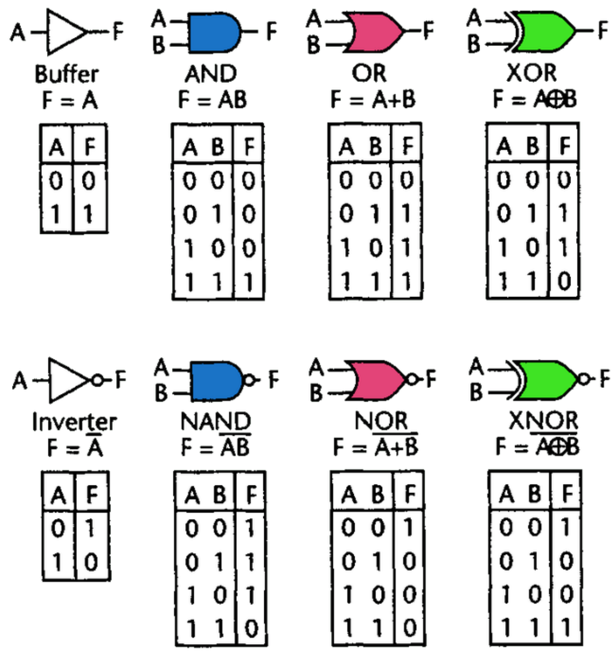
EE2000 Logic Circuit Design

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

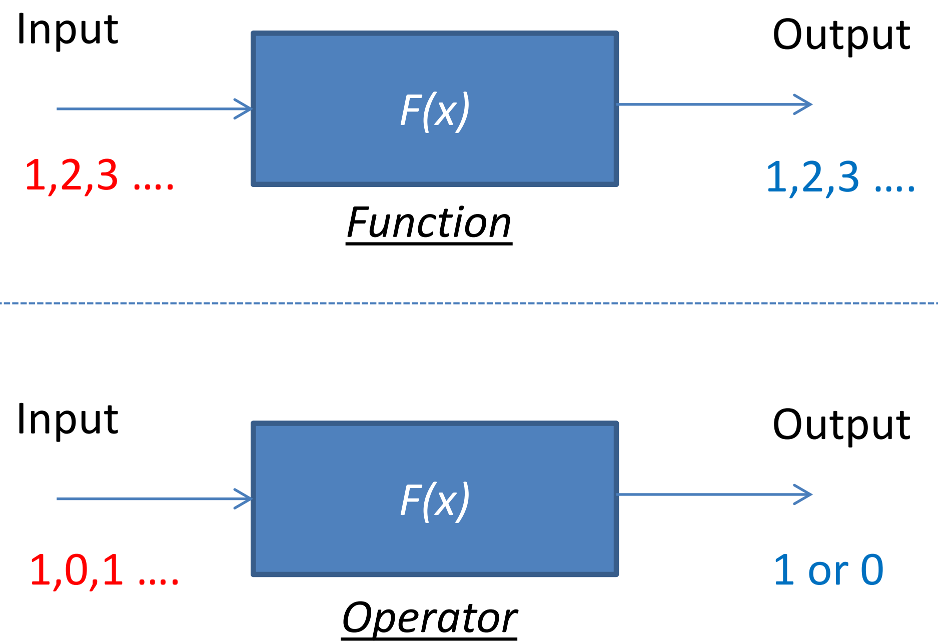
1.4 Simplification using Boolean Algebra

1.1 Basic Logic Gates

Gate describes a circuit that performs a basic logic operation.

Conventional numerical / mathematical function

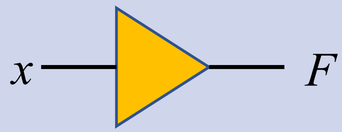
Logic operator/ function



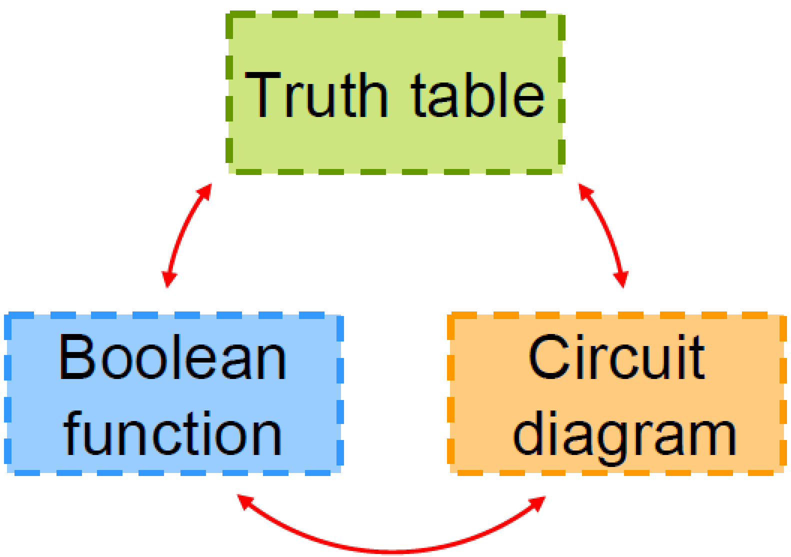
- Basic building blocks
- Perform certain function or logic
- One or more inputs
- One output

Binary decision output e.g, Yes/No ; True/False and 1/0

Logic Gate

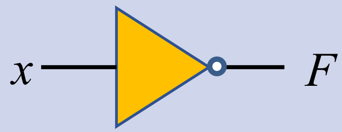
<table>
<tr><th>Logic Name</th><th>Circuit Diagram</th><th>Truth Table</th><th>Boolean Function</th></tr>
<tr><td>Buffer: Output (F) follows the same logic state as the Input (x)</td><td></td><td><table><tr><th>x</th><th>F</th></tr><tr><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td></tr></table></td>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

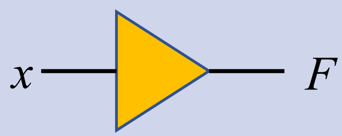
$$F(x, y) = x \cdot y + \overline{x \cdot y}$$



