CHAPTER 1

FUNDAMENTAL CONCEPTS

1.1 INTRODUCTION

All of us are familiar with the impact of modern digital computers, communication systems, digital display systems, internet, email etc. on society. One of the main causes of this revolution is the advent of *integrated circuits* (*ICs*), which became possible because of the tremendous progress in semiconductor technology in recent years. Most of us may not be familiar with the principles of working of computers, communication systems, internet, email, etc. even though these have become an important part of our daily life. The operation of these systems, and many other systems, is based on the principles of digital techniques and these systems are referred to as *digital systems*.

Some of us are familiar with electronic amplifiers. These are used to amplify electrical signals. This type of signals are continuous signals and can have any value in a limited range and are known as *analog signals*. The electronic circuits used to process (amplify) these signals are known as *analog circuits* and the systems built around this kind of operation are known as *analog systems*.

On the other hand, in an electronic calculator, the input is given with the help of switches. This is converted into electrical signals which have two discrete values or levels. One of these may be called as LOW level and the other one as HIGH level. The signal will always be of one of the two levels. Here, the actual value of the signal is immaterial as long as it is within the specified range of LOW or HIGH level. This type of signal is known as a *digital signal* and the circuits inside the calculator used to process these signals are known as *digital circuits*. A calculator is an example of a *digital system*.

There has been an unprecedented growth in digital techniques since Claude Shannon systematised and adapted George Boole's theoretical work in 1938. Developments in semiconductor technology, together with the progress in digital techniques, brought about a revolution in digital electronics when a single chip device known as *microprocessor* was introduced by Intel Corporation of America in 1971.

Since the introduction of microprocessors, the digital systems have gained tremendous power and importance. There is no field of knowledge which has affected our lives as much as the digital theory and applications, in such a short span of time. It has, infact, created a culture which nobody could have imagined till a few years ago. The rate of growth in this field has been unprecedented and the digital technology has become the most powerful technology for all future innovations in every walk of human endeavour whether it is computers, communication systems, information systems, entertainment and consumer products, business, banking and finance, office machines, homes, cars, education, industrial control systems, scientific and medical instruments, and defence equipment, etc.

Some of the principal reasons for the widespread use of digital techniques and systems are:

- The devices used in digital circuits generally operate in one of the two *states*, known as ON and OFF resulting in a very simple operation.
- There are only a few basic operations in digital circuits which are very easy to understand.
- Digital techniques require Boolean algebra which is very simple and can easily be learnt even in schools.
- Digital circuits require basic concepts of electric network analysis which can easily be learnt at the junior level in colleges. The principal electrical characteristics required are *switching speed* and *loading* considerations. On the other hand, analog circuits and systems involve frequency and time domain concepts, complicated circuit analysis techniques, etc. which make the understanding of these circuits much more difficult than the digital circuits.
- A large number of ICs are available for performing various operations. These are highly reliable, accurate, small in size and the speed of operation is very high. A number of programmable ICs are also available.
- Various ICs are available in a *logic family* with similar electrical characteristics which make the design and development of digital systems very simple and also reduces interfacing problems. Also, a number of logic families based on different technologies are available which help in optimising the system design from the point of view of power requirement and speed of operation.
- The effect of fluctuations in the characteristics of the components, ageing of components, temperature, and noise, etc. is very small in digital circuits.
- Digital circuits have capability of memory which makes these circuits highly suitable for computers, calculators, watches, telephones, etc.
- The display of data and other information is very convenient, accurate and elegant using digital techniques.
- Many students have an opportunity to learn programming of digital computers, hence they have a strong motivation to study the way the digital hardware works.
- It is a very fascinating and challenging field of study because most of the latest electronic systems are digital in nature.

1.2 DIGITAL SIGNALS

As mentioned above, a digital signal has two discrete levels or values. Two different representations of digital signals are shown in Fig. 1.1. In each case there are two discrete levels. These levels can be represented using the terms LOW and HIGH. In Fig. 1.1*a*, lower of the two levels has been designated as LOW level and the higher as HIGH level. In contrast to this, in Fig. 1.1*b*, higher of the two levels has been designated as LOW level and the lower as HIGH level. Digital systems using the representation of signal shown in Fig. 1.1*a* are said to employ *positive logic system* and those using the other representation of the signal shown in Fig. 1.1*b* are said to employ *negative logic system*. The genesis of the term *logic* is given later. In each of the two signals we observe that the voltage corresponding to a given level is not fixed, rather voltages in a limited range are designated as a level. As long as the voltage belongs to a level it will be taken as that level and the exact value of the voltage is immaterial. For example, any voltage in the range of 3.5 to 5 V will be considered as HIGH level in the positive logic system and LOW level in the negative logic system. Similarly, any voltage in the range of 0 to 1 V will be considered as LOW level in the positive logic system and HIGH level in the negative logic system. The actual voltage ranges corresponding to LOW and HIGH level are not same for all types of circuits and are different for different logic families (see Chapter 4). Unless otherwise specified, we shall be dealing with positive logic system.

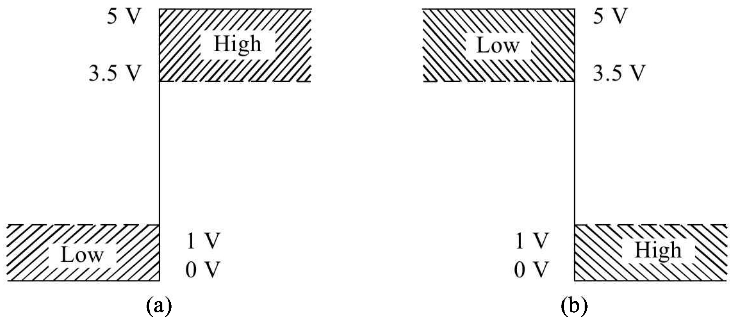


Fig. 1.1 ***Digital Signal Representation (a) Positive Logic (b) Negative Logic***

The above discussion brings out one of the main advantages of digital systems, viz. they are less susceptible to noise, fluctuations in the characteristics of components, etc.

The two discrete signal levels HIGH and LOW can also be represented by the *binary digits* 1 and 0 respectively. A *bi*nary digi*t* (0 or 1) is referred to as a *bit.* Since a digital signal can have only one of the two possible levels 1 or 0, the *binary number system* (see Chapter 2) can be used for the analysis and design of digital systems. The two levels (or states) can also be designated as ON and OFF or TRUE and FALSE. George Boole introduced the concept of binary number system in the studies of the mathematical theory of LOGIC in the work entitled *An Investigation of the Laws of Thought* in 1854 and developed its algebra known as *Boolean algebra.* These logic concepts have been adapted for the design of digital hardware since 1938 when Claude Shannon-organised and systematised Boole's work in *Symbolic Analysis of Relay and Switching Circuits.*

1.3 BASIC DIGITAL CIRCUITS

In a digital system there are only a few basic operations performed, irrespective of the complexities of the system. These operations may be required to be performed a number of times in a large digital system like digital computer or a digital control system, etc. The basic operations are AND, OR, NOT, and FLIP-FLOP. The AND, OR, and NOT operations are discussed here and the FLIP-FLOP, which is a basic memory element used to store binary information (one bit is stored in one FLIP-FLOP), will be introduced in Chapter 7.

