CSE370: Introduction to Digital Design
Winter 2000

Homework Set 5

DUE: MONDAY Feb. 14, 2000, 12:30 pm

a) **Please show *all* of your work. In certain problems, you may be asked to use Design Works. Otherwise, solutions do not have to be typeset, but may be if desired. In any case, your solutions must be legible. Please staple all the pages together. Make it clear which problem is which (especially important for the printouts from Design Works).**

For the first two problems: you can use the analysis and any relevant logic equations from the textbook. If it makes things easier, feel free to xerox the PAL or PLA programming maps (p.217, etc.) and write directly on them.

1. Katz problem 4.1 (p.231).

2. Katz problem 4.2 (p.231).

3. Katz problem 4.24 (p.237), ROM only. You don't need to write out the truth table separately if it is somehow shown as part of your schematic.

4. In a previous homework set, you created a DesignWorks project for a binary full-adder. You drew the adder schematic, created a block symbol for the adder, verified that your adder functioned correctly, and cascaded four full adders to make the ripple-carry adder shown on pg. 249 of Katz. Now you will construct three additional adders for your library.

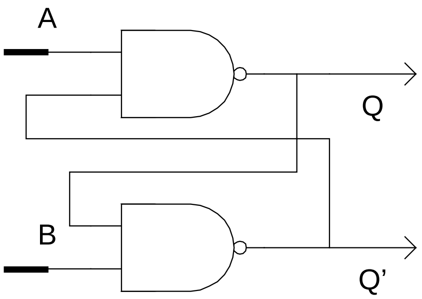
a) Construct a 4-bit carry-lookahead adder using pg. 255 of Katz as a reference. Use a reasonable level of modularization by creating your own sub-components as appropriate. For example, a flat schematic composed of a jumble of gates is probably not the most appropriate implementation (just as a C program with only a main() routine and no functions would probably not be an optimal solution for most assignments in a software course.) Turn in a DesignWorks schematics of your full-adder including the adder itself and the carry-lookahead logic. Verify that your adder operates correctly (you don't need to turn in schematics showing correct operation-- just verify that your circuit works right).

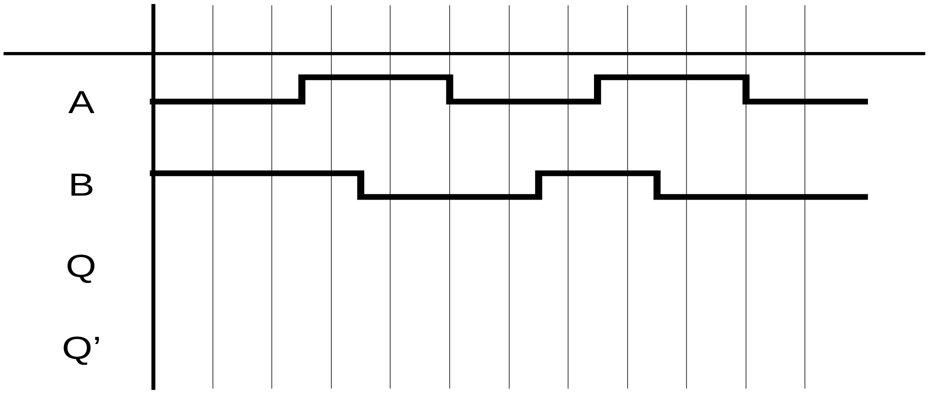
b) Construct an 8-bit carry-select adder. Use instances of your carry-lookahead adder from part a) in the implementation. Turn in a DesignWorks schematics. Verify that your adder operates correctly (you don't need to turn in schematics showing correct operation-- just verify that your circuit works right).

c) Construct a 4-bit adder as a Verilog module in Designworks. Use the Verilog "+" operator to implement addition. Refer back to the tutorial on the web page to the section on how to get going with Verilog. Make sure that your module includes a carry-out. Turn in your Verilog source code

d) Connect a keypad and probes to your Verilog adder block for testing. Verify that your adder works. Turn in printouts showing the following two tests:
i) "1111" + "0001"
ii) "1010" + "0101"

5. Fill in the output waveforms Q and Q' for the following circuit given the inputs A and B. Assume that each tick mark on the waveform graph represents 4 NAND-gate delays. In a concise paragraph, compare and contrast the behavior of this circuit with the cross-coupled NOR gates shown on pgs. 286-287 of the text.





6. Katz problem 6.11 (p.324). You can xerox the page in the book and write on it if you wish.