

Microprocessor and Computer Architecture Laboratory

UE19CS256

4th Semester, Academic Year 2020-21

Date: 27/01/2021

Name : Abhishek Aditya BS	SRN: PES1UG19CS019	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

a) For Addition :

Test case 1 : Addition of 7 and -5

`;Program using ARM instruction set to add two
32 bit numbers stored in registers`

`mov r0,#7 ;Storing 7 in the register r0`

`mov r1,#-5 ;Storing -5 in the register r1`

`;the result of addition of values(operands)
stored in r0 and r1 is stored in r2`

`add r2,r0,r1`

`swi 0x11`

Test case 2 : Addition of 9 and 8

`;Program using ARM instruction set to add two
32 bit numbers stored in registers`

`mov r0,#9 ;Storing 9 in the register r0`

`mov r1,#8 ;Storing 8 in the register r1`

`;the result of addition of values(operands)
stored in r0 and r1 is stored in r2`

`add r2,r0,r1`

`swi 0x11`

b) For Subtraction :

Test case 1 : Subtraction of 18 and 11

`;Program using ARM instruction set to subtract
two 32 bit numbers stored in registers`

`mov r0,#18 ;Storing value 18 in the register r0`

`mov r1,#11 ;Storing value 11 in the register r1`

`;the result of subtraction of values(operands)
stored in r0 and r1 is stored in r2`

```
sub r2,r0,r1
```

```
swi 0x11
```

Test case 2 : Subtraction of -22 and 8

;Program using ARM instruction set to subtract two 32 bit numbers stored in registers

```
mov r0,#-22;Storing value -22 in the register r0
```

```
mov r1,#8 ;Storing value 8 in the register r1
```

;the result of subtraction of values(operands) stored in r0 and r1 is stored in r2

```
sub r2,r0,r1
```

```
swi 0x11
```

11. 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)

For Addition : Test case 1 : Addition of 7 and -5

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 : 7
R1 : -5
R2 : 2
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) : 21504
R14 (lr) : 0
R15 (pc) : 4108

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

1_1a_FES1UG19CS019.s

;Program using ARM instruction set to add two 32 bit numbers stored in registers

```
00001000:E3A00007    mov r0,#7 ;Storing 7 in the register r0
00001004:E3E01004    mov r1,#-5 ;Storing -5 in the register r1

;the result of addition of values(operands) stored in r0 and r1 is stored in r2
00001008:E0802001    add r2,r0,r1

0000100C:EF000011    swi 0x11
```

OutputView

Console Stdin/Stdout/Stderr

Execution starting ...

Execution ending, Instruction Count:0 Elapsed Time:00:00:00.0049570
Instructions per second:0

For Addition : Test case 2 : Addition of 9 and 8

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 : 9
R1 : 8
R2 : 17
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) : 21504
R14 (lr) : 0
R15 (pc) : 4108

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

1_1b_FES1UG19CS019.s

;Program using ARM instruction set to add two 32 bit numbers stored in registers

```
00001000:E3A00009    mov r0,#9 ;Storing 9 in the register r0
00001004:E3A01008    mov r1,#8 ;Storing 8 in the register r1

;the result of addition of values(operands) stored in r0 and r1 is stored in r2
00001008:E0802001    add r2,r0,r1

0000100C:EF000011    swi 0x11
```

OutputView

Console Stdin/Stdout/Stderr

Execution starting ...

Execution ending, Instruction Count:0 Elapsed Time:00:00:00.0049561
Instructions per second:0

For Subtraction : Test case 1 :Subtraction of 18 and 11

The screenshot shows the ARMSim interface for Test case 1. The RegistersView on the left displays the state of the ARM registers. The main window shows the assembly code for the program. The OutputView at the bottom shows the execution log.

