

EE2000 Logic Circuit Design

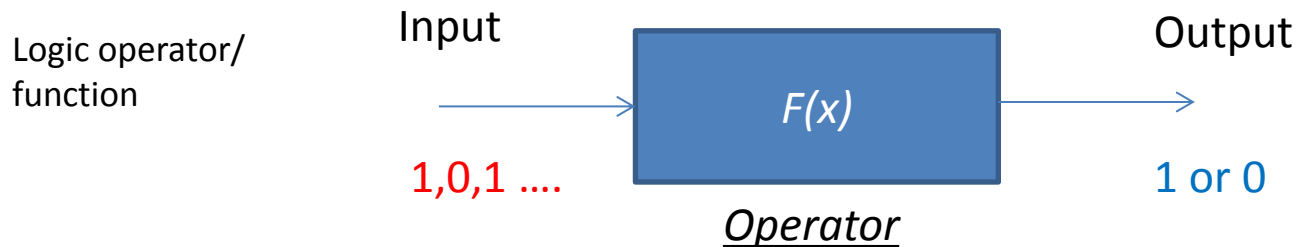
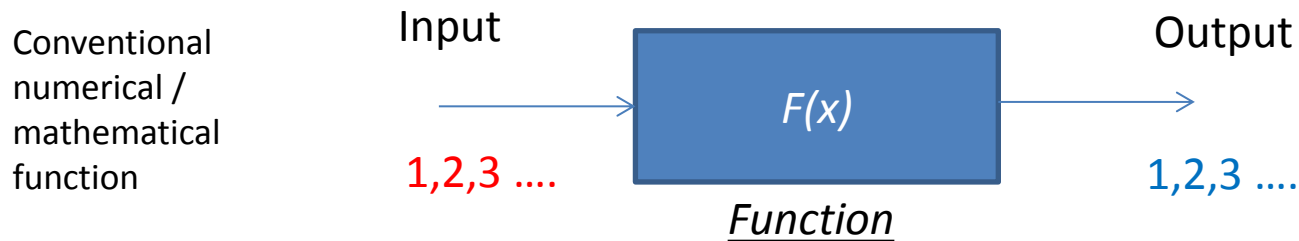
Boolean Algebra and Logic Function

Outline

- 2.1 Basic logic gates
- 2.2 Boolean algebra
- 2.3 Logic Circuit and Boolean Expression
- 2.4 Simplification using Boolean Algebra
 - SOP, POS, minterm, maxterm, canonical form

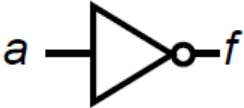
2.1 Logic Gate

- The term **gate** describes a circuit that performs a basic logic operation.



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

Logic Operator

	OR	AND	NOT
Binary / Unary operator?	Binary	Binary	Unary
Symbols	1: + 2: $\vee$	1: $\cdot$ 2: $\wedge$ 3: absence of an operator	1: ' (prime) 2: $\sim$ 3: $\bar{}$ (bar)
Examples	1: $a + b$ 2: $a \vee b$	1: $x \cdot y$ 2: $x \wedge y$ 3: xy	1: a' 2: $\sim a$ 3: $\bar{a}$
Logic Gate Symbol			

a	b	$a + b$
0	0	0
0	1	1
1	0	1
1	1	1

Input Output

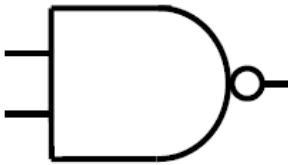
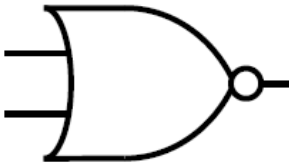
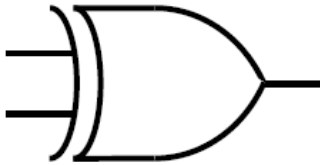
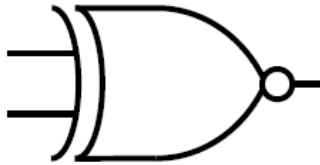
x	y	$x \cdot y$
0	0	0
0	1	0
1	0	0
1	1	1

