

Microprocessor and Computer Architecture Laboratory

UE19CS256

4th Semester, Academic Year 2020-21

Date:24-01-2021

Name: A.Narendiran	SRN:PES1UG19CS001	Section
		A

Week# 1 Program Number: 1

Title of the Program

Write an ALP using ARM instruction set to add and subtract two 32 bit numbers .Both numbers are in registers.

- I. ARM Assembly Code for each program
- II. Final Output Screen Shot (Register Window, Output window)
The output should be verified with 2 test cases (one example shown in class, one example of own choice)

```

.text

; addition
MOV r0, #63
MOV r1, #36
ADD r2, r0, r1

; subtraction
MOV r0, #96
MOV r1, #69
SUB r3, r0, r1
SUB r4, r1, r0

.end

```

RegistersView	RegistersView
General Purpose Floating Point	General Purpose Floating Point
Hexadecimal	Hexadecimal
Unsigned Decimal	Unsigned Decimal
Signed Decimal	Signed Decimal
R0 : 63 R1 : 36 R2 : 99 R3 : 27 R4 : -27 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 Carry (C) : 0 Overflow (V) : 0 IRQ Disable: 1 FIQ Disable: 1 Thumb (T) : 0 CPU Mode : System ----- 0x000000df	R0 : 10 R1 : 20 R2 : 30 R3 : -10 R4 : 10 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 Carry (C) : 0 Overflow (V) : 0 IRQ Disable: 1 FIQ Disable: 1 Thumb (T) : 0 CPU Mode : System ----- 0x000000df

Microprocessor and Computer Architecture Laboratory

UE19CS256

4th Semester, Academic Year 2020-21

Date:

Name: A.Narendiran	SRN: PES1UG19CS001	Section A
--------------------	-----------------------	------------------

Week# ____1____ Program Number: ____2____

Title of the Program

Write an ALP to demonstrate logical operations. All operands are in registers.

- I. ARM Assembly Code for each program
- II. Final Output Screen Shot (Register Window, Output window)
The output should be verified with 2 test cases
(one example shown in class, one example of own choice)

```

.text
MOV r0, #5
MOV r1, #6
AND r2, r1, r2
ORR r3, r1, r2
EOR r4, r1, r2
MVN r5, r0
.end

```

RegistersView	
General Purpose Floating Point	
Hexadecimal	
Unsigned Decimal	
Signed Decimal	
R0	:00000005
R1	:00000006
R2	:00000000
R3	:00000006
R4	:00000006
R5	:fffffffa
R6	:00000000
R7	:00000000
R8	:00000000
R9	:00000000
R10 (s1)	:00000000
R11 (fp)	:00000000
R12 (ip)	:00000000
R13 (sp)	:00005400
R14 (lr)	:00000000
R15 (pc)	:00011400

CPSR Register	
Negative (N)	:0
Zero (Z)	:0
Carry (C)	:0
Overflow (V)	:0
IRQ Disable	:1
FIQ Disable	:1
Thumb (T)	:0
CPU Mode	:System

0x000000df	

RegistersView	
General Purpose Floating Point	
Hexadecimal	
Unsigned Decimal	
Signed Decimal	
R0	:0000000a
R1	:0000000f
R2	:00000000
R3	:0000000f
R4	:0000000f
R5	:ffffff5
R6	:00000000
R7	:00000000
R8	:00000000
R9	:00000000
R10 (s1)	:00000000
R11 (fp)	:00000000
R12 (ip)	:00000000
R13 (sp)	:00005400
R14 (lr)	:00000000
R15 (pc)	:00011400

CPSR Register	
Negative (N)	:0
Zero (Z)	:0
Carry (C)	:0
Overflow (V)	:0
IRQ Disable	:1
FIQ Disable	:1
Thumb (T)	:0
CPU Mode	:System

0x000000df	

Microprocessor and Computer Architecture Laboratory

UE19CS256

4th Semester, Academic Year 2020-21

Date:

Name: A.Narendiran	SRN: PES1UG19CS001	Section A
--------------------	-----------------------	--------------

Week# 1 Program Number: 3

Title of the Program

Write an ALP to add 5 numbers where values are present in registers.