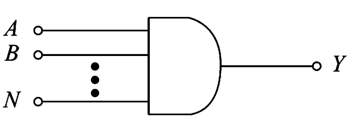


Fig. 1.2 ***The Standard Symbol for an AND Gate***

1.3.1 The AND Operation

A circuit which performs an AND operation is shown in Fig. 1.2. It has N inputs ($N \geq 2$) and one output. Digital signals are applied at the input terminals marked $A, B, \ldots, N$, the other terminal being ground, which is not shown in the diagram. The output is obtained at the output terminal marked Y (the other terminal being ground) and it is also a digital signal. The AND operation is defined as: the output of an AND gate is 1 if and only if all the inputs are 1. Mathematically, it is written as

$$\begin{aligned} Y &= A \text{ AND } B \text{ AND } C \ldots \text{ AND } N \\ &= A \cdot B \cdot C \cdot \ldots \cdot N \\ &= ABC \ldots N \end{aligned} \tag{1.1}$$

where A, B, C,... N are the input variables and Y is the output variable. The variables are binary, i.e. each variable can assume only one of the two possible values, 0 or 1. The *binary variables* are also referred to as *logical variables*.

Equation (1.1) is known as the *Boolean equation* or the *logical equation* of the AND *gate*. The term gate is used because of the similarity between the operation of a digital circuit and a gate. For example, for an AND operation the gate opens ($Y = 1$) only when all the inputs are present, i.e. at logic 1 level.

Truth Table Since a logical variable can assume only two possible values (0 and 1), therefore, any logical operation can also be defined in the form of a table containing all possible input combinations (2^N combinations for N inputs) and their corresponding outputs. This is known as a *truth table* and it contains one row for each one of the input combinations.

For an AND gate with two inputs A, B and the output Y, the truth table is given in Table 1.1. Its logical equation is $Y = AB$ and is read as "Y equals A AND B".

Table 1.1 ***Truth Table of a 2-Input AND Gate***

Inputs		Output
A	*B*	*Y*
0	0	0
0	1	0
1	0	0
1	1	1

Since, there are only two inputs, A and B, therefore, the possible number of input combinations is four. It may be observed from the truth table that the input–output relationship for a digital circuit is completely specified by this table in contrast to the input–output relationship for an analog circuit. The pattern in which the inputs are entered in the truth table may also be observed carefully, which is in the ascending order of binary numbers formed by the input variables. (See Chapter 2).

Logical Multiplication The AND operation is also referred to as logical multiplication and therefore, it is symbolised algebraically by a multiplication dot (·) as illustrated in Eq. (1.1).

Example 1.1

You have rented a locker in a bank. Express the process of opening the locker in terms of a digital operation.

Solution

The locker door (Y) can be opened by using one key (A) which is with you and the other key (B) which is with the bank executive. When both the keys are used, the locker door opens, i.e., the locker door can be opened ($Y = 1$) only when both the keys are applied ($A = B = 1$). Thus, this process can be expressed as an AND operation

$$Y = A \cdot B$$

Example 1.2

The voltage waveforms shown in Fig.1.3 are applied at the inputs of a 2-input AND gate. Determine the output waveform.

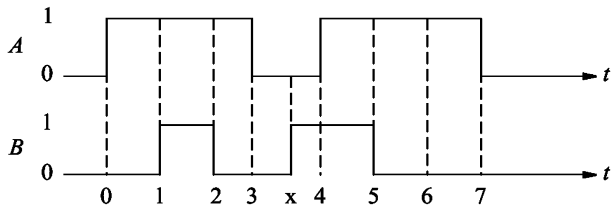


Fig. 1.3

Solution

Using Table 1.1, we find

From $t = 0$ *to* $t = 1$
$A = 1, B = 0$
Therefore, $Y = 0$

From $t = 1$ *to* $t = 2$
$A = B = 1$
Therefore, $Y = 1$

From $t = 2$ *to* $t = 3$
$A = 1, B = 0$
Therefore, $Y = 0$

From $t = 3$ *to* $t = \mathrm{x}$
$A = 0, B = 0$
Therefore, $Y = 0$

From $t = \mathrm{x}$ *to* $t = 4$
$A = 0, B = 1$
Therefore, $Y = 0$

From $t = 4$ *to* $t = 5$
$A = 1, B = 1$
Therefore, $Y = 1$

From $t = 5$ *to* $t = 7$
$A = 1, B = 0$
Therefore, $Y = 0$. It will be 0 for $t > 7$

The output waveform with reference to the input waveforms are shown in Fig. 1.4.

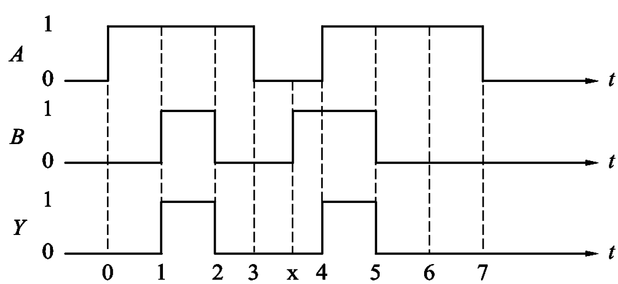


Fig. 1.4

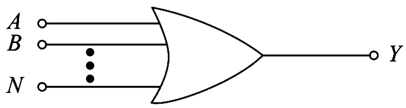


Fig. 1.5 ***The Standard Symbol for an OR Gate***

1.3.2 The OR Operation

Figure 1.5 shows an OR gate with N inputs ($N \geq 2$) and one output. The OR operation is defined as: the output of an OR gate is 1 if and only if one or more inputs are 1. Its logical equation is given by

$$Y = A \text{ OR } B \text{ OR } C \ldots \text{ OR } N = A + B + C + \ldots + N \tag{1.2}$$

The truth table of a 2-input OR gate is given in Table 1.2. Its logic equation is $Y = A + B$ and is read as "Y equals A OR B".

Table 1.2 ***Truth Table of a 2-Input OR Gate***

Inputs		Output
A	***B***	***Y***
0	0	0
0	1	1
1	0	1
1	1	1

Example 1.3

In a chemical process an ALARM is required to be activated if either temperature or pressure or both exceed certain limits. Is it possible to express this operation in terms of a digital operation? If yes, find the operation.

Solution

Let the temperature and pressure be converted into electrical signals and $T = 1$ if temperature exceeds the specified limit and $P = 1$ if pressure exceeds the specified limit. If $T = 1$ or $P = 1$ or both T and P are 1 then the ALARM is required to be activated, i.e., the signal applied to the ALARM $Y = 1$. This operation can be expressed as

$$Y = T \text{ OR } P$$
$$= T + P$$

Which is an OR operation.

Example 1.4

If the waveforms of Fig. 1.3 are applied at the inputs of a 2-input OR gate, determine the output waveform.

Solution

Using Table 1.2, we find

From $t = 0$ to $t = 1$
$A = 1, B = 0$
Therefore, $Y = 1$

From $t = 1$ to $t = 2$
$A = 1, B = 1$
Therefore, $Y = 1$

From $t = 2$ to $t = 3$
$A = 1, B = 0$
Therefore, $Y = 1$

From $t = 3$ to $t = \text{x}$
$A = 0, B = 0$
Therefore, $Y = 0$

From $t = x$ *to* $t = 4$
$A = 0, B = 1$
Therefore, $Y = 1$

From $t = 4$ *to* $t = 5$
$A = 1, B = 1$
Therefore, $Y = 1$

From $t = 5$ *to* $t = 7$
$A = 1, B = 0$
Therefore, $Y = 1$. It is 0 for $t > 7$.

