Telecom Australia
ENGINEERING TRAINING
210 Kingsway South Melbourne 3205

Training
Publication
ETP 0856
1980

# ELECTRONIC LOGIC PRINCIPLES 4



## 1. INTRODUCTION

1.1 The papers Electronic Logic Principles 1, 2 and 3 cover the operation of logic elements and only briefly discuss the role of Boolean statements in the theoretical description of logic element behaviour. This paper shows how Boolean statements may be manipulated by the use of Boolean Algebra to the extent that it is possible to analyse the operation of combinational logic circuits.

1.2 Boolean algebra is the mathematical technique used when considering problems of a logical nature. In 1847, an English mathematician, George Boole, developed the basic laws and rules for a mathematics which could be applied to problems of deductive logic. Until 1938 these techniques remained in use in the mathematical field. At this time a Bell Laboratories scientist, Claude Shannon, seeing the useful features of such an algebra, adapted it for analyzing multicontact networks such as those considered in telephone work.

In recent years electronic logic circuits are designed using Boolean algebra as a tool to simplify and manipulate the circuit in order to implement the required function.

## 2. BASIC DEFINITIONS

2.1 Summarising the basic definitions stated in 'Electronic Logic Principles 1',

:: All logic is confined to bi-valent or two state logic

:: A true state is logic 1, or contact closed

:: A false state is logic 0, or contact open

:: Letters of the alphabet A, B, C etc are used to represent the two state variable

:: Relationships between the variables use the symbols

(.) to represent AND

(+) to represent OR

(-) or (') to represent NOT

2.2 Combined Logic Functions. When a Boolean expression is used to represent a combined logic function the following rules are used in the order given:

:: Combine any bracketed terms with the type of gate indicated by the sign within the brackets

:: Combine any ANDed terms

:: Combine any ORed terms



ETS 20/0551

## 3. THE NEED FOR BOOLEAN MANIPULATION TECHNIQUES

3.1 Complex Logic Circuits. Logic circuits often contain more elements than would appear necessary to perform the required logic function. For example, Fig. 1 shows how three NAND gates are used to perform the OR function. In this example, a Boolean expression $\bar{A}.\bar{B}$ is developed which described the actual NAND gate combination used, and the equivalence of $\overline{\bar{A}.\bar{B}}$ to A+B is proven with truth tables 1 and 2.

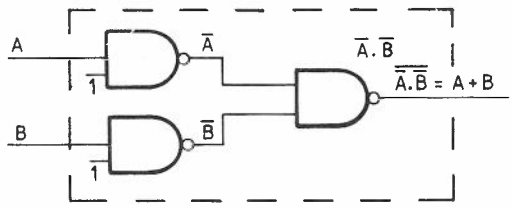



FIG. 1. NAND GATE COMBINATION WHICH PERFORMS THE OR FUNCTION

| Inputs | | Intermediate Points | | | Output |
|---|---|---|---|---|---|
| A | B | $\bar{A}$ | $\bar{B}$ | $\bar{A}.\bar{B}$ | $\overline{\bar{A}.\bar{B}}$ |
| 0 | 0 | 1 | 1 | 1 | 0 |
| 0 | 1 | 1 | 0 | 0 | 1 |
| 1 | 0 | 0 | 1 | 0 | 1 |
| 1 | 1 | 0 | 0 | 0 | 1 |

=

| Inputs | | Output |
|---|---|---|
| A | B | A+B |
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 1 |

TABLE 1. TRUTH TABLE FOR FIG. 1. TABLE 2. OR GATE TRUTH TABLE

In practice, however, the Boolean expression describing the actual gate combination in a complex circuit may not readily indicate the basic function of the circuit, and the basic function may not be known. For example, consider the circuit shown in Fig. 2. The Boolean expression $\overline{A + \bar{B}.\bar{C}}$ developed at the output of this circuit describes the actual circuit used, but the input conditions required to make the output significant are not immediately obvious. The operation of this complex circuit can be more readily understood if the basic function of the circuit is available.

