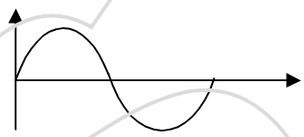
INDEX

CHAPTER 1

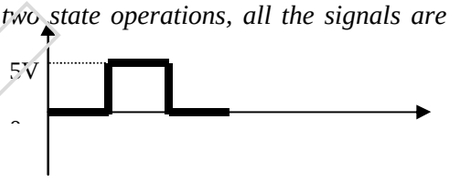
INTRODUCTION

1.1 Analog and Digital Signals

(i) Analog signal: A continuously varying signal *(voltage or current)* is called an analog signal. For example, an Alternating Current *varying sinusoidally is an analog signal. If such an analog signal is applied to the input of a transistor amplifier, the output voltage will also vary sinusoidally. This is the analog operation i.e. the output voltage can have an infinite number of values. Due to many valued output, the analog operation is less reliable.*



(ii) Digital signal: A signal *(voltage or current)* that can have only two discrete (separate) values is called a digital signal, for example, a square wave. *It is because this signal has only two values viz, +5V and 0V and no other value. These values are labeled as high and low. The high voltage is + 5V and the low voltage is 0V. If proper digital signal is applied to the input of a transistor, the transistor can be driven between cut off and saturation. In other words, the transistor will have two state operation i.e. output is either low or high. Since digital operation has only two states (i.e. ON or OFF),so it is far more reliable than many value analog operation. It is because with two state operations, all the signals are easily recognised as either low or high.*



1.2. Digital Circuit

An electronic circuit that handles only a digital signal is called a digital circuit.

The output voltage of a digital circuit is either low or high and no other value. In other words, digital operation is a two state operation. These are (High or Low) or (ON or OFF) or (1 or 0). The digital system uses only two digits, 1 and 0 and is called the binary number system.

1.3. Binary Number System

A number system is a code that uses symbols to count the number of items. The most common and familiar number system is the decimal number system. The decimal number system uses symbols 0 to 9. Thus the decimal system uses 10 digits for counting the items. A binary system uses only two digits 0 and 1 for counting the items.

1.4. Counting in Decimal and Binary System

Decimal	Binary
0	0
1	1
2	10
3	11
4	100
5	101
6	110
7	111
8	1000
9	1001

a) Each binary digit (Bit) 0 or 1 is referred to as a Bit. A string of four bits is called a Nibble and eight bits make a Byte.

b) The binary number system is the most useful in digital circuits because there are only two digits (0 and 1).

1.5. Place Value

For the decimal numbers, the digit to the extreme right is referred to as the Least Significant Digit (LSD) because its positional value is the lowest.

The left most digit in the decimal number is the Most Significant Digit (MSD) because its positional value is the highest.

For binary numbers, the digit at the extreme right is referred to as Least Significant Bit (LSB). The left most digit is called the Most Significant Bit (MSB).

EXERCISE 1

Q.1 Fill in the blanks

(i) A continuously varying signal is called _ _____. (Analog signals)

(ii) A signal that has only two different valued is called _____.(Digital Signal)

(iii) The number system used by the digital circuit is called _____. (Binary number system)

(iv) In a number, the digit to the extreme right is called _____. (Least Significant Digit).

CHAPTER 2

NUMBER SYSTEM

2.1. Introduction

Decimal number system is the most common number system about which we all are aware of but modern digital computers do not understand this system because they are based on the binary system i.e. 1 and 0.

a) Binary Number System - Binary system uses 2 digits or symbols i.e. 0 and 1, therefore its base is 2.

b) Octal Number System – Octal system uses 8 digits or symbols ie. 0 to 7, therefore its base is 8.

c) Hexadecimal Number System – Hexadecimal uses digits 0 to 9 and alphabets A to F ie. 16 symbols, therefore its base is 16.

2.2. Comparison of Decimal, Binary, Octal and Hexadecimal

Decimal	Binary	Octal	Hexadecimal
0	0000	0	0
1	0001	1	1
2	0010	2	2
3	0011	3	3
4	0100	4	4
5	0101	5	5
6	0110	6	6
7	0111	7	7
8	1000	10	8
9	1001	11	9
10	1010	12	A
11	1011	13	B
12	1100	14	C
13	1101	15	D
14	1110	16	E
15	1111	17	F

EXERCISE –2

Q.1 Fill in the blanks

(i) The number system, which we use in our day-to-day life is _____. (Decimal number system)

(ii) The number system used by the digital computers is _____. (Binary number system).

(iii) The number of symbols used by a number system is called _____. (Base)

(iv) Base of Hexadecimal number system is _____. (16)

(v) The Hexadecimal number system uses _____ (Digits 0 to 9 and alphabets A to F)

CHAPTER 3

CONVERSION OF NUMBERS

3.1. Decimal to Binary Conversion

Integer

a) Divide the decimal number by 2 until you get a quotient of 0.

b) Note down the remainders from bottom (MSB) to top (LSB).

Example: Convert $(14)_{10}$ into Binary.

2	**14**	**R**
2	7	0
2	3	1
2	1	1

The binary equivalent will be 1110, so $(14)_{10} = (1110)_{2.}$

Fraction

a) Multiply the fraction by 2.

b) Note down the carry from top (MSB) to bottom (LSB).

Example: Convert $(0.24)_{10}$ into Binary

0.24	x	2	=	0.48	-	0
0.48	x	2	=	0.96	-	0
0.96	x	2	=	1.92	-	1
0.92	x	2	=	1.84	-	1

$(0.24)_{10} = (0011)_2$

Note: If we will convert to base 8 or base 16, 8 and 16 would replace the value of 2.

3.2. **Decimal to Octal Conversion**

Integer

a) *Divide the decimal number by 8 until you get a quotient of 0.*

b) *Write down the remainders from bottom (MSD) to top (LSD).*

Example: Convert $(87)_{10}$ to Octal

8	87	R	
8	10	7	LSD
8	1	2	

$(87)_{10} = (127)_8$

Fraction

a) *Multiply the decimal fraction by 8.*

b) *Write down the carry from top (MSD) to bottom (LSD)*

Example: Convert (0.11) 10 to Octal

0.11	X	8	=	0.88	-	0
0.88	X	8	=	7.04	-	7
0.04	X	8	=	0.32	-	0
0.32	X	8	=	2.56	-	2

$(0.11)_{10} = (.0702)_8$

3.3. **Decimal To Hexadecimal Conversion**

Integer

a) *Divide the decimal number by 16 until you get a quotient of 0.*

b) *Write down the remainders from bottom (MSD) to top (LSD).*

Example: Convert $(87)_{10}$ to Hexadecimal

16	**87**	**R**	
16	5	7	LSD

$(87)_{10} = (57)_{16}$

Fraction

a) *Multiply the decimal fraction by 16.*

b) *Write down the carry from top (MSD) to bottom (LSD).*

0.02 X 16 = 0.32 - 0

0.32 X 16 = 5.12 - 5

0.12 X 16 = 1.92 - 1

0.92 X 16 = 14.72 - E

$(0.02)_{10} = (.051E)_{16}$

3.4. **Binary to Decimal Conversion**

Integer

a) Write the binary number.

b) Multiply each bit by 2.

c) Assign the powers to all 2s starting from LSB as 0, 1, 2 and so on.

d) Add all the values obtained by multiplication.

Example: Convert $(1110)_2$ to Decimal

$1110_2 = 1x2^3 + 1x2^2 + 1x2^1 + 0x2^0$

$= 1x8 + 1x4 + 1x2 + 0x1$

$= 8 + 4 + 2 + 0$

$= 14$

$(1110)_2 = (14)_{10}$

Fraction

a) Write the binary fraction.

b) Multiply each bit 2.

c) Assign powers to all 2s starting from MSB as –1, -2, -3 and so on.

d) Add all the values obtained by multiplication.

Example: Convert $(.111)_2$ to Decimal

0.111_2 = $1x2^{-1}$ + $1x2^{-2}$ + $1x2^{-3}$

= ½ ¼ 1/8

= 0.5 0.25 0.125

= 0.875

$(.111)_2 = (0.875)_{10}$

3.5. Octal to Decimal Conversion

Integer

a) Multiply each digit by 8.

b) Assign powers to all 8s starting from LSD as 0, 1, 2 and so on.

c) Add all the products to get the decimal equivalent.

Example: Convert $(165)_8$ to Decimal

$(165)_8$ = $1x8^2$ + $6x8^1$ + $5x8^0$

= 1x64 + 6x8 + 5x1

= 64 + 48 + 5

= 117

$(165)_8 = (117)_{10}$

Fraction

a) Multiply each fraction digit by 8.

b) Assign powers to all 8s starting from MSB as 0, 1, 2 and so on.

c) Add all the products to get the decimal equivalent.

Example: Convert (0.52)8 to Decimal

$0.52_8 = 5x8^{-1} + 2x8^{-2}$

$= 5/8 + 2/64$

$= 5/8 + 1/32$

$= \frac{20 + 1}{32} =$

$= \frac{21}{32} = 0.65625$

$(.52)_8$ = ($0.65625)_{10}$

3.6. **Hexadecimal to Decimal Conversion**

Integer

a) Multiply each hexadecimal digit by 16.

b) Assign the powers to all 16s starting from LSD as 0, 1, 2 and so on.

c) Add all the products to get decimal equivalent.

Example: Convert $(A9E)_{16}$ to Decimal

$A9E_{16} = Ax16^2 + 9x16^1 + Ex16^0$

$= 10x16^2 + 9x16^1 + 14x16^0$

$= 10x256 + 9x16 + 14x1$

$= 2560 + 144 + 14$

$= 2718$

$(A9E)_{16} = (2718)_{10}$

Fraction

a) Multiply each fraction digit by 16.

b) Assign powers to all 16sstarting from MSB as -1, -2, -3 and so on.

c) Add all the products to get decimal equivalent.

Example: Convert **(.1C)16** to Decimal.

$.1C_{16} = 1x16^{-1} + Cx16^{-2}$

$= 1/16 + 12/256$

$= \frac{16 + 12}{256} = \frac{28}{256}$

$= 0.109375$

$(.1C)_{16} = (0.109375)_{10}$

3.7. **Binary to Octal Conversion**

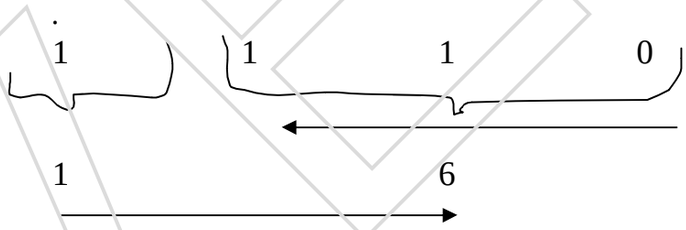
Integer

a) Make group of 3 bits from right (LSB) to left (MSB).

b) Convert these groups into their Octal equivalent.

c) Write the octal equivalents from left (MSD) to right (LSD).

Example: Convert $(1110)_2$ to octal



$(1110)_2 = (16)_8$

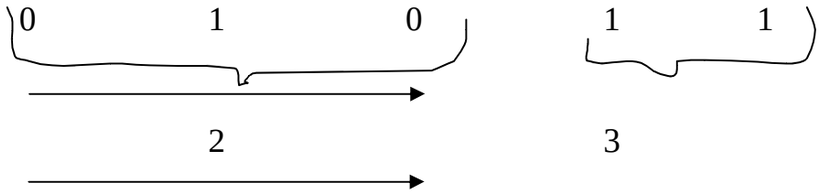
Fractions

a) Make group of 3 bits from left (MSB) to right (LSB).

b) Convert these groups into their octal equivalents.

c) Write the octal equivalents from left (MSD) to right (LSD).

Example: Convert $(.01011)_2$ to Octal.



$(.01011)_2 = (.23)_8$

3.8. **Binary to Hexadecimal Conversion**

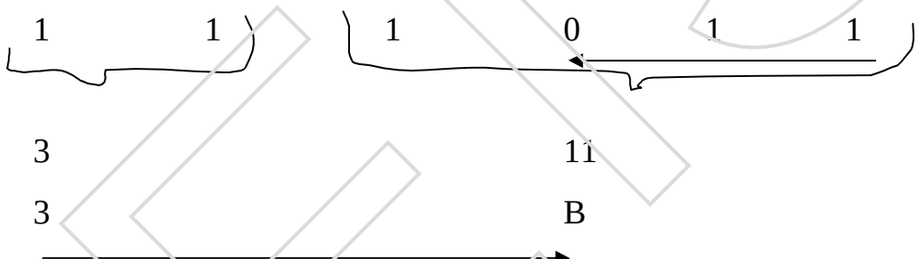
Integer

a) Make group of 4 bits from right (LSB) to (MSB).

b) Convert these groups into their hexadecimal equivalent.

c) Write the hexadecimal equivalents from left (MSD) to right (LSD).

