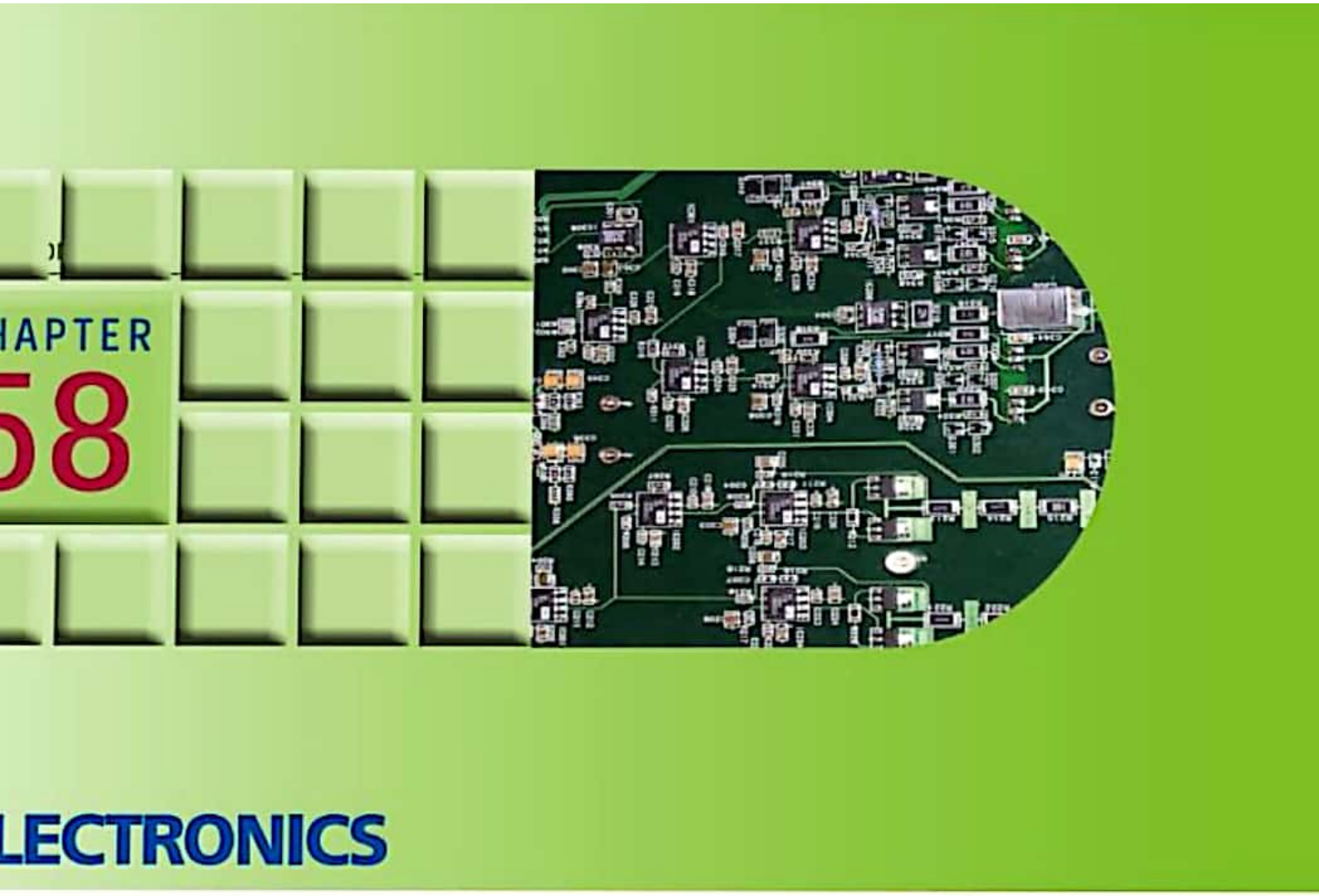


HAPTER
58

LECTRONICS

AT A GLANCE

	Formation of p-n Junction Diode	58.21	Condition for Self-Excitation (Barkhausen Criterion for Oscillations)
	Forward and Reverse Biasing of a Junction Diode	58.22	Hartley Oscillator
	V-I Characteristics of a Junction Diode	58.23	Colpitt's Oscillator
	The Zener Diode	58.24	Tunnel Diode Operation
	Experiment to Study the Characteristics of the Zener Diode	58.25	Tunnel Diode
	Zener Diode as Voltage Regulator	58.26	Field-Effect Transistors
	P-N Junction Diode as Rectifier	58.27	Drain Characteristics of an n Channel JFET
	Diode as a Half-Wave Rectifier	58.28	Principle of n-Channel FET Operation

AT A GLANCE			
70.1	Decimal Number System	70.2	Binary Number System
70.3	Conversion of Binary Number into Decimal Number		
70.4	Conversion of Decimal Number into Binary Number		
70.5	Binary Addition	70.6	Binary Subtraction
70.7	Complement Method of Subtraction	70.8	Binary Multiplication
70.9	Binary Division	70.10	Octal Number System
70.11	Hexadecimal Number System	70.12	Boolean Algebra
70.13	Boolean Addition and Multiplication	70.14	Logic Expressions

expressed in terms of the position or place values.

For example, the number 10523 is represented as

$10523 = 1 \times 10^4 + 0 \times 10^3 + 5 \times 10^2 + 2 \times 10^1 + 3 \times 10^0$.

The digit (3) has the position value 10^0 and is the *least significant digit* (*LSD*).

The digit (1) has the place value 10^4 and is the *most significant digit* (*MSD*).

Similarly, the number 2564.397 can be expressed as

$2564.397 = 2 \times 10^3 + 5 \times 10^2 + 6 \times 10^1 + 4 \times 10^0 + 3 \times 10^{-1} + 9 \times 10^{-2} + 7 \times 10^{-3}$.

That is the powers to the base 10 are numbered to the left of the decimal point starting with 0 and to the right of the decimal point starting with – 1.

70.2 Binary Number System

In a *binary system* of representations the *base* (or *radix*) is 2. It uses only two numerals 0 and 1. In a digital system there are only two possible states or conditions. For example, a situation may be True or False, a switch close or open, a voltage signal High or Low etc. These states or conditions are designated as 1 and 0 respectively. The binary digits 0 and 1 are termed as bits. Like the decimal system, the binary system also has a place or position value representation.

For example, the number 15 of decimal system is written in the binary system as 1111, since,

$$1111 = 1 \times 2^3 + 1 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 = 15.$$

This can also be written as $1111_2 = 15_{10}$, the subscript indicating the number system. In the binary 1111, the bit 1 at the extreme left is the MSB (most significant bit) and the bit 1 at the extreme right is the LSB.

Similarly, the binary 1011.011 is written in the decimal system as 11.375, since

$$1011.011 = 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 + 0 \times 2^{-1} + 1 \times 2^{-2} + 1 \times 2^{-3}$$

$$= 8+0+2+1+0+\frac{1}{4}+\frac{1}{8} = 11{\cdot}375$$

or $1011.011_2 = 11.375_{10}$

Thus, the place values of the bits in a binary number are given by ascending powers of 2 to the left of binary point starting from 0 and to the right of binary point in the descending powers of 2 starting from – 1.

70.3 Conversion of Binary Number into Decimal Number

(*i*) Conversion of integral binary numbers

Example 1. *Convert* $(1010)_2$ *into its decimal equivalent.*

$$1010 = 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 0 \times 2^0 = 8 + 0 + 2 + 0 = 10$$

$$\Rightarrow (1010)_2 = (10)_{10}$$

EXAMPLE 2. *Convert* $(1011.0101)_2$ *into its decimal equivalent.*

$$1011.0101 = 1\times 2^0 + 1\times 2^1 + 0\times 2^2 + 1\times 2^3 + 0\times 2^{-1} + 1\times 2^{-2} + 0\times 2^{-3} + 1\times 2^{-}$$

$$= 1+2+0+8+0+\frac{1}{4}+0+\frac{1}{16}$$

$$= 11.3125$$

$$(1011.0101)_2 = (11.3125)_{10}$$

70.4 Conversion of Decimal Number into Binary Number

(*i*) Conversions of integral decimal numbers

The given decimal number is divided progressively by 2, until we get zero. The remainders aken in the reverse order, give the binary number.

As an example let us convert the decimal 19 into its binary equivalent.

