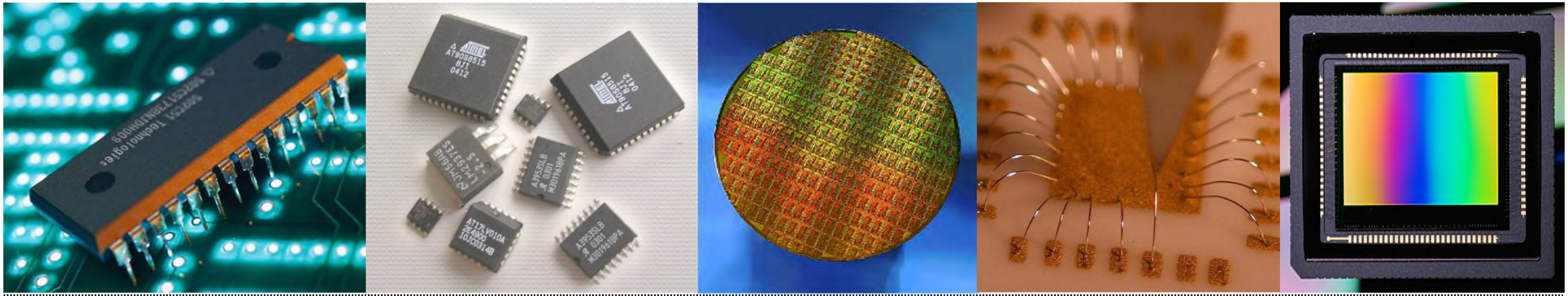


2. Boolean Algebra and Logic Gates

2.1 Boolean Algebra and Logic Gates

- Boolean Algebra
  - Define
    - a set of elements, a set of operators
    - a number of unproved axioms or postulates

2.2 Basic Definition

- Various algebraic structures

1. Closure : A set S is closed with respect to a binary operator if, for every pair of elements of S, the binary operator specifies a rule for obtaining a unique elements of S.
2. Associative law : $(x*y)*z=x*(y*z)$ for all $x,y,z \in S$
3. Commutative law : $x*y=y*x$ for all $x,y \in S$
4. Identity elements: for all $x \in S$, $e*x=x*e=x$

   ex) set of integers $I=\{\ldots, -3, -2, -1, 0, 1, 2, 3, \ldots\}$, $x+0=0+x=x$
5. Inverse : A set S having the identity elements

   for all $x \in S$, $y \in S$, $x*y=e$
6. Distributive law : $x*(y \bullet z)=(x*y) \bullet (x*z)$

불 값은 집합과 동일한 연산법칙이 적용된다.

2.3 Axiom Definition of Boolean Algebra

- Boolean algebra is an algebraic structure defined by a set of elements, B, together with two binary operators, + and·, provided that the following(Huntington) postulates are satisfied

1. (a) Closure with respect to the operator +. 덧셈, 곱셈에 대해 닫혀 있음

   (a) Closure with respect to the operator ·.
2. (a) An identity element with respect to +, designated by 0: $x+0=0+x=x$

   (b) An identity element with respect to ·, designated by 1: $x \cdot 1=1 \cdot x=x$
3. (a) Commutative with respect to + : $x + y = y + x$ 덧셈의 항등원은 0, 곱셈의 항등원은 1

   (a) Commutative with respect to · : $x \cdot y = y \cdot x$ 곱셈, 덧셈에 대해 교환법칙 성립
4. (a) · is distributive over + : $x \cdot (y+z)=(x \cdot y)+(x \cdot z)$ 곱셈, 덧셈에 대해 분배법칙 성립

   (a) + is distributive over · : $x+(y \cdot z)=(x+y) \cdot (x+z)$
5. For every element $x \in B$, there exists an element $x' \in B$ such that (a)$x+x'=1$ and (b) $x \cdot x' = 0$ 불 값의 집합에 속하는 모든 원소 x는 자신의 역과 합하면 1, 곱하면 0이 된다.
6. There exists at least two elements $x, y \in B$ such that $x \neq y$