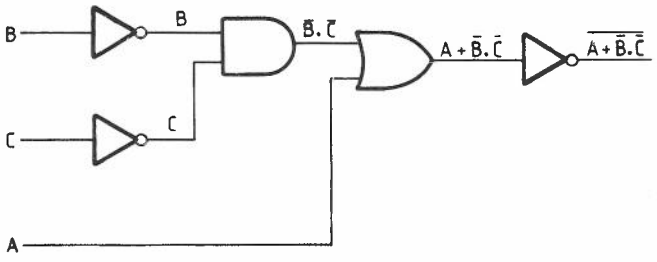



FIG. 2. CIRCUIT OF $\overline{A + \bar{B}.\bar{C}}$

3.2 Basic Function of Complex Logic Circuit. One method of obtaining the basic function of a complex Boolean expression is to apply known Boolean identities and laws to simplify the expression. For example, the application of Boolean identities and laws to $\overline{A + \bar{B}.\bar{C}}$ enables it to be simplified to $\bar{A}.(B + C)$. This expression reveals that a significant (logic 1) output is obtained in Fig. 2 when $\bar{A}$ is logic 1 AND when either B OR C is logic 1. Note that $\bar{A}$ is logic 1 when A is logic 0.

3.3 Functional Logic Diagram. A functional logic diagram can be drawn to represent a simplified Boolean expression. A functional logic diagram is a simplified diagram having the same truth table as the actual logic circuit from which it was derived. For example, Fig. 3 is a functional logic diagram of Fig. 2 as it represents the Boolean expression $\bar{A}.(B + C)$, which is the simplified expression of $\overline{A + \bar{B}.\bar{C}}$.

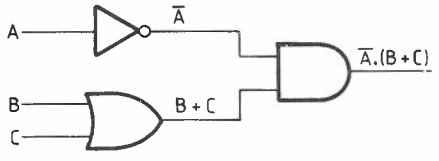



FIG. 3. FUNCTIONAL LOGIC DIAGRAM OF FIG. 2.

It should be realised that some logic circuits may already be in their simplest form. In these cases the function of the circuit is obtained directly from the Boolean expression describing the circuit.

4. BOOLEAN IDENTITIES

4.1 Boolean Identities. In this section of the paper it is intended to introduce some of the basic identities commonly used in converting complex Boolean expressions to their simplest function. A Boolean identity equates two expressions which are equal for all possible combinations of their variables. The equivalence of the two expressions in a Boolean identity can always be proven with the aid of truth tables.

Although the identities may seem self-evident and are, in fact, quite simple it is necessary to formally state them since the rules of Boolean algebra are built on them.

4.2 AND Function Identities. Consider a two input AND gate with one input permanently tied to logic 1 as shown in Fig. 4.

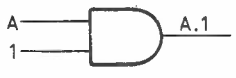



FIG. 4.

The identity derived from this expression is:-

$$A.1 = A \quad \ldots\ldots\ldots \quad (1)$$

This means that the output of the gate is always dependent on the logic condition at input A. This can be proven as follows. The variable A has two states, logic 1 and logic 0. Substituting these in the identity A.1 = A we get:-

1.1 = 1, when A is logic 1, and

0.1 = 0, when A is logic 0.

This identity is true because the output is equal to A for both states.

Now consider an AND gate with one input permanently at logic 0, as shown in Fig. 5. The output is always at logic 0 regardless of the condition of A. The identity derived from this circuit is:-

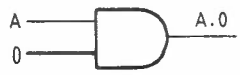



FIG. 5

$$A.0 = 0 \quad .. \quad .. \quad .. \quad .. \quad (2)$$

When both inputs of an AND gate have the same signal applied, as shown in Fig. 6, the identity applying to the circuit is:-

$$A.A = A \quad .. \quad .. \quad .. \quad .. \quad (3)$$

When A is logic 1 the output is logic 1, when A is logic 0 the output is logic 0, thus the output is always equal to the condition of A.

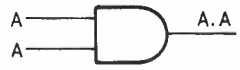



FIG. 6

Fig. 7 shows a signal $\bar{A}$, and its complement A applied to an AND gate. Since it is impossible for both A and $\bar{A}$ to be at logic 1 at the same time, the output of the gate must always be at logic 0. Therefore, the following identity applies.

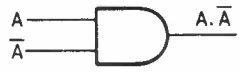



FIG. 7

$$A.\bar{A} = 0 \quad .. \quad .. \quad .. \quad .. \quad (4)$$

4.3 OR Function Identities. Consider an OR gate with one input permanently tied to logic 1, as shown in Fig. 8. An OR gate requires logic 1 at one input only to obtain a logic 1 output, therefore, the following identity applies:-


FIG. 8

$$A + 1 = 1 \quad .. \quad .. \quad .. \quad .. \quad (5)$$

The condition of the OR gate output in Fig. 9 depends on the logic condition of input A, because the other input is permanently tied to logic 0. If A is at logic 1 the output is at logic 1, and if A is at logic 0 the output is at logic 0, therefore the following identity applies.