Successive divisions		Remainders	
2 | 19			
2 | 9	→	1	Top
2 | 4	→	1	
2 | 2	→	0	
2 | 1	→	0	
0	→	1	Bottom

Reading the remainders from the bottom to the top, the binary equivalent of 19 is found t be 10011, or

$$19_{10} = 10011_2.$$

EXAMPLE 3. *Convert* $(25)_{10}$ *into binary number.*

		Remainders	
2 | 25			
2 | 12	→	1	
2 | 6	→	0	
2 | 3	→	0	$(25)_{10} = (11001)_2$
2 | 1	→	1	
0	→	1	

(*ii*) Conversion of fractional-decimal numbers

The given decimal number is multiplied by 2 progressively. For each step that results in a 1 in the units place, transfer the 1 to the binary record and repeat the process with the fractional number. For each multiplication by 2 that results in a product less than unity, record a 0 in the binary number and carry on the process. The last step is reached if the fractional part is zero or it is terminated when the desired accuracy is attained. The carries are taken in the forward (top to bottom) direction to give the equivalent binary.

Let us convert 0.9125 into its binary equivalent.

$0.9125 \times 2 = 1.8250 = 0.8250$ with a carry 1
$0.8250 \times 2 = 1.6500 = 0.6500$ with a carry 1
$0.6500 \times 2 = 1.3000 = 0.3000$ with a carry 1
$0.3000 \times 2 = 0.6000 = 0.6000$ with a carry 0
$0.6000 \times 2 = 1.2000 = 0.2000$ with a carry 1
$0.2000 \times 2 = 0.4000 = 0.4000$ with a carry 0 ↓

The process is terminated here to get an *approximate* result, namely, representation of 0.9125 by six binary digits.

$\therefore \quad 0.9125_{10} = 0.111010_2$

The point in front of the binary is referred to as the *binary point.*

To find the binary equivalent of a decimal number like 35.625, we split the number into an integer of 35 and a fraction of 0.625. Then the binary equivalent of each part is obtained separately by methods discussed above.

First let us find out the binary equivalent of the integer part 35 by divide-by-two method.

$35 \div 2 = 17 + 1$ remainder 1 — Top
$17 \div 2 = 8 + 1$ remainder 1 ↑
$8 \div 2 = 4 + 0$ remainder 0
$4 \div 2 = 2 + 0$ remainder 0
$2 \div 2 = 1 + 0$ remainder 0
$1 \div 2 = 0 + 1$ remainder 1

$\therefore \quad 35_{10} = 100011_2$

The binary equivalent of 0.625 is found by the multiply-by-two method,

0.625×2	$=$ 1.25	$=$	$1 + 0.25$	carry 1	Top
0.25×2	$=$ 0.50	$=$	$0 + 0.50$	carry 0	↓
0.50×2	$=$ 1.0	$=$	$0 + 1$	carry 1	Bottom

Taking the carrys from top to bottom

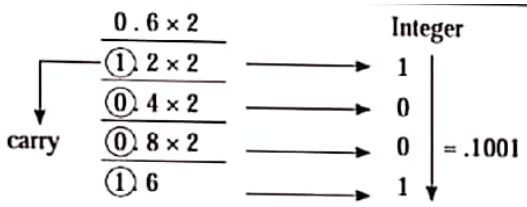
$0.625_{10} = 101$

$\therefore \quad 35.625_{10} = 100011.101_2$

Example 1. *Convert* $(21.6)_{10}$ *into binary number.*

Remainders

2	21		
2	10	→ 1	
2	5	→ 0	
2	2	→ 1	= 10101
2	1	→ 0	
	0	→ 1	



$\therefore \quad (21.6)_{10} = (10101.1001)_2$

70.5 Binary Addition

The sum of two binary numbers is calculated by the same rules as in decimal, except that the digits of the sum in any significant position can be only 0 or 1. Any "carry" obtained in a given significant position is used by the pair of digits one significant position higher. Binary addition is with the four rules :

1. $0 + 0 = 0$
2. $0 + 1 = 1$
3. $1 + 0 = 1$
4. $1 + 1 = 10$ (one-zero, not ten)

The last rule is often written as $1 + 1 = 0$ with a carry of 1.

Example of addition of two binary numbers is shown below :

augend :	101101
addend :	+ 100111
sum :	1010100

Consider **another example**

	1	0	1	1	(11)
+	1	0	0	1	(9)
1	0	1	0	0	(20)

Explanation. Addition is made column-wise.

70.6 Binary Subtraction

The *four rules* for binary subtraction are given below :

1. $0 - 0 = 0$
2. $1 - 0 = 1$
3. $1 - 1 = 0$
4. $10 - 1 = 1$.

The last rule indicates that when 1 is subtracted from 10_2 (= decimal 2), we get 1.

EXAMPLE 1.

minuend :	101101
subtrahend :	− 100111
difference :	000110

EXAMPLE 1. *Let us subtract* 1101_2 *from* 1010_2.

1's Complement Method

$$\begin{array}{r} 1010 \\ +\ 0010 \\ \hline \square 1100 \end{array}$$

↑

No carry

(1's complement of 1101)

Conventional Method

$$\begin{array}{r} 1010 \\ -\ 1101 \\ \hline -\ 0011 \end{array}$$

Since there is no end-around carry (EAC) in this case, we write 1's complement of 1100 and attach a minus sign to it. Thus

Final answer → − 0011.

EXAMPLE 2. *Consider the subtraction*

$$\begin{array}{rl} 1010 & \text{Minuend} \\ -\ 1001 & \text{Subtrahend} \end{array}$$

1's complement of the subtrahend is 0110

$$\therefore \quad \begin{array}{r} 1010 \\ -1001 \\ \hline \end{array} \quad \Rightarrow \begin{array}{r} 1010 \\ +\ 0110 \\ \hline 10000 \end{array} \quad \Rightarrow \begin{array}{r} 0000 \\ +\quad\ 1 \\ \hline 0001 \end{array}$$

↑

carry

$$\therefore \quad \begin{array}{r} 1010 \\ -\ 1001 \\ \hline 0001 \end{array}$$

(b) 2's complement. The 2's complement of a binary number is obtained by adding 1 to its 1's complement.

Subtraction by 2's complement

(*i*) Find the 2's complement of the subtrahend.

(*ii*) Add this complement to the minuend.

(*iii*) Drop the carry in the last position.

(*iv*) If the carry in the last position is 1, the result is positive.

(*v*) If there is no 1 carry in the last position, determine the 2's complement of the result of subtraction and attach a minus sign to it. Thus, the answer of subtraction is negative.

The following examples illustrate the procedure.

(*i*) Let us subtract 10001_2 from 10011_2.

2's Complement Method

$$\begin{array}{r} 10011 \\ +\ 01111 \\ \hline [1]\ 00010 \end{array}$$

(2's complement of 10001)

↓

Drop the carry.

Conventional Method

$$\begin{array}{r} 10011 \\ -\ 10001 \\ \hline 00010 \end{array}$$

Therefore, *final answer is* : 00010.

(*ii*) Let us subtract 1101_2 from 1010_2.

$$\begin{array}{rl} 1010 & \text{Minuend} \\ -\ 1101 & \text{Subtrahend} \end{array}$$

2's complement of subtraction = 1's complement + 1 = 0010 + 1 = 0011

$$\begin{array}{r} 1010 \\ +\ 0011 \\ \hline \square\ 1101 \end{array}$$

(2's complement of 1101)

↑

No carry

Thus, there is no 1 carry in the last position. Hence, we determine 2's complement of 1101 which is

$$0010 + 1 = 0011.$$

After attaching the minus sign, the final answer is – 0011.

70.8 Binary Multiplication

The following are the rules for binary multiplication.

(*i*) $0 \times 0 = 0$

(*ii*) $0 \times 1 = 0$

(*iii*) $1 \times 0 = 0$

(*iv*) $1 \times 1 = 1$

Binary multiplication is carried out as in decimal system.

Scanned with CamScanner

Example 1. *Multiply* 10110 *by* 110.

$$\begin{array}{r} 10110 \times 110 \\ 00000 \\ 10110 \\ 10110 \\ \hline 10000100 \end{array}$$

Verification

22×6

132

$\therefore$ The result is 10000100.

70.9 Binary Division

Binary division is done as in decimal system.