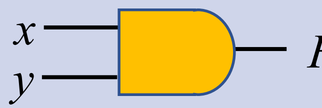
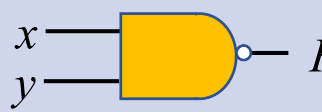
- Function (F): Operation
- Variable: Inputs (x, y)
- Complement: Inversion ($\bar{x}$, $\bar{y}$)
- Literal: Each appearance of a variable or its complement (4)
- Product term: One or more literals connected by · operator (2)

NOT Gate

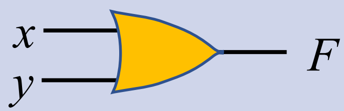
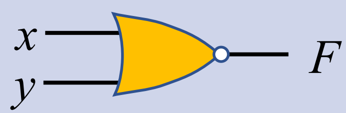
<table>
<tr><th>Logic Name</th><th>Circuit Diagram</th><th>Truth Table</th><th>Boolean Function</th></tr>
<tr><td>NOT gate (Inverter): Output (F) has opposite logic state of the Input (x)</td><td>
</td><td><table><tr><th>x</th><th>F</th></tr><tr><td>0</td><td>1</td></tr><tr><td>1</td><td>0</td></tr>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

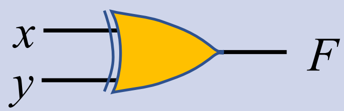
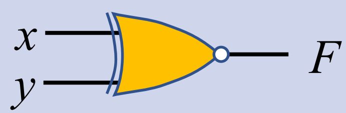
AND & NAND Gates

<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

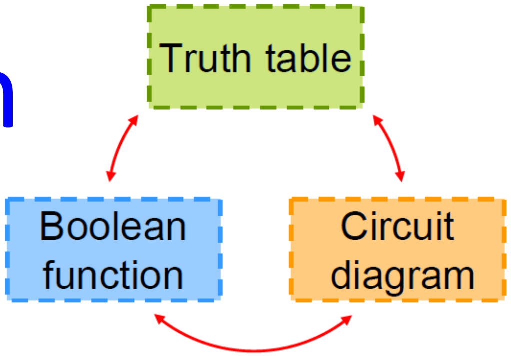
<table>
<tr><th>Logic Name</th><th>Circuit Diagram</th><th>Truth Table</th><th>Boolean Function</th></tr>
<tr><td>OR gate: Output (F) is 1 when either or all inputs are 1</td><td>
</td><td><table><tr><th>x</th><th>y</th><th>F</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</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

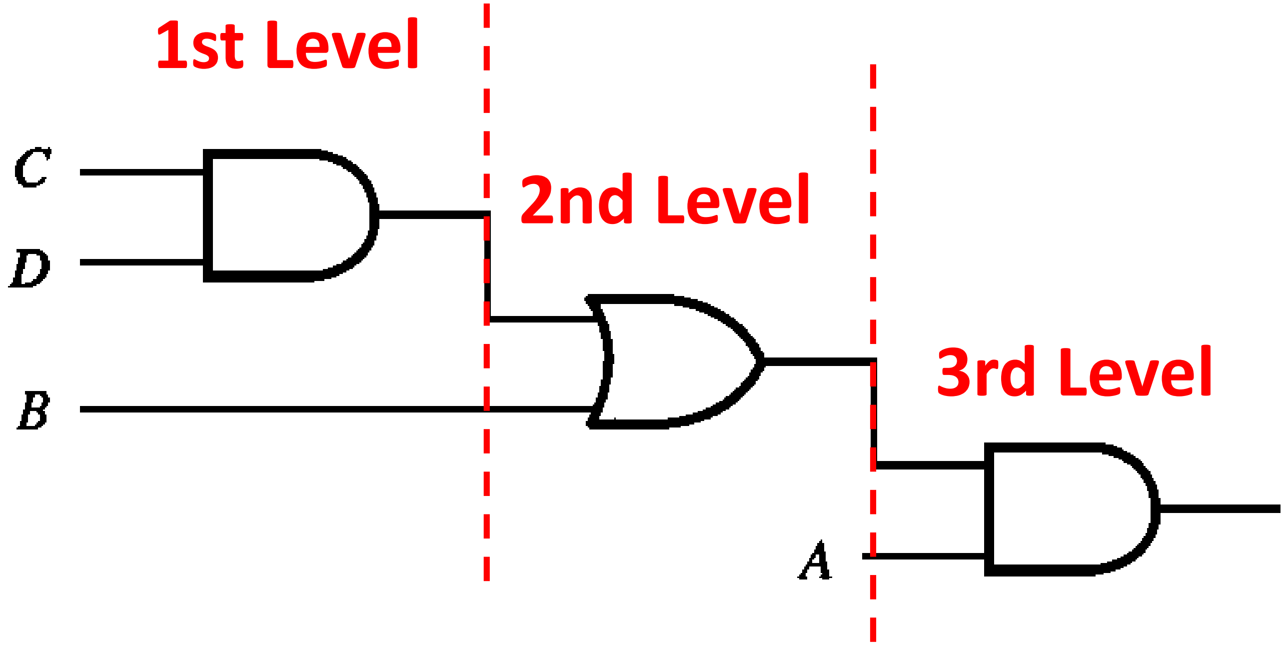
<table>
<tr><th>Logic Name</th><th>Circuit Diagram</th><th>Truth Table</th><th>Boolean Function</th></tr>
<tr><td>XOR gate: Output (F) is 1 only when one of the input is 1</td><td>
</td><td><table><tr><th>x</th><th>y</th><th>F</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</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>

