UNIT 10 NUMBER SYSTEM AND CODES

Structure

10. INTRODUCTION

Aim of any number system is to deal with certain quantities which can be measured, monitored, recorded, manipulated arithmetically, observed and utilised. Each quantity has to be represented by its value as efficiently and accurately as is necessary for any application. The numerical value of a quantity can be basically expressed in either analog (continuous) or digital (step by step) method of representation.

In analog method, a quantity is expressed by another quantity which is proportional to the first. For example, the voltage output of an amplifier is measured by a voltmeter. The angular position of the needle of the voltmeter is proportional to the voltage output of the amplifier. Yet another example is of a thermometer. The height to which the mercury rises is proportional to the temperature. In both these examples, the value of voltage and temperature can be anywhere between zero and the maximum limit.

In digital method, the value of a quantity is expressed by some symbols which are called digits, and not by a quantity which is proportionnl to the first. In a digital watch, the time, which changes continuously, is expressed by digits which do not change continuously. The hour-digits change every hour, and the minute-digits change every minute. But there is no measurement of time lapsed between two successive minute-digits. If we want to measure time more accurately then use can be made of watches which have second-digits as well. The second-digits change every second. The time passed between two seconds is not measured. If this is to be measured, we have to use sports watches where the time is measured upto 2 decimal places. Thus the time can be expressed by digits which change step by step (discrete). This step which is an interval of time, in this example, can be made by us as small as necessary, Hence, the analog quantities like time can be represented as digital approximations (e.g. 10 hour 40 minutes, or more accurately 10 hour 39 minutes 50 secondsj. As is clear from the examples above, the accuracy of the value of an analog quantity generally depends upon the judgement of the observer.

Many number systems are being used in digital technology. Most common amongst them are decimal, binary, octal, and hexadecimal systems. We are most familiar with the decimal number system, because we use it everyday. In this unit we shall describe these number systems, the conversion of a number from one system lo another, and finally binary arithmetic. This unit is intended to provide the first step in our understanding of digital electronics.

In the next unit you will be introduced to some of the gates which are fundamental in digital electronics. There you will be familiarised with Boolean algebra which is a mathematical method used in the design of digital systems.

Objectives

After studying this unit, you should be able to

- write binary number and convert it into its decimal equivalent and a decimal number into its binary equivalent,
- explain octal number system, understand octal counting, convert an octal number into its decimal and binary equivalents and decimal and binary numbers into their octal equivalents,
- explain hexadecimal number system, understand hex counting, convert hex number into its decimal, binary and octal equivalents and decimal, binary and octal numbers into their hex equivalents,
- write BCD code and convert a decimal number into its equivalent BCD code and vice versa,
- understand ASCII code,
- learn addition, subtraction, multiplication and division using binary numbers.

10.2 BINARY NUMBER SYSTEM

First let us consider the familiar decimal system. In this system there are ten distinct and different digits (0, 1, 2, 3, 4, 5, 6, 7, 8, and 9). For magnitudes greater than 9 the convention is to arrange digits in rows starting with the most significant on the left and concluding with the least significant on the right. The significance is determined by what is called the 'weighting' of a digit. Thus arises the concept of 'tens', 'hundreds', 'thousands', etc. For example $3458 = (3 \times 10^3) + (4 \times 10^2) + (5 \times 10^1) + (8 \times 10^0)$. Each digit is one of the symbols 0 *to* 9 and is multiplied by a power of ten, depending upon the position of digit. Thus decimal numbers are said to have a base of ten and the multiplying powers 10^0, 10^1, 10^2, 10^3 etc. are called 'weight' or 'positional values'.

In the binary number system (base of 2), there are only two digits: 0 and 1 and the place values are 2^0, 2^1, 2^2, 2^3 etc. Binary digits are abbreviated as bits. For example 1101 is a binary number of 4 bits (i.e., it is a binary number containing four binary digits.)

A binary number may have any number of bits. Consider the number 11001.011. Note the binary point (counterpart of decimal point in decimal number system) in this number. The bit on the extreme right is called least significant bit (LSB) and the bit on the extreme left is called most significant bit (MSB). Each bit has its positional value as shown in Fig. 10.1.

2^4	2^3	2^2	2^1	2^0		2^{-1}	2^{-2}	2^{-3}	Positional values or **weight**
1	1	0	0	1	.	0	1	1	
MSB					Binary Point			LSB	

Fie. 10.1: Binary number: showing positional values (weight) of each bit.