Example 2. *Divide* 1111 by 110.

$$\begin{array}{r|l} & 10 \\ \hline 110 & 1111 \\ & 110 \\ \hline & 11 \end{array}$$

$\therefore$ Quotient : 10

Remainder : 11

Verification

15 ÷ 6 gives

$$\begin{array}{r|l} & 2 \\ \hline 6 & 15 \\ & 12 \\ \hline & 3 \end{array}$$

Quotient : 2

Remainder : 3

70.10 Octal Number System

The radix or base of this system is 8. It uses the eight numerals 0, 1, 2, 3, 4, 5, 6, 7. For countin

the second and so on. For example, in the hexadecimal system 10 (second digit followed by the first) represents 16 in the decimal system. The position value for each digit is in ascending powers of 16 for integers and descending powers of 16 for fractions.

Examples of Hexadecimal to Decimal Conversion

(*i*) $3\,C\,8_{16} = 3 \times 16^2 + 12 \times 16^1 + 8 \times 16^0$
$= 768 + 192 + 8 = 968_{10}$

(*ii*) $E\,5\,F\,8_{16} = 14 \times 16^3 + 5 \times 16^2 + 15 \times 16^1 + 8 \times 16^0$
$= 57344 + 1280 + 240 + 8 = 58{,}872_{10}$

BOOLEAN ALGEBRA

70.12 Boolean Algebra

The algebra of logic prominently used in the operation of computer devices is the algebra developed by George Boole. It is a binary or two-valued logic, *i.e.*, it permits only two values or states for its variables. These two states are 'true' and 'false' in logic but are represented by 'on' and 'off' states of electronic circuits.

The two variables of the Boolean algebra are usually represented by 0 and 1. Hence, every variable is either a 0 or a 1. There are no negative or fractional numbers. Logically, we may write :

If $X = 0$ then $X \neq 1$

And if $X = 1$ then $X \neq 0$.

Boolean algebra uses only three operations on its variables. These operations are:

(*i*) The OR addition represented by a + (plus) sign.

(*ii*) The AND multiplication represented by a × (cross), or a . (dot) sign.

70.14 Logic Expressions

NOT

The operation of an invertor (NOT circuit) can be expressed with symbols as follows : If the input variable is called A and the output variable is called X, then $X = \overline{A}$. This expression states that the output is the complement of the input, so that if $A = 0$, then $X = 1$, and if $A = 1$, then $X = 0$ (Fig. 70.1).

Fig. 70.1

(*i*) $\overline{0} = 1$ (*ii*) $\overline{1} = 0$

AND

The operation of a two-input AND gate can be expressed in equation form as follows : If one input variable is A, the other input variable is B, and the output variable is X, then the Boolean expression for this basic gate function is $X = AB$ (Fig. 70.2).

Fig. 70.2

OR

The operation of a two-input OR gate can be expressed in equation form as follows : If one input is A, the other input is B, and the output is X, then the Boolean expression is $X = A + B$ (Fig. 70.3).

Fig. 70.3

70.15 Commutative Laws

(*i*) The commutative law of addition for two variables is written algebraically as

$$A + B = B + A$$

This states that the order in which the variables are ORed makes no difference.

This law states that ORing several variables and ANDing the result with a single variable is equivalen
o ANDing the single variable with each of the several variables and then ORing the products.

70.18 Rules for Boolean Algebra

Table 70.1 lists *Basic rules of Boolean algebra.*

Table 70.1

1.	$A + 0 = A$
2.	$A + 1 = 1$
3.	$A \,.\, 0 = 0$
4.	$A \,.\, 1 = A$
5.	$A + A = A$
6.	$A + \overline{A} = 1$
7.	$A \,.\, A = \mathrm{A}$
8.	$A \,.\, \overline{A} = 0$
9.	$\overline{\overline{A}} = A$
10.	$A + AB = A$
11.	$A + \overline{A}B = A + B$
12.	$(A + B)\ (A + C) = \mathrm{A} + BC$

Rule 1. A variable ORed with a 0 is equal to the value of the variable $(A + 0 = A)$ (Fig. 70.4).
Similarly, the other rules are proved.

70.19 De Morgan's Theorems

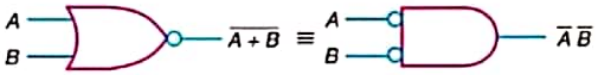


Fig. 70.6

Truth Table

A	B	$\overline{A+B}$	$\overline{A}\,\overline{B}$
0	0	1	1
0	1	0	0
1	0	0	0
1	1	0	0

LOGIC GATES

70.20 Digital Logic Gates

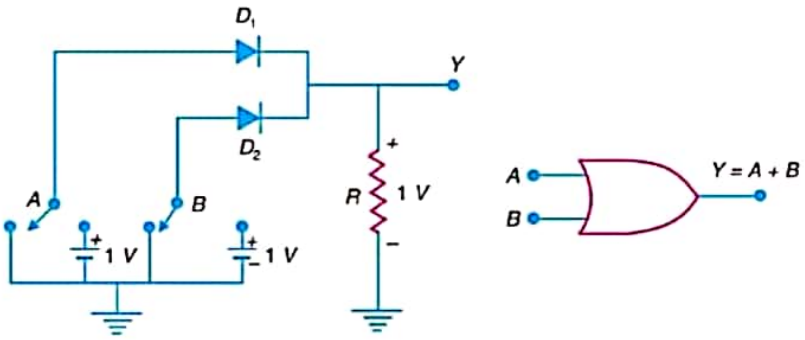
A logic gate is an electronic circuit which performs logic functions or takes a logic decision. It has one output and one or more inputs. Logic gates are the building blocks of the digital systems. They work on the logical algebra developed by George Boole. The Boolean operations namely 'OR' operation, 'AND' operation and 'NOT' operation are implemented by three logic gates called 'OR' gate, 'AND' gate and 'NOT' gate.

Binary Number and Logic Gates.

70.21 The OR Gate

It implements Boolean 'OR' operation. An 'OR' gate has two (or) more inputs and one output. *An 'OR' gate is a logic gate whose output is '1' state if any or all the inputs are in '1' state.*

Fig. 70.7 (*a*) shows a two input OR gate using two ideal diodes D_1 and D_2. Here A and B represent the two inputs and Y the output. R represents the output load resistor. Fig. 70.7 (*b*) gives the symbolic representation of the OR gate.



(*a*) (*b*)

conduct. Since D_1 is conducting, current flows through R. Hence, there is an output voltage, *i.e.*, $Y = 1$.

Case (*iii*). When $A = 0$; $B = 1$, the diode D_2 is forward biased and so it conducts. D_1 does not conduct. Since D_2 is conducting, current flows through R. Hence, there is an output voltage, *i.e.*, $Y = 1$.

Case (*iv*). When $A = 1$; $B = 1$, both the diodes D_1 and D_2 are forward biased. Hence, both are conducting. So current flows through R and there is an output voltage, *i.e.*, $Y = 1$.

The logic operation of the OR gate can be summarised in a tabular form known as Truth Table. A truth table may be defined as a table which gives the output state for all possible input combinations.

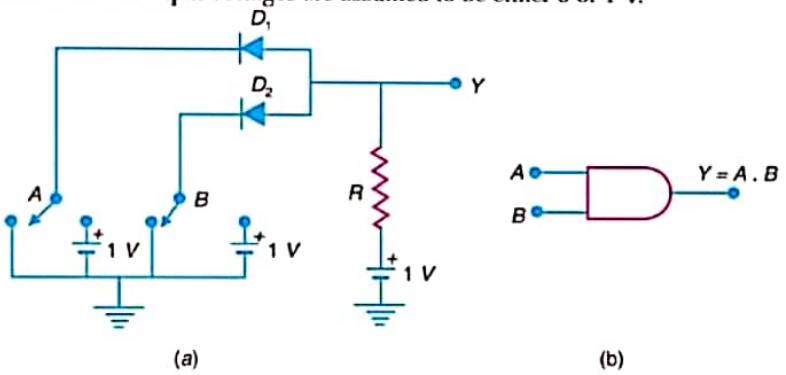
Truth Table

A	B	$Y = A + B$
0	0	0
1	0	1
0	1	1
1	1	1

70.22 The AND Gate

An "AND" gate implements Boolean 'AND' operation. It has two or more inputs and one output. *An 'AND' gate is a logic gate whose output is '1', if and only if all the inputs are in '1' state.*

Fig. 70.8 (*a*) *shows a two-input AND gate using two ideal diodes and Fig.* 70.8 (*b*) *gives its symbolic representation.* Here A and B represent the two inputs and Y the output. R represents the output load resistor. The two input voltages are assumed to be either 0 or 1 V.



