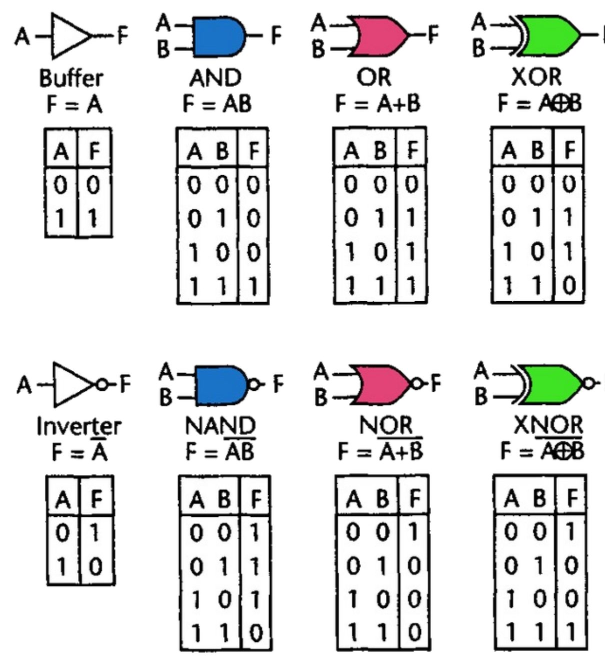
EE2000 Logic Circuit Design

Recap Lecture 1 – Logic Function and Boolean Algebra



Instructables.com

Outline

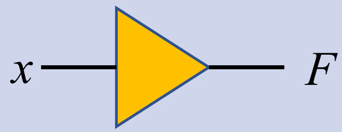
1.1 Basic Logic Gates

1.2 Logic Circuit and Boolean Expression

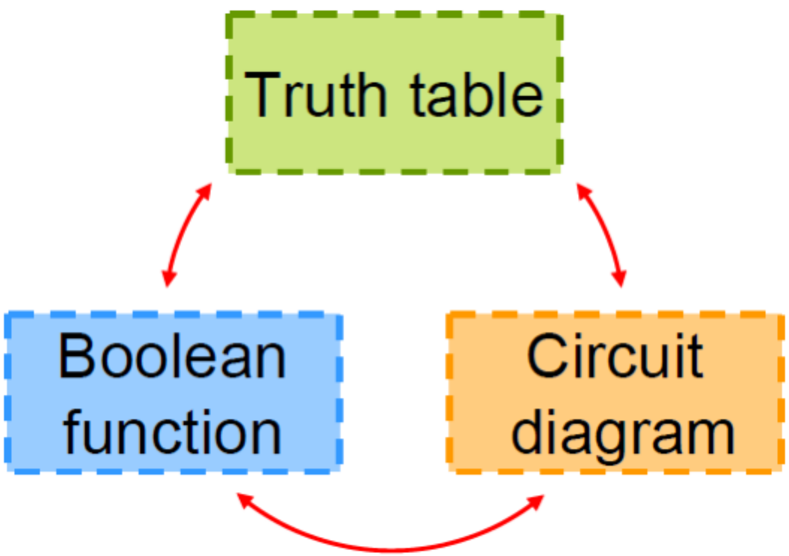
1.3 Sum of Products vs Product of Sums and Canonical Form

1.4 Simplification using Boolean Algebra

Logic Gate

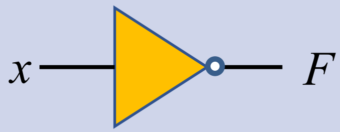
<table>
<tr><th>Logic Name</th><th>Circuit Diagram</th><th>Truth Table</th><th>Boolean Function</th></tr>
<tr>
<td>Buffer: Output (F) follows the same logic state as the Input (x)</td>
<td>
</td>
<td><table><tr><th>x</th><th>F</th></tr><tr><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td></tr></table></td>
<td>$F(x) = x$
$F = x$</td>
</tr>
</table>

$$F(x,y) = x \cdot y + \overline{x \cdot y}$$

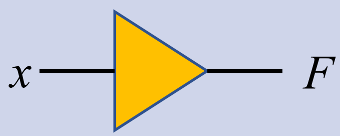


- Function (F): Operation
- Variable: Inputs (x, y)
- Complement: Inversion ($\bar{x}$, $\bar{y}$)
- Literal: Each appearance of a variable or its complement (4)
- Product term: One or more literals connected by $\cdot$ operator (2)

