Microprocessor and Computer Architecture Laboratory

UE19CS256

4th Semester, Academic Year 2020-21

Date: 24/1/21

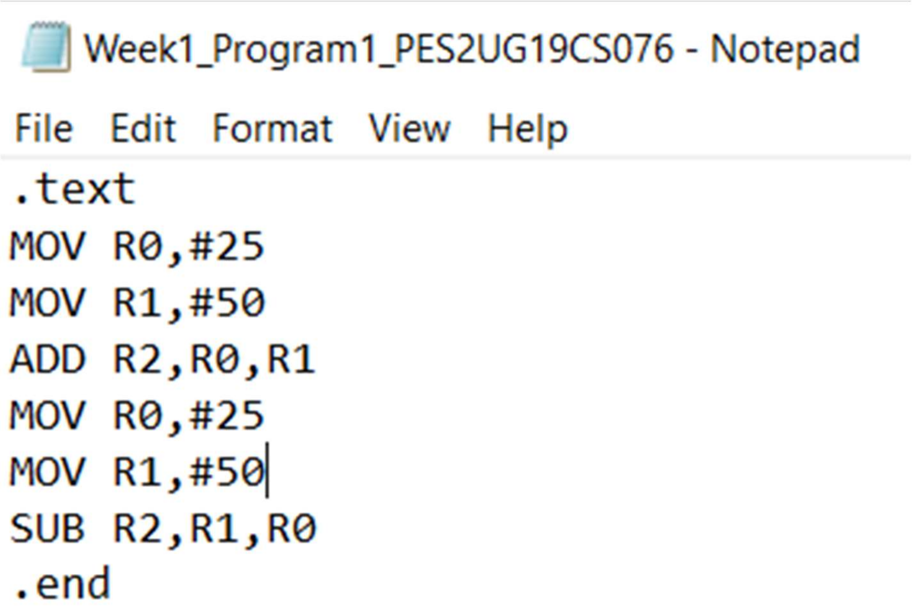
Name: B.Pravena	SRN: PES2UG19CS076	Section: B

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.