RegistersView:

Register	Value
R0	18
R1	11
R2	7
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)	21504
R14 (lr)	0
R15 (pc)	4108

CPSR Register:

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

Assembly Code:

```
;Program using ARM instruction set to subtract two 32 bit numbers stored in registers
00001000:E3A00012  mov r0,#18 ;Storing value 18 in the register r0
00001004:E3A0100B  mov r1,#11 ;Storing value 11 in the register r1

;the result of subtraction of values(operands) stored in r0 and r1 is stored in r2
00001008:E0402001  sub r2,r0,r1

0000100C:EF000011  swi 0x11
```

OutputView:

```
Execution starting ...
Execution ending, Instruction Count:0 Elapsed Time:00:00:00.0049936
Instructions per second:0
```

For Subtraction : Test case 2 :Subtraction of -22 and 8

The screenshot shows the ARMSim interface for Test case 2. The RegistersView on the left displays the state of the ARM registers. The main window shows the assembly code for the program. The OutputView at the bottom shows the execution log.

RegistersView:

Register	Value
R0	-22
R1	8
R2	-30
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)	21504
R14 (lr)	0
R15 (pc)	4108

CPSR Register:

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

Assembly Code:

```
;Program using ARM instruction set to subtract two 32 bit numbers stored in registers
00001000:E3E00015  mov r0,#-22 ;Storing value -22 in the register r0
00001004:E3A01008  mov r1,#8 ;Storing value 8 in the register r1

;the result of subtraction of values(operands) stored in r0 and r1 is stored in r2
00001008:E0402001  sub r2,r0,r1

0000100C:EF000011  swi 0x11
```

OutputView:

```
Execution starting ...
Execution ending, Instruction Count:0 Elapsed Time:00:00:00.0049561
Instructions per second:0
```

Microprocessor and Computer Architecture Laboratory

UE19CS256

4th Semester, Academic Year 2020-21

Date: 27/01/2021

Name : Abhishek Aditya BS	SRN: PES1UG19CS019	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

Test case 1 : 12 in r0 register and 4 in r1 register

`;Demonstration of Logical AND,OR,OR,XOR,NOT
operations`

`mov r0,#12 ;Storing value 12 in r0`

`mov r1,#4 ;Storing value 4 in r1`

`;Logical AND operation`

`and r2,r0,r1`

```
;Logical OR operation
orr r3,r0,r1

;Logical XOR operation
eor r4,r0,r1

;Logical NOT operation
mvn r5,r0

swi 0x11
```

Test case 2 : 5 in r0 register and 6 in r1 register

```
;Demonstration of Logical AND,OR,OR,XOR,NOT
operations

mov r0,#5 ;Storing value 5 in r0
mov r1,#6 ;Storing value 6 in r1

;Logical AND operation
and r2,r0,r1

;Logical OR operation
orr r3,r0,r1

;Logical XOR operation
eor r4,r0,r1

;Logical NOT operation
mvn r5,r0

swi 0x11
```

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)

(1) Test case 1 : 12 in r0 register , 4 in r1 register

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 :12
R1 :4
R2 :4
R3 :12
R4 :8
R5 :-13
R6 :0
R7 :0
R8 :0
R9 :0
R10 (sl):0
R11 (fp):0
R12 (ip):0
R13 (sp):21504
R14 (lr):0
R15 (pc):4120

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

2a_PES1UG19CS019.s

;Demonstration of Logical AND,OR,OR,XOR,NOT operations

```
00001000:E3A0000C  mov r0,#12 ;Storing value 12 in r0
00001004:E3A01004  mov r1,#4 ;Storing value 4 in r1

00001008:E0002001  ;Logical AND operation
                    and r2,r0,r1

0000100C:E1803001  ;Logical OR operation
                    orr r3,r0,r1

00001010:E0204001  ;Logical XOR operation
                    eor r4,r0,r1

00001014:E1E05000  ;Logical NOT operation
                    mvn r5,r0

00001018:EF000011  swi 0x11
```

OutputView

Console Stdin/Stdout/Stderr

Execution starting ...