The bits on the left of the binary point are positive powers of 2 and bits on the right of binary point are negative powers of 2. The decimal equivalent of this number is found by summing the products of each bit and its positional value as follows:

$$11001.011_2 = (1 \times 2^4) + (1 \times 2^3) + (0 \times 2^2) + (0 \times 2^1) + (1 \times 2^0) + (0 \times 2^{-1}) + (1 \times 2^{-2}) + (1 \times 2^{-3})$$

$$= 16 + 8 + 0 + 0 + 1 + 0 + 0.250 + 0.125$$

$$= 25.375_{10}.$$

Note that to avoid confusion the subscripts 2 and 10 are written with the numbers to indicate the base of the appropriate number system in which the number is expressed.

Any number can be expressed in binary form in the usual way as shown in Table 10.1.

Table 10.1: Counting in Binary System.

2^3	2^2	2^1	2^0	Binary Number	Decimal Number
0	0	0	0	0000	0
0	0	0	1	0001	1
0	0	1	0	0010	2
0	0	1	1	0011	3
0	1	0	0	0100	4
0	1	0	1	0101	5
0	1	1	0	0110	6
0	1	1	1	0111	7
1	0	0	0	1000	8
1	0	0	1	1001	9
1	0	1	0	1010	10
1	0	1	1	1011	11
1	1	0	0	1100	12
1	1	0	1	1101	13
1	1	1	0	1110	14
1	1	1	1	1111	15

Prom this Table, note that 4 binary digits are required to do counting upto 15_{10}. Thus if the number of bits is n, then wc can go upto 2^n counts and the largest decimal number represented will be $2^n - 1$. For example, in the above cash, $n = 4$ and therefore, the largest decimal number represented is $2^n - 1 = 15_{10}$. To write the next higher number in Table 10.1, we need an additional column for t. ■ ıext power of the base i.e. 2^4.

SAQ 1

What is the largest decimal number that can be represented using 10 bits?

The advantage of binary system is that it has made the job of designing the digital circuitry very easy because only two distinct states or levels of voltages have to be handled. For example, 'ON' state of a bulb may be represented by the bit '1' and 'OFF' state by '0'. In terms of voltages, 0 V or a 'LOW' voltage may represent bit '0' and 5 V or a 'HIGH' voltage may represent bit '1'. Actually, it is not necessary also to have precise voltages assigned to each bit. In analog system the exact value of voltage is very important which makes the design of accurate analog circuitry very difficult. However, in digital systems exact value of voltage is not important because a voltage of 3.9 V means the same thing as a voltage of 4.4 V or 5 V. This aspect will be dealt with in Unit 12.

Let us now see how binary ,numbers can be converted into equivalent decimal form and vice-versa.

10.2.1 Binary to Decimal Conversion

From the example discussed above it is clear that a binary number can be converted into its decimal equivalent by simply adding the weights of various positions in the binary number which have bit 1. For example, consider the conversion of 100011.101_2.

$$1 \quad 0 \quad 0 \quad 0 \quad 1 \quad 1\,.\,1 \quad 0 \quad 1$$

$$2^5 + 0 + 0 + 0 + 2^1 + 2^0 + 2^{-1} + 0 + 2^{-3}$$

$$= 32 + 2 + 1 + 0.5 + 0.125$$

$$= 35.625_{10}$$

Let us Lake up another example of conversion of 11100111.0101_2.

11100111.0101

$$2^7 + 2^6 + 2^5 + 0 + 0 + 2^2 + 2^1 + 2^0 + 0 + 2^{-2} + 0 + 2^{-4}$$

$$= 128 + 64 + 32 + 4 + 2 + 1 + 0.250 + 0.0625$$

$$= 231.3125_{10}.$$

Consider the following examples.

1111.00 = 15

11110.0 = 30

111100.0 = 60

From these examples it is clear that if the binary point is shifted towards right side, then the value of the number is doubled.

Now consider the following examples.

111.100 = 7.5

11.1100 = 3.75

1.11100 = 1.875

From these examples it is clear that if the binary point is shifted towards the left side, then the value of the number is halved.

SAQ 2

Convert 1011.101 into its decimal equivalent.

10.2.2 Decimal to Binary Conversion

A decimal number is converted into its binary equivalent by its repented divisions by 2. The division is continued till we get a quotient of 0. Then all the remainders are arranged sequentially with first remainder taking the position of LSB and the last one taking the position of MSB. Consider the conversion of 27 into its binary equivalent as follows.

```
2 | 27
2 | 13 – 1
2 |  6 – 1
2 |  3 – 0
2 |  1 – 1
     0 – 1

  1  1  0  1  1
```

Thus $27_{10} = 11011_2$.

