COS 126 Architecture Review Questions

I. Combinational Logic

1. Use AND gates, OR gates, and inverters as needed to implement the following logic expressions as stated:

   (a) x = AB + B'C
   (b) x = A(B + C')
   (c) x = A B' + AB
   (d) x = (ABC)' + B(EF + G')

2. Assume that X consists of 3 bits, $x_2x_1x_0$. Write three logic functions that are true if and only if

   (a) X contains only one 1
   (b) X when interpreted as an unsigned binary number is less than 3
   (c) X when interpreted as a signed (two's complement) number is less than -1

3. Assume that X consists of 2 bits, x_1x_0, and Y consists of 2 bits, y_1y_0. Write logic functions that are true if and only if:

   (a) X < Y, where X and Y are thought of as unsigned binary numbers
   (b) X < Y, where X and Y are thought of as signed (two's complement) numbers
   (c) X = Y

4. One logic function that is used for a variety of purposes (including adders and to compute parity) is exclusive OR. The output of a two-input exclusive-OR function is true if exactly one of the inputs is true. Show the truth table for a two-input exclusive-OR function and implement this function using AND gates, OR gates, and inverters.

5. A NAND gate is functionally equivalent to an AND gate that has an inverter attached to its output. So, NAND(x,y) = 0 when x=1 and y=1; NAND(x,y) = 1 in all other cases. Show that the NAND gate is universal by constructing AND, OR, and NOT functions using a two-input NAND gate.

6. Suppose you have a multiplexer with 8 inputs (and 3 address lines). If you want to use the multiplexer to compute the majority function, treating the 3 address lines as the inputs to the majority function, what values should be assigned to the input lines of the multiplexer? How could you similarly adapt the multiplexer to compute the odd parity function?

7. Suppose that you are given a decoder n to 2^n and you are asked to implement one or more boolean functions with n inputs. What other component(s) do you need?

8. Implement a switching network that has two data inputs (A and B), two data outputs (C and D), and a control input (S). If S equals 1, the network is in pass-through mode, and C should equal A, and D should equal B. If S equals 0, the network is in crossing mode, and C should equal B, and D should equal A.

9. Consider *all* possible boolean functions that take only one input variable (X). There are four of them:

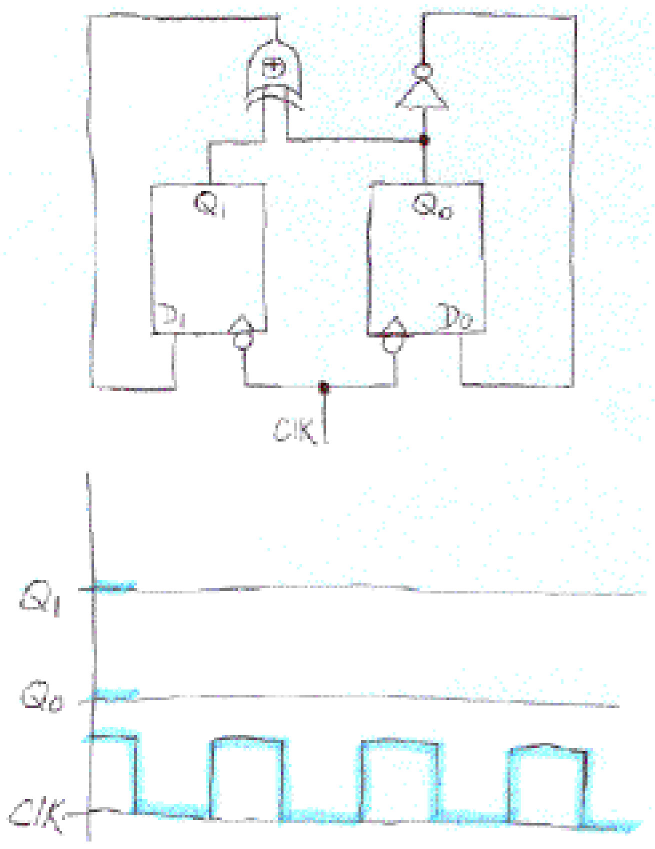
   F0(0) = 0, F0(1) = 0, in other words, F0(X) = 0, the 0-constant function.
   F1(0) = 0, F1(1) = 1, in other words, F1(X) = X, the identity function.
   F2(0) = 1, F2(1) = 0, in other words, F2(X) = X', the inverse function.
   F3(0) = 1, F3(1) = 1, in other words, F3(X) = 1, the 1-constant function.

