GATE-LEVEL MINIMIZATION

CANONICAL AND STANDARD FORMS

Boolean Functions

- Boolean algebra is an algebra that deals with binary variables and logic operations.
- A Boolean function described by an algebraic expression consists of binary variables, the constants 0 and 1, and the logic operation symbols.
- For a given value of the binary variables, the function can be equal to either 1 or 0.

- As an example, consider the Boolean function

  $$F_1 = x + y'z$$

- The function F_1 is equal to 1 if x is equal to 1 or if both y' and z are equal to 1. F_1 is equal to 0 otherwise. Therefore, $F_1 = 1$ if $x = 1$ or if $y = 0$ and $z = 1$.

- A Boolean function can be represented in a truth table.
- The number of rows in the truth table is 2^n, where n is the number of variables in the function.
- The binary combinations for the truth table are obtained from the binary numbers by counting from 0 through $2^n - 1$.

Table shows the truth tables for the function $F_1 = x + y'z$ and $F_2 = x'y'z + x'yz + xy'$

x	y	z	F_1	F_2
0	0	0	0	0
0	0	1	1	1
0	1	0	0	0
0	1	1	0	1
1	0	0	1	1
1	0	1	1	1
1	1	0	1	0
1	1	1	1	0

Example: **Draw the truth table for the Boolean function: $F = A + BC$.**

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

Example: **Draw the truth tables for 3-input Universal Gates.**

A	B	C	$\overline{ABC}$	$\overline{A+B+C}$
0	0	0	1	1
0	0	1	1	0
0	1	0	1	0
0	1	1	1	0
1	0	0	1	0
1	0	1	1	0
1	1	0	1	0
1	1	1	0	0

Canonical and Standard Forms

Minterms and Maxterms

- 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 *minterm*, or a *standard product*.

- In a similar manner, *n* variables can be combined to form 2^n minterms.
- The binary numbers from 0 to $2^n - 1$ are listed under the *n* variables.
- Each minterm 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 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 fashion, n variables forming an OR term, with each variable being primed or unprimed, provide 2^n possible combinations, called *maxterms*, or *standard sums*.

- Any 2^n maxterms for n variables may be determined similarly.
- It is important to note that (1) 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, and (2) each maxterm is the complement of its corresponding minterm and vice versa.

$$\overline{m_j} = M_j \quad \text{or} \quad \overline{M_j} = m_j$$

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

- 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 minterms.
- by 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.

Example:

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

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

$$= m_1 + m_4 + m_7$$

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

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

$$= M_0 . M_2 . M_3 . M_5 .M_6$$

$$f_2 = x'yz + xy'z + xyz' + xyz$$

$$= m_3 + m_5 + m_6 + m_7$$

$$f_2 = (x + y + z)(x + y + z')$$

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

$$= M_0 . M_1 . M_2 . M_4$$

Canonical Forms

- Boolean functions expressed as a sum of minterms or product of maxterms are said to be in *canonical form* .
- For n binary variables, one can obtain 2^n distinct minterms or maxterms and that any Boolean function can be expressed as a sum of minterms or product of maxterms.

Sum of Minterms

- The minterms whose sum defines the Boolean function are those which give the 1's of the function in a truth table.
- If the function is not in this form, it can be made so by first expanding the expression into a sum of AND terms. Each term is then inspected to see if it contains all the variables. If it misses one or more variables, it is ANDed with an expression such as $x + x'$, where x is one of the missing variables.

Product of Maxterms

- The maxterms whose product defines the Boolean function are those which give the 0's of the function in a truth table.
- 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'.

Example: Express the Boolean function $F = A + B'C$ as a sum of minterms.

The function has three variables: A, B, and C.

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

$$= ABC + ABC' + AB'C + AB'C'$$

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

Combining all terms, we have

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

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

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

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

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

First, convert the function into OR terms

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

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

The function has three variables: x, y, and z.

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

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

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

Combining all the terms, we have

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

$$= M_0 . M_2 . M_4 . M_5$$

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

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.
- This is because 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.

