CSE 3666

Homework 1

Due Date: By the end of Sat, 2/3/2024.

Total points: 100

Submit your work in a PDF file in HuskyCT.

Only use only the instructions in RV32I.

1. For each instruction in the table, write 8 hexadecimal digits that represent the 32 bits in the destination register after the instruction is executed.

   Assume s0 is 0x98ABCD6A, s1 is 0x20FF5A98.

   Find out the answers by working on bits/hexadecimal digits. Do not convert large numbers from hexadecimal to decimal.

   The submission should include the steps you take to find the answers.

   Check your answers in RARS.

Instructions	Dest. reg. in 8 hexadecimal digits
`add  t0, s0, s1`	
`and  t1, s0, s1`	
`or   t2, s0, s1`	
`xor  t3, s0, s1`	
`addi t4, s0, 0x2FA`	
`andi t5, s0, -16`	
`slli t6, s0, 12`	
`srai s2, s0, 8`	

2. Write RISC-V instructions to reverse the order of bytes in register `s2` and save the results in `s4`. For example, if `s2` is 0x12345678, the four bytes in s2 are 0x12, 0x34, 0x56, and 0x78. Register `s4` should be 0x78563412 after the execution of the instruction sequence. Use temporary registers like `t0` and `t1` to save intermediate values.

   Explain your strategy and test your code in RARS.

   Always write brief comments with your code. See the examples in slides and the code given in the next questions.

3. The following RISC-V instructions calculate the Hamming weight (the number of 1's) of `s0`. The result is saved in register `s1`.

```
      addi   s1, x0, 0          # s1 = 0
      addi   t0, x0, 1          # Use t0 as mask to test each bit in s0
loop:
      and    t1, s0, t0         # extract a bit with the mask
      beq    t1, x0, skip       # if the bit is 0, do not increment s1
      addi   s1, s1, 1          # increment the counter
skip:
      slli   t0, t0, 1          # shift mask to left by 1
      bne    t0, x0, loop       # if the mask is not 0, continue
```

   a. If s0 is 0xFF00FF00, how many instructions are executed? Does the number of executed instructions depend on the number of 1's in s0? Does it depend on the location of 1's? Explain your answers.

   b. There are many ways to compute Hamming weight. We could test the most significant bit (bit 31) of s0. For example, extract bit 31 with an AND instruction, and compare it with 0. This is similar to the method in the code given. However, we can save one instruction. **If we treat s0 as a 2's complement number, s0 is less than 0 if and only if bit 31 in s0 is 1.** Using this method, write RISC-V instructions to compute the Hamming weight of s0. Explain your method in comments. We can start with the following two instructions. How many instructions are executed if s0 is 0xFF00FF00? Explain how you get the answers.

```
      addi  s1, x0, 0        # s1 = 0
      add   t0, x0, s0       # make a copy so s0 is not changed
```

Continued on the next page.

4. Translate the following C code to RISC-V assembly code. Assume that `a`, `i`, and `r` are stored in registers s1, s2, and s3, respectively, and their values are already set in these registers. All the variables are signed. Write brief comments in your code. Clearly mark the instructions that control the loop (for example, using different colors), the instructions in if branch, and instructions in else branch.

   Use at most 12 instructions.

```
for (i = 0; i < a; i += 1)
    if ((i & 0xA5) != 0)
        r ^= i << 8;
    else
        r += i >> 4;
```

5. Read the following Wikipedia page about Collatz conjecture. The link is clickable. https://en.wikipedia.org/wiki/Collatz_conjecture.

   Write a RISC-V program in RARS. The program reads a positive integer *n* and prints out the total stopping time of *n*, i.e., the number of times we need to apply function *f* on *n* to reach 1. Function *f* is defined on the Wikipedia page. For example, if *n* is 1, the program outputs 0. If *n* is 9, the program outputs 19. There is no newline character after the number. You can find more expected results on the Wikipedia page.

   Note that we cannot use MUL and DIV instructions, which are not in RV32I.