

## EECS 2021 Lab B

Q1:

$A = 12, b = 10, c = 15, d = 7$

$E = c * (a - b) - d * (c - a)$

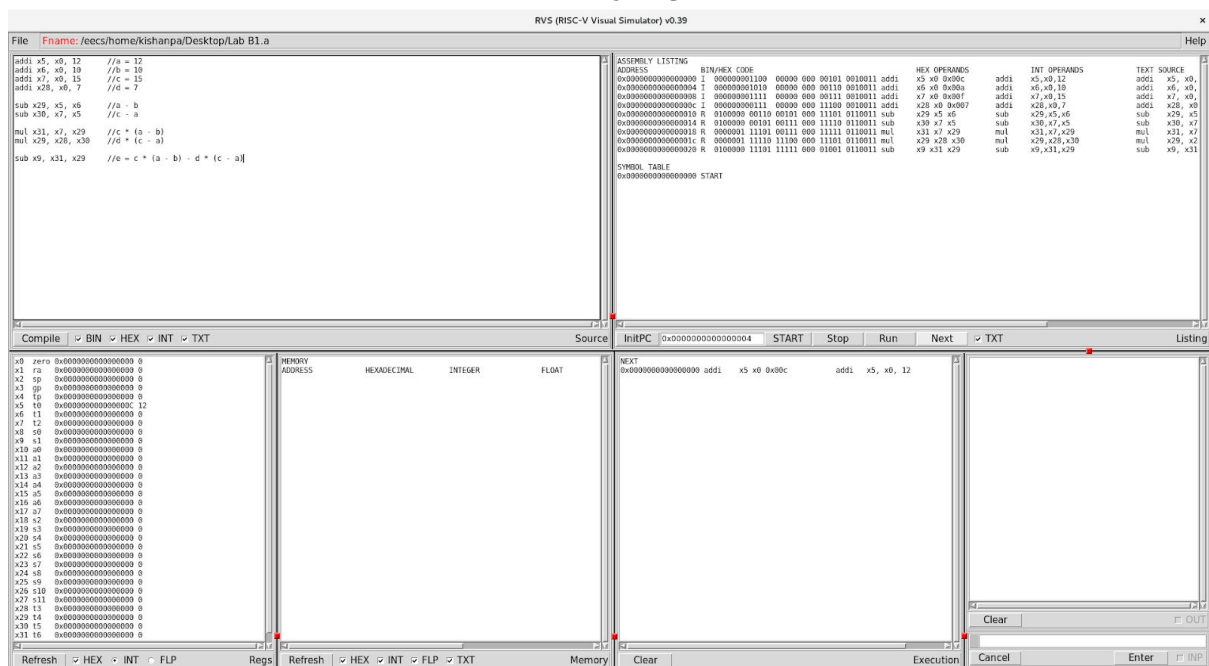
$E = 15 * (12 - 10) - 8 (15 - 12)$

$E = 15(2) - 7(3)$

$E = 30 - 21$

$E = 9$

**Step 1: Store 12 into x5 which is a temporary register**



**Step 2: Store 10 into x6 which is a temporary register**



RVS (RISC-V Visual Simulator) v0.39

File: /eecs/home/kishanpa/Desktop/Lab B1.a

ASSEMBLY LISTING

ADDRESS	BIN/HEX CODE	HEX OPERANDS	INT OPERANDS	TEXT SOURCE
0x0000000000000000	0000000000000000	addi x5, x0, 12	addi x5, x0, 12	addi x5, x0, 12
0x0000000000000004	0000000000000004	addi x6, x0, 10	addi x6, x0, 10	addi x6, x0, 10
0x0000000000000008	0000000000000008	addi x7, x0, 15	addi x7, x0, 15	addi x7, x0, 15
0x000000000000000c	000000000000000c	addi x28, x0, 7	addi x28, x0, 7	addi x28, x0, 7
0x0000000000000010	0000000000000010	sub x29, x5, x6	sub x29, x5, x6	sub x29, x5, x6
0x0000000000000014	0000000000000014	sub x30, x7, x5	sub x30, x7, x5	sub x30, x7, x5
0x0000000000000018	0000000000000018	mul x31, x7, x29	mul x31, x7, x29	mul x31, x7, x29
0x000000000000001c	000000000000001c	mul x29, x28, x30	mul x29, x28, x30	mul x29, x28, x30
0x0000000000000020	0000000000000020	sub x9, x31, x29	sub x9, x31, x29	sub x9, x31, x29

