Boolean Functions

Reference: Chapter 3
Digital Principles and Logic Design
By: A. Saha & N. Manna

Prepared By:
Nuren Zabin Shuchi
Lecturer
Department of EEE, SUST

Expressing Boolean Functions

- A Boolean function is an expression formed with binary variables, the two binary operators AND and OR, one unary operator NOT, parentheses and equal sign.

$$F = AB'C$$

- The value of a function may be 0 or 1, depending on the values of variables present in the Boolean function or expression.

- For example, the value of F will be 1, when A = 1, B = 0, and C = 1. For other values of A, B, C the value of F is 0.

Expressing Boolean Functions (Cont.)

- Boolean functions can also be represented by truth tables. A *truth table* is the tabular form of the values of a Boolean function according to the all possible values of its variables.

- For an *n* number of variables, 2^n combinations of 1's and 0's are listed and one column represents function values according to the different combinations.

- For example, for three variables the Boolean function F = AB + C truth table can be written as

A	B	C	F
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

Expressing Boolean Functions (Cont.)

A logic function can be expressed in the following forms.

$\left.\begin{array}{l}(i)\ \text{Sum of the Products (SOP)} \\ (ii)\ \text{Product of the Sums (POS)}\end{array}\right\}$ Standard form of Expressing a function.

Terms	Definition	Example
Product Term	An **AND** function is referred to as a product term or standard product.	AB'C
Sum Term	An OR function is referred to as a sum term.	A+B+C'
Sum of the Products (SOP)	The logical sum of two or more logical product terms is referred to as a sum of products expression.	F= ABC+A'BC+ABC' Y = AB + BC + AC
Product of the Sums (POS)	The logical product of two or more logical sum terms is called a product of sums expression.	$F = (A + B + C)(A + B' + C)(A + B + C')$

Expressing Boolean Functions (Cont.)

Boolean functions expressed as a sum of **minterms** or product of **maxterms** are said to be in **canonical form.**

Terms	Definition	Example
Minterms	A product term containing all *n* variables of the function in either true or complemented form is called the minterm.	For a two-variable function, four different combinations are possible, such as, A′B′, A′B, AB′, and AB. Each of these is called a minterm.
Maxterm	A sum term containing all *n* variables of the function in either true or complemented form is called the maxterm.	For a two-variable function, four different combinations are possible, such as, A′+B′, A′+B, A+B′, and A+B. Each of these is called a maxterm.

Expressing Boolean Functions (Cont.)

			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

- In the minterm, a variable will possess the value 1 if it is in true or uncomplemented form, whereas, it contains the value 0 if it is in complemented form.

- In the maxterm, a variable will possess the value 0, if it is in true or uncomplemented form, whereas, it contains the value 1, if it is in complemented form.

Expressing Boolean Functions (Cont.)

			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

Terms	Definition	Example
Canonical Sum of the Products Expression	When a Boolean function is expressed as the logical sum of all the minterms from the rows of a truth table, for which the **value of the function is 1**, it is referred to as the ***canonical sum of product expression***.	$F(A,B,C) = \Sigma(2,4,5,6)$ $= m_2 + m_4 + m_5 + m_6$ $= A'BC' + AB'C' + AB'C + ABC'.$
Canonical Product of Sum Expression	When a Boolean function is expressed as the logical product of all the maxterms from the rows of a truth table, for which the **value of the function is 0**, it is referred to as the ***canonical product of sum expression.***	$F(A,B,C = \Pi(0,2,5)$ $= M_0 M_2 M_5$ $= (A + B + C)(A + B' + C)(A' + B + C')$

Obtaining the canonical sum of product form of a Logic Function

The canonical sum of products form of a logic function can be obtained by using the following procedure.

1. Check each term in the given logic function. Retain if it is a minterm, continue to examine the next term in the same manner.

2. Examine for the variables that are missing in each product which is not a minterm. For Example,
If the missing variable in the minterm is X, multiply that minterm with (X+X').

3. Multiply all the products and discard the redundant terms.

$$F(A, B, C) = A + BC$$

Solution. Here neither the first term nor the second term is minterm. The given function contains three variables A, B, and C. The variables B and C are missing from the first term of the expression and the variable A is missing from the second term of the expression. Therefore, the first term is to be multiplied by $(B + B')$ and $(C + C')$. The second term is to be multiplied by $(A + A')$. This is demonstrated below.

$$\begin{aligned} F(A, B, C) &= A + BC \\ &= A(B + B')(C + C') + BC(A + A') \\ &= (AB + AB')(C + C') + ABC + A'BC \\ &= ABC + AB'C + ABC' + AB'C' + ABC + A'BC \\ &= ABC + AB'C + ABC' + AB'C' + A'BC \text{ (as } ABC + ABC = ABC) \end{aligned}$$

Hence the canonical sum of the product expression of the given function is

$$F(A, B) = ABC + AB'C + ABC' + AB'C' + A'BC.$$

Obtaining the canonical product of sum form of a logic Function

The canonical product of sums form of a logic function can be obtained by using the following procedure.