Example: Convert $(111011)_2$ to Hexadecimal



$(1110)_2 = (3B)_{16}$

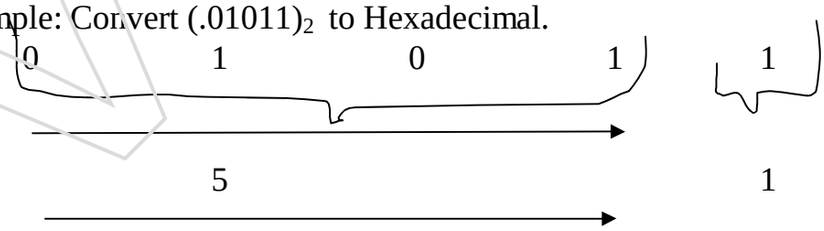
Fractions

a) Make group of 4 bits from left (MSB) to right (LSB).

b) Convert these groups into their hexadecimal equivalent.

c) Write the hexadecimal equivalents from left (MSD) to right (LSD).

Example: Convert $(.01011)_2$ to Hexadecimal.



$(.01011)_2 = (.51)_{16}$

3.9. **Hexadecimal Decimal to Binary Conversion**

Integer

a) Convert each hexadecimal digit into its 4 bit binary equivalent.

b) Write the binary bits in the same order to get the binary equivalent.

Example: Convert $(2D5)_{16}$ to Binary.

$(2D5)_{16}$	=	2	D	5
	=	0010	1101	0101
$(2D5)_{16}$	=	$(1011010101)_2$		

Fraction

a) Convert each hexadecimal digit into its 4 bit binary equivalent.

b) Write the binary bits in the same order to get the binary equivalent.

Example: Convert $(.47)_{16}$ to Binary.

(.47) 16 = (.0100 0111) 2

(.47) 16 = (.01000111) 2

3.10. **Hexadecimal to Octal Conversion**

Integer

a) First convert hexadecimal into its binary equivalent.

b) Then make groups of 3 bits from right (LSB) to left (MSB).

c) Then write the octal equivalents of these groups.

NOTE: This conversion is also possible by first converting hexadecimal into its decimal equivalent and later converting decimal equivalent into its octal equivalent.

Example: Convert $(47)_{16}$ to Octal

$(47)_{16} = (0100 \quad 0111)_2$

$= (01 \quad 000 \quad 111)_2$

$= (1 \quad 0 \quad 7)_8$

$(47)_{16} = (107)_8$

Fraction

a) First convert hexadecimal into its binary equivalent by converting each hexadecimal digit into its 4 bit binary equivalent.

b) Make groups of 3 bits from left (MSB) to right (LSB).

c) Convert each 3 bit group into its octal equivalent.

Example: Convert $(.4C)_{16}$ to Octal.

$(.4C)_{16} = (.0100 \ 1100)_2$

$(.01001100)_2 = (.010 \ 011 \ 000)_2$

2 3 0

$(.4C)_{16} = (.230)_8$

3.11. **Octal to Hexadecimal Conversion**

Integer

a) First convert octal into its binary equivalent by converting each octal digit into 3 bit binary equivalent.

b) Make group of 4 bits from right (LSB) to left (MSB).

c) Convert each 4 bit group into its hexadecimal equivalent.

Example: Convert $(67)_8$ to Hexadecimal.

$(67)_8 = (110\ \ 111)_2$

$(110111)_2 = 0011\ \ 0111$

3 7

←--------

$(67)_8 = (37)_{16}$

Fraction

a) First convert each fraction digit into its binary equivalent by converting each digit into 3 bit equivalent.

b) Make groups of 4 bits from left (MSB) to right (LSB).

c) Convert each 4 bit group into its hexadecimal equivalent.

Example: Convert $(.23)_8$ to Hexadecimal.

$(.23)_8 = (.010\ \ 011)_2$

$(.010011)_2 = .0100\ \ 1100$

4 12

C

--------→

$(.23)_8 = (.4C)_{16}$

Note: This conversion is also possible by first converting octal into its decimal equivalent and then decimal into hexadecimal.

3.12. **Octal to Binary Conversion**

Integer

a) Convert each octal digit into its 3 bit binary equivalent.

b) Write the binary bits in the same order to get octal equivalent.

Example: Convert $(23)_8$ to Binary.

$(23)_8 = (010\ \ 011)_2$

$(23)_8 = (010011)_2$

Fraction

a) Convert each fractional digit into its 3 bit binary equivalent.

b) Write the binary bits in the same order to get the octal equivalent.

Example: Convert $(.37)_8$ to Binary.

$(.37)_8 = (.011\ \ 111)_2$

$(.37)_8 = (.011111)_2$

EXERCISE 3

Q.1 Fill in the blanks:-

(i) Binary, octal and hexadecimal equivalent of decimal number 14 is _____.

(ii) Binary number can be converted into its octal equivalent by making _____. (sets of three bits from LSB).

(iii) Binary number can be converted into its hexadecimal equivalent by making _____. (sets of four bits from LSB).

Q. 2 Convert

(i) $(1110.111)_2$ = (_____)$_{10}$ [14.875]

(ii) $(87.11)_{10}$ = (_____)$_8$ [127.0702]

(iii) $(1110.01011)_2$ = (_____)$_8$ [16.26]

(iv) $(111011)_2$ = (_____)$_{16}$ [3B]

CHAPTER 4

ARITHMETIC OPERATIONS IN BINARY SYSTEM

4.1. Binary Addition

Rules

0 + 0 = 0

0 + 1 = 1

1 + 0 = 1

1 + 1 = 0 plus a carry of 1 to next higher column

Example 1: Add the binary number 10011 and 1001.

Carry		**1**	**1**		
MSB				LSB	
	1	**0**	**0**	**1**	**1**
	+	**1**	**0**	**0**	**1**
	---	**---**	**---**	**---**	**---**
	1	**1**	**1**	**0**	**0**
	---	**---**	**---**	**---**	**---**

Example 2: Add the binary number 100111 and 11011.

Carry	**1**	**1**	**1**	**1**	**1**	
	1	**0**	**0**	**1**	**1**	**1**
	+	**1**	**1**	**0**	**1**	**1**
	---	**---**	**---**	**---**	**---**	**---**
1	**0**	**0**	**0**	**0**	**1**	**0**
	---	**---**	**---**	**---**	**---**	**---**

4.2. **Binary Subtraction**

Subtraction is done by direct method and complement method. Computer does the subtraction by the complement method.

a) Direct Method

Rules

0 - 0 = 0

0 - 1 = 1 with a borrow from the next column

1 - 0 = 1

1 - 1 = 0

Example 1: Subtract the binary number 1110 from 10101.

```
    1   0   1   0   1
-   0   1   1   1   0
    ---------------------------------
    0   0   1   1   1
```

In the first column, o is subtracted from 1. No borrow is required in this case and the result is 1. In the second column, we have to subtract 1 from 0, borrow is necessary to perform this subtraction. So a 1 is borrowed from the third column, which becomes 2 in the second column because the base is 2. Now in the second column, we subtract 1 from 2 giving a result of 1. The borrow performed in the second column reduces the 1 in the third column to 0. So in the third column, once again we have to subtract 1 from 0 for which borrow is required. The fourth column contains a 0 and thus has nothing to borrow. Therefore we have to borrow from the fifth column. Borrowing 1 from the fifth column gives 2 in the fourth column. Now fourth column has something to borrow. When 1 of the 2 in the fourth column is borrowed, it becomes 2 in the third column. Now in the third column, we subtract 1 from 2 giving a result of 1. The borrow performed in the third column reduces the 1 in the fifth column to 0 and the 2 in the fourth column to 1. Hence, subtraction of the fourth column is now 1 from 1 giving 0. Thus the final result of subtraction is 00111_2.

Example 2: Subtract the binary number 01110000 from 101110.

```
    1   0   1   1   1   0   0
-   0   1   1   1   0   0   0
    -----------------------------------------------
    0   1   0   0   1   0   0
```

---

b) Complement Method

Complements method is used in digital computers for simplifying the subtraction operation and for logic manipulations.

Complements are of two types. These are as follows:

(i) 1's complement

(ii) 2's complement

(i) Subtraction by 1's complement

In the 1's complement each 1 is changed to 0 and 0 to 1.

Number	1's Complement
100	011
1010	0101
111	000

Example 1: Subtract 101 from 111

Take 1's complement of number to be subtracted. Add it to the given number.

```
    1   1   1
+   0   1   0     (1's complement of 101)
    -------------
1   0   0   1
    -------------
```

Fourth (Extra) bit (left bit) is called End Around Carry (EAC). Add this EAC to the remaining three bits. When EAC is present means answer is positive.

```
    0   0   1
+           1
    -------------
    0   1   0     (final answer)
    -------------
```

Example 2: Subtract 1101 from 1010

```
    1   0   1   0
+   0   0   1   0    (1's complement of 1101)
    -------------------------
    1   1   0   0
```

No EAC means answer will be negative. To get the answer, convert this result in the 1's complement.

```
-   0   0   1   1    (1's complement of 1100).
```

ii) Subtraction by 2's complement

2's complement is 1's complement +1.

Number	1's Complement	2's Complement
100	011	011+1=100
1010	0101	0101+1=0110
111	000	000+1=001

Example 1: Subtract 101 from 111

Take 2's complement of number to be subtracted. Add this to the given number.

```
    1   1   1
+   0   1   1    (2's complement of 101)
    -----------------
1   0   1   0
```

Discard this EAC. When EAC is present means answer is positive.

```
    0   1   0    (final answer)
```

Example 2: Subtract 1101 from 1010

```
    1   0   1   0
+   0   0   1   1   (2's complement of 1101)
    ---------------
    1   1   0   1
```

No EAC means answer will be in negative, to get the answer convert this value in the 2's complement.

```
-   0   0   1   1   (2's complement of 1101).
```

4.3. **Multiplication**

Multiply 101 with 101

```
                1   0   1
                1   0   1
        ---------------------
                1   0   1
            0   0   0   x
        1   0   1   x   x
        ---------------------
        1   1   0   0   1
        ---------------------
```

4.4. **Division**

```
101 | 11011 | 101
      101
    -------
      111
      101
    -------
       10
    -------
```

EXERCISE – 4

Q. 1 Fill in the blanks

(i) 1's complement of 100 is _____. (011)
(ii) 2's complement is _____. (1's complement + 1)
(iii) In 2's complement method, EAC is _____. (discarded)
(iv) In 2's complement method, if there is no EAC, it means answer is _____. (negative)
(v) EAC stands for _____. (End Around Carry).

Q. 2 Addition / Subtraction

(i) Add 10011 and 1001
(ii) Add 110011 and 1111
(iii) Subtract 101 from 111 by 1's complement
(iv) Subtract 1101 from 1010 by 1's complement
(v) Subtract 101 from 111 by 2's complement
(vi) Subtract 1101 from 1010 by 2's complement

CHAPTER 5
CODES

5.1. Codes

Codes are the symbolic representation of information, which may be present in the form of numbers or letters. The symbols used are the binary digits 1 or 0, which are arranged according to the rules of codes. These codes are used to communicate information to a digital computer to retrieve messages from it. A code is used to enable on operation to feed data into a computer directly in the form of decimal numbers, alphabets and special characters. The computer converts these data in to binary codes and after computation, transforms the data into its original format (decimal number alphabets & special character).

5.2. Classification of Codes

Codes are classified into the following four groups:

(i) Weighted binary codes

(ii) Non-weighted binary codes

(iii) Error detecting codes

(iv) Alpha numeric codes

5.3.. Weighted Binary codes

Each position of a number represents specific weight. In weighted binary codes, the bits are multiplied by the weights indicated. The sum of these weighted bits gives the decimal digit.

Decimal No	8421	5421	2421
0	0000	0000	0000
1	0001	0001	0001
2	0010	0010	0010
3	0011	0011	0011
4	0100	0100	0100
5	0101	1000	1011
6	0110	1001	1100
7	0111	1010	1101
8	1000	1011	1110
9	1001	1100	1111

BCD OR 8421 CODE

Binary coded decimal (BCD) specify the decimal numbers 0 to 9. It has four bits. The weights are assigned according to the positions occupied by the digits. The weight of the first (right most) is 2^0 ie. (1), the second is 2^1 ie. (2), the third is 2^2 ie.(4) and the fourth is 2^3 ie. (8). Reading from left to right it becomes 8421, so it is also called as 8421 code.

The binary equivalent of 7 is $[111]_2$ but BCD is $[0111]_2$ i.e. in four bits.

5.4. **Non–Weighted Codes**

Non- weighted codes are the codes that are not positionally weighted. This means that each position within a binary number in not assigned a fixed value. For example Excess-3 codes and Gray codes.

a) Excess- 3 code

An excess-3 code obtains by adding 3 to a decimal number and then converting it into the binary.

Decimal No.	BCD	Excess-3
0	0000	0011
1	0001	0100
2	0010	0101
3	0011	0110
4	0100	0111
5	0101	1000
6	0110	1001
7	0111	1010
8	1000	1011
9	1001	1100

b) Gray Code

It is a minimum change code in which only one bit in the code group changes when moving form one step to the next step.

Decimal No.	BCD	Gray Code
0	0000	0000
1	0001	0001
2	0010	0011
3	0011	0010
4	0100	0110
5	0101	0111
6	0110	0101
7	0111	0100
8	1000	1100
9	1001	1101

5.5. **Error–Detection Codes**

It is used to detect errors during transmission. An extra bit is included with the message to make the total number of 1's either odd or even. The extra bit is called parity. The parity is either odd or even.