   Design a circuit that can compute any boolean function of a single variable. This circuit has three inputs (X, f0, f1) and one output (F). The f0f1 inputs encode which one of the above four functions the circuit is computing.

II. Sequential Circuits

10. Build a register file. This register file has only two registers and only one read port, and each register has only 2 bits of data. Draw this register file so that every wire in your diagram corresponds to only 1 bit of data. Draw the registers using D flip-flops. You do not need to show how to implement a D flip-flop or a multiplexor.

11. Two negative edge-triggered D flip-flops are connected as shown in the figure. (The gate near top-center of the figure is an XOR-gate.) Both flip-flops are initialized to 0 at the beginning. Fill in the timing diagram for the two outputs Q1 and Q0.



12. The JK flip-flop is similar to the RS flip-flop except it takes care of the forbidden combination R=S=1. That is, J works as S and K works as R except when J=K=1. In that last case the value of Q^+ will be the inverse of Q: (Recall from lecture that Q denotes the current value stored in the flip-flop, and Q^+ denotes the value the flip-flop will hold at the next clock-tick, given the current values of its inputs (which include Q.)

J	K	Q^+
0	0	Q
0	1	0
1	0	1
1	1	Q'

a) Implement the JK flip-flop using AND, OR, NOT gates (use a clock as well). Do you see any problems with this circuit?

b) Now implement the JK using an RS flip-flop and a few gates. How can you fix the problems above with this (higher level) implementation?

c) Use the JK flip-flop to implement a toggle flip-flop (T). The T flip-flop has input, one output, and the clock, and it toggles its output from 1 to 0 or from 0 to 1 on every clock-cycle during which the input is 1. If the input is 0, the output is preserved.

III. TOY Architecture

13. The complete single cycle TOY Datapath given in lecture (slide 19) does not allow instructions like: 5801 to be performed. Why not? Fix the datapath to allow TOY instructions of this type to be executed.

14. What values should be assigned to each of the control lines in the single cycle TOY Datapath given in lecture when the instruction AF10 is executed?

15. Consider the TOY instruction set, suppose we have decided to add the division operation, how many control lines (ALUctrl) do we need?

    a. two
    b. three
    c. four
    e. five
    f. six

16. Consider the TOY datapath. If we read and write the same register in a single instruction (such as R1=R1+R2) under the single cycle design, how is the TOY datapath able to avoid repeatedly fetching and incrementing the same register (R1) multiple times in a cycle?

For the remaining questions, assume the following of the TOY datapath:

a) fetching an instruction from memory takes 3 ns;
b) reading from the register file takes 1 ns;
c) executing a multiplication in the ALU takes 4 ns;
d) executing any other ALU operation takes 2 ns;
e) reading from or writing to memory takes 3 ns;
f) writing the result back to the register file takes 1 ns;
g) all other delays (those of MUXes, controls, wires, etc.) are 0.

(Each of these operations must finish inside a single cycle.)

17. How long does each of the following instructions take?

a) add
b) multiply
c) store (to memory) using indexed addressing
d) load (from memory) using indexed addressing

18. Single cycle vs. multicycle designs.

a) What is the minimum cycle time for a single cycle design?
b) What is the minimum cycle time for a multicycle design?

19. Suppose I speed up the ALU multiply operation so it now only takes 2 ns.

a) What is the minimum cycle time for a single cycle design?
b) What is the minimum cycle time for a multicycle design?

20. Suppose I slow down ALU multiply operation so it now takes 8 ns.

a) What is the minimum cycle time for a single cycle design?
b) What is the minimum cycle time for a multicycle design?