SYMBOL TABLE

0x0000000000000000 START

Source: InitPC: 0x0000000000000010 START Stop Run Next TXT Listing

Refresh HEX INT FLP Regs Refresh HEX INT FLP TXT Memory Clear Execution Cancel Enter INP

Step 5: Store a - b into x29 which is a temporary register

RVS (RISC-V Visual Simulator) v0.39

File: /eecs/home/kishanpa/Desktop/Lab B1.a

ASSEMBLY LISTING

ADDRESS	BIN/HEX CODE	HEX OPERANDS	INT OPERANDS	TEXT SOURCE
0x0000000000000000	0000000000000000	addi x5, x0, 12	addi x5, x0, 12	addi x5, x0, 12
0x0000000000000004	0000000000000004	addi x6, x0, 10	addi x6, x0, 10	addi x6, x0, 10
0x0000000000000008	0000000000000008	addi x7, x0, 15	addi x7, x0, 15	addi x7, x0, 15
0x000000000000000c	000000000000000c	addi x28, x0, 7	addi x28, x0, 7	addi x28, x0, 7
0x0000000000000010	0000000000000010	sub x29, x5, x6	sub x29, x5, x6	sub x29, x5, x6
0x0000000000000014	0000000000000014	sub x30, x7, x5	sub x30, x7, x5	sub x30, x7, x5
0x0000000000000018	0000000000000018	mul x31, x7, x29	mul x31, x7, x29	mul x31, x7, x29
0x000000000000001c	000000000000001c	mul x29, x28, x30	mul x29, x28, x30	mul x29, x28, x30
0x0000000000000020	0000000000000020	sub x9, x31, x29	sub x9, x31, x29	sub x9, x31, x29

SYMBOL TABLE

0x0000000000000000 START

Source: InitPC: 0x0000000000000014 START Stop Run Next TXT Listing

Refresh HEX INT FLP Regs Refresh HEX INT FLP TXT Memory Clear Execution Cancel Enter INP

Step 6: Store c - a into x30 which is a temporary register

RVS (RISC-V Visual Simulator) v0.39

File: /home/kishanpa/Desktop/Lab B1.a

```

addi x5, x0, 12 //a = 12
addi x6, x0, 10 //b = 10
addi x7, x0, 15 //c = 15
addi x28, x6, 7 //d = 7
sub x29, x5, x6 //a - b
sub x30, x7, x5 //c - a
mul x31, x7, x29 //c * (a - b)
mul x29, x28, x30 //d * (c - a)
sub x9, x31, x29 //e = c * (a - b) - d * (c - a)

```

ASSEMBLY LISTING

ADDRESS	BIN/HEX CODE	HEX OPERANDS	INT OPERANDS	TEXT SOURCE
0x0000000000000000	0000000000000000	addi x5,x0,12	addi x5,x0,12	addi x5,x0,12
0x0000000000000004	0000000000000004	addi x6,x0,10	addi x6,x0,10	addi x6,x0,10
0x0000000000000008	0000000000000008	addi x7,x0,15	addi x7,x0,15	addi x7,x0,15
0x000000000000000c	000000000000000c	addi x28,x6,7	addi x28,x6,7	addi x28,x6,7
0x0000000000000010	0000000000000010	sub x29,x5,x6	sub x29,x5,x6	sub x29,x5,x6
0x0000000000000014	0000000000000014	sub x30,x7,x5	sub x30,x7,x5	sub x30,x7,x5
0x0000000000000018	0000000000000018	mul x31,x7,x29	mul x31,x7,x29	mul x31,x7,x29
0x000000000000001c	000000000000001c	mul x29,x28,x30	mul x29,x28,x30	mul x29,x28,x30
0x0000000000000020	0000000000000020	sub x9,x31,x29	sub x9,x31,x29	sub x9,x31,x29