If the number also has some figures on the right of the decimal point, then this part of the number is to be treated separately. Multiply this part repeatedly by 2. After first multiplication by 2, either 1 or 0 will appear on the left of the decimal point. Keep this 1 or 0 separately and do not multiply it by 2 subsequently. This should be followed for every multiplication. Continue multiplication by 2 till you get all 0s after the decimal point or upto the level of the accuracy desired. This will be clear from the following example. Consider the conversion of 27.625_{10} into its binary equivalent. We have already converted 27 into its binary equivalent which is 11011_2. Now for the conversion of 0,625, multiply it by 2 repeatedly as follows:

```
 .625
 × 2
 -----
 1.250
 |        .250
 |        × 2
 |        -----
 |        0.500
 |        |        .500
 |        |        × 2
 |        |        -----
 |        |        1.000
 |        |        |        .000
 1        0        1
```

Thus 27.625_{10} = 11011.101,.

Let us try another example, conversion of 58.0725_{10} into binary. Split this number in two parts, i.e. 58 and .0725 and convert them into binary separately as described above.

```
2 | 58
2 | 29 – 0
2 | 14 – 1
2 |  7 – 0
2 |  3 – 1
2 |  1 – 1
     0 – 1

        1  1  1  0  1  0
```

Now take up the conversion of ,0725

```
 .0725
 × 2
 ------
 0.1450
 |       .1450
 |       × 2
 |       ------
 |       0.2900
 |       |       .2900
 |       |       × 2
 |       |       ------
 |       |       0.5800
 |       |       |       .5800
 |       |       |       × 2
 |       |       |       ------
 |       |       |       1.1600
 |       |       |       |       .1600
 |       |       |       |       × 2
 |       |       |       |       ------
 |       |       |       |       0.3200
 0       0       0       1       0
```

Thus 58.0725_{10} = 111010.00010,

SAQ 3

What is the binary equivalent of 37.75_{10}?

Representing numbers in binary is very tedious since binary numbers often consist of a large chain of 0's and 1's. Imagine the length of the binary equivalent of a 10 digit decimal number !!. So, convenient shorthand forms for representing the binary numbers are developed such as octal system and hexadecimal system. With these number systems long strings of 0's and 1's can be reduced to a manageable form. Let us see what these systems are.

10.3 OCTAL NUMBER SYSTEM

The octal number system has base-8, that is there arc 8 digits in this system. These digits are 0, 1, 2, 3, 4, 5, 6, and 7. The weight of each octal digit is some power of 8 depending upon the position of the digit. This is explained in Fig. 10.2.

8^4	8^3	8^2	8^1	8^0		8^{-1}	8^{-2}	Weights
1	0	6	2	7	.	4	5	Octal Number
MSD					Octal Point		LSD	

Fig. 10.2: Octal number: showing positional values (weights) **of each digit.**

Octal number does not include the decimal digits 8 and 9. If any number includes decimal digits 8 and 9, then the number can not bc an octal number.

Now let us see how counting is done in octal system. You arc familiar with the counting in decimal system. In decimal system there are 10 digits from 1 to 9 hence the counting in such systcm is done as in Table 10.2.

Table 10.2: Counting in decimal system.

0	10	20	30	40	50	60	70	100	110	....	170
1	11	21	31	41	51	61	71	101	111		
2	12	22	32	42	52	62	72	102	112		
3	13	23	33	43	53	63	73	103	113		
4	14	24	34	44	54	64	74	104	114		
5	15	25	35	45	55	65	75	105	115		
6	16	26	36	46	56	66	76	106	116		
7	17	27	37	47	57	67	77	107	117		
8	18	28	38	48	58	68	78	108	118		
9	19	29	39	49	59	69	79	109	119	....	179

In the same style, counting can be done in octal system as shown in Table 10.3.

Table 10.3: Counting in octal system.

0	10	20	30	40	50	60	70	100
1	11	21	31	41	51	61	71	101
2	12	22	32	42	52	62	72	102
3	13	23	33	43	53	63	73	103
4	14	24	34	44	54	64	74	104
5	15	25	35	45	55	65	75	105
6	16	26	36	46	56	66	76	106
7	17	27	37	47	57	67	77	107

In the octal counting, if n is the number of digits then the total number of counts is 8^n. The largest decimal number represented by an octal number having n digits is $8^n - 1$. Thus with $n = 4$, the total number of counts is $8^4 = 4096$ and the largest decimal number represented is $4096 - 1 = 4095_{10}$.

SAQ 4

Can the number 128.96 be an octal number?

SAQ 5

What is the largest decimal number that can be represented by a three digit octal number?

10.3.1 Octal to Decimal Conversion

As has been done in case of binary numbers, an octal number can be converted into its decimal equivalent by multiplying the octal digit by its positional value. For example,

$$126.25_8 = (1 \times 8^2) + (2 \times 8^1) + (6 \times 8^0) + (2 \times 8^{-1}) + (5 \times 8^{-2})$$
$$= 64 + 16 + 6 + 0.25 + 0.078$$
$$= 86.328_{10}.$$

Let us convert 36.4_8 into decimal number.

$$36.4_8 = 3 \times 8^1 + 6 \times 8^0 + 4 \times 8^{-1}$$
$$= 24 + 6 + 0.5$$
$$= 30.5_{10}$$

SAQ 6

What is the decimal equivalent of 37.2_8?