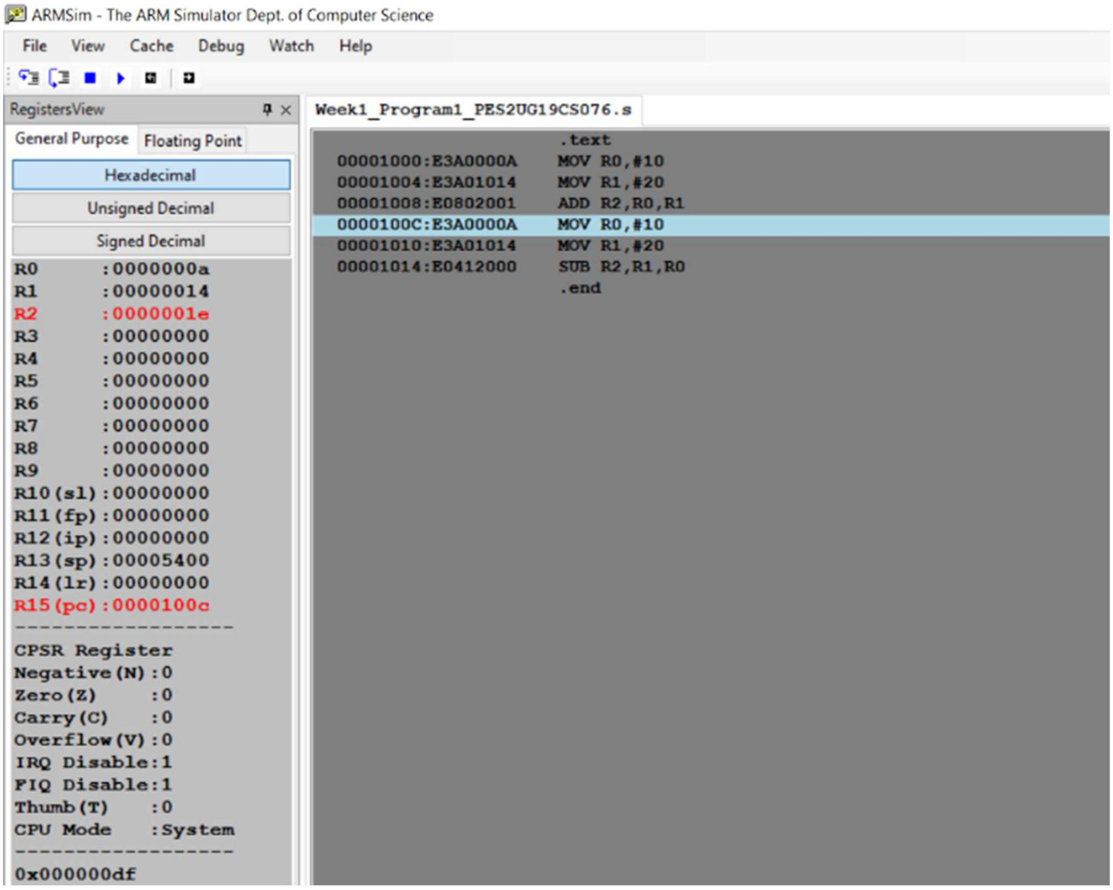
I. ARM Assembly Code



Week1_Program1_PES2UG19CS076 - Notepad
File Edit Format View Help

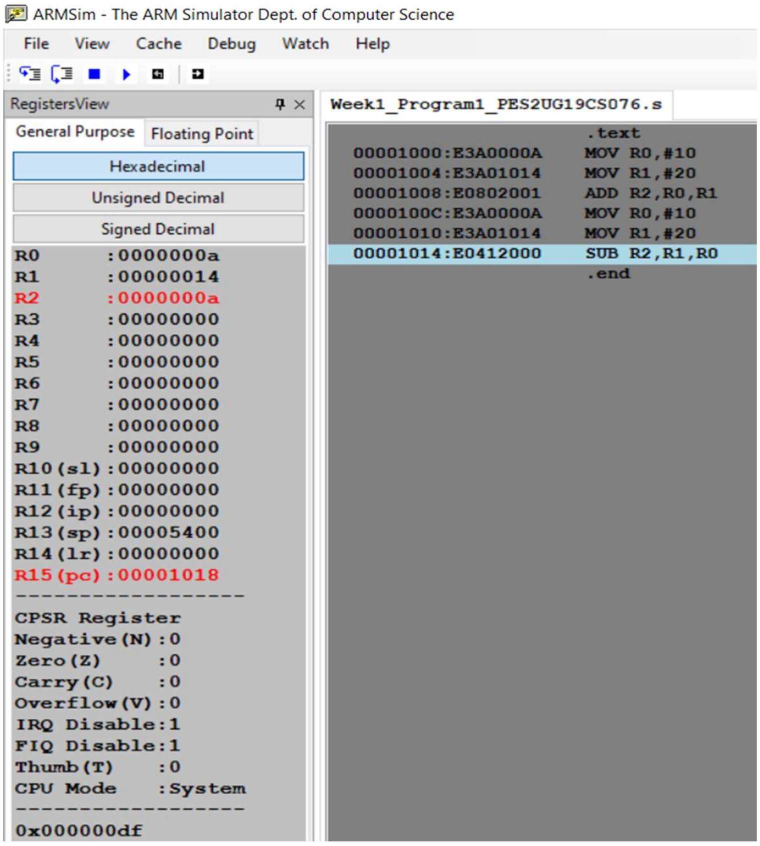
```
.text
MOV R0,#25
MOV R1,#50
ADD R2,R0,R1
MOV R0,#25
MOV R1,#50
SUB R2,R1,R0
.end
```

II. Output Screen Shot



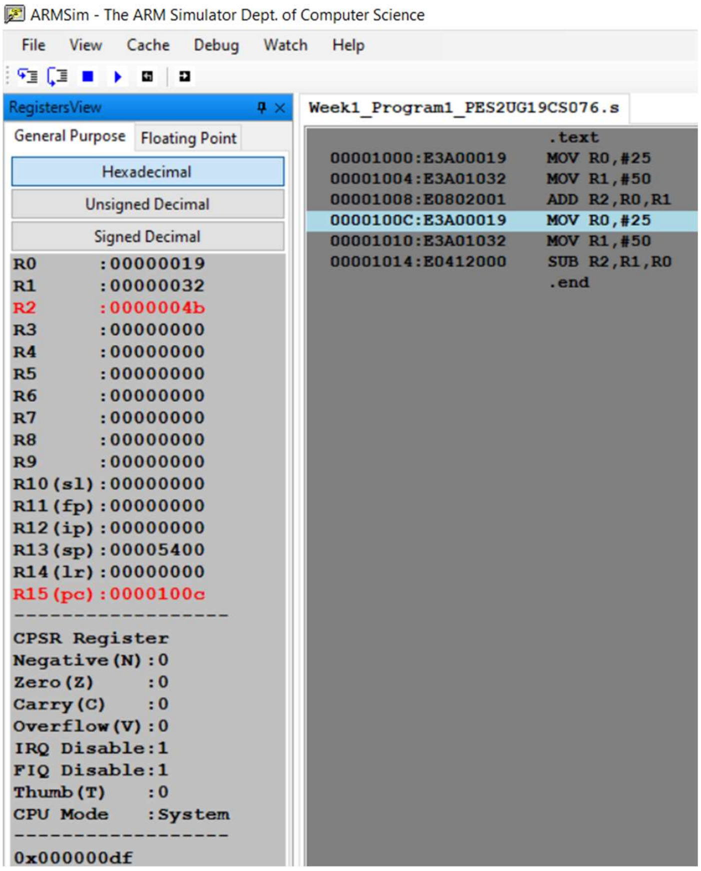
```
R0      :0000000a
R1      :00000014
R2      :0000001e
R3      :00000000
R4      :00000000
R5      :00000000
R6      :00000000
R7      :00000000
R8      :00000000
R9      :00000000
R10(sl):00000000
R11(fp):00000000
R12(ip):00000000
R13(sp):00005400
R14(lr):00000000
R15(pc):0000100c
------------------
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
```

```
                    .text
00001000:E3A0000A   MOV R0,#10
00001004:E3A01014   MOV R1,#20
00001008:E0802001   ADD R2,R0,R1
0000100C:E3A0000A   MOV R0,#10
00001010:E3A01014   MOV R1,#20
00001014:E0412000   SUB R2,R1,R0
                    .end
```