Input Output

a	a'
0	1
1	0

Input Output

Logic Operator

Operation		<i>NAND</i>	<i>NOR</i>	<i>XOR</i>	<i>XNOR</i>
<i>a</i>	<i>b</i>	$(ab)'$	$(a+b)'$	$a \oplus b$	$a \otimes b$
0	0	1	1	0	1
0	1	1	0	1	0
1	0	1	0	1	0
1	1	0	0	0	1
Symbol					

2.2 Boolean Algebra

- A set of element S with at least two different elements (x, y) satisfying binary operations $(+)$ and $(\cdot)$.
- For Boolean algebra in which $S = \{0,1\}$, the formulation is referred as switching function.

Basic Postulates

If $x, y \in S$,

$$x + y = y + x$$

$$x \cdot y = y \cdot x$$

commutative

If $x, y, z \in S$,

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

$$x \cdot (y \cdot z) = (x \cdot y) \cdot z$$

associative

If $x, y, z \in S$,

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

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

distributive

Distributive Law

- Proof $x + (y \cdot z) = (x + y) \cdot (x + z)$

x	y	z	$y \cdot z$	$x + y \cdot z$	$x + y$	$x + z$	$(x + y)(x + z)$
0	0	0	0	0	0	0	0
0	0	1	0	0	0	1	0
0	1	0	0	0	1	0	0
0	1	1	1	1	1	1	1
1	0	0	0	1	1	1	1
1	0	1	0	1	1	1	1
1	1	0	0	1	1	1	1
1	1	1	1	1	1	1	1

Identical

Duality

- If an expression is valid in Boolean algebra, the dual of the expression is also valid.

- Principle of duality:

$$0 \cdot x = 0$$

$$1 + x = 1$$

$$1 \cdot x = x$$

$$0 + x = x$$

$$x \cdot x = x$$

$$x + x = x$$

$$x \cdot x' = 0$$

$$x + x' = 1$$

The expressions are interchangeable by replacing “0” by “1” and “+” by “ $\cdot$ ”.

Theorems

Idempotent

$$x + x = x$$

$$x \cdot x = x$$

Involution

$$(x')' = x$$

Absorption

$$x + xy = x$$

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

Logical adjacency

$$xy + xy' = x$$

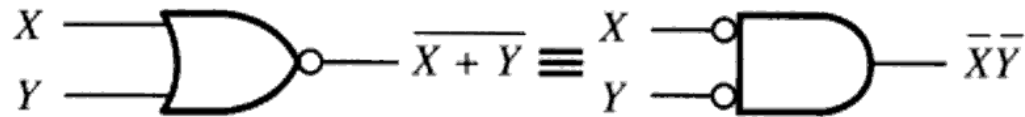
DeMorgan

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

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

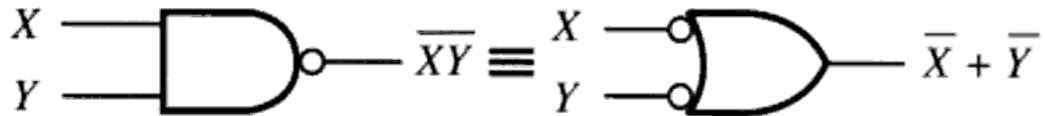
- The complement of sum is equal to the product of the complement
- The product of complement is equal to the sum of the complement