2.4 Basic Theorems and Properties of Boolean Algebra

- Duality

  - interchange OR and And operators and replace 1's by 0's and 0's by 1's

Table 2-1
Postulates and Theorems of Boolean Algebra

Postulate 2	(a)	$x + 0 = x$	(b)	$x \cdot 1 = x$
Postulate 5	(a)	$x + x' = 1$	(b)	$x \cdot x' = 0$
Theorem 1	(a)	$x + x = x$	(b)	$x \cdot x = x$
Theorem 2	(a)	$x + 1 = 1$	(b)	$x \cdot 0 = 0$
Theorem 3, involution		$(x')' = x$		
Postulate 3, commutative	(a)	$x + y = y + x$	(b)	$xy = yx$
Theorem 4, associative	(a)	$x + (y + z) = (x + y) + z$	(b)	$x(yz) = (xy)z$
Postulate 4, distributive	(a)	$x(y + z) = xy + xz$	(b)	$x + yz = (x + y)(x + z)$
Theorem 5, DeMorgan	(a)	$(x + y)' = x'y'$	(b)	$(xy)' = x' + y'$
Theorem 6, absorption	(a)	$x + xy = x$	(b)	$x(x + y) = x$

- Operator precedence

  1. Parentheses
  2. NOT
  3. AND
  4. OR

연산 순서는
1. 괄호
2. 부정(')
3. (논리)곱
4. (논리)합

2.5 Boolean Functions

Table 2-2
Truth Tables for F_1 and F_2

x	y	z	F_1	F_2
0	0	0	0	0
0	0	1	1	1
0	1	0	0	0
0	1	1	0	1
1	0	0	1	1
1	0	1	1	1
1	1	0	1	0
1	1	1	1	0

- $F_1 = x + y'z$
- $F_2 = x'y'z + x'yz + xy'$
  $= x'z(y' + y) + xy'$
  $= x'z + xy'$

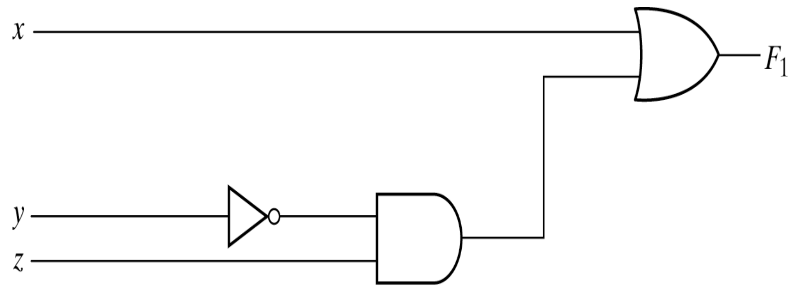
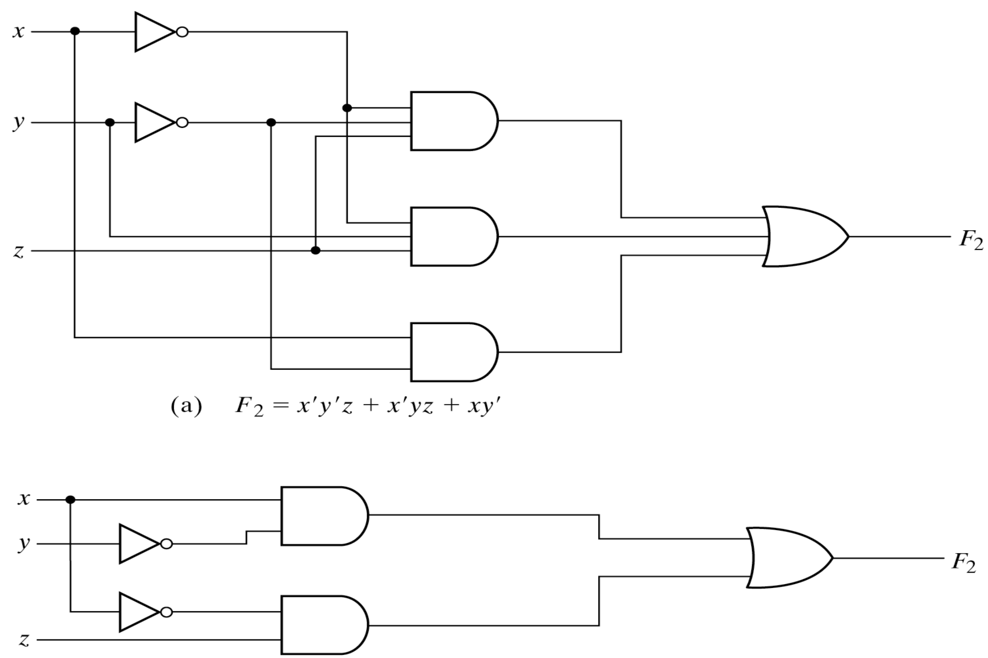


Fig. 2-1 Gate implementation of $F_1 = x + y'z$



(a) $F_2 = x'y'z + x'yz + xy'$

(b) $F_2 = xy' + x'z$

Fig. 2-2 Implementation of Boolean function F_2 with gates

2.5 Boolean Functions – Algebraic Manipulation

- Ex 2-1) Simplify the following Boolean functions to a minimum number of literals.