5.6. **Alphanumeric Codes**

The numbers, letters and the special symbols are called alphanumeric codes. The alphanumeric codes are ASCII and EBCDIC.

ASCII: The ASCII is pronounced as "as–kee". ASCII stands for American Standard Code for Information Interchange. Most of the microcomputers manufacturers use this code. The ASCII code represents a character with seven bits with one additional zero bit.

EBCDIC (Extended Binary Coded Decimal Interchange Code): It is an 8-bit code. It differs from ASCII only in its code grouping for different alphanumeric characters.

Exercise – 5

Q. 1 Fill in the blanks

(i) Different groups of codes are _____. (weighted, non-weighted, error detecting, alpha numeric).
(ii) Decimal number 9 can be written in 8421 code as ____. (1001)
(iii) BCD stands for _____. (binary coded decimal).
(iv) Another name of BCD code is _____. (8421 code)
(v) Excess-3 code is a _____. (non-weighted code)
(vi) In _____ code only one bit changes in the subsequent steps. (Gray)

Q. 2 Short notes:

(i) Computer codes
(ii) Weighted binary codes
(iii) Error detecting codes
(iv) Alpha numeric codes

CHAPTER 6
BOOLEAN ALGEBRA

6.1. Boolean Algebra

George Boole invented Boolean algebra in 1854. Boolean algebra is used in simplifying logic circuits. Simplification of Boolean logic reduces the hardware required to design a specific circuit.

Digital circuits perform the binary arithmetic operations with binary digits 0 and 1. These operations are called logic functions or logical operations. The algebra used to symbolically describe logic functions is called Boolean Algebra. Boolean algebra is a set of rules and theorems by which logical operations can be expressed symbolically in equation form and can be manipulated mathematically. There are four connecting symbols used in Boolean algebra which are as follows:

a) Equal Sign (=) –

It refers to the standard mathematical equality e.g. A = B.

b) Plus Sign (+) –

It refers to the logical OR operation e.g. A+B = 1.

c) Multiply Sign (.) –

It refers to the logical AND operation e.g. A.B = 1.

d) Bar Sign (_ /') –

It refers to NOT operation e.g. if A = 1, then A' = 0.

6.2. Boolean Theorems

There are some basic Boolean theorems that are useful in manipulating and simplifying Boolean expressions. For convenience, the theorems can be divided into two groups, which are as follows:

a) Single Variable Theorems

b) Multi Variable Theorems

6.3. **Single Variable Theorems**

These theorems refer to the condition when only one input to the logic gate is variable.

a) Theorem 1 : A+0 = A

b) Theorem 2 : A.1 = A

c) Theorem 3 : A+A' = 1

d) Theorem 4 : A . A'= 0

e) Theorem 5 : A+A = A

f) Theorem 6 : A . A = A

g) Theorem 7 : A+1 = 1

h) Theorem 8 : A .0 = 0

i) Theorem 9 : A" = A

6.4 **Multivariable Theorems**

These theorems refer to the condition when more than one input to the logic gate are variable.

a) Theorem 10 : A+B = B+A (Commutative Law)

b) Theorem 11: A.B = B.A ('')

c) Theorem 12 : A+(B+C) = (A+B)+C (Associative Law)

d) Theorem 13: A.(B.C) = (A.B) .C (,,)

e) Theorem 14 : A.(B+C) = AB+A.C (Distributive Law)

f) Theorem 15 : (A+B).(C+D)= A.C+B.C+A.D+B.D (,,)

g) Theorem 16 : A+A.B=A

h) Theorem 17: (A+B)' = A'. B' (De Morgan's Theorem)

i) Theorem 18: (A.B)' = A'+ B' (,,)

6.5. De Morgan's Theorems

(i) The complement of a product is equal to the sum of the complements. $\overline{AB} = A'+B'$

(ii) The complement of a sum is equal to the product of the complements.

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

6.6. Boolean Expressions

Boolean expressions can be solved by two methods, which are as follows:

a) Using Boolean Laws/Theorems.

b) By Karnaugh Map (K-Map).

6.7. Solving Boolean Expressions Using Boolean Laws

Simplify:

AB + BC +B'C

=AB + C(B+B') (∴B+B' =1)

=AB+C

Simplify

A'.B + A.B + A'.B'

=B(A+A') +A'.B'

=B+A'.B'

=B+A' (∴ A+A'B = A+B)

Simplify:

A+A.B'+A'.B

=A(1+B')+A'.B

=A+A'. B

=A+B

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

$=\overline{(A'.\ B)'\ (C.\ D')'}$

=(A''+ B') (C'+D'')

=(A+B') (C'+D)

6.8 Min term (SOP)

A product term of all variables in either complimented or un-complimented form is called a min term.

A two variable function has four possible combinations, viz. AB, A'B, AB' and A'B'

6.9 Max term (POS)

A sum term of all variables in either complimented or un-complimented form is called a max tem.

A two variable function has four possible combinations viz A+B, A'+B, A+B'and A'+B'.

6.10 Karnaugh Map (K-Map)

It is used to simplify the Boolean expressions. Karnaugh map is made up of squares. Each square represents one min term. Map is designed as per the number of variables. A variable is represented by 1 and its complement is represented by 0.

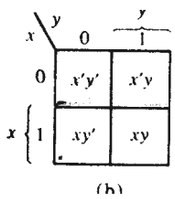
6.11 Two Variable Map

a) Write 1 in the corresponding squares of the minterms.

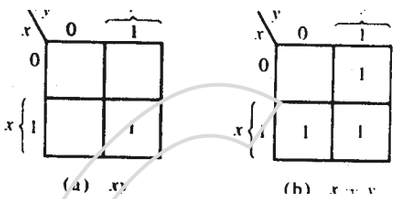
b) Try to make the pairs. If pair is complete then encircle it.

c) Take the common variable from the pairs, which is the solution of the expression.

There are four minterms for 2 variables; hence the map consists of four squares, one for each minterm.



Two-variable map



Representation of functions in the map

6.12 **Three Variable Map**

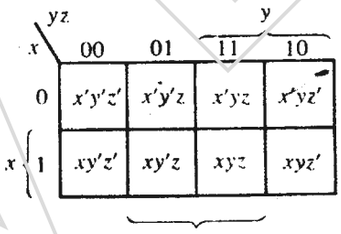
a) Write 1 in the corresponding squares of the minterms.

b) Try to make quads and pairs. Encircle the quads and pairs.

c) Take the common variables from the quads and pairs and add them.

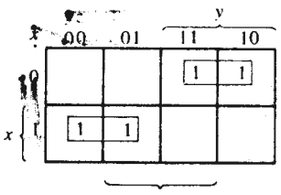
d) Map can be rolled for making quads and pairs.

There are 8 minterms for three variables. Therefore, a map consists of eight squares. Minterms are arranged as per minimum change code (Gray Code) i.e. only one bit changes in the next step.



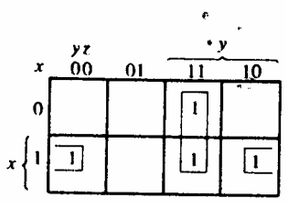
Three-variable map

Simplify: x'yz + x'yz' + xy'z' + xy'z



F = x'y + xy'

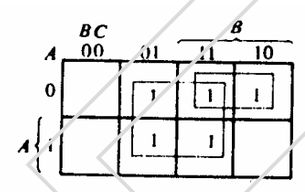
Simplify: F = x'yz + xy'z' + xyz + xxyz'



F = yz -- xz'

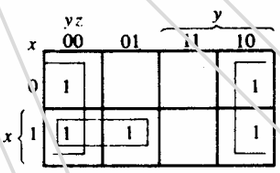
Simplify: F = A'C + A'B + AB'C + BC *is equal to*

IF = A'BC + A'B'C + A'BC + A'BC' + AB'C + ABC + AB'C



F = C + A'B

Simplify: x'y'z' + xy'z' + xy'z + x'yz' + xyz'



F = z' + xy'

6.13 **Four Variable Map**

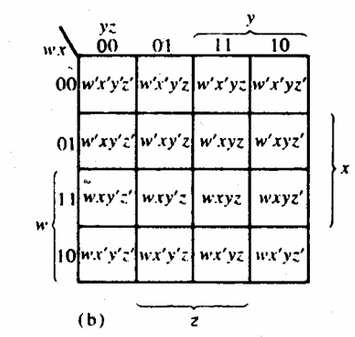
a) Write 1 in the corresponding squares of the minterms.

b) Try to make octets, quads and pairs. Encircle them.

c) Take the common variables from the octets, quads and pairs and add them.

d) Map can be rolled to make octets, quads and pairs.

It represents 16 minterms. Each minterm represent a square.



Important points:

(i) One square represents one minterm, giving a term of four literals.

(ii) Two adjacent squares represent a term of three literals.

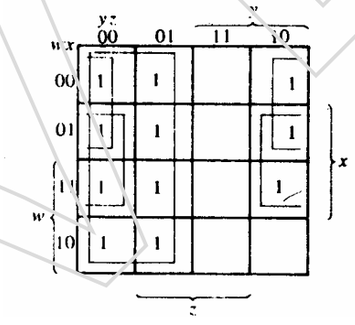
(iii) Four adjacent squares represent a term of two literals.

(iv) Eight adjacent squares represent a term of one literal.

Sixteen adjacent squares represent the function equal to 1.

Simplify.

F= w'x'y'z' + w'x'y'z + w'x'yz' + w'xy'z' + w'xy'z + w'xyz' + wxy'z' + wxy'z + wxyz' + wx'y'z' + wx'y'z

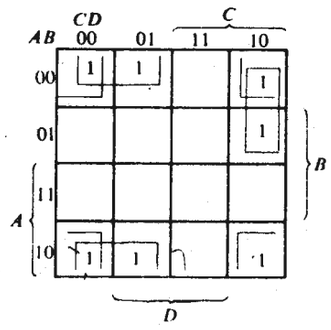


F = y' + w'z' + xz'

Simplify:

F = A'B'C' + B'CD' + A'BCD' + AB'C' *is equal to*

F = A'B'C'D + A'B'C'D' + AB'CD' + A'B'CD' + A'BCD' + AB'C'D + AB'C'D'



F = B'D' + B'C' + A'CD'

Exercise 6

Q. 1 Fill in the blanks

(i) Boolean expression can be solved by _____ and _____. (Boolean Laws, K-map)

(ii) A + 1 = _____ (1)

(iii) A + A' = _____ (1)

(iv) 4-variable map consists of _____ squares. (16)

(v) A quad represents a group of _____ (four 1s)

Q. 2 Write short notes

(i) Demorgan's theorems

(ii) Min term and Max term

(iii) Karnaugh map

Q. 3 Solve the following using K-map

(i) X'Y'Z' + XY'Z' + XY'Z + X'YZ' + XYZ'

(ii) A'B'C'D + A'B'C'D' + AB'CD' + A'B'CD' + A'BCD' + AB'C'D + AB'C'D'

CHAPTER 7
LOGIC GATES

7.1. Logic Gates

A logic gate is an electronic circuit, which makes logical decision. The common logic gates are OR, AND, NOT, NAND & NOR gates. The NAND and NOR gates are called universal gates because using NAND and NOR gates you can design any gate. OR, AND and NOT gates are called basic gates. The Exclusive-OR (X-OR) gate is another logic gate which can be constructed by using AND, OR and NOT gates.

Logic gates have two or more inputs and only one output except NOT Gate, which has only one input. The logic gates are the building blocks of hardware, which are available in the form of various ICs. Each Gate has its own logic symbol. The relationship between input and output variables of each Gate can be represented in a tabular form called a truth table.

7.2. OR Gate

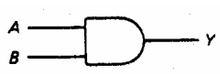
OR Gate performs logical addition. *OR gate has two or more inputs and only one output.* The output of an OR gate will be high when any of the inputs is HIGH (1). The output will be low when all the inputs are LOW (0). If A and B are the input variables of an OR gate and Y is its output, then Y=A+B.

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

7.3. AND Gate

AND gate performs logical multiplication known as AND function. *AND gate has two or more inputs and a single output*. The output of an AND gate will be high only when all the inputs are high. If any of the inputs is low, the output will be low.

If A and B are the input variables of an AND gate and Y is its output, then Y=A.B.



Inputs		Output
A	B	$Y = A \cdot B$
0	0	0
0	1	0
1	0	0
1	1	1

7.4. NOT Gate

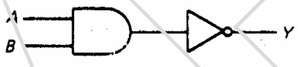
NOT gate performs logical function called complementation (inversion). This gate converts one logic level into the opposite logic level. It has one input and one output.

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

7.5. NAND Gate

It consists of an AND gate followed by a NOT gate. Its function is just opposite to that of an AND gate.