Execution ending, Instruction Count:0 Elapsed Time:00:00:00.0049594
Instructions per second:0

(2) Test case 2 : 5 in r0 register , 6 in r1 register

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 :5
R1 :6
R2 :4
R3 :7
R4 :3
R5 :-6
R6 :0
R7 :0
R8 :0
R9 :0
R10 (sl):0
R11 (fp):0
R12 (ip):0
R13 (sp):21504
R14 (lr):0
R15 (pc):4120

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

2b_PES1UG19CS019.s

;Demonstration of Logical AND,OR,OR,XOR,NOT operations

```
00001000:E3A00005  mov r0,#5 ;Storing value 5 in r0
00001004:E3A01006  mov r1,#6 ;Storing value 6 in r1

00001008:E0002001  ;Logical AND operation
                    and r2,r0,r1

0000100C:E1803001  ;Logical OR operation
                    orr r3,r0,r1

00001010:E0204001  ;Logical XOR operation
                    eor r4,r0,r1

00001014:E1E05000  ;Logical NOT operation
                    mvn r5,r0

00001018:EF000011  swi 0x11
```

OutputView

Console Stdin/Stdout/Stderr

Execution starting ...

Execution ending, Instruction Count:0 Elapsed Time:00:00:00.0049543
Instructions per second:0

Microprocessor and Computer Architecture Laboratory

UE19CS256

4th Semester, Academic Year 2020-21

Date: 27/01/2021

Name : Abhishek Aditya BS	SRN: PES1UG19CS019	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

Test case 1 :12 in r0,6 in r1,3 in r2,8 in r3 ,1 in r4

**;Adding 5 values(operands) present in registers
r0,r1,r2,r3,r4**

mov r0,#12 ;Storing value 12 in r0

mov r1,#6 ;Storing value 6 in r1

mov r2,#3 ;Storing value 3 in r2

mov r3,#8 ;Storing value 8 in r3

mov r4,#1 ;Storing value 1 in r4

```

add r5,r0,r1 ; r5 = r0+r1
add r6,r5,r2 ;r6 = r5+r2 = r0+r1+r2
add r7,r6,r3 ;r7 = r6+r3 = r0+r1+r2+r3
add r8,r7,r4 ;r8 = r7+r4 = r0+r1+r2+r3+r4
swi 0x11

```

Test case 2 :71 in r0,10 in r1,32 in r2,5 in r3 ,13 in r4

;Adding 5 values(operands) present in registers
r0,r1,r2,r3,r4

```

mov r0,#71 ;Storing value 71 in r0
mov r1,#10 ;Storing value 10 in r1
mov r2,#32 ;Storing value 32 in r2
mov r3,#5 ;Storing value 5 in r3
mov r4,#13 ;Storing value 13 in r4
add r5,r0,r1 ;r5 = r0+r1
add r6,r5,r2 ;r6 = r5+r2 = r0+r1+r2
add r7,r6,r3 ;r7 = r6+r3 = r0+r1+r2+r3
add r8,r7,r4 ;r8 = r7+r4 = r0+r1+r2+r3+r4
swi 0x11

```

11. 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)

