K-map with don't care conditions:

Consider 4-bit binary $\rightarrow 0000 \uparrow 0, 3$ digits

Possible combinations: 1111

1010 don't care input conditions because they never occur in the input

we'll leverage this fact (don't cares) to simplify better.

30, 13

don't cares can be treated as a 1 or 0 on the k-map.
\[ f(w, x, y, z) = \Sigma(0, 2, 3, 5, 6, 7, 8, 9) \]

don't cares \( \Delta(w, x, y, z) = \Sigma(10, 11, 12, 13, 14, 15) \)

\[ w'y' + w'x + wz + wx'y' + x'yz' \]

4 terms
11 literals

without don't using don't care conditions:

now let's repeat this problem but with don't cares

\[ f(w, x, y, z) \]

\[ y' + w' + x'z' + xz' \]

= 4 terms
= 6 literals

Implement using NAND

\[ y = (y')' \]
\[ w = (w')' \]
\[ x = (x')' \]
\[ z = (z')' \]

\[ f(w, x, y, z) \]

\[ w' + x + y + z \]
\[ f(w, x, y, z) = \Sigma(0, 2, 6, 8) \]
\[ g(w, x, y, z) = \Sigma(10, 11, 12, 13, 14, 15) \]

2 terms \( \Rightarrow \) 3 literals each
6 literals

**Lesson:** You don't have to cover all the "don't cares"

2 groups of 4 1's:

\[ f(w, x, y, z) = x'z + yz' \]

**Sum of minterms**
**Canonical form**
**Sum of products** - **Standard form**

**NAND only implementation**
3.23

\[ F(A, B, C, D) = \Sigma(2, 4, 10, 12, 14) \]
\[ d(A, B, C, D) = \Sigma(0, 1, 5, 8) \]

Implement the simplified function using only 2 gates: (Hint: NOR gates)

- 3 terms \( \Rightarrow \) 3 gates
- 3 groups of 4 1's

With NAND implementation we cannot meet the constraint.

\[ A + 1: \text{ Cover only the 0's + 1's} \]

\[ f' = D \cdot 1 + A \cdot B \cdot C \]

\[ f'(D' + 1)^2 - D' = D' \]

\[ f: D + A + B \cdot C = (f')' = f \]
\[ (C \cdot D + A \cdot B \cdot C)' = D' \]
\[ F' = D + A'BC \]

\[ F = (F')' = (D + A'BC)' \]

\[ = (D + (A'BC))' \]

\[ A'BC = ((A'BC)')' \]

\[ = (A + B' + C')' \]

De Morgan's law