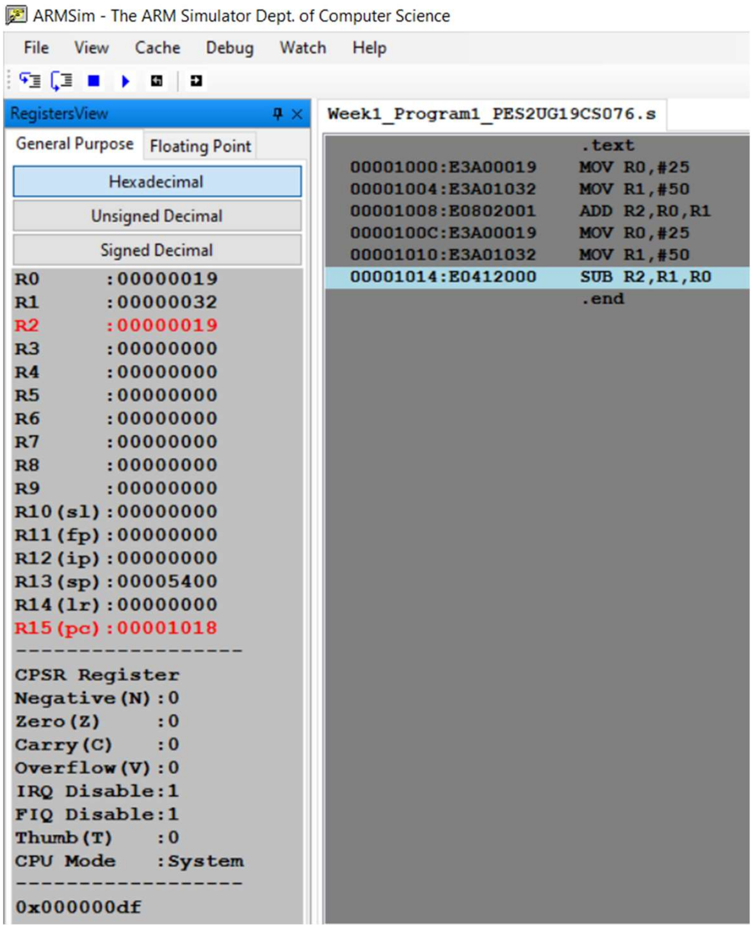
```
R0      :0000000a
R1      :00000014
R2      :0000000a
R3      :00000000
R4      :00000000
R5      :00000000
R6      :00000000
R7      :00000000
R8      :00000000
R9      :00000000
R10(sl):00000000
R11(fp):00000000
R12(ip):00000000
R13(sp):00005400
R14(lr):00000000
R15(pc):00001018
------------------
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
```

```
                    .text
00001000:E3A0000A   MOV R0,#10
00001004:E3A01014   MOV R1,#20
00001008:E0802001   ADD R2,R0,R1
0000100C:E3A0000A   MOV R0,#10
00001010:E3A01014   MOV R1,#20
00001014:E0412000   SUB R2,R1,R0
                    .end
```



```
R0        :00000019
R1        :00000032
R2        :0000004b
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):0000100c
------------------
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
```

```
                            .text
   00001000:E3A00019       MOV R0,#25
   00001004:E3A01032       MOV R1,#50
   00001008:E0802001       ADD R2,R0,R1
   0000100C:E3A00019       MOV R0,#25
   00001010:E3A01032       MOV R1,#50
   00001014:E0412000       SUB R2,R1,R0
                            .end
```



```
R0        :00000019
R1        :00000032
R2        :00000019
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):00001018
------------------
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
```

```
                            .text
   00001000:E3A00019       MOV R0,#25
   00001004:E3A01032       MOV R1,#50
   00001008:E0802001       ADD R2,R0,R1
   0000100C:E3A00019       MOV R0,#25
   00001010:E3A01032       MOV R1,#50
   00001014:E0412000       SUB R2,R1,R0
                            .end
```

III. Output table for each program

R0	R1	Arithmetic operation	Result
0x0a	0x14	ADD	0x1e
0x0a	0x14	SUB	0x0a
0x19	0x32	ADD	0x4b
0x19	0x32	SUB	0x19

Microprocessor and Computer Architecture Laboratory

UE19CS256

4th Semester, Academic Year 2020-21

Date: 24/1/21

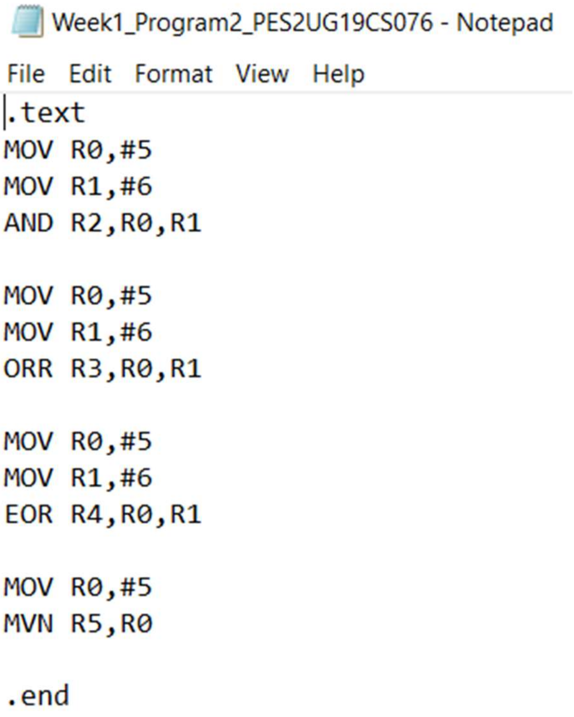
Name: B.Pravena	SRN: PES2UG19CS076	Section: B

Week#____1_______ Program Number: ____2___

Title of the Program

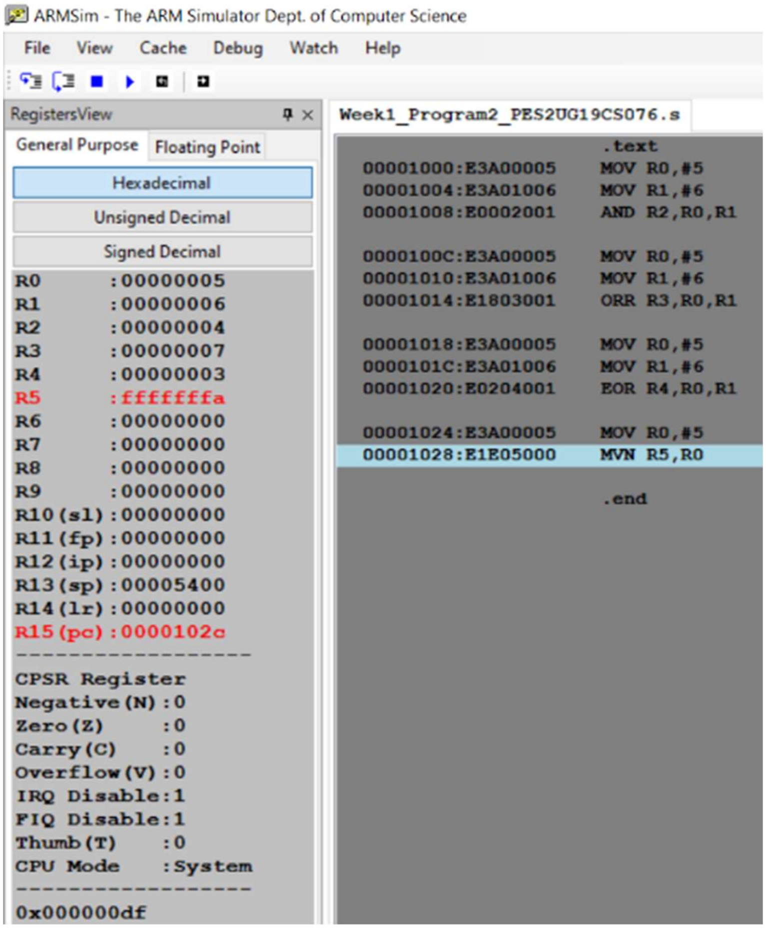
Write an ALP to demonstrate logical operations. All operands are in registers.