Example:

- Consider the function

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

- This function has a complement that can be expressed as

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

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

$$F = (m_0 + m_2 + m_3)' = m'_0 \,.\, m'_2 \,.\, m'_3$$

$$= M_0 M_2 M_3 = \prod(0, 2, 3)$$

- From the definition of minterms and maxterms as shown in Table

$$m'_j = M_j$$

- That is, the maxterm with subscript *j* is a complement of the minterm with the same subscript *j* and vice versa.

Standard Forms

- The two canonical forms of Boolean algebra are basic forms that one obtains from reading a given function from the truth table.
- These forms are very seldom the ones with the least number of literals, because each minterm or maxterm must contain, by definition, *all* the variables, either complemented or uncomplemented.

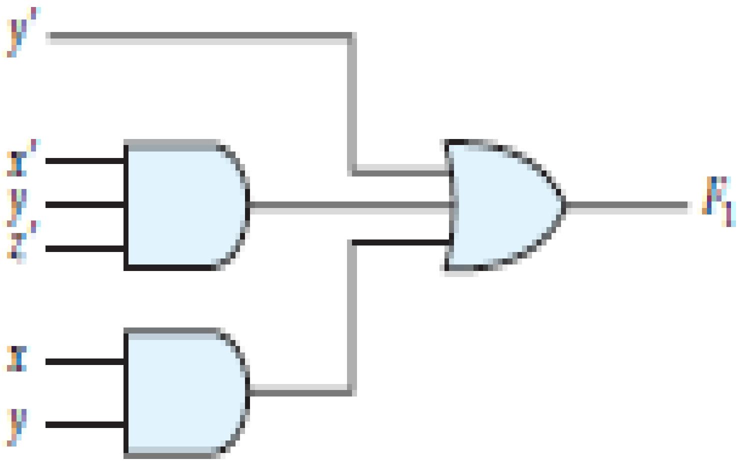
- Another way to express Boolean functions is in *standard* form.
- In this configuration, the terms that form the function may contain one, two, or any number of literals.
- There are two types of standard forms: the sum of products and products of sums.

- The *sum of products* is a Boolean expression containing AND terms, called *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 $F_1 = y' + xy + x'yz'$
- The expression has three product terms, with one, two, and three literals. Their sum is, in effect, an OR operation.

- 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.
- An example of a function expressed as a product of sums is $F_2 = x(y' + z)(x' + y + z')$
- This expression has three sum terms, with one, two, and three literals. The product is an AND operation.

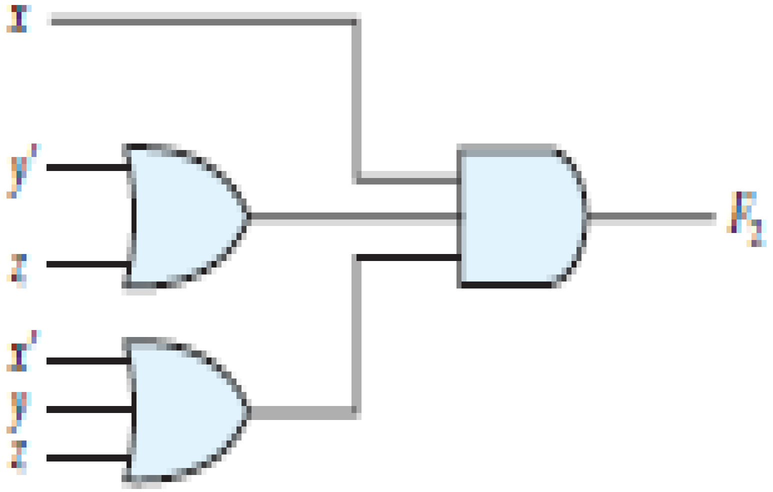
Sum of Products

$$F_1 = y' + xy + x'yz'$$



Product of Sums

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



Example: **Convert each of the following expressions into sum of products and product of sums:**