NOT Gate

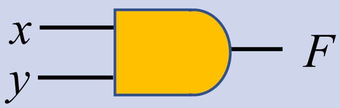
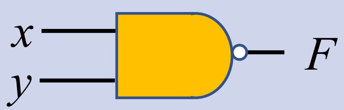
<table>
<tr><th>Logic Name</th><th>Circuit Diagram</th><th>Truth Table</th><th>Boolean Function</th></tr>
<tr><td>NOT gate (Inverter): Output (F) has opposite logic state of the Input (x)</td><td></td><td><table><tr><th>x</th><th>F</th></tr><tr><td>0</td><td>1</td></tr><tr><td>1</td><td>0</td></tr></table></td><td>$F = \bar{x}$

$F = x'$</td></tr>
</table>

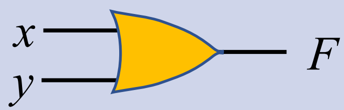
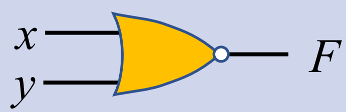
<table>
<tr><th>Logic Name</th><th>Circuit Diagram</th><th>Truth Table</th><th>Boolean Function</th></tr>
<tr><td>Buffer: Output (F) follows the same logic state as the Input (x)</td><td></td><td><table><tr><th>x</th><th>F</th></tr><tr><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td></tr></table></td><td>$F(x) = x$

$F = x$</td></tr>
</table>

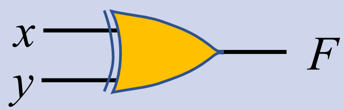
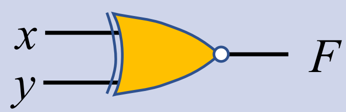
AND & NAND Gates

<table>
<tr><th>Logic Name</th><th>Circuit Diagram</th><th>Truth Table</th><th>Boolean Function</th></tr>
<tr>
<td>AND gate: Output (F) is 1 only when all inputs are 1</td>
<td>
</td>
<td><table>
<tr><th>x</th><th>y</th><th>F</th></tr>
<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>1</td><td>1</td></tr>
</table></td>
<td>$F = x \cdot y$
$F = xy$</td>
</tr>
<tr>
<td>NAND gate: Output (F) is 0 only when all inputs are 1</td>
<td>
</td>
<td><table>
<tr><th>x</th><th>y</th><th>F</th></tr>
<tr><td>0</td><td>0</td><td>1</td></tr>
<tr><td>0</td><td>1</td><td>1</td></tr>
<tr><td>1</td><td>0</td><td>1</td></tr>
<tr><td>1</td><td>1</td><td>0</td></tr>
</table></td>
<td>$F = \overline{x \cdot y}$
$F = \overline{xy}$</td>
</tr>
</table>

OR & NOR Gates

<table>
<tr><th>Logic Name</th><th>Circuit Diagram</th><th>Truth Table</th><th>Boolean Function</th></tr>
<tr>
<td>OR gate: Output (F) is 1 when either or all inputs are 1</td>
<td>
</td>
<td><table><tr><th>x</th><th>y</th><th>F</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table></td>
<td>$F = x + y$</td>
</tr>
<tr>
<td>NOR gate: Output (F) is 1 only when all inputs are 0</td>
<td>
</td>
<td><table><tr><th>x</th><th>y</th><th>F</th></tr><tr><td>0</td><td>0</td><td>1</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>1</td><td>0</td></tr></table></td>
<td>$F = \overline{x + y}$</td>
</tr>
</table>

