

Problem 1: Consider the pair of logic functions (yes, they are from Homework 5).

$$f_0 = (a + bc + \bar{b}cd) \bar{e}$$

$$f_1 = (a + bc + b\bar{c}d) \bar{e}$$

(a) Write a Verilog structural description of a single module implementing these functions. Take advantage of the common terms in the two functions ($a + bc$).

(b) Notice that one function contains a $\bar{b}cd$ term and the other contains a $b\bar{c}d$ term. Though the variables are different the two terms represent the same operation (a three-input AND with one input to the AND inverted). Write a Verilog structural description for these functions taking advantage of this fact by having two modules. One module will be the three-input AND gate just mentioned. The other will compute the functions using two instances of the AND gate.

Problem 2: Answer the following questions about EDA (CAD).

(a) How does a synthesis program know the capabilities of the chip it is targeting, such as the number of gates it can hold?

(b) Comment on the correctness the statement below:

“Before writing a design in Verilog one should simplify the combinational logic (perhaps using Karnaugh maps) so that the resulting chip will have lower cost.”

(c) Why must a timing simulation be performed after synthesis? Why can't the simulator read some files that contain gate propagation delays and use that to perform a timing simulation before synthesis is ever done?

(d) The same Verilog file for a design (or VHDL) is read by simulation programs (for functional simulation) and synthesis programs, meaning that Verilog must be good for both simulation and synthesis. Why not have separate languages for simulation and synthesis?

Problem 3: A 2^n -input multiplexer can be build using $2^n n + 1$ -input AND gates and an 2^n -input OR gate. Each input to this mux is just one bit. In practical applications one might want to switch larger inputs, for example, each input might be an 8-bit quantity. Explain why the cost of an 8-bit 2^n -input mux is less than 8 times the cost of a 1-bit 2^n input mux. Draw a logic diagram to illustrate your answer.

Problem 4: *Note: Don't attempt this problem until you are comfortable with all other material in this course.* Design logic to determine if one 16-bit unsigned integer is greater than another one.

(a) Do this by designing a block with four inputs, a , b , e , g . There will be 16 blocks. Inputs a and b in block i connect to a_i and b_i , the bit at position i in a and b , respectively. Input e in block i is true if bits i up to 15 in a and b are equal, it is false otherwise. (That is, if $a_j = b_j$ for $i < j < 16$.) Input g in block i is true if $a > b$ looking at bits $> i$. (That is, there exists some k such that $a_j = b_j$ for $k < j < 16$ and $a_k = 1$ and $b_k = 0$.) It is false otherwise.

The block will have two outputs, E and G . As the alert student has already guessed, output E of block i connects to input e of block $i - 1$, similarly for G . For block 15 (corresponding to the most significant digit) input e_{15} is 1 and g_{15} can be set to any value (0 if that's less confusing).

Design the block, and draw a diagram showing how they are connected. Also show how the overall output is generated, a one-bit signal that is true if $a > b$.

(b) Design the $a > b$ circuit using the binary full adder (BFA) as a building block. Negate b using inverters and by taking advantage of the unused carry-in. This design should consist of a bunch of BFAs (just show a box for them) and inverters. Don't forget to show the one-bit $a > b$ signal. One might need to review 2's complement arithmetic before completing the problem.