FIG. 9

$$A + 0 = A \quad .. \quad .. \quad .. \quad .. \quad (6)$$

The OR gate in Fig. 10 has the same signal applied to both inputs. Therefore, when A is logic 1 the output is logic 1, and when A is logic 0 the output is logic 0. Thus the output is equal to the logic condition of A, and the following identity applies:-


FIG. 10

$$A + A = A \quad .. \quad .. \quad .. \quad .. \quad (7)$$

Fig. 11 shows an OR gate with signal A applied to one input, and its complement $\bar{A}$ applied to the other input. Since one of these signals must always be at logic 1, the output of the gate must always be at logic 1. Therefore, the following identity applies:-


FIG. 11

$$A + \bar{A} = 1 \quad .. \quad .. \quad .. \quad .. \quad (8)$$

4.4 Application of AND and OR Identities. The following is a summary of identities applying to the AND and OR functions:-

| AND | OR |
|---|---|
| $A.1 = A$ .. .. .. .. (1) | $A + 1 = 1$ .. .. .. .. (5) |
| $A.0 = 0$ .. .. .. .. (2) | $A + 0 = A$ .. .. .. .. (6) |
| $A.A = A$ .. .. .. .. (3) | $A + A = A$ .. .. .. .. (7) |
| $A.\bar{A} = 0$ .. .. .. .. (4) | $A + \bar{A} = 1$ .. .. .. .. (8) |

Some examples of the application of these identities are as follows:-

Example 1. Simplify $A.1 + A.A + B$.

Substitute A for A.1 (identity 1).

$= A + A.A + B$.

Substitute A for A.A (identity 3)

$= A + A + B$.

Substitute A for A + A (identity 7).

$= A + B$ (answer).

Example 2. Simplify $(B + 0).(B + 1).C$

Substitute B for B + 0 (identity 6) and 1 for B + 1 (identity 5).

$= B.1.C$

Substitute B for B.1 (identity 1).

$= B.C$ (answer).

5. LAWS OF BOOLEAN ALGEBRA

5.1 Section 4 dealt with the relationship between a variable and a logical 1 or 0 state. The laws in this section show the relationships of two or more variables and allow simplifications and manipulation of logic circuits into a variety of forms.

It is worthwhile to note that the laws of commutation, association and distribution in ordinary algebra are also true in Boolean algebra. The negation law is only true for two valued logic and the redundance law comes from the simple nature of logical combination.

5.2 Commutative Law. This law states that the inputs to a logic gate may be listed in any order without affecting the logical operation. For example, consider the three input AND gates in Figs. 12a and 12b. Since both these gates have the same signals applied, they perform the same logical operation. Therefore it is true to say that:-

$A.B.C. = C.B.A.$ .. .. .. .. (9)


(a) (b)

FIG. 12. COMMUTATIVE LAW APPLIED TO AND FUNCTION

The commutative law also applies to ORed variables, as shown in Fig. 13. The logic function remains the same regardless of the order in which the input variables are listed. This can be expressed in the following identity.

$$A + B + C = C + B + A \quad .. \quad .. \quad .. \quad .. \quad (10)$$

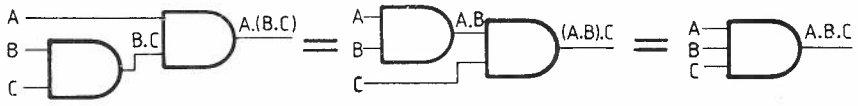

(a) (b)

FIG. 13. COMMUTATIVE LAW APPLIED TO OR FUNCTION

5.3 Associative Law. This law states that ANDed variables or ORed variables can be grouped together in any order in a Boolean expression. This can be expressed in logical form for three ANDed variables as follows:-

$$A.(B.C) = (A.B).C = A.B.C. \quad .. \quad .. \quad .. \quad .. \quad (11)$$

Each of the expressions has a different logic circuit, as shown in Fig. 14. However the basic logic function of Figs. 14a and 14b is exactly the same as Fig. 14c.



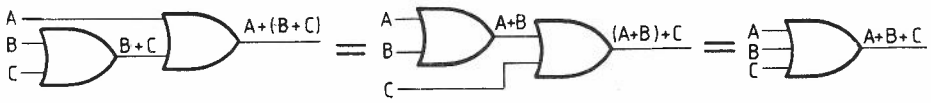

(a) (b) (c)

FIG. 14. ASSOCIATIVE LAW APPLIED TO AND FUNCTION

The associative law can be applied to ORed variables as follows:-

$$A + (B + C) = (A + B) + C = A + B + C \quad .. \quad .. \quad .. \quad .. \quad (12)$$

The logic diagrams representing each of the expressions in law 12, are shown in Fig. 15. Each of these circuits has exactly the same basic function which is A + B + C, as shown in Fig. 15c.