DeMorgan



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

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



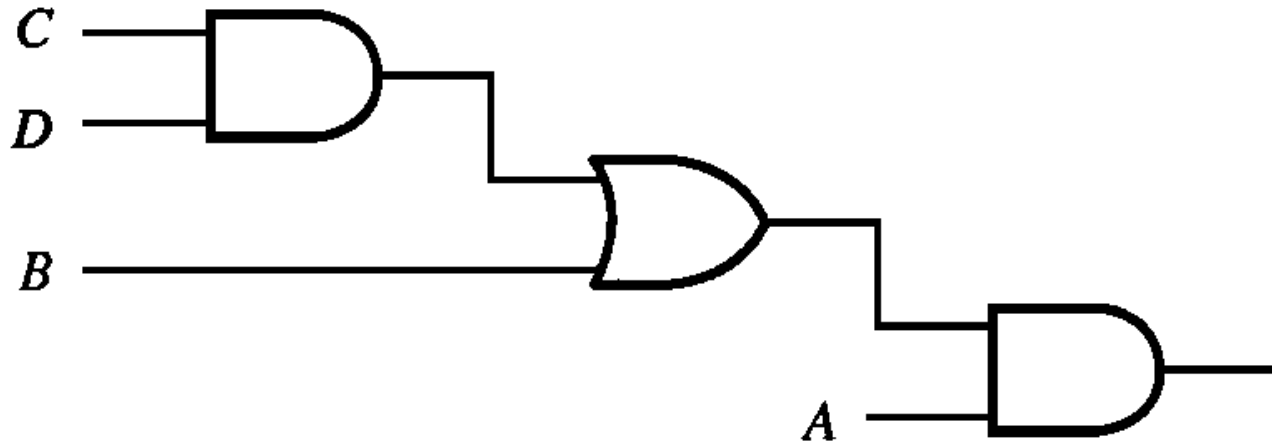
$$(xy)' = x' + y'$$

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

Consensus

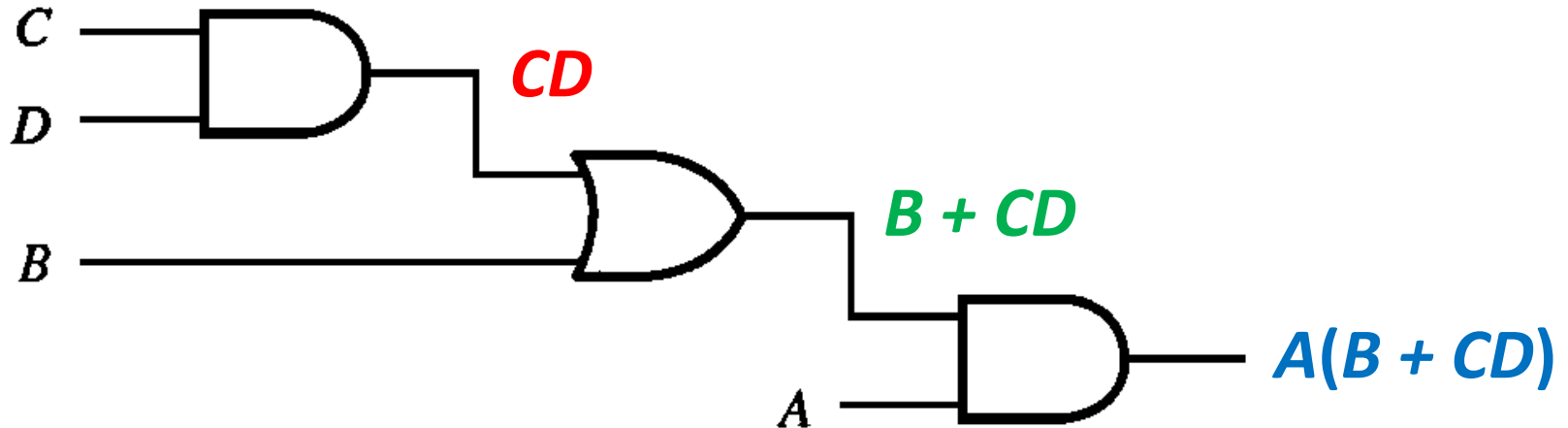
- $at_1 + a't_2 + t_1t_2 = at_1 + a't_2$
- $(a + t_1)(a' + t_2)(t_1 + t_2) = (a + t_1)(a' + t_2)$
- The theorem shows that the consensus term, t_1t_2 , is redundant and can be eliminated
- Proof: $at_1 + a't_2 + t_1t_2$
 - $= at_1 + a't_2 + t_1t_2 \cdot 1$ (identity)
 - $= at_1 + a't_2 + t_1t_2 \cdot (a + a')$ (complementation)
 - $= at_1 + a't_2 + at_1t_2 + a't_1t_2$ (distributivity)
 - $= at_1 + a't_2$ (absorption)