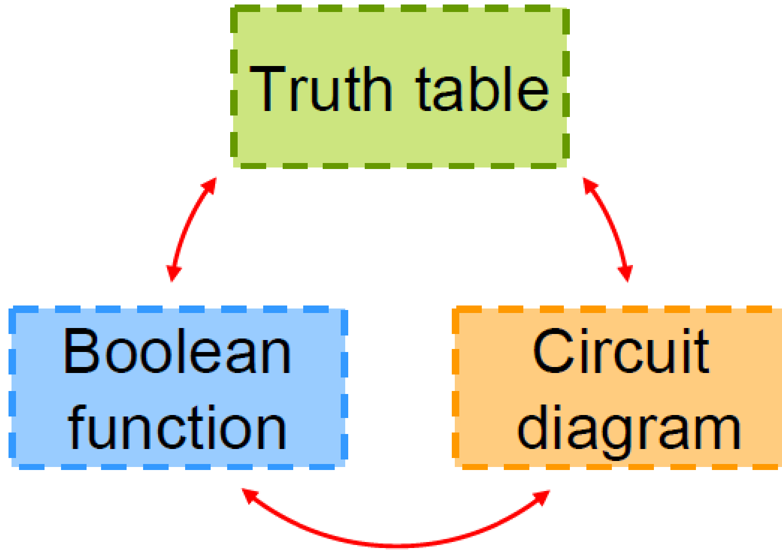
1.2 Logic Circuit and Boolean Expression

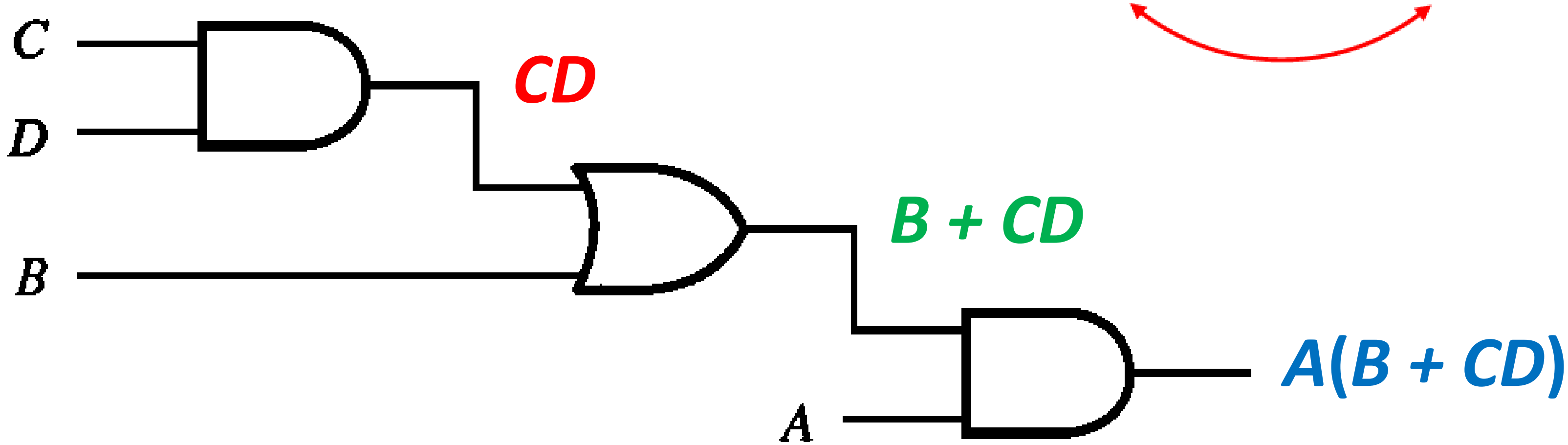


To derive the Boolean expression of a given logic circuit, begin at the left-most inputs and work towards the final output, writing the expression for each gate.



Boolean Expression

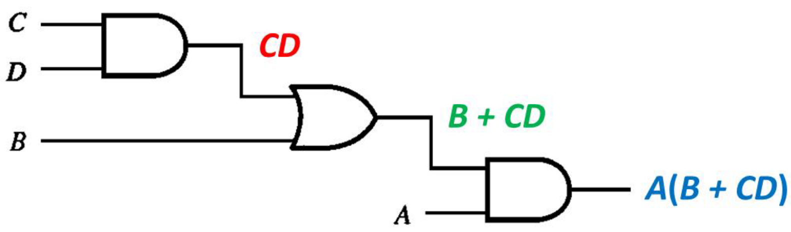




$$F = A(B + CD)$$

NEXT: Construct a truth table for above logic circuit

Truth Table



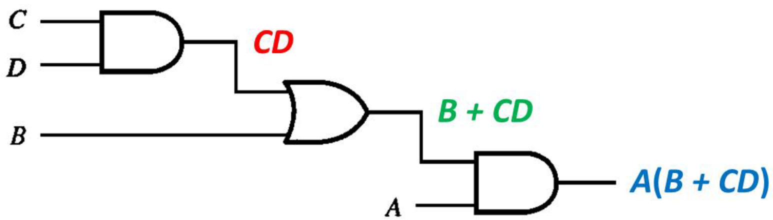
$$F = A(B + CD)$$

For the truth table, find the output following the steps:

1. Write down all input possibility
2. Write down the stage output (i.e. *CD*, *B + CD*)
3. Write down the final stage output (i.e. *A(B + CD)*)

Inputs				*Operation*		*Output*
A	*B*	*C*	*D*	*CD*	*B+CD*	*A(B+CD)*
0	0	0	0			
0	0	0	1			
0	0	1	0			
0	0	1	1			
0	1	0	0			
0	1	0	1			
0	1	1	0			
0	1	1	1			
1	0	0	0			
1	0	0	1			
1	0	1	0			
1	0	1	1			
1	1	0	0			
1	1	0	1			
1	1	1	0			
1	1	1	1			

Complete Solution



$$F = A(B + CD)$$

Examples of numbering systems		Inputs				Output
Decimal	Hexadecimal	A	B	C	D	A(B+CD)
0	0	0	0	0	0	0
1	1	0	0	0	1	0
2	2	0	0	1	0	0
3	3	0	0	1	1	0
4	4	0	1	0	0	0
5	5	0	1	0	1	0
6	6	0	1	1	0	0
7	7	0	1	1	1	0
8	8	1	0	0	0	0
9	9	1	0	0	1	0
10	A	1	0	1	0	0
11	B	1	0	1	1	1
12	C	1	1	0	0	1
13	D	1	1	0	1	1
14	E	1	1	1	0	1
15	F	1	1	1	1	1