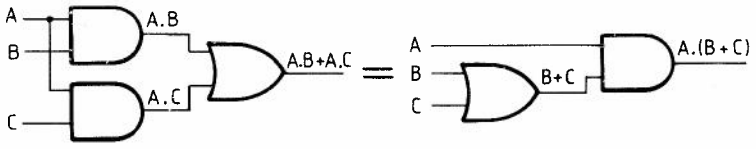

(a) (b) (c)

FIG. 15. ASSOCIATIVE LAW APPLIED TO OR FUNCTION

5.4 Distributive Law. The distributive law is expressed as follows:-

$$A.B + A.C = A.(B + C) \quad .. \quad .. \quad .. \quad .. \quad (13)$$

Figs. 16a and 16b show the logic circuits used to represent each expression in the identity. Although each circuit is different, they both have the same function. The distributive law shows that normal algebraic factorising techniques can be applied to a Boolean expression.




(a) (b)

FIG. 16. DISTRIBUTIVE LAW

5.5 The following are further examples of simplification using some of the previously stated identities and laws.

Example 1. Simplify $A.(A + B)$.

Substitute $A.A + A.B$ for $A.(A + B)$(law 13).

$= A.A + A.B$

Substitute A for A.A (identity 3).

$= A + A.B$

Factorise the expression (identity 13).

$= A.(1 + B)$.

Substitute 1 for (1 + B)(identity 5).

$= A.1$

Substitute A for A.1 (identity 1).

$= A$ (answer).

Example 2. Simplify $\bar{A}.B + \bar{A}.\bar{B}$

Factorise the expression (law 13).

$= \bar{A}.(B + \bar{B})$.

Substitute 1 for $B + \bar{B}$ (identity 8).

$= \bar{A}.1$

Substitute $\bar{A}$ for $\bar{A}.1$ (identity 1).

$= \bar{A}$ (answer).

5.6 Negation Law applies only to bi-valent logic and states that when a variable is negated twice the effect is that the variable is unchanged.

$\bar{\bar{A}} = A$ .. .. .. .. (14)

Two inverters in cascade achieve the function as shown in Fig. 17.

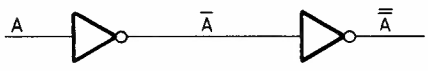



FIG. 17. NEGATION LAW

5.7 Redundance Law or absorptive law is derived from the proof of law 13 where the terms A.B and A.C do not appear in the final form.

$$A + A.B = A \quad .. \quad .. \quad .. \quad .. \quad (15)$$

Example 1 in Para 5.5 is another form of law 15. In terms of combinations of gates, Fig. 18 is a representation of the left hand side of law 15 but on application of the redundance law, variable B disappears and there is no function applied to A.

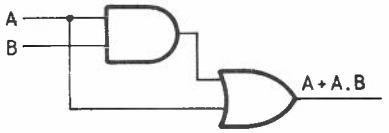



FIG. 18. REDUNDANCE LAW

5.8 De Morgan's Laws. Generally, the basic function of a logic circuit can not be readily interpreted from a negated expression situated at an intermediate point, or at an output. For example, the basic function of the negated expression $\overline{(A + B).C.D}$ is not immediately obvious. De Morgan's theorem, which is usually expressed in terms of two laws, provides a method by which a complex negated expression can be replaced by an alternative simplified expression. De Morgan's theorem implies that when an expression is negated, it can be replaced by another expression in which each variable is negated, each AND is changed to OR, and each OR is changed to AND. The two laws derived from this theorem are as follows:-

$$\overline{A.B} = \bar{A} + \bar{B} \quad .. \quad .. \quad .. \quad .. \quad (16)$$

$$\overline{A + B} = \bar{A}.\bar{B} \quad .. \quad .. \quad .. \quad .. \quad (17)$$

De Morgan's Laws are frequently used identities in Boolean manipulation.

