CSE370: Introduction to Digital Design
Autumn 2001

Homework Set 6

DUE: Friday, November 16, 2001

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).

1. (6 points) Katz problem 4.24 (p.237) (Gray code). Rather than give detailed schematics, it suffices to "size" the parts – i.e., what are the parameters of the ROM and the PLA? State generally how they would be wired, in English prose and/or with a rough schematic. You don't have to give full internal details. [Hint: Gray code is already worked out pretty well in the lecture slides.]

2. 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) (6 points) 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. Turn in at least two printouts with inputs and outputs, as you did for HW5.

b) (5 points) Construct an 8-bit carry-select adder. Use instances of your carry-lookahead adder from part a) in the implementation. Turn in DesignWorks schematics, including some examples showing correct operation.

c) (4 points) Challenge: construct a 4-bit adder as a Verilog module in DesignWorks. Suggestion: use the Verilog "+" operator to implement addition. Refer 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. If you can't get it to actually work in DesignWorks, that's OK (this time). In that case, just write out the code as best you can, even in longhand if necessary. Indicate what parts of it you are uncertain about. You'll get credit for making any kind of decent effort. [Hint: it can be a *very* short program if you use the "+" operator as suggested. But do it any way you like.]

3. (8 points, 2 per part) Katz problem 6.11 (p.324) (flip-flop waveforms). You can Xerox the page in the book and write on it if you wish.