SYMBOL TABLE

0x0000000000000000 START

Source: InitPC: 0x0000000000000018 START Stop Run Next TXT Listing

Refresh HEX INT FLP Regs Refresh HEX INT FLP TXT Memory Clear Execution Cancel Enter INP

Step 7: Store  $c * (a - b)$  into x31 which is a temporary register

RVS (RISC-V Visual Simulator) v0.39

File: /home/kishanpa/Desktop/Lab B1.a

```

addi x5, x0, 12 //a = 12
addi x6, x0, 10 //b = 10
addi x7, x0, 15 //c = 15
addi x28, x6, 7 //d = 7
sub x29, x5, x6 //a - b
sub x30, x7, x5 //c - a
mul x31, x7, x29 //c * (a - b)
mul x29, x28, x30 //d * (c - a)
sub x9, x31, x29 //e = c * (a - b) - d * (c - a)

```

ASSEMBLY LISTING

ADDRESS	BIN/HEX CODE	HEX OPERANDS	INT OPERANDS	TEXT SOURCE
0x0000000000000000	0000000000000000	addi x5,x0,12	addi x5,x0,12	addi x5,x0,12
0x0000000000000004	0000000000000004	addi x6,x0,10	addi x6,x0,10	addi x6,x0,10
0x0000000000000008	0000000000000008	addi x7,x0,15	addi x7,x0,15	addi x7,x0,15
0x000000000000000c	000000000000000c	addi x28,x6,7	addi x28,x6,7	addi x28,x6,7
0x0000000000000010	0000000000000010	sub x29,x5,x6	sub x29,x5,x6	sub x29,x5,x6
0x0000000000000014	0000000000000014	sub x30,x7,x5	sub x30,x7,x5	sub x30,x7,x5
0x0000000000000018	0000000000000018	mul x31,x7,x29	mul x31,x7,x29	mul x31,x7,x29
0x000000000000001c	000000000000001c	mul x29,x28,x30	mul x29,x28,x30	mul x29,x28,x30
0x0000000000000020	0000000000000020	sub x9,x31,x29	sub x9,x31,x29	sub x9,x31,x29

SYMBOL TABLE

0x0000000000000000 START

Source: InitPC: 0x000000000000001c START Stop Run Next TXT Listing

Refresh HEX INT FLP Regs Refresh HEX INT FLP TXT Memory Clear Execution Cancel Enter INP

Step 8: Store  $d * (c - a)$  into x29 which is a temporary register

RVS (RISC-V Visual Simulator) v0.39

File: /eecs/home/kishanpa/Desktop/Lab B1.a

ASSEMBLY LISTING

```
addi x5, x0, 12 //a = 12
addi x6, x0, 10 //b = 10
addi x7, x0, 15 //c = 15
addi x28, x6, 7 //d = 7
sub x29, x5, x6 //a - b
sub x30, x7, x5 //c - a
mul x31, x7, x29 //c * (a - b)
mul x29, x28, x30 //d * (c - a)
sub x9, x31, x29 //e = c * (a - b) - d * (c - a)
```

Source

InitPC: 0x0000000000000020

START

Stop

Run

Next

Cancel

Enter

INP

**Step 9: Save  $e = c * (a - b) - d * (c - a)$  into x9 which is a saved register**

RVS (RISC-V Visual Simulator) v0.39

File: /eecs/home/kishanpa/Desktop/Lab B1.a

ASSEMBLY LISTING

```
addi x5, x0, 12 //a = 12
addi x6, x0, 10 //b = 10
addi x7, x0, 15 //c = 15
addi x28, x6, 7 //d = 7
sub x29, x5, x6 //a - b
sub x30, x7, x5 //c - a
mul x31, x7, x29 //c * (a - b)
mul x29, x28, x30 //d * (c - a)
sub x9, x31, x29 //e = c * (a - b) - d * (c - a)
```