The validity of the first of De Morgan's laws ($\overline{A.B} = \bar{A} + \bar{B}$) can be shown with the aid of a two input NAND gate (Fig. 19) and truth tables. The normal expression on the output of this gate is $\overline{A.B}$ , but this expression does not readily reveal the input conditions required to make the output significant. However, an equivalent expression for $\overline{A.B}$ can be obtained by applying De Morgan's theorem as follows; A is changed to $\bar{A}$, B is changed to $\bar{B}$, and the AND is changed to an OR. The new expression on the output of the NAND gate becomes $\bar{A} + \bar{B}$, which indicates that the output is significant when $\bar{A}$ is logic 1 OR $\bar{B}$ is logic 1, that is, when A is logic 0 OR B is logic 0. The equivalence of the two expressions $\overline{A.B}$ and $\bar{A} + \bar{B}$ is proven in truth tables 3 and 4.

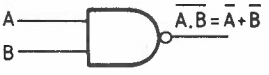



FIG. 19.

| A | B | A.B | $\overline{A.B}$ |
|---|---|---|---|
| 0 | 0 | 0 | 1 |
| 0 | 1 | 0 | 1 |
| 1 | 0 | 0 | 1 |
| 1 | 1 | 1 | 0 |

TABLE 3. TRUTH TABLE FOR $\overline{A.B}$.

| A | B | $\bar{A}$ | $\bar{B}$ | $\bar{A} + \bar{B}$ |
|---|---|---|---|---|
| 0 | 0 | 1 | 1 | 1 |
| 0 | 1 | 1 | 0 | 1 |
| 1 | 0 | 0 | 1 | 1 |
| 1 | 1 | 0 | 0 | 0 |

TABLE 4. TRUTH TABLE FOR $\bar{A} + \bar{B}$.

The validity of the second of De Morgan's Laws ($\overline{A + B} = \bar{A}.\bar{B}$) can be shown with the aid of a NOR gate (Fig. 20) and truth tables. The input conditions required to make the output significant are not obvious from the output expression $\overline{A + B}$. Another expression can be obtained for $\overline{A + B}$ by applying De Morgan's theorem as follows; A is inverted to $\bar{A}$, B is inverted to $\bar{B}$, and the OR is changed to an AND. The new expression on the output of the gate becomes $\bar{A}.\bar{B}$, which indicates that the output is significant when $\bar{A}$ is logic 1 and $\bar{B}$ is logic 1, that is, when A is logic 0 and B is logic 0. The equivalence of $\overline{A + B}$ and $\bar{A}.\bar{B}$ is shown in truth tables 5 and 6.


FIG. 20.

| A | B | A + B | $\overline{A + B}$ |
|---|---|---|---|
| 0 | 0 | 0 | 1 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 1 | 0 |

TABLE 5. TRUTH TABLE FOR $\overline{A + B}$

| A | B | $\bar{A}$ | $\bar{B}$ | $\bar{A}.\bar{B}$ |
|---|---|---|---|---|
| 0 | 0 | 1 | 1 | 1 |
| 0 | 1 | 1 | 0 | 0 |
| 1 | 0 | 0 | 1 | 0 |
| 1 | 1 | 0 | 0 | 0 |

TABLE 6. TRUTH TABLE FOR $\bar{A}.\bar{B}$

It should be realised that De Morgan's Laws (identities 16 and 17) can be written as follows,

$\overline{A.B.C.D.\ etc} = \bar{A} + \bar{B} + \bar{C} + \bar{D}\ \overline{etc}$ (identity 16)

$\overline{A + B + C + D\ etc} = \bar{A}.\bar{B}.\bar{C}.\bar{D}.\ \overline{etc}$ (identity 17)

5.9 Some Examples of the applications of De Morgan's Laws are as follows.

Example 1. Simplify $\overline{D + E + F}$

Applying law 17 ($\overline{A + B} = \bar{A}.\bar{B}$)

$= \bar{D}.\bar{E}.\bar{F}.$ (answer).

When applying De Morgan's Laws to complex expressions, it is essential that parts of the expressions which are bracketed, or would normally be considered as being bracketed, are considered as complete units until all 'De Morganising' has been completed, then the units can be simplified. This is demonstrated in example 2.

Example 2. Simplify $\overline{A.B + C.D}$

Since an AND function takes precedence over an OR function in a Boolean expression, A.B and C.D are considered as two separate units for the application of De Morgan's Laws, as follows:-

Let x = A.B and,

y = C.D.

Substitute x and y into the expression

$= \overline{x + y}$

Apply identity 15 ($\overline{A + B} = \bar{A}.\bar{B}$) by substituting x for A, and y for B.

