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. (20) Draw lines from each item in the left column, which is discussed in the context of caches, to the analogous item in the right column, which is for the virtual memory context:

miss	virtual page number
main memory	disk
block	page offset
index	page fault
	address translation
	page
	physical page number

2. (20) Look at the table at the bottom of p.547. Suppose we have the same sequence of memory references (leftmost column) on the same machine, except that 2-way set associativity is used, i.e. we have sets of size 2. The cache still has 8 one-word lines, same as before. Suppose that if both lines in a set are unused when a new block is brought in, the new block is always placed in the lower-numbered (i.e. left-side) line. Assume an LRU replacement policy. Show how the last two columns of the table would change, if any. (Your last column will still consist of numbers in the range 0-7.)

3. (15) Look at the MUX in the picture on p.574. Label the four top inputs a31,a30,...,a1,a0; b31,b30,...,b1,b0; c31,c30,...,c1,c0; and d31,d30,...,d1,d0, from left to right. Label the four left-side inputs e0,e1,e2,e3, from top to bottom. Label the output of the MUX o31,o30,...,o1,o0. Give a boolean expression (sum-of-product form or otherwise) for o31 in terms of all the inputs.

4. Suppose our system memory is to consist of 8M 8-bit words, and we use 4Mx1 DRAM chips, high-order interleaved.

(a) (15) How many chips will we need?

(b) (10) We will have a 23-bit address bus, with lines A22,A21,...,A1,A0. A22 will be used for the interleaving. We will also have a 1-of-2 MUX, with two sets of 11-bit inputs, an 11-bit output, and a 1-bit control bit. One set of inputs will be connected to A21-A11 and the other to A10-A0. (The control bit will have to be connected to some other logic, which we will not worry about here.) The output of the MUX will lead to the DRAMs. Explain why we need a MUX.

5. (10) The Unix **time** command will tell us how long a program takes, and some other information about the program's performance as well. For example

```
time gcc x.c
```

will produce output like

```
0.120u 0.130s 0:00.27 92.5% 0+0k 0+0io 438pf+0w
```

Here the operating system tells us that **gcc** took 0.120 seconds of CPU time. It gives some other information too, including a report that the program had 438 page faults. However, it does not report the number of cache misses, and in fact *could not do so*—why not? Explain fully.

6. (10) Suppose we have a virtual memory machine, but with no cache (and no TLB; don't worry if you don't know what a TLB is here). The machine word size is 32 bits. Assume we use SRAMs for the memory. Assume the setup in the figure on p.584. Each line of the page table is stored in a separate word. Suppose the CPU issues a read from virtual address 0x00005032. Suppose further that the corresponding page is currently resident in memory, and that the physical address of the requested item is 0x00012032. The content of the page table register is 0x16800000. Pretend you are a microscopic "gnome," watching activity on the system address and data buses. State which values—and in which order—will go onto the buses. Use hex ("0x") to express the numbers.