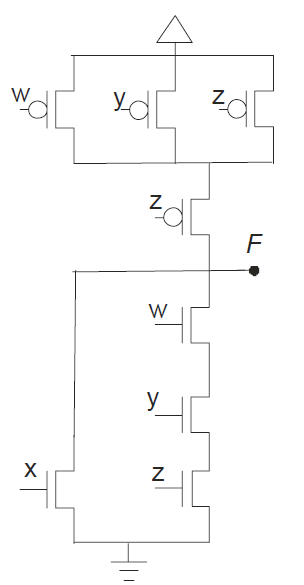
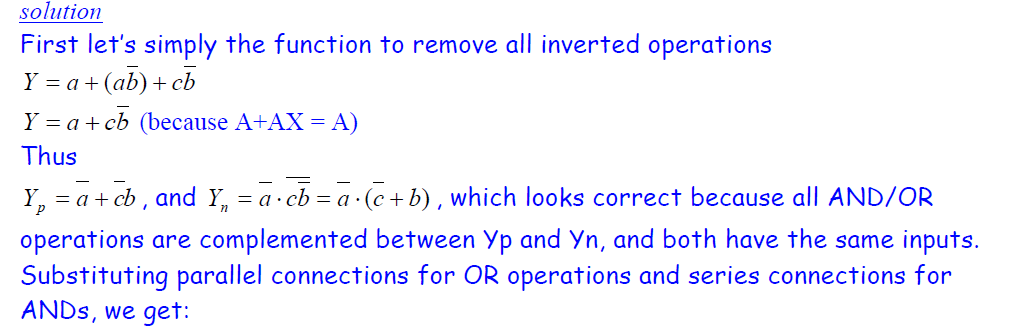
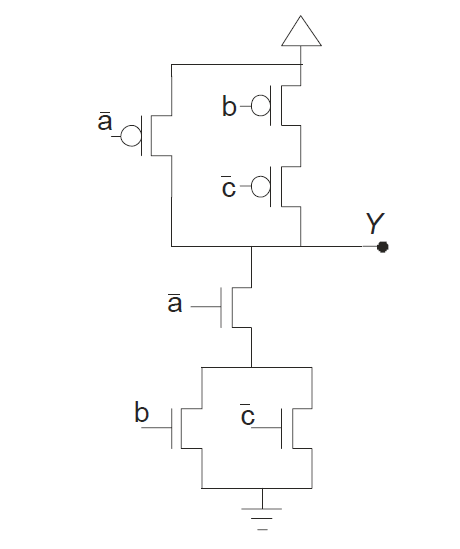
Q1: Design a transistor-level CMOS logic circuit to implement the function using the least number of transistors. HINT: Consider that you may need to expand the equation in order to reduce it using the logic properties shown in the Chapter 2 lecture notes.





Q2:Construct the CMOS logic circuit that implements using the fewest possible transistors. You are allowed to use inverted inputs (e.g., *a’*) rather than adding inverters to create these signals.



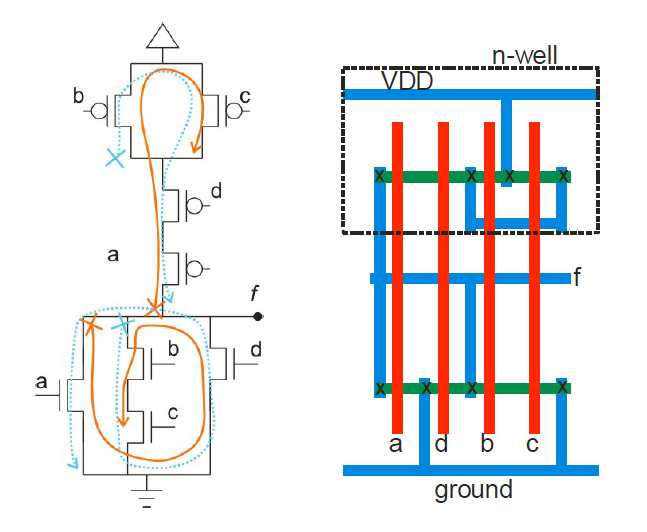


Q3:

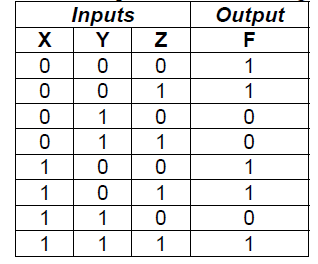
Sketch a color-coded stick diagram for the circuit that implements the function *f=(~(A+(D+(BC))*. Organize the layout so that the transistors can be implemented on a continuous strip of active (i.e., do not break the active).

**solution:**

An interesting feature of this function is that you have to organize the transistors within the schematic correctly in order to achieve the ‘one continuous strip of active’ layout goals. The schematic below shows one possible implementation. Notice the b || c pMOS devices must be directly attached to either the ground or output nodes in order to draw a loop that does not violate the rules. Two possible loops (with X at the start and an arrow at the end) for this schematic implementation are shown (solid-orange line and dotted-blue line). The solid-orange loop is used to set the order of poly traces in the stick diagram below to a, d, b, c. However, there are many possible layouts for this function.



Q5: Draw the schematic for the CMOS circuit that implements the function ***F*** described by the truth table below. Use the least possible number of transistors. Explain your procedure and show the reduced function equation used to design the schematic.



**solution (corrected)**

There are two basic approaches: we can construct a K-map or write a sum of products expression and reduce it. Let’s try the sum of products option.

We can write the sum of products for either the high terms (F) or the low terms (F’). Because there are fewer low/zero terms, let’s try those.

F’ = (X’YZ’ + X’YZ + XYZ’) = X’Y (Z’+Z) + XYZ’

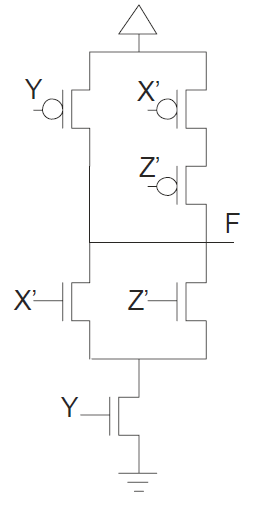
F’ = X’Y + XYZ’ = Y (X’+XZ’) 􀃆**F’ = Y (X’+Z’)**

Thus **F = Y’ + XZ**

Alternatively, by observation we can see that F = 1 when Y’ OR XYZ is true. Thus, F = Y’ + XYZ. Using the property covered in HW1 we can reduce this to

**F = Y’ +XZ**

which matches our sum of products results. Note that we should get the same results if we started with the high terms sum of products



Q6: **Lab:**

Implement using Micrwind or any layout editor and 35nm Process a 2 input :

* NOR
* XOR
* TURN ON :
  + Layout screenshot
  + Power
  + Area of the cell

Use this link to help you out

<https://www.smohanty.org/Teaching/2008Fall_VLSI/MohantyVLSI5Microwind.pdf>