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:

- \bigcirc (a) Sets x1 to x1 & x2, and sets x2 to x1 | x2
- \bigcirc (b) Sets x1 to x1 ^ x2, and sets x2 to \sim (x1 ^ x2)
- \bigcirc (c) Sets x1 to \sim (x1 $^$ x2), and sets x2 to x1 $^$ x2
- (d) Sets x2 to -1
- \bigcirc (e) Sets x1 = x2
- (f) Sets **x1** to 0
- (g) Sets x2 to 0
- \bigcirc (h) Sets x1 to x1 | x2, and sets x2 to x1 & x2
- (i) Swaps x1 and x2
- (j) Sets x1 to -1

Q2.1:

x12=0x

2: After the following code is run, what is contained in $\times 12$? 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
```

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.

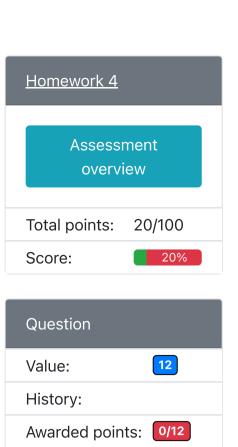


Save & Grade 20 attempts left

Save only

Additional attempts available with new variants 3

3



Previous question

Report an error in this

question 모

Next question

