

Name: \_\_\_\_\_

# CIS 351 Practice Test 2

Updated October 29, 2018

Test 2 is Tuesday, 6 November. Problems 18, 19, and 20 are due for credit Friday, 2 November.

1. Show how to build a register file using only AND, OR, NOT, NOR, NAND, and XOR gates.
2. What is a computer's "word size"?
3. List and explain the direct and indirect effects of a computer's word size (16-bit, 32-bit, 64-bit, etc.).
4. How does a computer's word size affect its design?
5. How do "fixed-width" and "variable width" CPUs differ? What are the advantages and disadvantages of each?
6. Explain the difference between CISC and RISC computers. What are the advantages and disadvantages of each?
7. Explain how MIPS applies each of the four design principles discussed in Chapter 6.
8. What are implicit parameters? Name some instructions that have implicit parameters.
9. Why does the MIPS instruction set not contain a subtract immediate instruction?
10. Look at each MIPS pseudo-instruction and explain why it is a pseudo-instruction instead of a "real" instruction.
11. A colleague notices that the jump instruction `j label` can be replaced by a pseudo-instruction `beq R0, R0, label`. He then proposes eliminating the MIPS jump instruction `j` (the compiler would then replace any `j` instructions with the corresponding `beq` instruction.)
  - (a) Which design principles suggest that you should keep the `j` instruction? Why?
  - (b) Which design principles suggest that you should eliminate the `j` instructions? Why?
12. Give examples of how the MIPS instruction set exhibits each of the four design principles.
13. Why is there not a `sla` instruction?
14. What is the difference between `slti` and `sltiu`?
15. Explain how `add` and `addu` behave differently.
16. Convert the following line of Java code to assembly: `t0 = t1 + t2 + t3 - t4 + t5`

Name: \_\_\_\_\_

17. Convert the following line of Java code to assembly:  $t0 = (t1 \wedge t2) \& (t3 \mid !t4)$
18. Convert the following Java code to assembly. Your answer *must* use `slt`. Do not use any pseudoinstructions.
- ```
if (t1 - 6 < t2) {  
    t0 = t1;  
} else {  
    t0 = t2 + 4;  
}  
t1 = t1 + 7
```
19. Describe in common English what the following function does. Hint: It takes three parameters, all integers.

```
mysteryFunction1:  
slt $t0, $a0, $a1  
slt $t1, $a1, $a2  
and $v0, $t0, $t1  
jr $ra
```

20. Describe in common English what the following function does. Hint: It takes two integer parameters. `sra` stands for “shift right arithmetic”. It moves all the bits in the register to the right the specified amount.

```
mysteryFunction2:  
add $v0, $a0, $a1  
sra $v0, $v0, 1  
jr $ra
```

21. Convert the following Java code to assembly:

```
t1 = 0;  
for (int t0 = a0; t0 >= 0; t0-- = a1) {  
    t1 += t0;  
}  
return t1;
```

22. Review the assembly programs from each of the assembly labs.
23. Review the Single Cycle homework.