2.3 Logic Circuit and Boolean Expression



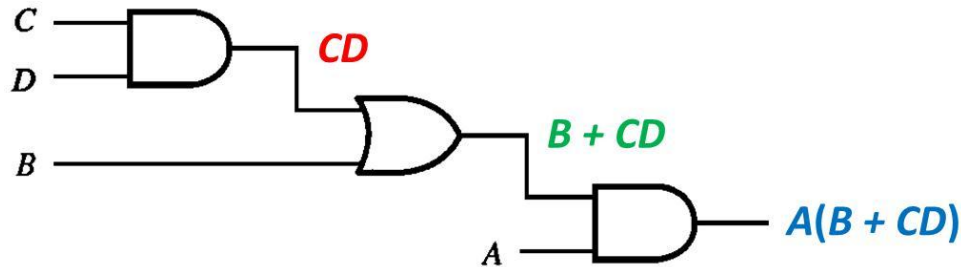
- To derive the Boolean expression of a given logic circuit, begin at the left-most inputs and work towards the final output, writing the expression for each gate.

Boolean expression from a logic circuit



- Write down the output expression from all logic operators
- The Boolean function of this circuit is $A (B + CD)$
- Construct a truth table for above logic circuit

Truth table for a logic circuit



Completed solution for a logic circuit design must include:

1. Boolean Algebra
2. Circuit schematic
3. Truth table
4. Table Assignment

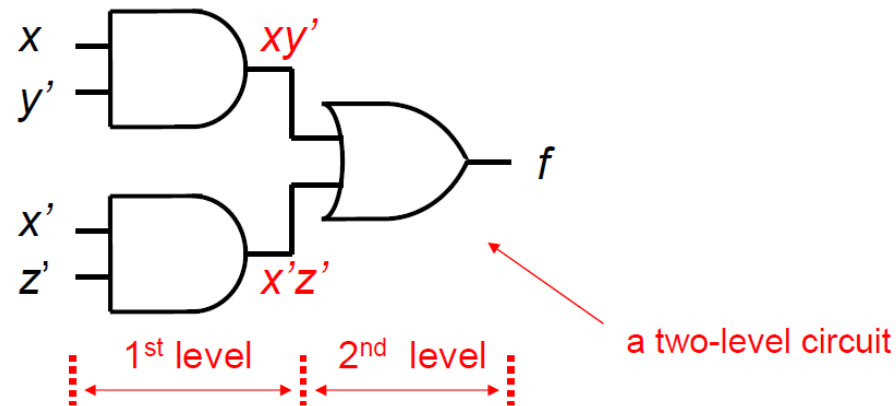
For the truth table, find the output as a following:

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. $A(B + CD)$)