Source

InitPC: 0x0000000000000024

START

Stop

Run

Next

Cancel

Enter

INP

**Q2:**

**Step 1: Store 12 into x8. Since ori compared to x0 will just store the immediate number**

RV5 (RISC-V Visual Simulator) v0.39

File: Fname:/eecs/home/kishanpa/Desktop/Lab B2.a

ASSEMBLY LISTING

ADDRESS	BIN/HEX CODE	HEX OPERANDS	INT OPERANDS	TEXT SOURCE
0x0000000000000000	00000001100 00000 110 01000 0010011	ori x8 x0 0x0c	ori x8,x0,12	ori x8,x0,
0x0000000000000004	011000000000 00000 110 00111 0010011	x7 x0 0x600	ori x7,x0,1536	ori x7,x0,
0x0000000000000008	00000000 01000 00111 110 01001 0110011	ori x9 x7 x8	ori x9,x7,x8	ori x9,x7,
0x000000000000000c	000011111111 01001 100 01010 0000011	x10 x8 0xffff	ori x10,x8,255	ori x10,x8,

SYMBOL TABLE

0x0000000000000000 START

Compile BIN HEX INT TXT Source InitPC 0x0000000000000004 START Stop Run Next TXT Listing

Refresh HEX INT FLP Regs Refresh HEX INT FLP TXT Memory Clear Execution Cancel Enter INP

**Step 2: Store 1536 into x7. Since ori compared to x0 will just store the immediate number**

RV5 (RISC-V Visual Simulator) v0.39

File: Fname:/eecs/home/kishanpa/Desktop/Lab B2.a

ASSEMBLY LISTING

ADDRESS	BIN/HEX CODE	HEX OPERANDS	INT OPERANDS	TEXT SOURCE
0x0000000000000000	00000001100 00000 110 01000 0010011	ori x8 x0 0x0c	ori x8,x0,12	ori x8,x0,
0x0000000000000004	011000000000 00000 110 00111 0010011	x7 x0 0x600	ori x7,x0,1536	ori x7,x0,
0x0000000000000008	00000000 01000 00111 110 01001 0110011	ori x9 x7 x8	ori x9,x7,x8	ori x9,x7,
0x000000000000000c	000011111111 01001 100 01010 0000011	x10 x8 0xffff	ori x10,x8,255	ori x10,x8,