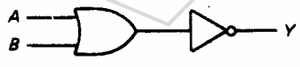
Inputs		Output
A	B	$Y = \overline{A \ B}$
0	0	1
0	1	1
1	0	1
1	1	0



7.6. NOR Gate

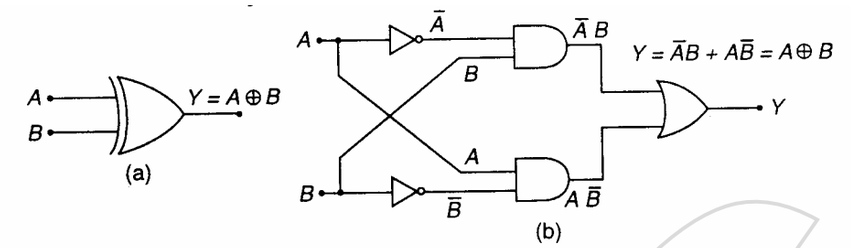
It consists of an OR gate followed by a NOT gate. Its function is just opposite to that of an OR gate.

Inputs		Output
A	B	$Y = \overline{A + B}$
0	0	1
0	1	0
1	0	0
1	1	0



7.7. Exclusive –OR Gate (EX-OR Gate)

It consists of two NOT gates, two AND gates and one OR gate. Its output will be high when inputs are different. Its output will be low when all inputs are similar.



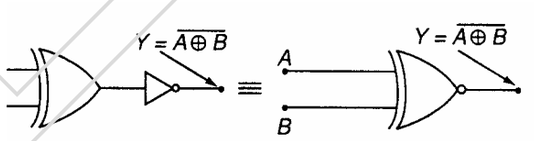
Inputs		Output
A	*B*	$Y = A \oplus B$
0	0	0
0	1	1
1	0	1
1	1	0

.

7.8. Exclusive –NOR Gate (EX-NOR Gate)

It consists of an Ex-OR gate followed by a NOT gate. Its output will be high when all inputs are similar. Its output will be low when inputs are different.

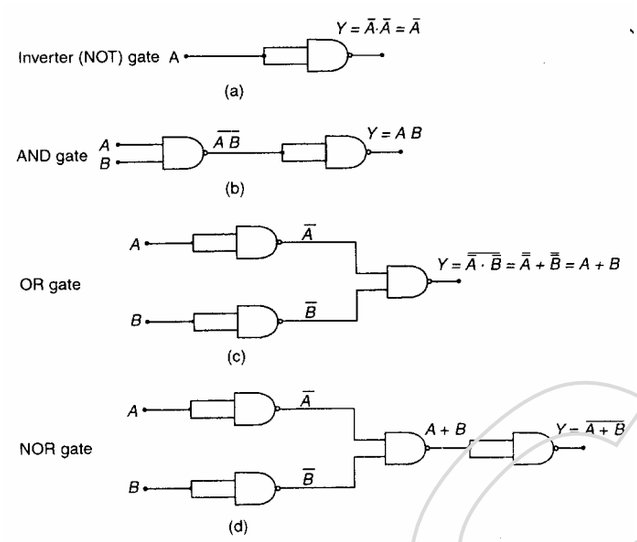
Inputs		Output
A	*B*	$Y = \overline{A \oplus B}$
0	0	1
0	1	0
1	0	0
1	1	1



7.9. Building Different Gates Using Universal Gates

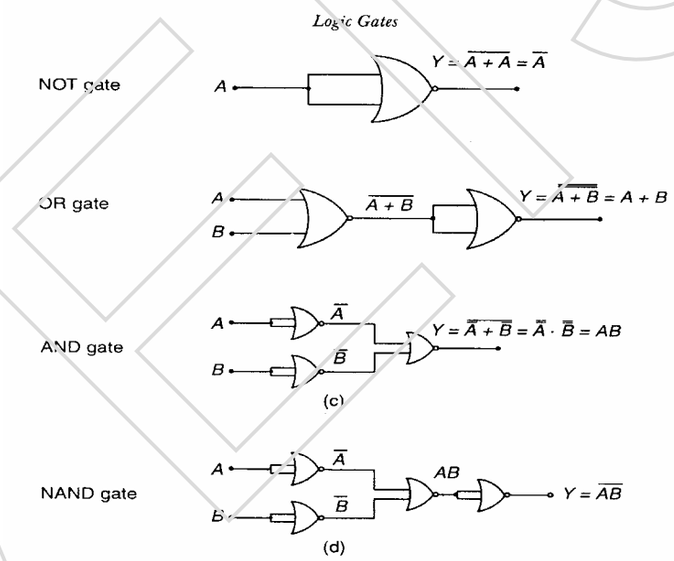
NAND and NOR gates are called universal gates because by using these gates we can make all gates like NOT, AND, OR and combination of these.

a) Using NAND Gate -



Building of NOT, AND, OR, NOR gates using NAND gate

b) Using NOR Gate



Building of NOT, AND, OR, NOR gates using NOR gate

Exercise 7

Q.1 Fill in the blanks

(i) Basic gates are _____ (OR, AND, NOT)
(ii) Universal gates are _____ (NOR, NAND)
(iii) _____ gate has only one input and one output.(NOT)
(iv) Output of a OR gate is high when _____. (any of the input is high).
(v) Output of an AND gate low when _____. (any of the input is low)
(vi) NOT gate performs the _____ function. (complementation)
(vii) NOR and NAND gates are called universal gates because _____ (all gates can be designed by using these gates)

Q. 2 Write short notes

(i) NOT Gate
(ii) Universal gates
(iii) NAND gate
(iv) NOR gate
(v) Ex-OR gate

Q.3 Build AND, OR, and NOT gates using NAND gates.

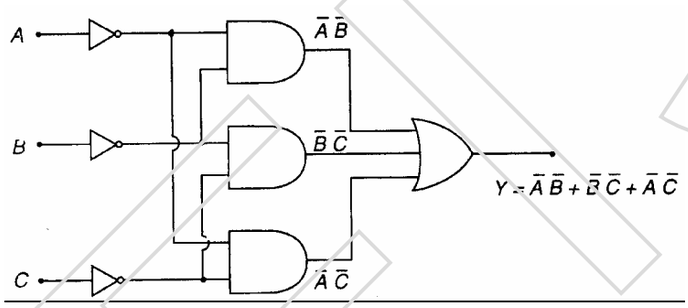
CHAPTER 8
DESIGNING OF LOGIC CIRCUITS

8.1. **Introduction**

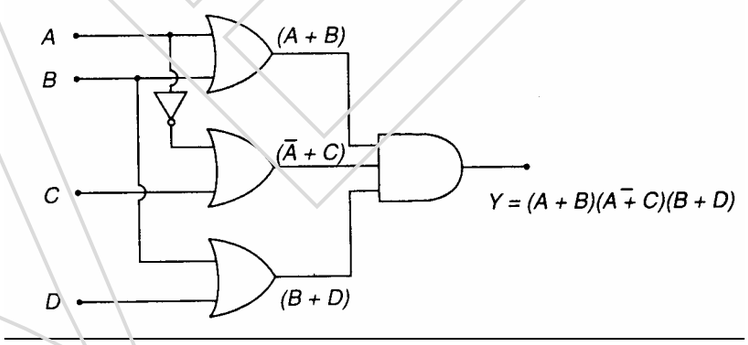
A logic circuit is constructed by using different logic gates. Logic circuits are designed as per the Boolean expressions. Depending upon the number of variables and logical operations used in the Boolean expression, the type and number of gates is decided. Before designing a logic circuit a Boolean expression should first be simplified so as to reduce the complexity of the circuit and the number of gates used.

Examples:

Realise the logic expression Y = A'B' + B'C'+A'C'



Realise the logic expression Y = (A+B) (A'+C) (B+D)



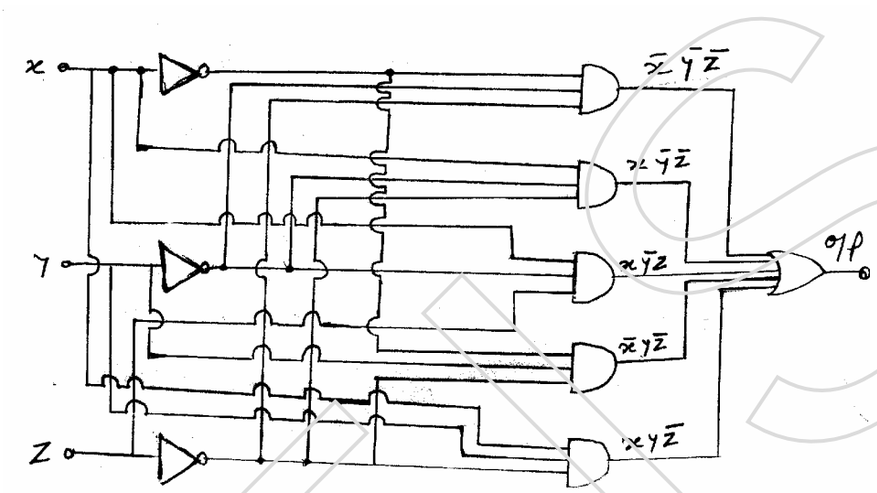
8.2. **Need of Simplification**

Boolean expressions are simplified to reduce the hardware requirement.

Given Boolean expression is

x'y'z' + xy'z' + xy'z + x'yz' + xyz'

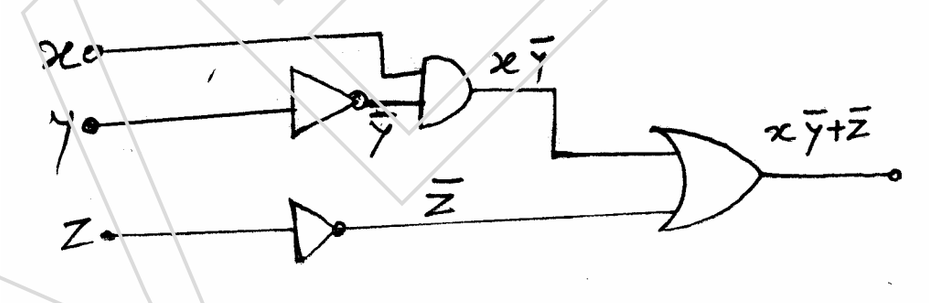
Its logic circuit without simplification will be as drawn below:



By simplifying this Boolean expression we get

z' + xy'

So logic circuit for this will be as shown below:



Without simplification number of gates used were ----- AND – 5, NOT – 3, OR – 1. After simplification number of gates have been reduced to --- AND – 1, NOT – 2, OR –1.

CHAPTER 9
ARITHMETIC CIRCUITS

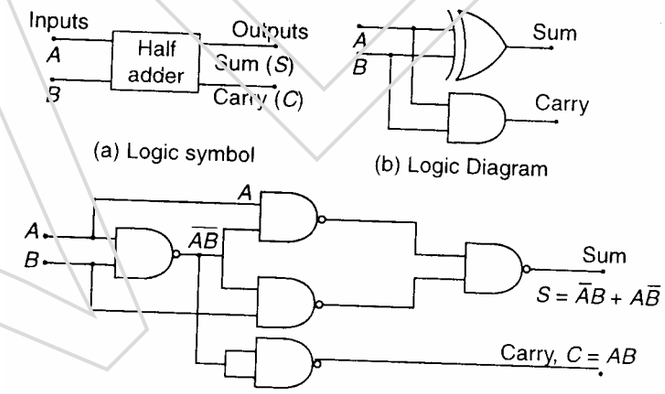
9.1. Arithmetic Circuits

Digital computers and calculators consist of arithmetic and logic circuits, which contain logic gates that add, subtract, multiply and divide the binary numbers.

9.2. Half – Adder

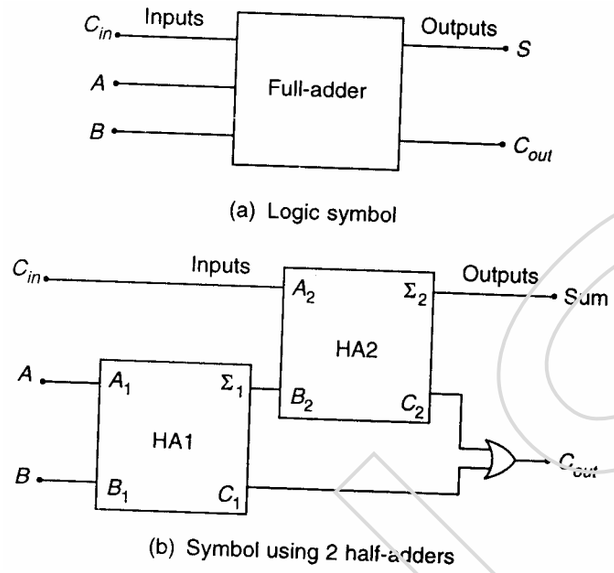
The combinational circuit, which performs the arithmetic addition of two binary bits, is called half-adder. Half adder has two inputs and two outputs. *In the truth table, the sum output is 1 when AB =01 and AB =10, therefore the expression for sum is S = A'B+ AB' and carry is 1 when A=1, B=1, means AB=1 therefore, C=AB* . It consists of an Ex-OR gate and an AND gate.