I. ARM Assembly Code


Week1_Program2_PES2UG19CS076 - Notepad
File Edit Format View Help

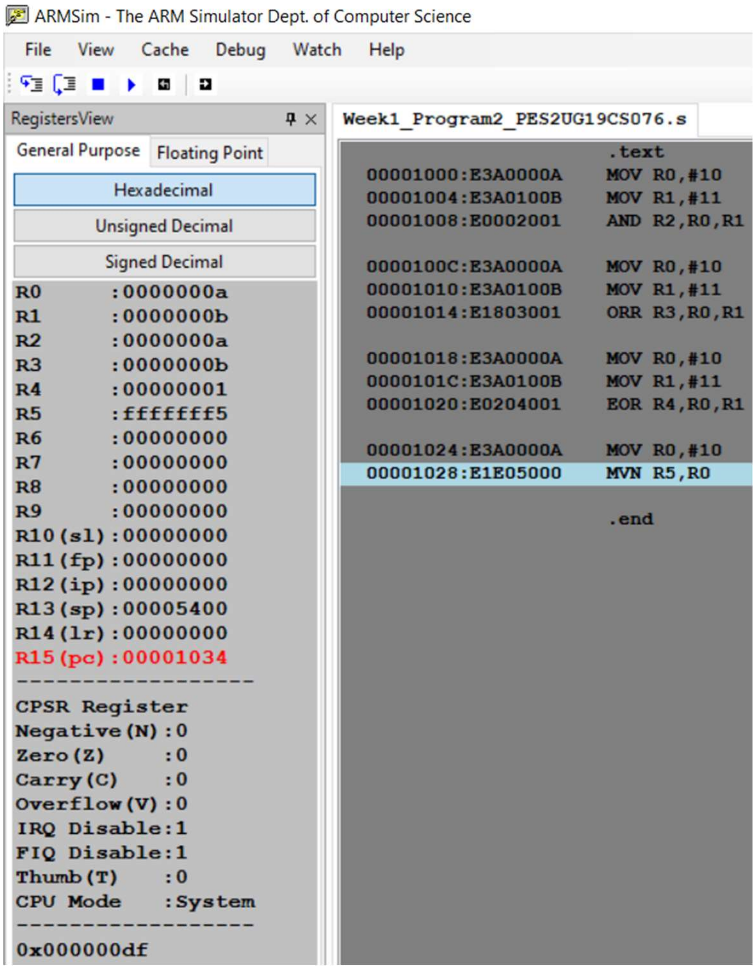
```
.text
MOV R0,#5
MOV R1,#6
AND R2,R0,R1

MOV R0,#5
MOV R1,#6
ORR R3,R0,R1

MOV R0,#5
MOV R1,#6
EOR R4,R0,R1

MOV R0,#5
MVN R5,R0

.end
```

II. Output Screen Shot


ARMSim - The ARM Simulator Dept. of Computer Science
File View Cache Debug Watch Help
RegistersView
General Purpose
Floating Point
Hexadecimal
Unsigned Decimal
Signed Decimal
Week1_Program2_PES2UG19CS076.s

```
R0        :00000005
R1        :00000006
R2        :00000004
R3        :00000007
R4        :00000003
R5        :fffffffa
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):0000102c
------------------
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
```

```
                          .text
00001000:E3A00005         MOV R0,#5
00001004:E3A01006         MOV R1,#6
00001008:E0002001         AND R2,R0,R1

0000100C:E3A00005         MOV R0,#5
00001010:E3A01006         MOV R1,#6
00001014:E1803001         ORR R3,R0,R1

00001018:E3A00005         MOV R0,#5
0000101C:E3A01006         MOV R1,#6
00001020:E0204001         EOR R4,R0,R1

00001024:E3A00005         MOV R0,#5
00001028:E1E05000         MVN R5,R0

                          .end
```



```
R0        :0000000a
R1        :0000000b
R2        :0000000a
R3        :0000000b
R4        :00000001
R5        :fffffff5
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):00001034
------------------
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
```

```
                              .text
00001000:E3A0000A         MOV R0,#10
00001004:E3A0100B         MOV R1,#11
00001008:E0002001         AND R2,R0,R1

0000100C:E3A0000A         MOV R0,#10
00001010:E3A0100B         MOV R1,#11
00001014:E1803001         ORR R3,R0,R1

00001018:E3A0000A         MOV R0,#10
0000101C:E3A0100B         MOV R1,#11
00001020:E0204001         EOR R4,R0,R1

00001024:E3A0000A         MOV R0,#10
00001028:E1E05000         MVN R5,R0

                              .end
```

III. Output table for each program

R0	R1	Logical operation	Instruction	Result
0x05	0x06	AND	AND	0x04
0x05	0x06	OR	ORR	0x07
0x05	0x06	EX-OR	EOR	0x03
0x05		NOT	MVN	0xfffffffa
0x0a	0x0b	AND	AND	0x0a
0x0a	0x0b	OR	ORR	0x0b
0x0a	0x0b	EX-OR	EOR	0x01
0x0a		NOT	MVN	0xfffffff5