10.3.2 . Decimal to Octal Conversion

A decimal number can be converted by repeated division by 8 into equivalent octal number. This method is similar to that adopted in decimal to binary conversion,
If the decimal number has some digits on the right of the decimal point, then this part of the number is converted into its octal equivalent by repeatedly multiplying it by 8. The process is same as has been followed in binary number system. Consider the conversion of 126.38_{10} into its decimal equivalent. Split it into two parts, that is 126 and .38

```
8 | 126
8 | 15 - 6
8 |  1 - 7
8 |  0 - 1

     1   7   6
```

Now the conversion of **.38** is as follows:

.38 × 8 = 3.04	.04 × 8 = 0.32	.32 × 8 = 2.56	.56 × 8 = 4.48
3	0	2	4

.48

Thus 126.38_{10} = 176.3024,

SAQ 7

What is the octal equivalent of 15.250_{10}?

10.3.3 Octal to Binary Conversion

In the octal number system the highest octal digit i.e. 7 can be expressed as a 3-bit binary number. Therefore, all the octal digits have to be represented by a 3-bit binary number. The binary equivalent of each octal digit is shown in Table 10.4. The main advantage of the octal number system is the easiness with which any octal number can be converted into its binary equivalent.

Table 10.4: Binary equivalent of each octal digit.

Octal digit	3-bit binary equivalent
0	000
1	001
2	010
3	011
4	100
5	101
6	110
7	111

Using this conversion of octal digit into 3-bit binary number, any octal number can be converted into its binary equivalent by simply replacing each octal digit by a 3-bit binary number. For example, conversion of 567, into its binary equivalent is:

567, = 101110111

= 101110111,

Thus 567, = 101110111_2.

Another example:

Conversion of 672.27, into its binary equivalent.

672.27, = 110 111 010.010 111

= 110111010.010111,

Thus 672.27, = 110111010.0101 11,

SAQ 8

Represent 10027.12, in binary number.

10.3.4 Binary to Octal Conversion

A binary number can be converted into its octal equivalent by first making groups of 3-bits starting from the LSB side. If the MSB side does not have 3 bits, then add 0s to make the last group of 3 bits. Then by replacing each group of 3 bits by its octal equivalent, a binary number can be converted into its binary equivalent. For example, consider the conversion of 1100011001, into its octal equivalent as follows:

1100011001_2 = 1 100 011 001

= 001 100 011 001 [As the MSB side does not have 3 bits, we have added two 0's to make the last group of 3 bits]

= 1 4 3 1

= 1431,

Thus 1100011001, = 1431,

SAQ 9

What is the octal equivalent of 10010_2?

10.4 HEXADECIMAL NUMBER SYSTEM.

The hexadecimal number system has base-16, that is it has 16 digits (Hexadecimal means '16'). These digits are 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, and F. The digits A, B, C, D, E, and F have equivalent decimal values 10, 11, 12, 13, 14, and 15 respectively. Each Hex (hexadecimal is popularly known as hex) digit in a hex number has a positional value that is some power of 16 depending upon its position in the number. This is illustrated in Fig. 10.3.

16^4	16^3	16^2	16^1	16^0	16^{-1}	16^{-2}	Weights
1	2	4	A	F .	B	9	Hex number
MSD				Hex Point		LSD	

Pig. 10.3: Hexadecimal number: showing positional values (weight) of each digit.

The relationship of hex digits with decimal and binary numbers is given in Table 10.5. Note that to represent the largest hex digit we require four binary bits. Therefore, the binary equivalent of all the hex digits have to be written in 4 bit numbers.

Table 10.5: Binary and Decimal equivalent of each Hex Digit.

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

While doing counting in hex number system if n is the number of hex digits then counting can be done upto 16^n counts and the largest decimal number represented by a hex number is $16^n - 1$. The hex counting is shown in Table **10.6.**

Table 10.6: Counting in Hexadecimal system.

0	10	20	10	40	...	90	A0	B0	C0	D0	E0	F0	100
1	11	21	31	41	...	91	A1	B1	C1	D1	E1	F1	
2	12	22	32	42	...	92	A2	B2	C2	D2	E2	F2	
3	13	23	33	43	...	93	A3	B3	C3	D3	E3	F3	
9	19	29	39	49	...	99	A9	B9	C9	D9	E9	F9	
A	1A	2A	3A	4A	...	9A	AA	BA	CA	DA	EA	FA	
B	1B	2B	3B	4B	...	9B	AB	BB	CB	DB	EB	FB	
C	1C	2C	3C	4C	...	9C	AC	BC	CC	DC	EC	FC	
D	1D	2D	3D	4D	...	9D	AD	BD	CD	DD	ED	FD	
E	1E	2E	3E	4E	...	9E	AE	BE	CE	DE	EE	FE	
F	1F	2F	3F	4F	...	9F	AF	BF	CF	DF	EF	FF	

SAQ 10

What is the number next to $835F_{16}$?