1. Check each term in the given logic function. Retain it if it is a maxterm, continue to examine the next term in the same manner.

2. Examine for the variables that are missing in each sum term that is not a maxterm.
If the missing variable in the maxterm is X, multiply that maxterm with (X.X').

3. Expand the expression using the properties and postulates as described earlier and discard the redundant terms.

$$F(A, B, C) = (A + B')(B + C)(A + C')$$

Solution. In the above three-variable expression, C is missing from the first term, A is missing from the second term, and B is missing from the third term. Therefore, CC′ is to be added with first term, AA′ is to be added with the second, and BB′ is to be added with the third term. This is shown below.

$$\begin{aligned} F(A, B, C) &= (A + B')(B + C)(A + C') \\ &= (A + B' + 0)(B + C + 0)(A + C' + 0) \\ &= (A + B' + CC')(B + C + AA')(A + C' + BB') \\ &= (A + B' + C)(A + B' + C')(A + B + C)(A' + B + C)(A + B + C') \\ &\quad (A + B' + C') \end{aligned}$$

[using the distributive property, as X + YZ = (X + Y)(X + Z)]

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

[as (A + B′ + C′) (A + B′ + C′) = A + B′ + C′]

Hence the canonical product of the sum expression for the given function is

$$F(A, B, C) = (A + B' + C)(A + B' + C')(A + B + C)(A' + B + C)(A + B + C')$$

Deriving a Sum of Products (SOP) Expression from a Truth Table

The procedure of deriving the output expression in SOP form from a truth table can be summarized as below.

1. Form a product term for each input combination in the table, containing an output value of 1.

2. Each product term consists of its input variables in either true form or complemented form. If the input variable is 0, it appears in complemented form and if the input variable is 1, it appears in true form.

3. To obtain the final SOP expression of the output, all the product terms are OR operated.

Inputs			*Output*	*Product terms*	*Sum terms*
A	*B*	*C*	*Y*		
0	0	0	0		$A + B + C$
0	0	1	0		$A + B + C'$
0	1	0	1	$A'BC'$	
0	1	1	0		$A + B' + C'$
1	0	0	1	$AB'C'$	
1	0	1	1	$AB'C$	
1	1	0	1	ABC'	
1	1	1	0		$A' + B' + C'$

The final sum of products expression (SOP) for the output Y is derived by summing or performing an OR operation of the four product terms as shown below.

$$Y = A'BC' + AB'C' + AB'C + ABC'$$

Deriving a Product of Sum (POS) Expression from a Truth Table

The procedure of deriving the output expression in POS form from a truth table can be summarized as below.

1. Form a sum term for each input combination in the table, containing an output value of 0.

2. Each sum term consists of its input variables in either true form or complemented form. If the input variable is 0, it appears in true form and if the input variable is 0, it appears in complemented form.

3. To obtain the final POS expression of the output, all the sum terms are AND operated.

Inputs			*Output*	*Product terms*	*Sum terms*
A	*B*	*C*	*Y*		
0	0	0	0		$A + B + C$
0	0	1	0		$A + B + C'$
0	1	0	1	$A'BC'$	
0	1	1	0		$A + B' + C'$
1	0	0	1	$AB'C'$	
1	0	1	1	$AB'C$	
1	1	0	1	ABC'	
1	1	1	0		$A' + B' + C'$

$$Y = (A + B + C)\,(A + B + C')\,(A + B' + C')\,(A' + B' + C')$$

Deriving a the expression of a function from a Truth Table

Inputs			Output	Product terms	Sum terms
A	B	C	Y		
0	0	0	0		$A+B+C$
0	0	1	0		$A+B+C'$
0	1	0	1	$A'BC'$	
0	1	1	0		$A+B'+C'$
1	0	0	1	$AB'C'$	
1	0	1	1	$AB'C$	
1	1	0	1	ABC'	
1	1	1	0		$A'+B'+C'$

$$Y = (A+B+C)(A+B+C')(A+B'+C')(A'+B'+C')$$

Or

$$Y = A'BC' + AB'C' + AB'C + ABC'$$

Which one of these is the expression of the function expressed by the truth table?

