Boolean Algebra

Complement of a 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

Finding the complement of the functions (Method 1: *By De-Morgan's laws*)

- $F1 = x'yz' + x'y'z$

  $F1' = (x'yz' + x'y'z)' = (x'yz')'(x'y'z)'$

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

- $F2 = x(y'z' + yz)$

  $F2' = [x(y'z' + yz)]' = x' + (y'z' + yz)' = x' + (y'z')(yz)'$

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

  $= x' + yz' + y'z$

Finding the complement of the functions (Method 2:By Dual & complement)

- A simpler procedure for deriving the complement of a function is to
  - take the dual of the function
  - then complement each literal.
- $F1 = x'yz' + x'y'z$
  - The dual of $F1$ *is* $(x' + y + z')(x' + y' + z)$.
  - Complement each literal: $(x + y' + z)(x + y + {'z}) = F1'$
- $F2 = x(y'z' + yz)$
  - The dual of $F2$ *is* $x + (y' + z')(y + z)$.
  - Complement each literal: $x' + (y + z)(y' + {'z}) = F2'$

Dual of a function is obtained from the interchange of AND and OR operators and 1's and 0's.

Minterm or Standard Product

- A binary variable may appear either in its normal form *(x) or in its complement form (x).*
- *n variables forming an AND term, with each variable being primed* or unprimed, provide 2^n *possible combinations, called minterm, or a standard product*
- If 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

			Minterms	
x	**y**	**z**	**Term**	**Designation**
0	0	0	$x'y'z'$	m_0
0	0	1	$x'y'z$	m_1
0	1	0	$x'yz'$	m_2
0	1	1	$x'yz$	m_3
1	0	0	$xy'z'$	m_4
1	0	1	$xy'z$	m_5
1	1	0	xyz'	m_6
1	1	1	xyz	m_7

Maxterm or Standard Sum

- A binary variable may appear either in its normal form (*x) or in its complement form (x).*
- *n variables forming an OR term, with each variable being primed* or unprimed, provide 2^n *possible combinations, called maxterm, or a standard sum*
- If The binary numbers from 0 to 2^n - 1 are listed under the n variables
  - Each minterm is obtained from an OR term of the n variables, with each variable being unprimed if the corresponding bit of the binary number is a 0 and primed if a 1

			Maxterms	
x	***y***	***z***	**Term**	**Designation**
0	0	0	$x + y + z$	M_0
0	0	1	$x + y + z'$	M_1
0	1	0	$x + y' + z$	M_2
0	1	1	$x + y' + z'$	M_3
1	0	0	$x' + y + z$	M_4
1	0	1	$x' + y + z'$	M_5
1	1	0	$x' + y' + z$	M_6
1	1	1	$x' + y' + z'$	M_7

Boolean function using minterms & maxterms & canonical form

- 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.

    (or)

  - forming a maxterm for each combination of the variables that produces a 0 in the function and then taking the AND of all those terms.
- Boolean functions expressed as a ***sum of minterms*** or ***product of maxterms*** are said to be in ***canonical form***

Example

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

- Sum of minterms:

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

$= m1 + m4 + m7$

- *Product of maxterms:*

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

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

- *Complement of f1:*

$f1' = x'y'z' + x'yz' + x'yz + xy'z + xyz'$ *(minterms not present in f1)*

- Complement of *f1'*:

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

$= M_0 \,.\, M_2 \,.\, M_3 \,.\, M_5 \,.\, M_6 = f1$

Sum of Minterms

- It is sometimes convenient to express a Boolean function in its sum-of-minterms form.
- If the function is not in minterm 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.*
- The summation symbol g stands for the ORing of terms
- An alternative procedure for deriving the minterms of a Boolean function
  - obtain the truth table of the function directly from the algebraic expression and then read the minterms from the truth table.

Sum of Minterms - Example

- Express the Boolean function $F = A + BC'$ *as a sum of minterms.*
  - The function has three variables: *A, B, and C.*
  - *The first term A is missing two variables;*
    - $A = A(B + B') = AB + AB'$
  - The above function is still missing one variable;
    - $A = AB(C + C') + AB'(C + C') = ABC + ABC' + AB'C + AB'C'$
  - The second term *BC is missing one variable;*
    - $B'C = B'C(A + A') = AB'C + A'B'C$
  - Combining all terms:
    - $F = A + BC = ABC + ABC' + AB'C + AB'C' + A'B'C$
  - Rearranging the minterms in ascending order
    - $F = ABC + ABC + ABC + ABC + ABC$

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

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

But *AB'C appears twice, hence keep only* one of those occurrences.