XOR & XNOR Gates

<table>
<tr><th>Logic Name</th><th>Circuit Diagram</th><th>Truth Table</th><th>Boolean Function</th></tr>
<tr>
<td>XOR gate: Output (F) is 1 only when one of the input is 1</td>
<td>
</td>
<td><table><tr><th>x</th><th>y</th><th>F</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></table></td>
<td>$F = x \oplus y$</td>
</tr>
<tr>
<td>XNOR gate: Output (F) is 1 only when all inputs are 0 or 1</td>
<td>
</td>
<td><table><tr><th>x</th><th>y</th><th>F</th></tr><tr><td>0</td><td>0</td><td>1</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>1</td><td>1</td></tr></table></td>
<td>$F = \overline{x \oplus y}$
$F = x \otimes y$</td>
</tr>
</table>

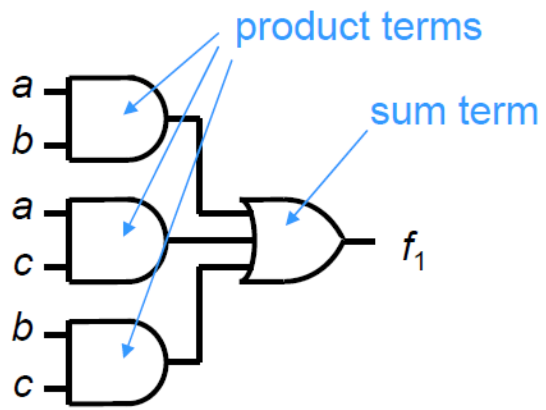
1.3 Sum of Products vs Product of Sums, Canonical Form

- Product terms: One or more literals connected by AND operator.

$$x, xy', xyz', y'z'$$

- Standard product terms: A product term that includes each variable of the problem, either uncomplemented or complemented.

$$xyz, xy'z', x'yz', x'y'z'$$



$f_1(a, b, c) = ab + ac + bc$

- Sum of Products (SOP): A group of AND gates followed by a single OR gate.

$xy + yz + x'z + z'$	4 product terms
$xy + yz + x'z$	3 product terms
$xy + yz$	2 product terms
xy	1 product term

Canonical (Standard) Form in SOP

Inputs			Minterms		Output
x	y	z	Term	Designation	$f(x, y, z)$
0	0	0	$x'y'z'$	m_0	1
0	0	1	$x'y'z$	m_1	0
0	1	0	$x'yz'$	m_2	1
0	1	1	$x'yz$	m_3	0
1	0	0	$xy'z'$	m_4	1
1	0	1	$xy'z$	m_5	1
1	1	0	xyz'	m_6	0
1	1	1	xyz	m_7	0

- Minterms: Standard product terms. Uncomplement = 1; Complement = 0.
- Canonical Sum (Sum of standard product terms): Sum of products expression with minterms only when output is 1

