Chapter2: Boolean Algebra and Logic Gates

Lecture4- Other Logic Operations

Engr. Arshad Nazir, Asst Prof
Dept of Electrical Engineering
SEECS

Objectives

- Study other Logic Operations
- Different Function Gates
- Extension to Multiple Inputs

Other Logic Operations

- Given two Boolean variables:
  - When binary operators AND and OR are placed between two variables, they form two Boolean functions x . y and x + y
  - there are 2^{2x2} = 16 combinations of the two variables as there are 2^{2n} possible functions for n binary variables (we will see the details of these 16 functions in next slides)
  - each combination of the variables can result in one of two values, 0 or 1, therefore there are 2^4=16 functions (combinations of 0's and 1's for the four combinations, 00,01,10,11)
- These 16 functions listed can be subdivided into three categories:
  - Two functions that produce a constant 0 or1.
  - Four functions with unary operations: complement and transfer.
  - Ten functions with binary operators that define eight different operations: AND, OR, NAND, NOR, exclusive-OR, equivalence, inhibition, and implication.

Function Combinations

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

- F_1 represents the AND Operation
- F_7 represents the OR Operation
- There are 14 other functions

16 Two-Variable 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

Function Gate Implementations

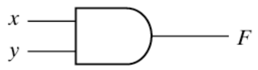
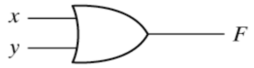
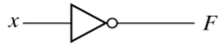
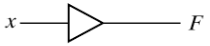
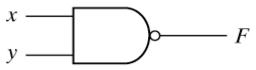
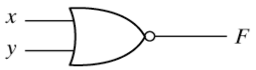
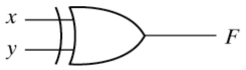
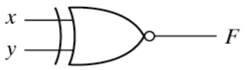
Name	Graphic symbol	Algebraic function	Truth table
AND		$F = xy$	x y F 0 0 0 0 1 0 1 0 0 1 1 1
OR		$F = x + y$	x y F 0 0 0 0 1 1 1 0 1 1 1 1
Inverter		$F = x'$	x F 0 1 1 0
Buffer		$F = x$	x F 0 0 1 1
NAND		$F = (xy)'$	x y F 0 0 1 0 1 1 1 0 1 1 1 0
NOR		$F = (x + y)'$	x y F 0 0 1 0 1 0 1 0 0 1 1 0
Exclusive-OR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

Fig. 2-5 Digital logic gates

Function Gate Implementations

- It is easier to implement a Boolean function with these types of gates (as seen on last slide)
- Inverter (Complement), Buffer (transfer), AND, OR, NAND, NOR, X-OR, and XNOR (equivalence) are used as standard gates in digital design
- NAND and NOR are extensively used logic gates and are more popular than AND and OR gates because these gates are easily constructed with transistor circuits and digital circuits are easily implemented with them.
- Implication and inhibition are not commutative or associative and thus are impractical to use as standard logic gates.

Multiple Inputs

- All the previously defined gates, with the exception of the inverter and the buffer, can have multiple inputs.
  - A gate can have multiple inputs provided it is a binary operation that is commutative ($x + y = y + x$ and $xy = yx$) and associative ($x + (y + z) = (x + y) + z$ and $x(yz) = (xy)z$)
  - NAND and NOR functions are commutative but not associative.

    For example, for NOR

    $X \downarrow Y = Y \downarrow X$ commutative

    $(X \downarrow Y) \downarrow Z \neq X \downarrow (Y \downarrow Z)$ not associative

    $xz' + yz' \neq x'y + x'z$
  - To overcome this difficulty we define multiple NOR (or NAND) gate as a complemented OR (or AND) gate e.g., as $(x+y+z)'$ or $(xyz)'$

Multiple Inputs (Non-associative NOR operation)