Microprocessor and Computer Architecture Laboratory

UE19CS256

4th Semester, Academic Year 2020-21

Date: 24/1/21

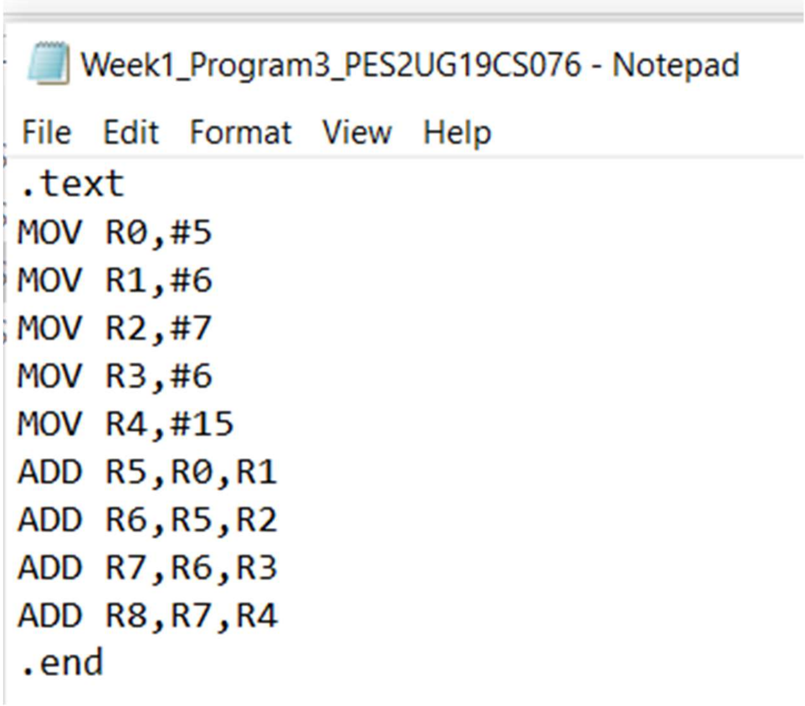
Name: B.Pravena	SRN: PES2UG19CS076	Section: B

Week#____1_______ Program Number: ____3___

Title of the Program

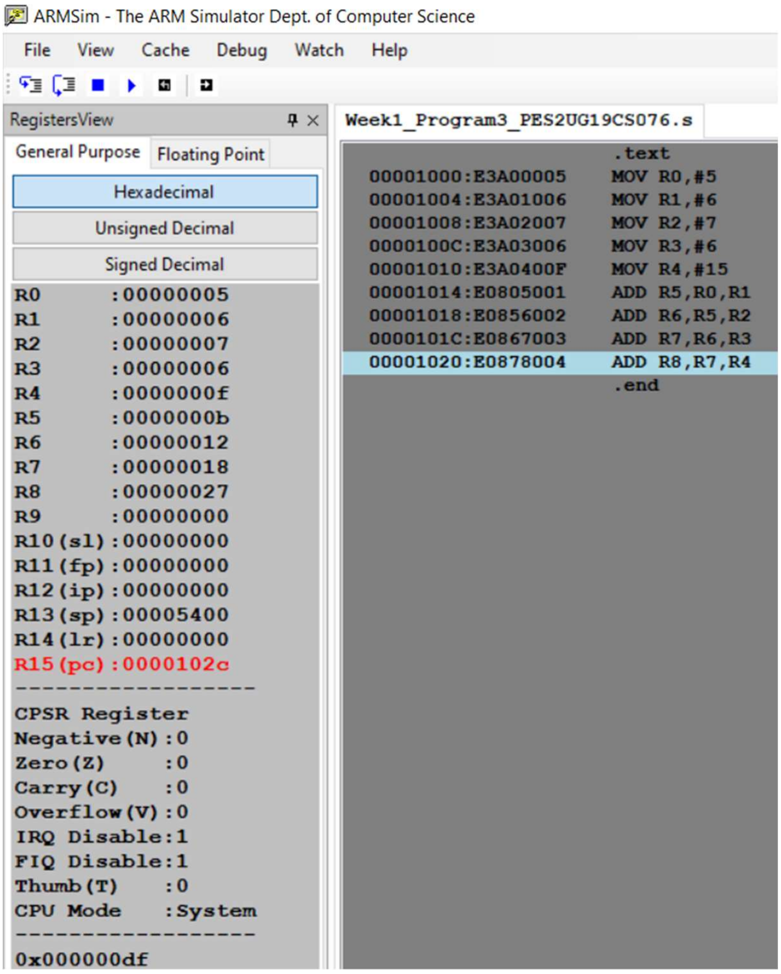
Write an ALP to add 5 numbers where values are present in registers.

I. ARM Assembly Code


Week1_Program3_PES2UG19CS076 - Notepad
File Edit Format View Help

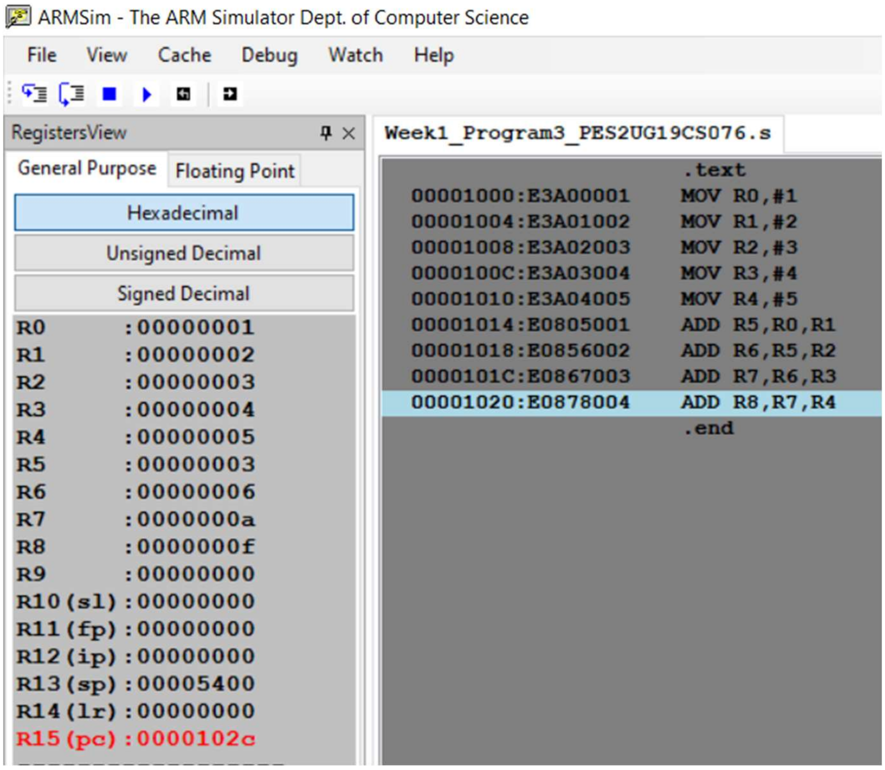
```
.text
MOV R0,#5
MOV R1,#6
MOV R2,#7
MOV R3,#6
MOV R4,#15
ADD R5,R0,R1
ADD R6,R5,R2
ADD R7,R6,R3
ADD R8,R7,R4
.end
```