From Truth Table

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

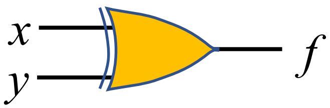
$f(x, y, z) = 1$
if $(x = 0 \text{ AND } y = 0 \text{ AND } z = 1)$
OR
$(x = 1 \text{ AND } y = 0 \text{ AND } z = 0)$
OR
$(x = 1 \text{ AND } y = 0 \text{ AND } z = 1)$

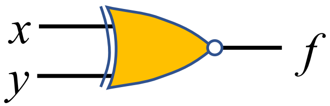
$$f(x, y, z) = x'y'z + xy'z' + xy'z$$

Questions:

1. How many literals?
2. How many product terms?

Exercises





x	y	f
0	0	0
1	0	1
0	1	1
1	1	0

x	y	f
0	0	1
1	0	0
0	1	0
1	1	1

Given the Truth Tables of XOR and XNOR gates, write down the Boolean Function (using sum and product) and draw the Logic Circuit using NOT, AND and OR gates.

Exercises

Given the Boolean function $f(x, y, z) = xy' + x'z'$, draw the Logic Circuit and work out the truth table.

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

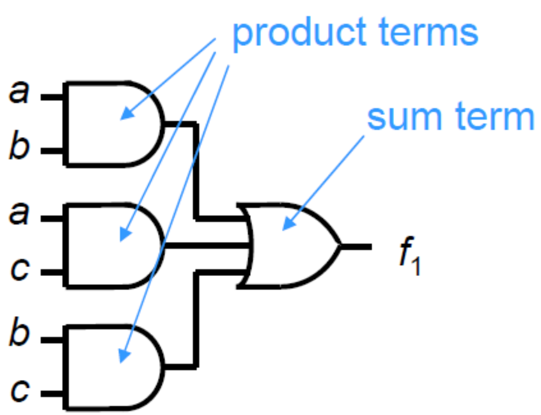
1.3 Sum of Products vs Product of Sums, Canonical Form

- 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

$$f(x, y, z) = x'y'z' + x'yz'$$
$$+xy'z' + xy'z$$
$$= m_0 + m_2 + m_4 + m_5$$
$$= \sum m(0,2,4,5)$$

Question

Which function(s) is formed by minterm?

A) $f = a + b + c$

B) $f = ab + bc + ac$

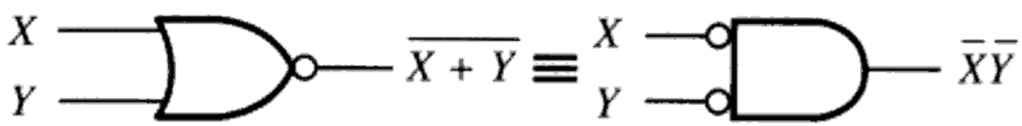
C) $f = abc + ab'c'$

D) $f = ab + ab'c'$

DeMorgan's Theorem

$$(a+b)' = a'b'$$

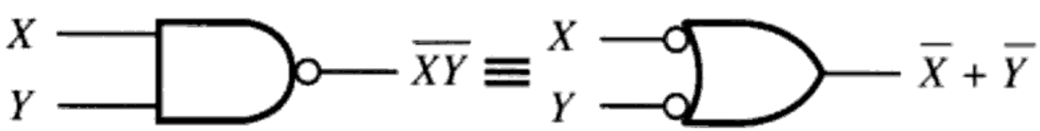
Complement of Sum is equal to Product of Complement



X	Y	$\overline{X+Y}$	$\bar{X}\bar{Y}$
0	0	1	1
0	1	0	0
1	0	0	0
1	1	0	0

$$(ab)' = a' + b'$$

Product of Complement is equal to 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! $(ab)' \neq a'b'$ **and** $(a+b)' \neq a'+b'$

Example

$$(a + b)' = a'b'$$

$$(ab)' = a' + b'$$

$$f = wx'y + xy' + wxz$$

$$f' = (wx'y + xy' + wxz)'$$
$$= (wx'y)'(xy')'(wxz)'$$
$$= (w' + x + y')(x' + y)(w' + x' + z')$$

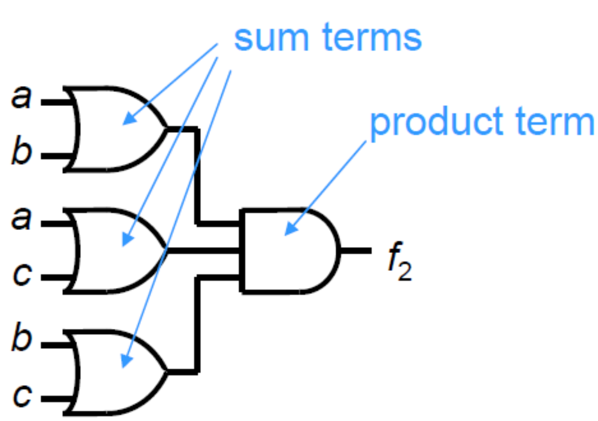
Product of Sums

- 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

SOP to POS

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

$(a+b)' = a'b'$ $\quad$ $(ab)' = a' + b'$

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

$$= m_0 + m_1 + m_3 + m_5 + m_6$$

$$= \sum m(0,1,3,5,6)$$

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

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

$$= (x'yz')'(xy'z')'(xyz)'$$

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

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

$$f(x, y, z)$$
$$= (x + y' + z)(x' + y + z)(x' + y' + z')$$
$$= M_2 M_4 M_7 = \prod M(2,4,7)$$

Thought (SOP or POS?)

