GROUP 7

- EE24BTECH11012 - BHAVANISANKAR G S
- EE24BTECH11006 - ARNAV MAHISHI
- EE24BTECH11007 - ARNAV MAKARAND YADNOPAVIT
- EE24BTECH11010 - BALAJI B
- EE24BTECH11021 - ESHAN RAY

LOGIC DESIGN FUNDAMENTALS
UNIT 3 - BOOLEAN ALGEBRA

February 3, 2025

1 Multiplying Out and Factoring Expressions

Following theorems are useful in factoring and multiplying out (Very Useful for Boolean Expression Manipulation):

$$X(Y+Z) = XY + XZ \tag{1}$$
$$(X+Y)(X+Z) = X + YZ \tag{2}$$
$$(X+Y)(X'+Z) = XZ + X'Y \tag{3}$$

1.1 Proving

Theorem 23:

$$(X+Y)(X+Z) = X + XY + XZ + YZ \tag{4}$$
$$= X(1+Y+Z) + YZ \tag{5}$$
$$= X + YZ \tag{6}$$

Theorem 24:

$$(X+Y)(X'+Z) = 0 + XZ + YX' + YZ \tag{7}$$
$$\tag{8}$$

Putting X=0

$$X = 0 \tag{9}$$
$$Y = Y + YZ \tag{10}$$
$$Y = Y \tag{11}$$

Putting X=1

$$X = 1 \tag{12}$$
$$Z = Z + YZ \tag{13}$$
$$Z = Z \tag{14}$$

Hence Proved

1.2 Examples

- $(Q + AB')(C'D + Q')$

$$= QC'D + QQ' + AB'C'D + AB'Q' \tag{15}$$
$$= QC'D + AB'C'D \tag{16}$$

- (A + B + C′)$(A + B + D)](A + B + E)(A + D' + E)(A' + C)$

Using 23

$$= [(A + B + C'D)(A + B + E)][(A + D' + E)(A' + C)] \tag{17}$$

Using 23 and 24

$$= (A + B + C'DE)(AC + A'(D' + E)) \tag{18}$$
$$= [AC + ABC] + A'BD' + A'BE + A'C'DE \tag{19}$$
$$= AC + A'BD' + A'BE + A'C'DE \tag{20}$$

2 Map Method

- The complexity of digital logic gates depends on the algebraic expression of a Boolean function.
- While truth tables provide a unique representation, Boolean expressions can have multiple equivalent forms.
- Algebraic simplification is possible but lacks systematic rules.
- The Karnaugh map (K-map) method offers a visual way to minimize Boolean functions by grouping minterms efficiently. It allows users to derive simpler expressions, reducing the number of gates and inputs required in a circuit.

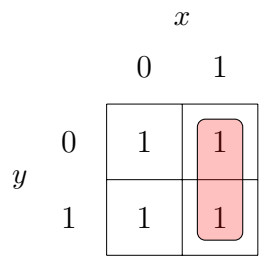
– The minimized expressions follow either the sum of products or product of sums form. Multiple minimal expressions may exist, and any valid solution is acceptable.

2.1 Two Variable k-map

– A two-variable Karnaugh map (K-map) consists of four squares, each representing a minterm of the two variables x and y .

– The map visually organizes these **minterms**, with rows and columns corresponding to the variable values. By marking the squares containing minterms of a given Boolean function, the K-map provides an intuitive way to represent any of the 16 possible Boolean functions for two variables.

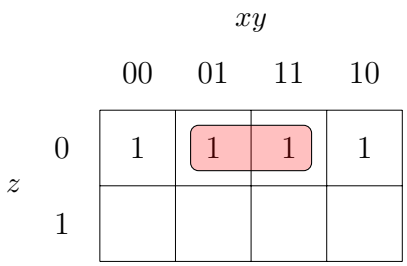
– For example, the function xy is represented by marking the minterm m_3 , while $x + y$ is represented by marking minterms m_1, m_2, and m_3 . This visualization helps in understanding Boolean expressions and their simplifications.



2.2 Three Variable k-Map

– A three-variable Karnaugh map (K-map) consists of eight squares, each representing a minterm of three binary variables.

– The minterms are arranged in a sequence similar to the Gray code , ensuring that only one variable changes between adjacent squares. Each square corresponds to a unique minterm, with variables labeled to indicate their presence or complement.



- The key property of the K-map is that adjacent squares differ by only one variable, allowing for Boolean simplifications.
- When two adjacent minterms are summed, the differing variable is eliminated, reducing the expression.
- For example, summing minterms m_5 and m_7 simplifies to xz . This property makes the K-map a powerful tool for minimizing Boolean functions efficiently.

3 Consensus Theorem

- The Consensus Theorem is very useful in simplifying Boolean expressions. Given an expression of the form:

$$XY + X'Z + YZ,$$

the term YZ is redundant and can be eliminated to form the equivalent expression:

$$XY + X'Z.$$

- The term that was eliminated is referred to as the consensus term .
- Given a pair of terms where a variable appears in one term and the complement of that variable in another, the consensus term is formed by multiplying the two original terms together while leaving out the selected variable and its complement.

For example:

- The consensus of ab and $a'c$ is bc.

- The consensus of abd and $b'de'$ is ade', since:

$$(ad)(de') = ade'.$$

- The consensus of $ab'd$ and $a'bd'$ is 0, meaning no further simplification can be done.

