

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) Alter equation (9), p.5 of the digital logic handout to show the formula for Z6.

2. Look at the figure on p.19 of the digital logic handout.

(a) (10) Fill in the blank with an official component term from our course: Each line leading from a DCDR to a register is connected specifically to a(n) _____.

(b) (15) Look at the MUX. Let U equal 1 if the upper-right input is selected, 0 if the lower-right one is selected. U is produced as a result of running the top inputs, which we will denote by Op-Code0, OpCode1, OpCode2, OpCode3, N and Z, through some combinational logic. Show this logic by writing U as a boolean expression of Op-Code0 etc.; use sum-of-products form.

3. (10) Equation (8) on p.4 of the digital design handout could be written in the equivalent form $A1(A0D3 + \overline{A0}D2) + \overline{A1}(A0D1 + \overline{A0}D0)$. Fill in the blank with an official acronym from our course: The advantage of the form in equation (8) over this other form is that (8) is easy to implement as a(n) _____.

4. (15) Design a 1-bit ripple counter with asynchronous clear, defined as follows: It has two 1-bit inputs, named PULSE and CLEAR, and a 1-bit output named MOD2. MOD2 is a count of the number of pulses mod 2.

You are allowed to use only a D flip-flop and AND, OR and NOT gates in your design, no other components. The inputs to the DFF will be named D and CLK, and the outputs Q and \overline{Q} .

Give your answer in terms of boolean equations, not as a picture. (Of course, you will probably want to draw a picture on scratch paper to help your thinking.) Specifically, your answer should consist of three boolean equations, whose left-hand sides are "MOD2 = ", "D = ", and "CLK = ", and whose right-hand sides are boolean expressions in PULSE, CLEAR, Q and \overline{Q} .

5. Consider the 32-word memory system and 8-word cache described on pp.546-548 of Patterson and Hen-

nessy, except that block/line size is now 4 words instead of 1. Assume 8-bit words.

(a) (10) How many lines will the cache have?

(b) (10) Suppose the cache is initially empty, and then it is presented with the sequence of memory references 3,4,5,6,3,12. How many misses will occur? Which reference will produce the SECOND miss? (Both read misses and write misses count as misses.)

(c) (10) What is the total number of bits (of all types) stored in this cache? In other words, if we were to build this cache from flip-flops (including packaged sets of flip-flops such as SRAMs), how many flip-flops would be in the cache?

6. (10) Consider the following C code:

```
for (i = 0; i < 10; i++) {
    tmp = 0;
    for (j = 0; j < 3; j++)
        tmp += x[4*i+j];
    x[4*i+3] += tmp;
}
```

where x is an **int** array.

Assume: separate instruction and data caches; no cache misses due to instruction fetches during the execution of the above code; the compiler has stored the variables i, j and tmp in registers rather than in memory; the data cache consists of just 1 line; line size is 4 words; x[0] is the first word in its memory block (i.e. its address is an even multiple of 4 times the word size); the data cache is initially empty. State the numbers of memory accesses which would be made under write-through and write-back schemes. (Both reads and writes count as memory accesses.)

Solutions:

1. $Z6 = X2X1\overline{X0}$.

2.

(a) Tri-state buffer.

(b)

$$U = N(\overline{OC0}OC1OC2\overline{OC3}) + Z(\overline{OC0}OC1OC2OC3) + OC0\overline{OC1}$$

3. PLA.

4. $MOD2 = Q, D = \overline{CLEAR}, CLK = CLEAR + PULSE$.

5.

(a) $8/4 = 2$ lines.

- (b) At the first reference to 3, which is a miss since the cache is empty, the block containing 3 is brought in, which consists of memory locations, 0, 1, 2 and 3. The next reference, 4, is still not in the cache; this is the second miss, and 4's block—4, 5, 6 and 7—is brought in. The next four references—5, 6 and 3—will then be hits. The reference to 12 will then be a miss, the third.
- (c) Since the memory consists of 32 words, the address size is 5 bits. Of these, the least-significant 2 bits give the Word Offset within the block/line, since there are 4 words per block/line. The next least-significant bit is the line number, i.e. the Index (we only need 1 bit for this since there are only 2 lines). The remaining $5 - 2 - 1 = 2$ bits form the Tag. Then there is 1 Valid bit per line, and 4 Data items per line. Thus each line consists of $2 + 1 + 4(8) = 35$ bits, thus a total of 70 for the entire cache.

6. First, consider write-through, when $i = 0$. The read of `x[0]` when $j = 0$ produces a cache miss; this means that the block containing `x[0]`—`x[0]`, `x[1]`, `x[2]` and `x[3]`—must be brought in; this is already 4 memory accesses. When $j = 1$ and $j = 2$, we have two more reads but they are cache hits, thus no more memory accesses. However, the final statement within the “ i ” loop,

```
x[4*i+3] += tmp;
```

generates both a read and a write of `x[3]`. These are cache hits, but the write must be written-through, thus another access to memory. Thus, for each value of i , we will have 5 memory accesses, for a total of 50.

The write-back case is similar. There will still be 4 read memory accesses per value of i . However, the write-back involves writing back the entire block of 4 words, so there will be 4 write memory accesses per value of i . So, with a write-back design, we will have 80 memory accesses in all.