5) Create a DesignWorks project for a binary full-adder. Be sure and save your finished project since you will use it again on a later assignment. You can find a block diagram, a truth table, and a schematic diagram for a full adder on Katz pgs.14, 19, and 250, respectively. Perform the following:

   a) Draw the full-adder schematic in DesignWorks, using standard logic gates from the PrimGate library. Label *A*, *B*, and *Cin* as inputs; *Sum* and *Cout* as outputs. Use "port in" and "port out" symbols from the Pseudo library to label your inputs and outputs. Turn in your schematic drawing.

   b) Create a block symbol for your full adder (as on pg. 14 of Katz). Label the symbol "FullAdder". Create a new schematic and test your adder with the "binary probe" and hex keypad method as described in question 4). Turn in this schematic showing the probe and keypad.

   c) Draw a schematic where you cascade four full-adders (four of your blocks from part (b)) as in Fig. 6 on pg. 249 of Katz. Label the inputs *A0, A1, A2, A3, B0, B1, B2, B3*; label the outputs *S0, S1, S2, S3,* and *C4*. Attach five "binary probes" from the Primio library to the four *Sum* digits and to *Cout*. Connect *Cin* for the first adder to logic "0". Attach two "hex keyboards" from the Primio library to your *A* and *B* inputs. Verify that your adder computes the sums *A+B* correctly, where
      i) *A*=0 and *B*=3
      ii) *A* =7 and *B* =7
      iii) *A* =9 and *B* =7

Turn in three schematic sheets, showing the correct outputs for these three cases on the binary

b). The inputs should be A3, A2, A1, A0 and output F. Note that F should be true if an odd number of the inputs are true. Turn in your schematic.

CSE370: Introduction to Digital Design
Winter 2000

Homework Set 3

DUE: Jan 26, 2000, 12:30 pm

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. Katz exercise 2.3 (a) and (b).

2. Katz exercise 2.28 (a) and (b).

1) Follow through the design process to create a logic circuit that detects pairs of 1's in nonadjacent positions of a 4-bit binary word. Specifically, given a 4-bit word ABCD, the circuit output F shall be logic true when (A=1 and C=1) or when (B=1 and D=1) or when (A=1 and D=1). The output F shall be logic false for all other cases.
   a) Draw the truth table and K-map.
   b) Express F in *minimized* sum-of-products form.
   c) Express F in *minimized* product-of-sums form.
   d) Using your expression for F from (b), draw a circuit (logic) schematic for F. You don't have to use DesignWorks here, but may if you wish.
   e) Modify your schematic so that you are using only NAND gates or their equivalents. Do not use unnecessary gates.
   f) Modify your schematic once more so that you are using only NOR gates or their equivalents. Again, do not use unnecessary gates.

2) Katz Exercise 3.13

3) Katz Exercise 3.16 (a) and (b). Draw the truth tables, do the K-maps, and write the hazard-free functions from the K-maps. You do not need to draw the schematics.

4) Create a DesignWorks projects for some XOR components.

   a) Draw a DesignWorks schematic for a two-input XOR function with inputs A, B and output F using only NAND gates from the PrimGate library. Use "port in" and "port out" symbols from the Pseudo library to label your inputs and outputs. Turn in this schematic.

   b) Create a block symbol for your XOR component and label it appropriately. Now produce a new schematic with your XOR block on the sheet. Attach a "binary probe" from the Primio library to *F*. Attach a "hex keyboard" from the Primio library to *A* and *B* (note: the bottom wire on the hex keyboard is the least significant bit). Select digits from the keyboard, and verify that your XOR block outputs are correct. Turn in your schematic drawing, with the keyboard and probes.

   c) Draw a schematic for a four-input parity function similar to what you did in part a), but instead of using NAND gates, reuse only instances of your 2 input XOR block from part