SAQ 11

What is the largest decimal number represented by a 3-digit hex number?

10.4.1 Hex to Decimal Conversion

Hex to decimal conversion is done in the same way as in the cases of binary and octal to decimal conversions. **A** hex number is converted into its equivalent decimal number by summing the products of the weights of each digit and their values. This is clear from the example of conversion of $514.AF_{16}$ into its decimal equivalent.

$$
\begin{aligned}
514.AF_{16} &= 5 \times 16^2 + 1 \times 16^1 + 4 \times 16^0 + 10 \times 16^{-1} + 15 \times 16^{-2} \\
&= 1280 + 16 + 4 + 0.625 + 0.0586 \\
&= 1300.6836_{10}.
\end{aligned}
$$

Another example:

$$
\begin{aligned}
3BE.1A_{16} &= 3 \times 16^2 + 11 \times 16^1 + 14 \times 16^0 + 1 \times 16^{-1} + 10 \times 16^{-2} \\
&= 768 + 176 + 14 + 0.0625 + 0.0391 \\
&= 958.1016_{10}.
\end{aligned}
$$

SAQ 12

What is decimal equivalent of $1BE2_{16}$?

10.4.2 Decimal to Hex Conversion

A decimal number is converted into hex number in the same way as a decimal number is converted into its equivalent binary and octal numbers. The part of the number on the left of the decimal point is to be divided repeatedly by 16 and the part an the right of the decimal point is to be repeatedly multiplied by **16.** This will be clear from the examples of conversion of 579.26_{10} into hex equivalent. Split the number into two parts 579 and .26.

```
16 | 579
16 | 36 - 3 ----------+
16 |  2 - 4 -------+   |
   |  0 - 2 ---+   |   |
               2   4   3
```

Thus $579_{10} = 243_{16}$.

Now .26 is converted into hex number as follows:

```
  .26
× 16
-----
 4.16
  |        .16
  |      × 16
  |      -----
  |       2.56
  |        |        .56
  |        |      × 16
  |        |      -----
  |        |       8.96
  |        |        |          .96
  4        2        8
```

Thus $579.26_{10} = 243.428_{16}$.

SAQ 13

What is the hex equivalent of 37_{10}.

10.4.3 Hex to Binary Conversion

As in octal number system, a hex number is converted into its binary equivalent by replacing each hex digit by its equivalent 4-bit binary number, This is clear from the following example:

$BA6_{16}$ = B A 6

= 1011 1010 0110

= 101110100110_2.

SAQ 14

What is the binary equivalent of $6F10_{16}$?

10.4.4 Binary to Hex Conversion

By a process that is reverse of the process described in section 10.4.4 above, a binary number can be converted into its hex equivalent. Starting from the LSB side, group the binary number bits into groups of four bits. If towards the MSB side, the number of bits is less than Sour then add zeros on the left of the **MSB** so that the group of four is complete. Replace each group by its equivalent hex digit. This is clear from the following example:

1001101110, = 0010 0110 1110

= 2 6 E

= $26E_{16}$.

SAQ 15

What is the hex equivalent of $1100101010011l1_2$?

10.4.5 Hex to Octal Conversion

Each digit of the hex number is first converted into its equivalent four bit binary number. Then the bits of the equivalent binary number are grouped into groups of three bits. Then each group is replaced by its equivalent oclal digit to get the octal number. For example:

$$
\begin{aligned}
5AF_{16} &= 0101 \quad 1010 \quad 1111 \\
&= 010110101111 \\
&= 010 \quad 110 \quad 101 \quad 111 \\
&= 2 \quad 6 \quad 5 \quad 7 \\
&= 2657_8
\end{aligned}
$$

SAQ 16

What is the octal equivalent of $5A9_{16}$?

10.4.6 Octal to Hex Conversion

For octal to hex chpversion, just reverse the process described in section 10.4.6 above. This is clear from the following example:

$$
\begin{aligned}
5457_8 &= 101 \quad 100 \quad 101 \quad 111 \\
&= 1011 \quad 0010 \quad 1111 \\
&= B \quad 2 \quad F \\
&= B2F_{16}.
\end{aligned}
$$

This method can also be applied to hex to decimal and decimal to hex conversions. For example consider the conversion of $3C_{16}$ into its decimal equivalent:

$$
\begin{aligned}
3C_{16} &= 0011 \quad 1100 \\
&= 111100_2.
\end{aligned}
$$

Check the conversion.

$$
\begin{aligned}
3C_{16} &= 3 \times 16^1 + C \times 16^0 \\
&= 3 \times 16^1 + 12 \times 16^0 \\
&= 48 + 12 \\
&= 60_{10}. \\
111100_2 &= 2^5 + 2^4 + 2^3 + 2^2 \\
&= 32 + 16 + 8 + 4 \\
&= 60_{10}.
\end{aligned}
$$

Thus $3C_{16} = 111100_2 = 60_{10}$.

