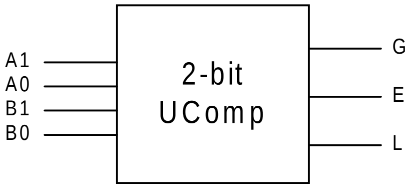
g) What is the gate count and the propagation delay of the 4-bit UComp blocks that you have designed so far (the one in part d and in part f)

h) Given the results from part g, you decide not to show the Bit Slice design to your boss. One of your colleagues though, comes up with the same exact idea (BSUComp, etc.), and decides to pompously present it at the group meeting. The next day, this colleague of yours is fired! BUT, as luck would have it, the day after getting fired, your colleague gets a job at SFEC inc. (Slow but Flexible and Extensible Chips). What happened? (Hint: What happens when the Zentium ™ branch prediction engine makes the fantastic leap to 8 bits, and you will need to redesign your comparator? What will happen to the gate count and the propagation delay in each of the two designs? How much effort will be required to design the new comparator? What are the tradeoffs here?)

You start writing the truth table for this, but you quickly realize that there are 8 variables, so that's 256 rows in the truth tables. And on top of that, those K-maps that were advocated so much in cse370 back when you were in school don't even work on 8 variables!!! So, you decide to start with a smaller more manageable problem, say that of comparing two 2-bit numbers.

a) Design the following 2-bit UComp component, which has the exact same specification as above, except that A and B are only 2-bit numbers (design here means do a truth table, use K-maps to minimize, and draw the resulting circuit)



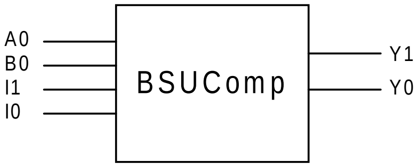
b) Show how you would construct the 4-bit UComp block using **two** 2-bit UComp blocks, and a third block with **6 inputs and 3 outputs** (you do not need to design the 6 input block, just say what it does)

All you need to do now is design the block with 6 inputs! This is doable with K-maps, you say… But real designers like you don't waste their time doing 6 variable K-maps! They find better solutions. Thus:

c) Design a new 2-bit UComp block which has only 2 outputs (hint: let your outputs be Y_1 and Y_0, and have Y_0Y_1 = 00 mean smaller, Y_0Y_1 = 01 mean equal and Y_0Y_1 = 10 mean greater).

d) Do b) again, but this time use **two** 2-bit UComp blocks and a third block with **4 inputs, and 3 outputs**. Design this third block using truth tables and K-maps.

Great! Now, you've done the design for this new comparator. Your boss really likes the design, but isn't impressed enough to give you a bonus. So you keep on thinking about the problem, in the hope of finding something better (something, shall we say, bonus-inducing?). Finally, you come up with this incredible idea: the Bit Slice UComp, or BSUComp for short (shown below).



The BSUComp compares two **n-bit numbers**, of which the **least significant bits** are A_0 and B_0. *What, you say? How can we compare two n-bit numbers according to only the lsb?* Well, notice that there are two additional inputs. These two inputs (encoded according to part d) will give you the result of **comparing all the bits in A and B that are more significant than the lsb**.

e) Design the BSUComp block (using, you've guessed it, truth tables and K-maps. Hope you're seeing a pattern here about what the design process is…)

f) Show how you would construct a 4-bit UComp block using **four** BSUComp blocks and a block with **2 inputs and 3 outputs**. Design this 2 input, 3 output block (using what?… you've guessed it, truth tables and K-maps, of course)

CSE370: Introduction to Digital Design
Winter 2000

Homework Set 4

DUE: Feb. 2, 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).

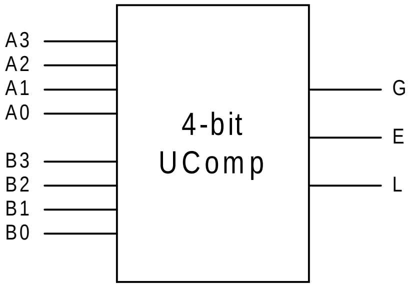
1) [Leftover from Homework #3. If you handed it in last week, please extract the page with this problem and hand in again (make sure the page is labeled we don't confuse it with something from this set.] Katz Exercise 3.16 (a) and (b). Draw the truth tables, do the K-maps, and write the hazard-free functions from the K-maps. You do not need to draw the schematics.

2) Katz problem 4.15 (a), (b), (c) and (d). Draw your solution to part (a) in DesignWorks, using a Mux8 device from the Primlog library with its EN input tied to 0. In (b), you may use an OR gate that has less than 16 inputs. In (c), draw a block diagram of the ROM, label the address lines, and explain what bit values you would store at each memory location. In (d), don't forget to minimize the logic before implementation. Only part (a) must be in DesignWorks, but the others may be if you wish.

PROBLEM 3

You have just been hired by RFC Inc. (Really Fast Chips) as an engineer in a top secret group: the one responsible for the next generation of RFC processors, the Zentium VII ™. One of your tasks is to design a 4-bit unsigned number comparator, called UComp. This block is an essential part of the new processor since it is used in one the most critical components, RFC's new Allervassen branch prediction algorithm. So, it is important that you get this design really well done.

The block you need to design looks like this:



It will take two 4-bit unsigned numbers, and compare them. There are 8 inputs, which are the bits of the two numbers to compare: A3..0 is one number and B3..0 is another (A0 is the lsb, A3 is the msb). There are three outputs: G is asserted iff A > B, E is asserted iff A=B and L is asserted iff A < B (there is never more than one output asserted at once)