$= \bar{x}.\bar{y}$

Re-substitute A.B for x and C.D for y

$= \overline{A.B}.\overline{C.D}$ Substitute $\bar{A} + \bar{B}$ for $\overline{A.B}$; and $\bar{C} + \bar{D}$ (law 17)

$= (\bar{A} + \bar{B}).(\bar{C} + \bar{D})$ (Answer).

Generally, the procedure shown in example 2 can be simplified by mentally recognising the units in an expression, and then applying De Morgan's Laws direct to the expression, as shown in examples 3 and 4.

Example 3. Simplify $\overline{(C + D).(E + F)}$.

As (C + D) and (E + F) are bracketed, they are considered as complete units.

Apply law 16 ($\overline{A.B} = \bar{A} + \bar{B}$) by substituting (C + D) for A, and (E + F) for B.

$= \overline{C + D} + \overline{E + F}$. Substitute $\bar{C}.\bar{D}$ for $\overline{C + D}$; and $\bar{E}.\bar{F}$

for $\overline{E + F}$ (law 17)

$= \bar{C}.\bar{D} + \bar{E}.\bar{F}$ (Answer).

Example 4. Simplify $\overline{(A + B + C).D}$

(A + B + C) and (D) are considered as separate units.

Apply law 16 ($\overline{A.B} = \bar{A} + \bar{B}$) by substituting

(A + B + C) for A and (D) for B, $= \overline{A + B + C} + \bar{D}$

Substitue $\bar{A}.\bar{B}.\bar{C}$ for $\overline{A + B + C}$ (law 16)

$= \bar{A}.\bar{B}.\bar{C} + \bar{D}$ (Answer)

Sometimes further simplification can be applied after De Morgan's Laws have been used on an expression, as shown in example 5.

Example 5.

Simplify $\overline{A.B}.\overline{A.C}$

In this case $\overline{A.B}$ and $\overline{A.C}$ are considered as complete units.

Apply law 16 ($\overline{A.B} = \bar{A} + \bar{B}$) by substituting $\overline{A.B}$ for A,

and $\overline{A.C}$ for B

$= \overline{\overline{A.B}} + \overline{\overline{A.C}}$

$= A.B + A.C$ (law 14)

Simplify this by applying law 13

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

## 6. NAND AND NOR GATE IMPLEMENTATION

6.1 It may be noticed that sometimes the logic circuit contains a predominance of one type of logic element. This is usually the result of the circuit designer's strive for economy in the implementation of a logic circuit. Broadly, the steps taken in logic design are:

:: Clearly state the problem and define input and output requirements

:: Prepare a truth table from the problem description

:: Obtain a logic expression from the truth table

:: Simplify the logic expression

:: Manipulate the logic expression to a desired form

:: Draw the logic circuit diagram

Integrated circuit manufacturers put a number of gates of the one type on to the one chip and this makes it convenient to use all NAND or all NOR type logic elements in the implementation of a design. Therefore once the simplest logic expression has been obtained it is usual to change it so that for example, all NAND gates are used.

6.2 NAND Gate Implementation of a Comparator. From Electronic Logic Principles 1 the logic diagram for a comparator is shown in Fig 21.

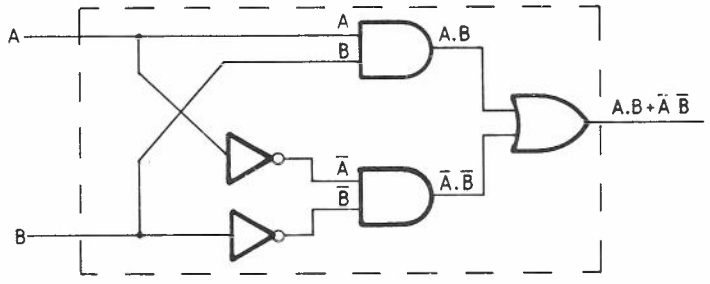



FIG. 21. COMPARATOR

The Boolean expression for the above circuit is: $E = A.B + \bar{A}.\bar{B}$

Applying the Negation Law in reverse: $E = \overline{\overline{A.B + \bar{A}.\bar{B}}}$

From De Morgan's Laws: $E = \overline{\overline{A.B} \; . \; \overline{\bar{A}.\bar{B}}}$

which can be drawn as shown in Fig 22.