The consensus theorem, as given in Equation (2-18), is:

$$XY + X'Z + YZ = XY + X'Z. \tag{21}$$

3.1 Dual Form of Consensus Theorem

The dual form of the theorem is given by:

$$(X + Y)(X' + Z) = (X + Y)(Y + Z).$$

This follows a similar elimination principle.

3.2 Recognizing Consensus Terms

To apply the consensus theorem, identify pairs of terms where:

- One term contains a variable, and another contains its complement.
- The consensus term is formed by combining these terms and omitting the selected variable and its complement.

For example:

$$(a + b + c')(a + b + d')(b + c + d') = (a + b + c')(b + c + d').$$

Here, $a + b + d'$ is a consensus term and can be eliminated.

3.3 Example Application

- Given the Boolean expression:

$$AC'D + A'BD + BCD + ABC + ACD'.$$

- Using the Consensus Theorem, redundant terms can be removed step by step. The order of elimination affects the final simplification.

– In some cases, additional terms may need to be added before applying the theorem effectively.

– For example:

$$F = ABCD + B'CDE + A'B' + BCE'.$$

By adding the consensus term $ACDE$, we can further simplify the expression to:

$$F = A'B' + BCE' + ACDE.$$

Thus, the term $ACDE$ is essential and cannot be removed.

3.4 Conclusion

– The Consensus Theorem is a powerful tool in Boolean algebra that helps simplify expressions by removing redundant terms.

– Recognizing consensus terms and applying the theorem properly can significantly reduce the complexity of logical expressions, which is beneficial in digital circuit design and optimization.

– By understanding and utilizing this theorem effectively, one can achieve minimal representations of Boolean functions efficiently.

4 Algebraic Simplification of Switching Expressions

– Simplification of switching Boolean expressions algebraically are done based on the following properties -

$$XY + XY' = X \text{ Combining terms} \tag{22}$$
$$X + XY = X \text{ Eliminating terms} \tag{23}$$
$$X + X'Y = X + Y \text{ Eliminating literals} \tag{24}$$

– Simplification can, sometimes, also be done by,

$$\text{Addition of } xx' \tag{25}$$
$$\text{Multiplying by } x + x' \tag{26}$$
$$\text{Adding } yz \text{ to } xy + x'z \tag{27}$$

– **Examples :**

1. WX + XY + $X'Z' + WY'Z'$

$$
\begin{aligned}
F &= WX + XY + X'Z' + WY'Z' && (28)\\
&= WX + XY + X'Z' + WY'Z' + WZ' \text{ from (21)} && (29)\\
&= WX + XY + X'Z'WZ' \text{ from (23)} && (30)\\
&= WX + XY + X'Z' \text{ from (23)} && (31)
\end{aligned}
$$

5 Proving Validity of an Equation

Often, we need to determine whether an equation holds for all possible values of its variables. Several methods can be used to prove its validity:

- **Truth Table Method:** Construct a truth table and evaluate both sides of the equation for all combinations of variable values. This is not efficient for large numbers of variables.
- **Algebraic Manipulation:** Use Boolean theorems to transform one side of the equation until it matches the other side.
- **Reduction Method:** Independently simplify both sides of the equation to check if they reduce to the same expression.
- **Applying Reversible Operations:** Perform the same operation on both sides of the equation, ensuring that the operation is reversible. For example, complementing both sides is valid, but multiplying both sides by an expression is not allowed in Boolean algebra.

5.1 Disproving an Equation

To show that an equation is not valid , it is sufficient to find at least one assignment of variable values that results in different values on the two sides of the equation.

5.2 Useful Strategy for Proof

When using algebraic manipulation, the following strategy is helpful:

- Express both sides in sum of products or product of sums form.
- Compare the two sides for equality.
- Add or simplify terms logically to make both sides match.
- Continue until the expressions are identical.

5.3 Cancellation Law in Boolean Algebra

Some theorems of Boolean algebra do not hold in ordinary algebra and vice versa. One such case is the cancellation law:

5.4 Addition Cancellation Law

In ordinary algebra, the cancellation law states:

$$x + y = x + z \Rightarrow y = z$$

However, this is not true in Boolean algebra. For example, let:

$$x = 1, y = 0, z = 1$$

Then,

$$1 + 0 = 1 + 1, \quad \text{but} \quad 0 \neq 1$$

Thus, the **cancellation law does not hold in Boolean algebra**.

5.5 Multiplication Cancellation Law

In ordinary algebra, the cancellation law for multiplication states:

$$xy = xz \Rightarrow y = z, \quad \text{if } x \neq 0$$

In Boolean algebra, this law does not hold when $x = 0$. For example, if $x = 0, y = 0, z = 1$, then:

$$0 \cdot 0 = 0 \cdot 1, \quad \text{but} \quad 0 \neq 1$$

Because $x = 0$ occurs frequently in Boolean algebra, **the cancellation law for multiplication is not valid**.

5.6 Converse Statements

Although the cancellation laws are generally false in Boolean algebra, their converses are true:

$$y = z \Rightarrow x + y = x + z$$

$$y = z \Rightarrow xy = xz$$

- This means that adding the same term to both sides of a Boolean equation or multiplying both sides by the same term results in a valid equation.
- However, the reverse is not true: subtracting or dividing both sides by the same term is not permissible, as these operations are not reversible in Boolean algebra.