DeMorgan Revisited
Last week, we discussed DeMorgan's Theorem and its application to Boolean expressions. As a refresher, the theorem is duplicated below.

\[ \overline{A + B} = \overline{A} \cdot \overline{B} \]

\[ \overline{A + B} = \overline{A} \cdot \overline{B} \]

In terms of the logic diagram, they look something like this:

\[ \overline{A + B} = \overline{A} \cdot \overline{B} \]

\[ \overline{A + B} = \overline{A} \cdot \overline{B} \]

Let's take the first equation, where the inverse of the sum is equal to the product of the inverses, and invert both sides of the equation.

\[ A + B = \overline{A} \cdot \overline{B} \]

\[ A + B = \overline{A} \cdot \overline{B} \]

This shows that the sum (OR-ing) of two variables is equal to the inverse of the product of their inverses. (Huh?) Well, for now, just make sure you see how we got the equation above simply by inverting both sides of the first of DeMorgan's Theorems listed above. The same idea can be applied to multiple inputs, once again, with DeMorgan's Theorem.
Okay, now that we have thoroughly baffled you, let's turn our attention to the sum of products format. Assume I have an equation like the one below.

\[ X = (\overline{A} \overline{B} \overline{C}) + (A \overline{B} C) + (A B \overline{C}) \]

The circuit might look something like the one below.

If we apply the same equation for the OR gate as we did above, we get:
Now, take each of the inverter circles at the inputs to the rightmost gate (the NAND gate that replaced the OR gate), and move them to the outputs of the AND gates. You should get:

Okay, well it's not like the discovery of penicillin or anything, but it is rather cool. It shows us that every SOP expression can be implemented with NAND gates replacing the AND gates and the OR gates. This results in only requiring one type of gate to implement ANY truth table. In addition, NAND gates tend to be one of the fastest gates available, hence this circuit should be faster than the direct SOP implementation.