Canonical and Standard form

Lecture-9

By
Bhagyalaxmi Behera
Asst. Professor (Dept. of ECE)

Contents

- Complement of Boolean function
- HDL Description of Boolean Function
- Canonical form representation of Boolean Functions
- Standard form representation of Boolean Functions

Complement of Boolean function

- The complement of a function *F is F' and is obtained from an interchange of 0's for 1's* and 1's for 0's in the value of *F.*
- *The complement of a function may be derived algebraically* through DeMorgan's theorems.
- DeMorgan's theorems can be extended to three or more variables.

$$
\begin{aligned}
(A + B + C)' &= (A + x)' && \text{let } B + C = x \\
&= A'x' && \text{by theorem 5(a) (DeMorgan)} \\
&= A'(B + C)' && \text{substitute } B + C = x \\
&= A'(B'C') && \text{by theorem 5(a) (DeMorgan)} \\
&= A'B'C' && \text{by theorem 4(b) (associative)}
\end{aligned}
$$

Complement of Boolean function

DeMorgan's theorems for any number of variables resemble the two-variable case in form and can be derived by successive substitutions similar to the method used in the preceding derivation. These theorems can be generalized as follows:

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

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

The generalized form of DeMorgan's theorems states that the complement of a function is obtained by interchanging AND and OR operators and complementing each literal.

Ex_1: $F_1 = x'yz' + x'y'z$

applying DeMorgan's theorem.

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

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

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

A simpler procedure for deriving the complement of a function is to take the dual of the function and complement each literal.

Ex_2: $F_2 = x'y z' + x\, y' z$

Dual of F_2 $= (x' + y + z')(x' + y' + z)$

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

Find the complement of the function *F2 by* Applying DeMorgan's theorems

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

Find the complement of the function F2 by taking dual of the function and complementing each literal.

$F_2 = x(y'z' + yz)$.
The dual of F_2 is $x + (y' + z')(y + z)$.
Complement each literal: $x' + (y + z)(y' + z') = F_2'$.

HDL Description of Boolean Function

- Boolean equations describing combinational logic are specified in Verilog with a continuous assignment statement consisting of the keyword **assign followed by a Boolean** expression.

Example: Write HDL Description of the function

F=x ′y z ′+ x ′y ′z

- // Verilog model: Circuit with Boolean expressions

```
module Circuit_Boolean_CA ( F, X, Y, Z);

output  F;

input X, Y, Z;

assign F = ((!X) && Y && (!Z) ) || ((!X)  && (!Y) && Z);

endmodule
```

Minterms

- A binary variable may appear either in its normal form (*x) or in its complement form (x ’).*
- Now consider two binary variables *x and y combined with an AND operation. Since each* variable may appear in either form, there are four possible combinations: *x’y’, x’y, xy’,* and *xy. Each of these four AND terms is called a min-term, or a standard product. In a* similar manner, *n variables can be combined to form* 2^n *min-terms.*
- Each min-term is obtained from an AND term of the *n variables, with each variable being* primed if the corresponding bit of the binary number is a 0 and un-primed if a 1.

Maxterms

- *n variables forming an OR term, with each variable being primed* or un-primed, provide 2^n *possible combinations, called max-terms, or standard sums.*
- Now consider two binary variables *x and y combined with an OR operation. Since each* variable may appear in either form, there are four possible combinations: *x'+y', x'+y, x+y',* and *x+y. Each of these four OR terms is called a maxterm. In a* similar manner, *n variables can be combined to form* 2^n *maxterms*.
- Each maxterm is obtained from an OR term of the *n variables, with each variable being* primed if the corresponding bit of the binary number is a 1 and unprimed if a 0.

Min-terms and Max-terms 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

Canonical form

- A truth table consists of a set of inputs and outputs. If there are ‘n’ input variables, then there will be 2^n possible combinations with zeros and ones. So the value of each output variable depends on the combination of input variables. So, each output variable will have ‘1’ for some combination of input variables and ‘0’ for some other combination of input variables.
- Therefore, we can express each output variable in following two ways.
  - Sum of minterms form (SOP)
  - Product of maxterms form (POS)
- **Boolean functions expressed as a sum of minterms or product of maxterms are said to be in *canonical form*.**

Sum of Minterms

A Boolean function can be expressed algebraically from a given truth table by forming a min-term for each combination of the variables that produces a 1 in the function and then taking the OR of all those terms.

$f1 = x' y' z + x y' z' + x y z$

$= m1 + m4 + m7$

$= \sum(m1 , m4 , m7)$

$= \sum m(1,4,7)$

x	y	z	Function f_1
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

Ex. Express the Boolean function F = A + B′ C as a sum of minterms.

