

# Homework 1

**Due Date:** Tuesday, April 12, 2005, 2:00

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

Your Email: \_\_\_\_\_

## Reading Assignment

Cooper and Torczon: Chapter 5 and 6

**Question 1** Is SPARC big- or little- Endian? \_\_\_\_\_

When a word of data is stored in memory, which byte is stored in the first byte (i.e., in the byte with the lowest address)? \_\_\_\_\_

**Question 2** A word aligned address must be divisible by what? \_\_\_\_\_

How many of the least significant bits must be zero in the address? \_\_\_\_\_

**Question 3** What is this binary value, in hex?

0101 1100 0011 1011 1010 0010 0100 1001

\_\_\_\_\_

What is this hex value, in binary?

84AF DC3B

\_\_\_\_\_

**Question 4** Using 32-bit unsigned numbers, which is the largest number that can be represented? Give it in hex and in decimal. How do we often write this number (for example, 2K+4).

\_\_\_\_\_ (in hex)

\_\_\_\_\_ (in decimal)

\_\_\_\_\_ (as an expression)

**Question 5** Using 32-bit signed numbers, which is the largest number that can be represented?

\_\_\_\_\_ (in hex)  
 \_\_\_\_\_ (in decimal)  
 \_\_\_\_\_ (as an expression)

**Question 6** Using 32-bit signed numbers, which is the smallest number that can be represented?

\_\_\_\_\_ (in hex)  
 \_\_\_\_\_ (in decimal)  
 \_\_\_\_\_ (as an expression)

**Question 7** Using 32-bit signed numbers, how is -1 represented? Give it in hex.

\_\_\_\_\_

**Question 8** How long is each SPARC instruction (ignore “set”)?

in bits? \_\_\_\_\_

in bytes? \_\_\_\_\_

in words? \_\_\_\_\_

**Question 9** There are four sets of 8 registers in the SPARC. What are they called?

\_\_\_\_\_  
 \_\_\_\_\_  
 \_\_\_\_\_  
 \_\_\_\_\_

**Question 10** In the full SPARC architecture, each register is 64 bits long. In the subset we are using, how many bits are we using in each register? \_\_\_\_\_

**Question 11** Some instructions allow the second operand to be either in a register or specified literally, by including a value directly in the instruction. For example:

```
add    %g4, %g7, %o2
add    %g4, 123, %o2
```

When a literal (immediate) value is used, what is the range of allowable values? (Specify in decimal.) \_\_\_\_\_

How would you describe this value? (Specify signed/unsigned and number of bits.)

\_\_\_\_\_

**Question 12** Consider these SPARC instructions: sll, srl, and sra. What do these letters stand for?

sll: \_\_\_\_\_

srl: \_\_\_\_\_

sra: \_\_\_\_\_

**Question 13** Consider the difference between srl and sra... What does srl do? What does sra do? (Hint: your answer should use words like: shift, zero, one, most-significant, least-significant.)

srl: \_\_\_\_\_

sra: \_\_\_\_\_

**Question 14** When using instructions that alter flow of control (like ble and call), the processor will always execute the instruction in the word following the instruction before the transfer is made. What is this word called?

\_\_\_\_\_

Filling it is an important optimization, but is sometimes difficult. To just get the program working, you can always put what instruction in this position?

\_\_\_\_\_

**Question 15** Here is code with an annulled branch:

```

cmp      %o7, 100
bge, a   myLoop
add      %o5, 1, %o5

```

If the value of register %o7 is greater than or equal to 100, will %o5 be incremented?

\_\_\_\_\_

If the value of register %o7 is less than 100, will %o5 be incremented?

\_\_\_\_\_

**Question 16** Show in hex the bytes that this pseudo-op will place in memory?

```
.asciz   "abc"
```

\_\_\_\_\_

**Question 17** The instruction "not" is a synthetic instruction. Consider

```
not      %o4, %o5
```

The assembler will produce the same code for this instruction as for what instruction?

\_\_\_\_\_

**Question 18** Which synthetic instruction is used to move a full 32 bit value into a register? Give an example moving 0x12345678 into %g5.

\_\_\_\_\_

Also show the two instructions it will assemble into.

\_\_\_\_\_

\_\_\_\_\_

**Question 19** Consider this SPARC instruction:

```
call printf
```

This instruction saves the address of what instruction?

\_\_\_\_\_

It puts it in which register before transferring control to the routine called “printf”?