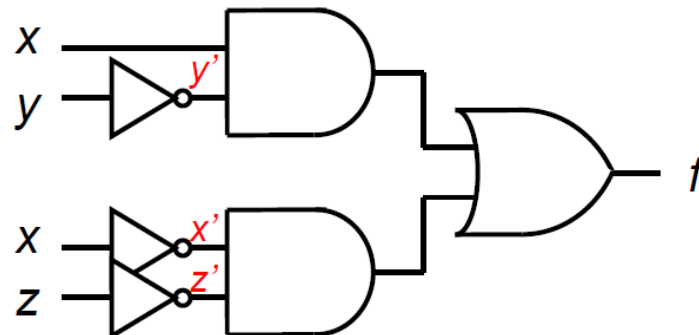
Examples of numbering systems		Inputs				Output
Decimal	Hexadecimal	A	B	C	D	$A(B+CD)$
0	0	0	0	0	0	0
1	1	0	0	0	1	0
2	2	0	0	1	0	0
3	3	0	0	1	1	0
4	4	0	1	0	0	0
5	5	0	1	0	1	0
6	6	0	1	1	0	0
7	7	0	1	1	1	0
8	8	1	0	0	0	0
9	9	1	0	0	1	0
10	A	1	0	1	0	0
11	B	1	0	1	1	1
12	C	1	1	0	0	1
13	D	1	1	0	1	1
14	E	1	1	1	0	1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

Logic circuit from a Boolean expression

Provided that a Boolean function $f(x,y,z)=xy'+x'z'$, then the logic circuit can be formed as:



or



Boolean function $\rightarrow$ Truth Table

■ e.g. $f(x, y, z) = xy' + x'z'$

The number of rows is 2^n (n is the number of variables)

Input(s)			Output		
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

Truth Table → Boolean function

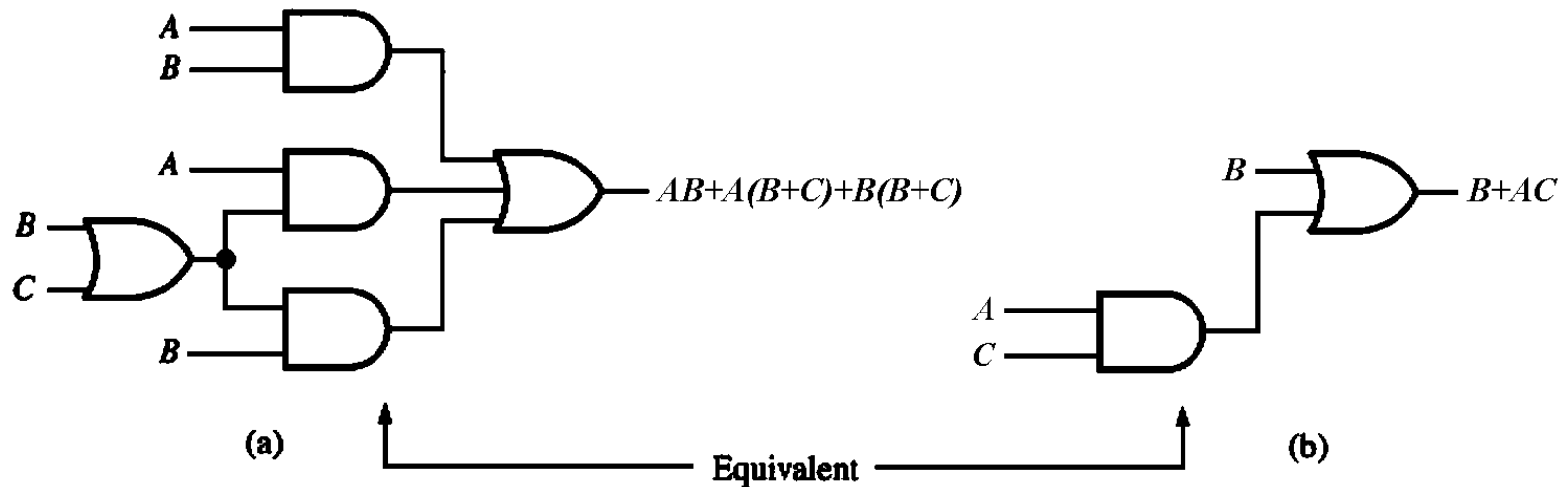
Inputs			Output
<i>a</i>	<i>b</i>	<i>c</i>	<i>f</i>
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	0

f is 1 if $(a = 0 \text{ AND } b = 0 \text{ AND } c = 1) \text{ OR}$
 $(a = 1 \text{ AND } b = 0 \text{ AND } c = 0) \text{ OR}$
 $(a = 1 \text{ AND } b = 0 \text{ AND } c = 1)$

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

Is it the simplest form?