Inputs		Outputs	
Augend *A*	Addend *B*	Sum *S*	Carry *C*
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1



9.3. **Full – Adder**

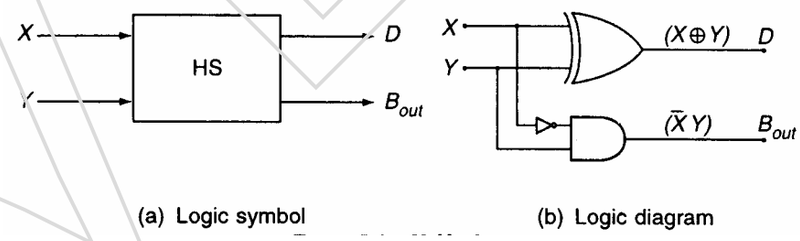
A full-adder is a combinational circuit that performs the arithmetic sum of three input (two variables and a carry) bits and produces a sum (S) output and a carry (C) output. Ii consists of two half – adders and an OR gate.



9.4. **Half Subtractor**

The half subtractor is a combinational circuit, which is used to perform subtraction of two bits. It has two inputs, (X, Y) and two outputs (difference & borrow). *It is clear from the truth table that the difference (D) output is 0, if X = Y and 1 if X # Y. The borrow (B) output is 1 whenever X<Y. If X is less than Y, then subtraction is done by borrowing 1 from the next higher bit.*

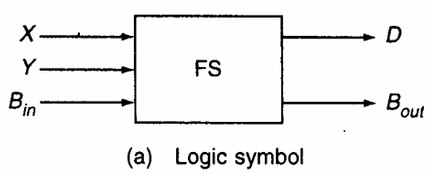
It consists of an Ex-OR gate, a NOT gate and an AND gate.



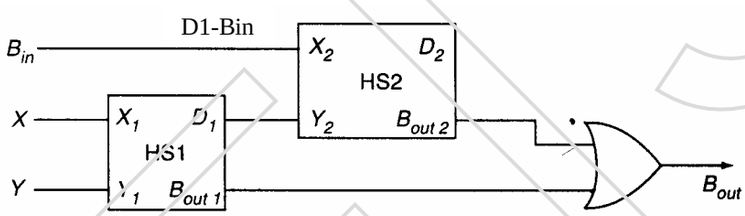
Inputs		Outputs	
Minuend X	Subtrahend Y	Difference D	Borrow B_{out}
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

9.5. **Full Subtractor**

It has three inputs (two variables and a borrow) and two outputs. It consists of two half subtractors and an OR gate.



(a) Logic symbol



Inputs			Outputs	
Minuend bit X	Subtrahend bit Y	Borrow in B_{in}	Difference D	Borrow out B_{out}
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

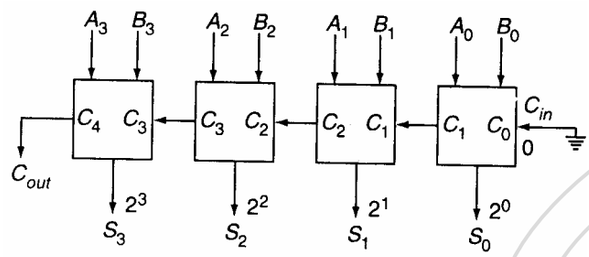
X-Y = X2

Bin = Y2

D = X2-Y2

9.6. **Parallel Binary Adder**

In the least significant stage, A_0, B_0 and C_0 are added resulting in sum S_0 and carry C_1. This carry C_1 becomes the carry input to the second (next) stage. Similarly in the second stage A_1, B_1 and C_1 are added resulting in sum S_1 and carry C_2 and the similar process continues till the last stage.



Example: 1111 +1000

Exercise 9

Q. 1 Fill in the blanks

(i) A half adder performs the arithmetic addition of _____ binary bits. (two)

(ii) Outputs of an adder are called _____ and _____ (sum, carry)

(iii) Number of outputs of a half adder are _____ (two)

(iv) Number of outputs of a half subtractor are _____ (two)

(v) Outputs of a subtractor are called _____ and _____. (difference, borrow)

Q.2 Write short notes

(i) Half adder

(ii) Full adder

(iii) Half subtractor

(iv) Parallel binary adder

CHAPTER 10
DATA PROCESSING CIRCUITS

10.1. **Introduction**

There are number of data processing circuits used in digital circuits. A few of them are as follows:

a) Multiplexers

b) Demultiplexers

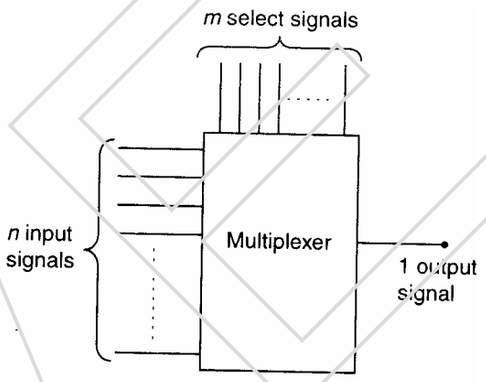
c) Decoders

d) Encoders

e) Parity Checkers

10.2. **Multiplexer (MUX)**

The term ‘multiplex’ means “ many into one”. Multiplexing is the process of transmitting a large number of information over a single line.

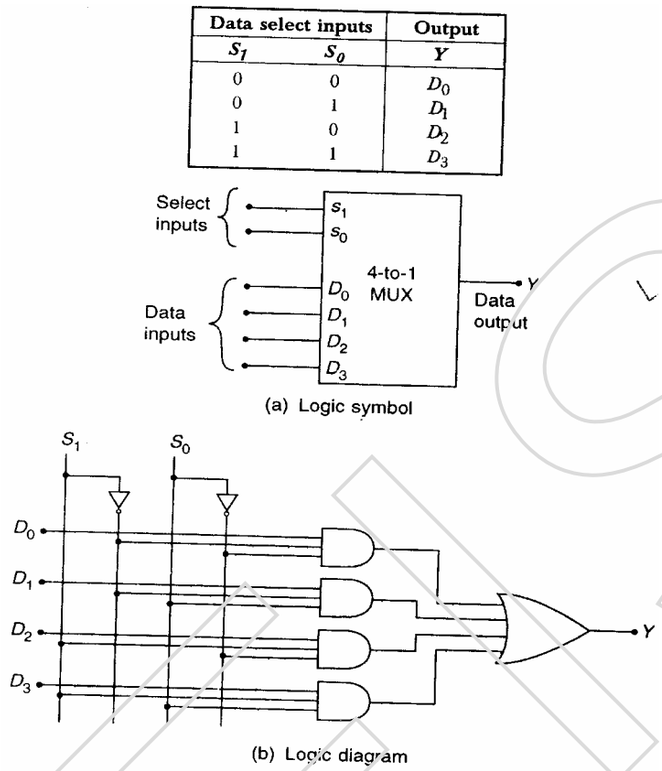


The digital multiplexer (mux) has several (many) data input lines and a single output line. The selection of a particular input line is controlled by a set of selection lines. The selection lines decide the number of input lines of a particular multiplexer. For example, to select 1 out of 4 input lines,

two select lines are required, to select 1 out of 8 input lines, three select lines are required and so on.

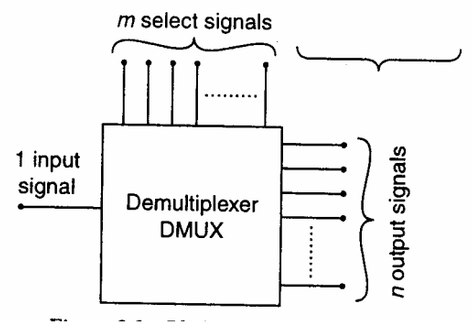
10.3. 4 to 1 MUX

Data select inputs		Output
S_1	S_0	Y
0	0	D_0
0	1	D_1
1	0	D_2
1	1	D_3



A 4 to 1 multiplexer has four inputs (D0 to D3), one output (Y) and two select inputs (S0 and S1). Data input D0 is connected to the output Y. When S1=0 and S0=0 and data input D1 is connected to the output Y. When S1=0 and S0=1 and so on. Four input data lines are connected to the different AND gates to which select lines S1 and S0 are also connected. But select lines S0 and S1 enable only one gate at a time, so only one input data line appears at the output.

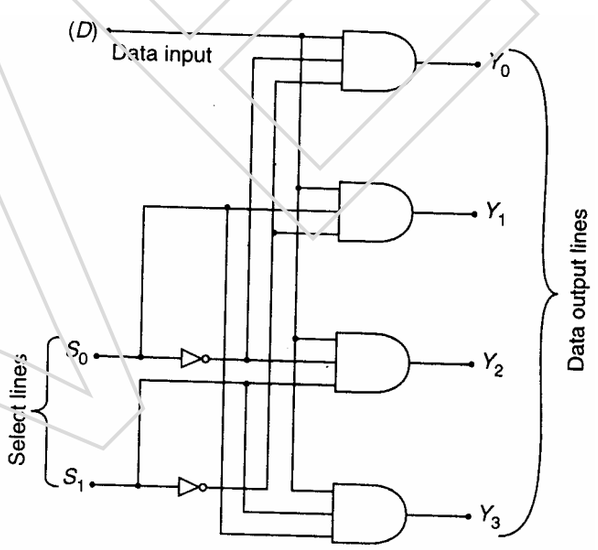
10.4. **Demultiplexers (DMUX)**



The word " demultiplex" means "one into many". De-multiplexing is the process of taking information from one input and transmitting the same over one of the several (many) outputs. The circuit has one input signal, m select signals and n output signals. The select inputs determine to which output the input data will be connected.

10.5. **1 to 4 Demultiplexer**

Data input	Select inputs		Outputs			
D	S_1	S_0	Y_3	Y_2	Y_1	Y_0
D	0	0	0	0	0	D
D	0	1	0	0	D	0
D	1	0	0	D	0	0
D	1	1	D	0	0	0



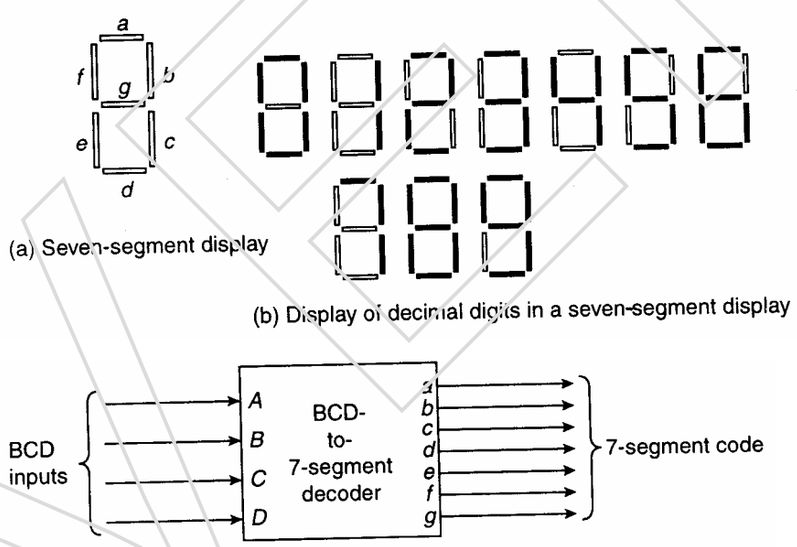
A 1-to-4 demultiplexer has a single input (D), four outputs (Y0 to Y3) and two select inputs (S0 and S1). Data input is connected to output Y0 when S1=0 and S0=0 and the data input is connected to output Y1 when S1=0 and S0=1 and so on.

Input data line is connected to all the AND gates. Two select lines S1 and S0 enable only one gate at a time and the data that appears on the input line passes through the selected gate to the associated output line.