$$f(x, y, z) = x'y'z' + x'yz'$$

$$+xy'z' + xy'z$$

$$= m_0 + m_2 + m_4 + m_5$$

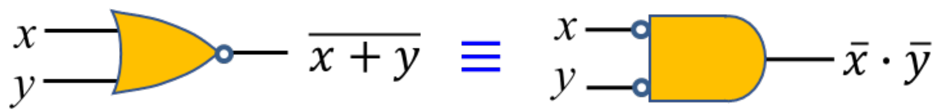
$$= \sum m(0, 2, 4, 5)$$

DeMorgan's Theorem

$$(x + y)' = x'y'$$

Complement of Sum = Product of Complement

x	y	$\overline{x+y}$	$\bar{x} \cdot \bar{y}$
0	0	1	1
0	1	0	0
1	0	0	0
1	1	0	0



$$(xy)' = x' + y'$$

Complement of Product = Sum of Complement

x	y	$\overline{xy}$	$\bar{x} + \bar{y}$
0	0	1	1
0	1	1	1
1	0	1	1
1	1	0	0

Common mistake! $(xy)' \neq x'y'$ **and** $(x + y)' \neq x' + y'$

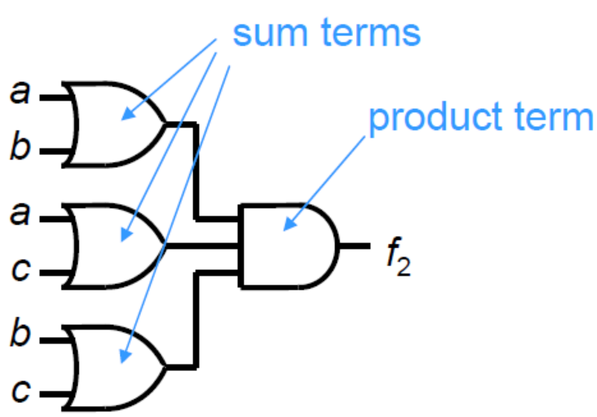
Product of Sums

- Sum terms: One or more literals connected by OR operator.

$$x, x + y', y + z', x + y + z'$$

- Standard sum terms: A sum term that includes each variable of the problem, either uncomplemented or complemented.

$$x + y + z, x + y' + z', x' + y + z', x' + y' + z'$$



$f_2(a, b, c) = (a + b)(a + c)(b + c)$

- Product of Sums (POS): A group of OR gates followed by a single AND gate.

$(w + z)(w' + y')xy$	4 sum terms
$(w + z)(w' + y')x$	3 sum terms
$(w + z)(w' + y')$	2 sum terms
$(w + z)$	1 sum term

Canonical (Standard) Form in POS

Inputs			Maxterms		Output
x	y	z	*Term*	*Designation*	$f(x, y, z)$
0	0	0	$x+y+z$	M_0	1
0	0	1	$x+y+z'$	M_1	1
0	1	0	$x+y'+z$	M_2	0
0	1	1	$x+y'+z'$	M_3	1
1	0	0	$x'+y+z$	M_4	0
1	0	1	$x'+y+z'$	M_5	1
1	1	0	$x'+y'+z$	M_6	1
1	1	1	$x'+y'+z'$	M_7	0

- Maxterms: Standard sum terms. Uncomplement = 0; Complement = 1.
- Canonical Product (Product of standard sum terms): Product of sums expression with maxterms only when output is 0

$$f(x,y,z)$$
$$= (x+y'+z)(x'+y+z)$$
$$(x'+y'+z')$$
$$= M_2M_4M_7 = \prod M(2,4,7)$$

Summary

- $\overline{m_i} = M_i$ and $\overline{M_i} = m_i$

$$\overline{m_0} = (x'y'z')' = x + y + z = M_0$$

- If a f is in SOP form, its complement is in POS form (*vice versa*).

$$f = xyz + xy'z$$

$$f' = (x' + y' + z')(x' + y + z')$$

- Canonical SOP (all minterms with output 1)

$$f = xyz + xy'z = \sum m(5,7) \quad \text{and} \quad f' = \sum m(0,1,2,3,4,6)$$

- Canonical POS (all maxterm with output 0)

$$f = \prod M(0,1,2,3,4,6) \qquad\qquad \text{and} \quad f' = \prod M(5,7)$$

Summary (Given in Test and Exam)

Commutative	$a + b = b + a$	$ab = ba$
Associative	$a + (b + c) = (a + b) + c$	$a(bc) = (ab)c$
Identity	$a + 0 = a$	$a(1) = a$
Null	$a + 1 = 1$	$a(0) = 0$
Complement	$a + a' = 1$	$a(a') = 0$
Idempotency	$a + a = a$	$a(a) = a$
Involution	$(a')' = a$	
Distributive	$a(b + c) = ab + ac$	$a + bc = (a + b)(a + c)$
Adjacency	$ab + ab' = a$	$(a + b)(a + b') = a$
Simplification	$a + a'b = a + b$	$a(a' + b) = ab$
DeMorgan	$(a + b)' = a'b'$	$(ab)' = a' + b'$
Absorption	$a + ab = a$	$a(a + b) = a$
Consensus	$ab + a'c + bc = ab + a'c$	

Exercise

1. Express the Canonical Sum and Product based on the Truth Table provided.
2. Simplify the Function in SOP form.
3. Design the logic circuit using
   NAND and NOT gates.

Inputs			*Output*
x	y	z	$f(x, y, z)$
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1