The output waveform with reference to the input waveforms are shown in Fig. 1.6

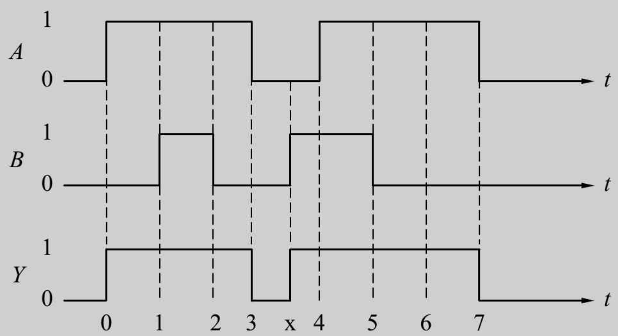


Fig. 1.6

1.3.3 The NOT Operation

Figure 1.7 shows a NOT gate, which is also known as an *inverter*. It has one input (A) and one output (Y). Its logic equation is written as

$$\begin{aligned} Y &= \text{NOT } A \\ &= \overline{A} \end{aligned} \tag{1.3}$$

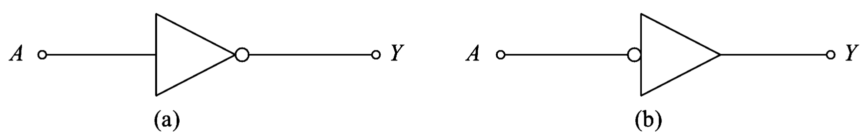


Fig. 1.7 ***The Standard Symbols for a NOT Gate***

and is read as "Y equals NOT A" or "Y equals complement of A". The truth table of a NOT gate is given in Table 1.3.

The NOT operation is also referred to as an *inversion* or *complementation*. The presence of a small circle, known as the *bubble*, always denotes inversion in digital circuits.

Table 1.3 ***Truth Table of a NOT Gate***

Input A	Output Y
0	1
1	0

Example 1.5

If the waveform shown in Fig. 1.8 is applied at the input of an inverter, find its output waveform.

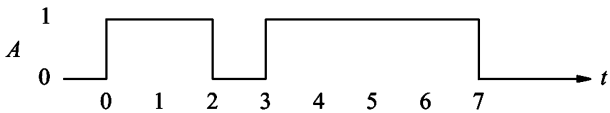


Fig. 1.8

Solution

Using Table 1.3, we find

From $t = 0$ *to* $t = 2$
$A = 1$
Therefore, Y = NOT A = NOT 1 = 0

From $t = 2$ *to* $t = 3$
$A = 0$
Therefore, Y = NOT A = NOT 0 = 1

From $t = 3$ *to* $t = 7$
$A = 1$
Therefore, Y = NOT A = NOT 1 = 0
Y will be 1 for $t > 7$

The output waveform with reference to the input waveform is shown in Fig. 1.9

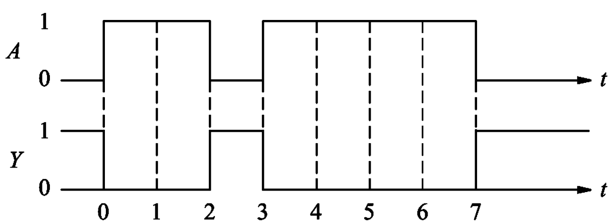


Fig. 1.9

1.4 NAND AND NOR OPERATIONS

Any Boolean (or logic) expression can be realised by using the AND, OR and NOT gates discussed above. From these three operations, two more operations have been derived: the NAND operation and NOR operation.

These operations have become very popular and are widely used, the reason being the only one type of gates, either NAND or NOR are sufficient for the realisation of any logical expression. Because of this reason, NAND and NOR gates are known as *universal gates*.

1.4.1 The NAND Operation

The NOT–AND operation is known as the NAND operation. Figure 1.10*a* shows an N input ($N \geq 2$) AND gate followed by a NOT gate. The operation of this circuit can be described in the following way:

The output of the AND gate (Y') can be written using Eq. (1.1)

$$Y' = AB \ldots N \tag{1.4}$$

Now, the output of the NOT gate (Y) can be written using Eq. (1.3)

$$Y = \overline{Y'} = \overline{AB \ldots N} \tag{1.5}$$

The logical operation represented by Eq. (1.5) is known as the NAND operation. The standard symbol of the NAND gate is shown in Fig. 1.10*b*. Here, a bubble on the output side of the NAND gate represents NOT operation, inversion or complementation.

The truth table of a 2-input NAND gate is given in Table 1.4. Its logic equation is $Y = \overline{A \cdot B}$ and, is read as "Y equals NOT (A AND B)".

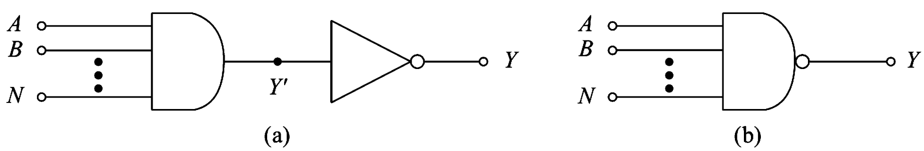


Fig. 1.10 ***(a)* NAND *Operation as* NOT–AND *Operation,* *(b)* *Standard Symbol for the* NAND *Gate***

Table 1.4 ***Truth Table of a 2-Input* NAND *Gate***

Inputs		Output
A	***B***	***Y***
0	0	1
0	1	1
1	0	1
1	1	0

The three basic logic operations, AND, OR and NOT can be performed by using only NAND gates. These are given in Fig. 1.11.

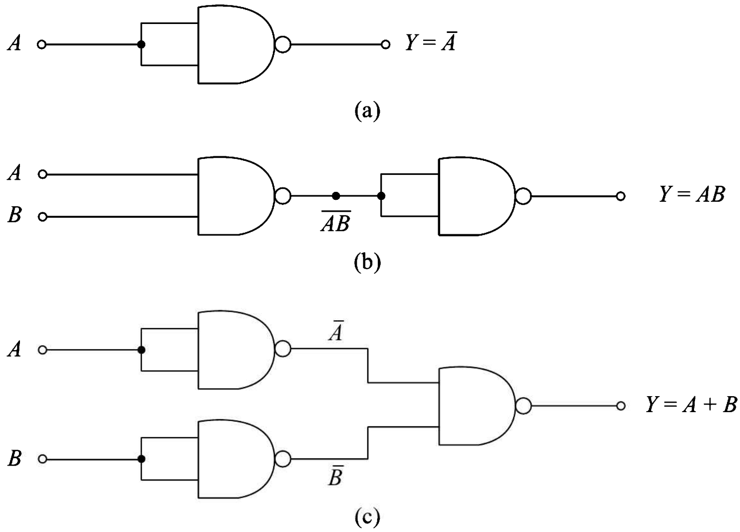


Fig. 1.11 ***Realisation of Basic Logic Operations Using NAND Gates (a) NOT (b) AND (c) OR***

Example 1.6

If the voltage waveforms of Fig. 1.3 are applied at the inputs of a 2-input NAND gate, find the output waveform.

Solution

The output waveform will be inverse of the output waveform Y of Fig. 1.4.

Example 1.7

What will be the output of a 2-input NAND gate, if one of its inputs is permanently connected to

(a) logic 0 voltage
(b) logic 1 voltage

Solution

(a) Figure 1.12*a* shows a 2-input NAND gate with one of its inputs connected to logic 0 voltage.
In this circuit if $A = 0$, $Y = 1$. Similarly, if $A = 1$, then also the output $Y = 1$.
This shows that the output Y is 1, irrespective of the other input. In fact, a NAND gate is *disabled* or *inhibited* if one of its inputs is connected to logic 0.