  1. $x(x'+y) = xx' + xy = 0 + xy = xy.$
  2. $x + x'y = (x+x')(x+y) = 1(x+y) = x + y.$
  3. $(x+y)(x+y') = x + xy + xy' + yy' = x(1+y+y') = x.$
  4. $xy + x'z + yz = xy + x'z + yz(x+x')$
     $= xy + x'z + xyz + x'yz$
     $= xy(1+z) + x'z(1+y)$
     $= xy + x'z$
  5. $(x+y)(x'+z)(y+z) = (x+y)(x'+z)$: by duality from function 4.

- $(A + B + C)' = (A+x)'$ let $B+C=x$

  $= A'x'$ by theorem 5(a)(DeMorgan) 분배법칙, 드모르간의 법칙

  $= A'(B+C)'$ substitute $B+C=x$

  $= A'(B'C')$ by theorem 5(a)(DeMorgan) 분배법칙, 드모르간의 법칙

  $= A'B'C'$ by theorem 4(b)(associative) 결합 법칙에 의해 괄호 벗김

  → $(A+B+C+D+\ldots+F)' = A'B'C'D'\ldots F'$

  $(ABCD\ldots F)' = A' + B' + C' + D' + \ldots + F'$

2.5 Boolean Functions – Complement of a Function

- Ex 2-2) Find the complement of the functions

  $F_1=x'yz'+x'y'z$, $F_2=x(y'z'+yz)$.

  $F_1' = (x'yz'+x'y'z)' = (x'yz')'(x'y'z)' = (x+y'+z)(x+y+z')$

  $F_2' = [x(y'z'+yz)]' = x'+(y'z'+yz)' = x'+(y'z')'(yz)' = x'+(y+z)(y'+z')$

- Ex 2-3) Find the complement of the functions F_1 And F_2 Ex 2-2 by taking their duals and complementing each literal.

1. $F_1 = x'yz' + x'y'z$.

   The dual of F_1 is $(x'+y+z')(x'+y'+z)$

   Complement each literal : $(x+y'+z)(x+y+z')=F_1'$

2. $F_2 = x(y'z'+yz)$.

   The dual of F_2 is $x+(y'+z')(y+z)$이다.

   Complement each literal : $x'+(y+z)(y'+z')=F_2'$

dual ????????????

- Minterms and Maxterms

Table 2-3
Minterms and Maxterms for Three Binary Variables

			Minterms		Maxterms	
x	**y**	**z**	**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

불값 0이면 '첨자, 1이면 그대로

불값 0이면 그대로, 1이면 '첨자

2.6 Canonical and Standard Forms

Table 2-4
Functions of Three Variables

x	*y*	*z*	Function f_1	Function f_2
0	0	0	0	0
0	0	1	1	0
0	1	0	0	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

$$f_1 = x'y'z + xy'z' + xyz = m_1 + m_4 + m_7$$

$$f_2 = x'yz + xy'z + xyz' + xyz = m_3 + m_5 + m_6 + m_7$$

$$f_1 = (x+y+z)(x+y'+z)(x'+y+z')(x'+y'+z)$$

$$= M_0 M_2 M_3 M_5 M_6$$

minterm - 곱이 1이 되는 것들의 합
maxterm - 합이 0이 되는 것들의 곱

$$f_2 = (x+y+z)(x+y+z')(x+y'+z)(x'+y+z)$$

$$= M_0 M_1 M_2 M_4$$

2.6 Canonical and Standard Forms

- Sum of Minterms
- Ex 2-4) Express the Boolean function F=A+B'C in a sum of minterms.

$$A = A(B+B') = AB + AB'$$
$$= AB(C+C') + AB'(C+C')$$
$$= ABC + ABC' + AB'C + AB'C'$$
$$B'C = B'C(A+A') = AB'C + A'B'C$$
$$F = A + B'C$$
$$= A'B'C + AB'C' + AB'C + ABC' + ABC$$
$$= m_1 + m_4 + m_5 + m_6 + m_7$$
$$= \sum(1, 4, 5, 6, 7)$$