II. Output Screen Shot


ARMSim - The ARM Simulator Dept. of Computer Science
File View Cache Debug Watch Help
RegistersView
General Purpose
Floating Point
Hexadecimal
Unsigned Decimal
Signed Decimal
Week1_Program3_PES2UG19CS076.s

```
R0      :00000005
R1      :00000006
R2      :00000007
R3      :00000006
R4      :0000000f
R5      :0000000b
R6      :00000012
R7      :00000018
R8      :00000027
R9      :00000000
R10(sl):00000000
R11(fp):00000000
R12(ip):00000000
R13(sp):00005400
R14(lr):00000000
R15(pc):0000102c
------------------
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
```

```
                      .text
00001000:E3A00005     MOV R0,#5
00001004:E3A01006     MOV R1,#6
00001008:E3A02007     MOV R2,#7
0000100C:E3A03006     MOV R3,#6
00001010:E3A0400F     MOV R4,#15
00001014:E0805001     ADD R5,R0,R1
00001018:E0856002     ADD R6,R5,R2
0000101C:E0867003     ADD R7,R6,R3
00001020:E0878004     ADD R8,R7,R4
                      .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
Week1_Program3_PES2UG19CS076.s

```
R0      :00000001
R1      :00000002
R2      :00000003
R3      :00000004
R4      :00000005
R5      :00000003
R6      :00000006
R7      :0000000a
R8      :0000000f
R9      :00000000
R10(sl):00000000
R11(fp):00000000
R12(ip):00000000
R13(sp):00005400
R14(lr):00000000
R15(pc):0000102c
```