2.4 Simplification using Boolean Algebra



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

Solution by Boolean Algebra Simplification

$$AB + A(B + C) + B(B + C)$$

$$AB + AB + AC + BB + BC$$

$$AB + AB + AC + B + BC$$

$$AB + AC + B + BC$$

$$AB + AC + B$$

$$B + AC$$

$$BB=B$$

$$AB+AB=AB$$

$$B+BC=B$$

$$AB+B=B$$

Idempotent

Idempotent

Absorption

Absorption

Boolean Algebra Simplification

Example 1

$$\begin{aligned}\text{Simplify } f &= \overline{AB} + \overline{AC} + \overline{A} \overline{B} C \\&= (\overline{AB})(\overline{AC}) + \overline{A} \overline{B} C \\&= (\overline{A} + \overline{B})(\overline{A} + \overline{C}) + \overline{A} \overline{B} C \\&= \overline{A} \overline{A} + \overline{A} \overline{C} + \overline{A} \overline{B} + \overline{B} \overline{C} + \overline{A} \overline{B} C \\&= \overline{A} + \overline{A} \overline{C} + \overline{A} \overline{B} + \overline{B} \overline{C} \\&= \overline{A} + \overline{A} \overline{B} + \overline{B} \overline{C}\end{aligned}$$

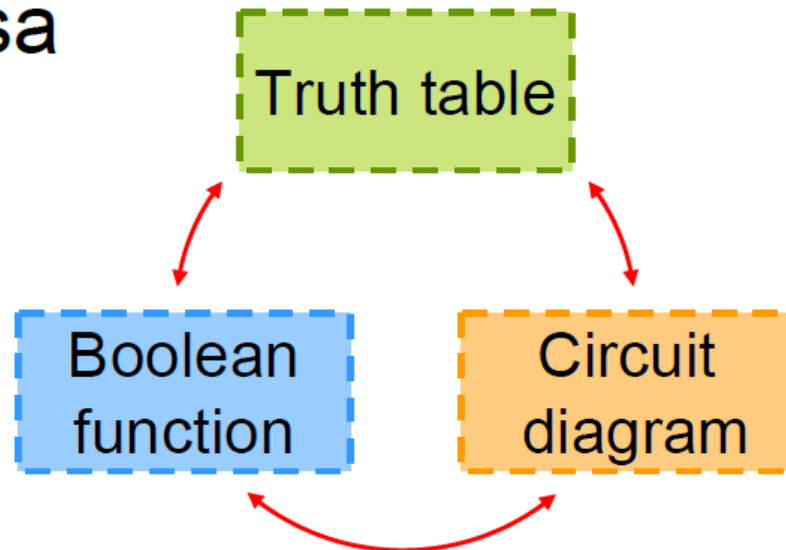
Example 2

$$\begin{aligned}\text{Simplify } f &= \overline{(A + B) \overline{C} \overline{D} + E + \overline{F}} \\&= [\overline{(A + B) \overline{C} \overline{D}}][\overline{E + \overline{F}}] \\&= [\overline{(A + B)} + \overline{\overline{C} \overline{D}}][\overline{E} \overline{\overline{F}}] \\&= (\overline{A} \overline{B} + \overline{\overline{C}} + \overline{\overline{D}}) \overline{E} F \\&= (\overline{A} \overline{B} + C + D) \overline{E} F\end{aligned}$$

- Please write the properties of switching algebra for every steps

Relationship

- Relation between Boolean function, truth table and logic circuit diagram
- A Boolean function can be represented by truth table and logic circuit diagram, and vice versa



Minterm and Maxterm

[**Minterm**] For a function of n variables, if a product term contains each of the n variables **exactly one time** in complemented or uncomplemented form, the product term is called *minterm*. Complement = 0 and Uncomplement = 1.