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

SOP & POS

Inputs			Minterms		Output	Maxterms		Output
x	y	z	Term	Designation	$f(x, y, z)$	*Term*	*Designation*	$f'(x, y, z)$
0	0	0	$x'y'z'$	m_0	0	$x+y+z$	M_0	1
0	0	1	$x'y'z$	m_1	0	$x+y+z'$	M_1	1
0	1	0	$x'yz'$	m_2	0	$x+y'+z$	M_2	1
0	1	1	$x'yz$	m_3	1	$x+y'+z'$	M_3	0
1	0	0	$xy'z'$	m_4	0	$x'+y+z$	M_4	1
1	0	1	$xy'z$	m_5	1	$x'+y+z'$	M_5	0
1	1	0	xyz'	m_6	0	$x'+y'+z$	M_6	1
1	1	1	xyz	m_7	0	$x'+y'+z'$	M_7	1

$$f(x,y,z) = x'yz + xy'z = m_3 + m_5 = \sum m(3,5) = \prod M(0,1,2,4,6,7)$$

$$f'(x,y,z) = (x+y'+z')(x'+y+z') = \prod M(3,5) = \sum m(0,1,2,4,6,7)$$

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)

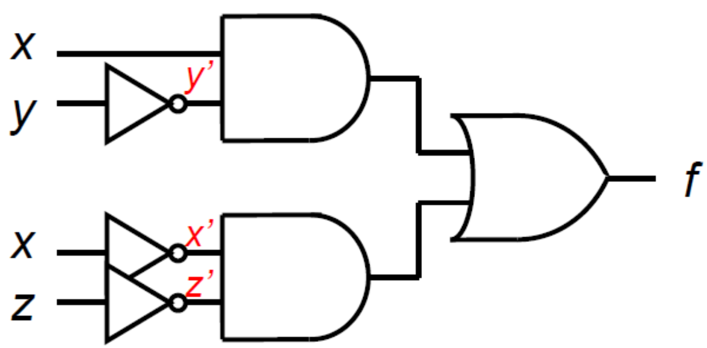
$$f = xyz + xy'z = \sum m(5,7) \quad \text{and} \quad f' = \sum m(0,1,2,3,4,6)$$

- Canonical POS (all maxterm with output 0)

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

Recap

Given the Boolean function $f(x, y, z) = xy' + x'z'$,
draw the Logic Circuit and work out the truth table.



x	y	z	xy'	x'z'	xy' + x'z'
0	0	0	0	1	1
0	0	1	0	0	0
0	1	0	0	1	1
0	1	1	0	0	0
1	0	0	1	0	1
1	0	1	1	0	1
1	1	0	0	0	0
1	1	1	0	0	0

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

1.4 Simplification using Boolean Algebra

- Obtain a simple (or simplest) logic circuit
- Reduce the cost of circuit

Cost in logic circuit

- Gate cost (number of gates in the implementation)

- Gate-input cost (number of inputs to the gates)

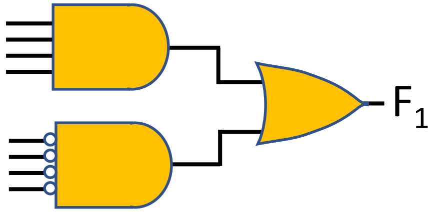
- Total cost = Gate cost + Gate-input cost

Gate-input cost

- The number of gate-input is proportional to the number of transistors in the logic circuit
- The cost can be determined by checking logic diagram / schematic and the Boolean function

$F_1 = abcd + a'b'c'd'$

$F_2 = (a' + b)(b' + c)(c' + d)(d' + a)$

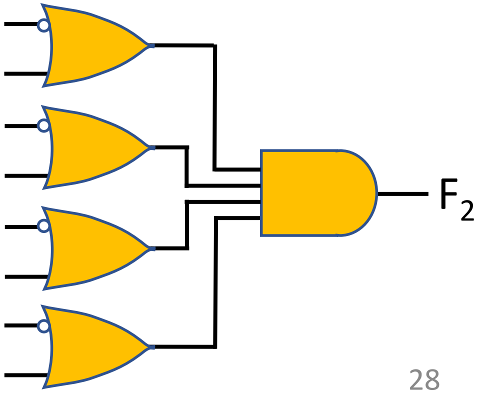


F_1 has 3 no. of gate and 10 no. of gate-input

Total cost = 13

F_2 has 5 no. of gate and 12 no. of gate-input

Total cost = 17



How to reduce cost?

The key of simplifying logic functions is to reduce the no. of terms and no. of literals

$\downarrow$no. of literals = $\downarrow$ no. of gate inputs
$\downarrow$ no. of terms = $\downarrow$ no. of gates

In addition, costs in space and power consumption can be reduced.

Simplification

$$f(a,b,c) = a'bc' + a'bc + ab'c' + ab'c + abc$$

5 product terms, 15 literals

$$f(a,b,c) = a'b + ab' + abc$$

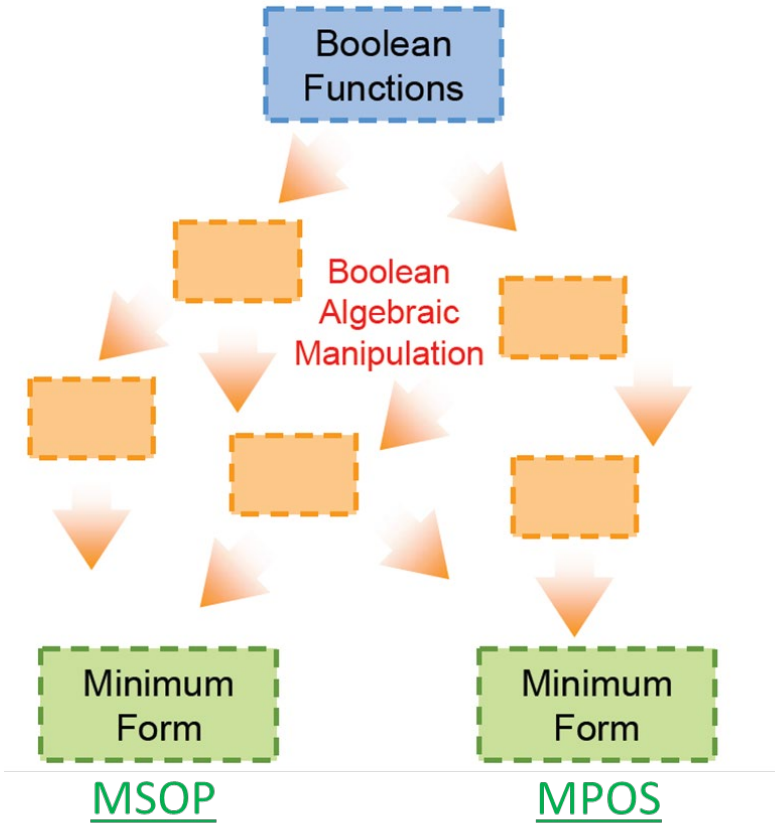
3 product terms, 7 literals

$$f(a,b,c) = a'b + ab' + ac$$

3 product terms, 6 literals

$$f(a,b,c) = a'b + ab' + bc$$

3 product terms, 6 literals



Boolean or Switching Algebra

- A system of mathematical logic to perform different operations in binary system with elements of $S = \{0, 1\}$
- {0 ,1} represents a light off or on, a switch up or down, a low or high voltage, *etc.*
- The formulation is referred as Switching or Boolean Function

