
CSCI-21 Lab 2, due 2/12/20

Please work alone on the paper parts of this first, then work in groups of two for the test code. Agree on your paper answers **before** you write the code to test these.

Bit operation instructions:

or ori and andi xor xori sll srl sra nor (there is NO nori!)

Two useful macros:

not \$Rd, \$Rs is implemented as nor \$Rd, \$Rs, \$0, a macro (or pseudoinstruction).

nop (no-operation, which has no operands) is a macro of either or \$0, \$0, \$0 or even more simply, sll \$0, \$0, 0 which is actually 0x00000000 in MIPS machine code. And this is, in fact, a very useful instruction!

The immediate bitwise logical instructions left-zero-fill the 16-bit immediate field to 32 bits.

1. For the following hexadecimal bit patterns, determine the result of executing or, and, xor and nor on the patterns. After trying by hand, write a little MIPS program to check your results.

0x1234 and 0xFFFF

0xFEDC and 0x5678

For these, you may load the 16-bit values using ori with \$zero to get them into registers to test.

2. Verify, using a truth table, that the result of NORing a value with 0 is the same as NOTing (bitwise complementing) the value. Then extend the program you wrote in part 1 to test this with a few values.
3. Given the following bit patterns, try to predict the results of executing sll, srl and sra on them to move the patterns left or right 2, 3, 4 and 5 places. You can use the li (load immediate) macro for 16- or 32-bit values (li is a macro that expand into one or two instructions, depending on the size of the macro's immediate field). If you set Accept Pseudo Instructions ON in QtSpim, this will work. Check your predictions by extending the program again.

0xffffffff

0x96969696

0x87654321

I don't care to see your paper work so long as you agree on the answers, but e-mail me the program you write. Make sure all your names are in the notes at the top of the program and in the subject line/body of the email. Please, remember the rules for email...