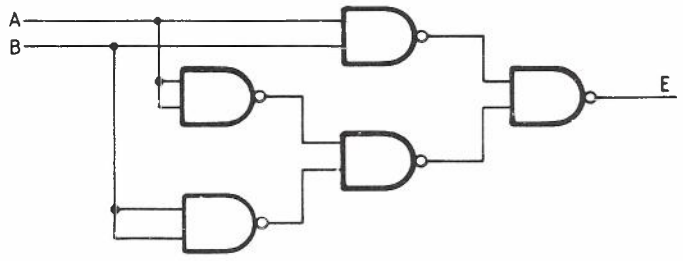



FIG. 22. COMPARATOR USING NAND GATES

6.3 NOR Gate Implementation of a Comparator. On the other hand a series of NOR gates may be used to perform the function of Fig 21.

$$E = A.B + \bar{A}.\bar{B}$$

$$= \overline{\overline{A.B + \bar{A}.\bar{B}}} \qquad \text{(law 14)}$$

$$= \overline{\overline{A.B} \;.\; \overline{\bar{A}.\bar{B}}} \qquad \text{(law 17)}$$

$$= \overline{(\bar{A} + \bar{B}).(\bar{\bar{A}} + \bar{\bar{B}})} \qquad \text{(law 16)}$$

$$= \overline{(\bar{A} + \bar{B}).(A + B)} \qquad \text{(law 14)}$$

$$= \overline{\bar{A} + \bar{B}} + \overline{A + B} \qquad \text{(law 16)}$$

$$= \overline{\overline{\bar{A} + \bar{B}} + \overline{A + B}} \qquad \text{(law 14)}$$

Fig. 23 is a comparator which uses NOR gates only.

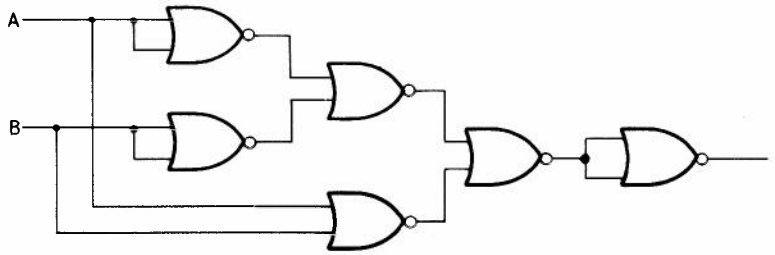



FIG. 23. COMPARATOR USING NOR GATES

## 7. TEST QUESTIONS

7.1 Answers to test question are in Section 8.

Q.1 Simplify the following expressions.

(a) $A.B.\bar{A}. + C.B.C.$

(b) $A + B + \bar{A} + B$

(c) $B.(C + 1).C.B$

Q.2 Simplify the following expressions.

(a) $A.B + B.C + \bar{C}.B$

(b) $A.(B.C + B.\bar{C})$

(c) $A.\bar{A}. + A.B + \bar{A}.B + B.B + C.A.$

Q.3 Simplify the following expressions.

(a) $\overline{\bar{A} + \bar{B}}$

(b) $\overline{\bar{A}.\bar{B}.\bar{C}.\bar{D}.}$

(c) $\overline{(\bar{A} + \bar{B}).(\bar{A} + \bar{C}).(\bar{A} + \bar{D})}$

Q.4 Manipulate the following expressions into a form suitable for NAND logic operation and draw the logic diagram of the NAND implementation.

(a) $A.B + \bar{A}.C + A.\bar{C}$

(b) $(A.C + \bar{B}).(\bar{B} + C)$

8. ANSWERS TO TEST QUESTIONS

ANS.1 (a) $A.B.\bar{A} + C.B.C$ .. .. (Substituting $A.\bar{A} = 0$; $C.C = C$)

$= B.0 + B.C$ .. .. ( " $B.0 = 0$)

$= 0 + B.C.$ .. .. ( " $0 + B.C = B.C$)

$= B.C$

(b) $A + B + \bar{A} + B$ .. .. ( " $A + \bar{A} = 1$; $B + B = B$)

$= 1 + B$ .. .. ( " $B + 1 = 1$)

$= 1$

(c) $B.(C + 1).C.B$ .. .. ( " $C + 1 = 1$)

$= B.1.C.B$ .. .. ( " $B.B = B$; $C.1 = C$)

$= B.C$

ANS.2 (a) $A.B + B.C + \bar{C}.B$

$= B.(A + C + \bar{C})$ .. ( " $C + \bar{C} = 1$)

$= B.(A + 1)$ .. .. ( " $A + 1 = 1$)

$= B.1$ .. .. ( " $B.1 = B$)