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

$= AB + AB' + AB'C + A'B'C$

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

$= ABC + ABC' + AB'C + AB'C' + AB'C + A'B'C$

Combining all term

$F = A'B'C + AB'C' + AB'C + ABC' + ABC$

$= m_1 + m_4 + m_5 + m_6 + m_7$

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

The summation symbol $\sum$ stands for the ORing of terms; the numbers following it are the indices of the minterms of the function. The letters in parentheses following *F form* a list of the variables in the order taken when the minterm is converted to an AND term.

Product of Maxterms

Now consider the complement of a Boolean function.

The complement of f_1 *is*

$f_1' = x'y'z' + x'yz' + x'yz + xy'z + xyz'$

If we take the complement of f_1',

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

$= M_0 \cdot M_2 \cdot M_3 \cdot M_5 \cdot M_6$

$= \Pi (M_0, M_2, M_3, M_5, M_6)$

$= \Pi (0,2,3,5,6)$

Form a maxterm for each combination of the variables that produces a 0 in the function, and then form the AND of all those maxterms.

x	y	z	Function f_1
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

Ex. Express the Boolean function F= xy + x'z as a product of maxterms.

Using distributive law: $F = xy + x'z = (xy+x')(xy+z)$

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

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

Each OR term missing one variable

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

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

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

Combining all the terms

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

$$= M_0M_2M_4M_5$$

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

Conversion between canonical forms

Ex. Boolean expression: F = xy + x′z

xy = 11 or xz = 01

sum of minterms is

F(x, y, z) = Σ(1, 3, 6, 7)

Product of maxterms is

F(x, y, z) = Π(0, 2, 4, 5)

Take complement of F by DeMorgan's theorem

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

To convert from one canonical form to another, interchange the symbols ∑ and ∏ and list those numbers missing from the original form.

Standard SOP and POS forms

- We discussed two canonical forms of representing the Boolean outputs.
- Another way to express Boolean functions is in *standard form. In this configuration*, the terms that form the function may contain one, two, or any number of literals. There are two types of standard forms: the sum of products and products of sums.

Standard SOP form

- Standard SOP form means **Standard Sum of Products** form. In this form, each product term need not contain all literals. So, the product terms may or may not be the minterms.
- The sum of products is a Boolean expression containing AND terms, called product terms, with one or more literals each. The sum denotes the ORing of these terms. An example of a function expressed as a sum of products is

  $F1 = y + xy + xy'z$

- The expression has three product terms, with one, two, and three literals. Their sum is, in effect, an OR operation.

Standard POS form

- Standard POS form means **Standard Product of Sums** form.

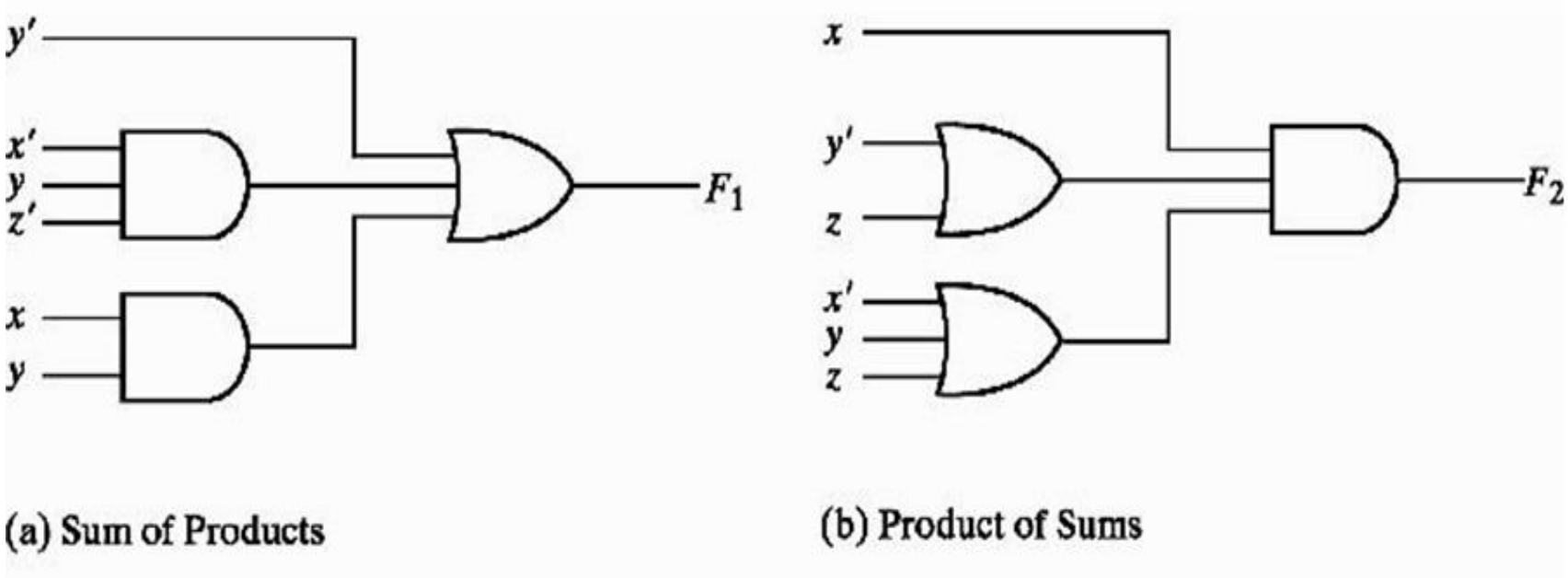
- *A product of sums is a Boolean expression containing OR terms, called sum terms.*

- Each term may have any number of literals. The *product denotes the ANDing of these* terms.

