Microprocessor and Computer Architecture Laboratory

UE19CS256

4th Semester, Academic Year 2020-21

Date: 24/1/21

Name: Adithya M S	SRN:PES1UG19CS027	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.

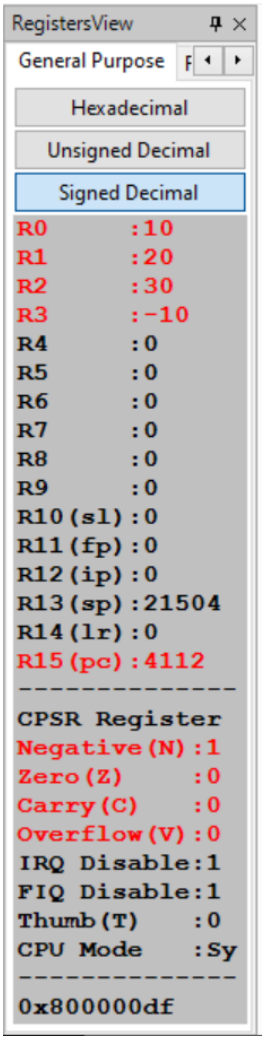
I. ARM Assembly Code for each program
II. Final Output Screen Shot (Register Window, Output window)
The output should be verified with 2 test cases (one example shown in class, one example of own choice)

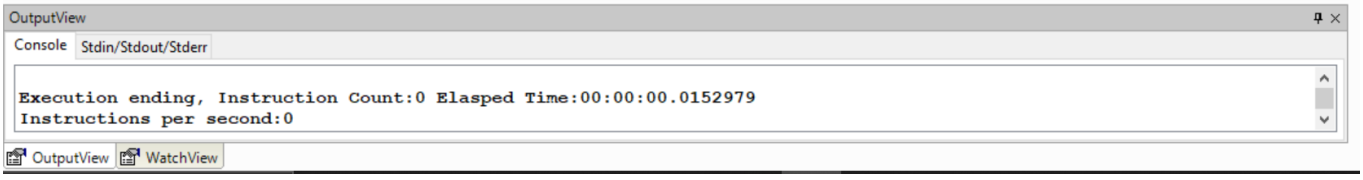
Example1:

1_PES1UG19CS027.s

```
00001000:E3A0000A    mov R0,#0x0A
00001004:E3A01014    mov R1,#0x14
00001008:E0802001    add R2,R0,R1

0000100C:E0503001    subs R3,R0,R1
00001010:EF000011    swi 0x11
```



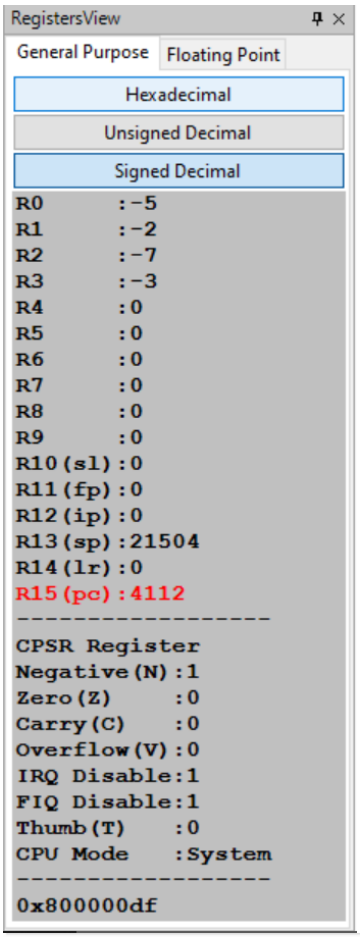


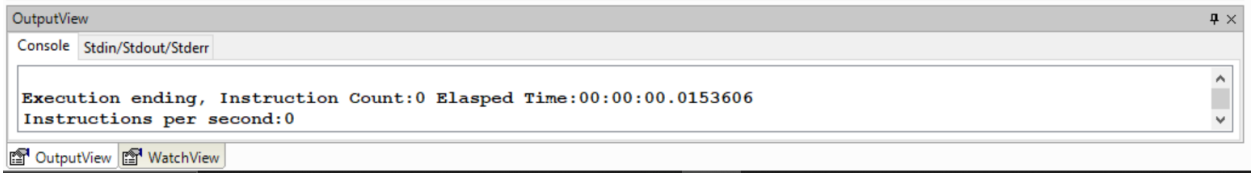
Example 2:

1_PES1UG19CS027.s

```
00001000:E3E00004    mov R0,#-5
00001004:E3E01001    mov R1,#-2
00001008:E0802001    add R2,R0,R1

0000100C:E0503001    subs R3,R0,R1
00001010:EF000011    swi 0x11
```