(*a*) (b)

Fig. 70.8

70.23 The NOT Gate

The NOT circuit has only one input and one output. It inverts the polarity of a pulse applied to it. If the input is 1, the output is 0 and when the input is 0 the output is 1. That is, the NOT gate inverts the input. A 'NOT' gate using transistor is shown in Fig. 70.9 (*a*) and its symbol in Fig. 70.9 (*b*).

(*i*) When no signal is applied at the input, *i.e.*, $A = 0$ V, the transistor is cut OFF, making the collector current zero. Thus, the potential drop across R is zero. The supply voltage of V_{cc} appears at the output terminal. Hence, the output $Y = V_{cc}$.

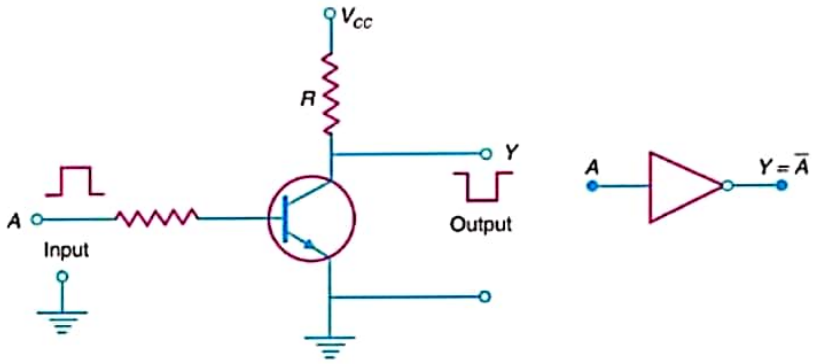
Thus, *when input is low, output is high.*

(*ii*) When a positive pulse is applied to A, *i.e.*, $A = 1$, the transistor conducts (fully ON) drawing maximum collector current. Hence, whole of V_{cc} drops across R and output $Y = 0$ V. *Thus, when input is high, the output is held at a low value.*

The results are tabulated in the Truth table.

Truth Table for NOT Gate

Input	*Output*
A	$Y = \overline{A}$
0	1
1	0



(*a*) (*b*)

Fig. 70.9

FLIP FLOPS

voltage levels that are complementary. If Q is high then $\overline{Q}$ is low and *vice-versa*. The output remains stable till it is changed by the input signal. The two output stable conditions are $Q = 1$, $\overline{Q} = 0$, and $\overline{Q} = 0$, $Q = 1$, and these are called states. $Q = 1$ is called the Set state or 1 state and $Q = 0$ is called the Reset state or 0 state.

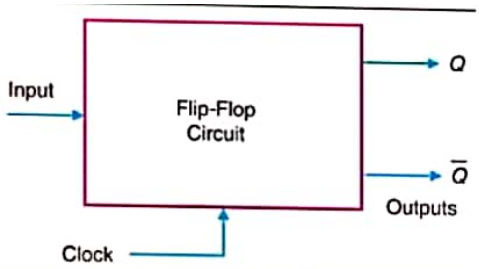


Fig. 70.10

70.25 Master-Slave Flip-Flop

A *master-slave flip-flop* is constructed from two separate flip-flops. One circuit serves as a master and the other as a slave. Fig 70.11 shows the logic diagram of an *RS* master-slave flip-flop. It consists of a master flip-flop, a slave flip-flop, and an inverter.

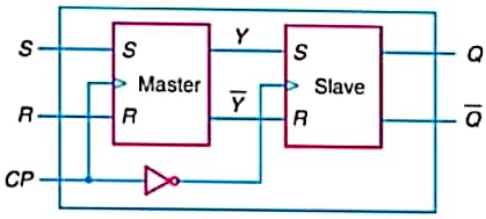


Fig. 70.11

When clock pulse CP is 0, the output of the inverter is 1. Since the clock input of the slave is 1, the flip-flop is enabled and output Q is equal to Y, while $\overline{Q}$ is equal to $\overline{Y}$. The master flip-flop is disabled because $CP = 0$. When the pulse becomes 1, the information then at the external R and S inputs is transmitted to the master flip-flop. The slave flip-flop, however, is isolated as long as the pulse is at its 1 level, because the output of the inverter is 0. When the pulse returns to 0, the master flip-flop is isolated, which prevents the external inputs from affecting it. The slave flip-flop then goes to the same state as the master flip-flop.

70.26 Master-Slave JK Flip-Flop

The circuit diagram is shown in Fig. 70.12. The clocks to the Master and Slave flip-flops are complementary to each other. It consists of two flip-flops. Gates 1 through 4 form the master flip-flop. Gates 5 through 8 form the slave flip-flop. The information present at the J and K inputs is transmitted to the master flip-flop on the positive edge of a clock pulse. It is held there until the negative edge of the clock pulse occurs, after which it is allowed to pass through to the slave flip-flop. The clock input is normally 0, which keeps the outputs of gates 1 and 2 at the 1 level. This prevents the J and K inputs from affecting the master flip-flop.

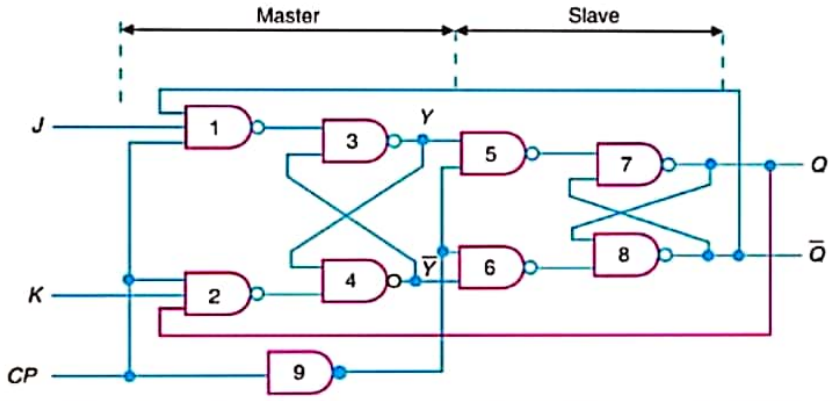


Fig. 70.12

The slave flip-flop is a clocked *RS* type, with the master flip-flop supplying the inputs and the clock input being inverted by gate 9. When the clock is 0, the output of gate 9 is 1. So, output $Q = Y$, and $\overline{Q} = \overline{Y}$.

When the positive edge of a clock pulse occurs, the master flip-flop is affected and may switch states. The slave flip-flop is isolated as long as the clock is at the 1 level, because the output of gate 9 provides a 1 to both inputs of the NAND basic flip-flop of gates 7 and 8. When the clock input returns to 0, the master flip-flop is isolated from the *J* and *K* inputs and the slave flip-flop goes to the same state as the master flip-flop.

70.27 Counters

In digital systems possibly the most widely used block is a counter. Counter is an instrument used for measuring time. A counter, which is driven by a clock can be used to count the number of clock pulses. There are two types of counters :

(*i*) Asynchronous (or) Binary ripple counters.

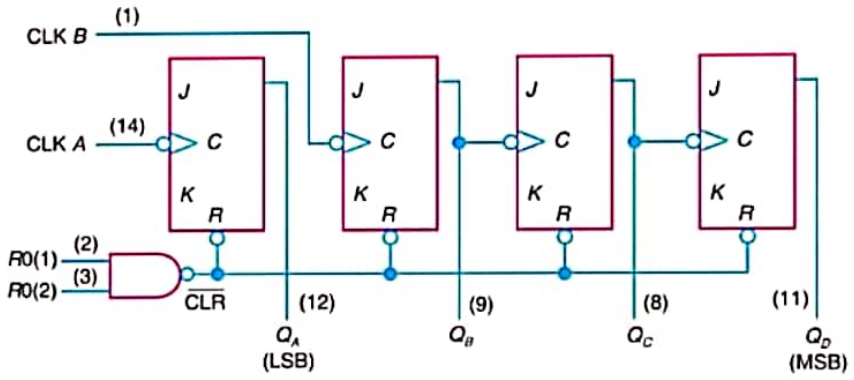
(*ii*) Synchronous counter (parallel counter).

