EE 330

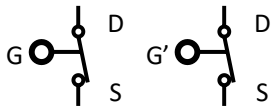
Homework 3 Solutions

Corrected Sept 27

Fall 2020

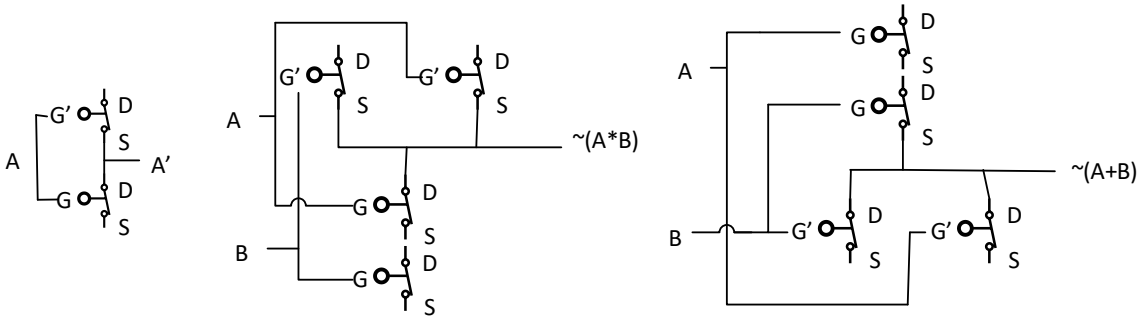
Solution 1

I will use the following for an NMOS and PMOS respectively



The NOT, NAND and NOR gates are drawn like:

NOT NAND NOR



To create the AND gate, we need to cascade the NAND and NOT such that the output of the NAND is the input of the NOT gate, and the output of that will be the AND logic operation of A and B. Similarly you can do the same for the NOR with the NOT gate.

Solution 2

The switch-level model for a MOSFET is the model frequently used in courses such as CPR E 281, where it is assumed that MOSFETs act like perfect switches which close or open depending on some input voltage. This model is extremely simplistic, making it good for basic back-of-the-napkin designs and prototypes. It is also a good model to start with when debugging since it allows you to quickly look at a schematic and say, “does the logic make sense?”

The improved switch-level model is similar to the switch-level model, but with a little bit more complexity. The improved switch-level model also models the ability for a MOSFET to switch between two states, but now it also includes the fact that MOSFETs cannot switch instantly unless they’re driven by an ideal source. This is because of the gate-drain capacitance and drain-source resistance of the MOSFET. This model is useful for basic digital logic design where you are concerned about the timing of individual transistors, although it is still extremely simplistic.

Solution 3

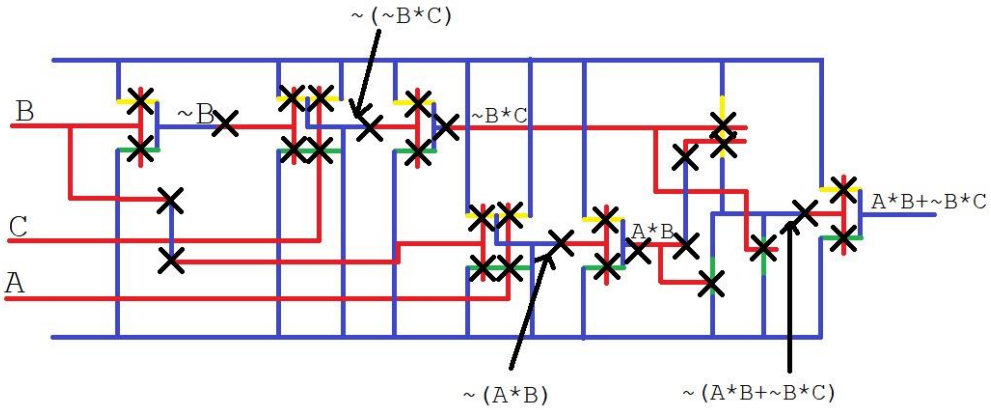
For the first inverter to drive 1, it would be:

