Due: Monday, January 25, 12:00PM (except A.3, C.2 as noted)

For all assignments in this class: Writeups must be done in electronic form and submitted through blackboard. Use CAD or drawing tools where appropriate. Handwritten assignments and hand-drawn figures are not acceptable.

You may do sections (A and B) or (B and C). C is primarily intended as a more challenging (interesting) alternative for students who have already had considerable experience with digital logic.

You may use hierarchial schematics. Use of a schematic drawing program for circuits is encouraged.

A: Basic Logic

1. Implement $A > B$ out of 2-input NAND gates; assume $A$ and $B$ are 4b unsigned numbers. (Hint: design the bit slice and show how the bit slices compose for 4b numbers.)

2. Using your comparison function from A.1, show logic for a spatial sorting function to sort 4, 4b inputs into ascending order.

3. Show the logic (RTL logic – i.e. logic equations and registers) for a simple vending machine. [Defer this to Monday, February 1 – return with HW2.]
   
   Inputs: n, d, and q, (nickle, dime, quarter)
   Output: v (vend), nc (nickle change)
   Function: Collect $\geq$30 cents, then vend and give change in nickles.

   • Don’t worry about running out of nickles to provide as change.
   • Include a diagram of your state-transition graph in your writeup.
   • Hint: It is probably easier not to use adders and datapath logic for this problem.

Course Web Page: <http://www.seas.upenn.edu/~ese534/>
B: Properties of Boolean Functions

1. Consider all two-input functions. (How many functions are there?)
   For each function, identify if the function is universal; you may tie the inputs of a
   function to a constant 0 or 1. Your writeup should be a table, with the following
   entries for each two-input function:
   
   • list on-set minterms (i.e. the truth table)
   • logic expression for function
   • universal?
   • explanation of why or why not

2. Counting each gate as unit size, give a bound on the size ratio between an optimal
   implementation of an arbitrary \( n \)-input function when the implementation may use an
   optimal mixture of the full set of 2-input functions from B.2 as gates compared to an
   implementation which uses only 2-input NOR gates.

C: Advanced Logic Problems

1. Using only two-input NOR gates, give a bound on the number of different functions
   that can be implemented with depth \( l \). (Your bound should be non-trivial, but does
   not need to be tight.)

2. Firing Squad – Design the logic for an FSmodule. [Defer this to Monday, February 1
   – return with HW2.]

   • FSmodules can be assembled into a 1d array of arbitrary length.
   • Each FSmodule is connected exclusively to his left and right neighbors.
   • The leftmost FSmodule will get a start input.
   • FSmodules may have configuration input bits which distinguish the leftmost and
     rightmost modules from the rest (i.e. a module will be leftmost, rightmost, or a
     chained element).
   • All FSmodules are clocked together.
   • Data can travel from one FSmodule to his adjacent neighbor in one cycle.
   • You can have a constant number of wires between adjacent FSmodules (independent
     of the length of the 1d array).
   • The state in an FSmodule is finite and independent of the length of the 1d array.
   • In response to an input pulse on the leftmost module, the array of FSmodules
     should all, simultaneously flash an output light.
   • The number of cycles between the input pulse and the synchronized firing of the
     FSmodules’ lights is not restricted.

Show your state-transition graph and gate logic (you may write equations for the logic
as long as the equations identify the primitive gates). Describe the operation of your
solution.