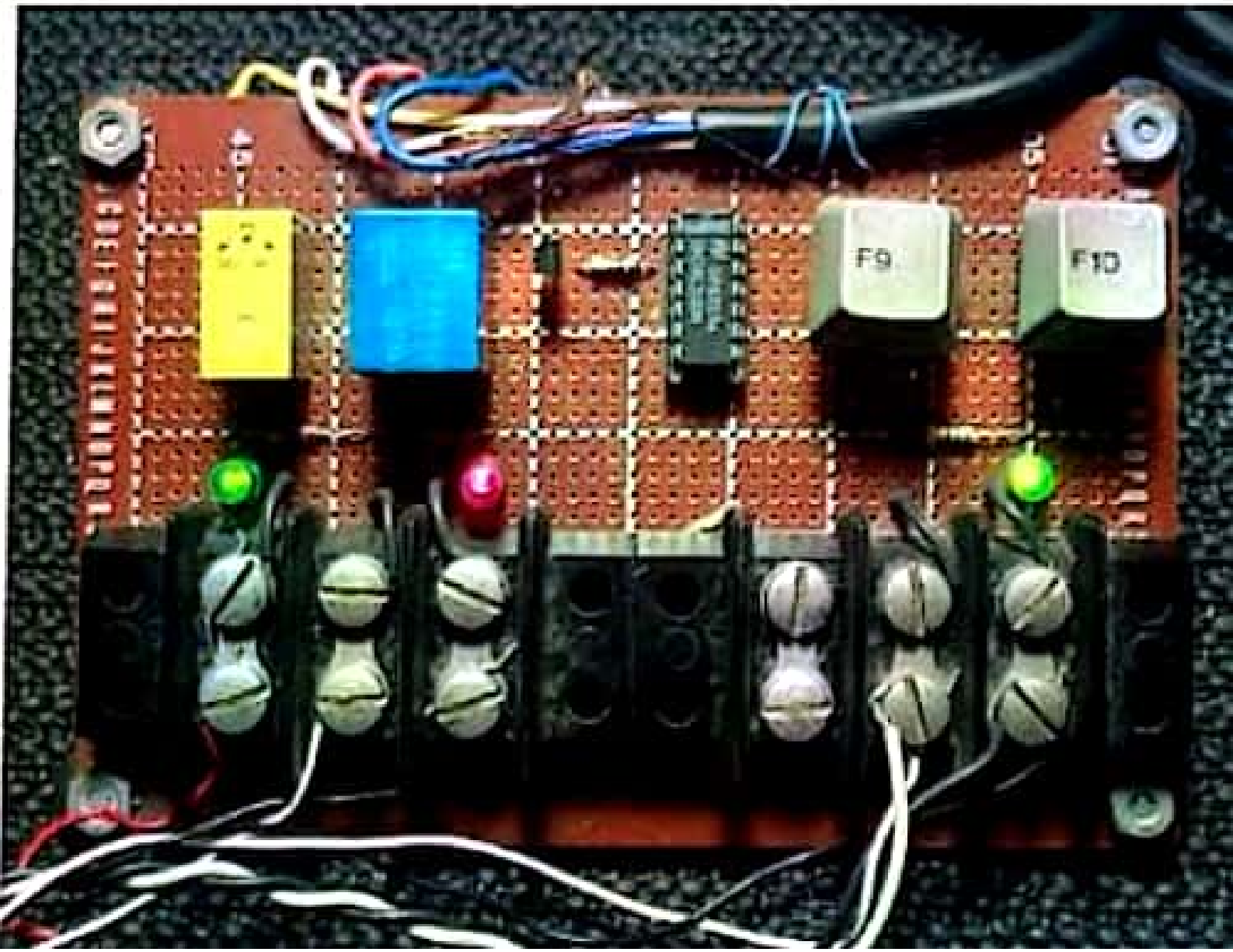
In asynchronous counters, the clock pulses (CK) are applied to the first Flip-Flop and the output of the first Flip-Flop is used as CK for the next Flip-Flop and so on. In asynchronous counters, the FFs are connected in series and each FF takes its own propagation time. So the net output occurs only after the total propagation time. Hence, its speed is limited.

In synchronous counters (or parallel counters), the CKs are applied simultaneously to all the FFs. So the net propagation time becomes less and hence its speed is increased.

70.28 The 7493 A Four-Bit Binary Counter

It is an integrated circuit asynchronous counter. The logic diagram is shown in Fig. 70.13. Pin numbers are in parentheses, and all J-K inputs are internally connected HIGH.





EXERCISE

1. What is binary number system? How does it differ from decimal number system? Why is the binary number system used in computers?
2. What are binary numbers? Explain with illustrations the methods of conversion from decimal to binary and binary to decimal numbers.
3. Write four basic rules for adding binary digits. Give the truth table for binary addition.
4. State the rules for binary subtraction. Explain 1's complement and 2's complement method with examples.
5. What are octal and hexadecimal number systems?
6. What is Boolean algebra? Mention its unique feature. Discuss the fundamental laws of Boolean algebra. [*Karnatak University, Dharwad, Oct. 93*]
7. Write and explain the Boolean expressions for OR, AND and NOT circuits.
8. State and prove De Morgan's theorems.
9. (*a*) Draw logic symbols and truth tables of AND, OR, NOT logic gates. Explain its operation for two-input condition.

   (*b*) Draw the circuits of two-input AND, OR gates. Explain these two circuits. [*April 95, KUD*]
10. Explain the operation of a NOT circuit. Give its truth table and logic symbol.
11. Discuss the operation of a JK master-slave flip-flop with necessary circuit.
12. Explain the operation of a "Four Bit Binary Counter".
13. Define a register. What is a shift register? Explain the operation of a Serial in-Serial out shift register.
14. Solve the Boolean expression $X = A + B$ for the following inputs:

    (*i*) $A = 0, B = 0$; (*ii*) $A = 1, B = 0$; (*iii*) $A = 1, B = 1$.

Sol. (*i*) When $A = 0, B = 0, X = 0 + 0 = 0$.

(*ii*) When $A = 1, B = 0, X = 1 + 0 = 1$.

(*iii*) When $A = 1, B = 1, X = 1 + 1 = 1$.

15. Prove the following Boolean identities.

    (*i*) $AC + ABC = AC$

    (*ii*) $A + \overline{A}B = A + B$

Sol. (*i*) $AC + ABC = AC(1 + B) = AC (\because 1 + B = 1)$

(*ii*) $A + \bar{A}B = A.1 + \bar{A}B$ [Law (4)]

$= A(1 + B) + \bar{A}B$ [Law (2)]

$= A.1 + AB + \bar{A}B$ (Distributive law)

$= A + B(A + \bar{A})$ (commutative law and distributive law)

$= A + B.1$ [Law (6)]

$= A + B.$

$\therefore \quad A + \bar{A}B = A + B.$

16. Reduce $AB + ABC + \bar{A}B + A\bar{B}C$ using laws of Boolean algebra.

Sol. $AB + ABC + \bar{A}B + A\bar{B}C$

$= AB + \bar{A}B + ABC + A\bar{B}C$ [commutative law]

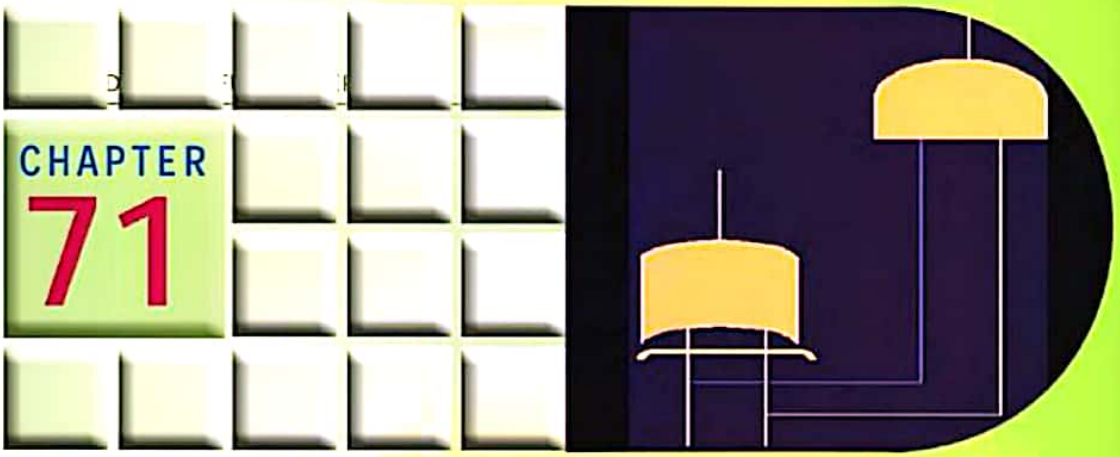
$= B(A + \bar{A}) + AC(B + \bar{B})$ [Distributive law and commutative Law]

$= B.1 + AC.1$ [Law (6)]

$= B + AC$ [Law (4)]

$\therefore \quad AB + ABC + \bar{A}B + A\bar{B}C = B + AC.$

Scanned with CamScanner



CHAPTER 71

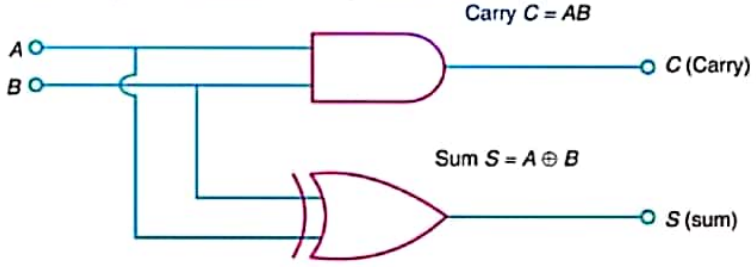
HALF-ADDER AND FULL ADDER

AT A GLANCE			
71.1	Half-Adder	71.2	Full Adder
71.3	Parallel Binary Adder		

71.1 Half-Adder

A logic circuit that adds two bits producing a sum and a carry to be used in the next higher position is called a *half-adder*.