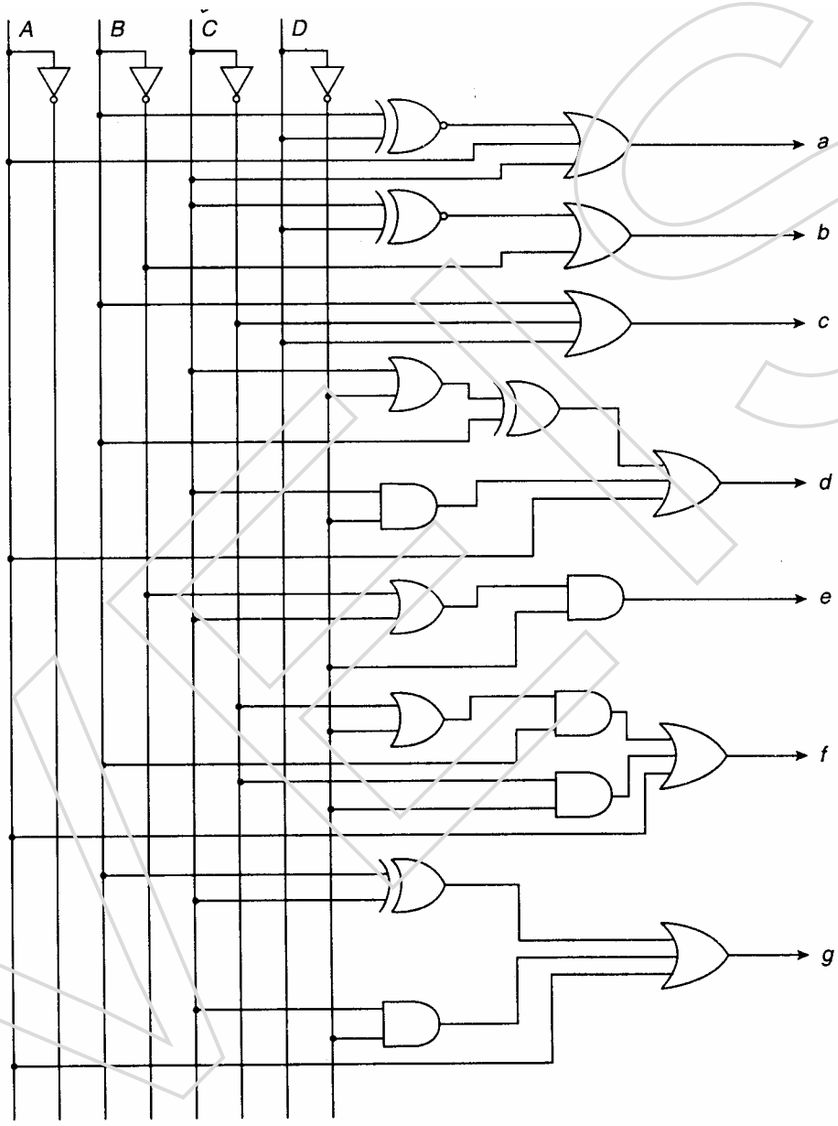
10.6 **Decoders**

A decoder is a logic circuit that converts an n-bit binary input code (data) into 2^n output lines, such that each output line will be activated for only one of the possible combination of inputs. In decoder, number of outputs is greater than number of inputs.

10.7. **BCD to 7-Segment Decoder**



BCD inputs				Seven segment outputs						
A	*B*	*C*	*D*	*a*	*b*	*c*	*d*	*e*	*f*	*g*
0	0	0	0	1	1	1	1	1	1	0
0	0	0	1	0	1	1	0	0	0	0
0	0	1	0	1	1	0	1	1	0	1
0	0	1	1	1	1	1	1	0	0	1
0	1	0	0	0	1	1	0	0	1	1
0	1	0	1	1	0	1	1	0	1	1
0	1	1	0	1	0	1	1	1	1	1
0	1	1	1	1	1	1	0	0	0	0
1	0	0	0	1	1	1	1	1	1	1
1	0	0	1	1	1	1	1	0	1	1



A seven-segment display is used for displaying any one of the decimal digits, 0 to 9. A BCD-to-7-segment decoder accepts a decimal digit in BCD code and generates the corresponding 7-segment code.

A seven–segment display is composed of seven elements or segments. Each segment is made up of a material that emits light when current is passed through it. Most commonly used displays are LEDs.

10.8 Encoders

An encoder is a digital circuit that performs the inverse operation of a decoder. Hence, the opposite of the decoding process is called encoding. In an encoder, the number of output are less than the number of inputs.

Figure 6.24 BCD-to-7-segment decoder driving a LCD

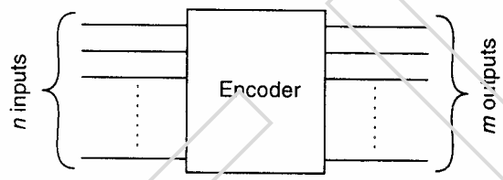
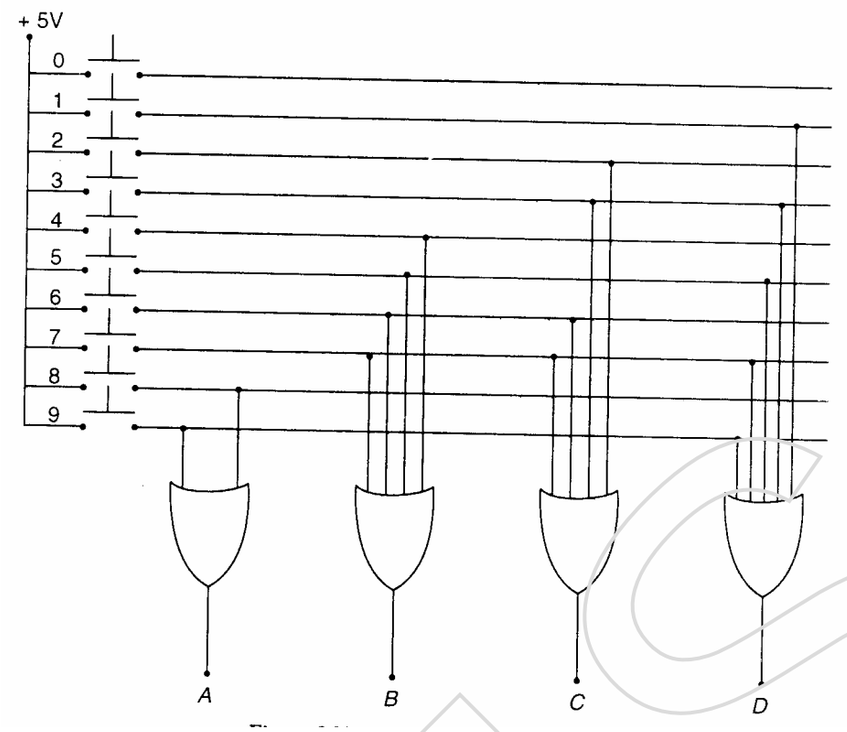


Figure 6.25 Block diagram of an encoder

10.9 Decimal to BCD Encoder

It has 10 inputs corresponding to ten decimal digits (0 to 9) and four outputs (A, B, C, D) representing the BCD value of input decimal digit.

Decimal inputs										BCD outputs			
0	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*A*	*B*	*C*	*D*
1	0	0	0	0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0	0	0	0	1
0	0	1	0	0	0	0	0	0	0	0	0	1	0
0	0	0	1	0	0	0	0	0	0	0	0	1	1
0	0	0	0	1	0	0	0	0	0	0	1	0	0
0	0	0	0	0	1	0	0	0	0	0	1	0	1
0	0	0	0	0	0	1	0	0	0	0	1	1	0
0	0	0	0	0	0	0	1	0	0	0	1	1	1
0	0	0	0	0	0	0	0	1	0	1	0	0	0
0	0	0	0	0	0	0	0	0	1	1	0	0	1



10.10 **Parity**

When digital data is transmitted from one location to another, it is necessary to know at the receiving end whether the received data is free of error.

Error detection is achieved by adding extra bit to the transmitted signal. The additional bit is known as parity bit. There are two types of parity namely even parity and odd parity.

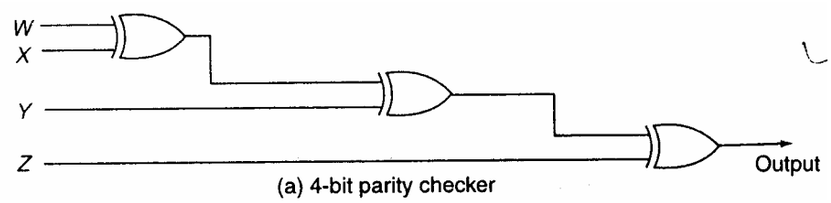
In an even parity, the parity bit is added to the word to make the number of 1s even in the modified word.

Odd parity means number of 1s is odd in the modified word.

10.11. **Parity Checkers**

Parity checker can be designed by using Exclusive–OR gate. It is used to detect errors in storage media such as magnetic tapes, paper tapes etc.

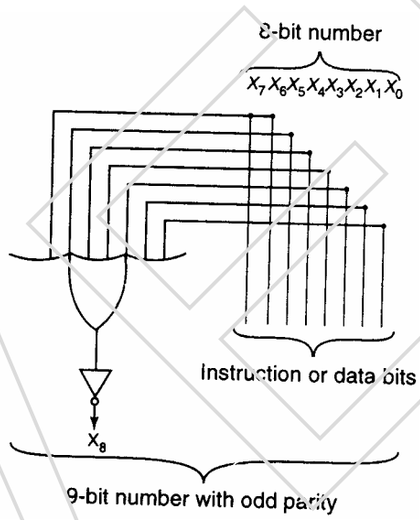
Ex-OR gate produce an output when the input has an odd number of 1s. Therefore, an even-parity input to an Exclusive-OR gate produces a low output, while an odd-parity input produces a high output.



(a) 4-bit parity checker

10.12. Odd Parity Generator

Extra bit can be generated using an Ex-NOR gate. Suppose the 8-bit number is 010000010. This number has a even parity, it means when applied to an Ex-OR gate, it will produce an output of 0, then NOT gate will convert this 0 to 1 and the final 9-bit output is 101000001. Now it has odd parity.



Exercise – 10

Q. 1 Fill in the blanks

(i) "Multiplex" means _____. (many into one)
(ii) "Demultiplex" means _____. (one into many)
(iii) Number of outputs of a multiplexer is _____. (one)
(iv) Number of inputs of a demultiplexer is _____. (one)
(v) Types of parity are _____ and _____ (odd, even)

Q. 2 Write short notes

(i) Multiplexer
(ii) Demultiplexer
(iii) Decoder
(iv) Encoder
(v) Parity checker

CHAPTER 11
FLIP FLOPS

11.1 Introduction

The logic circuits whose outputs at any instant depend on the input signals present at that time are known as combinational circuits. Combinational circuits do not have capacity to retain the information. The logic circuits whose output depends upon present inputs as well as previous output are called sequential circuits. In sequential circuits output signals are fed back to the input side.

11.2 Sequential Circuits

Sequential circuits are of two types. They are as follows:

(i) Synchronous or Clocked

(ii) Asynchronous or Un-clocked

In synchronous sequential circuits, a timing device called a master–clock generator that generates pulses continuously, achieves synchronization. An asynchronous sequential circuit is a memory cell that has only two states, which can be either 1 or 0. Such two state sequential circuit is called flip-flop.

11.3. Flip – Flop

Flip-flop has one or more inputs and two outputs Q and Q'. The two outputs are complementary to each other. When the flip-flop output Q is 0 or 1, it will remain in that stable state until one or more inputs are given to change the output. The flip-flop output will remain set/reset until the

trigger pulse is given to change the state. It can be used as a memory device to store one binary bit.

Flip- flops are classified on the basis of the way their inputs and clock pulses cause pulses cause transition (Change) between two states. Major types of F/Fs are as follows:

(i) S-R flip- flop

(ii) Clocked S-R flip- flop

(iii) J-K flip-f lop

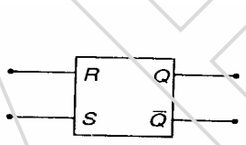
(iv) D flip- flop

(v) T flip- flop

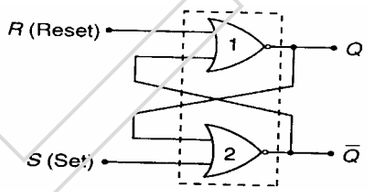
(vi) Master/Slave J- K flip- flop

11.4. S- R Flip_ Flop (Set- Reset F/F)

The SR flip–flop has two inputs namely SET (S) and RESET (R), and two outputs Q and Q'. The two outputs are complement to each other. The SR flip–flop can be designed using NOR gates or NAND gates. As we know the output of a NOR gate is 0 if any input is 1, and output is 1 when all inputs are 0. Assume Q=0 and Q'=1 at the initial state.



(a) Block diagram



Inputs		Output	Action
S	R	Q	
0	0	Last State	No Change
0	1	0	Reset
1	0	1	Set
1	1	?	Forbidden

]

CASE 1:

For S=0 and R=0, the flip–flop remains in its present state. The inputs of NOR gate 1 are 0 and 1 (because Q' is 1), thus its output is 0. This output is fed to the NOR gate 2, therefore inputs of NOR gate 2 are 0 and 0, thus its output is 1. So, the condition S=0 and R=0 will not affect the outputs of flip-flop.

CASE 2:

For S=0 and R=1, the inputs of NOR gate 1 are 1 (as R=1) and 1 (as Q'=1), thus its output Q is 0. The inputs of NOR gate 2 are 0 (as S=0) and 0 (as Q=0), so its output is 1 (Q'=1). Therefore, the input condition S=0 & R=1 always reset the flip- flop.

CASE 3:

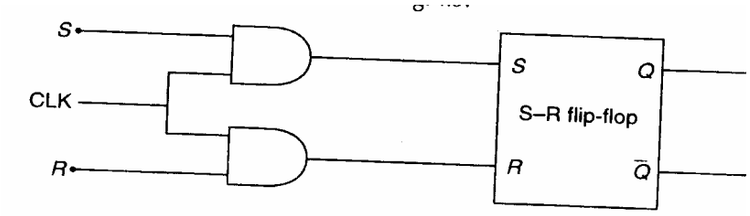
For S=1 and R=0, the inputs of NOR gate 2 are 1 (as S=1) and 0 (Q=0) so its output is 0 (Q'=0). Now the inputs of NOR gate 1 are 0 and 0 (:. R = 0 Q'=0), thus output is 1(Q=1). Therefore, the input condition S=1 and R=0 will always set the flip-flop.