\_\_\_\_\_

**Question 20** Which instruction opens a new register window? \_\_\_\_\_

Which closes it? \_\_\_\_\_

**Question 21** Assume that a routine named “foo” calls a routine named “bar” using the standard SPARC calling conventions. Where will the code for “foo” place the first argument to be passed? \_\_\_\_\_

In the code for “bar”, where will that argument be found? \_\_\_\_\_

Assume “bar” returns a value; where will “bar” put the returned value? \_\_\_\_\_

Where will “foo” look to find the result? \_\_\_\_\_

**Question 22** There are two stacks involved in the SPARC procedure calling. One concerns the register windows and the other is the in-memory stack of “frames”. Does the in-memory stack grow from lower addresses toward higher, or from higher addresses toward low memory? \_\_\_\_\_

**Question 23** What is the other name for “stack frames”?

\_\_\_\_\_

**Question 24** Each stack frame in the SPARC must be a multiple of how many bytes?

\_\_\_\_\_

**Question 25** Assume that a routine named “foo” calls a routine named “bar” using the standard SPARC calling conventions. Within “bar”, what does %fp point to? What does %sp point to?

%fp points to \_\_\_\_\_

%sp points to \_\_\_\_\_

**Question 26** When returning from a routine, what instruction is normally placed after the “ret” instruction? \_\_\_\_\_

**Question 27** Assume that a routine named “foo” calls a routine named “bar” using the standard “C” calling conventions. Where will the code for “bar” find any additional arguments (beyond the first 6)? \_\_\_\_\_

**Question 28** Some small routines (called “leaf” routines) may execute especially efficiently by not allocating a new frame on the in-memory stack. Which registers must such a routine leave unchanged when it returns?

\_\_\_\_\_

\_\_\_\_\_

**Question 29** Consider the “gdb” debugger. Which command do you use to... (Just show the command name, not any arguments.)

Set a breakpoint? \_\_\_\_\_

Continue execution after a breakpoint is reached? \_\_\_\_\_

Begin execution of a program? \_\_\_\_\_

Display a data value? \_\_\_\_\_

Examine memory contents? \_\_\_\_\_

Change a register? \_\_\_\_\_

Display a routine’s instructions? \_\_\_\_\_

Execute the next (machine) instruction, skipping over any called subroutines?

\_\_\_\_\_

Execute the next (machine) instruction, including all instructions in called routines? \_\_\_\_\_

Execute the next (source language) statement, skipping over any called subroutines? \_\_\_\_\_

Execute the next (source language) statement, including all instructions in called routines? \_\_\_\_\_

Display the contents of the routine calling stack? \_\_\_\_\_

Display the contents of all registers (show the parameter)? \_\_\_\_\_

**Question 30** How many bits in a single precision floating point number? \_\_\_\_\_

In a double precision floating point number? \_\_\_\_\_

**Question 31** What is the second to the last double precision floating point register named?

\_\_\_\_\_

**Question 32** What SPARC instruction will move a quad precision value from memory to a floating point register? \_\_\_\_\_

What instruction will move a single precision value from a floating point register to memory? \_\_\_\_\_

**Question 33** What is the value (expressed as a decimal number) of this fixed-point binary number: 1100.1011? \_\_\_\_\_

**Question 34** Can every decimal fraction be represented exactly using a binary number (with finite precision)? \_\_\_\_\_

**Question 35** What does NAN stand for? \_\_\_\_\_

**Question 36** What are 3 values that can be represented with a floating point number that are not themselves numbers? (Given symbolic names, not actual hex values.)

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

**Question 37** About how many decimal digits of accuracy are available in a double precision floating point number? \_\_\_\_\_

What is the range of the exponent, in decimal? \_\_\_\_\_

**Question 38** What are the SPARC instructions for...

subtracting two single precision floats? \_\_\_\_\_

multiplying two quad precision floats? \_\_\_\_\_

negating a double precision float? \_\_\_\_\_

moving a double precision float from one register to another? \_\_\_\_\_

computing the square root of a double precision float? \_\_\_\_\_

comparing two double precision floats? \_\_\_\_\_

**Question 39** Are the condition codes that the fcmpd instruction modifies the same as, or different from, the condition codes modified by the cmp instruction?

\_\_\_\_\_

**Question 40** Do instructions like fadds and fmuld set the condition code bits?

\_\_\_\_\_

**Question 41** When converting from a double-precision floating point value to an integer value, the source must be in a floating point register. Which instruction is used?

\_\_\_\_\_

Where is the result placed? \_\_\_\_\_

**Question 42** To “trap” to the OS (i.e., to invoke an operating system function), which instruction is used? \_\_\_\_\_