```
                      .text
00001000:E3A00001     MOV R0,#1
00001004:E3A01002     MOV R1,#2
00001008:E3A02003     MOV R2,#3
0000100C:E3A03004     MOV R3,#4
00001010:E3A04005     MOV R4,#5
00001014:E0805001     ADD R5,R0,R1
00001018:E0856002     ADD R6,R5,R2
0000101C:E0867003     ADD R7,R6,R3
00001020:E0878004     ADD R8,R7,R4
                      .end
```

III. Output table for each program

R0		0x05
R1		0x06
R2		0x07
R3		0x06
R4		0x0f
R5	R0+R1	0x0b
R6	R5+R2	0x12
R7	R6+R3	0x18
R8	R7+R4	0x27
R0		0x01
R1		0x02
R2		0x03
R3		0x04
R4		0x05
R5	R0+R1	0x03
R6	R5+R2	0x06
R7	R6+R3	0x0a
R8	R7+R4	0x0f

Microprocessor and Computer Architecture Laboratory

UE19CS256

4th Semester, Academic Year 2020-21

Date: 24/1/21

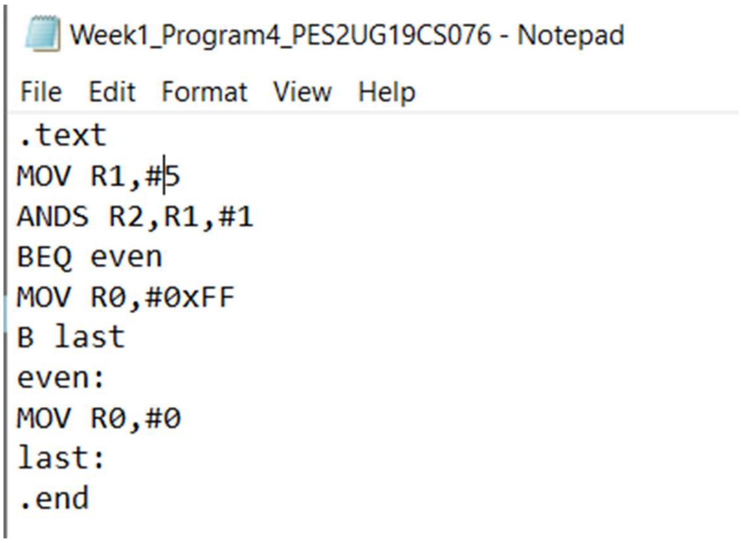
Name: B.Pravena	SRN: PES2UG19CS076	Section: B

Week#____1_______ Program Number: ____4___

Title of the Program

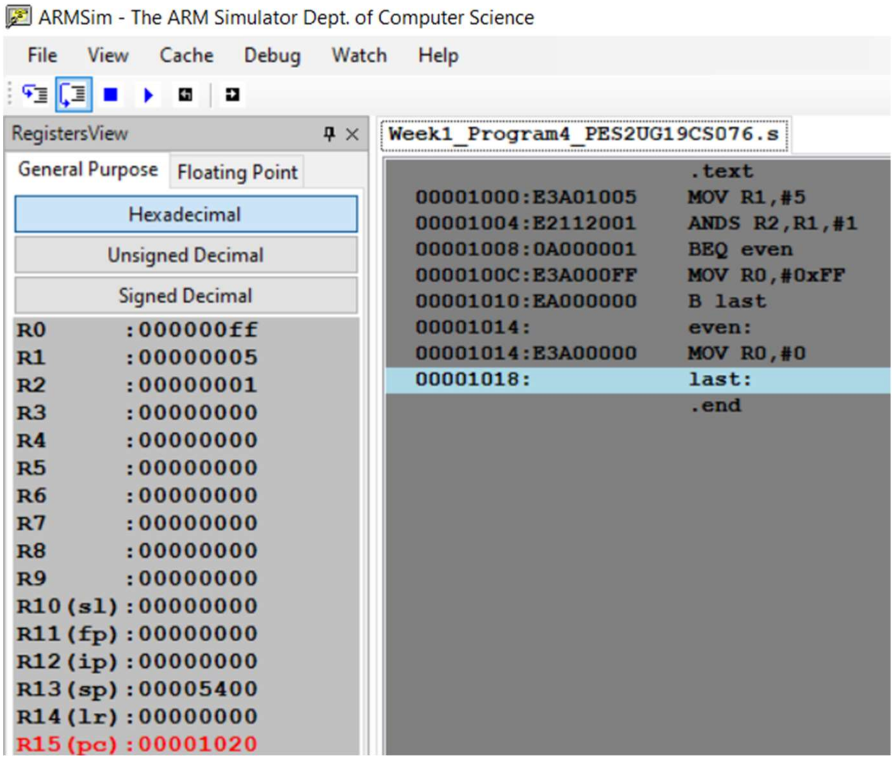
Write an ALP using ARM instruction set to check if a number stored in a register is even or odd. If even, store 00 in R0, else store FF in R0

I. ARM Assembly Code for each program


Week1_Program4_PES2UG19CS076 - Notepad
File Edit Format View Help

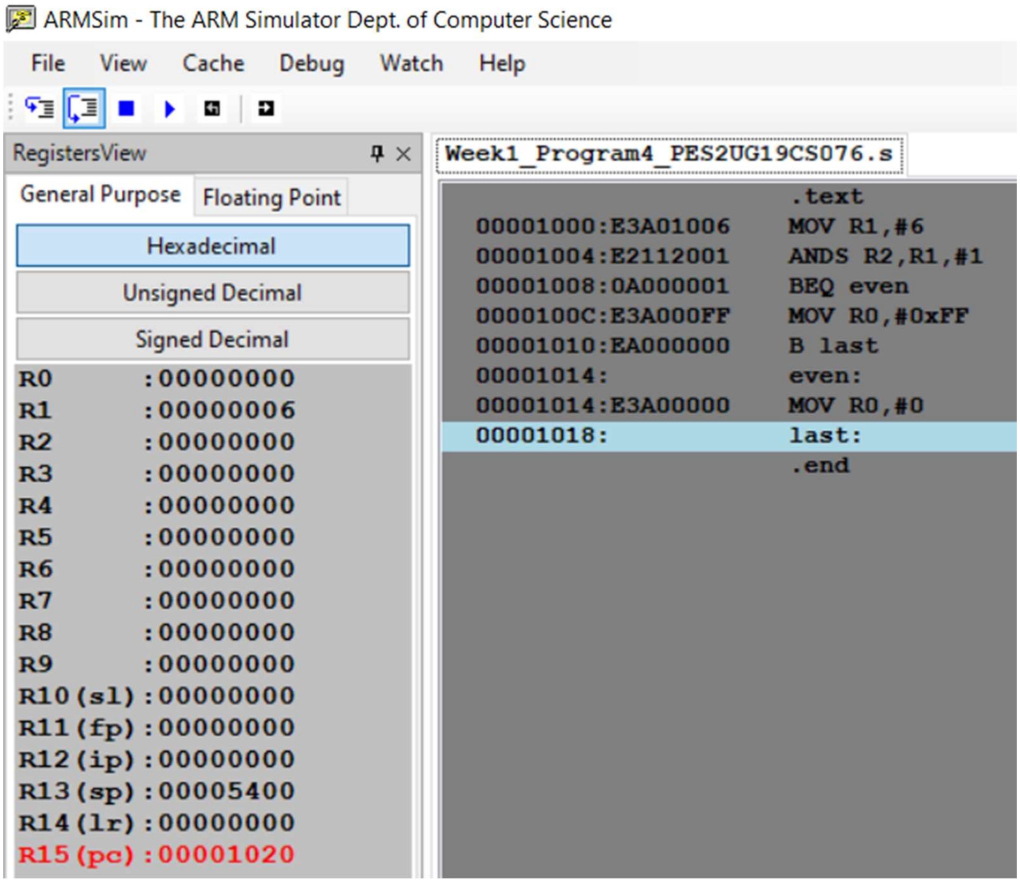
```
.text
MOV R1,#5
ANDS R2,R1,#1
BEQ even
MOV R0,#0xFF
B last
even:
MOV R0,#0
last:
.end
```

II. Output Screen Shot



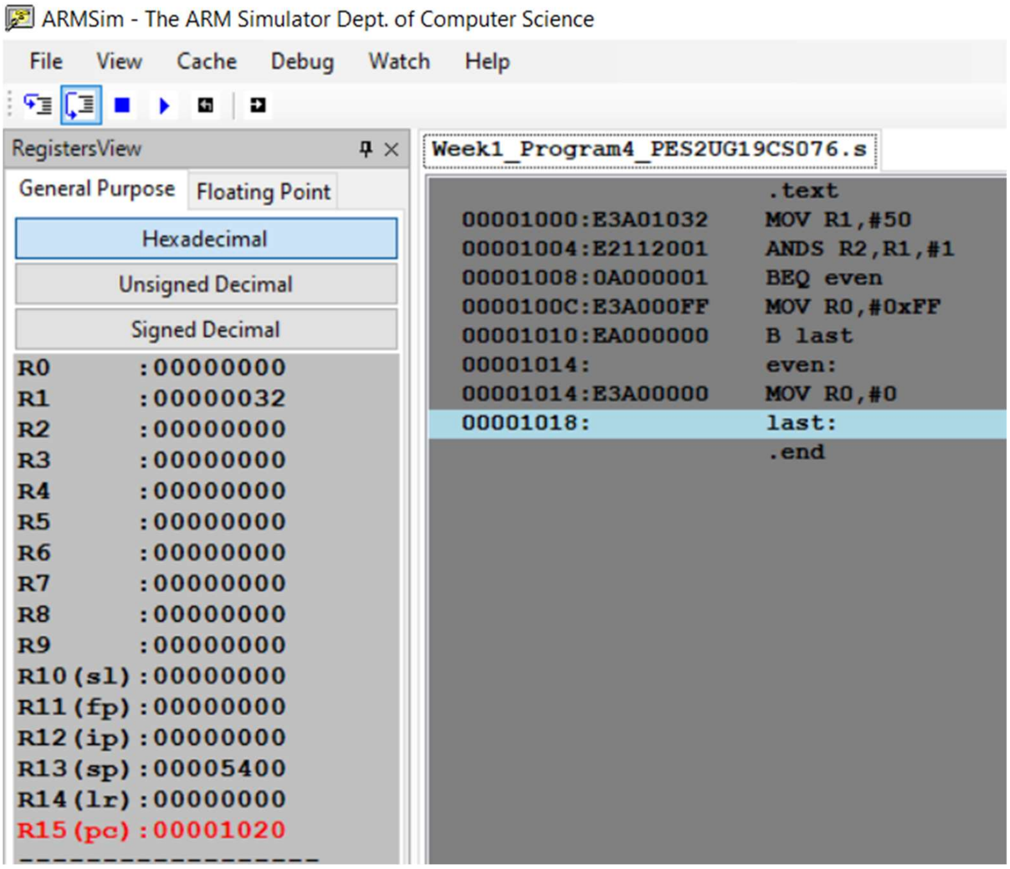
```
R0       :000000ff
R1       :00000005
R2       :00000001
R3       :00000000
R4       :00000000
R5       :00000000
R6       :00000000
R7       :00000000
R8       :00000000
R9       :00000000
R10(sl)  :00000000
R11(fp)  :00000000
R12(ip)  :00000000
R13(sp)  :00005400
R14(lr)  :00000000
R15(pc)  :00001020
```

