Bit 16Bit 32Bit
0000105c
0000105C 00000003 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 00001094 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 000010CC 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 00001104 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181
OutputView
Console Stdin/Stdout/Stderr
Execution ending, Instruction Count:113 Elapsed Time:00:00:00.0160183
Instructions per second:7054

OutputView WatchView

```

Microprocessor and Computer Architecture

UE22CS251B

4th Semester, Academic Year 2023-24

Date:

Name: Harshit Chandak	SRN: PES2UG24CS185	Section C
-----------------------	-----------------------	--------------

LAB # 5

Assignment Question 1

Title of the Program

Write an ALP using conditional ARM instructions to sort an array of numbers using Bubble Sort Algorithm. Use STMFD to save registers before swap and LDMFD to restore the registers

- I. ARM Assembly Code
- II. Output Screen Shot
(Screenshot including Register Window, Memory Window and Code Window)

.DATA

A: .WORD 3,4,1,6,5

.TEXT

LDR R1,=A

MOV R5,#5

MOV R0,#0

LOOP:

CMP R0,#4

BEQ EXIT

MOV R2,#0

INNERLOOP:

MOV R4, R5

SUB R4, R4, R0

SUB R4, R4, #1

CMP R2, R4

BEQ NEXT_OUTER

BL ADDRESS

LDR R6,[R3]

LDR R7,[R3,#4]

CMP R6,R7

STRGT R7,[R3]

STRGT R6,[R3,#4]

ADD R2,R2,#1

B INNERLOOP

NEXT_OUTER:

ADD R0,R0,#1

B LOOP

ADDRESS:

STMFD R13!,{R0,R2,LR}

ADD R3,R1,R2,LSL #2

LDMFD R13!,{R0,R2,LR}

MOV PC,LR

EXIT:

SWI 0x11

.END

The screenshot shows the ARMSim interface with the following details:

- Registers View:** Shows general purpose registers R0-R15 and CPSR register. Values for R0-R15 are mostly zero, except R1 (lr) which is 4204, R2 (r1) which is 1, R3 (r2) which is 4204, R4 (r3) which is 1, R5 (r4) which is 5, R6 (r5) which is 1, R7 (r6) which is 3, R8 (r7) which is 0, R9 (r8) which is 0, R10 (r9) which is 0, R11 (fp) which is 0, R12 (ip) which is 0, R13 (sp) which is 21504, R14 (lr) which is 4144, and R15 (pc) which is 4196.
- Memory View:** Shows memory starting at address 0x0000106C. The first few bytes are 0000106C 00000001 00000003 00000004 00000005 00000006. The instruction at 0000106C is SWI 0x11.
- Stack View:** Shows the stack starting at address 0000539C, containing a series of 16-bit values.
- Output View:** Shows the console output: "Execution ending. Instruction Count:216 Elapsed Time:00:00:00.3993240 Instructions per second:540".

Microprocessor and Computer Architecture

UE22CS251B

4th Semester, Academic Year 2023-24

Date:

Name: Harshit Chandak	SRN: PES2UG24CS185	Section 2
-----------------------	-----------------------	--------------

LAB # 3

Assignment Question 2

Title of the Program

Write an ALP using ARM7TDMI to find the length of string

- I. ARM Assembly Code
- II. Output Screen Shots

(One Screenshot including Register Window, Memory Window and Code Window)

.DATA

A: .ASCIZ "HARSHITCHANDAK"

COUNT: .WORD 0

.TEXT

LDR R0,A

MOV R3,#0

LOOP:

LDRB R1,[R0],#1

CMP R1,#0

BEQ EXIT

ADD R3,R3,#1

BNE LOOP

EXIT:

LDR R2,=COUNT

STR R3,[R2]

SWI 0x11

.END

The screenshot shows the ARMSSim interface with the following windows and content:

- RegistersView**: Shows general purpose registers (R0-R15) in Unsigned Decimal format. Values: R0: 4159, R1: 0, R2: 4160, R3: 14, 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): 4132.
- MemoryView**: Shows memory starting at address 00001040. The first few bytes are 00001040: 0000000E, 00001041: 81818181, 00001042: 81818181, 00001043: 81818181, 00001044: 81818181, 00001045: 81818181, 00001046: 81818181, 00001047: 81818181, 00001048: 81818181, 00001049: 81818181, 0000104A: 81818181, 0000104B: 81818181, 0000104C: 81818181, 0000104D: 81818181, 0000104E: 81818181, 0000104F: 81818181.
- OutputView**: Displays the final output message: "Execution ending. Instruction Count:78 Elapsed Time:00:00:00.0172877 Instructions per second:4511".

The assembly code window contains the following code:

```
.DATA
A: .ASCIZ "HARSHITCHANDAK"
COUNT: .WORD 0

.TEXT
00001000:E59F0020 LDR R0,=A
00001004:E3A03000 MOV R3,#0

00001008: LOOP:
00001008:E4D01001 LDRB R1,[R0],#1
0000100C:E3510000 CMP R1,#0
00001010:0A000001 BEQ EXIT
00001014:E2833001 ADD R3,R3,#1
00001018:1AFFFFFFA BNE LOOP

0000101C: EXIT:
0000101C:E3A02D41 LDR R2,=COUNT
00001020:E5823000 STR R3,[R2]
00001024:EP000011 SWI 0x11
00001028:00001030 .END
0000102C:00000000
```

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:

SRN:

Section:

Date: