Lecture 3: 
Combinational Logic Specification and Simplification:

CSE 140: Components and Design Techniques for Digital Systems

Diba Mirza 
Dept. of Computer Science and Engineering 
University of California, San Diego
What you should know at the end of this lecture…. 

1. How to derive the Sum of Product and Product of Sum canonical forms from the truth table.

2. How to design a combinational circuit starting with the truth table.

3. How to simplify switching functions using Boolean Algebra axioms and theorems
Q: Does the following SOP canonical expression correctly express the above truth table:

\[ Y(A,B) = \Sigma m(2,3) \]

A. Yes
B. No
Sum of Product Canonical Form

<table>
<thead>
<tr>
<th>Minterm</th>
<th>A</th>
<th>B</th>
<th>Carry</th>
<th>Sum</th>
</tr>
</thead>
<tbody>
<tr>
<td>A'B'</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>A'B</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>AB'</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>AB</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

I. \[ \text{sum}(A,B) = \]

II. \[ \text{carry}(A,B) = \]
DeMorgan’s Theorem

- $Y = \overline{A \cdot B} = \overline{A} + \overline{B}$

- $Y = \overline{A + B} = \overline{A} \cdot \overline{B}$
Product of Sum Canonical Form

<table>
<thead>
<tr>
<th>Max term</th>
<th>A</th>
<th>B</th>
<th>Carry</th>
<th>Sum</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

We will derive another canonical form using the SOP expression for the compliment of the outputs: Carry and Sum

carry(A,B)=

Product of Sum Canonical Form

<table>
<thead>
<tr>
<th>Max term</th>
<th>A</th>
<th>B</th>
<th>Carry</th>
<th>Sum</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0 1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1 0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1 1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

The POS expression for sum(A,B)

A. \((A' + B) \cdot (A + B')\)
B. \(A' B + AB'\)
C. \((A+B') \cdot (A' +B)\)
D. Either A or C
PI Q: When would you use the SOP instead of the POS to express the switching function?

A. When the output of the function is TRUE for most input combinations.
B. When the output of the function is FALSE for most input combinations.
C. We always prefer the SOP form because its more compact
PI Q: When would you use the SOP instead of the POS to express the switching function?

A. When the output of the function is TRUE for most input combinations.
B. When the output of the function is FALSE for most input combinations.
C. We always prefer the SOP form because its more compact

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>Carry</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>
Re-deriving the truth table

Switching Expressions:

Sum \((A,B) = A'B + AB'\)

Carry \((A, B) = AB\)

Ex:

Sum \((0,0) = 0' .0 + 0.0' = 0 + 0 = 0\)

Sum \((0,1) = 0' 1 + 0.1' = 1 + 0 = 1\)

Sum \((1,1) = 1' 1 + 1.1' = 0 + 0 = 0\)

Logic circuit for half adder:
The SOP and POS forms don’t usually give the optimal circuit or the simplest Boolean expression of the switching function

To optimize the circuit, we need to simplify the Boolean expression using:

1. Boolean Algebra axioms and theorems
2. Karnaugh Maps (K-Maps) (next lecture)
Axioms of Boolean Algebra

1. \( B = 0, \) if \( B \) not equal to 1
2. \( 0' = 1 \)
3. \( 1.1 = 1 \)
4. \( 0.1 = 0 \)
5. \( a+0=a, \ a.1=a \) Identity law
6. \( a+a' =1, \ a.a' =0 \) Complement law
Theorems of Boolean Algebra

I. Commutative Law: \( A + B = B + A \)  
\( AB = BA \)

II. Distributive Law
\( A(B+C) = AB + AC \)

\( A+BC = (A+B)(A+C) \)
III Associativity

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

\[(AB)C = A(BC)\]
IV: Consensus Theorem: \[ AB + B' C + AC = AB + B' C \]

Venn Diagrams
PI Q: Which of the following is AC’ +BC +BA equal to?
A. AB+C’A
B. AC’+CB
C. BC+AB
D. None of the above
Proof of consensus Theorem using Boolean Algebra

\[ AB + B'C + AC = AB + B' C \]
V. DeMorgan’s Theorem

- \( Y = \overline{AB} = \overline{A} + \overline{B} \)

- \( Y = \overline{A + B} = \overline{A} \cdot \overline{B} \)
Circuit Transformation: Bubble Pushing

- Pushing bubbles backward (from the output) or forward (from the inputs) changes the body of the gate from AND to OR or vice versa.
- Pushing a bubble from the output back to the inputs puts bubbles on all gate inputs.
- Pushing bubbles on all gate inputs forward toward the output puts a bubble on the output and changes the gate body.
Example of transforming circuits using bubble pushing