(1) Test case 1 :12 in r0,6 in r1,3 in r2,8 in r3 ,1 in r4

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 :12

R1 :6

R2 :3

R3 :8

R4 :1

R5 :18

R6 :21

R7 :29

R8 :30

R9 :0

R10 (s1) :0

R11 (fp) :0

R12 (ip) :0

R13 (sp) :21504

R14 (lr) :0

R15 (pc) :4132

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

3a_PES10G19CS019.s

;Adding 5 values(operands) present in registers r0,r1,r2,r3,r4

```
00001000:E3A0000C  mov r0,#12 ;Storing value 12 in r0
00001004:E3A01006  mov r1,#6 ;Storing value 6 in r1
00001008:E3A02003  mov r2,#3 ;Storing value 3 in r2
0000100C:E3A03008  mov r3,#8 ;Storing value 8 in r3
00001010:E3A04001  mov r4,#1 ;Storing value 1 in r4

00001014:E0805001  add r5,r0,r1 ; r5 = r0+r1
00001018:E0856002  add r6,r5,r2 ;r6 = r5+r2 = r0+r1+r2
0000101C:E0867003  add r7,r6,r3 ;r7 = r6+r3 = r0+r1+r2+r3
00001020:E0878004  add r8,r7,r4 ;r8 = r7+r4 = r0+r1+r2+r3+r4

00001024:EF000011  swi 0x11
```

OutputView

Console Stdin/Stdout/Stderr

Execution starting ...

Execution ending, Instruction Count:0 Elapsed Time:00:00:00.0049516

Instructions per second:0

(2) Test case 2 :71 in r0,10 in r1,32 in r2,5 in r3 ,13 in r4

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 :71

R1 :10

R2 :32

R3 :5

R4 :13

R5 :81

R6 :113

R7 :118

R8 :131

R9 :0

R10 (s1) :0

R11 (fp) :0

R12 (ip) :0

R13 (sp) :21504

R14 (lr) :0

R15 (pc) :4132

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

3b_PES10G19CS019.s

;Adding 5 values(operands) present in registers r0,r1,r2,r3,r4

```
00001000:E3A00047  mov r0,#71 ;Storing value 71 in r0
00001004:E3A0100A  mov r1,#10 ;Storing value 10 in r1
00001008:E3A02020  mov r2,#32 ;Storing value 32 in r2
0000100C:E3A03005  mov r3,#5 ;Storing value 5 in r3
00001010:E3A0400D  mov r4,#13 ;Storing value 13 in r4

00001014:E0805001  add r5,r0,r1 ; r5 = r0+r1
00001018:E0856002  add r6,r5,r2 ;r6 = r5+r2 = r0+r1+r2
0000101C:E0867003  add r7,r6,r3 ;r7 = r6+r3 = r0+r1+r2+r3
00001020:E0878004  add r8,r7,r4 ;r8 = r7+r4 = r0+r1+r2+r3+r4

00001024:EF000011  swi 0x11
```

OutputView

Console Stdin/Stdout/Stderr

Execution starting ...

Execution ending, Instruction Count:0 Elapsed Time:00:00:00.0049554

Instructions per second:0

Microprocessor and Computer Architecture Laboratory

UE19CS256

4th Semester, Academic Year 2020-21

Date: 27/01/2021

Name : Abhishek Aditya BS	SRN: PES1UG19CS019	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

Test case 1 : value 3 is stored in register r1

`;To check if a number stored in a register is even or odd`

`;If the number is even then store 00 in r0 else store FF in r0`

`mov r1,#3 ;value 3 is stored in the register r1`

`ands r2,r1,#1 ;performing Logical and of value stored in r1 & 1 and updating flags`

```
cmp r2,#0 ;Checking if the value in r2 is 0 ,
by comparing the value in r2 and 0

beq label ;If the value in r2 is 0, then the
number is even, jump to label

mov r0,#0xFF ;If the number is odd, store FF in r0

b end ;jump to end, after storing

label: mov r0,#0x00 ;If the number is even,
store 00 in r0

end: swi 0x11
```

Test case 2 : value 12 is stored in register r1

```
;To check if a number stored in a register is
even or odd

;If the number is even then store 00 in r0 else
store FF in r0

mov r1,#12 ;value 12 is stored in the register r1

ands r2,r1,#1 ;performing Logical and of value
stored in r1 & 1 and updating flags

cmp r2,#0 ;Checking if the value in r2 is 0 ,
by comparing the value in r2 and 0

beq label ;If the value in r2 is 0, then the
number is even, jump to label

mov r0,#0xFF ;If the number is odd, store FF in r0

b end ;jump to end, after storing
```

```

label: mov r0,#0x00 ;If the number is even,
store 00 in r0

end: swi 0x11