SAQ 17

What is the hex equivalent of 327_8?

10.5 CODES

So far you have learnt about binary, octal and hexadecimal number system. For any number system with a base B and digits N_0 (LSB), N_1, N_2 ...... N_m (MSB), the decimal equivalent N_{10} is given by

$$N_{10} = N_m \times B^m + \ldots N_3 \times B^3 + N_2 \times B^2 + N_1 \times B^1 + N_0 B^0 \quad (10.1)$$

You have also observed that a number in any system can be written in the binary form. A number code is a relationship between the binary digits and the number represented. Thus, all number systems are codes and the decimal equivalent is given by Eq. (10.1) But there are other relationships or codes that relate decimal numbers and groups of binary digits that do not obey Eq. (10.1) These relationships are called codes. We will now discuss some of the important codes used in digital work.

10.5.1 BCD Code

In BCD (BCD stands for binary coded decimal) code, each digit of a decimal number is converted into its four bit binary equivalent. The largest decimal digit is 9, therefore the largest binary equivalent is 1001. This is illustrated as follows:

$$951_{10} = 1001 \quad 0101 \quad 0001$$
$$= 100101010001_{BCD}.$$

Remember that the conversion of a decimal number into its binary equivalent and BCD equivalent leads to two different numbers. For example:

$$158_{10} = 0001 \quad 0101 \quad 1000$$
$$= 101011000_{BCD}.$$

158_{10} = 10011110, (obtained by repeated division method).

Thus we see that it is quite easy to convert from decimal to BCD and from BCD to decimal. It is much easier to convert from BCD to decimal than from straight binary to decimal, because we only have to count upto 9 in binary to do so. However, it takes more bits to represent a number in BCD than in binary.

A BCD number is converted into its decimal equivalent by the reverse process. For example:

$$1010101110010_{BCD} = 0001 \quad 0101 \quad 0111 \quad 0010$$
$$= 1 \quad 5 \quad 7 \quad 2$$
$$= 1572_{10}.$$

Although the main function of a computer is to perform arithmetic operations, it also processes messages and information in a language that uses letters of the alphabet (e.g. English) and data of other kinds. Computers operate by coding letters of the alphabet, other symbols, and data into binary Sorni. The code used for this purpose is ASCII code about which you will study now.

10.5.2 ASCII Code

The word ASCII is an acronym of American Standard Code for Information Interchange. This is the alphanumeric code most widely used in computers. The alphanumeric code is one that represents alphabets, numerical numbers, punctuation marks and other special characters recognised by a computer. The ASCII code is a 7-bit code representing 26 English alphabets, 0 through 9 digits, punctuation marks, etc, A 7-bit code has 2^7 = 128 possible code groups which are quite sufficient. A partial ASCII code listing is shown in Table 10.6.

Table 10.6: Some of the ASCII codes for numbers, alphabets and other common symbols.

$A_6A_5A_4$						$A_3A_2A_1A_0$
010	011	100	101	110	111	
SP	0	@	P		p	0000
!	1	A	Q	a	q	0001
"	2	B	R	b	r	0010
#	3	C	S	c	s	0011
$	4	D	T	d	t	0100
%	5	E	U	e	u	0101
&	6	F	V	f	v	0110
'	7	G	W	g	w	0111
(	8	H	X	h	x	1000
)	9	I	Y	i	y	1001
*	:	J	Z	j	z	1010
+	;	K		k		1011
,	<	L		l		1100
–	=	M		m		1101
.	>	N		n		1110
/	?	O		o		1111

The code is $A_6A_5A_4A_3A_2A_1A_0$. For example, A has $A_6A_5A_4$ of 100 and an $A_3A_2A_1A_0$ of 0001. Therefore, its ASCII code is

100 0001 = A.

The ASCII code for a is 110 0001.

SAQ **18**

What is the ASCII code of SHARMA'?

10.6 BINARY ARITHMETIC

Digital computers can perform arithmetic operations using only binary numbers. We will learn how to add, subtract, multiply and divide binary numbers. We will first review this in the familiar decimal system and apply the same ideas to binary system.