Table 2-5
Truth Table for $F = A + B'C$

A	*B*	*C*	*F*
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	1
1	1	1	1

2.6 Canonical and Standard Forms

- Product of maxterms
- Ex 2-5) Express the Boolean function $F = xy + x'z$ in a product of maxterm form.

$$F = xy + x'z = (xy + x')(xy + z)$$
$$= (x + x')(y + x')(x + z)(y + z)$$
$$= (x' + y)(x + z)(y + z)$$
$$x' + y = x' + y + zz' = (x' + y + z)(x' + y + z')$$
$$x + z = x + z + yy' = (x + y + z)(x + y' + z)$$
$$y + z = y + z + xx' = (x + y + z)(x' + y + z)$$
$$F = (x + y + z)(x + y' + z)(x' + y + z)(x' + y + z')$$
$$= M_0M_2M_4M_5$$
$$F(x, y, z) = \prod(0, 2, 4, 5)$$

2.6 Canonical and Standard Forms

- Conversion between Canonical Forms

  $F(A, B, C) = \sum(1, 4, 5, 6, 7)$

  $F'(A, B, C) = \sum(0, 2, 3) = m_0 + m_2 + m_3$

  $F = (m_0 + m_2 + m_3)' = m_0' m_2' m_3' = M_0 M_2 M_3 = \prod(0, 2, 3)$, $m_j' = M_j$

Ex) $F = xy + x'z$

$F(x, y, z) = \sum(1, 3, 6, 7)$

$F(x, y, z) = \prod(0, 2, 4, 5)$

Table 2-6

Truth Table for $F = xy + x'z$

x	*y*	*z*	*F*
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

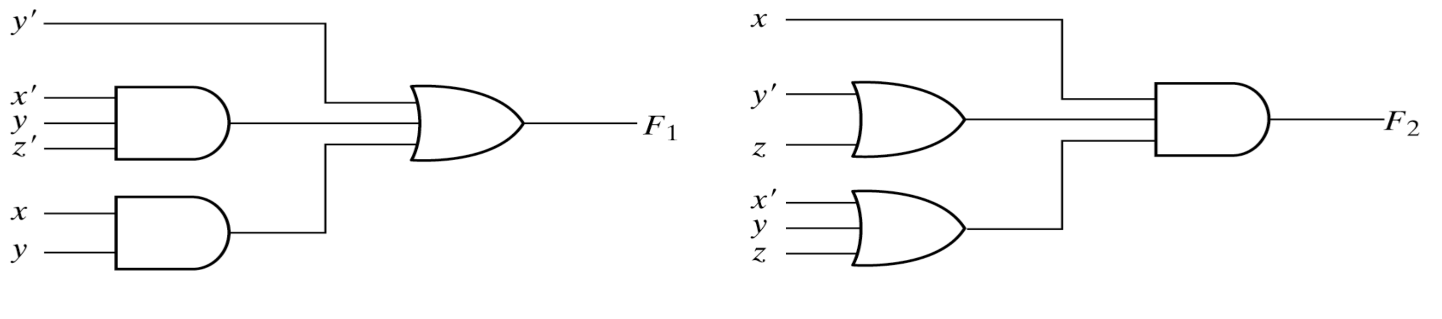
2.6 Canonical and Standard Forms

Standard Forms

- Sum of product : $F_1 = y' + xy + x'yz'$
- Product of sum : $F_2 = x(y' + z)(x' + y + z' + w)$

