

HW4.3. RISC-V Bit Manipulation and Endianness

Feel free to check out the [guide](#) that we have prepared to help you in this problem.

1: What does the following code do? Feel free to test random numbers on the two registers.

```
xor x1, x1, x2
xor x2, x1, x2
xor x1, x1, x2
```

Q1.1:

☐ (a) Sets **x1** to **x1 & x2**, and sets **x2** to **x1 | x2**

☐ (b) Sets **x1** to **x1 ^ x2**, and sets **x2** to **~(x1 ^ x2)**

☐ (c) Sets **x1** to **~(x1 ^ x2)**, and sets **x2** to **x1 ^ x2**

☐ (d) Sets **x2** to -1

☐ (e) Sets **x1 = x2**

☐ (f) Sets **x1** to 0

☐ (g) Sets **x2** to 0

☐ (h) Sets **x1** to **x1 | x2**, and sets **x2** to **x1 & x2**

☐ (i) Swaps **x1** and **x2**

☐ (j) Sets **x1** to -1

2: After the following code is run, what is contained in **x12**? Answer with 8 digits of hexadecimal.

*Hint: Note that RISC-V registers contain 32-bit data. **srai** performs a sign-extended right shift.*

```
li x10, 0x84FF
slli x12, x10, 0x10
srai x13, x12, 0x10
srli x14, x12, 0x08
and x12, x13, x14
```

Q2.1:

x12=0x

?

3: Given the following code sequence (assume **x5** is somewhere on the stack):

```
addi x11, x0, -1261
sw x11, 0(x5)
sh x11, 2(x5)
lb x12, 1(x5)
lbu x13, 2(x5)
```

What is the 32-bit value stored in **x12** and **x13**? Answer with 8 digits of hexadecimal.

Hint: RISC-V uses little endian. Take note of sign-extension when doing load instructions.

Q3.1:

x12=0x

?

Q3.2:

x13=0x

?

Save & Grade 20 attempts left

Save only

Additional attempts available with new variants ?

Homework 4

Assessment overview

Total points: 20/100

Score:

20%

Question

Value: 12

History:

Awarded points: 0/12

Report an error in this question ?

Previous question

Next question

Attached files

No attached files

Attach a file ?

Attach text ?