Microprocessor and Computer Architecture Laboratory

UE19CS256

4th Semester, Academic Year 2020-21

Date:24/1/21

Name: Adithya M S	SRN:PES1UG19CS027	Section: A

Week#____1_____ Program Number: ____2___

Title of the Program

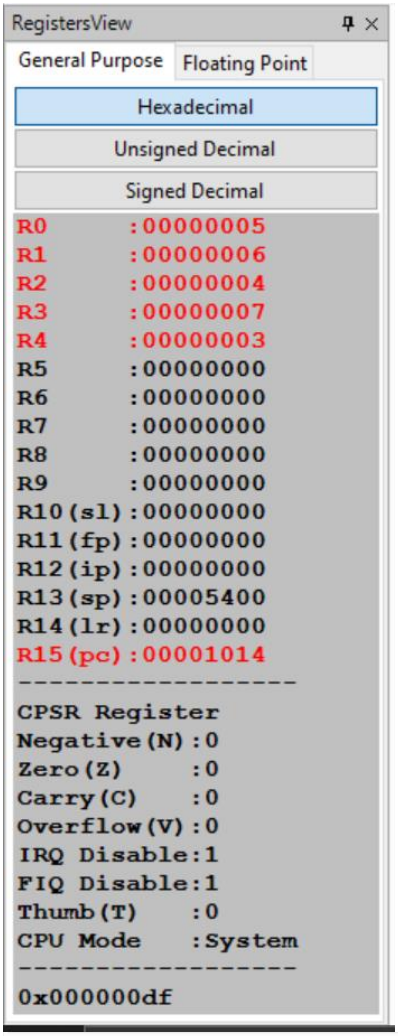
Write an ALP to demonstrate logical operations. All operands are in registers.

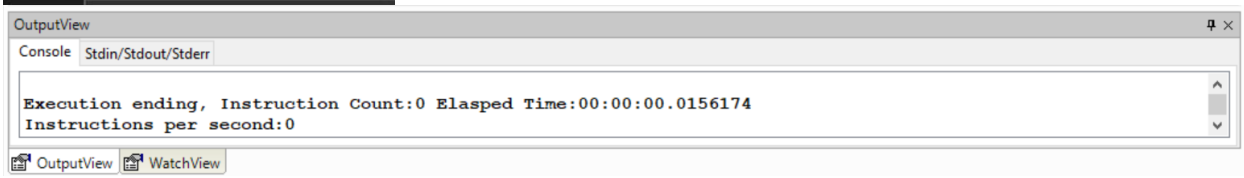
I. ARM Assembly Code for each program
II. Final Output Screen Shot (Register Window, Output window)
The output should be verified with 2 test cases (one example shown in class, one example of own choice)

Example 1:

2_PES1UG19CS027.s

```
00001000:E3A00005    mov R0,#0x05
00001004:E3A01006    mov R1,#0x06
00001008:E0002001    and R2,R0,R1
0000100C:E1803001    orr R3,R0,R1
00001010:E0204001    eor R4,R0,R1
00001014:EF000011    swi 0x11
```

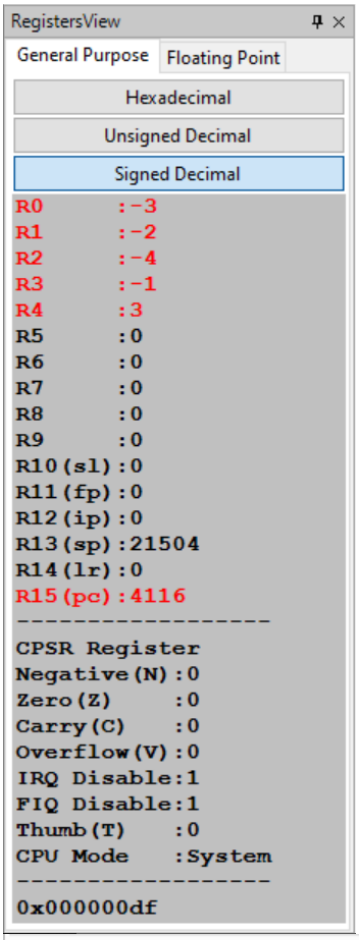


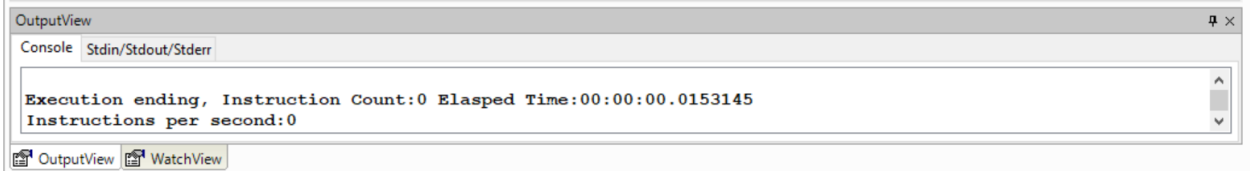


Example 2:

2_PES1UG19CS027.s

```
00001000:E3E00002    mov R0,#-3
00001004:E3E01001    mov R1,#-2
00001008:E0002001    and R2,R0,R1
0000100C:E1803001    orr R3,R0,R1
00001010:E0204001    eor R4,R0,R1
00001014:EF000011    swi 0x11
```