$$t_{LH} = 2 * 6k\Omega * 1.5fF = 18psec$$

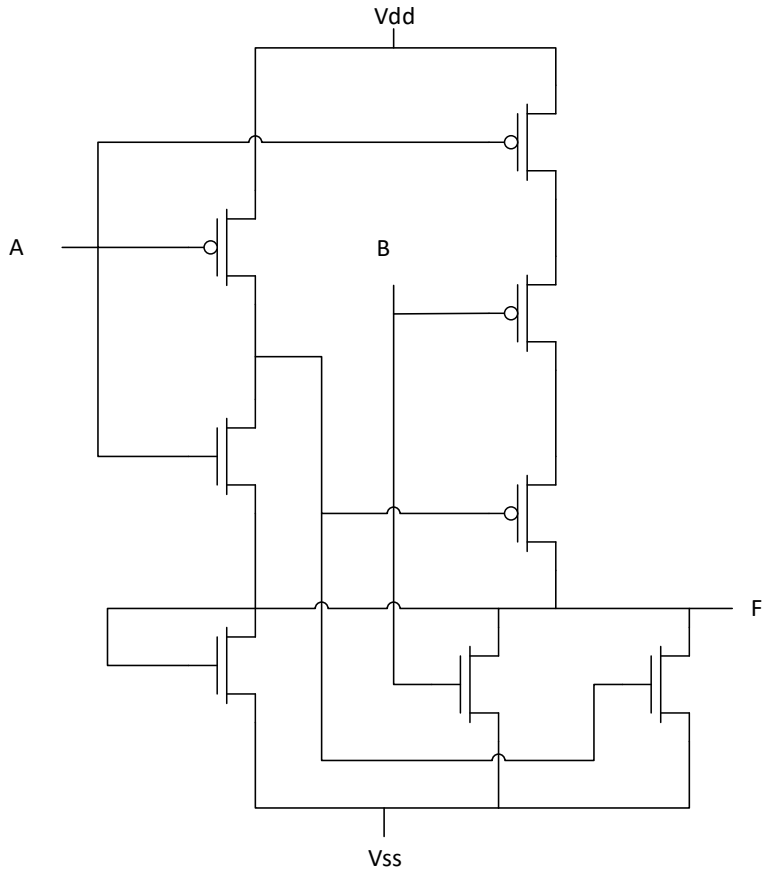
Since it is driving 6,

$$t_{LH} = 2 * 6 * 6k\Omega * 1.5fF = 108psec$$

Solution 4



Solution 5



Solution 6

To do this problem, let's take a look at each stick diagram individually and see what transistors we can spot inside of it. To begin, let's look at the top-left stick diagram. It has both a p-active and an n-active strip in it, meaning that it contains both a NMOS and a PMOS device. That makes sense. We can also see that the p-active strip has two gates intersecting it, forming two series PMOS devices between VDD and VOUT. In comparison, the n-active strip is configured so that current can flow from VOUT to VSS in one of two ways; either through the A Poly or through the B Poly. This implies that the NMOS devices formed by the n-active strip are placed in parallel. Two series PMOS devices followed by two parallel NMOS devices is indicative of a NOR gate.

We can follow the same train of thought for the bottom-left stick diagram. This diagram is the same as the top-left stick diagram, except that the n-active and p-active strips have been swapped, along with VDD and VSS. This means that there are now two parallel PMOS devices followed by two series NMOS devices, which is the configuration for a NAND gate.

The top-right stick diagram is the same as the top-left diagram, except with an extra set of n-active and p-active strips added in. These strips form a PMOS and NMOS pair which are connected together, creating an inverter. Thus, this stick diagram is that of an OR gate.

With the other three stick diagrams figured out, the bottom-right stick diagram can only be the AND logic function.

Solution 7

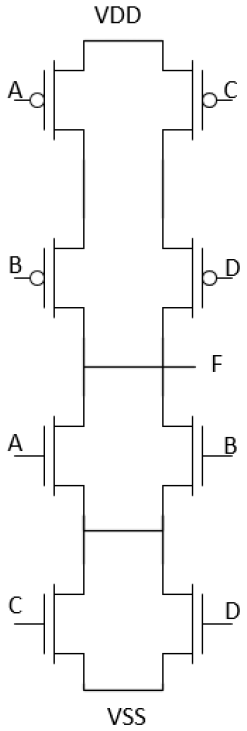
To begin, we must create the pull up (PU) and pull down (PD) networks for this expression. Let's start with the PU network. To do so, take the Boolean expression and invert each of the inputs (since the PU network is implemented with PMOS devices) and apply De Morgans' Law:

$$F = \overline{(A+B)*(C+D)} = \overline{(\bar{A}+\bar{B})*(\bar{C}+\bar{D})} = \overline{(\overline{AB})*(\overline{CD})} = AB + CD$$

Now, create the PD network. To do so, solve for $\bar{F}$ instead of F since the PD network pulls the logic output to 0 instead of 1:

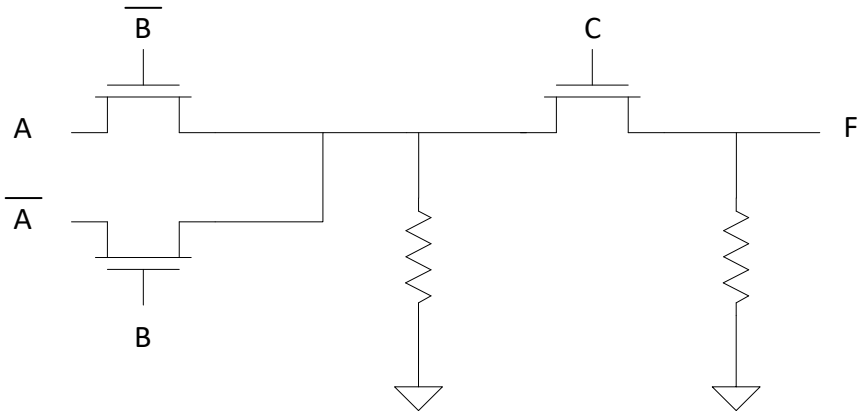
$$\bar{F} = \overline{\overline{(A+B)*(C+D)}} = (A+B)*(C+D)$$

