Canonical Forms

- Canonical Means "Unique"
- All Possible Functions can be Expressed in One and Only One Way in a Canonical Form
- Canonical Forms may be Circuit Diagrams or Algebraic Equations

Minterms and Maxterms

- Consider a function of Three variables x, y, and z
- Since each Variable may be Complemented or Uncomplemented there are $2^3=8$ Different Combinations
- When Combinations are Combined with AND they are Called Minterms
- When Combinations are combined with OR they are Called Maxterms

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

For n Variables there are 2^n Minterms/Maxterms

Sum-of-Minterms Form

- Canonical Form – *Standard Products*
- Determine the Set of Minterms for Which a Function is 1-valued
  - These are called “Minterms of the Function”
- Combine all Minterms with a + Operation
- This is a 2-Level Form

Sum-of-Minterms Example

x	y	z	f_1	
0	0	0	0	
0	0	1	1	$\leftarrow \bar{x}\,\bar{y}\,z$
0	1	0	0	
0	1	1	0	
1	0	0	1	$\leftarrow x\,\bar{y}\,\bar{z}$
1	0	1	0	
1	1	0	0	
1	1	1	1	$\leftarrow x\,y\,z$

$$f_1 = \bar{x}\,\bar{y}\,z + x\,\bar{y}\,\bar{z} + x\,y\,z \qquad f_1 = \sum(1,4,7)$$

Product-of-Maxterms Form

- Canonical Form – *Standard Sums*
- Determine the Set of Maxterms for Which a Function is 0-valued
  - These are called “Maxterms of the Function”
- Must Complement Each Literal
- Combine all Maxterms with a • Operation
- This is a 2-Level Form

Product-of-Maxterms Example

x	y	z	f_1	
0	0	0	0	← $x + y + z$
0	0	1	1	
0	1	0	0	← $x + \overline{y} + z$
0	1	1	0	← $x + \overline{y} + \overline{z}$
1	0	0	1	
1	0	1	0	← $\overline{x} + y + \overline{z}$
1	1	0	0	← $\overline{x} + \overline{y} + z$
1	1	1	1	

$$f_1 = (x + y + z)(x + \overline{y} + z)(x + \overline{y} + \overline{z})$$
$$(\overline{x} + y + \overline{z})(\overline{x} + \overline{y} + z)$$

$$f_1 = \prod (0, 2, 3, 5, 6)$$

Other Notation

$$f_1 = \sum (1, 4, 7) = m_1 + m_4 + m_7$$

$$f_1 = \prod (0, 2, 3, 5, 6) = M_0 \bullet M_2 \bullet M_3 \bullet M_5 \bullet M_6$$

What is the function:

$$f = m_0 + m_1 + m_2 + m_3 + m_4 + m_5 + m_6 + m_7$$

Conversion of Canonic Forms

$$f_1 = \sum(1,4,7) = m_1 + m_4 + m_7$$

$$\overline{f}_1 = \sum(0,2,3,5,6) = m_0 + m_2 + m_3 + m_5 + m_6$$

$$\overline{\overline{f}}_1 = \overline{m_0 + m_2 + m_3 + m_5 + m_6}$$

$$f_1 = \overline{m_0 + m_2 + m_3 + m_5 + m_6}$$

DeMorgan's Theorem:

$$f_1 = \overline{m_0} \bullet \overline{m_2} \bullet \overline{m_3} \bullet \overline{m_5} \bullet \overline{m_6}$$

$$\overline{m_i} = M_i$$

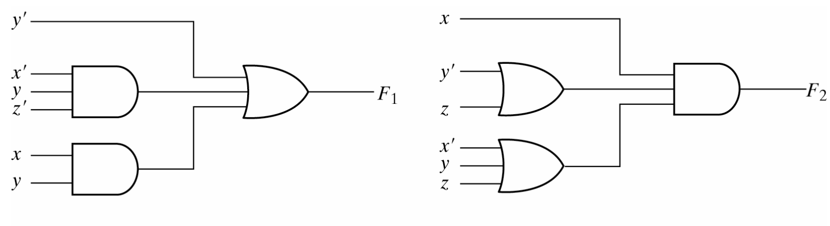
$$f_1 = M_0 \bullet M_2 \bullet M_3 \bullet M_5 \bullet M_6$$