Microprocessor and Computer Architecture Laboratory

UE19CS256

4th Semester, Academic Year 2020-21

Date: 24/1/21

Name: Adithya M S	SRN: PES1UG19CS027	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)

Example 1:

3_PES1UG19CS027.s

```
00001000:E3A00005    mov r0,#0x05
00001004:E3A01006    mov r1,#0x06
00001008:E3A02007    mov r2,#0x07
0000100C:E3A03006    mov r3,#0x06
00001010:E3A04015    mov r4,#0x15

00001014:E0805001    add r5,r0,r1
00001018:E0856002    add r6,r5,r2
0000101C:E0867003    add r7,r6,r3
00001020:E0878004    add r8,r7,r4
00001024:EF000011    swi 0x11
```

RegistersView

General Purpose | Floating Point

Hexadecimal | Unsigned Decimal | Signed Decimal

```
R0      :5
R1      :6
R2      :7
R3      :6
R4      :21
R5      :11
R6      :18
R7      :24
R8      :45
R9      :0
R10(sl):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
------------------
0x000000df
```

OutputView

Console | Stdin/Stdout/Stderr

```
Execution ending, Instruction Count:0 Elasped Time:00:00:00.0153770
Instructions per second:0
```

OutputView | WatchView

Example 2:

3_PES1UG19CS027.s

```
00001000:E3E00000    mov r0,#-1
00001004:E3A0100A    mov r1,#10
00001008:E3E02004    mov r2,#-5
0000100C:E3A0300E    mov r3,#14
00001010:E3A04003    mov r4,#3

00001014:E0805001    add r5,r0,r1
00001018:E0856002    add r6,r5,r2
0000101C:E0867003    add r7,r6,r3
00001020:E0878004    add r8,r7,r4
00001024:EF000011    swi 0x11
```

RegistersView

General Purpose | Floating Point

Hexadecimal | Unsigned Decimal | Signed Decimal

```
R0      :-1
R1      :10
R2      :-5
R3      :14
R4      :3
R5      :9
R6      :4
R7      :18
R8      :21
R9      :0
R10(sl):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
------------------
0x000000df
```

OutputView

Console | Stdin/Stdout/Stderr

```
Execution ending, Instruction Count:0 Elasped Time:00:00:00.0152974
Instructions per second:0
```

OutputView | WatchView

Microprocessor and Computer Architecture Laboratory

UE19CS256

4th Semester, Academic Year 2020-21

Date: 24/1/21

Name: Adithya M S	SRN: PES1UG19CS027	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)

Example 1:

4_PES1UG19CS027.s

```
00001000:E3A01006    mov r1,#0x06
00001004:E2012001    and r2,r1,#1
00001008:E3520000    cmp r2,#0
0000100C:0A000001    beq loop
00001010:E3A000FF    mov r0,#0xFF
00001014:EF000011    swi 0x11

00001018:E3A00000    loop: mov r0,#0
0000101C:EF000011    swi 0x11
```

RegistersView

General Purpose | Floating Point

Hexadecimal | Unsigned Decimal | Signed Decimal

```
R0       :00000000
R1       :00000006
R2       :00000000
R3       :00000000
R4       :00000000
R5       :00000000
R6       :00000000
R7       :00000000
R8       :00000000
R9       :00000000
R10(sl):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
------------------
0x600000df
```

OutputView

Console | Stdin/Stdout/Stderr

```
Execution ending, Instruction Count:0 Elasped Time:00:00:00.0153435
Instructions per second:0
```

OutputView | WatchView

Example 2:

4_PES1UG19CS027.s

```
00001000:E3A01005    mov r1,#0x05
00001004:E2012001    and r2,r1,#1
00001008:E3520000    cmp r2,#0
0000100C:0A000001    beq loop
00001010:E3A000FF    mov r0,#0xFF
00001014:EF000011    swi 0x11

00001018:E3A00000    loop: mov r0,#0
0000101C:EF000011    swi 0x11
```

RegistersView

General Purpose | Floating Point

Hexadecimal | Unsigned Decimal | Signed Decimal

```
R0       :000000ff
R1       :00000005
R2       :00000001
R3       :00000000
R4       :00000000
R5       :00000000
R6       :00000000
R7       :00000000
R8       :00000000
R9       :00000000
R10(sl):00000000
R11(fp):00000000
R12(ip):00000000
R13(sp):00005400
R14(lr):00000000
R15(pc):00001014
------------------
CPSR Register
Negative(N):0
Zero(Z)     :0
Carry(C)    :1
Overflow(V):0
IRQ Disable:1
FIQ Disable:1
Thumb(T)    :0
CPU Mode    :System
------------------
0x200000df
```

OutputView

Console | Stdin/Stdout/Stderr

```
Execution starting ...

Execution ending, Instruction Count:0 Elasped Time:00:00:00
```

OutputView | WatchView

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: Adithya M S
SRN: PES1UG19CS027
Section: A
Date: 24/1/21