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. The earlier problems, and the earlier parts within each problem, are intended to be easier—MAKE SURE YOU WORK ON THEM FIRST!**

1. (10) Suppose we have a large digital circuit, part of which consists of a 4-bit register, encoding 16 different cases. We will have 16 different wires leading to other parts of the circuit, dealing with the 16 cases. What kind of component should we place in between the register and the 16 wires?

2. (10) Fill in the blank: Although Patterson and Hennessy refer to the fundamental storage cells in a cache as **blocks**, in industry they are more commonly called ______________.

3. (10) Look at the circle containing an equal sign on p.553 of Patterson and Hennessy. Its left input is the upper 16 bits of the address generated by the ALU, which we will denote by u31, u30, ..., u16. Its right input is the tag from the cache entry, denoted by t15, t14, ..., t0. The output, which we will call e, consists of a single wire whose value will be 1 in the case of a hit and 0 in the case of a miss. Write a "sum of products" equation for e as a function of the ui and tj. Feel free to use an ellipsis ("...") as long as your meaning is clear. (Note: Write an *equation*, not a table. The second expression for E on p.B-10 is an example of a sum-of-products equation.)

4. Look at the set of pictures on p.548, and let x denote the next memory address to be presented by the CPU to the cache after picture (f).

(a) (10) Give a complete list of values of x which would result in hits.

(b) (10) Give a complete list of values of x which would result in evictions of items currently in the cache.

5. (5) Fill in the blank with either the word *wide* or *narrow*: Programs having a high degree of spatial locality benefit from caches with ______________ blocks.

6. Supppose the following: word size is 16 bits; address size is 14 bits; consecutive word addresses for our memory system differ by 1 (i.e. individual bytes within a word are not separately addressable); our compiler stores all variables in "non-reverse" order as discussed in our handout; the entire set of declarations in our program is

```
int I,Sum,X[5000];
```

with &I being 0; our main memory uses SRAMs which have 8K words of one byte each. This last point implies that our memory system will consist of pairs of SRAMs; denote the pair contains word 0 of our system by pair 0, then pair 1, etc.

(a) (10) How many pairs of SRAMs will we need?

(b) (10) If our system uses high-order interleaving, in which pair of SRAMs will X[11] be stored?

(c) (10) If our system uses low-order interleaving, in which pair of SRAMs will X[11] be stored?

In parts (d)-(f), assume also that: our cache is direct-mapped; block size is a single word; and there are 64 blocks in the cache. Also for (d)-(f), consider the code

```
Sum = 0;
for (I = const1; I < 500; I += const2)
   Sum += X[I];
```

where const1 and const2 are literal constants.

(d) (5) Suppose const1 is 0. Give a list of all values of const2 which would result in at least one cache eviction per iteration of the loop.

(e) (5) Give an example of values of const1 and const2 which result in two cache misses per iteration of the loop.

(f) (5) Suppose const1 is 63 and const2 is 64. What change could we make to the declarations above (NOT to the code) which would improve our cache hit rate? **Explain fully.**