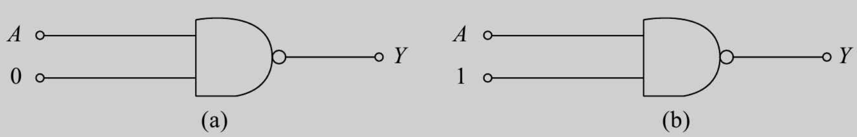


Fig. 1.12 ***Circuits for Example 1.7***

(b) Figure 1.12*b* shows a 2-input NAND gate with one of its inputs connected to logic 1 voltage
Here, if $A = 1$, $Y = 0$,
and if $A = 0$, $Y = 1$
This means $Y = \overline{A}$

This condition *enables* a NAND gate.

1.4.2 The NOR Operation

The NOT−OR operation is known as the NOR operation. Figure 1.13*a* shows an N input ($N \geq 2$) OR gate followed by a NOT gate. The operation of this circuit can be described in the following way:

The output of the OR gate Y' can be written using Eq. (1.2) as

$$Y' = A + B + \ldots + N \tag{1.6}$$

and the output of the NOT gate (Y) can be written using Eq. (1.3)

$$Y = \overline{Y'} = \overline{A + B + \ldots + N} \tag{1.7}$$

The logic operation represented by Eq. (1.7) is known as the NOR operation.

The standard symbol of the NOR gate is shown in Fig. 1.13*b*. Similar to the NAND gate, a bubble on the output side of the NOR gate represents the NOT operation.

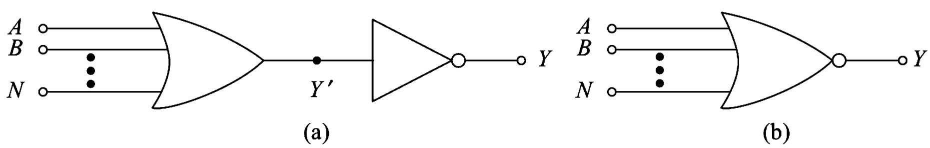


Fig. 1.13 ***(a) NOR Operation as NOT−OR Operation***
(b) Standard Symbol for the NOR Gate

Table 1.5 gives the truth table of a 2-input NOR gate. Its logic equation is $Y = \overline{A + B}$ and is read as "Y equals NOT (A OR B)".

Table 1.5 ***Truth Table of a 2-Input NOR Gate***

Inputs		Output
A	**B**	**Y**
0	0	1
0	1	0
1	0	0
1	1	0

The three basic logic operations, AND, OR, and NOT can be performed by using only the NOR gates. These are given in Fig. 1.14.

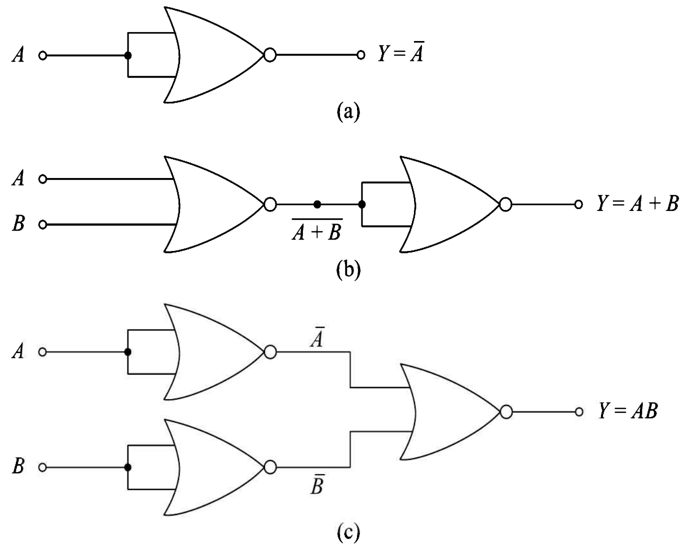


Fig. 1.14 ***Realisation of Basic Logic Operations Using* NOR *Gates (a)* NOT *(b)* OR *(c)* AND**

Example 1.8

If the voltage waveforms of Fig. 1.3 are applied at the inputs of a 2-input NOR gate, determine the output waveform.

Solution

The output waveform will be inverse of the output waveform of Fig. 1.6.

Example 1.9

(a) If one of the inputs of a NOR gate is connected to logic 0 voltage, find the output voltage in terms of the other input.
(b) Repeat (a) if one of the inputs is connected to logic 1 voltage.

Solution

(a) From Table 1.5, we observe that if $A = 0$ then $Y = \overline{B}$. This operation is used for enabling a NOR gate.
(b) Similarly from Table 1.5, we observe that if $A = 1$, then $Y = 0$. This operation inhibits or disables a NOR gate. Here, the output is 0 irrespective of the B input.

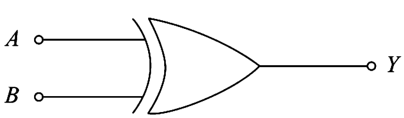


Fig. 1.15 ***Standard Symbol for* EX-OR *Gate***

1.5 EXCLUSIVE-OR AND EXCLUSIVE-NOR OPERATIONS

1.5.1 The EXCLUSIVE-OR Operation

The EXCLUSIVE-OR (EX-OR) operation is widely used in digital circuits. It is not a basic operation and can be performed using the basic gates—AND, OR and NOT or universal gates NAND or NOR. Because of its importance, the standard symbol shown in Fig. 1.15 is used for this operation and it is treated as basic logic element.

The truth table of an EX-OR gate is given in Table 1.6 and its logic equation is written as

$$Y = A \text{ EX-OR } B = A \oplus B \tag{1.8}$$

Table 1.6 ***Truth Table of* EX-OR *Gate***

Inputs		Output
A	***B***	***Y***
0	0	0
0	1	1
1	0	1
1	1	0

If we compare the truth table of an EX-OR gate with that of an OR gate given in Table 1.2, we find that the first three rows are same in both. Only the fourth row is different. This circuit finds application where two digital signals are to be compared. From the truth table we observe that when both the inputs are same (0 or 1) the output is 0, whereas when the inputs are not same (one of them is 0 and the other one is 1) the output is 1.

Example 1.10

The voltage waveforms, shown in Fig.1.3, are applied at the inputs of an EX-OR gate. Determine the output waveform.

Solution

Using Table 1.6, we find

From $t = 0$ *to* $t = 1$
$A = 1, B = 0$
Therefore, $Y = 1$

From $t = 1$ *to* $t = 2$
$A = 1, B = 1$
Therefore, $Y = 0$

From $t = 2$ *to* $t = 3$
$A = 1, B = 0$
Therefore, $Y = 1$

From $t = 3$ *to* $t = \text{x}$
$A = 0, B = 0$
Therefore, $Y = 0$

From $t = \text{x}$ *to* $t = 4$
$A = 0, B = 1$
Therefore, $Y = 1$

From $t = 4$ *to* $t = 5$
$A = 1, B = 1$
Therefore, $Y = 0$

From $t = 5$ *to* $t = 7$
$A = 1, B = 0$
Therefore, $Y = 1$

For $t > 7$
$A = 0, B = 0$
Therefore, $Y = 0$

The output waveform with reference to the input waveforms are shown in Fig. 1.16.

