Name: ____________________

Directions: Work on this sheet (both sides, if needed) only; **do not turn in any supplementary sheets of paper**. There is actually plenty of room for your answers, as long as you organize yourself BEFORE starting writing. In order to get full credit, **WRITE LEGIBLY** (∞ points off for illegible handwriting!), and **SHOW YOUR WORK**. Make sure you work on the easier problems first (which will usually be the earlier problems, and within multi-part problems, the earlier parts).

1. (15 pts) Show how to construct an XOR gate from AND gates, OR gates and NOT gates. The two input lines will be labeled I1 and I0, and the output line labeled X. X should be equal to 1 if and only if exactly one of I1 and I0 is equal to 1.

2. Look at the computation yielding 23 clock cycles near the top of p.475 of Patterson & Hennessy.

(a) (10 pts) Suppose the time to send something along the data bus is two cycles instead of one. What will the 23 change to?

(b) (10 pts) Suppose we have the change in (a), <u>and</u> that the block size is eight instead of four. What will the 23 change to?

3. (15 pts) In Figure 2.3, p. 41 of the PC book, show the wiring between the registers CS, IP and MAR, and a 20-bit adder. In your diagram, be sure to label the bit numbers, e.g. 19 (msb) to 0 (lsb) in MAR, so that it is clear which bits get attached where.

4. (10 pts) Look at Fig. 7.8, p.468 of Patterson & Hennessy. For the program gcc, what percentage of memory accesses consisted of instruction fetches?

5. Suppose we have a machine (unrealistically simple) with 8-bit addresses, 1-byte words, 1-word block size and a 4-line cache (direct-mapped). Suppose we are running a program compiled from code which includes the following:

```
int x,y,z[4],*p;

x = 2;
y = 9;
z[2] = 3;
y += x;
p = &z[2];
x = *(p-3);
```

Suppose the compiler assigns storage for the variables contiguously and in the order of declaration, with x in word 0, y in word 1, and so on, and that the cache is initially empty.

(a) (10 pts) How many bits will be needed for each tag entry?

(b) (10 pts) Which of the six executable C statements above will result in <u>read</u> (not write) misses?

(c) (10 pts) Show the numerical contents of line 0 (valid bit, tag bits and data bits) after the program has finished execution.

6. (10 pts) Re-read the Elaboration on p.465, Patterson & Hennessy. Recall that in our class discussion of this, we noted that it could be implemented by adding a Busy bit to each register. When the execution of an instruction causes a data cache miss, the Busy bit for the destination register (if any) is set to 1 until the cache miss completes processing. During the processing of the cache miss, subsequent instructions are checked for executability by checking whether any of their source registers have their Busy bit set. Set up circuitry to do this. Assume that the current instruction is in the instruction register (IR), consisting of 11 bits: a 2-bit op code, and three 3-bit register codes. The three register codes specify the destination and two source registers for the instruction. Your circuit's inputs should be IR10-IR0, the bits of IR, and B7-B0, the Busy bits of the machine's eight registers, and the output should be one bit labeled OK, with OK = 1 meaning that we can go ahead and execute this instruction. Feel free to use standard MSI components such as MUXs, etc.