Example: CMOS Logic Gate Synthesis

**Problem:** Design a CMOS digital circuit that realizes the Boolean function:

\[ Y = \overline{A} + B + \overline{A} \overline{C} \]

**Solution:** Follow the steps of the *design synthesis* handout!

**Step 1:** Design the PDN

First, we must rewrite the Boolean function as:

\[ \overline{Y} = f(A, B, C) \]

In other words, write the *complemented output* in terms of un-complemented inputs.

*Time to recall our Boolean algebra skills!*
We must first complement this equation, and then apply **DeMorgan’s Theorem** (several times!).

\[
Y = \overline{A + B + \overline{AC}} \\
\overline{Y} = \overline{A + B + \overline{AC}} \\
= (\overline{A + B})(\overline{\overline{A} + \overline{C}}) \\
= (A + B)(A + C) \\
= AA + AC + BA + BC \\
= A(A + B + C) + BC \\
= A + BC
\]

Logically, this result says:

\[Y \text{ is low if } A \text{ is high, OR if both } B \text{ AND } C \text{ are high.}\]

We can thus realize this logic with the following **NMOS PDN**: 
**Step 2:** Design the PUN

First, we must **rewrite** the Boolean function as:

\[ Y = \overline{A} + \overline{B}C \]

In other words, write the un-complemented output in terms of complemented inputs.
Again, using DeMorgan's Theorem:

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

Logically, this result says:

\[ Y \text{ is high if } A \text{ is low AND either } B \text{ OR } C \text{ are low.} \]

We can thus realize this logic with the following PMOS PUN:
Thus, the entire CMOS realization is:

\[ Y = \overline{A(\overline{B} + \overline{C})} \]
$Y = A + B + \overline{A \cdot C}$