

The UNIVERSITY of NORTH CAROLINA at CHAPEL HILL

Comp 411 Computer Organization

Fall 2012

Prof. Montek Singh

Exam #1: SAMPLE QUESTIONS — Solutions

(The actual exam will have 10-15 questions.)

1. (2 points) How many bits does it take to encode a 9-digit social security number? Assume all possible combinations of digits are legal. Fractional bits should be rounded up. (If you do not have a calculator, you can give the full expression instead of the final answer.)

$$\lceil \log_2(10^9) \rceil = \lceil 29.89 \rceil = 30 \text{ bits}$$

2. (4 points) Using 6-bit 2's-complement binary arithmetic, add the following two numbers: $(-9_{10}) + (-10_{10})$. Fill in the entries in the table below to show: (i) the 2's complement representation of each number, (ii) the result in binary of adding them together, and (iii) convert the binary sum back to decimal, showing both its sign and magnitude.

<i>Decimal</i>	<i>Action</i>	<i>6-bit Binary</i>
-9	Convert to binary →	110111
-10	Convert to binary →	110110
	Sum in binary →	101101
-19	← Convert to decimal	

3. (1 point) Which field determines the operation of an R-type instruction?

The FUNC field determines the actual operation (the opcode is 0 for all R-type instructions).

4. (2 points) Suppose the program counter, PC, has the value 0x00001234. What is the value of PC after executing the following branch instruction?

beq \$0, \$0, 4

$$\begin{aligned} \text{PC} &\leftarrow \text{PC} + 4 + \text{sign-ext}(4 * \text{imm}) \\ &= 0x1234 + 4 + 16 = 0x1248 \end{aligned}$$

5. (2 points) Without making any assumptions about the contents of registers or memory, which of the following operations **cannot** be performed by a **single** MIPS instruction and why?

- (A) $\text{Memory}[\text{R}[\text{rs}] + 0x1000] \leftarrow 0$
- (B) $\text{Memory}[\text{R}[\text{rs}]] \leftarrow 0$
- (C) $\text{Memory}[0x1000] \leftarrow 0$
- (D) $\text{Memory}[\text{R}[\text{rs}] + \text{R}[\text{rt}]] \leftarrow 0$
- (E) $\text{R}[\text{rt}] \leftarrow \text{Memory}[\text{R}[\text{rs}] + 0x1000]$

The answer is (D) because there is no MIPS instruction that can add two registers and use the sum as a memory address. All other choices can be implemented using a `sw` or `lw` instruction.

6. (3 points) Suppose you execute the following instruction sequence:

```
addi $t0, $0, -1
sll  $t0, $t0, 16
srl  $t1, $t0, 16
sra  $t2, $t0, 16
```

What are the values of `$t0`, `$t1` and `$t2` after execution (in binary or hex)?

```
$t0 = 0xFFFF0000
$t1 = 0x0000FFFF
$t2 = 0xFFFFFFFF
```

7. (1 point) Assuming the standard MIPS procedure calling conventions, if we see an instruction of the form `lw $t0, 4($fp)`, the program is most likely

- (A) accessing the return address
- (B) accessing one of its own local variables
- (C) accessing a local variable belonging to its caller
- (D) accessing its fifth argument
- (E) none of the above

The answer is (D) because the fifth argument (`arg[4]`) would typically be found one word above the location to which the frame pointer is pointing.

For the next two questions, consider the following assembly language procedure:

```
foo:
    addi $sp, $sp, -4
    sw   $ra, 0($sp)
    beq  $a0, $0, L1
    addi $a0, $a0, -1
    add  $a1, $a1, $a1
    jal  foo
    add  $a1, $v0, $0
L1: add  $v0, $a1, $0
    lw   $ra, 0($sp)
    addi $sp, $sp, 4
    jr   $ra
```

Suppose there is a procedure called `main` which calls `foo(4, 3)`. [Assume that `main` places 4 in `$a0`, and 3 in `$a1` before calling `foo`.]

8. (4 points) What is the entire sequence of calls to `foo`, starting with `foo(4, 3)`?

`foo(4, 3) → foo(3, 6) → foo(2, 12) → foo(1, 24) → foo(0, 48)`

9. (1 point) What is the final value returned to `main`?

The value returned to `main` is 48.