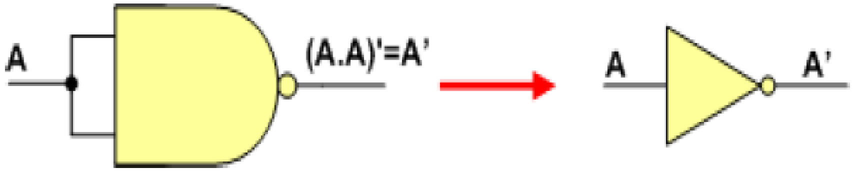
Universal Gates

- A universal gate is a gate which can implement any Boolean function without need to use any other gate type.

- The **NAND** and **NOR** gates are universal gates.

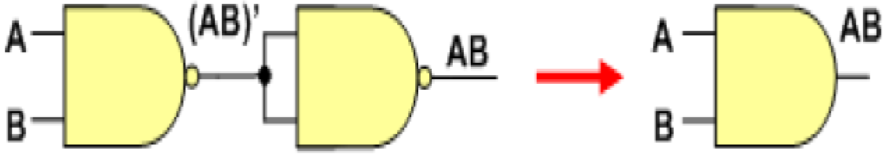
NAND Gate as Universal Gate

- **Implementing an Inverter Using only NAND Gate**



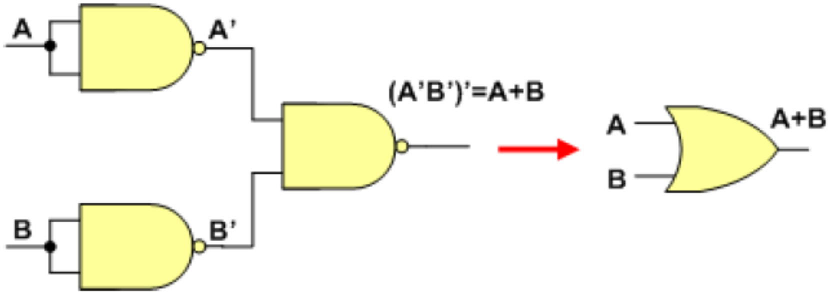
All NAND input pins connect to the input signal **A** gives an output **A**'.

- **Implementing an AND Gate Using only NAND Gate**



The AND is replaced by a NAND gate with its output complemented by a NAND gate inverter.

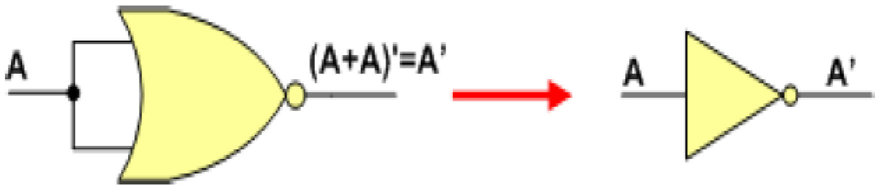
- **Implementing an OR Gate Using only NAND Gate**



The OR gate is replaced by a NAND gate with all its inputs complemented by NAND gate inverters.

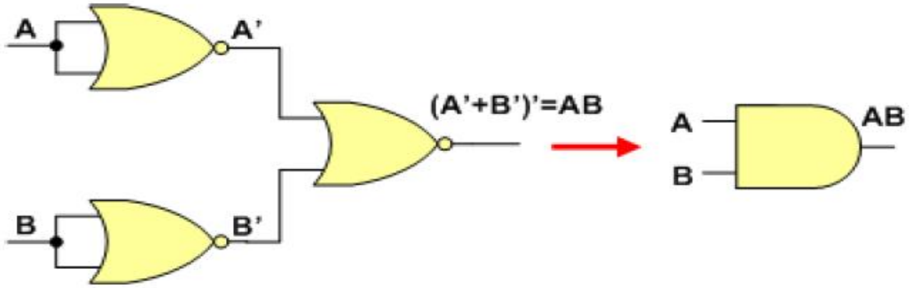
NOR Gate as Universal Gate

- **Implementing an Inverter Using only NOR Gate**



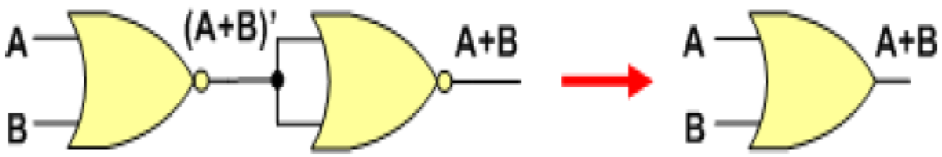
All NOR input pins connect to the input signal **A** gives an output **A'**.

- **Implementing an AND Gate Using only NOR Gate**



The AND gate is replaced by a NOR gate with all its inputs complemented by NOR gate inverters.

- **Implementing an OR Gate Using only NOR Gate**



The OR is replaced by a NOR gate with its output complemented by a NOR gate inverter.

Examples: 3.1-3.20

Exercises: 3.31-3.35, 3.37-3.44