CS103 – Monsoon 2018 — Homework 3

Neeraj Pandey

Collaborators: NONE

1. Assume the following register contents:
$t0 = 0x89ABCDEF, $t1 = 0x12345678
For the register values shown above, what is the value of $t2 for the following sequence of instructions? There are come instructions whose functionality you might have to look up in the textbook, or on the Internet. Show your work.

(a) srl $t2,t0, 3
andi $t2,t2, 0xFFEF

Solution: Binary value of \$t0 = 1000100110101011111100110111101111
According to the first instruction, \$t0 is shifted by 3bits:

$$\$t2 = 00010001001101010111100110111101$$

Binary value of 0xFFEF is 1111111111101111
According to the second instruction,

00010001001101010111100110111101 **AND** 00000000000000001111111111101111

$$= 00000000000000001111100110101101$$

$$\implies \$t2 = 0x000079AD$$

(b) sll $t2, $t0, 10
or $t2, $t2, $t1

Solution: Binary value of \$t0 = 10001001101010111110011011111101111
According to the first instruction, \$t0 is shifted by 10bits:

$$\$t2 = 10101111001101111011110000000000$$

According to the second instruction:

10101111001101111011110000000000 **OR** 00010010001101000101011001111000

$$= 10111111001101111111111001111000$$

$$\implies \$t2 = 0xBF37FE78$$

1

2. instructions to save/restore values on the stack and update the stack pointer. Assume that procA and procB were written independently by two different programmers who are following the MIPS guidelines for caller-saved and callee-saved registers. In other words, the two programmers agree on the input arguments and return value of procB, but they can't see the code written by the other person. Be sure to read the textbook and lecture slides so you understand the MIPS guidelines for caller-saved and callee-saved registers.

Solution: procA:
$s0 = ...
$t1 = ...
$s1 = ...
$t2 = ...
$s2 = ...
$t0 = ...
L
$a1 = ..
$a0 = ...
addi $sp, $sp, -12
sw $ra, 8($sp)
sw $a0, 4($sp)
sw $a1, 0($sp)
jal procB
lw $ra, 8($sp)
lw $a0, 4($sp)
lw $a1, 0($sp)
addi $sp, $sp, 12
M
... =$s1
... =$t0
... =$t1
... =$a0
... =$s3
jr $ra

procB:
U
addi $sp, $sp, -8
sw $s2, 4($sp)
sw $s3, 0($sp)
... = $a0
... = $a1
$s2 = ...

$s3 = ...
$t0 = ...
lw $s2, 4($sp)
lw $s3, 0($sp)
addi $sp, $sp, 8

X
jr $ra