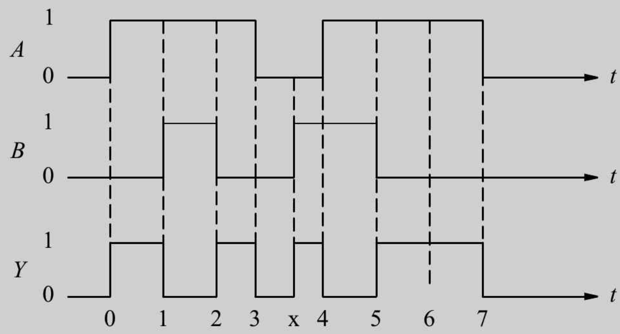


Fig. 1.16

Example 1.11

Determine the relation between the output Y and one of its inputs, if its other input is connected to
(a) logic 0
(b) logic 1

Solution

(a) Let us connect input A to 0, from the first two rows of the Table 1.6, we observe that $Y = B$.
(b) Similarly if input A is connected to logic 1 voltage, from the last two rows of the truth table, we observe that $Y = \overline{B}$

1.5.2 The EXCLUSIVE-NOR Operation

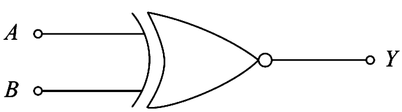


Fig. 1.17 ***Standard Symbol for EX-NOR Gate***

Figure 1.17 shows the standard symbol of EXCLUSIVE-NOR (EX-NOR) gate. Its truth table is given in Table 1.7.

Its logic operation is specified as

$$Y = A \text{ EX-NOR } B = \overline{A \text{ EX-OR } B} = \overline{A \oplus B} = A \odot B \tag{1.9}$$

Similar to EX-OR gate, EX-NOR gate is not a basic operation and can be performed using the basic gates or universal gates, but because of its importance, it has been given a standard symbol. The EX-NOR operation is also referred to as the *coincidence* operation because it produces output of 1 when its inputs coincide in value, both 0 or both 1.

Table 1.7 ***Truth Table of EX-NOR Gate***

Inputs		Output
A	**B**	**Y**
0	0	1
0	1	0
1	0	0
1	1	1

Example 1.12

Repeat Example1.10 for EX-NOR gate.

Solution

The output waveform will be inverse of the output waveform, as shown in Fig. 1.16.

Example 1.13

Repeat Example1.11 for EX-NOR gate

Solution

(a) $Y = \overline{B}$
(b) $Y = B$

1.6 BOOLEAN ALGEBRA

As discussed above, the digital signals are discrete in nature and can only assume one of the two values 0 or 1. A number system based on these two digits is known as *binary number system*. In the middle of 19th century, an English mathematician George Boole developed rules for manipulations of binary variables, known as *Boolean algebra*. This is the basis of all digital systems like computers, calculators, etc.

Binary variables can be represented by a letter symbol such as $A, B, X, Y, \ldots$ The variable can have only one of the two possible values at any time, viz. 0 or 1. The Boolean algebraic theorems are given in Table 1.8.

From these theorems, we observe that the even numbered theorems can be obtained from their preceding odd numbered theorems by (*i*) interchanging + and $\cdot$ signs, and (*ii*) interchanging 0 and 1.

Theorems which are related in this way are called *duals*.

Table 1.8 ***Boolean Algebraic Theorems***

Theorem No.	Theorem
1.1	$A + 0 = A$
1.2	$A \cdot 1 = A$
1.3	$A + 1 = 1$
1.4	$A \cdot 0 = 0$
1.5	$A + A = A$
1.6	$A \cdot A = A$
1.7	$A + \overline{A} = 1$
1.8	$A \cdot \overline{A} = 0$
1.9	$A \cdot (B + C) = AB + AC$
1.10	$A + BC = (A + B)(A + C)$
1.11	$A + AB = A$

(*Continued*)

Table 1.8 **(*Continued*)**

Theorem No.	Theorem
1.12	$A(A+B)=A$
1.13	$A+\overline{A}B=(A+B)$
1.14	$A(\overline{A}+B)=AB$
1.15	$AB+A\overline{B}=A$
1.16	$(A+B)\cdot(A+\overline{B})=A$
1.17	$AB+\overline{A}C=(A+C)(\overline{A}+B)$
1.18	$(A+B)(\overline{A}+C)=AC+\overline{A}B$
1.19	$AB+\overline{A}C+BC=AB+\overline{A}C$
1.20	$(A+B)(\overline{A}+C)(B+C)=(A+B)(\overline{A}+\mathrm{C})$
1.21	$\overline{A\cdot B\cdot C\cdot\ldots}=\overline{A}+\overline{B}+\overline{C}+\ldots$
1.22	$\overline{A+B+C+\ldots}=\overline{A}\cdot\overline{B}\cdot\overline{C}\ldots$

Theorems 1.1 to 1.8 involve a single variable only. Each of these theorems can be proved by considering every possible value of the variable. For example, in Theorem 1.1,

if $A=0$ then $0+0=0=A$

and if $A=1$ then $1+0=1=A$

and hence the theorem is proved.

Theorems 1.9 to 1.20 involve more than one variable and can be proved by making a truth table. For example, Theorem 1.10 can be proved by making the truth table given in Table 1.9.

Table 1.9 ***Truth Table to Prove Theorem 1.10***

A	***B***	***C***	***BC***	***A + BC***	***A + B***	***A + C***	***(A + B) • (A + C)***
0	0	0	0	0	0	0	0
0	0	1	0	0	0	1	0
0	1	0	0	0	1	0	0
0	1	1	1	1	1	1	1
1	0	0	0	1	1	1	1
1	0	1	0	1	1	1	1
1	1	0	0	1	1	1	1
1	1	1	1	1	1	1	1

From the table we observe that there are 8 (= 2^3) possible combinations of the three variables A, B, and C. For each combination, the value of $A + BC$ is the same as that of $(A + B)(A + C)$, which proves the theorem. Theorems 1.21 and 1.22 are known as De Morgan's theorems. These theorems can be proved by first considering the two variable case and then extending this result. From the truth table given in Table 1.10 we get the relations

$$\overline{A \cdot B} = \overline{A} + \overline{B} \tag{1.10}$$

and

$$\overline{A + B} = \overline{A} \cdot \overline{B} \tag{1.11}$$

Table 1.10 ***Truth Table to Prove De Morgan's Theorems***

A	B	$\overline{A}$	$\overline{B}$	$\overline{AB}$	$\overline{A} + \overline{B}$	$\overline{A + B}$	$\overline{A} \cdot \overline{B}$
0	0	1	1	1	1	1	1
0	1	1	0	1	1	0	0
1	0	0	1	1	1	0	0
1	1	0	0	0	0	0	0

Now consider the NAND operation of three variables,

$$\begin{aligned} \overline{ABC} &= \overline{(AB) \cdot C} \\ &= \overline{(A \cdot B)} + \overline{C} \\ &= \overline{A} + \overline{B} + \overline{C} \quad \text{using Eq. (1.10)} \end{aligned} \tag{1.12}$$

In a similar way, the NOR operation of three variables gives

$$\begin{aligned} \overline{A + B + C} &= \overline{(A + B) + C} \\ &= \overline{(A + B)} \cdot \overline{C} \\ &= \overline{A} \cdot \overline{B} \cdot \overline{C} \quad \text{using Eq. (1.11)} \end{aligned} \tag{1.13}$$

The above results can be easily extended to any number of variables.