minterm - 곱이 1이 되는 것들의 합

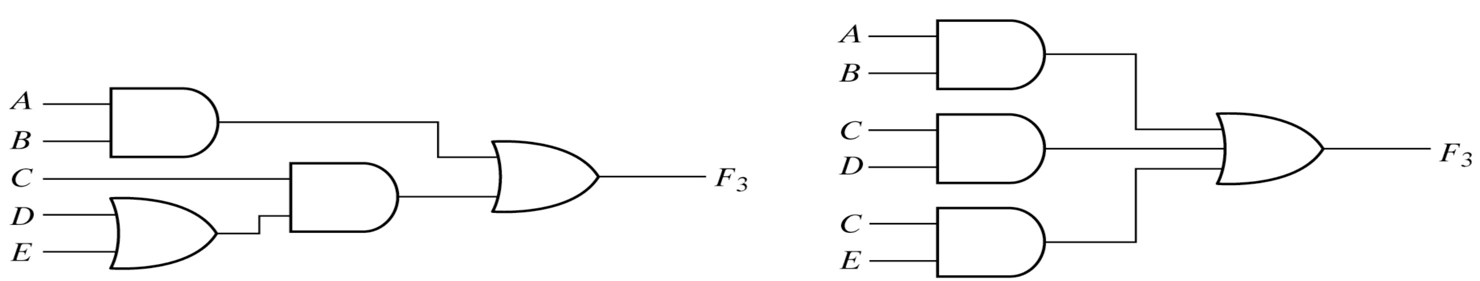
maxterm - 합이 0이 되는 것들의 곱



(a) Sum of Products (b) Product of Sums

Fig. 2-3 Two-level implementation

- Ex) $F_3 = AB + C(D + E) = AB + CD + CE$



(a) $AB + C(D + E)$ (b) $AB + CD + CE$

Fig. 2-4 Three- and Two-Level implementation

2.7 Other Logic Operations

Truth Tables for the 16 Functions of Two Binary Variables

x	y	F_0	F_1	F_2	F_3	F_4	F_5	F_6	F_7	F_8	F_9	F_{10}	F_{11}	F_{12}	F_{13}	F_{14}	F_{15}
0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
0	1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
1	0	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
1	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1

Boolean Expressions for the 16 Functions of Two Variables

Boolean functions	Operator symbol	Name	Comments
$F_0 = 0$		Null	Binary constant 0
$F_1 = xy$	$x \cdot y$	AND	x and y
$F_2 = xy'$	x/y	Inhibition	x, but not y
$F_3 = x$		Transfer	x
$F_4 = x'y$	y/x	Inhibition	y, but not x
$F_5 = y$		Transfer	y
$F_6 = xy' + x'y$	$x \oplus y$	Exclusive-OR	x or y, but not both
$F_7 = x + y$	$x + y$	OR	x or y
$F_8 = (x + y)'$	$x \downarrow y$	NOR	Not-OR
$F_9 = xy + x'y'$	$(x \oplus y)'$	Equivalence	x equals y
$F_{10} = y'$	y'	Complement	Not y
$F_{11} = x + y'$	$x \subset y$	Implication	If y, then x
$F_{12} = x'$	x'	Complement	Not x
$F_{13} = x' + y$	$x \supset y$	Implication	If x, then y
$F_{14} = (xy)'$	$x \uparrow y$	NAND	Not-AND
$F_{15} = 1$		Identity	Binary constant 1

2.8 Digital Logic Gate

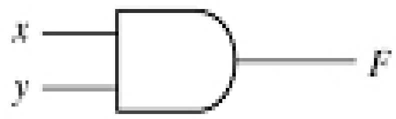
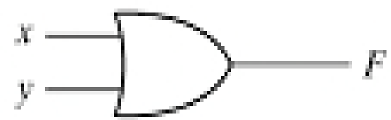
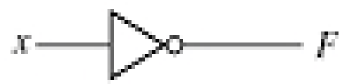
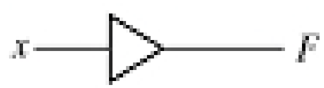
Name	Graphic symbol	Algebraic function	Truth table
AND		$F = xy$	x y F 0 0 0 0 1 0 1 0 0 1 1 1
OR		$F = x + y$	x y F 0 0 0 0 1 1 1 0 1 1 1 1
Inverter		$F = x'$	x F 0 1 1 0
Buffer		$F = x$	x F 0 0 1 1

Fig. 2-5 Digital logic gates

2.8 Digital Logic Gate

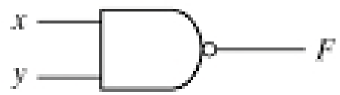
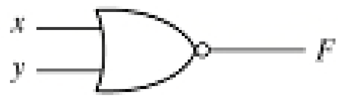
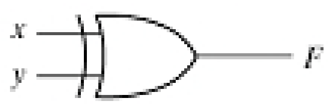
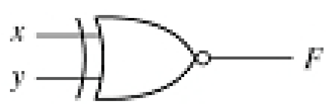
Name	Graphic symbol	Algebraic function	Truth table x	y	F
NAND		$F = (xy)'$	0	0	1
			0	1	1
			1	0	1
			1	1	0
NOR		$F = (x + y)'$	0	0	1
			0	1	0
			1	0	0
			1	1	0
Exclusive-OR (XOR)		$F = xy' + x'y$ $= x \oplus y$	0	0	0
			0	1	1
			1	0	1
			1	1	0
Exclusive-NOR or equivalence		$F = xy + x'y'$ $= (x \oplus y)'$	0	0	1
			0	1	0
			1	0	0
			1	1	1

Fig. 2-5 Digital logic gates

- Extension to Multiple Inputs

NAND 게이트와 NOR 게이트는 결합법칙이 성립되지 않는다

- The NAND and NOR operators are not associative.

