CS 2733/2731, Computer Organization II
Spring Semester, 2004
First Examination

1. Below are questions about number representations and conversions: **(25)**

   (a) Convert the (decimal) number −46 to 16-bit two's complement binary. (The binary representation for 46 is 101110.)

   (b) Consider the floating point number (a `double`) with representations:

   ```
   1011 1111 1101 1100  (48 more 0's)  (binary)
   b    f    d    c     (12 more 0's)  (hex)
   ```

   i. What is the sign of this number?

   ii. What is its exponent (power of 2)? (Remember that the bias for a `double` is 1023, and that an exponent of 1 is represented by `100 0000 0000`.)

   iii. What is the significant part?

   iv. Put i, ii, and iii together to get the number.

---

2. Consider the following MIPS code fragment: **(25)**

   ```
         .data
   A:    .word 2, 3, 5, 7, 11, 13, 17, 19, 23, 0
         .text
   # insert MIPS instructions here.
   ```

   For insertion at the comment, write MIPS instructions that will do the following:

   (a) Load the address of `A` into `$s0`.

   (b) Use a loop to add the 10 values of the array `A`.
   (You must access the array `A` and you must use a loop for credit. You may use the fact that the last number is a zero if you wish. Your MIPS code should do what is asked for above and *nothing more*.)

---

3. Write a *single* MIPS function **`Addup`** that does a., b., and c. below. **(25)**

   (a) **`Addup`** saves register **`$ra`** *on the stack*.

   (b) **`Addup`** adds its two input parameters and returns the sum.

   (c) **`Addup`** restores the register **`$ra`** saved above and returns.

   (d) Separately show a call to **`Addup`** with input parameters 7 and 19.
   Note: You should just give code for the call to **`Addup`** and for the definition of the function **`Addup`** that do the above items and *nothing more*. You should follow MIPS parameter convensions.

(10)

4. The MIPS assembler instruction `lui` (Load Upper Immediate) will load an immediate constant into the upper 16 bits of a register. Thus `lui    $s3, 0x344` will load the hexadecimal number `0x344` (which is 836 decimal) into the upper 16 bits of register `$s3`, leaving 0s in the lower 16 bits. Show how to use two instructions to load the hexadecimal constant `0x034407ff` into the register `$s3`.

(15)

5. Consider the following logic gate constructed out of CMOS transistors.

   (a) In case A is a 1 (voltage high) and B is also a 1, what will be the output at C? Explain your answer in terms of the diagram and the properties of the transistors. (Show which switches are open (don't conduct current), which are closed (conduct current), and explain why the output at C is what it is.)

   (b) Say what kind of logic gate this represents.