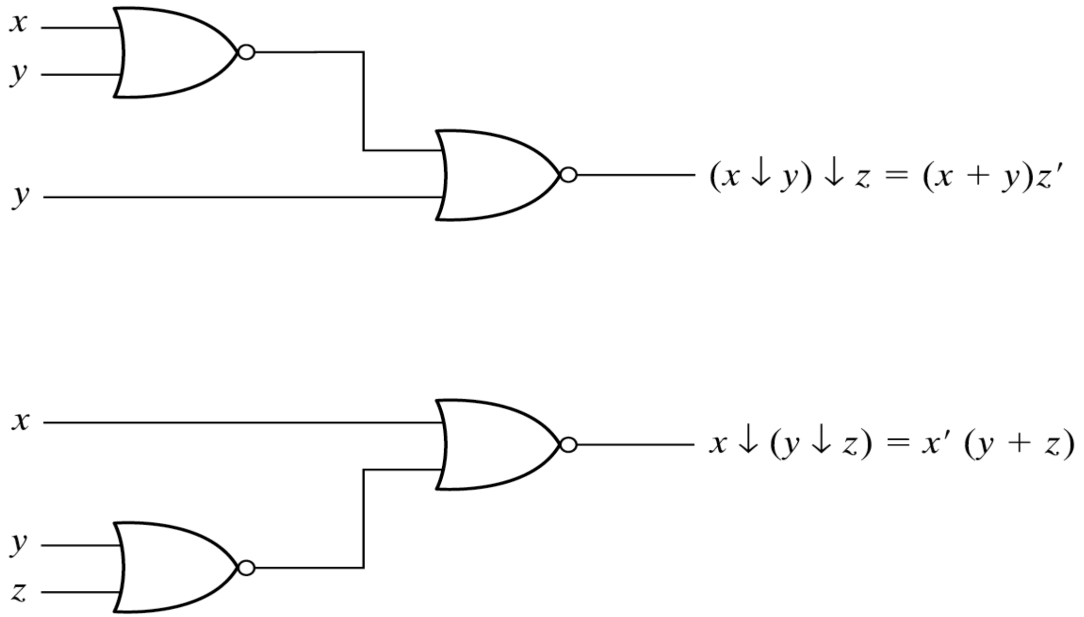
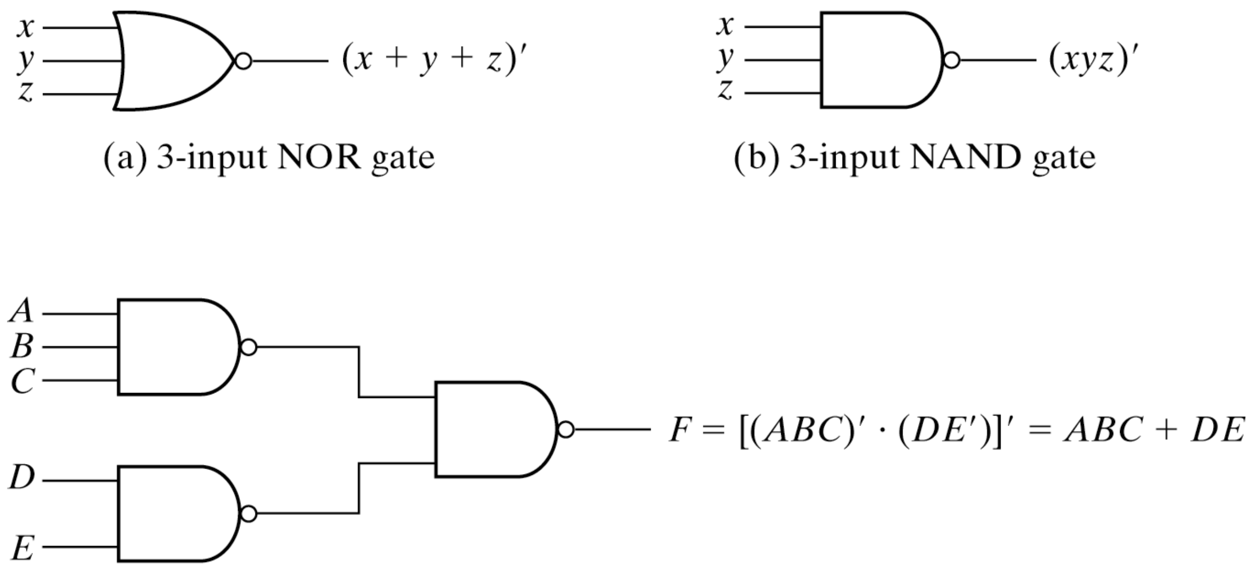


Fig. 2-6 Demonstrating the nonassociativity of the NOR operator; $(x \downarrow y) \downarrow z \neq x \downarrow (y \downarrow z)$

Multiple Inputs NOR and NAND gates



(a) 3-input NOR gate

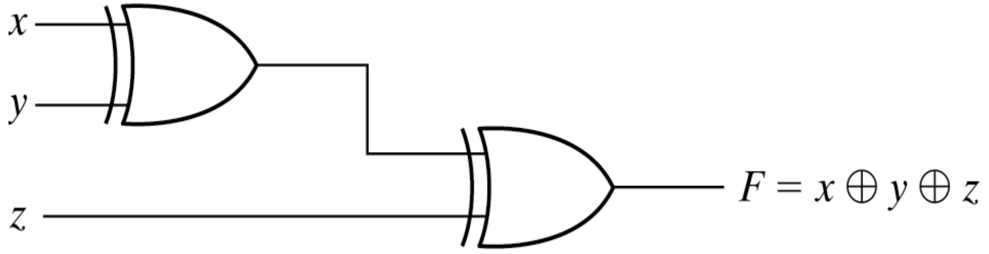
(b) 3-input NAND gate

(c) Cascaded NAND gates

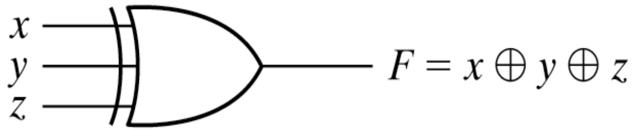
Fig. 2-7 Multiple-input and cascated NOR and NAND gates

Multiple Inputs XOR gate

- 3-input XOR gate is normally implemented by cascading 2-input gates (multiple inputs XOR is uncommon from hardware point)



(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

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

The End