  e.g. If $x, y \in S$,

$$f(x, y) = x + y$$

Basic Postulates

If $\boldsymbol{x}, \boldsymbol{y} \in \boldsymbol{S}$,

Commutative	$x + y = y + x$	$xy = yx$
Associative	$x + (y + z)$ $= (x + y) + z$	$x(yz) = (xy)z$
Identity	$x + 0 = x$	$x \cdot 1 = x$
Null	$x + 1 = 1$	$x \cdot 0 = 0$
Complement	$x + x' = 1$	$x \cdot x' = 0$
Idempotency	$x + x = x$	$x \cdot x = x$
Involution	$(x')' = x$	

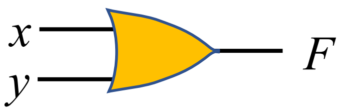
Duality Principle

Commutative	$x + y = y + x$	$xy = yx$
Associative	$x + (y + z)$ $= (x + y) + z$	$x(yz) = (xy)z$
Identity	$x + 0 = x$	$x \cdot 1 = x$
Null	$x + 1 = 1$	$x \cdot 0 = 0$
Complement	$x + x' = 1$	$x \cdot x' = 0$
Idempotency	$x + x = x$	$x \cdot x = x$

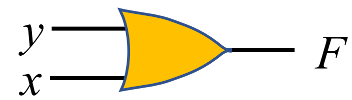
Boolean algebra remains true when the operators OR and AND is interchanged and the identity elements 0 and 1 are interchanged.

Commutative

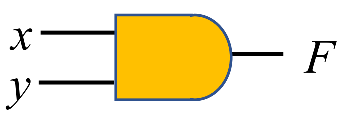
$x + y = y + x$



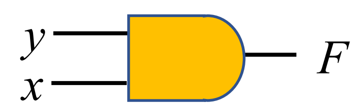
is equivalent to



$xy = yx$

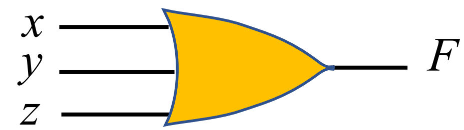


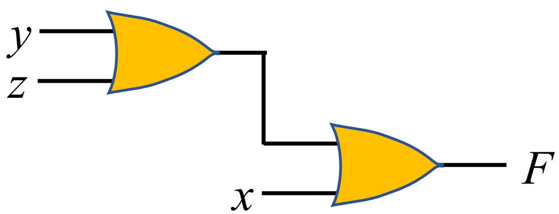
is equivalent to



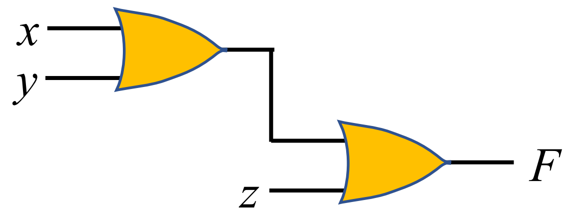
Associative

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

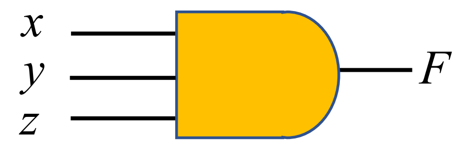


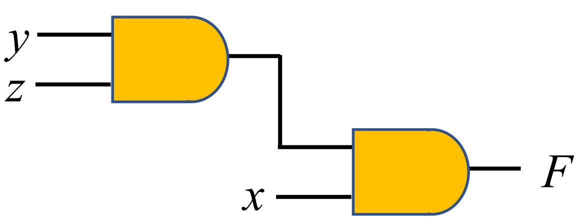


is equivalent to

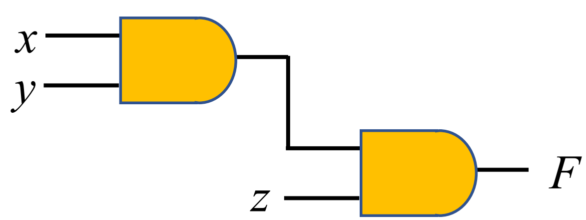


$$x(yz) = (xy)z$$





is equivalent to



Distributive

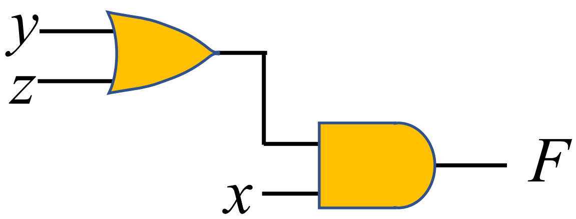
$$x(y + z) = xy + xz$$

$$\boxed{x + yz = (x + y)(x + z)}$$

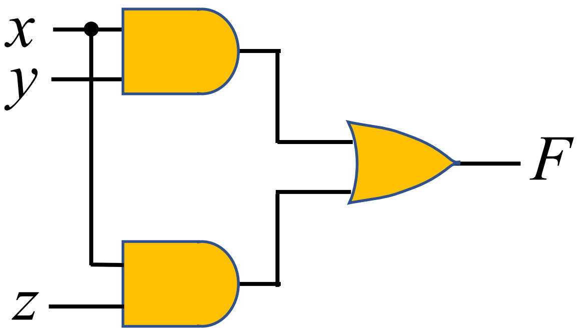
Inputs			$x + yz$			$(x + y)(x + z)$		
x	y	z	x	yz	f	$x + y$	$x + z$	f
0	0	0	0	0	0	0	0	0
0	0	1	0	0	0	0	1	0
0	1	0	0	0	0	1	0	0
0	1	1	0	1	1	1	1	1
1	0	0	1	0	1	1	1	1
1	0	1	1	0	1	1	1	1
1	1	0	1	0	1	1	1	1
1	1	1	1	1	1	1	1	1