Fig. 71.1 (*a*) shows the circuit of a half-adder. It consists of an EXCLUSIVE OR gate and an AND gate. The output of the exclusive OR gate is called the *SUM*, while the output of the AND gate is called the CARRY. Fig. 71.1 (*b*) shows the symbol of a half-adder.



(*a*) Half Adder

(*b*)

Fig. 71.1

The two inputs A and B represent the bits to be added.

S and C are the two outputs.

S represents the output of EXCLUSIVE OR gate.

C represents the CARRY bit. It is the output of AND gate.

With two inputs A and B, there are four distinct cases.

(I) When $A = 0$ and $B = 0$,

Carry $C = AB = 0$

Sum $S = A \oplus B = 0 \oplus 0 = 0$.

(II) When $A = 0$ and $B = 1$,

Carry $C = AB = 0.1 = 0$

Sum $S = A \oplus B = 0 \oplus 1 = 1$.

(III) When $A = 1$ and $B = 0$,

Carry $C = AB = 1.0 = 0$

Sum $S = A \oplus B = 1 \oplus 0 = 1$.

(IV) When $A = 1$ and $B = 1$,

Carry $C = AB = 1.1 = 1$

Sum $S = A \oplus B = 1 \oplus 1 = 0$.

The truth table for the Half-Adder operation is given in Table 1.

TRUTH TABLE 1

Input		*Output*	
A	*B*	*Sum (S)*	*Carry (C)*
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

In the least significant column,

$$1 + 1 = 10, \text{Sum} = 0, \text{Carry} = 1.$$

Thus for this operation, we need a half-adder.

In the next column, we must add 3 bits because of carry,

$$1 + 0 + 1 = 10, \text{Sum} = 0, \text{Carry} = 1.$$

By connecting two half-adders and an OR gate, we get a full adder (Fig. 71.2).

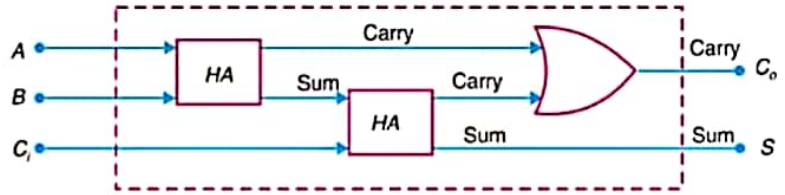
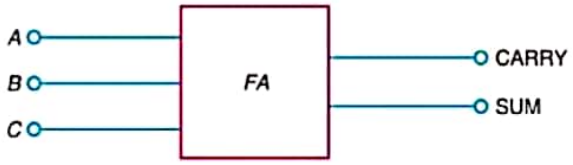


Fig. 71.2

It can add three bits at a time.

Fig. 71.3 shows the symbol of a full-adder. It has two inputs *A* and *B*, plus a third input *C*. Input *C* is also called the CARRY IN (C_i). It comes from a lower-order column. There are two outputs, *Sum* and *Carry*. The output *carry* is also called the *CARRY OUT* (C_o).



Working

To illustrate its operation, let us take two examples:

1. $A = 1, B = 1, C_I = 0$

Fig. 71.4 shows the full adder with these three inputs.

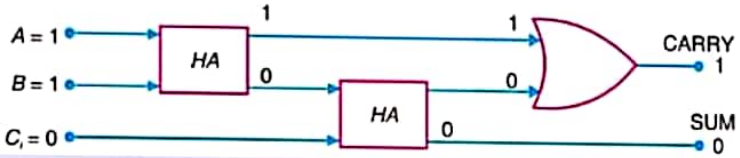


Fig. 71.4

(*i*) The output of first half adder consists of a sum of 0 with a carry of 1.

(*ii*) The sum 0 of first half adder and carry 0 when fed to the second half adder, give a sum of 0 with a carry of 0.

(*iii*) The carry outputs of both the half adders is fed into the input of OR gate.

The final output is : SUM 0, CARRY 1.

We get the same result from binary addition: $1 + 1 + 0 = 10_2$.

2. $A = 1, B = 1, C_I = 1$.

(*i*) The output of the first half adder is a sum of 0 with a carry 1 (Fig. 71.5).

(*ii*) The output of second half adder is a sum of 1 with a carry of 0.

(*iii*) The final output is SUM 1 with a CARRY 1.

We get the same result from binary addition: $1 + 1 + 1 = 11_2$.

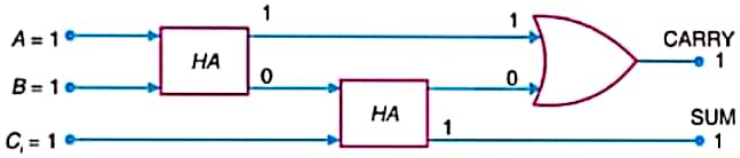


Fig. 71.5

71.3 Parallel Binary Adder

A parallel adder is used to add two numbers in parallel form and to produce the sum bits as parallel outputs. By cascading a number of full adders, we can get an n-bit binary adder circuit which is called a *parallel adder.*

Suppose we are required to add 4-bit binary numbers $A_4\ A_3\ A_2\ A_1$ and $B_4\ B_3\ B_2\ B_1$. Then we get a sum $S_4\ S_3\ S_2\ S_1$ such that

$$\begin{array}{r} A_4\ A_3\ A_2\ A_1 \\ +\ B_4\ B_3\ B_2\ B_1 \\ \hline S_5\ S_4\ S_3\ S_2\ S_1 \end{array}$$

Here, S_5 indicates overflow carry if the sum exceeds four bits. For adding them (two 4-bit numbers), we need *four full adders* (FA) *connected in parallel.* The first adder could, however, be a half adder (HA) because only two bits are to be added and there is no carry. But all subsequent columns need full adder because we have to handle three bits at a time (two binary digits and a carry generated by the previous column).

Fig. 71.7 shows a parallel four-bit binary adder.

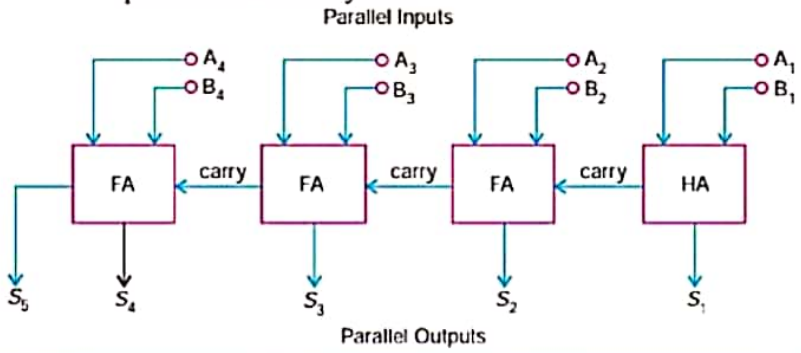


Fig. 71.7

To increase the capacity, more full adders may be connected. For example, for adding 6-bit numbers, we will have to add two more full adders to the left, thus making a total of six.

CHAPTER 72

BJT TRANSISTOR MODELING

AT A GLANCE			
72.1	Introduction	72.2	The Important Parameters: Z_i, Z_o, A_v, A_i
72.3	The r_e Transistor Model		

72.1 Introduction

...equivalent circuits (models).