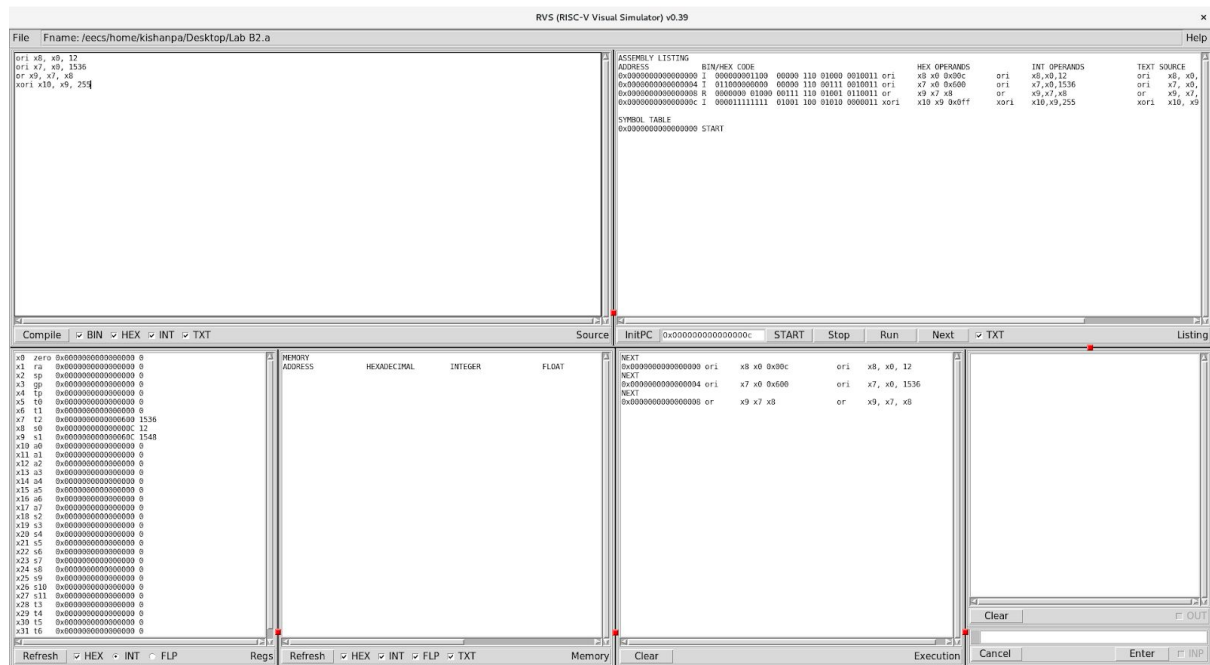
SYMBOL TABLE

0x0000000000000000 START

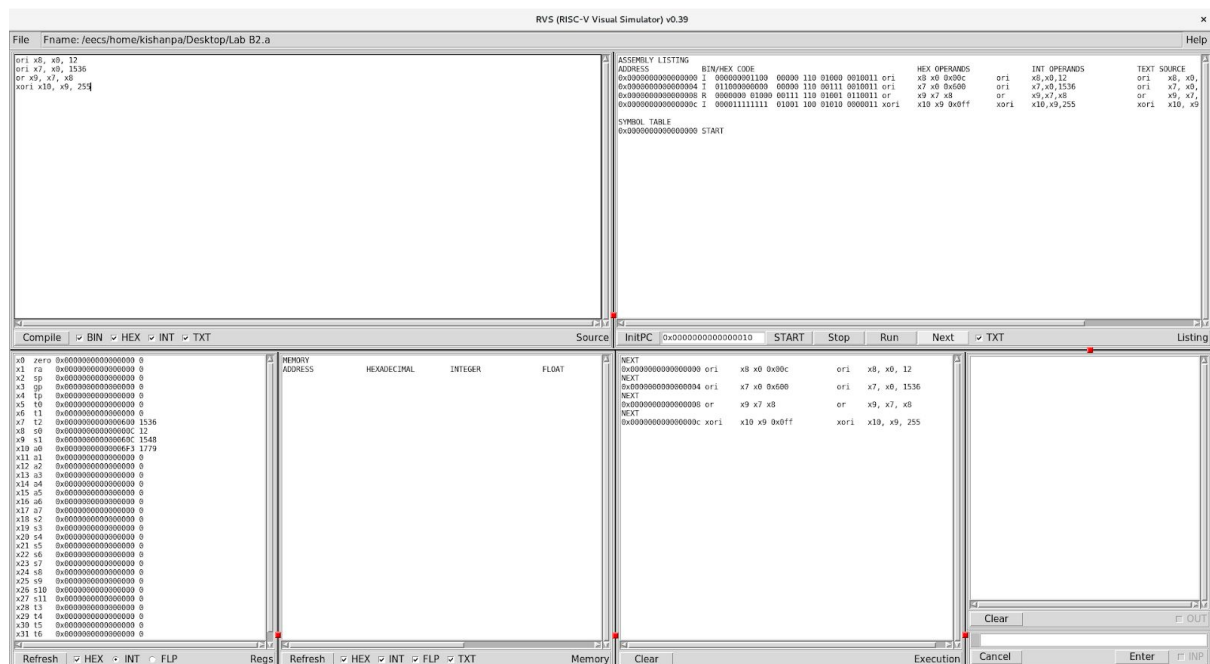
Compile BIN HEX INT TXT Source InitPC 0x0000000000000008 START Stop Run Next TXT Listing

Refresh HEX INT FLP Regs Refresh HEX INT FLP TXT Memory Clear Execution Cancel Enter INP

**Step 3: Store the addition of x8 and 1536 ( x7 ) into x9. Since or will save when either number has bit 1, it is the same as adding**



**Step 4: Store the inversion of 8 bits into x10. Since xori compared to 255 which in binary is all 1s, will just store the xor value of the bit which is the same as inversion.**



**This lab taught me ways to store values without using the obvious operators and logical operators only. So, it teaches us how to do arithmetic operations without the use of arithmetic operators and via the use of logical operators instead.**