1. $(AB + C)(B + C'D)$
2. $x' + x(x + y')(y + z')$

Solution:

1. Sum of Products

$$(AB + C)(B + C'D)$$

$$= ABB + ABC'D + BC + CC'D$$

$$= AB + ABC'D + BC$$

$$= AB(1 + C'D) + BC = AB + BC$$

Product of Sums

$$(AB + C)(B + C'D)$$

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

Solution:

2. Sum of Products

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

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

$= x' + xxy + xxz' + xy'z'$

$= x' + xy + xz' + xy'z'$

$= x' + xy + xz' (1 + y') = x' + xy + xz'$

Product of Sums

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

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

$= 1 . 1 . (x' + y + z')$

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

Example: **Express each function in sum-of-minterms and product-of-maxterms form.**

1. $(xy + z)(y + xz)$
2. $(A' + B)(B' + C)$

Solution:

1. $(xy + z)(y + xz)$

$= xyy + xyxz + yz + xzz$

$= xy + xyz + yz + xz$

$= xy(z + z') + (x + x')yz + x(y + y')z + xyz$

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

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

$= m_3 + m_5 + m_6 + m_7$

$F(x, y, z) = \sum(3, 5, 6, 7) = \prod(0, 1, 2, 4)$

Solution:

2. $(A' + B)(B' + C)$

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

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

$= M_4 \,.\, M_5 \,.\, M_2 \,.\, M_6$

$F(A, B, C) = \prod(2, 4, 5, 6) = \sum(0, 1, 3, 7)$

or

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

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

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

$= m_1 + m_0 + m_3 + m_7$

$F(A, B, C) = \sum(0, 1, 3, 7) = \prod(2, 4, 5, 6)$

Example: **Convert each of the following to the other canonical form:**

1. $F(x, y, z) = \sum(1, 3, 5)$
2. $F(x, y, z) = \prod(0, 3, 6, 7)$
3. $F(A, B, C, D) = \sum(0, 2, 6, 11, 13, 14)$
4. $F(A, B, C, D) = \prod(0, 1, 3, 4, 6, 8, 11, 12)$

Solution:

1. $F(x, y, z) = \sum(1, 3, 5)$
   $= \prod(0, 2, 4, 6, 7)$
2. $F(x, y, z) = \prod(0, 3, 6, 7)$
   $= \sum(1, 2, 4, 5)$
3. $F(A, B, C, D) = \sum(0, 2, 6, 11, 13, 14)$
   $= \prod(1, 3, 4, 5, 7, 8, 9, 10, 12, 15)$
4. $F(A, B, C, D) = \prod(0, 1, 3, 4, 6, 8, 11, 12)$
   $= \sum(2, 5, 7, 9, 10, 13, 14, 15)$

Example: **Express the complement of the following functions in sum-of-minterms form:**

1. $F(x, y, z) = \sum(0, 3, 6, 7)$
2. $F(A, B, C, D) = \sum(2, 4, 7, 10, 12, 14)$
3. $F(A, B, C) = \prod(3, 5, 7)$
4. $F(w, x, y, z) = \prod(0, 1, 3, 4, 6, 8, 11, 12)$

Solution:

1. $F(x, y, z) = \sum(0, 3, 6, 7) = \prod(1, 2, 4, 5)$

   $F'(x, y, z) = \sum(1, 2, 4, 5) = \prod(0, 3, 6, 7)$
2. $F(A, B, C, D) = \sum(2, 4, 7, 10, 12, 14)$

   $F'(A, B, C, D) = \sum(0, 1, 3, 5, 6, 8, 9, 11, 13, 15)$
3. $F(A, B, C) = \prod(3, 5, 7) = \sum(0, 1, 2, 4, 6)$

   $F'(A, B, C) = \sum(3, 5, 7) = \prod(0, 1, 2, 4, 6)$
4. $F(w, x, y, z) = \prod(0, 1, 3, 4, 6, 8, 11, 12)$

   $F'(w, x, y, z) = \sum(0, 1, 3, 4, 6, 8, 11, 12)$