```
                        .text
00001000:E3A01005       MOV R1,#5
00001004:E2112001       ANDS R2,R1,#1
00001008:0A000001       BEQ even
0000100C:E3A000FF       MOV R0,#0xFF
00001010:EA000000       B last
00001014:               even:
00001014:E3A00000       MOV R0,#0
00001018:               last:
                        .end
```



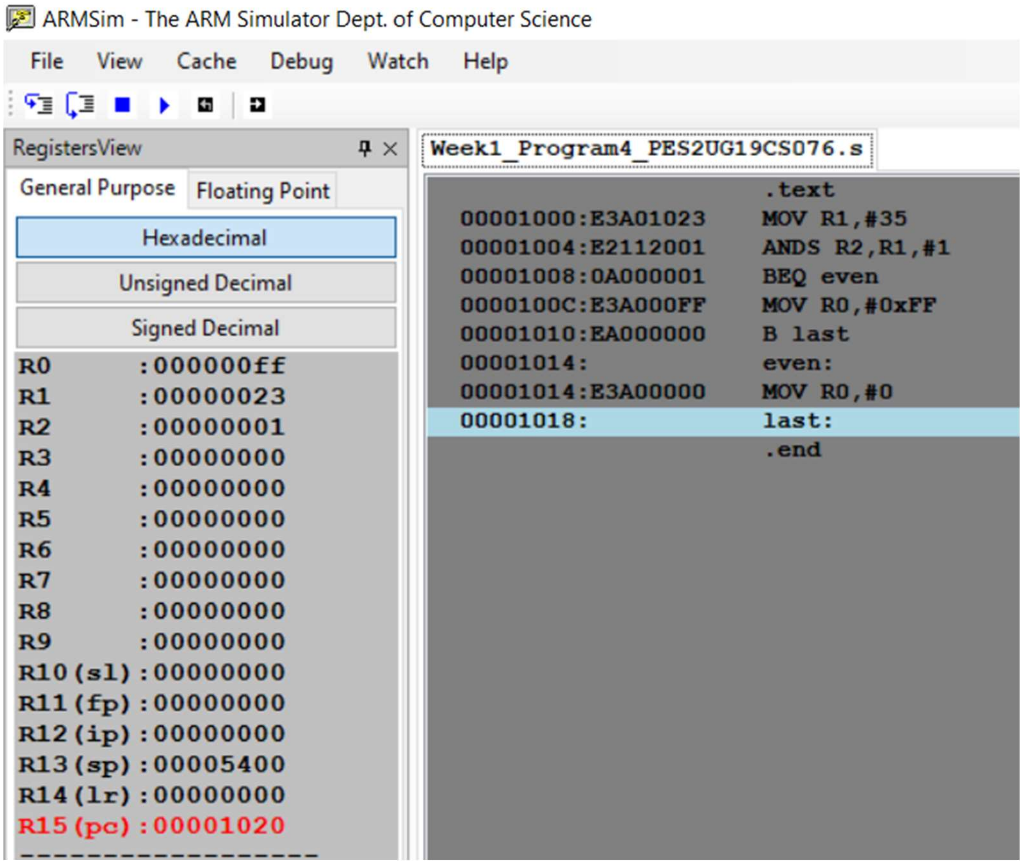
```
R0       :00000000
R1       :00000006
R2       :00000000
R3       :00000000
R4       :00000000
R5       :00000000
R6       :00000000
R7       :00000000
R8       :00000000
R9       :00000000
R10(sl)  :00000000
R11(fp)  :00000000
R12(ip)  :00000000
R13(sp)  :00005400
R14(lr)  :00000000
R15(pc)  :00001020
```

```
                        .text
00001000:E3A01006       MOV R1,#6
00001004:E2112001       ANDS R2,R1,#1
00001008:0A000001       BEQ even
0000100C:E3A000FF       MOV R0,#0xFF
00001010:EA000000       B last
00001014:               even:
00001014:E3A00000       MOV R0,#0
00001018:               last:
                        .end
```



```
R0       :00000000
R1       :00000032
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):00001020
------------------
```

```
                        .text
00001000:E3A01032       MOV R1,#50
00001004:E2112001       ANDS R2,R1,#1
00001008:0A000001       BEQ even
0000100C:E3A000FF       MOV R0,#0xFF
00001010:EA000000       B last
00001014:               even:
00001014:E3A00000       MOV R0,#0
00001018:               last:
                        .end
```



```
R0       :000000ff
R1       :00000023
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):00001020
------------------
```

```
                        .text
00001000:E3A01023       MOV R1,#35
00001004:E2112001       ANDS R2,R1,#1
00001008:0A000001       BEQ even
0000100C:E3A000FF       MOV R0,#0xFF
00001010:EA000000       B last
00001014:               even:
00001014:E3A00000       MOV R0,#0
00001018:               last:
                        .end
```

III. Output table for each program

Case 1	R1		0x06
	R2	After AND operation	0x00
	R0		0x00
Case 2	R1		0x05
	R2	After AND operation	0x01
	R0		0xFF
Case 1	R1		0x32
	R2	After AND operation	0x00
	R0		0x00
Case 2	R1		0x23
	R2	After AND operation	0x01
	R0		0xFF