JOINT INSTITUTE
交大密西根学院

ECE3700J Introduction to Computer Organization

Homework 1

Assigned: September 22, 2022

Due: 2:00pm on September 29, 2022

Submit a PDF file on Canvas

1. (5 points) For the following C statement, write the corresponding RISC-V assembly code. Assume that the C variables z, x, and y, have already been placed in registers x16, x17, and x18 respectively. Use a minimal number of RISC-V assembly instructions.

```
z = x + (y - 12);
```

2. (5 points) For the following C statement, write the corresponding RISC-V assembly code. Assume that the variables i and j are assigned to registers x5 and x6 respectively. Assume that the base addresses of the arrays A and B are in registers x16 and x17, respectively.

```
B[7] = A[i+j];
```

3. (10 points) For the RISC-V assembly instructions below, what is the corresponding C statement? Assume that the variables f, a, b, c, and d are assigned to registers x5, x6, x7, x28, and x29, respectively. Assume that the base address of the arrays Y and Z are in registers x10 and x11, respectively.

```
slli x30, x5, 2
add  x30, x10, x30
slli x31, x6, 2
add  x31, x11, x31
lw   x5, 0(x30)
addi x12, x30, 4
lw   x30, 0(x12)
add  x30, x30, x5
sw   x30, 0(x31)
```

4. (5 points) Show how the value `0x5678abcd` would be arranged in memory of a little-endian and a big-endian machine. Assume the data are stored starting at word address 0.

5. (10 points) Find the shortest sequence of RISC-V instructions that extracts bits 20 down to 11 from register x15 and uses the value of this field to replace bits 31 down to 22 in register x16 without changing the other bits of registers x15 or x16. (Be sure to test your code using x15 = 0 and x16 = `0xffffffff`. Doing so may reveal a common oversight.)

6. (5 points) Assume x5 holds the value 0x11010000. What is the value of x6 after the following instructions?

```
        addi x6, x0, 1
        bge x5, x0, ELSE
        jal x0, DONE
ELSE:   ori x6, x0, 2
DONE:   lui x6, 0xFFFFF
```

7. Consider the following loop in RISC-V assembly:

```
LOOP:   beq x6, x0, DONE
        addi x6, x6, -1
        addi x5, x5, 2
        jal x0, LOOP
DONE:   ......
```

   (1) (10 points) Assume that the register x6 is initialized to the value 5. What is the final value in register x5 assuming the x5 is initially zero?

   (2) (10 points) For the loop above, write the equivalent C code. Assume that the registers x5 and x6 are integers i and j, respectively.

   (3) (5 points) For the loop written in RISC-V assembly above, assume that the register x6 is initialized to the value N. How many RISC-V instructions are executed?

   (4) (5 points) For the loop written in RISC-V assembly above, replace the instruction "`beq x6, x0, DONE`" with the instruction "`blt x6, x0, DONE`" and write the equivalent C code.

8. (20 points) Translate function `f` into RISC-V assembly language following function calling convensions. Assume the function declaration for `g` is `int g(int a, int b)`. The code for function f is as follows:

```
int f(int a, int b, int c, int d){
   return g(g(a,c), b-d);
}
```

9. (10 points) Right before your function `f` from Problem 8 returns, what do we know about contents of registers `x10-x14, x8, x1`, and `sp`? Keep in mind that we know what the entire function `f` looks like, but for function `g` we only know its declaration.

叶子木 520370910037

1. add x16, x17, x18
   addi x16, x16, -12

2. add x5, x5, x6
   slli x7, x5, 2
   add x7, x16, x7
   lw x9, 0(x7)
   sw x9, 28(x17)

3. z[a] = Y[f] + Y[f+1]

4.

Address:	0	1	2	3
Big Endian:	56	78	ab	cd
Little Endian:	cd	ab	78	56

5. add x5, x15, x0
   srli x5, x5, 14
   slli x5, x5, 22
   slli x16, x16, 10
   srli x16, x16, 10
   add x16, x8, x16

6. x6 = 0x 00000001

x5 ≥ x0 => ELSE

x6 = 0x 0000 0010

x6 = 0x FFFFFF00

---

7 (1) loop1: x6 = 4, x5 = 2

loop2: x6 = 3, x5 = 4

3:	2	6
4	1	8
5	0	10

final value of x5 is 10

(2) while (j != 0) {

j -= 1 ;

i += 2 ;

}

(3) 4N + 1

(4) while (j >= 0) { j -= 1; i += 2; }

8. arguments: f(a, b, c, d) in x12, x13, x14, x15

g(a, b) in x16, x17

results f in x10, g in x11

f:

addi sp, sp, -4

sw x1, 0(sp)

add x12, x16, x0

```
add  x13, x17, x0
jal  x1, g
add  x10, x16, x0
sub  x17, x13, x15
jal  x1, g
lw   x1, 0(sp)
addi sp, sp, 4
jalr x0, 0(x1)
```

9. x10: g(g(a, c), b-d)

x11 is not used. so is default value

x12: value of a, x13: value of b, x14: value of c

x1: the f's caller's address + 4

sp: original value before calling f

x8: same as sp