Three- and two-level implementation of Standard forms

- Another way to express Boolean functions is in standard form.

1. Sum of products(SOP): $F_1 = y' + xy + x'yz'$
2. Product of sums(POS): $F_2 = x(y' + z)(x' + y + z')$

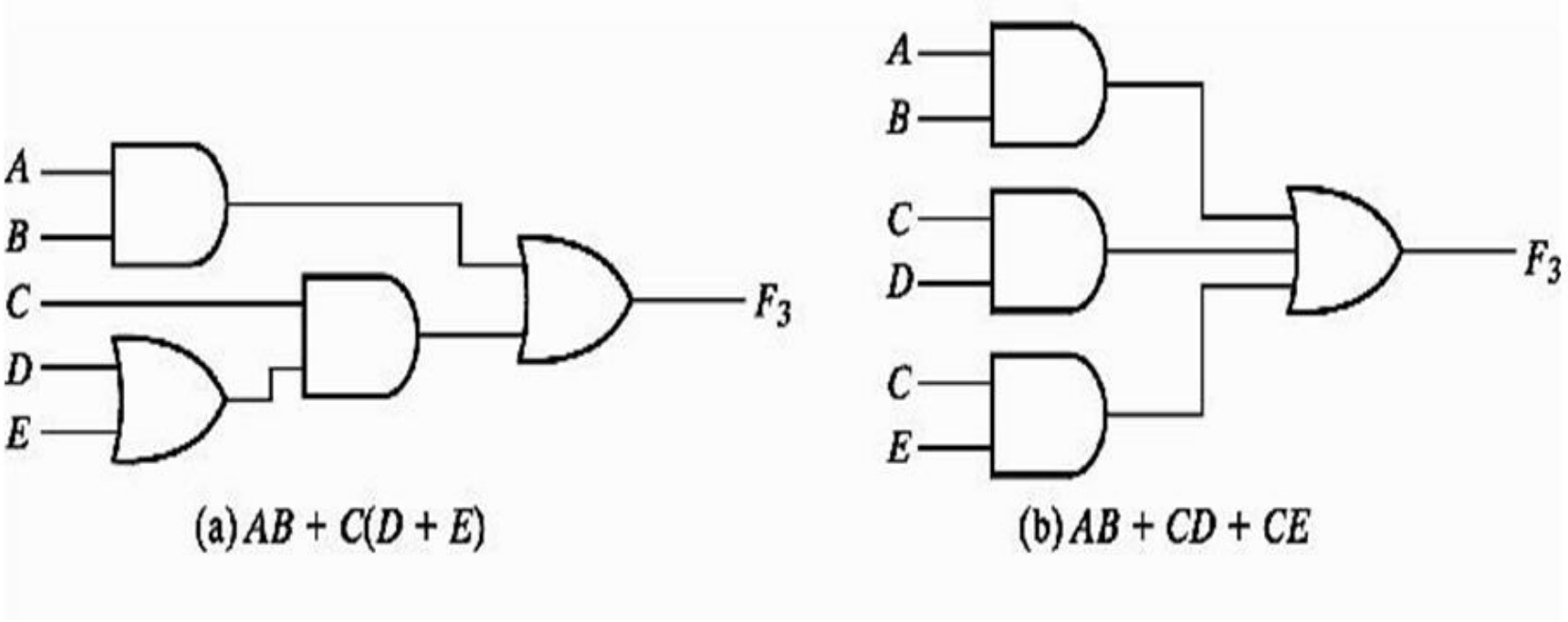


(a) Sum of Products

(b) Product of Sums

Standard forms

- F_3 is a non-standard form, neither in SOP nor in POS.
- F_3 can change to a standard form by using distributive law.



HDL Description of Boolean Function

- Write HDL Description of the function

  F=AB+CD+CE

- // Verilog model: Circuit with Boolean expressions

```
module Circuit_CA ( F, A, B, C,D,E);

output  F;

input A, B, C, D, E;

assign F = (A && B) || (C && D) || (C && E) ;

endmodule
```

HDL Description of Boolean Function

- Write HDL Description of the function

  F=AB+C(D+E)

- // Verilog model: Circuit with Boolean expressions

```
module Circuit_CA ( F, A, B, C, D, E);

output  F;

input A, B, C, D, E;

assign F = (A && B) || (C && (D || E)) ;

endmodule
```