3

Boolean Algebra and Reduction Techniques

Boolean Algebraic Laws

Commutative Law

$$A + B = B + A \quad \text{and} \quad A \cdot B = B \cdot A$$

Associative Law

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

and

$$A \cdot (B \cdot C) = (A \cdot B)\, C = A\, B\, C$$

Distributive Law

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

$$(A + B)(C + D) = AC + AD + BC + BD$$

Boolean Algebraic Theorems

AND-Operation Theorem

$$A \cdot A = A \qquad A \cdot 0 = 0$$

$$A \cdot 1 = A \qquad A \cdot \bar{A} = 0$$

Involution Theorem

$$(A')' = \bar{\bar{A}} = A$$

OR-Operation Theorem

$$A + A = A \qquad A + 0 = A$$

$$A + 1 = A \qquad A + \bar{A} = 1$$

De Morgan's Theorem

$$\overline{(A_1 \cdot A_2 \cdot A_3 \cdot \ldots . A_n)} = \bar{A}_1 + \bar{A}_2 + \bar{A}_3 + \ldots . + \bar{A}_n$$

$$\overline{(A_1 + A_2 + A_3 + \cdots + A_n)} = \bar{A}_1 \cdot \bar{A}_2 \cdot \bar{A}_3 \cdots \bar{A}_n$$

Transposition Theorem

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

Distribution Theorem

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

Consensus Theorem

- Used to eliminate redundant term.
- It is applicable only when if a boolean function,
  1. Contains 3-variables
  2. Each variable used 2-times
  3. Only one variable is in complemented or uncomplemented form.
  4. Then the related terms to that complemented or uncomplemented variable is the answer.

  Ex: $AB + \bar{A}C + BC = AB + \bar{A}C$

Boolean Algebraic Theorems

Theorem No.	Theorem
1.	$(A + B) \cdot (A + \bar{B}) = A$
2.	$AB + \bar{A}C = (A + C)(\bar{A} + B)$
3.	$(A + B)(\bar{A} + C) = AC + \bar{A}B$
4.	$AB + \bar{A}C + BC = AB + \bar{A}C$
5.	$(A + B)(\bar{A} + C)(B + C) = (A + B)(\bar{A} + C)$
6.	$\overline{A \cdot B \cdot C \cdot \ldots} = \bar{A} + \bar{B} + \bar{C} + \ldots$
7.	$\overline{A + B + C + \ldots} = \bar{A} \cdot \bar{B} \cdot \bar{C} \ldots$

Duality Theorem

- "Dual expression" is equivalent to write a negative logic of the given boolean relation. For this we,

1. Change each OR sign by an AND sign and vice-versa.
2. Complement any '0' or '1' appearing in expression.
3. Keep literals as it is.

Note:

- If a dual of an expression results the expression itself then it is called "self dual expression".
- For N-variables, maximum possible Self-Dual Function = $(2)^{2^{n-1}} = 2^{(2^n/2)}$.
- With N-variables, maximum possible distinct logic functions = 2^{2^n}.

Complementary Theorem

For obtaining complement expression we

1. Change each OR sign by AND sign and vice-versa.
2. Complement any '0' or '1' appearing in expression.
3. Complement the individual literals.

Boolean Function Representation

Canonical Form

All the terms contains each literal.

Ex.: $F(A,B,C) = \bar{A}BC + ABC + \bar{A}B\bar{C}$

Standard Form

All the term do not have each literals.

Ex.: $F(A,B,C) = \bar{A} + BC + A\bar{B}C$

Note:

A binary-variable is called "LITERALS".

Minterms and Maxterms

- n-binary variables have 2^n possible combinations and each of these possible combination is called "Minterm or Standard Product form".
- "Maxterm" is the complement of corresponding "Minterm" i.e. $\boxed{M = \bar{m}}$; with don't care conditions remain same.

Reduction Techniques

SOP (Sum of Product)

In SOP form each product term is known as min term. SOP form is used when output is logic 1.

e.g. $\underbrace{AB\bar{C}}_{\text{Min term}} + A\bar{B}C + ABC$

POS (Product of SUM)

In POS form each product term called as max term. POS form is used when output is 0 logic '0'.

e.g. $\underbrace{(A+B+C)}_{\text{Max term}} \cdot (A+\bar{B}+C) \cdot (\bar{A}+B+C)$

Note:

- AND-OR Logic ≡ NAND-NAND Logic and is used in SOP.
- OR-AND Logic ≡ NOR-NOR Logic and is used in POS.

