Standard Forms

- **Standard Sum-of-Products (SOP) form:** equations are written as an OR of AND terms
- **Standard Product-of-Sums (POS) form:** equations are written as an AND of OR terms
- Examples:
  - SOP: \( \bar{A} \bar{B} \bar{C} + \bar{A} \bar{B} \bar{C} + B \)
  - POS: \( (A + B) \cdot (A + B + C) \cdot C \)
- These “mixed” forms are **not** SOP or POS
  - \( (A \bar{B} + C)(A + C) \)
  - \( A \bar{B} \bar{C} + A \bar{C} (A + B) \)
Standard Sum-of-Products (SOP)

- **A Sum of Minterms** form for \( n \) variables can be written down directly from a truth table.
  - Implementation of this form is a two-level network of gates such that:
    - The first level consists of \( n \)-input AND gates, and
    - The second level is a single OR gate (with fewer than \( 2^n \) inputs).
- **This form:**
  - is usually *not* a minimum literal expression, and
  - leads to a more expensive implementation (in terms of two levels of AND and OR gates) than needed.

Standard Sum-of-Products (SOP)

- Therefore, we try to combine terms to get a lower literal cost expression, leading to a less expensive implementation.
- **Example:** \( F(A, B, C) = \Sigma m(1,4,5,6,7) \)
- **Simplifying:**

The Canonical Sum-of-Minterms form has (5 * 3) = 15 literals and 5 terms. The reduced SOP form has 3 literals and 2 terms.
AND/OR Two-level Implementation of SOP Expression

- The two implementations for F are shown below: (Which is simpler?)

![Diagram of two-level implementation](image)

Standard Product-of-Sums (POS)

- A Product of Maxterms form for $n$ variables can be written down directly from a truth table.
- Implementation of this form is a two-level network of gates such that:
  - The first level consists of $n$-input OR gates, and
  - The second level is a single AND gate (with fewer than $2n$ inputs).
- This form:
  - is usually not a minimum literal expression, and
  - leads to a more expensive implementation (in terms of two levels of AND and OR gates) than needed.
Standard Product-of-Sums (POS)

- Therefore, we try to combine terms to get a lower literal cost expression, leading to a less expensive implementation.
- Example: \( F(A, B, C) = \Pi_M(0,2,3) \)
- Simplifying

The Canonical Product-of-Maxterms form had \((3 * 3) = 9\) literals and 3 terms. The reduced POS form had 4 literals and 2 terms.

OR/AND Two-level Implementation

- The two implementations for \( F \) are shown below: (Which is simpler?)
SOP and POS Observations

- The previous examples show several things:
  - Canonical Forms (Sum-of-minterms, Product-of-Maxterms), or other standard forms (SOP, POS) can differ in literal cost.
  - Boolean algebra can be used to manipulate equations into simpler forms.
  - Simpler equations lead to simpler two-level implementations

- Questions:
  - How can we attain a minimum literal expression?
  - Is there only one minimum cost circuit?

Equivalent Cost Circuits

- Given \( F(A, B, C) = \sum (0,2,3,4,5,7) \)
\[
F = \overline{A} \overline{B} \overline{C} + \overline{A} B \overline{C} + \overline{A} B C + A \overline{B} \overline{C} + A \overline{B} C + A B C
\]
\[
= A \overline{C} B + A \overline{C} B + A \overline{B} C + A \overline{B} C + \overline{A} B C + A B C
\]
\[
= A \overline{C} (B + \overline{B}) + A \overline{B} (C + \overline{C}) + (A + \overline{A}) B C
\]
\[
= A \overline{C} + A \overline{B} + B C
\]
- By pairing terms **differently** at the start:
\[
F = A \overline{B} C + A \overline{B} C + A \overline{B} C + A \overline{B} C + A B C + A B C + A B C
\]
- We arrive at:
\[
F = A C + \overline{A} B + B C
\]
- BOTH HAVE THE SAME LITERALS COUNTS AND NUMBER OF TERMS !!
Boolean Function Simplification

- Reducing the literal cost of a Boolean Expression leads to simpler networks.
- Simpler networks are less expensive to implement.
- Boolean Algebra can help us minimize literal cost.
- When do we stop trying to reduce the cost?
- Do we know when we have a minimum?
- We will introduce a systematic way to arrive at a minimum cost, two-level POS or SOP network.