- I. ARM Assembly Code for each program
- II. Final Output Screen Shot (Register Window, Output window)

The output should be verified with 2 test cases
(one example shown in class, one example of own choice)

```

.text
MOV r0, #0x05
MOV r1, #0x06
MOV r2, #0x07
MOV r3, #0x06
MOV r4, #0x15
ADD r5, r0, r1
ADD r6, r5, r2
ADD r7, r6, r3
ADD r8, r7, r4
.end

```

RegistersView	
General Purpose	Floating Point
Hexadecimal	
Unsigned Decimal	
Signed Decimal	
R0	: 00000005
R1	: 00000006
R2	: 00000007
R3	: 00000006
R4	: 00000015
R5	: 0000000b
R6	: 00000012
R7	: 00000018
R8	: 0000002d
R9	: 00000000
R10 (sl)	: 00000000
R11 (fp)	: 00000000
R12 (ip)	: 00000000
R13 (sp)	: 00005400
R14 (lr)	: 00000000
R15 (pc)	: 00011400

CPSR Register	
Negative (N)	: 0
Zero (Z)	: 0
Carry (C)	: 0
Overflow (V)	: 0
IRQ Disable	: 1
FIQ Disable	: 1
Thumb (T)	: 0
CPU Mode	: System

0x000000df	

RegistersView	
General Purpose	Floating Point
Hexadecimal	
Unsigned Decimal	
Signed Decimal	
R0	: 10
R1	: 20
R2	: 30
R3	: 40
R4	: 50
R5	: 30
R6	: 60
R7	: 100
R8	: 150
R9	: 0
R10 (sl)	: 0
R11 (fp)	: 0
R12 (ip)	: 0
R13 (sp)	: 21504
R14 (lr)	: 0
R15 (pc)	: 70656

CPSR Register	
Negative (N)	: 0
Zero (Z)	: 0
Carry (C)	: 0
Overflow (V)	: 0
IRQ Disable	: 1
FIQ Disable	: 1
Thumb (T)	: 0
CPU Mode	: System

0x000000df	

Microprocessor and Computer Architecture Laboratory

UE19CS256

4th Semester, Academic Year 2020-21

Date:

Name: A.Narendiran	SRN: PES1UG19CS001	Section A
--------------------	-----------------------	--------------

Week# 1 Program Number: 4

Title of the Program

Write an ALP using ARM instruction set to check if a number stored in a register is even or odd. If even, store 00 in R0, else store FF in R0

- I. ARM Assembly Code for each program
- II. Final Output Screen Shot (Register Window, Output window)

The output should be verified with 2 test cases
(one example shown in class, one example of own choice)

```

.text
MOV r0, #0
MOV r1, #23
ANDs r1, r1, #1

BEQ cond
    MOV r0, #0xFF
    B exit
cond:
    MOV r0, #0
exit:
    SWI 0x011
.end

```

ARMSim - The ARM Simulator Dept. of Computer Science

File View Cache Debug Watch Help

RegistersView

General Purpose Floating Point

Hexadecimal

Unsigned Decimal

Signed Decimal

R0 : 255

R1 : 1

R2 : 0

R3 : 0

R4 : 0

R5 : 0

R6 : 0

R7 : 0

R8 : 0

R9 : 0

R10 (s1) : 0

R11 (fp) : 0

R12 (ip) : 0

R13 (sp) : 0

R14 (lr) : 4128

R15 (pc) : 8

CPSR Register

Negative (N) : 0

Zero (Z) : 0

Carry (C) : 0

Overflow (V) : 0

IRQ Disable : 1

FIQ Disable : 1

Thumb (T) : 0

CPU Mode : Supervisor

0x000000d3

p4.s

.text

```

00001000:E3A00000    MOV r0, #0
00001004:E3A01017    MOV r1, #23
00001008:E2111001    ANDs r1, r1, #1

0000100C:0A000001    BEQ cond
00001010:E3A000FF    MOV r0, #0xFF
00001014:EA000000    B exit
00001018:                cond:
00001018:E3A00000    MOV r0, #0
0000101C:                exit:
0000101C:EF000011    SWI 0x011

.end

