# Canonical and Standard Forms

## Minterms and Maxterms

→ A binary variable may appear either in its normal form (x) or in its complement form (x').

→ 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 minterm, or a standard product.

→ In a same manner, n variables can be combined to form $2^n$ minterms. The $2^n$ different minterms may be determined by binary numbers from 0 to $2^n - 1$.

→ Each minterm is obtained from an AND term of the n variables, with each variable being primed (or ~~unprimed~~) if the corresponding bit of the binary number is a 0 and unprimed if a 1.

→ A symbol for each minterm is of the form $m_j$, where the subscript j denotes the decimal equivalent of the binary number of the minterm designated.

→ In a similar way, n variables forming an OR term, with each variable being primed or unprimed, provide $2^n$ possible combinations, called maxterms or standard sums.

→ A symbol for each maxterm is of the form $M_j$, where the subscript j denotes the decimal equivalent of the binary number of the maxterm designated.

→ Each maxterm is obtained from an OR term of the n variables, with each variable being unprimed if the corresponding bit is a 0 and primed if a 1.

→ Each maxterm is the complement of its corresponding minterm and viceversa.

| x | y | z | Minterms Term | Minterms Designation | Maxterms Term | Maxterms 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 | $xyz$ | $m_7$ | $x'+y'+z'$ | $M_7$ |

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

→ Consider the functions

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

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

| 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 |

→ The complement of the Boolean function can be read from the truth table by forming a minterm for each combination that produces a 0 in the function and then ORing those terms.

→ To express a Boolean function as a product of maxterms, it must first be brought into a form of OR terms. This may be done by using the distributive law $x + yz = (x+y)(x+z)$

→ Then any missing variable $x$ in each OR term is ORed with $xx'$.

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

$$F = xy + x'z = (xy + x')(xy + z)$$
$$= (x'+x)(x'+y)(z+x)(z+y)$$
$$= (x'+y+zz')(x+z+yy')(y+z+xx')$$
$$F = (x'+y+z)(x'+y+z')(x+y+z)(x+y'+z)(x+y+z)(x'+y+z)$$
$$F = (x+y+z)(x+y'+z)(x'+y+z)(x'+y+z')$$
$$= M_0 M_2 M_4 M_5$$

→ A convenient way to express this function is:

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

→ The product symbol $\Pi$ denotes the ANDing of maxterms, the numbers are the indices of the maxterms of the function

## Conversion between Canonical Forms

→ The complement of a function expressed as the sum of minterms equals the sum of minterms missing from the original function.

→ The original function is expressed by those minterms which make the function equal to 1, whereas its complement is a 1 for those minterms for which the function is a 0.

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

→ The complement of this function can be expressed as

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

→ Now if we take complement of $F'$ by DeMorgan's theorem, we obtain $F$ in a different form:

$$F = (m_0 + m_2 + m_3)' = m_0' \cdot m_2' \cdot m_3' = M_0 M_2 M_3$$
$$= \Pi(0, 2, 3)$$

→ From the definition of minterms $m_j' = M_j$
i.e., the maxterm with subscript j is a complement of the minterm with the same subscript j and vice versa.

→ To convert from one canonical form to another, interchange the symbols $\Sigma$ and $\Pi$ and list those numbers missing from the original form.

→ In order to find the missing terms, one must realize that the total number of minterms or maxterms is $2^n$ where n is number of binary variables in the function.

→ A Boolean function can be converted from an algebraic expression to a product of maxterms by means of truth table and the canonical conversion procedure.

→ Consider the Boolean function $F = xy + x'z$

| 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 |

Minterms

Maxterms

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

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

## Standard Forms

→ 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.

