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.**

1. (10) Cite a reason why virtual memory is useful even if memory is cheap and plentiful.

2. For each of the following items, answer either Yes or No to the question as to whether the item is in memory or not:

(a) (5) Page table.

(b) (5) Page table register.

(c) (5) TLB.

(d) (5) Process table.

3. (5) In a machine without hardware Reference bits, the OS typically sets things up so that the role of Reference bits is played by _____.

4. Since cache operation is purely in hardware, the OS designer cannot take any measures to improve cache performance. By contrast, in the case of virtual memory, the OS designer has control over many aspects of the system. For each of the following items, answer either Yes or No to the question as to whether the OS designer has control over that item:

(a) (5) Page-replacement policy.

(b) (5) Page size.

(c) (5) Degree of associativity of the TLB.

(d) (5) Location of page images on disk.

(e) (5) Location of a process' page table in memory.

5. Consider a system with virtual memory, including a 2-way set-associative TLB. Assume: 16-bit addresses; 8-bit words; the TLB has 128 entries (i.e. 128 sets); the page size is 256 bytes; the disk contains 256 sectors; main memory consists of 4 pages; a pure LRU page-replacement policy is used.

(a) (5) Consider the C code

```
int x[5000];
...
for (i = 0; i < 10000; i++) y += x[100*i];
```

Suppose the only memory requests are the reads of x. (Say i and y are in registers; we have an instruction cache but no data cache; there are no cache misses during the execution of this loop.) The page offset of x[0] is 0.

Which value of i will produce the second page fault?

(b) (5) Fill in the blank with a number: In examining an entry in the TLB for a match to a given virtual page number, the hardware will check _____ places within the entry.

(c) (10) What will be the size in bits of a tag field in a TLB entry?

(d) (5) What will be the total size in bits of a TLB entry, excluding Valid, Dirty, Reference and Protection bits?

6. Consider the timing analysis at the top of p.249.

(a) (5) A standard AND gate has a **fan-in** (number of input signals) value of 2, but AND gates do exist with larger fan-in values, as is true for OR gates as well. We could implement the circuitry here using various fan-in values.

For example, if we had available AND gates with fan-in values of 3 and 2, we could evaluate the expression P3 P2 P1 P0 c0 in C4, p.247, as follows: Input P3, P2 and P1 into a 3-input AND gate; input P0 and c0 into a 2-input AND gate; and then input the outputs of those two gates into another 2-input AND gate.

Fill in the blanks (with possibly different numbers): The analysis here assumes AND fan-in values as large as _____ and OR fan-in values as large as _____.

(b) (10) In the analysis here, what is the worst-case time for G2 to become valid?

Answers:

1. Protection; sharing; relocation.

2.a. Yes. 2.b. No. 2.c. No. 2.d. Yes.

3. The Valid bits.

4.a. Yes. 4.b. No. 4.c. No. 4.d. Yes. 4.e. Yes.

5.a. Since initially x[0] through x[1023] are in memory, i = 0,1,...,10 don't cause page faults. But i = 11 will cause a fault, resulting in x[1024] through x[1279] being brought into memory. That means i = 12 won't

cause a fault, but i = 13 will; that will be the second fault. (For various complex reasons, I did not grade the second part of this problem, thus reducing its weight to 5; I then added 5 to 5.c.)

5.b. 2 (i.e. the set size).

5.c. The page offset field is 8 bits ($log_2(256)$), so we have 16 - 8 = 8 bits for the virtual page number. Of these bits, 7 ($log_2(128)$) serve as the index. That leaves 1 bit for the tag. (By the way, note that in this case the TLB will always hold the entire page table.)

5.d. 2 (1 + 8) = 18.

6.a.,6.b. At the time you did your regular reading for the course, during your verification of the 5-cycle time claimed by the authors at the top of p.249, you would have discovered the following. First, the authors must be using fan-in larger than 2, because otherwise the time for C4 would be much more than 5. Upon closer inspection (trial and error, trying to match the authors' figure of 5 cycles for C4), one finds that they are always assuming as much fan-in as possible. The p_i and g_i are ready after Cycle 1. Then if we use AND gates with fan-in of 4, the Pi are ready after Cycle 2 and the Gi are ready after Cycle 3. Then using AND and OR gates of fan-in 5, C4 will be ready 2 cycles after the Gi, i.e. after Cycle 5.