Standard Forms

- Canonical Forms USUALLY NOT Smallest (in terms of literals)
- Each minterm/maxterm contains n literals
- Standard Forms Contain Terms with n or Fewer Literals
- Sum-Of-Products (SOP) form
- Product-Of-Sums (POS) form
- These are Also Two-level Forms

Standard Forms Examples

$$F_1 = \overline{y} + xy + \overline{x}\,y\,\overline{z} \qquad F_2 = x(\overline{y} + z)(\overline{x} + y + \overline{z})$$



(a) Sum of Products

(b) Product of Sums

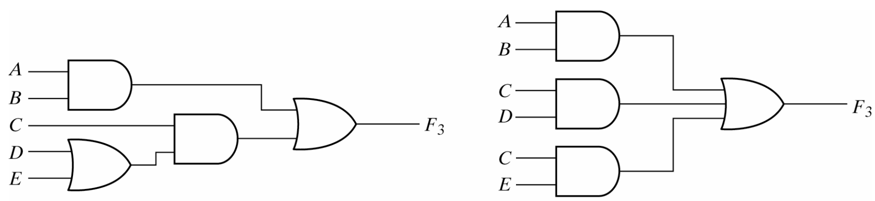
Fig. 2-3 Two-level implementation

Standard Forms

- Can Use Algebra to Find a Standard Form from a Canonical Form
- We Will Learn Other Methods to do this
- Commonly Known as “Simplification”
  - Seems Easy for Small Functions
  - Computationally Complex
  - Classic Problem in Switching Theory

Multi-level Forms

- All Possible Functions can be Expressed in a Standard Two-level Form
- Multi-level Forms have more than 2 Levels



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

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

All Functions of 2-Variables

- There are 2^{2^n} functions of n Variables
- AND and OR Happen to be Two of 16 Possible Functions of 2 Variables

Table 2-7
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

Names and Symbols of Functions

Table 2-8
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

Common Circuits

- Certain Subsets Form ANY Function
  - AND, OR, NOT
  - NAND
  - NOR
  - AND, XOR

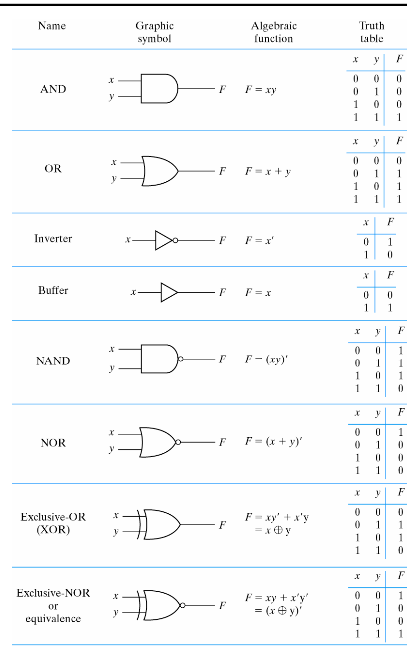


Fig. 2-5 Digital logic gates

Multi-input Logic Gates

- Many Logic Gates Can Have More than One Input
- Examples are AND and OR Gates
- Associativity Holds for These
- Can Build with Cascade of 2-input Gates