```

11. 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)

Test case 1 : value 3 is stored in register r1

The screenshot displays the ARM Simulator interface. The Register Window on the left shows the state of registers R0 through R15. R0 is 000000ff, R1 is 00000003, R2 is 00000001, and R15 (PC) is 0000101c. The CPSR Register shows Negative (N) as 0, Zero (Z) as 0, Carry (C) as 1, and Overflow (V) as 0. The CPU Mode is set to System.

The main window displays the assembly code for the test case. The code is as follows:

```

;To check if a number stored in a register is even or odd
;If the number is even then store 00 in r0 else store FF in r0

00001000:E3A01003  mov r1,#3 ;value 3 is stored in the register r1
00001004:E2112001  ands r2,r1,#1 ;performing Logical and of value stored in r1 & 1 and updating flags
00001008:E3520000  cmp r2,#0 ;Checking if the value in r2 is 0 , by comparing the value in r2 and 0

0000100C:0A000001  beq label ;If the value in r2 is 0, then the number is even, jump to label

00001010:E3A000FF  mov r0,#0xFF ;If the number is odd, store FF in r0
00001014:EA000000  b end ;jump to end, after storing

00001018:E3A00000  label: mov r0,#0x00 ;If the number is even, store 00 in r0

0000101C:EF000011  end: swi 0x11

```

The Output View at the bottom shows the execution starting and ending, with an instruction count of 0 and an elapsed time of 00:00:00.0059581.

Test case 2 : value 12 is stored in register r1

The screenshot shows the ARM Simulator interface. On the left, the 'RegistersView' panel displays the state of various registers. R0 through R14 are at 0x00000000, and R15 (PC) is at 0x000101c. The CPSR register shows flags: Negative (N) is 0, Zero (Z) is 1, Carry (C) is 1, and Overflow (V) is 0. The CPU Mode is set to System. The main window displays assembly code for a program that checks if a number in register r1 is even or odd. The code includes instructions like 'mov r1, #12', 'ands r2, r1, #1', 'cmp r2, #0', 'beq label', 'mov r0, #0xFF', 'b end', 'label: mov r0, #0x00', and 'end: swi 0x11'. The 'OutputView' panel at the bottom shows the execution log, indicating that execution started and ended successfully with 0 instructions per second.

```
4b_PES1UG19CS019.s
;To check if a number stored in a register is even or odd
;If the number is even then store 00 in r0 else store FF in r0

00001000:E3A0100C    mov r1,#12 ;value 12 is stored in the register r1
00001004:E2112001    ands r2,r1,#1 ;performing Logical and of value stored in r1 & 1 and updating flags
00001008:E3520000    cmp r2,#0 ;Checking if the value in r2 is 0 , by comparing the value in r2 and 0

0000100C:0A000001    beq label ;If the value in r2 is 0, then the number is even, jump to label

00001010:E3A000FF    mov r0,#0xFF ;If the number is odd, store FF in r0
00001014:EA000000    b end ;jump to end, after storing

00001018:E3A00000    label: mov r0,#0x00 ;If the number is even, store 00 in r0

0000101C:EF000011    end: swi 0x11

RegistersView
General Purpose Floating Point
Hexadecimal
Unsigned Decimal
Signed Decimal
R0 : 00000000
R1 : 0000000c
R2 : 00000000
R3 : 00000000
R4 : 00000000
R5 : 00000000
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): 0000101c
-----
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
-----

OutputView
Console Stdin/Stdout/Stderr
Execution starting ...
Execution ending, Instruction Count:0 Elapsed Time:00:00:00.0049922
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 : Abhishek Aditya BS

Name : Abhishek Aditya BS

SRN : PES1UG19CS019

Section : A

Date : 27/01/2021