```

OutputView

Console Stdin/Stdout/Stderr

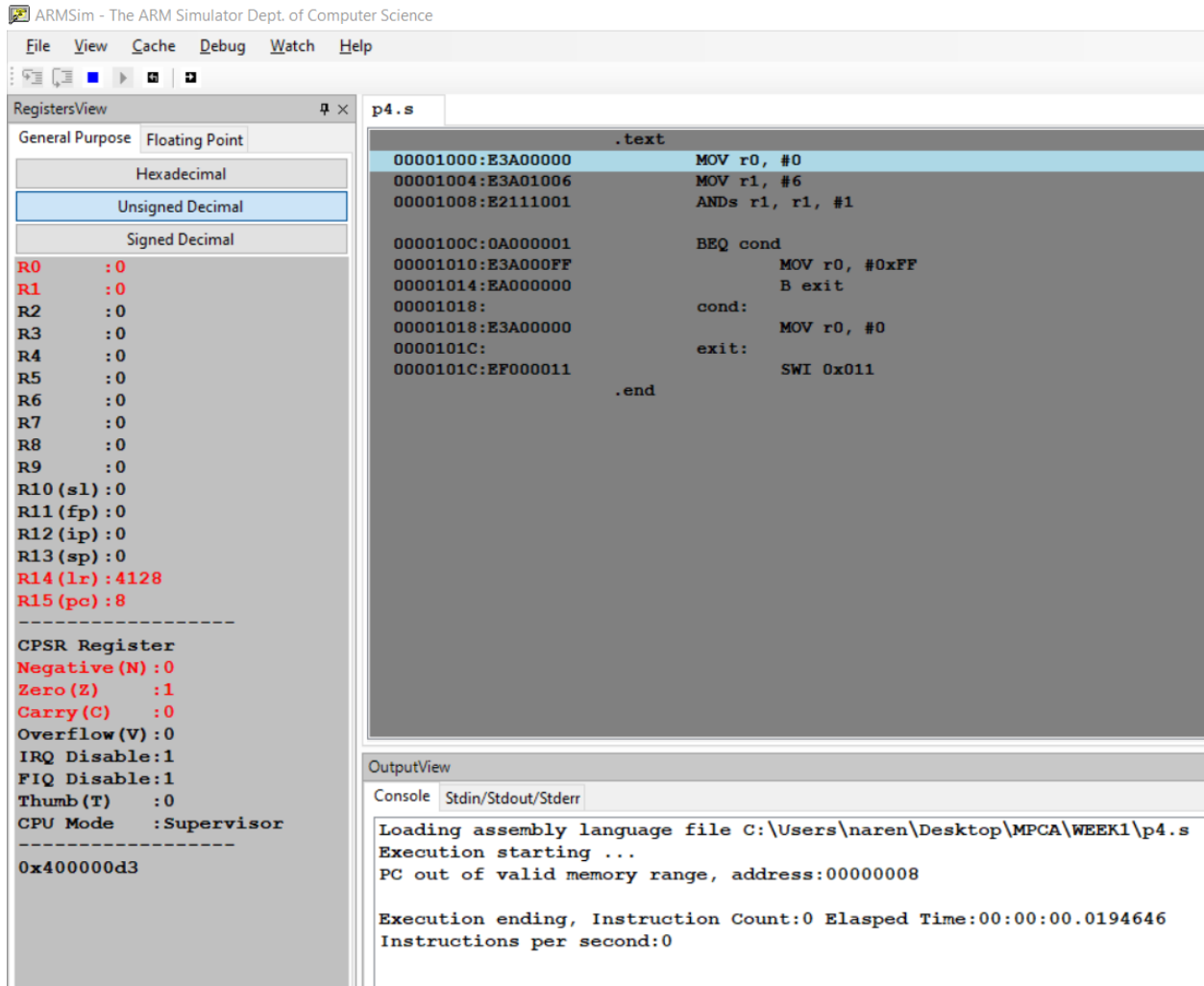
Loading assembly language file C:\Users\naren\Desktop\MPCA\WEEK1\p4.s

Execution starting ...

PC out of valid memory range, address:00000008

Execution ending, Instruction Count:0 Elapsed Time:00:00:00.0239998

Instructions per second:0



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: A.Narendiran

SRN: PES1UG19CS001

Section: A

Date:24-01-2021