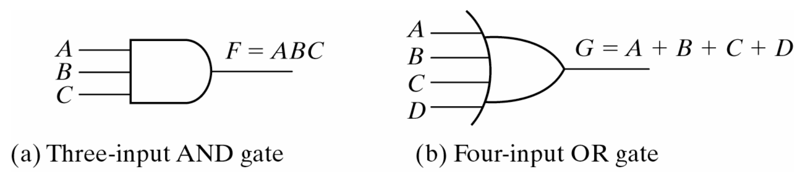


Fig. 1-6 Gates with multiple inputs

Multi-input Logic Gates

- Associativity Does Not Hold for NOR

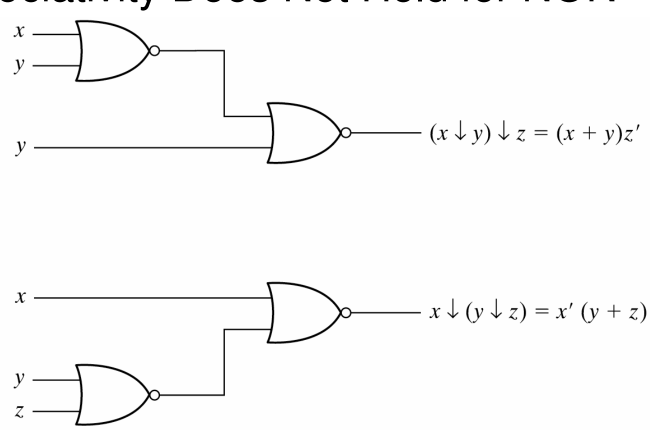


Fig. 2-6 Demonstrating the nonassociativity of the NOR operator; $(x \downarrow y) \downarrow z \neq x \downarrow (y \downarrow z)$

Multi-input Logic Gates

- Associativity Does Not Hold for NAND

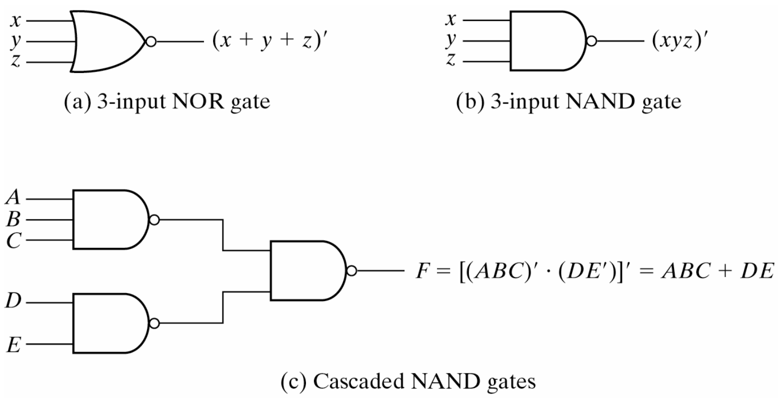
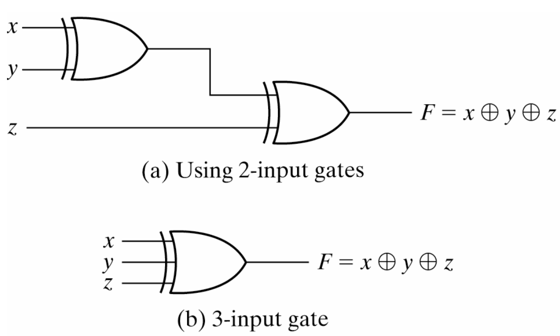


Fig. 2-7 Multiple-input and cascaded NOR and NAND gates

Exclusive-OR Gates

- Associativity Holds
- “Programmable” Inverter



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

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

Positive and Negative Logic

- 0 and 1 are Models to use Algebra
- Circuits use High (H) and Low (L) Values
  - Voltage or Current
- Up to Designer to Interpret if H→0 or H→1
- Interpretations Called ‘Positive’/’Negative’ Logic

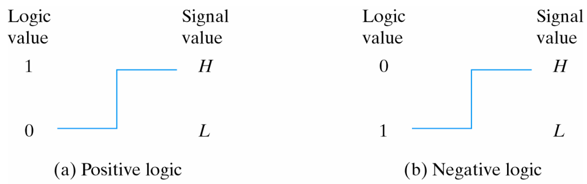


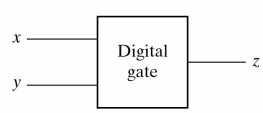
Fig. 2-9 signal assignment and logic polarity

Positive and Negative Logic Example and Notation

Positive-Logic AND becomes Negative-Logic OR

x	y	F
L	L	L
L	H	L
H	L	L
H	H	H

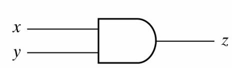
(a) Truth table with H and L



(b) Gate block diagram

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

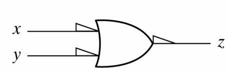
(c) Truth table for positive logic



(d) Positive logic AND gate

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

(e) Truth table for negative logic



(f) Negative logic OR gate

Fig. 2-10 Demonstration of positive and negative logic