NOR

$(x \downarrow y) \downarrow z \neq x \downarrow (y \downarrow z)$

$(x \downarrow y) \downarrow z = [(x+y)'+z]'$

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

$x \downarrow (y \downarrow z) = [x+(y+z)']'$

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

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

$x \uparrow y \uparrow z = (xyz)'$

NAND

$F = [(ABC)'(DE)']' = ABC + DE$

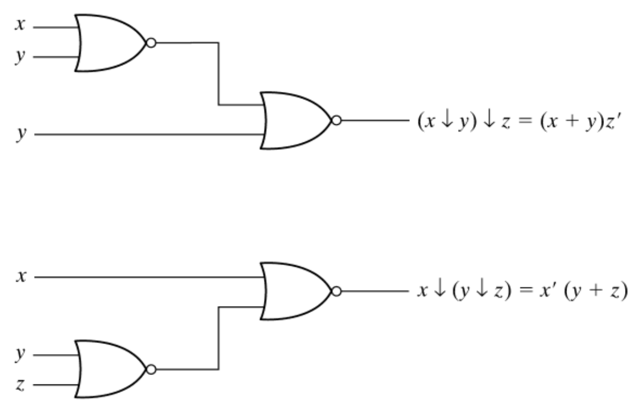
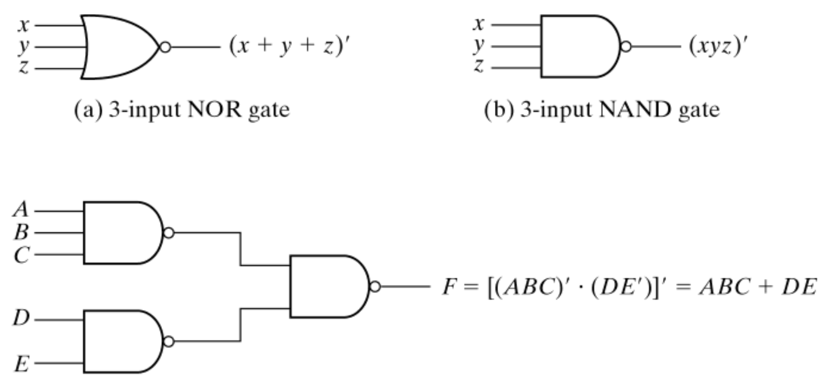


Fig. 2-6 Demonstrating the nonassociativity of the NOR operator; $(x \downarrow y) \downarrow z \neq x \downarrow (y \downarrow z)$



(a) 3-input NOR gate

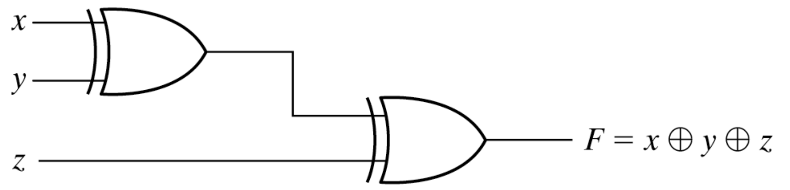
(b) 3-input NAND gate

(c) Cascaded NAND gates

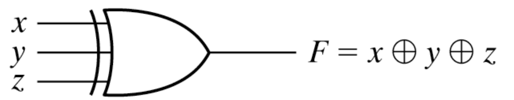
Fig. 2-7 Multiple-input and cascaded NOR and NAND gates

2.8 Digital Logic Gate

- exclusive-OR



(a) Using 2-input gates



(b) 3-input gate

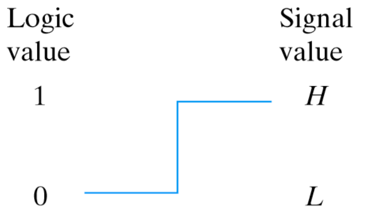
x	y	z	F
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

(c) Truth table

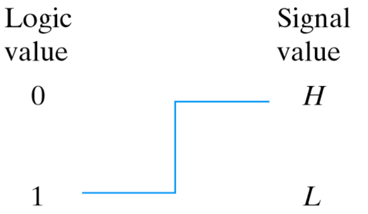
1의 갯수가 홀수개이면 참

Fig. 2-8 3-input exclusive-OR gate

Positive and Negative Logic



(a) Positive logic



(b) Negative logic

Fig. 2-9 signal assignment and logic polarity