Function	Minterm	Not minterm	Not minterm
$f(A, B, C)$	$A' B' C$	$(A B)' C$	$A' B'$

If the function is represented as a sum of minterms only, the function is in *standard sum of product (SOP)* form.

$$f(A, B, C) = \overline{A} \overline{B} \overline{C} + A \overline{B} \overline{C} + \overline{A} B \overline{C} + A B \overline{C}$$

Minterm	Code	Number
$A' B C'$	010	m_2
$A B C'$	110	m_6
$A' B C$	011	m_3
$A B C$	111	m_7

$$f(A, B, C) = m_2 + m_3 + m_6 + m_7$$

Canonical form $f(A, B, C) = \sum m(2, 3, 6, 7) = \sum (2, 3, 6, 7)$

Minterm and Maxterm

[**Maxterm**] If a sum term of a function of n variables contains each of the n variables **exactly one time** in complemented or uncomplemented form, the sum term is called a *maxterm*. Complement = 1 and Uncomplement = 0.

Function	Maxterm	Not maxterm	Not maxterm
$f(A, B, C)$	$A' + B' + C$	$(A + B)' C$	$A' + B'$

If a function is represented as a product of maxterms only, the function is in *standard product of sum (POS)* form.

$$f(A, B, C) = (A + B + C)(A + B + \bar{C})(\bar{A} + B + C)(\bar{A} + B + \bar{C})$$

Minterm	Code	Number
$A + B + C$	000	M_0
$A + B + C'$	001	M_1
$A' + B + C$	100	M_4
$A' + B + C'$	101	M_5

$$f(A, B, C) = M_0 M_1 M_4 M_5$$

Canonical form $f(A, B, C) = \prod M(0, 1, 4, 5) = \prod (0, 1, 4, 5)$

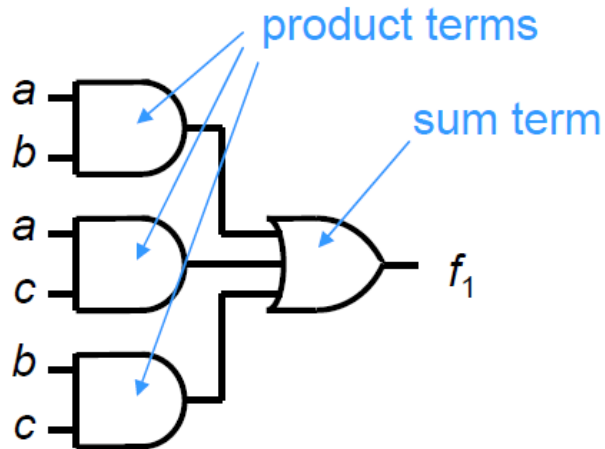
Description for minterms and maxterms for 3 variables logic function

x	y	z	Minterms		Maxterms	
			Term	designation	term	designation
0	0	0	$x'y'z'$	m_0	$x + y + z$	M_0
0	0	1	$x'y'z$	m_1	$x + y + z'$	M_1
0	1	0	$x'yz'$	m_2	$x + y' + z$	M_2
0	1	1	$x'yz$	m_3	$x + y' + z'$	M_3
1	0	0	$xy'z'$	m_4	$x' + y + z$	M_4
1	0	1	$xy'z$	m_5	$x' + y + z'$	M_5
1	1	0	xyz'	m_6	$x' + y' + z$	M_6
1	1	1	xyz	m_7	$x' + y' + z'$	M_7