Dual

Dual expression is used to convert positive logic into negative logic or vice-versa.

Procedure

1. AND logic $\longleftrightarrow$ OR
2. 1 $\longleftrightarrow$ 0
3. Keep variable as it is.

Note:

- Two time dual results in same expression.
- Positive logic AND = Negative logic OR and vice-versa.

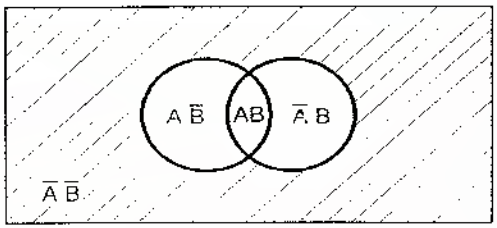
Complement

1. AND $\longleftrightarrow$ OR
2. 1 $\longleftrightarrow$ 0
3. Complement each variable.

Remember:

- With 'n' variables maximum possible logical expressions are 2^{2^n}.
- With 'n' variables maximum possible self dual expressions are $2^{2^{n-1}}$.

Venn Diagram



Here, the AND operation is considered as an intersection and the OR operation is considred as a union.

K-Map

Complete Simplification Rules (K-Map)

- Construct the K-map and place 1's in the cells corresponding to the 1's in the truth table. Place 0's in the remaining cells.
- Examine the map for adjacent 1's and loop those 1's which are not adjacent to any other 1's. These are called isolated 1's.
- Next, look for those 1's which are adjacent to only one other 1. Loop any pair containing such a 1.
- Loop any octet even it contains some 1's that have already been looped.
- Loop any quad that contains one or more 1's which have not already been looped, making sure to use the minimum number of loops.
- Loop any pairs necessary to include any 1's that have not yet been looped, making sure to use the minimum number of loops.
- Form the OR sum of all the terms generated by each loop.
- In an n-variable Karnaugh-map there are 2^n cells.

Don't Care Condition

- Some logic circuit can be designed so that there are certain input condition for which there is no specified output levels, usually because these conditions will never occur.
- So, a circuit designer is free to make the output for any "don't care" condition either a '0' or '1' in order to produce the simplest output expression.

Implicant

Each individual min-term in canonical SOP form is called implicant.

Prime Implicant

Prime implicant is an implicant, which is obtained by combining maximum possible adjacent cell in K-map.

Essential Prime Implicant

An essential prime implicant is a prime implicant in which one or more min term are unique i.e. which is obtain by combining only one way in K-map.

Digital Number Representation

- In unsigned magnitude representation with 'n' bits, the possible integer values are $[0 - (2^n - 1)]$.
- Extra bit → sign bit → MSB
  if MSB = 0 → +ve number
  if MSB = 1 → -ve number
- In a sign magnitude representation the range of number.

$$-(2^{n-1} - 1) \text{ to } +(2^{n-1} - 1)$$

Compliment: For base 'r' → (r – 1)'s compliment, r's compliment

Binary: 1's, 2's
Octal: 7's, 8's
Decimal: 9's, 10's
Hexadecimal: F's, 16's

1. To determine (r – 1)'s compliment, the given number is subtracted from the maximum possible number in given base.
2. To determine r's compliment, first write (r – 1)'s compliment then add 1 to the LSB (Least Significant Bit).

- In 1's complement representation, the positive number are represented similar to positive number in sign magnitude, but for representing negative number, first to write positive number and then take 1's complement of that.

  ❑ Range of 1's complement number

$$\boxed{-(2^{n-1}-1) \text{ to } +(2^{n-1}-1)}$$

  ❑ Range of 2's complement representation number

$$\boxed{-2^{n-1} \text{ to } +(2^{n-1}-1)}$$

Binary Arithmetic

- When both number have same sign then we add only magnitudes and use the sign as MSB.

1's complement addition

- When the numbers have different sign, keep one number as it is and 1's complement the other then we add magnitude only.
- If carry is present
  1. add carry to LSB
  2. sign of the result is sign of the uncomplemented number.
- If carry is not present
  1. take 1's complement the result.
  2. sign of the result is sign of the complemented number.

2's complement addition

In 2's compliment addition if any carry is present, then discard it.

Note:

- In 2's compliment there is only one way to represent '0'.
- 2's compliment of the 2's complimented number is equal to the number itself.
- In 2's compliment representation to extend the number of bits MSB bit is is always copied.

■■■■