A logic problem can be specified in terms of a set of statements. This set of statements can be represented in terms of an equation called the logic equation or in terms of a truth table. A digital circuit using the gates discussed above can be designed to realise a logic equation. In general, it is possible to simplify (minimise) a logic equation. The minimised logic equation will probably need less number of gates and/or less number of inputs for the gates. The techniques used to minimise logic equations will be discussed in Chapter 5. An example of the realisation of a circuit for a given logic equation is given below:

Example 1.14

Realise (design) a digital circuit for the logic equation

$$Y = \overline{A} \cdot B + A \cdot \overline{B} \tag{1.14}$$

Solution

This equation consists of two input variables A and B and one output variable Y. The variable A appears as $\overline{A}$ in the term $\overline{A} \cdot B$ and as A in the term $A \cdot \overline{B}$. Similarly the variable B appears as B in the term $\overline{A} \cdot B$ and as $\overline{B}$ in the term $A \cdot \overline{B}$. The two terms $\overline{A} \cdot B$ and $A \cdot \overline{B}$ are obtained by AND operations and the output Y is the result of OR operation performed on $\overline{A} \cdot B$ and $A \cdot \overline{B}$ terms. The realisation of this equation is obtained using NOT, AND and OR gates as shown in Fig. 1.18.

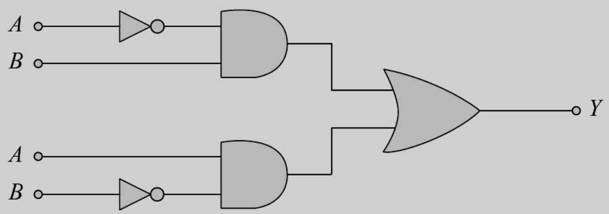


Fig. 1.18 ***Realisation of Eq. (1.14)***

1.7 EXAMPLES OF IC GATES

All the logic functions introduced in this chapter are commercially available in integrated circuit (IC) form. For example, 7400 IC chip is a quadruple 2-input NAND gate available in 14-pin DIP. It has four identical, independent 2-input NAND gates arranged as shown in Fig. 1.19. It requires a +5 V d.c. supply (to be connected between V_{CC} and GND pins) for the operation of the gates. Table 1.11 gives some of the available gate ICs. The details of their pin connexions, electrical characteristics, etc. can be obtained from the manufacturers' data catalogues.

Some of the manufacturers are:

- Texas Instruments (www.ti.com)
- Philips (www.philips.com)
- Fairchild semiconductor (www.fairchildsemi.com)

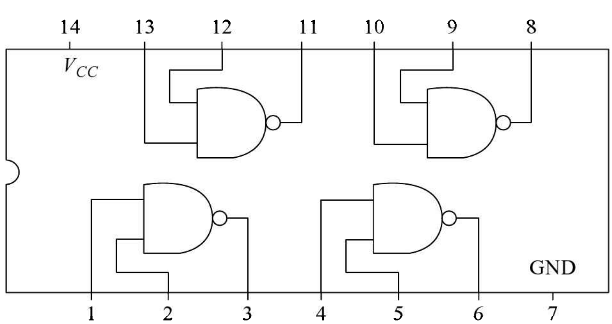


Fig. 1.19 ***Block Diagram of 7400 IC***

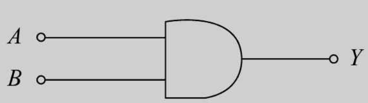
Table 1.11 ***Some of the Available IC Gates***

IC No.	Description
7400	Quad 2-input NAND gates
7402	Quad 2-input NOR gates
7404	Hex inverters
7408	Quad 2-input AND gates
7410	Triple 3-input NAND gates
7411	Triple 3-input AND gates
7420	Dual 4-input NAND gates
7421	Dual 4-input AND gates
7427	Triple 3-input NOR gates
7430	8-input NAND gate
7432	Quad 2-input OR gates
7486, 74386	Quad EX-OR gates
74133	13-input NAND gate
74135	Quad EX-OR/NOR gates
74260	Dual 5-input NOR gates

SUMMARY

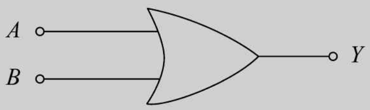
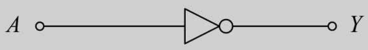
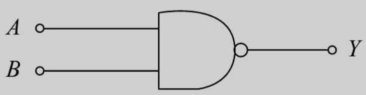
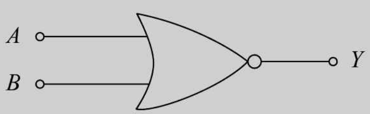
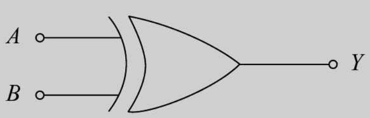
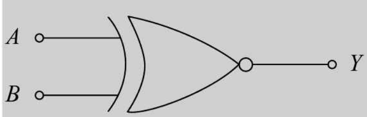
In this chapter, the basic concepts of the digital systems have been discussed. The basic features and advantages of these systems have been given briefly. The level of the treatment has been kept low to avoid any confusion. Table 1.12 summarises the operation of all the gates introduced in this chapter. For convenience, two input gates have been taken and the different symbols used for various operations are also given. A brief exposure to Boolean algebra has also been given. The techniques for the simplification of logic equations will be discussed in Chapter 5.

Table 1.12 ***Summary of Logic Gates***

Gate	Logic diagram	Function	Truth table		
			Inputs		Output
			A	B	Y
AND		$Y = A$ AND B $= A \cdot B$ $= A \cap B$ $= A \wedge B$ $= AB$	0	0	0
			0	1	0
			1	0	0
			1	1	1

(*Continued*)

Table 1.12 **(*Continued*)**

Gate	Logic diagram	Function	Truth table
OR		$Y = A$ OR B $= A + B$ $= A \cup B$ $= A \vee B$	Inputs: A, B; Output: Y 0 0 0 0 1 1 1 0 1 1 1 1
NOT (inverter)		$Y =$ NOT A $= \overline{A}$	Input: A; Output: Y 0 1 1 0
NAND		$Y = A$ NOT AND B $= A$ NAND B $= \overline{A \cdot B}$ $= \overline{A \cap B}$ $= \overline{A \wedge B}$ $= A \uparrow B$ $= \overline{AB}$	Inputs: A, B; Output: Y 0 0 1 0 1 1 1 0 1 1 1 0
NOR		$Y = A$ NOT OR B $= A$ NOR B $= \overline{A + B}$ $= \overline{A \cup B}$ $= \overline{A \vee B}$ $= A \downarrow B$	Inputs: A, B; Output: Y 0 0 1 0 1 0 1 0 0 1 1 0
EX-OR		$Y = A$ EX-OR B $= A \oplus B$ $= A\overline{B} + \overline{A}B$	Inputs: A, B; Output: Y 0 0 0 0 1 1 1 0 1 1 1 0
EX-NOR		$Y = \overline{A \text{ EX-OR } B}$ $= A$ EX-NOR B $= A \odot B$ $= \overline{A\overline{B} + \overline{A}B}$ $= \overline{A}\,\overline{B} + AB$	Inputs: A, B; Output: Y 0 0 1 0 1 0 1 0 0 1 1 1

GLOSSARY

Active-high input The input terminal is active (or enabled) when held at HIGH logic level.

Active-high output The output terminal is at HIGH logic level when active (or enabled).

Active-low input The input terminal is active (or enabled) when held at LOW logic level.

Active-low output The output terminal is at low logic level when active (or enabled).

Analog circuit An electronic circuit that processes analog signals.

Analog signal A continuous signal that can have any value in a given range. It is also known as continuous signal.

Analog system An electronic system that consists of analog circuits and/or devices.

AND gate A logic circuit whose output is 1 if and only if all its inputs are 1.

Binary Having two states.

Binary number system A number system with base (or radix) 2, i.e. only two symbols 0 and 1 are used to represent any number.

Binary variable A variable that can have only two values 0 and 1.

Bit A binary digit 0 and 1.

Boolean algebra The algebra of binary variables.

Boolean function A well defined relationship between binary variables specified by either a Boolean equation or a truth table.