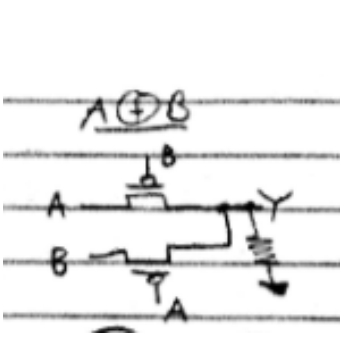
This creates the following logic circuit:

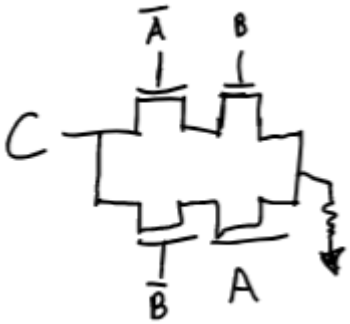


As we can see, using Complex Gates, it only takes 8 transistors and 1 level of logic to implement the Boolean expression Complex Logic. In comparison, to implement the Static CMOS version of the Boolean expression, it would take 1 NAND gate (4 transistors each), 2 NOR gates (4 transistors each), and 2 NOT gates (2 transistors each), bringing the device count up to 16.

Solution 8







Solution 9

The expression is $\bar{A} * B + \overline{(A * \bar{C})} + C * \bar{B}$

$$\bar{A} * B + \sim(A * \bar{C}) + C * \bar{B} = \bar{A} * B + \bar{A} + C + C * \bar{B} = \bar{A} * (B + 1) + C * (1 + \bar{B}) = \bar{A} + C$$

The expression can be simplified to $\bar{A} + C$

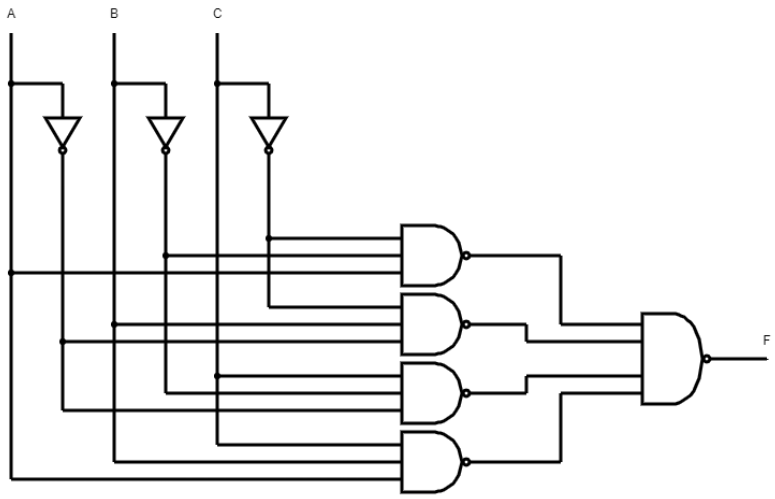
Solution 10

To begin, lets write the basic expression for a three-input XOR gate using SOP (sum of products) form: $F = A\bar{B}\bar{C} + \bar{A}B\bar{C} + \bar{A}\bar{B}C + ABC$.

We need to get this expression to use only NAND, NOR, and NOT functions, so let's start by not-ing the overall sum twice. Then, we'll use De Morgans' Law to simplify:

F = $\overline{\overline{A\bar{B}\bar{C} + \bar{A}B\bar{C} + \bar{A}\bar{B}C + ABC}} = \overline{\overline{(A\bar{B}C)}\ \overline{(\bar{A}B\bar{C})}\ \overline{(\bar{A}\bar{B}C)}\ \overline{(ABC)}}$

This expression implements a three-input XOR function using only NAND, NOR, and NOT functions. Now, we will create the circuit for it:



Solution 11

There are a number of reasons why this is not practical, at least at this point. One of the most significant reasons, however, is manufacturability. As a result of imperfect features, stray particulates, and other process effects, defects occur all the time on wafers. Most of the time, in an integrated circuit, any defect is unacceptable and significantly deteriorates the circuit's performance. It is simply too difficult to manufacture a defect-free die the size of a wafer with our current technology. As an example of this, consider a small $100mm$ wafer (approx. 4" in diameter) which is manufactured at a fab house which

generally sees defect densities of about 0.5 defects per square centimeter ($0.5cm^{-2}$). Note that this is a relatively small wafer and a relatively good defect density. Even so, the Hard Fault model predicts a yield of only $8.882 * 10^{-18}$, effectively meaning that you would need to produce 10^{18} wafers to get just 8 working wafers. That doesn't even account for other sources of error!

Solution 12

$0.8= e^{-1*d}$

$d= 0.223cm^{-2}$