Distributive

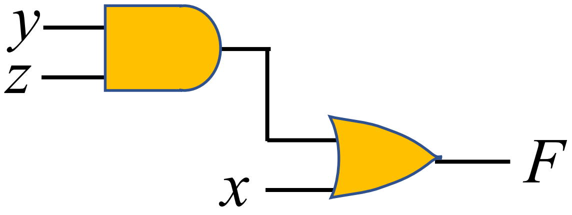
$$x(y + z) = xy + xz$$



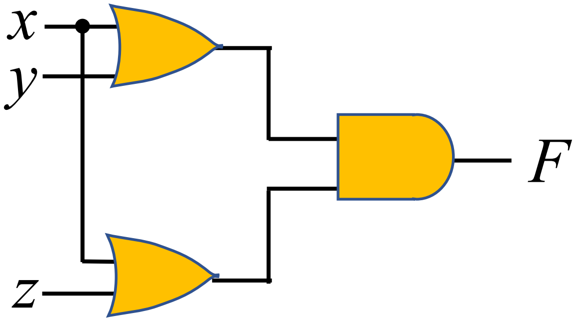
is equivalent to



$$x + yz = (x + y)(x + z)$$



is equivalent to



Question

- Which of the following has the same function as $x + x'y$?

    a) $x + xy'$

    b) $x + y$

    c) $x' + y$

    d) y

Simplification

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

$$= 1 \cdot (x + y)$$ complement

$$= x + y$$ identity

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

$$= 0 + xy$$ complement

$$= xy$$ identity

Adjacency

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

$$= x \cdot 1$$ complement

$$= x$$ identity

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

$$= xx + xy + xy' + yy'$$

complement

$$= x + x(y + y') + 0$$

$$= x + x(1) = x$$ Identity and idempotency

Exercise

- Simplify the following expression.

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

Exercise

- Simplify the following expression.

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

Adsorption

$$x + xy = x \cdot 1 + xy$$ identity

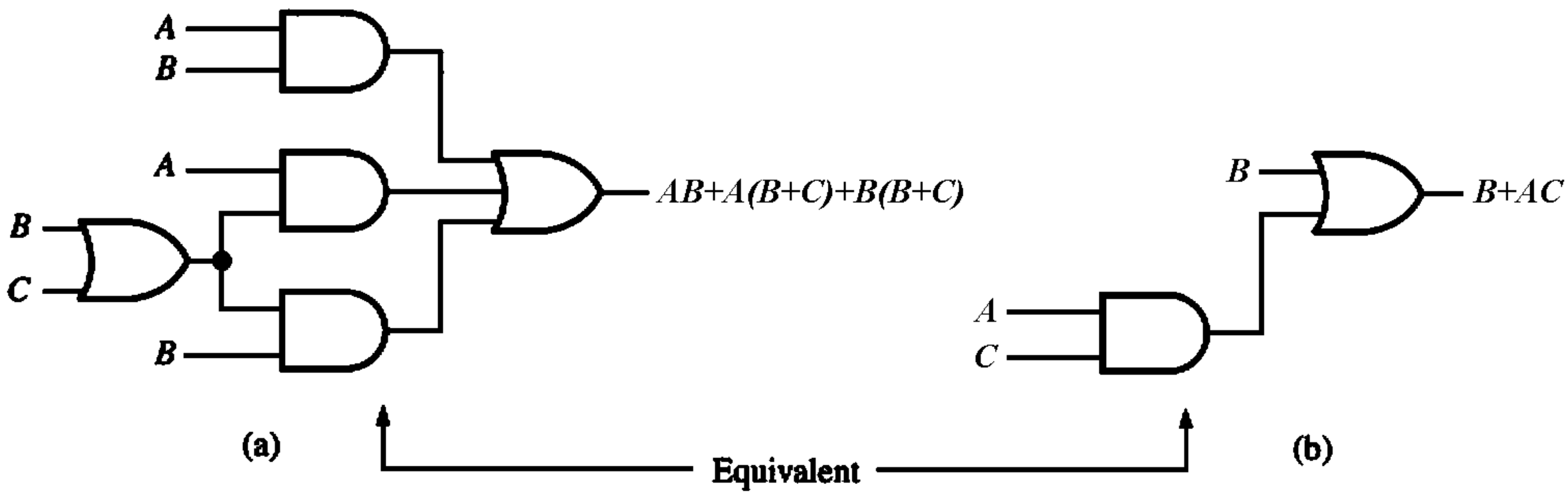
$$= x(1 + y)$$ distributive

$$= x$$ Null

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

Exercise

Prove that the above Circuit (a) is equivalent to Circuit (b).



Summary

Distributive	$x(y+z) = xy + xz$	$x + yz = (x+y)(x+z)$
Simplification	$x + x'y = x + y$	$x(x'+y) = xy$
Adjacency	$xy + xy' = x$	$(x+y)(x+y') = x$
Adsorption	$x + xy = x$	$x(x+y) = x$

Consensus

$$xy + x'z + yz = xy + x'z + yz(x + x')$$ complement

$$= xy + xyz + x'z + x'yz$$ distributive

$$= xy + x'z$$ adsorption

Consensus term: For any two product terms where **exactly** one variable appears uncomplemented in one and complemented in the other, the Consensus term is the product of the remaining literals. **The consensus term could be eliminated.**

Term 1	Term 2	Consensus Term
$xy'z$	wx'	$wy'z$
wxy'	xyz'	wxz'
wxy'	$xy'z$	$-$
$xy'z$	$wx'y$	$-$

Question

Which of the following can be simplified using consensus theorem?