$= B$

(b) $A.(B.C + B.\bar{C})$

$= A.B.(C + \bar{C})$ .. .. ( " $C + \bar{C} = 1$)

$= A.B.1$ .. .. ( " $B.1 = B$)

$= A.B$

(c) $A.\bar{A} + A.B + \bar{A}.B + B.B + C.A$ (Substituting $A.\bar{A} = 0$; $B.B = B$)

$= 0 + A.B + \bar{A}.B + B + C.A$

$= B.(A + \bar{A} + 1) + C.A$ .. .. ( " $A + \bar{A} = 1$)

$= B.(1 + 1) + C.A$ .. .. ( " $1 + 1 = 1$)

$= B + C.A$

ANS.3 (a) $\overline{\bar{A} + \bar{B}}$

Apply law 17.

$= \bar{\bar{A}}.\bar{\bar{B}}$

$= A.B$ (Answer)

ANS. 3 (Continued)

(b) $\overline{\bar{A}.\bar{B}.\bar{C}.\bar{D}.}$

Apply law 16.

$= \bar{\bar{A}} + \bar{\bar{B}} + \bar{\bar{C}} + \bar{\bar{D}}$

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

(c) $\overline{(\bar{A} + \bar{B}).(\bar{A} + \bar{C}).(\bar{A} + \bar{D})}$

Apply law 16 ($\overline{A.B.C.} = \bar{A} + \bar{B} + \bar{C}$) by substituting $(\bar{A} + \bar{B})$ for A, $(\bar{A} + \bar{C})$ for B and $(\bar{A} + \bar{D})$ for C.

$= \overline{\bar{A} + \bar{B} + \bar{A} + \bar{C} + \bar{A} + \bar{D}}$

Apply law 17 to each unit.

$= \bar{\bar{A}}.\bar{\bar{B}} + \bar{\bar{A}}.\bar{\bar{C}} + \bar{\bar{A}}.\bar{\bar{D}}$

$= A.B + A.C + A.D$

Apply law 13.

$= A.(B + C + D)$ (Answer).

ANS.4 (a) $A.B + \bar{A}.C + A.\bar{C}$

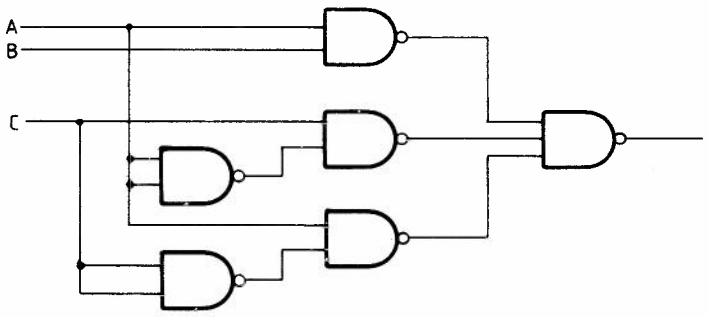
Apply law 14.

$= \overline{\overline{A.B + \bar{A}.C + A.\bar{C}}}$

Apply law 17.

$= \overline{\overline{A.B}.\ \overline{\bar{A}.C}\ .\ \overline{A.\bar{C}}}$ (Answer)

Logic Diagram:



(b) $(A.C + \bar{B}).(\bar{B} + C)$

Apply law 14.

$$= \overline{\overline{(A.C + \bar{B}).(\bar{B} + C)}}$$

Apply law 16.

$$= \overline{\overline{A.C + \bar{B} + \bar{B} + C}}$$

Apply law 17.

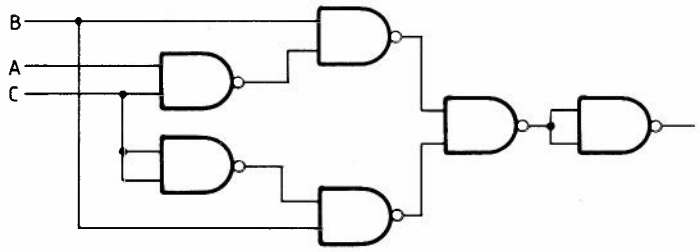
$$= \overline{\overline{A.C} \;.\; \bar{\bar{B}} \;.\; \overline{\bar{B}} \;.\; \bar{C}}$$

$$= \overline{\overline{A.C} \;.\; B \;.\; B \;.\; \bar{C}}$$

Apply law 14.

$$= \overline{\overline{\overline{A.C} \;.\; B \;.\; B \;.\; \bar{C}}}$$

Logic diagram:




END OF PAPER