Canonical forms

◆ Last lecture
  - Logic gates and truth tables
  - Implementing logic functions
  - CMOS switches

◆ Today’s lecture
  - deMorgan’s theorem
  - NAND and NOR
  - Canonical forms
    - Sum-of-products (minterms)
    - Product-of-sums (maxterms)

de Morgan’s theorem

◆ Replace
  - • with +, + with •, 0 with 1, and 1 with 0
  - All variables with their complements

◆ Example 1: \( Z = A'B' + A'C' \)

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

◆ Example 2: \( Z = A'B'C + A'BC + AB'C + ABC' \)

\[
Z' = (A'B'C + A'BC + AB'C + ABC')' \\
= (A'B'C)' \cdot (A'BC)' \cdot (AB'C)' \cdot (ABC')' \\
= (A+B+C)' \cdot (A+B'+C) \cdot (A'+B+C) \cdot (A'+B'+C)
\]
**NAND and NOR**

\[(X + Y)' = X' \cdot Y'\]

NOR is equivalent to AND with inputs complemented

<table>
<thead>
<tr>
<th>X</th>
<th>Y</th>
<th>X'</th>
<th>Y'</th>
<th>(X + Y)'</th>
<th>X' \cdot Y'</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

\[(X \cdot Y)' = X' + Y'\]

NAND is equivalent to OR with inputs complemented

<table>
<thead>
<tr>
<th>X</th>
<th>Y</th>
<th>X'</th>
<th>Y'</th>
<th>(X \cdot Y)'</th>
<th>X' + Y'</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

**NAND, NOR, and de Morgan’s theorem**

- **de Morgan’s**
  - Standard form: \(A'B' = (A + B)’\) \(A' + B' = (AB)’\)
  - Inverted: \(A + B = (A'B')'\) \((AB) = (A' + B')'\)
  - AND with complemented inputs \(\equiv\) NOR
  - OR with complemented inputs \(\equiv\) NAND
  - OR \(\equiv\) NAND with complemented inputs
  - AND \(\equiv\) NOR with complemented inputs

**pushing the bubble**
Converting between forms

- Introduce inversions ("bubbles")
  - Introduce bubbles in pairs
    - Conserve inversions
    - Do not alter logic function

\[ Z = AB + CD \]
\[ = (A' + B')' + (C' + D')' \]
\[ = [(A' + B')(C' + D')]' \]
\[ = [(AB)(CD)']' \]

- Example
  - AND/OR to NAND/NAND

![Diagram of AND/OR circuit to NAND/NAND conversion]

Converting between forms (con't)

- Example: AND/OR network to NOR/NOR

\[ Z = AB + CD \]
\[ = (A' + B')' + (C' + D')' \]
\[ = [(A' + B')' + (C' + D')]' \]
\[ = {[(A' + B')' + (C' + D')]'}' \]

![Diagram of AND/OR network to NOR/NOR conversion]
Converting between forms (con’t)

- Example: OR/AND to NAND/NAND

![Logic Circuit Diagram]

Converting between forms (con’t)

- Example: OR/AND to NOR/NOR

![Logic Circuit Diagram]
Why convert between forms?

- CMOS logic gates are inverting
  - Get NAND, NOR, NOT
  - Don’t get AND, OR, Buffer

Canonical forms

- Canonical forms
  - Standard forms for Boolean expressions
  - Unique algebraic signatures
  - Generally not the simplest forms
    - Can be minimized
    - Derived from truth table

- Two canonical forms
  - Sum-of-products (minterms)
  - Product-of-sum (maxterms)
Sum-of-products canonical form

- Also called disjunctive normal form
- Commonly called a **minterm expansion**

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>F</th>
<th>F'</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

\[ F = A'B'C + A'BC + AB'C + ABC' + ABC \]

\[ F' = A'B'C' + A'BC' + AB'C' \]

- Variables appears exactly once in each minterm
- In true or inverted form (but not both)

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>minterms</th>
<th>F in canonical form:</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>A'B'C' m0</td>
<td>( F(A,B,C) = \Sigma m(1,3,5,6,7) )</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>A'BC m1</td>
<td>( = m1 + m3 + m5 + m6 + m7 )</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>A'BC' m2</td>
<td>( = A'B'C' + A'BC + AB'C' + ABC' + ABC )</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>A'BC m3</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>AB'C' m4</td>
<td>canonical form ( \rightarrow ) minimal form</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>AB'C m5</td>
<td>( F(A,B,C) = A'B'C' + A'BC + AB'C + ABC' + ABC )</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>ABC' m6</td>
<td>( = (A'B' + A'B + AB' + AB)C + ABC' )</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>ABC m7</td>
<td>( = (A' + A)(B' + B)C + ABC' )</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>( = ABC' + C )</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>( = AB + C )</td>
</tr>
</tbody>
</table>

CSE370, Lecture 5

11, 12
Product-of-sums canonical form

- Also called conjunctive normal form
  - Commonly called a maxterm expansion

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

CSE370, Lecture 5

Maxterms

- Variables appears exactly once in each maxterm
  - In true or inverted form (but not both)

\[ \begin{array}{ccc|c|c}
A & B & C & \text{maxterms} & F \text{ in canonical form:} \\
0 & 0 & 0 & A+B+C & \text{M0} \\
0 & 0 & 1 & A+B+C' & \text{M1} \\
0 & 1 & 0 & A+B'+C & \text{M2} \\
0 & 1 & 1 & A+B'+C' & \text{M3} \\
1 & 0 & 0 & A'+B+C & \text{M4} \\
1 & 0 & 1 & A'+B'+C & \text{M5} \\
1 & 1 & 0 & A'+B'+C & \text{M6} \\
1 & 1 & 1 & A'+B'+C' & \text{M7} \\
\end{array} \]

- canonical form \( \rightarrow \) minimal form

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

short-hand notation

CSE370, Lecture 5
Canonical implementations of $F = AB + C$

These are not reduced forms for $F$

CSE370, Lecture 5

SOP, POS, and de Morgan's theorem

- Sum-of-products
  - $F' = A'B'C' + A'BC' + AB'C'$
  - Apply de Morgan's to get POS
    - $(F')' = (A'B'C' + A'BC' + AB'C')'$
    - $F = (A+B+C)(A+B'+C)(A'+B+C)$

- Product-of-sums
  - $F' = (A+B+C')(A+B'+C')(A'+B+C')(A'+B'+C)$
  - Apply de Morgan's to get SOP
    - $(F')' = ((A+B+C')(A+B'+C')(A'+B+C')(A'+B'+C'))'$
    - $F = A'B'C + A'BC + AB'C + ABC$
Conversion between canonical forms

- **Minterm to maxterm**
  - Use maxterms that aren’t in minterm expansion
  - \( F(A, B, C) = \sum m(1, 3, 5, 6, 7) = \prod M(0, 2, 4) \)

- **Maxterm to minterm**
  - Use minterms that aren’t in maxterm expansion
  - \( F(A, B, C) = \prod M(0, 2, 4) = \sum m(1, 3, 5, 6, 7) \)

- **Minterm of \( F \) to minterm of \( F' \)**
  - Use minterms that don’t appear
  - \( F(A, B, C) = \sum m(1, 3, 5, 6, 7) \quad F'(A, B, C) = \sum m(0, 2, 4) \)

- **Maxterm of \( F \) to maxterm of \( F' \)**
  - Use maxterms that don’t appear
  - \( F(A, B, C) = \prod M(0, 2, 4) \quad F'(A, B, C) = \prod M(1, 3, 5, 6, 7) \)