CASE 4:

For S=1 and R=1. This condition will produce 0 at the output of both the NOR gates. This condition violates the fact that Q and Q' are the complements of each other. In normal, this condition must be avoided.

Limitation – Forbidden condition for input S=1, R=1

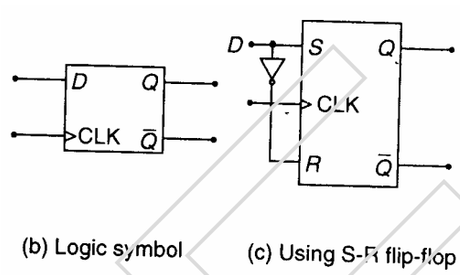
11.5. **Clocked S- R Flip- Flop**



Clocked S- R F/F is used to deliver output when clock pulse strikes.

In this circuit when the clock input is low, the output of both the AND gates is low, therefore the changes in S and R inputs will not affect the output (Q) of the flip-flop. When the clock input becomes high, the value at S and R inputs will be passed to the output of the AND gates and output (Q) of the flip-flop will change according to the changes in S and R inputs.

11.6. **D Flip- Flop**



(b) Logic symbol (c) Using S-R flip-flop

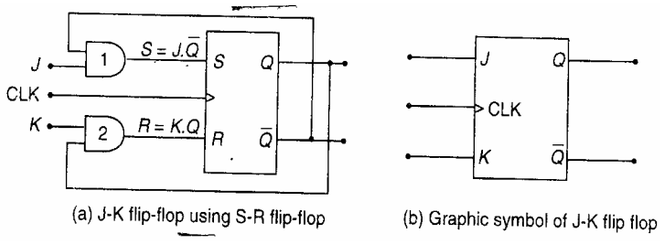
CLK	Input D	Output Q_{n+1}
1	0	0
1	1	1
0	×	No change

D F/F was developed to avoid forbidden condition because it has only one I/P.

The D (delay) flip-flop has only one input and two outputs Q and Q'. D-flip-flop is an SR flip-flop designed by inserting a NOT gate between S and R.

When D is 0 then S=0 and R=1 and if D is 1 then S=1 and R=0. It will work like a clocked SR flip-flop.

11.7. J- K Flip- Flop



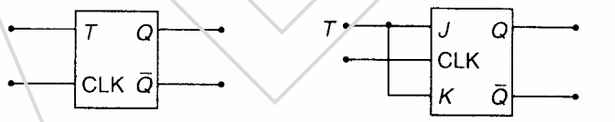
(a) J-K flip-flop using S-R flip-flop (b) Graphic symbol of J-K flip flop

It is similar to SR flip-flop but when J=K=1, the flip-flop output toggles i.e. changes to its complement state, if Q=0, it changes Q=1 and vice versa. Thus it also does not has forbidden condition.

When J=1, K=1 and previous state is a set state (i.e. Q=1, Q'=0), then S=JQ' = 1x0=0 and R =KQ =1x1=1. Since S=0 and R =1, the flip–flop RESETS on the application of clock pulse, i.e. the flip-flop toggles from SET to RESET state. When J=1. K=1 and if the previous state is RESET state (i.e. Q=0, Q' =1), then S =JQ'= 1x1=1 and R = KQ =1x0=0. Since S=1 and R=0, the flip-flop sets. So, when J=1 and K=1, the flip-flop toggles on the application of the next clock pulse.

11.8. T Flip- Flop

T or trigger or toggle flip-flop has only a single data (T) input, a clock input and two outputs Q and Q'. The T-type flip-flop is obtained from a J-K flip-flop by connecting its J and K inputs together.



(a) Block diagram of T flip-flop (b) T flip-flop using a J-K flip-flop

Q_n	T	Q_{n+1}
0	0	0
0	1	1
1	0	1
1	1	0

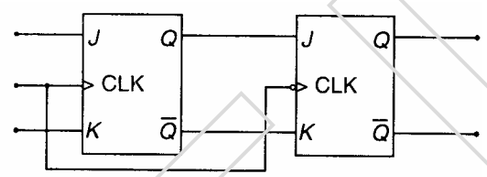
When the T input is in the 0 state (i.e. J=K=0) prior to a clock pulse the Q output will not change with clocking. When the T input is at a 1 state (i.e. J=K=1) prior to clocking, the output will change its state after clocking. If the input is 1 and device is clocked, the output will change state regardless of what output was prior to clocking. This is called toggling.

RACING: Toggling more than once during a clock cycle is called racing.

Limitation – It has racing problem.

11.9. Master- Slave J-K Flip- Flop

Master slave J–K flip-flop is the way to avoid the racing. In this, the first flip-flop is the master and the second is the slave. The master is positive edge triggered and the slave is negative edge triggered.



When the clock input has a positive edge, the master acts according to its J-K inputs, but the slave does not respond, since it requires a negative edge at the clock input. When the clock input has a negative edge, the slave flip-flop copies the master outputs but master does not respond, since it requires a positive edge at its clock input.

Exercise – 11

Q. 1 Fill in the blanks

(i) A device / circuit capable of storing one binary bit is called _____. (flip-flop)

(ii) A flip-flop has _____inputs. (one or more)

(iii) A flip-flop has _____ outputs.

(iv) Toggling occurs in _____ flip-flop when both the inputs are 1. (J_K)

(v) Forbidden condition occurs in _____ flip-flops when both the inputs are 1.

Q. 2 Write short notes

(i) Sequential circuits
(ii) Flip-flops
(iii) Master slave flip-flop
(iv) S-R flip-flop
(v) J-K flip-flop

CHAPTER 12

REGISTERS AND COUNTERS

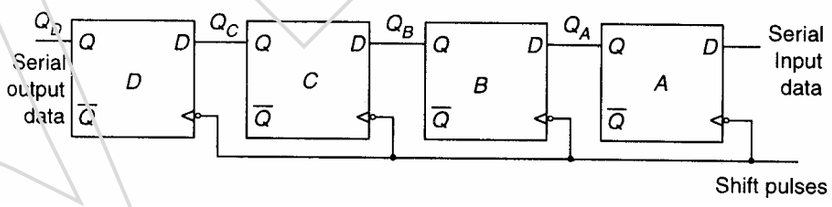
12.1. **Registers**

A register is a group of flip –flops suitable for storing binary information. Each flip-flop is a binary cell capable of storing one bit of information. An n-bit register consists of n flip-flops, which is capable of storing any binary information containing n bits. The register is mainly used for storing and shifting binary data entered into it from an external source.

12.2. **Shift Registers**

A shift register moves the stored bit left or right. This bit shifting is essential for certain arithmetic and logic operations used in microcomputers.

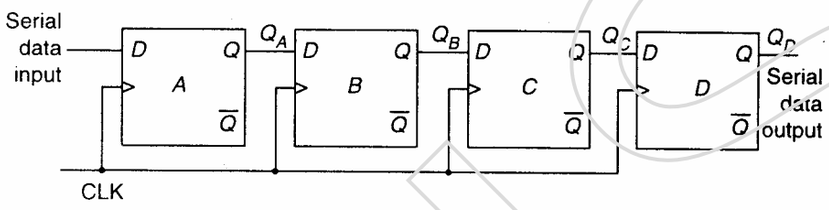
a) Shift Left Register



Input serial data sets up the right flip–flop A, Q_A sets up the second flip-flop B and so on. When the positive clock strikes the stored bit move one

position to the left. For example, when 4-Bit binary number 0001 is to be entered into the register before starting the shift operation. When input serial data is 1 and Q_A,Q_B,Q_C, Q_D = 0000. During first clock the right flip-flop sets up and the stored word becomes 0001. Now Q_A is 1 & serial input data is 0. When the next positive clock hits the B flip-flop, the register contents become 0010; the third positive clock edge results in 0100 and the fourth clock results in 1000.

b) Shift Right Register



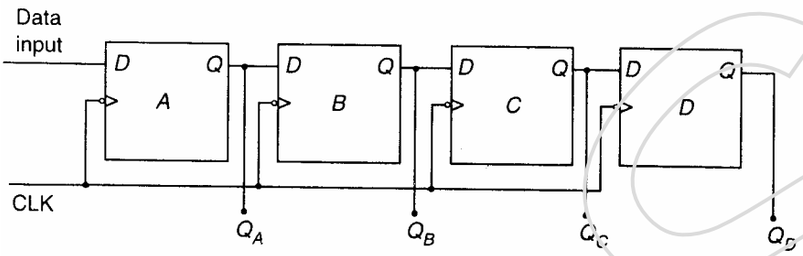
For example, when a 4-bit binary number 1101 is to be entered into the register. When 1 is applied at the serial data input, at the input of flip-flop A and clock is applied, then flip-flop A is set thus storing 1.

Next, 0 is applied to the serial input of flip-flop A and D=1 for flip-flop B because D input of flip-flop B is connected to the output of flip-flop A.

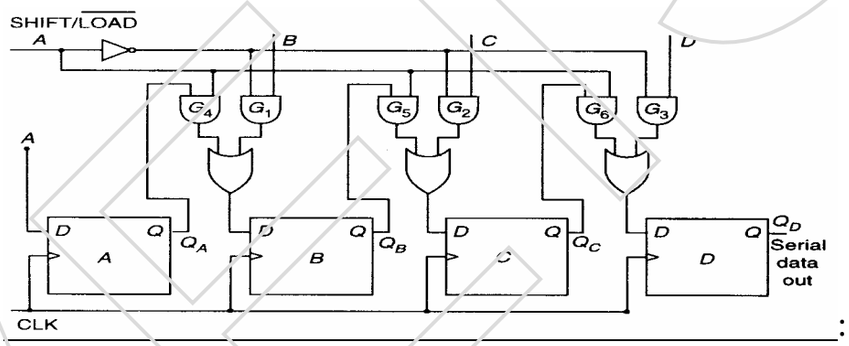
When the second clock pulse is applied, the 0 on the data input is shifted to the flip-flop A and the 1 stored in flip-flop 'A' is shifted to flip-flop B. Then next 1 is applied at the serial input line, and a clock pulse is applied. This 1 is entered into the flip-flop A, and the 0 stored in flip-flop A is shifted to flip-flop B and the 1 stored in flip-flop B is shifted to flip-flop C. Now apply the last binary bit i.e. 1 at the serial input and apply a clock pulse. Now, this 1 is entered into the flip-flop A, and the 1 stored in flip-flop A is shifted to flip-flop B, the 0 stored in flip-flop B is shifted to flip-

flop C, and 1 stored in flip-flop C is shifted to flip-flop D. This completes the serial entry of the 4-bit binary number into the shift resister.

12.3. Serial –In Parallel –Out Shift Register



12.4. Parallel –In Serial –Out Shift Register



12.5. Counters

A counter consists of a set of flip-flops. It is used to count the sequence of input pulses.

There are three types of counters, which are as follows:

a) Asynchronous and Synchronous Counters

b) Single and Multi Mode Counters

c) Modulus Counters

12.6. Asynchronous and Synchronous Counters

In an asynchronous counter, each F/F is triggered by the output from the previous F/F. The setting time of asynchronous counters is the cumulative (collective) sum of the individual setting times of F/Fs. It is also called as Serial Counter.

In synchronous counters, the clock pulses are simultaneously applied to all the F/Fs. Hence synchronous counters are also called Parallel Counters.

12.7. Single Mode and Multi Mode

Single mode counters operate in in a single mode ie. Up or Down mode. Multi mode counters operate in both in Up and Down modes.

12.8. Modulus Counter

Modulus counters are based on the ability to count the number of states.

For example, MOD-10 counter has 10 states.

EXERCISE – 12

Q. 1 Fill in the blanks

(i) _____ is a group of flip-flops capable of storing binary information. (Register)

(ii) _____ is used to count the sequence of pulses. (Counter)

(iii) In _____ counters, the clock pulses are simultaneously applied to all the flip-flops. (Synchronous)

Q. 2 Short notes

(i) Registers

(ii) Counters

(iii) Asynchronous counters

(iv) Synchronous counters

Q. 3 Explain shift registers with the help of diagram.

CHAPTER 13
LOGIC FAMILIES

13.1. Logic Family

A logic family is a specific class of logic circuits that are manufactured using that same manufacturing technology. Logic gates are fabricated as integrated circuits etc.

Depending upon the type of Semiconductor devices, logic families are classified as follows:

(i) Bipolar (ii) Unipolar

13.2. Bipolar

Bipolar means having two types of charge carriers i.e. electrons and holes. ICs, which are manufactured by using bipolar technology, are called bipolar logic families. These are:

(i) Resistor–Transistor Logic (RTL)

(ii) Diode–Transistor Logic (DTL)

(iii) Transistor-Transistor Logic (TTL)

13.3. Unipolar

ICs manufactured by using metal oxide semiconductors (MOS) i.e. which have only one type of charge carriers electrons or holes are called unipolar logic families. These are:

(i) CMOS (Complementary MOSFET): It uses P and N channel devices. It has the greatest complexity and lowest packing density among the MOS families. It has low power dissipation and very high input impedance. The CMOS logic gates are used

in battery operated portable equipment. Its disadvantage is low speed due to high input impedance.

