Keeyan Haghshenas → [keeyanhaghshenas9@gmail.com](mailto:keeyanhaghshenas9@gmail.com)

Thomas Wray → tpw31@scarletmail.rutgers.edu

Homework 2:

Problem 1: (12 pts, 3 pts each subpart)

Compile the RISC-V assembly code for the following C code.

LOOP: beq x6, x0, DONE

addi x6, x6, -1

addi x5, x5, 2

jal x0, LOOP

DONE:

Part1: Assume that the register x6 is initialized to the value 10. What is the final value in register x5 assuming the x5 is initially zero?

X6 = 10, x5 =0

When the loop begins we say that x6 is equal to 10 which is not equal to zero so we aren’t done. Then we store 9 in the x6 register. Then 2 in the x5 register. Then we jump to the next loop.

When the loop begins we say that x6 is equal to 9 which is not equal to zero so we aren’t done. Then we store 8 in the x6 register. Then 4 in the x5 register. Then we jump to the next loop.

When the loop begins we say that x6 is equal to 9 which is not equal to zero so we aren’t done. Then we store 8 in the x6 register. Then 4 in the x5 register. Then we jump to the next loop.

The loop will iterate all the way til we get to x6 is equal to 1.

On the final iteration of the loop x6 is equal to one. Initially so the loop can commence once again. As the code commences x6 is added to negative one again, making it zero, so on the next run the loop we will jump to done. The amount of iteration has caused x5 to equal 20.

Therefore x5 is equal to 20.

Part 2: For the loop above, write the equivalent C code. Assume that the registers x5 and x6 are integers acc and i, respectively.

for(i=10,i>=0;i--)

acc = acc +2;

Part 3: 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?

Two instances can occur. If the value N is initialized to zero then the code will jump right to done, where two instructions have been executed. The second instance is when its initialized to anything but zero, so the lines to initialize the variable N and run the code would end in 6 instructions being executed.

Part 4: 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.

int acc;

for (int i =10, i<=0; i—){

if(i<0){

goto done

else

acc += 2

}

done: exit

}

Problem 2:

Srli x7,x5,11

Srli x7,x7,26

Slli x6,x6,6

Slli x6,x6,6

Add x6,x7,6

3:

Part 0:

The jal instruction yields a PC+8 counter. Therefore a 20 bit intermediate in the J format the range would be +- 2^17 bits.

The j instruction would be the same at the first part, 2^17

The Beq instruction have 12 bit imediates so +- 2^10 bits because the PC is 2^2

Part 1:

No you cannot. A jump is only allowed to go to PC + 4, which isnt enough to jump from the 0x00004000 to PC=0x20014924.

Part 2:

No we can because, branch lets us use PC + 4 + H where H is a signed 32 bit. Which still does not allow us the range to make the jump.

Part 3:

Yes there are 2^12 bits available for the program counter

Beq x0,x0,1682

4:

Func:

Slt x0, x1, x3

Addi x0, 0, 4

Beq x0, zero, elseif

Jr, ra

elseif:

Beq x0,0, else

Addi, x0, 0, 8

Jr ra

else

Add,x0,x1,x3

Jr ra

5:

Part 2:

The number of instructions depends on the value that is input for n.

Depending on which function is run the output could either be 9 instructions or 10 instructions.

7:

Part 2:

There is no possible way for us to know those registers. The function can change the registers and the location of the data stored in the registers therefore there is no way to tell.

Thomas Wray - Problems 1-8

Keeyan Haghshenas - Problems 1-8