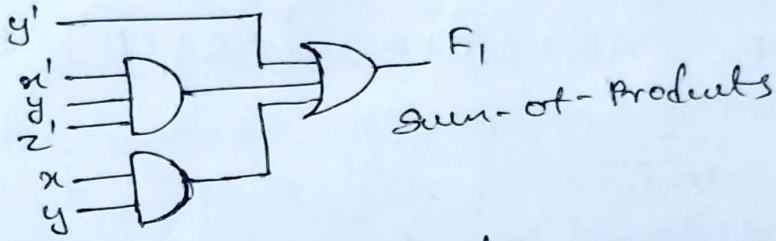
→ There are two types of standard forms: the sum of products and products of sums.

→ The sum of products is a Boolean expression containing AND terms, called products terms, with one or more literals each.

→ The sum denotes the ORing of these terms.

→ Consider $F_1 = y' + xy + x'yz'$, is an example of function expressed as a sum of products

→ The logic diagrams of a sum-of-products expression consists of a group of AND gates followed by a single OR gate. This circuit configuration is referred to as a two-level implementation.




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

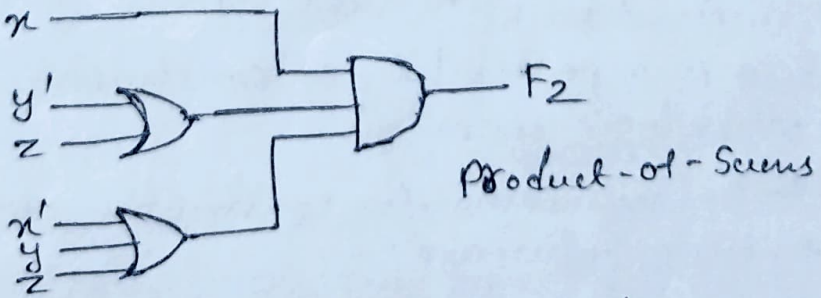
→ Each term may have any number of literals.

→ The product denotes the ANDing of these terms.

→ Consider $F_2 = x(y'+z)(x'+y+z')$, is an example of a function expressed as a product of sums

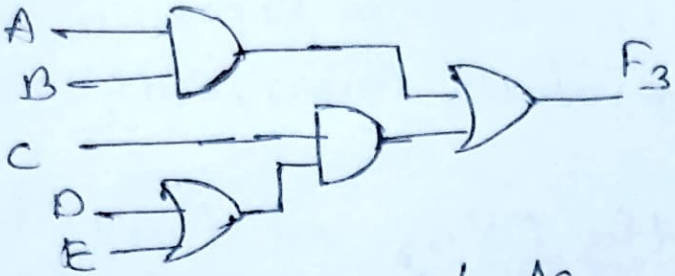
→ The gate structure of the product-of-sums expression consists of a group of OR gates for the sum terms, followed by an AND gate.

→ This standard type of expression results in a two-level structure of gates.




→ A Boolean function may be expressed in a nonstandard form.

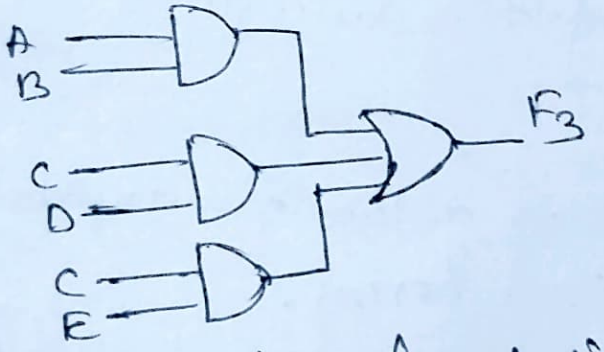
→ For example, the function $F_3 = AB + C(D+E)$ is neither in sum-of-products nor in product-of-sums form.




→ The implementation of this expression requires two AND gates and two OR gates. There are three levels of gating in this circuit

→ It can be changed to a standard form by using the distributive law to remove the paratheses:

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




→ In general, a two-level implementation is preferred because it produces the least amount of delay through the gates when the signal propogates from the inputs to the output.