10.6.1 Addition

Let us recall the addition in decimal numbers. Suppose we want to add 563 and 146. We start adding the digits in the least significant column. We get,

```
  563
+ 146
-----
    9   (no carry to tlie next column)
-----
```

Next, the digits of the second column are added and we get,

```
 563
+146
----
  09   (carry 1 to the next column)
----
```

In this case 6 + 4 gives 0, with a carry 1 to the next column. Then the digits of the last column and the 'carry' from the previous column are added. We get,

```
563
146
1     ........... carry from previous column
----
709   (no carry)
----
```

Addition of binary numbers can be carried out in a similar way by the column method. But before we do this, we need to discuss four simple cases. We known in the decimal number system, 3 + 6 = 9 symbolizes the combining of ... with ...... to get a total of .... Let us now discuss the four simple cases.

Case 1: When nothing is combined with nothing, we get nothing. The binary representation of this is 0 + 0 = 0.

Case 2: When nothing is combined with ., we get. Using binary numbers to denote this gives 0 + 1 = 1.

Case 3: Combining. with nothing gives. The binary equivalent of this is 1 + 0 = 1.

Case 4: When we combine. with ., the result is .. Using binary numbers, we symbolize 1 + 1 = 10.

The last result is sometimes confusing because of our long time association with decimal numbers. But it is correct qnd makes sense because we are using binary numbers. Binary number 10 stands for.. and not for .... (ten).

To summarize our results for binary addition,

0 + 0 = 0

0 + 1 = 1

1 + 0 = 1

1 + 1 = 10

To add large binary numbers, carry into higher-order columns as is done with decimal numbers. As an example, add 10 to 10 as follows

```
  10
+ 10
----
 100
----
```

In the first column, 0 plus 0 is 0. In the second column, 1 plus 1 is 0, carry a 1. As another example, take 1 + 1 + 1, Add two of the 1's to get 10 + 1.

Adding again gives 11 as follows:

1 + 1 + 1 = 10 + 1 = 11

See another example

```
  101   first column: 1 + 0 = 1
+ 110   secondcolumn: 0 + 1 = 1
-----
 1011   third column: 1 + 1 = 10 (zero, carry one)
-----
```

Further examples are

```
   110          101.011
 + 111        + 111.110
 -----        ---------
  1101         1101.001
```

In all digital networks or computers only two binary numbcrs arc added at a time. To add more than two numbers, first two numbcrs are added, then to their sum the third number is added, and so on. Therefore, wc should not worry about the addition of more than two numbcrs. The computer can add numhcrs in a few microseconds or even less. You will see that the multiplication, division and subtractions arc actually done by the computers by way of addition.

SAQ 19

Add the following : (a) 1010 and 1101

(b) 1011 and 1010

10.6.2 Subtraction

Binary subtraction is done in the same way as in decimal system. Let us recall the decimal subtraction, for example:

```
  56
- 49
----
   7
```

In this example, a 1 is borrowed from the ten's position giving 16 in the LSD. Then 16 − 9 = 7. Borrowing a 1 from the ten's position leaves 4 in place of 5. Then 4 − 4 = 0. In the same way the binary subtraction can be done.

To subtract binary numbers, we first need to discuss four simple cases.

Cise 1 0 − 0 = 0

Case 2 1 − 0 = 1

Case 3 1 − 1 = 0

Case 4 10 − 1 = 1

The last result represents . . − . = . which makes sense. To subtract large binary numbcrs, subtract column by column, borrowing from the adjacent column when necessary. For example, in subtracting 101 from 111, we proceed as follows:

111	first column :	1 − 1 = 0
− 101	second column:	1 − 0 = 1
010	third column:	1 − 1 = 0

Here is another example: subtract 1010 from 1101

1101	first column:	1 − 0 = 1
− 1010	second column:	10 (after borrow) − 1 = 1
	third column:	0 (after borrow) − 1 = 0
	fourth column:	1 − 1 = 0.

SAQ *20*

Subtract binary 100011 from 110011.

10.6.3 Multiplication and Division

The multiplication of binary numbers is also done in the same manner as in decimal system. It is rather easier, because the multiplication table for binary has only four cases.

Case 1 $0 \times 0 = 0$

Case 2 $0 \times 1 = 0$

Case 3 $1 \times 0 = 0$

Case 4 $1 \times 1 = 1$

For example, in multiplying 1101 by 1001, we proceed as follows:

```
    1101
    1001
   -----
    1101
   0000
  0000
 1101
 -------
 1110101
```

In the beginning the first partial product is written. Subsequently each partial product is written below the previous one by shifting one place towards left relative to the previous place. However, the digital circuits or computers add only two binary numbers at a time. Therefore, to the suin of first two partial products is added the third partial product. To this sum is added the third partial product to give the final sum.

The process of dividing a binary number is once again the same as followed in the decimal system. To divide 1100 by 10, we proceed as follows.

```
      110
    ------
 10| 1100
     10
     --
      10
      10
      --
       00
```

SAQ **21**

Multiply 10110 by 110.