(ii) PMOS (P-Channel MOSFET): It uses only p-channel MOSFET. Holes are the current carriers for PMOS.

(iii) NMOS (N-channel MOSFET): It uses n-channel MOSFET. Free electrons are the current carriers in NMOS.

13.4. Scale of Integration of ICs

(i) SSI (Small Scale Integration): It includes upto 10 gates on the same chip.

(ii) MSI (Medium Scale Integration): It includes 11 to 100 gates per chip.

(iii) LSI (Large Scale Integration): It includes 101 to 1000 gates per chip.

(iv) VLSI (Very Large Scale Integration): It includes 1001 to 10000 gates per chip.

(v) SLSI (Super Large Scale Integration) : It includes 10001 to 100000 gates per chip.

13.5. Classification of ICs

ICs have been mainly classified into the following two classes:

(i) Linear IC: It operates with continuously varying signals. For example, ICs used in amplifier circuits.

(ii) Digital ICs: It operates with binary signals.

13.6. IC Packages

Following are the popular IC packages:

(i) Dual–In-Line Package (DIP)

(ii) Leadless Chip Carrier (LCC)

(iii) Pin Grid Array (PGA)

(iv) Single Edge Cartridge (SEC)

13.7. **Characteristics of Digital ICs -**

(i) Speed of Operation: It operates with high speed. Speed is defined as the time taken for the output of a gate to change after the input has changed ie. called propagation delay.

(ii) Power Dissipation: It is measure of power consumed by the logic gate when fully driven by all its inputs. It is expressed in miliwatts and microwatts.

(iii) Fan–In: The fan-in of a gate is the number of inputs connected to the gate.

(iv) Fan-Out: The fan–out represents the maximum number O/Ps of a gate. logic gates (One logic gate has one output only).

(v) Operating Temperature: All IC gates are semiconductor devices. These are temperature–sensitive by nature. The operating temperature ranges from 0^0 C to 70^0 C (for military -55^0 C to 125^0 C).

Exercise 13

Q. 1 Fill in the blanks

(i) Depending upon the type of semiconductor devices, logic families are classified as _____ and _____ logic families. (unipolar, bipolar)

(ii) Unipolar logic families are _____. (CMOS, PMOS, NMOS)

(iii) CMOS stands for _____. (Complementary MOSFET)

(iv) DIP stands for _____. (Dual In line Package)

(v) PGA stands for _____. (Pin Grid Array).

Q. 2 Write short notes

(i) Scale of Integration of Ics.

(ii) IC Packages

(iii) Classification of Ics.

CHAPTER 14 (only for general discussion)

D/A AND A/D CONVERSION

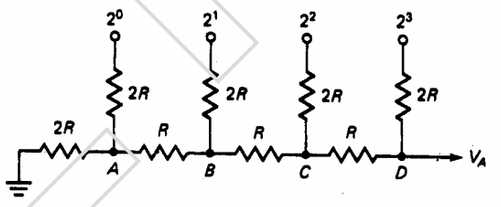
1. Introduction

Digital-to-Analog (D/A) and Analog-to-Digital (A/D) conversion form two very important aspects of digital data processing. A D/A converter (DAC) is sometime considered a decoding device. An A/D converter (ADC) is often referred to as an encoding device.

Digital to analog conversion is a straightforward conversion process and is easier than A/D conversion. In fact, a D/A converter is usually an integral part of any A/D converter.

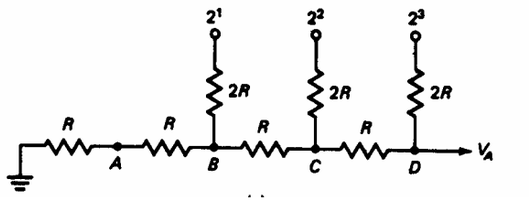
2. Binary Ladder

The binary ladder is a resistive network whose output voltage is a properly weighted sum of the digital input. Figure shows 4-bits binary ladder.



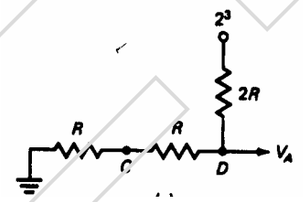
It is made of resistors that have only two values. The left end of the ladder is terminated with a resistance of 2R, and at the moment let us considers that the right end of the ladder (the output) is open circuited.

All the digital inputs are at ground. Beginning at node A, the total resistance looking into the terminating resistor is 2R. The total resistance looking out toward the 2^0 inputs is also 2R. These two resistors can be combined to form an equivalent resistor of value R as shown in figure.



Now, moving to node B, we see that the total resistance looking into the branch toward node B is 2R, as is the total resistance looking out towards the 2^1 inputs. These resistors can be combined to simplify the work as shown in fig below:

From the above figure it can be seen that the total resistance looking from nodes C down the branch towards node B or out the branch towards the 2^2 inputs is still 2R. The circuit shown first above can be further reduced to the equivalent as shown on figure below:

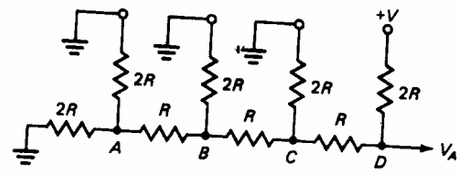


From this equivalent circuit it is clear that the resistance looking back from node. D is 2R, as is the resistance looking out towards the 2^3 inputs.

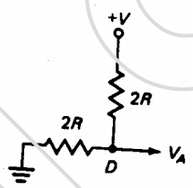
From the above discussion, it can be concluded that the total resistance looking from any node back towards the digital input is 2R. We can use the resistance characteristics of the ladder to determine the output voltages for the various digital inputs.

3. **Determination of output voltage:**

Assume that the digital input signal is 1000. With this input signal the binary ladder can be drawn as shown in figure below:



Since these are no voltage sources to the left of node D, the entire network to the left of this node can be replaced by a resistance of 2R to form the equivalent circuit as shown in fig. below:

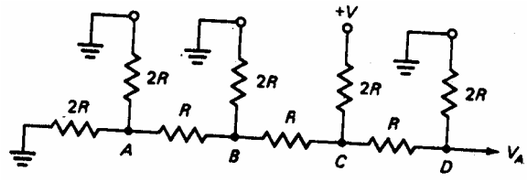


From this equivalent circuit, it can be easily seen that the output voltage (VA) can be calculated using Thevenin's theorem in the following manner:

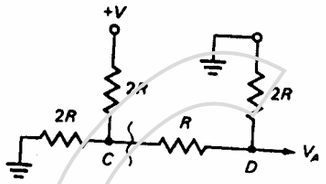
$$V_A = V \times \frac{2R}{2R + 2R} = V \times \frac{2R}{4R} = \frac{V}{2}$$

Thus a 1 in the MSB position will provide an output voltage of V/2.

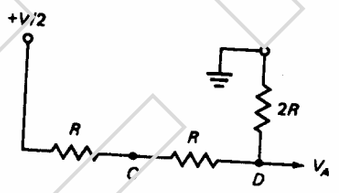
To determine the output voltage due to the second MSB, assume a digital input signal of 0100. This can be represented by the circuit as shown in figure below:



Since these are no voltage sources to the left of node C, the entire network to the left of this node can be replaced by a resistance of 2R as shown in figure below:



Let us now replace the network to the left of node C with its Thevenin equivalent by cutting the circuit on the jagged line shown in figure above. The Thevenin equivalent is clear a resistance R in series with voltage source +V/2. The final equivalent circuit with the Thevenin equivalent included is shown in figure below.



From this circuit the output voltage can be calculated as:

$$VA = \frac{V}{2} \times \frac{2R}{R+R+2R} = \frac{V}{2} \times \frac{2R}{4R} = \frac{V}{4}$$

Thus the second MSB provides an output Voltage of +V/4.

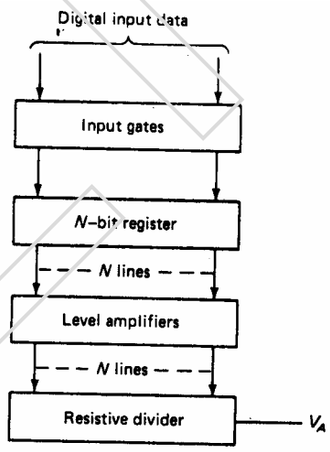
This process continues and it can be shown that the third MSB provides an output Voltage of +V/8, the fourth MSB

provides an output Voltage g +V/16 and so on. The output voltages for the binary ladder can be summarized as follows:

Bit position	Binary weight	Output voltage
MSB	½	V/2
2nd MSB	¼	V/4
3rd MSB	1/8	V/8
4th MSB	1/16	V/16
Nth MSB	$1/2^n$	$V/2^n$

4. D/A Converter

It is in the resistive network (binary ladder) that the actual translation from a digital signal to an analog voltage takes place. But there is need of additional or circuitry to complete the design of the D/A Converter.



As an integral part of the D/A converter there must be a Register that can be used to store the digital information. The simplest register is formed by use of RS flip flops with one flip flop per bit but other types of registers can also be used as discussed in previous chapters. There must also be level amplifiers between the register and the resistive network to ensure that the digital signals presented to the network are all of the same level and are constant. Finally, there must be some form of gating on the input of the register so that the

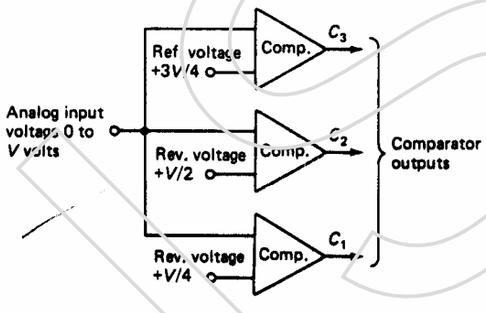
flip-flops can be set with the proper information from the digital system.

.

5. A/D Converter

This operation is more complicated than the D/A Conversion. A number of different methods have been developed, the simplest of which is the simultaneous method, which has than discussed in the subsequent paragraphs.

The simultaneous method of A/D conversion is based on the use of a number of comparator circuits. One such system using three comparator circuits has been shown in the figure below.



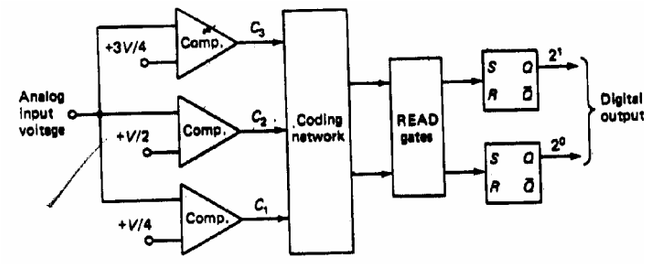
The analog signal to be digitalized serves as one of the inputs to each comparator. The second input is a standard reference voltage. The reference voltages used are +V/4, + V/2 and +3V/4. The system is then capable of accepting an analog input voltage between 0 and +V.

If the analog input signal exceeds the reference voltage to any comparator, that comparator turn on i.e output of the comparator goes High. If all the comparators are off, the analog signal must be between 0 and +V/4. If C_1 is High, C_2 and C_3 are Low, the input must be between +V/2 and +V/2. If C_1 and C_2 are high and C_3 is low, the input must be between

+V/2 and +3V/4. Finally if all comparator inputs are high the input signal must be between +3V/4 and +V. The comparator output levels for the various ranges of input voltages have been summarized in the table below:

Input voltage	Comparator Output		
	C_1	C_2	C_3
0 to +V/4	LOW	LOW	LOW
+V/4 to +V/2	HIGH	LOW	LOW
+V/2 to +3V/4	HIGH	HIGH	LOW
+3V/4 to +V	HIGH	HIGH	HIGH

The above figure shows that there are four Voltage ranges that can be detected by this converter. Four ranges can be easily distinguished by two binary bits. The three comparators outputs are then fed into a coding network to provide 2 bits, which are equivalent to the input analog Voltage. The bits of the coding network are then entered into a flip-flop register for storage. In general it can be said that 2^n-1 comparators are required to convert to a digital signal that has N bits. The block diagram of a 2 bits A/D converter has been shown below:



Exercise

Q 1. Fill in the blanks

(i) For D/A conversion, a resistive network is used called _____. (Binary Ladder)

(ii) Binary ladder is a resistive network, which uses only _____ values of resisters. (Two)

(iii) In a binary ladder the values of resistors are _____ and _____. (R, 2R)

(iv) Thevenin's theorem is associated with _____. (D/A Conversion / Binary ladder)

(v) In a binary ladder output voltage for N^{th} MSB is given by the formula _____. ($V/2^n$)

(vi) A D/A converter consists of _____, _____ , _____, and _____. (Input Gates, Register, Level Amplifiers, Binary Ladder)

(vii) The simplest method used for A/D conversion is _____. (Simultaneous method).

(viii) The number of comparator required to convert to an n-bit digital signal in A/D converter are given by the formula _____. (2^n-1)

Q 2. Short notes

(i). Binary Ladder.

(ii). D/A Converter (DAC)

(iii). A/D Converter (ADC)

Q 3. Explain

(i) A/D Converter giving suitable example and diagram.

(ii) Block diagram of a D/A Converter.