(i) Input Impedance, Z_i

For the input side, the input impedance Z_i is defined by Ohm's law as the following :

$$Z_i = \frac{V_i}{I_i} \qquad ...(1)$$

If the input signal V_i is changed, the current I_i can be computed using the same level of input impedance. In other words :

For small-signal analysis, once the input impedance has been determined the same numerical value can be used for changing levels of applied signal.

In particular, for frequencies in the low to mid-range (typically ≤ 100 *kHz*):

The input impedance of a BJT transistor amplifier is purely resistive in nature, and depending on the manner in which the transistor is employed, can vary from a few ohms to megohms.

In addition :

An ohmmeter cannot be used to measure the small-signal ac input impedance since the ohmmeter operates in the dc mode.

Determination of Z_i

Eq. (1) provides a method for measuring the input resistance in the ac domain. In Fig. 72.2 a sensing resistor has been added to the input side to permit a determination of I_i using Ohm's law. An oscilloscope or sensitive digital multimeter (DMM) can be used to measure the voltage V_s and V_i. Both voltages can be the peak-to-peak, peak, or rms values, as long as both levels use the same standard. The input impedance is then determined in the following manner :

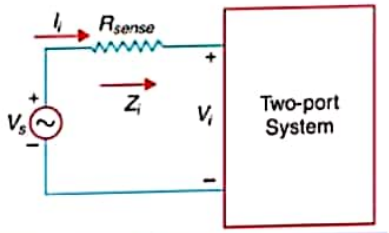


Fig. 72.2

$$I_i = \frac{V_s - V_i}{R_{sense}} \qquad ...(2)$$

and

$$Z_i = \frac{V_i}{I_i} \qquad ...(3)$$

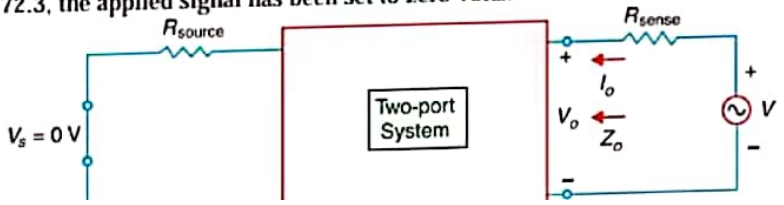
(ii) Output Impedance, Z_o

The output impedance of any amplifier is defined as the ratio of output voltage to the output current keeping the input current zero (input terminals left open).

Determination of Z_o

The output impedance is determined at the output terminals looking back into the system with the applied signal set to zero.

In Fig. 72.3, the applied signal has been set to zero volts.



$$I_o = \frac{V - V_o}{R_{sense}} \quad ...(4)$$

and

$$Z_o = \frac{V_o}{I_o} \quad ...(5)$$

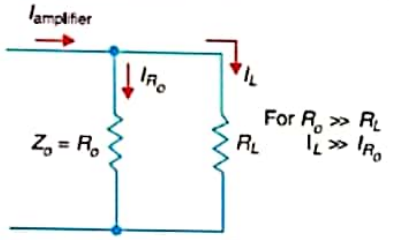


Fig. 72.4

In particular for frequencies in the low to mid-range (typically ≤ 100 kHz) :

The output impedance of a BJT transistor amplifier is resistive in nature and depending on the configuration and the placement of the resistive elements.

Z_o can vary from a few ohms to a level that can exceed 2 MΩ.

In addition :

An ohmmeter cannot be used to measure the small-signal ac output impedance since the ohmmeter operates in the dc mode.

Effect of $Z_o = R_o$ on the load or output current I_L.

For amplifier configurations where significant gain in current is desired, the level of Z_o should be as large as possible. If $Z_o \gg R_L$, the majority of the amplifier output current will pass on to the load (Fig. 72.4). Z_o is frequently so large compared to R_L that it can be replaced by an open-circuit equivalent.

(*iii*) Voltage Gain, A_v

The voltage gain of any amplifier is defined as the ratio of the output voltage to the input voltage.

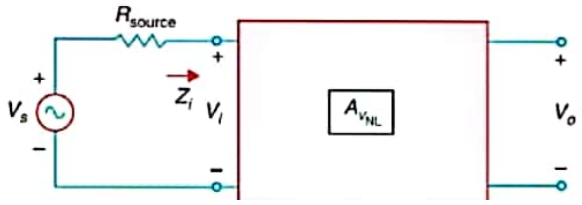
The small-signal ac voltage gain of an amplifier is

$$A_v = \frac{V_o}{V_i} \quad ...(6)$$

Determination of the no-load voltage gain.

For the system of Fig. 72.5, a load has not been connected to the output terminals. The level of gain determined by Eq. (6) is referred to as the no-load voltage gain. That is,

$$A_{v_{NL}} = \left.\frac{V_o}{V_i}\right|_{R_L = \infty\,\Omega\,(\text{open-circuit})} \quad ...(7)$$



$$A_{v_s} = \frac{V_o}{V_s} = \frac{V_i}{V_s} \cdot \frac{V_o}{V_i}$$

$$\therefore \quad A_{v_s} = \frac{V_o}{V_s} = \frac{Z_i}{Z_i + R_s} A_{v_{NL}} \qquad ...(8)$$

Experimentally, the voltage gain A_{v_s} or $A_{v_{NL}}$ can be determined simply by measuring the appropriate voltage levels with an oscilloscope or sensitive DMM and substituting into the appropriate equation.

Depending on the configuration, the magnitude of the voltage gain for a loaded single-stage transistor amplifier typically ranges from just less than 1 to a few hundred. A multistage (multiunit) system, however, can have a voltage gain in the thousands.

(*b*) Current Gain, A_i

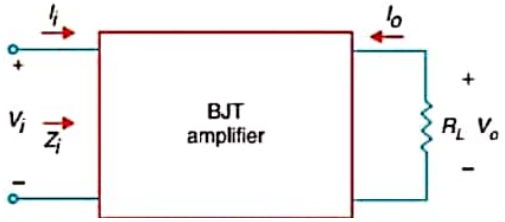
The current gain is defined by

$$A_i = \frac{I_o}{I_i} \qquad ...(9)$$

For BJT amplifiers, the current gain typically ranges from a level just less than 1 to a level that may exceed 100.

Determination of the loaded current gain.

For the loaded situation of Fig. 72.6,



BJT transistor amplifiers are referred to as current-controlled devices.

72.3.1. Common Base Configuration

In Fig. 72.7 (*a*), a common-base *pnp* transistor has been inserted within the two-port structure. In Fig. 72.7 (*b*), the r_e model for the transistor has been placed between the same four terminals.

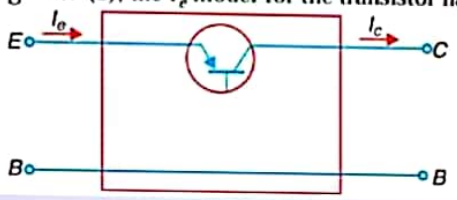


Fig. 72.7 (a)

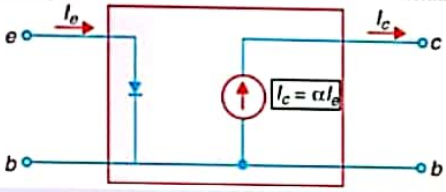


Fig. 72.7 (b)

One junction of an operating transistor is forward-biased while the other is reverse-biased.

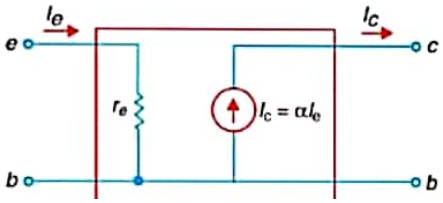
(*i*) The forward-biased junction will behave much like a diode (ignoring the effects of changing levels of V_{CE}). For the base-to-emitter junction of the transistor of Fig. 72.7 (*a*), the diode equivalence of Fig. 72.7 (*b*) between the same two terminals is quite appropriate.

(*ii*) The output characteristics for a common-base transistor amplifier indicate that $I_c \cong I_e$ (as derived from $I_c = \alpha I_e$) for the range of values of V_{CE}. The current source of Fig. 72.7(*b*) establishes the fact that $I_c = \alpha I_e$ with the controlling current I_e appearing in the input side of the equivalent circuit as dictated by Fig. 72.7 (*a*). We have therefore established an equivalence at the input and output terminals with the current-controlled source, providing a link between the two. So the model of Fig. 72.7 (*b*) is a valid model of the actual device.