10.7 SUMMARY

- There are mainly four number systems naincly binary, octal, decimal and hexadecimal which have 2, 8, 10 and 16 digits respectively. But it is the ease in applications that decides which kind of number system should tie defined and used. Every computer uses two or more of the above mentioned number systems simultaneously.
- The binary number system has only two digits: 0 and 1. A binary digit is called bit, A binary number can be converted into its equivalent octal, decimal and hex numbers as described in the text. And also octal, decimal and hex numbers can be converted into equivalent binary numbers.
- The octal number system has 8 digits: 0 through 7. An octal number can be converted into its equivalent binary, decimal and hex numbers and vice versa as described in the text.

- The hex number system has 16 digits: 0 through 9, A (10) through F (15). As in the other systems, the hex numbers can be converted as described in the text into their binary, octal and decimal equivalents and vice versa.

  It is possible to arrange sets of binary digits to represent numbers, letters of the alphabet or other information by using a given code. Some of the important codes are BCD and ASCII codes.

- In the BCD code, each decimal digit is replaced by its 4-bit binary equivalent. The conversion of BCD code into its decimal equivalent and vice versa is quite easy. Therefore, it is quite often used in computers.

- The ASCII code is the most widely used alphanumeric code. It is a 7-bit binary number–and has $2^7 = 128$ possible 7-bit binary numbers which are quite sufficient to describe the capital and small letters of the alphabet, digits, punctuation marks, and other symbols.

  The fundamental arithmetic of binary addition is contained in four rules:

  1. $0 + 0 = 0$
  2. $0 + 1 = 1$
  3. $1 + 0 = 1$
  4. $1 + 1 = 0$ but 1 must be carried over to next higher (more significant) bit.

- The fundamental arithmetic of, binary subtraction is contained in four rules:

  1. $0 - 0 = 0$
  2. $0 - 1 = 1$ and borrow 1 from the next more significant bit
  3. $1 - 0 = 1$
  4. $1 - 1 = 0$

- The four rules for binary multiplication are:

  1. $0 \times 0 = 0$
  2. $0 \times 1 = 1$
  3. $1 \times 0 = 1$
  4. $1 \times 1 = 0$

10.8 TERMINAL QUESTIONS

1) In the binary sequence, what is number that followp 10111?
2) What is the largest decimal number that can be expressed by 6 bits'?
3) Convert 11011011010.1101, into its decimal equivalent.
4) Convert 372.125_{10} into its binary equivalent.
5) Convert 89.875_{10} into its binary equivalent.
6) What is the largest decimal number represented by a five digit octal number?
7) Convert 7777_8 into its decimal equivalent.
8) Convert 6789_{10} into its octal equivalent,
9) Convert 23401, into its binary equivalent.
10) Convert 1100110111001010_2 into its octal equivalent.
11) Add the following binary numbers 1110001 and 1010101
12) Multiply 101.1 by 11.01
13) Divide 11011 by 100

10.9 SOLUTIONS AND ANSWERS

SAQs

1. Largest decimal number $=2^n - 1$. With $n = 10$, $2^{10} - 1 = 1024 - 1 = 1023_{10}$.
2. 11.625_{10}.
3. 100101.11.
4. No. Octal numbers do not have digits 8 and 9.
5. The largest decimal number is $8^3 - 1 = 512 - 1 = 511_{10}$.
6. 31.250_{10}
7. 17.2_8.
8. $0010000000101\,1.001010_2$.
9. 22_8.
10. 8360_{16}.
11. Largest decimal number $= 16^3 - 1 = 4096 - 1 = 4095_{10}$.
12. $1BE2_{16} = 1 \times 16^3 + 11 \times 16^2 + 14 \times 16^1 + 2 \times 16^0$

    $= 4096 + 2816 + 224 + 2$

    $= 7138_{10}$.
13. 25_{16}.
14. $6F10_{16}$ = 0110 1111 0001 0000 = 110111100010000_2.
15. 110010101001111, = 0110 0101 0100 1111 = $654F_{16}$.
16. $5A9_{16}$ = 0101 1010 1001

    = 010 110 101 001

    = 2651,.
17. 3278 = 011 010 111

    = 0 1101 0111

    = $D7_{16}$
18. SHARMA = 1010011 1001000 1000001 1010010 1001101 1000001
19. (a) 10111 (h) 10101
20. 10000.
21. 10000100.

TQs

1) 11000_2.
2) 63_{10}.
3) 1754.8125_{10}.
4) 101110100.001_2.
5) 1011001.111_2.

6) 32767_{10}.

7) 4095_{10}.

8) 15205_8.

9) 10011100000001_2.

10) 1100110111001010_2 = 001 100 110 111 001 010

= 1 4 6 7 1 2

= 146712_8.

11) 11000110

12) 10001.111

13) 110.11