Boolean variable Same as the binary variable.

Bubble A small circle indicating inversion operation or active-low condition.

Chip A piece of silicon (or any semiconductor material) on which an integrated circuit is fabricated.

Computer An electronic digital system that processes data. See also digital computer.

Digital circuit An electronic circuit that processes digital signals.

Digital computer A programmable digital system capable of performing various arithmetic and logical operations.

Digital electronics Branch of electronics which deals with the digital devices, circuits and systems.

Digital signal A signal with only two discrete values. These are usually represented by LOW and HIGH or 0 and 1.

Digital system An electronic system consisting of digital circuits and/or devices.

DIP (Dual-in-line package) A semiconductor chip package having two rows of pins perpendicular to the edges of the package.

Disable Prohibits an activity from proceeding or output(s) to be produced in a digital circuit. Also known as INHIBIT.

Dual Two boolean functions are said to be dual of each other if any one of these is obtained from the other one by (i) interchanging + and · signs, and (ii) interchanging 0 and 1.

Enable Allows activity to proceed or output(s) to be produced in a digital circuit.

EX-NOR (Exclusive-NOR) gate A two input gate that produces a high output only when both the inputs are same.

EX-OR (Exclusive-OR) gate A two input gate whose output is logic 0 when both the inputs are equal and logic 1 when they are unequal.

False One of the states of a logic circuit, the other state is true.

FILP-FLOP It is a basic memory element in digital systems. It is same as bistable multivibrator.

Gate A logic circuit in which the output depends upon the inputs according to some logic rules.

High-level The voltage corresponding to logic 1.

IC package An integrated circuit chip packaged as a single multilead component. For example DIP.

Inhibit Same as disable.

Integrated circuit (IC) A small semiconductor chip containing several electronic circuits.

Inversion The process of complementing a logic signal.

Inversion circle Same as bubble.

Inverter A logic gate whose output is the complement of its input.

Logic circuit An electronic circuit that operates on digital signals in accordance with a logic function.

Logic family A group of logic circuits built around a standardised integrated circuit technology. For example, transistor-transistor logic (TTL), complementary metal-oxide-semiconductor logic (CMOS) etc.

Logical variable Same as binary variable.

Logic gate Same as gate.

LOW-level The voltage corresponding to logic 0.

Memory A device that stores binary information.

Microprocessor A semiconductor IC chip with the capabilities of CPU of a computer.

NAND gate A logic gate whose output is logic 0 if and only if all of its inputs are logic 1.

Negative logic system Logic system in which the lower of the two levels is represented by 1 and the higher level is represented by 0.

Noise Unwanted electrical signals which are random in nature.

NOR gate A logic gate whose output is logic 1 if and only if all of its inputs are logic 0.

NOT gate Same as inverter.

OFF One of the states of a logic circuit, the other state is ON.

ON One of the states of a logic circuit, the other state is OFF.

OR gate A logic gate whose output is logic 0 if and only if all its inputs are logic 0.

Positive logic system Logic system in which the higher of the two levels is represented by 1 and the lower level is represented by 0.

Programmable ICs ICs which can be programmed for desired purpose by a user.

Switching speed Speed with which an electronic switch can change from ON to OFF and vice-versa. It is usually expressed in terms of the propagation delay time.

True One of the states of a logic circuit, the other state is *False*.

Truth table A table that gives outputs for all possible combinations of inputs to a logic circuit.

Universal gate A gate that can perform all the basic logical operations, such as NAND, and NOR.

REVIEW QUESTIONS

1.1 Ordinary electrical switch is ______________ device. (analog/digital)
1.2 A train of pulses is ______________ signal. (analog/digital)
1.3 The output of an AND gate is *high* if and only if all its inputs are ______________ . (high/low)
1.4 If one of the inputs to an OR gate is *high* its output will be ______________ . (high/low)
1.5 An AND gate output will always differ from an OR gate output for the same input conditions. (True/False)
1.6 An OR gate is DISABLED by connecting one of its inputs to logic level ______________ . (0/1)
1.7 To ENABLE an OR gate, one of its inputs is connected to logic level ______________ . (0/1)
1.8 To INHIBIT (or DISABLE) an AND gate one of its inputs is connected to logic level ______________. (0/1)
1.9 To ENABLE an AND gate one of its inputs is connected to logic level ______________ . (0/1)
1.10 An AND gate is ENABLED by connecting one of its inputs to logic level ______________ . (0/1)
1.11 To DISABLE a NOR gate one of its inputs needs to be connected to logic level ______________ . (0/1)
1.12 One of the inputs of an AND gate is labelled as ENABLE. This control input is ______________ . (active-low/active-high)
1.13 One of the inputs of a NOR gate is labelled as ENABLE. This control input is ______________ . (active-low/active-high)
1.14 The universal gates cannot be used as inverters. (True/False)
1.15 EXCLUSIVE-OR and EXCLUSIVE-NOR gates can be used as inverters. (True/False)
1.16 An EX-OR gate can be used to compare digital signals. (True/False)
1.17 If one of the inputs of an EX-OR gate is *high*, its output will be ______________ . (same as other input/inverse of other input)
1.18 The number of rows in a truth table of 4 variables is ______________ .
1.19 A 3-input NOR gate is required to detect the simultaneous occurrence of all the inputs in the LOW state. Its output is ______________ . (active-low/active-high)
1.20 The number of 3-input NAND gates in a 14-pin IC is ______________ .
1.21 The minimum member of bits required to distinguish 108 distinct objects is ______________ .

PROBLEMS

1.1 Which of the following systems are analog and which are digital? Why?
(a) Pressure gauge
(b) An electronic counter used to count persons entering an exhibition
(c) Clinical thermometre
(d) Electronic calculator
(e) Transistor radio receiver
(f) Ordinary electric switch.
(g) Electronic Voting Machine (EVM)

1.2 In the circuits of Fig. 1.20 the switches may be ON (1) or OFF (0) and will cause the bulb to be ON (1) or OFF (0).
(a) Determine all possible conditions of the switches for the bulb to be ON (1)/OFF (0) in each of the circuits.
(b) Represent the information obtained in part (*a*) in the form of truth table.
(c) Name the operation performed by each circuit (refer to Table 1.12).

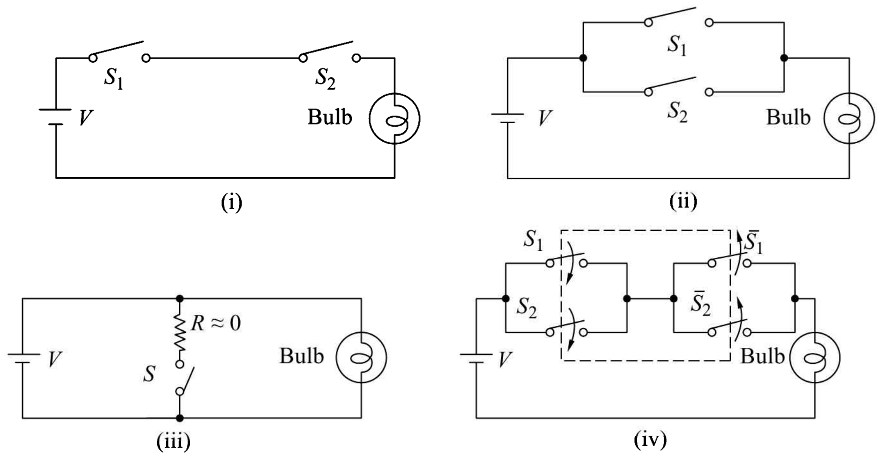


Fig. 1.20 ***Circuits for Problem 1.2***

1.3 The voltage waveforms shown in Fig. 1.21 are applied at the inputs of 2-input AND, OR, NAND, NOR, EX-NOR, and EX-OR gates. Determine the output waveform in each case.