a) $xy'z + wx' + wy'z' + wy$

b) $wxy' + xyz' + w'xz' + yz$

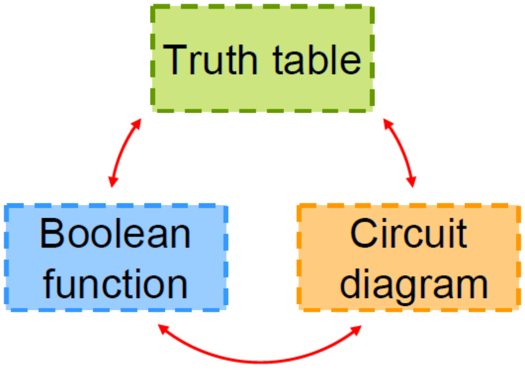
c) $wx' + xy'z + w'yz' + x'yz'$

Exercise

Simplify the following function.

$$f(w,x,y,z) = wxy' + w'y'z + wx'y' + xy'z + w'z$$

Conclusion



1.1 Basic Logic Gates

- Circuit diagram, Boolean function, Truth table

1.2 Logic Circuit and Boolean Expression

- Work out complete solution

1.3 Sum of Products vs Product of Sums and Canonical Form

- Express function in Canonical SOP or POS

1.4 Simplification using Boolean Algebra

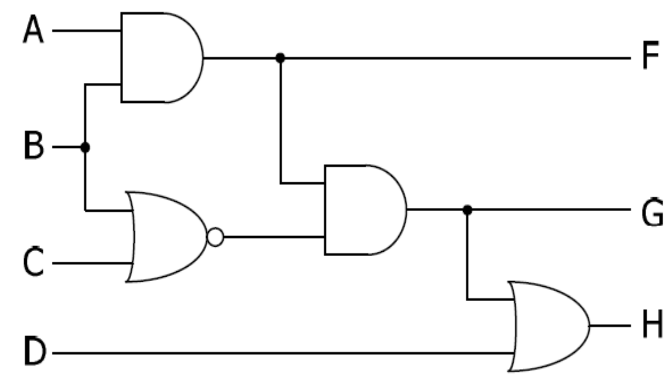
- Simplify the function for simpler circuit diagram

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	$ab + ac + bc = ab + ac$	

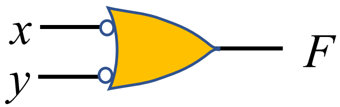
Exercise

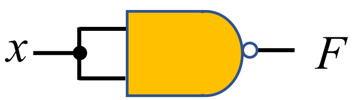
1. Derive the Boolean functions to describe the operations of the logic circuit as shown.
2. Simplify the functions and draw the circuit.

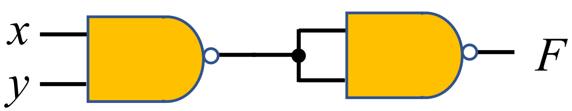


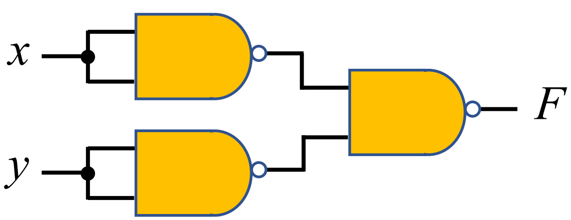
Exercise

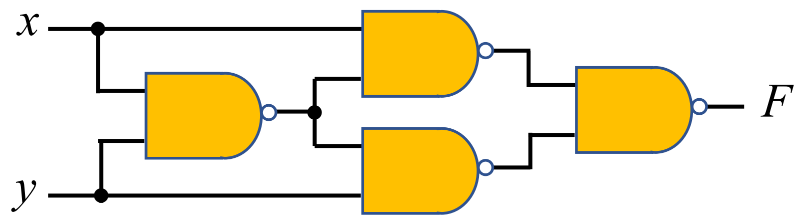
Work out the Boolean functions of the following circuits. Which standard logic gate does each of them represent?









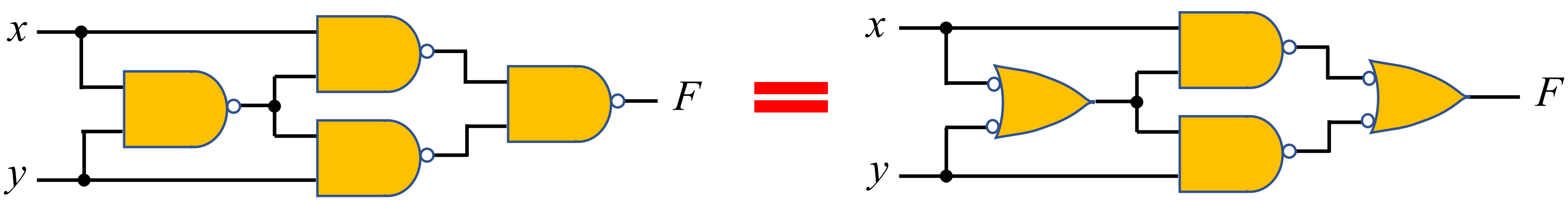


Exercise

Work out the Boolean functions of the following circuits. Which standard logic gate does each of them represent?

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

$$\begin{aligned} F &= x(x' + y') + y(x' + y') \\ &= xy' + yx' = x \oplus y \end{aligned}$$



- ➢ Any Boolean function can be implemented using NAND gates (Functional completeness)
- ➢ NAND gates are used for SOP function
- ➢ Same for NOR gate but for POS function

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

Next Lecture

Simplify $f(a, b, c, d) = \Sigma m(0, 1, 2, 5, 8, 9, 10)$ in POS form

cd \ ab	00	01	11	10
00	1	0	0	1
01	1	1	0	1
11	0	0	0	0
10	1	0	0	1

Fill the 1s and 0s into the map

cd \ ab	00	01	11	10
00	1	0	0	1
01	1	1	0	1
11	0	0	0	0
10	1	0	0	1

Group the 0s using the same procedure as grouping the 1s

$f'(a, b, c, d) = ab + cd + bd'$

$f(a, b, c, d) = (a'+b')(c'+d')(b'+d)$