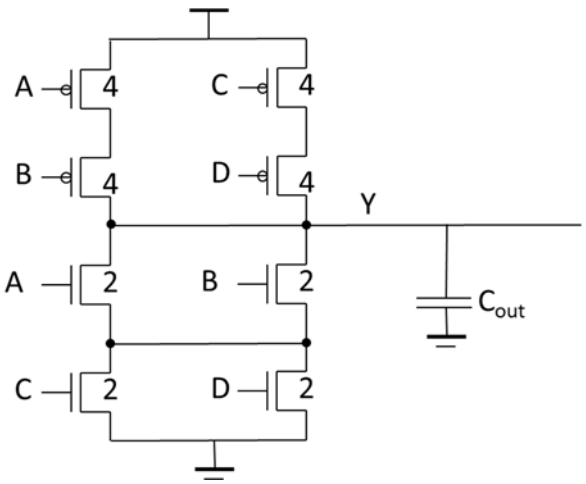
NAME_____

EN1600: Design and Implementation of VLSI Systems

Homework Assignment #3

Due **Friday, October 21, 2016, by 5pm**

1) Consider the following circuit. The sizes for all the transistors in the gate are given next to each transistor relative to a minimum sized transistor:



(a) If B=0, C=1, D=0, draw the switch model you would use to calculate the delay of the gate when A transitions from 0 →1. Include all relevant resistances and capacitances in the switch model for this input pattern (do not ignore internal capacitances). Label all resistances and capacitances uniquely (e.g., R_{pA}, R_{nB}, C_{int1}). What are the initial/final values of the output?

(b) Assume that C_{diff} = 0.1fF, R_n=2kΩ, and R_p=4kΩ for a minimum sized transistor. Also, C_{out}=7fF (but does not include parasitic capacitances). Ignore parasitic gate capacitance and assume no sharing/merging of diffusion regions for adjacent transistors. Compute all the capacitance and resistance values for all elements in your switch model from part (a). Clearly state all assumptions for your calculations.

(c) Compute the delay of the gate for this case using the Elmore delay model. Show all your work. How does this delay value compare to one that ignores internal capacitances?

(d) Repeat parts (a), (b), and (c) for the case where A=1, B=0, D=0, and C transitioning from 1→0. Again, clearly state all your assumptions.

2) Consider again the function from problem 1) where $Y = \overline{(A + B)(C + D)}$. Transistor sizes are the same as above for the complex gate. The following problems use logical effort to minimize path delay.

   (a) Compute the logical effort for the gate.

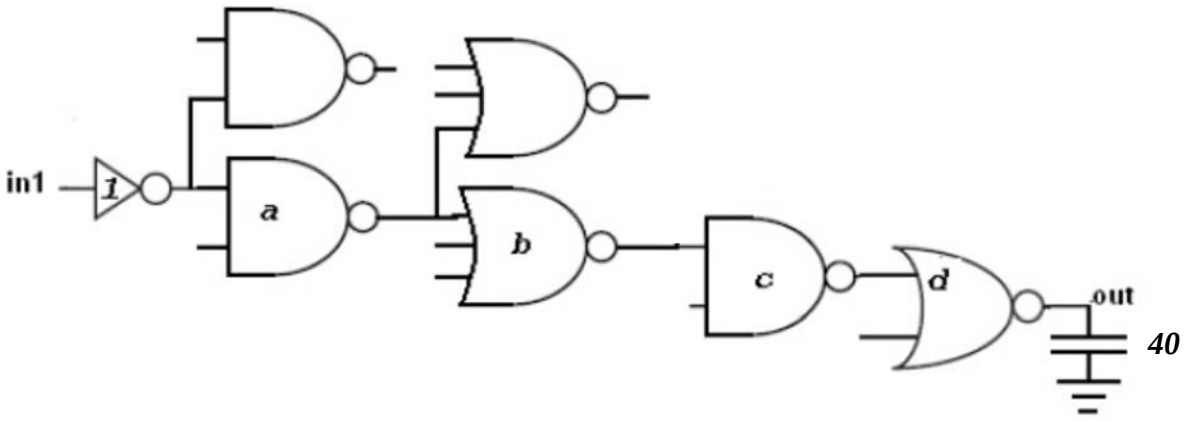
   (b) Assume that $C_L = 200C_{in}$ and that you can add as many inverters as you'd like after the complex gate (you don't need to maintain the logical polarity at the output), size the chain for the minimum delay (HINT: use the FO4 rule of thumb to find the optimal number of inverters). What is the minimum delay for this design in terms of t_{p0} (the delay of a minimum sized inverter)?

   (c) Now implement the function Y using only NAND, NOR, and inverter gates. Draw the gate-level implementation.

   (d) Assuming C_{in} for the chain from part (c) is the same as for part (b) and that $C_L = 200C_{in}$, size the chain from (c) for minimum delay (again, you can add inverters if needed and don't need to maintain the logical polarity). What is the minimum delay with this sizing, and what is the total transistor width of all of the gates?

   (e) Now redo parts (b), (c), and (d) assuming $C_L = 50C_{in}$. Comment on the relative tradeoffs of these two implementations given your results for all parts of this problem.

3) Consider the logic network shown below. The output of the network is loaded with a capacitance that is 40 times larger than the input capacitance of the first gate (a minimum-sized inverter). Find the gate sizes a, b, c and d that lead to minimum delay of the path.



4) Consider the function $Y = (A \bullet B + \overline{C \bullet D + E}) \bullet (F + \overline{G})$

(a) Sketch the transistor-level implementation using ***multi-stage domino logic*** (with precharge and evaluate transistors). Note that there are multiple ways to decompose this function into a multi-stage domino circuit. Try to balance stage complexity, series path resistance, and number of stages when coming up with a suitable design. Also, remember that domino logic can only implement positive logic in order for it to evaluate correctly. You can assume you have both true and complementary inputs available.

(b) Given your design from part (a), size the transistors as appropriate, so worst case fall delays are approximately equal for each stage (1^{st} order approximation).

(c) Determine the worst-case input vector pattern for the circuit.

(d) In Virtuoso, create a schematic view for the domino circuit you designed. Create a separate view for each stage in the resulting domino logic. Your top view should be obtained connecting the symbol views for each stage.

(e) For each of the views you have created in part (d), design a layout cell view. Use the same pitch for each cell view. Create a layout top view by connecting instances of the stage layout views. You will have to use metal2 for routing the stages of your domino logic. When routing the different stages, your goal is to use the same metal layer for a certain routing direction (for instance, use metal1 for the horizontal routing and metal2 for the vertical routing). Run DRC and LVS at each step.

(f) Extract the netlist from the layout top view of your domino circuit using Calibre PEX and simulate it using the input vector found in part (c).

Hand in a hardcopy of your schematic, layout, and simulation results. Also, please remember to put your libraries in the /gpfs/data/engn1600 directory. It is easier if you do this initially, when you first start working on the assignment, instead of moving your libraries afterward when your designs are complete.