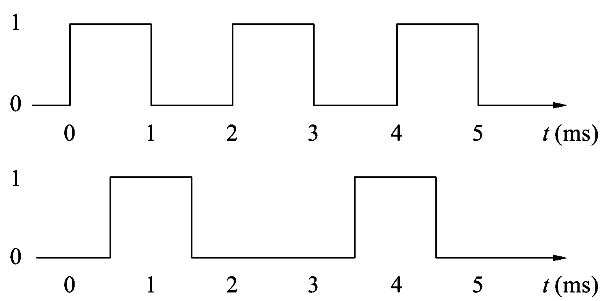


Fig. 1.21 ***Waveforms for Problem 1.3***

1.4 Find the relationship between the inputs and output for each of the gates shown in Fig. 1.22. Name the operation performed in each case (refer to Table 1.12).

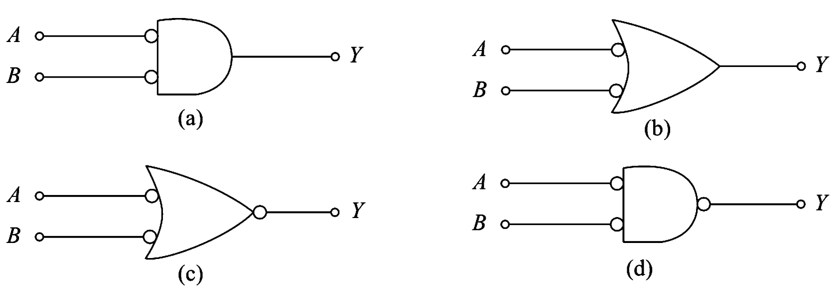


Fig. 1.22 ***Circuits for Problem 1.4***

1.5 Make the truth tables for each of the circuits of Figs. 1.11 and 1.14 and verify the operation performed.

1.6 For each of the following statements indicate the logic gate(s) AND, OR, NAND, NOR for which it is true.
(a) All LOW inputs produce a HIGH output.
(b) Output is HIGH if and only if all inputs are HIGH.
(c) Output is LOW if and only if all inputs are HIGH.
(d) Output is LOW if and only if all inputs are LOW.

1.7 For the logic expression,

$$Y = A\overline{B} + \overline{A}B$$

(a) Obtain the truth table.
(b) Name the operation performed.
(c) Realise this operation using AND, OR, NOT gates.
(d) Realise this operation using only NAND gates.

1.8 Prove the following:
(a) A positive logic AND operation is equivalent to a negative logic OR operation and vice-versa.
(b) A positive logic NAND operation is equivalent to a negative logic NOR operation and vice-versa.

1.9 Prove the following using the Boolean algebraic theorems:
(a) $A + \overline{A} \cdot B + A \cdot \overline{B} = A + B$
(b) $A \cdot B + \overline{A} \cdot B + \overline{A} \cdot \overline{B} = \overline{A} + B$
(c) $\overline{A}BC + A\overline{B}C + AB\overline{C} + ABC = AB + BC + CA$

1.10 Prove the logic equations of Problem 1.9 using the truth table approach.

1.11 Realise the left hand side and the right hand side of the logic equations of problem 1.9 using AND, OR, and NOT gates and find the saving in hardware in each case (number of gates and the number of inputs for the gates).

1.12 Prove the following using De Morgan's theorems:
(a) $AB + CD = \overline{\overline{AB} \cdot \overline{CD}}$
(b) $(A + B) \cdot (C + D) = \overline{\overline{(A + B)} + \overline{(C + D)}}$

And hence, prove the following statements:
(i) An AND-OR configuration is equivalent to a NAND-NAND configuration.
(ii) An OR-AND configuration is equivalent to a NOR-NOR configuration.

1.13 (a) Realise the logic equation (a) of Problem 1.12 using
(i) AND and OR gates.
(ii) only NAND gates.
(b) Realise the logic equation (b) of Problem 1.12 using
(i) OR and AND gates.
(ii) only NOR gates.

1.14 Verify that the following operations are commutative and associative
(a) AND (b) OR (c) EX-OR

1.15 Verify that the following operations are commutative but not associative.
(a) NAND (b) NOR

1.16 Realise the logic expression

$$Y = A \oplus B \oplus C \oplus D$$

using EX-OR gates.

1.17 Consider the expression: $Z = A \oplus B \oplus C \oplus D \oplus \ldots$ Show that $Z = 1$ if an odd number of variables are 1 and that $Z = 0$ if an even number of variables are 1.

1.18 For a gate with N inputs, how many combinations of inputs are possible? State the general rule to obtain the possible combinations.

1.19 Determine the number of pins in the following ICs.

(a) 7402 (d) 7410 (g) 7427
(b) 7404 (e) 7411 (h) 7432
(c) 7408 (f) 7420 (i) 7486

1.20 Determine the IC chips required for the implementation of each of the circuits of Problem 1.13.

1.21 The logic levels for two typical logic circuits A and B are given below:

A: 0.4 V and 2 V
B: –0.75 V and –1.55 V

Express these levels in binary form assuming positive logic system.

1.22 Make truth table for a 3-input

(a) AND gate (b) OR gate (c) NAND gate (d) NOR gate

1.23 Is it possible to use a 3-input gate as a 2-input gate for the following gates? If yes, how?

(a) AND (b) OR (c) NAND (d) NOR

1.24 Is it possible to INHIBIT (or DISABLE) AND, OR, NAND, NOR gates? If yes, how?

1.25 One of the inputs of a gate is used to control the operation of the gate and is labelled as ENABLE. Is it active-high or active-low if the gate is

(a) AND?
(b) OR?
(c) NAND?
(d) NOR?

1.26 Is the INHIBIT input active-high or active-low in Prob. 1.24.

1.27 Realise a 3-input gate using 2-input gates for the following gates:

(a) AND (b) OR (c) NAND (d) NOR

1.28 Prove the following:

(a) $A \oplus B = \overline{A} \oplus \overline{B}$
(b) $\overline{A \oplus B} = A \oplus \overline{B} = \overline{A} \oplus B$
(c) $B \oplus (B \oplus A \cdot C) = A \cdot C$

1.29 Is it possible to use the following gates as inverters? If yes, how?

(a) NAND
(b) NOR
(c) EX-OR
(d) EX-NOR
(e) AND
(f) OR

1.30 For the logic circuit shown in Fig. 1.23, find out the logic function performed using

(a) Boolean algebraic theorems
(b) truth table

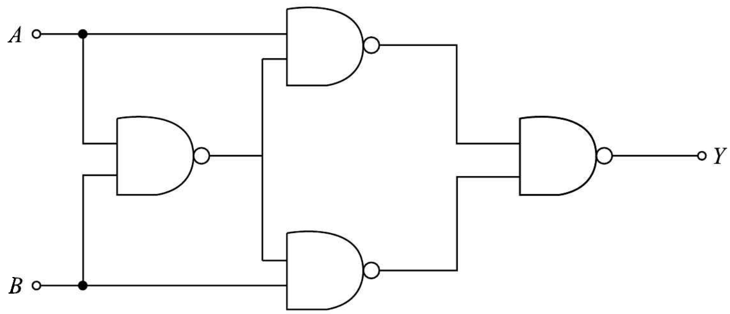


Fig. 1.23 ***Circuit for Problem 1.30***