The ac resistance of a diode can be determined by the equation $r_{ac} = 26\ \text{m}V/I_D$, where I_D is the dc current through the diode at the Q (quiescent) point. This same equation can be used to find the ac resistance of the diode of Fig. 72.7 (*b*) if we simply substitute the emitter current as follows:

$$r_e = \frac{26\ mV}{I_E} \qquad \ldots(1)$$

The subscript e of r_e emphasizes that it is the dc level of emitter current that determines the ac level of the resistance of the diode. Substituting the resulting value of r_e in Fig. 72.7 (*b*) results in the model of Fig. 72.8.



(*ii*) Consider the r_e Transistor Model of Fig. 72.8. If we set the signal to zero, then $I_e = 0\ A$ and $I_c = \alpha I_e = \alpha(0A) = 0A$, resulting in an open-circuit equivalence at the output terminals.

$$\therefore \qquad \boxed{Z_o \approx \infty\ \Omega}_{CB} \qquad \ldots(3)$$

For the common-base configuration, typical values of Z_o are in the megohm range.

Defining Z_o. The output resistance of the *C–B* configuration is determined by the slope of the characteristic lines of the output characteristics (Fig. 72.9). If Z_o is measured graphically or experimentally, levels typically in the range 1- to 2-MΩ are obtained. If we assume the lines to be perfectly horizontal (an excellent approximation), $Z_o \cong \infty\ \Omega$.

In general, for the common-base configuration the input impedance is relatively small and the output impedance quite high.

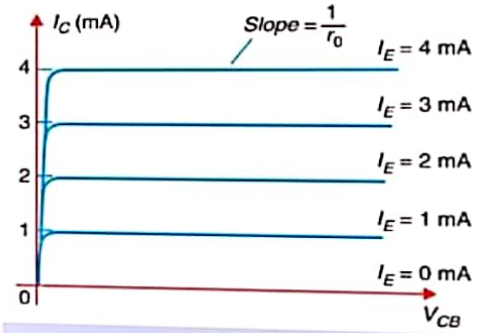
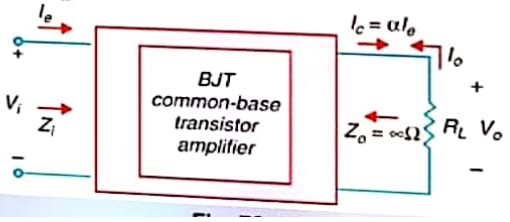


Fig. 72.9

(*iii*) We shall determine the voltage gain $A_v = V_o/V_i$ for the network of Fig. 72.10.



72.3.2. Common Emitter Configuration

Fig. 72.12 (*a*) shows the *C–E Configuration for an npn* transistor.

(*i*) The input terminals are the base and emitter terminals.

(*ii*) The output terminals are the collector and emitter terminals.

The emitter terminal is common between the input and output ports of the amplifier.

Fig. 72.12 (*b*) shows the approximate model for the configuration of Fig. 72.12 (*a*).

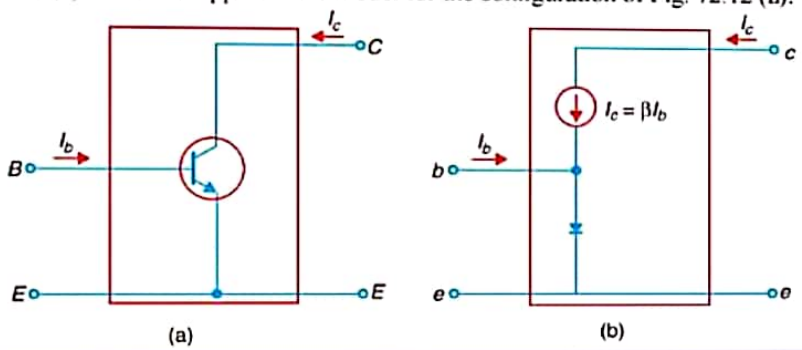


Fig. 72.12

The controlled-current source is connected between the collector and base terminals.

The diode is connected between the base and emitter terminals.

The base current I_b is the input current. The output current is I_c.

The base and collector currents are related by the following equation:

$$I_c = \beta I_b \quad \ldots(1)$$

The current through the diode is therefore determined by

$$I_e = I_c + I_b = \beta I_b + I_b$$

$$I_e = (\beta + 1) I_b \quad \ldots(2)$$

Since ac $\beta >> 1$, we will use the following approximation for the current analysis:

$$I_e \cong \beta I_b \quad \ldots(3)$$

The input impedance is determined by the

EXAMPLE. *If r_e = 6.5 Ω and β = 160, the input impedance has increased to a level of*

$Z_i = \beta r_e = (160)\ (6.5\ \Omega) = 1.04\ k\Omega$

For the C–E configuration, typical values of Z_i defined by βr_e range from a few hundred ohms to the kilohm range, with maximums of about 6 – 7 kΩ.

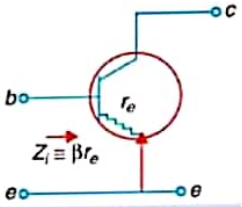


Fig. 72.14

(*ii*) Defining r_o for the C–E configuration

Fig. 72.15 shows the output characteristics of a silicon transistor. The slope of the curves increases with increase in collector current. The steeper the slope, the less the level of output impedance (Z_o).

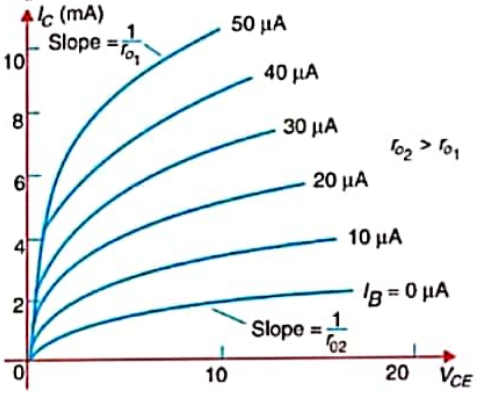


Fig. 72.15

Including r_0 in the transistor equivalent circuit

The r_e model of Fig. 72.12 does not include an output impedance, but if available from a graphical analysis or from data sheets it can be included as shown in Fig. 72.16.

For the C–E configuration, typical values of Z_o are in the range of 40 to 50kΩ.

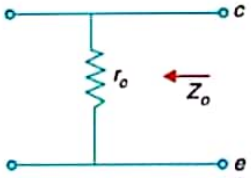


Fig. 72.16

For the model of Fig. 72.16, if the applied signal is set to zero, the current I_c is 0*A*.

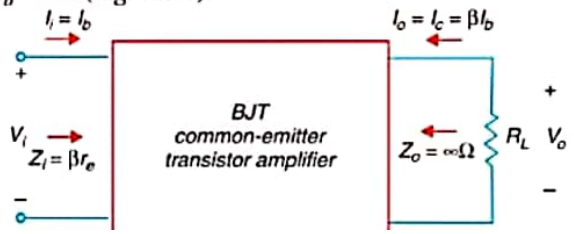
The output impedance is

$$\boxed{Z_o = r_o}_{CE} \qquad \text{...(5)}$$

If the contribution due to r_o is ignored as in the r_e model, the output impedance is defined by $Z_o = \infty\ \Omega$.

(*iii*) Determination of voltage gain for the C–E transistor amplifier.

We assume that $Z_o = \infty\Omega$ (Fig. 72.17).



For the defined direction of I_o and polarity of V_o

$$V_o = -I_o R_L$$

The minus sign simply reflects the fact that the direction of I_o in Fig. 72.17 would establish a voltage V_o with the opposite polarity.

$$V_o = -I_o R_L = -I_c R_L = -\beta I_b R_L$$

and

$$V_i = I_i Z_i = I_b \beta r_e$$

$$A_v = \frac{V_o}{V_i} = -\frac{\beta I_b R_L}{I_b \beta r_e}$$

$\therefore$

$$\boxed{A_v = -\frac{R_L}{r_e}}_{CE,\ r_o = \infty \Omega} \qquad ...(6)$$

The minus sign for the voltage gain reveals that the output and input voltages are 180° out of phase.

(*ii*) Determination of current gain

The current gain for the configuration of Fig. 72.17 is

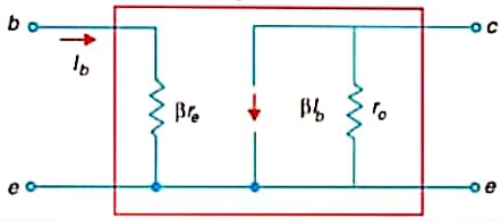


Fig. 72.18