SOP & POS Implementation

Sum of products

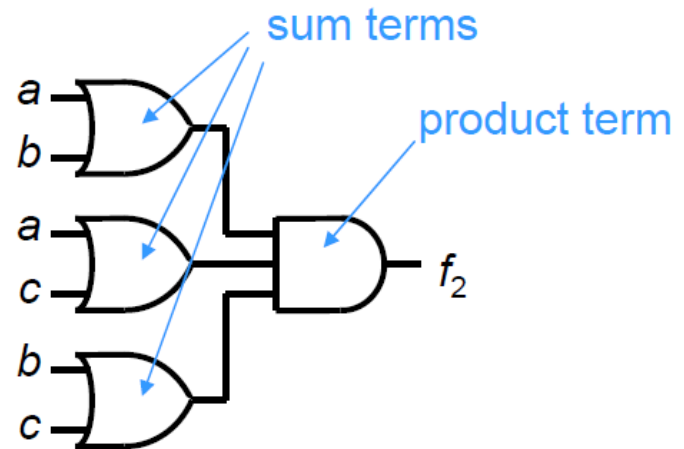
A group of AND gates followed by a single OR gate



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

Product of sums

A group of OR gates followed by a single AND gate



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

Relationship

- $\overline{\text{maxterm}_i} = \text{minterm}_i$ (i.e. $\overline{M_i} = m_i$)
- $\overline{\text{minterm}_i} = \text{maxterm}_i$ (i.e. $\overline{m_i} = M_i$)

- e.g. $M_3' = (a + b' + c')'$
- $= a'bc$ (De Morgan's Theorem)
- $= m_3$

Example: Find f , f' in SSP form

Inputs			Outputs	
a	b	c	f	f'
0	0	0	0	1
0	0	1	1	0
0	1	0	0	1
0	1	1	0	1
1	0	0	1	0
1	0	1	1	0
1	1	0	0	1
1	1	1	0	1

f is 1 if $(a = 0 \text{ AND } b = 0 \text{ AND } c = 1) \text{ OR}$
 $(a = 1 \text{ AND } b = 0 \text{ AND } c = 0) \text{ OR}$
 $(a = 1 \text{ AND } b = 0 \text{ AND } c = 1)$

$$\begin{aligned}
 f(a, b, c) &= a'b'c + ab'c' + ab'c \\
 &= m_1 + m_4 + m_5 \\
 &= \Sigma m(1, 4, 5)
 \end{aligned}$$

$$\begin{aligned}
 f'(a, b, c) &= a'b'c' + a'bc' + a'bc + abc' + abc \\
 &= m_0 + m_2 + m_3 + m_6 + m_7 \\
 &= \Sigma m(0, 2, 3, 6, 7)
 \end{aligned}$$

Abbreviated form:

Σ = logical sum (Boolean OR)

Example: Find f , f' in SPS form

Take the complement of f' to obtain f :

$$\begin{aligned}f(a, b, c) &= (f')' \\&= (m_0 + m_2 + m_3 + m_6 + m_7)' \\&= m_0' \cdot m_2' \cdot m_3' \cdot m_6' \cdot m_7' \\&= M_0 \cdot M_2 \cdot M_3 \cdot M_6 \cdot M_7 \\&= \Pi M(0, 2, 3, 6, 7) \quad \text{or} = (a+b+c)(a+b'+c)(a+b'+c')(a'+b'+c)(a'+b'+c')\end{aligned}$$

Following the same idea, we can obtain f' by:

$$\begin{aligned}f'(a, b, c) &= (f)' = (m_1 + m_4 + m_5)' \\&= m_1' \cdot m_4' \cdot m_5' \\&= M_1 \cdot M_4 \cdot M_5 \\&= \Pi M(1, 4, 5) \quad \text{or} = (a+b+c')(a'+b+c)(a'+b+c')\end{aligned}$$

↑
Abbreviated form: Π = logical product (Boolean AND)