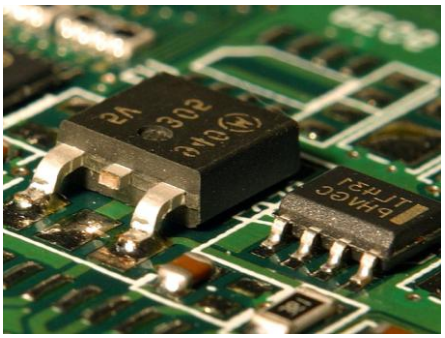


The number systems you will learn in this unit form the basis of all digital circuitry used around us. This is a photograph of computer chips.

Source: http://www.public-domain-image.com/public-domain-images-pictures-free-stock-photos/objects-public-domain-images-pictures/electronics-devices-public-domain-images-pictures/computer-components-pictures/computers-chips-circuits.jpg created by Jon Sullivan

UNIT 6

NUMBER SYSTEMS

Structure

STUDY GUIDE

In this unit, you will learn about the number systems being used in digital technology: decimal, binary, octal, and hexadecimal and the conversion of a number from one system to another. You know the decimal number system very well. You have to also learn binary arithmetic, binary addition, subtraction and multiplication. You have to practice the inter-conversions between these number systems and binary arithmetic thoroughly by solving problems. Therefore, our advice: Do all steps given in the unit and solved Examples yourself, solve all SAQs and Terminal Questions on your own. Only then you can learn the concepts of this unit properly.

"Science is a way of thinking much more than it is a body of knowledge."

Carl Sagan

6.1 INTRODUCTION

You know the decimal system right from your early school days and you use it all the time. You know that numbers in Physics deal with physical quantities that can be observed, measured, monitored, recorded, manipulated arithmetically, and utilized. 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.

So far, you have learnt about the analog method of expressing a quantity by another quantity, which is proportional to the first. For example, the voltage in a circuit across a resistor is measured by a voltmeter. The angular position of the needle of the voltmeter is proportional to the voltage across that resistor. When you measure temperature using 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 proportional to another quantity. For example, 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 the time elapsed between two successive minute-digits.

If we want to measure time more accurately then we can use watches which have second-digits as well. The second-digits change every second. But, even in this case, the time elapsed between two seconds is not measured. If this is to be measured, we have to use sports watches where the time is measured up to 2 decimal places.

Thus, 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 as small as necessary by us. 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 59 seconds).

Since digital technology is based entirely on discrete digits, you must first learn about the number systems that are used in digital electronics. Many number systems are being used and the most common among them are: decimal, binary, octal and hexadecimal systems.

Since you know the decimal system, in Secs. 6.2 to 6.4 of this unit, you will learn about these systems and the conversion of a number from one system to another. In Sec. 6.5, we introduce the BCD and ASCII codes and in Sec. 6.6, 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 logic gates, which are fundamental in digital electronics.

Expected Learning Outcomes

After studying this unit, you should be able to:

- write a binary number and convert it into its decimal equivalent and a decimal number into its binary equivalent;
- define octal number system, perform octal counting, convert an octal number into its decimal and binary equivalents, and decimal and binary numbers into their octal equivalents;
- define hexadecimal number system, perform hex counting, convert hex number into its decimal, binary and octal equivalents and decimal, binary and octal numbers into their hex equivalents;
- write BCD and ASCII codes and convert a decimal number into its equivalent BCD and ASCII codes, and vice versa; and
- solve simple problems on binary arithmetic.

6.2 BINARY NUMBER SYSTEM

Consider the decimal system you have used so far. Note that in this system there are ten distinct and different digits (0, 1, 2, 3, 4, 5, 6, 7, 8, and 9). To represent magnitudes greater than 9, we arrange digits in rows starting with the most significant one on the left and ending with the least significant one on the right. The significance is determined by what is called the 'weight' of a digit. Thus arise the concepts of 'ten', 'hundreds', 'thousands', etc.

For example, in the number 1234, each digit is one of the symbols 0 to 9 and is multiplied by a power of ten, depending upon the position of digit. So, the number is one thousand two hundred and thirty four. You know very well how to express numbers in the decimal system.

Thus, decimal numbers having ten digits (0 to 9) are said to have **a base of ten** and the multiplying powers of 10 are called '**weights**' or '**positional values**'.

You must learn that a **string** is a group of characters (either letters or numbers) written in a sequence. For example, 1234 or 1A23 are strings.

Now, you can learn the binary system. Binary means **two**. In the binary number system there are only **two digits**: 0 and 1 and so, it is **of base two**. Binary digits are called **bits** in short.

All binary numbers are expressed as strings of 0s and 1s. For example, 1101 and 100100 are binary numbers of 4 bits and 6 bits, respectively. So, these are binary numbers containing four and six binary digits or bits. You may ask: **What is meant by base two**?

You know that in decimal number, the base is ten (there being ten digits). So, we express decimal numbers, say 12, 123 and 1234, as

$12 = 1\times10^1 + 2\times10^0$ (6.1a)

$123 = 1\times10^2 + 2\times10^1 + 3\times10^0$ (6.1b)

$1234 = 1\times10^3 + 2\times10^2 + 3\times10^1 + 4\times10^0$, and so on. (6.1c)

The powers of 10 in Eqs. (6.1a to c) are the weights: The weight of the right-most digit is 10^0 (ones), the second digit has a weight of 10^1 (tens), the third digit a weight of 10^2 (hundreds), the fourth digit a weight of 10^3 (thousands), and so on.

The binary number with **base 2 uses powers of 2 as weights.** These weights are: 2^0 (ones), 2^1 (twos), 2^2 (fours), 2^3 (eights), and so on. So, with these weights, we express decimal numbers as binary numbers as follows:

0 as 0, (6.2a)

1 as 1, since $1 = 1\times 2^0$, (6.2b)

2 as 10, since $2 = 1\times 2^1 + 0\times 2^0$, (6.2c)

4 as 100, since $4 = 1\times 2^2 + 0\times 2^1 + 0\times 2^0$, (6.2d)

9 as 1001, since $9 = 1\times 2^3 + 0\times 2^2 + 0\times 2^1 + 1\times 2^0$, and so on. (6.2e)

Notice that we have replaced base 10 of Eqs. (6.1a to c) with base 2 in expressing the numbers 1, 2, 4 and 9 above. The binary equivalents of the decimal numbers 1 to 9 are given in Table 6.1.

Table 6.1: Equivalence of decimal and binary numbers

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

Verify the equivalence between decimal and binary numbers given in Table 6.1 before studying further. Try SAQ 1.

SAQ 1 – Equivalence of binary and decimal numbers

Verify the equivalence of respective binary numbers given in Table 6.1 with the decimal numbers 3, 5, 6, 7 and 8.

We strongly advise you to memorise Table 6.1 for better recall and ease of calculation. Using the basic method shown above, we can convert any binary number expressed as a string of any number of bits to a decimal number. Let us learn how to do that.

6.2.1 Binary to Decimal Conversion

Suppose we wish to express numbers beyond 9 in a string of bits using base 2. What is the highest decimal number that we can express using a string of 4 bits, 1111? Let us check using the method shown from Eqs. (6.2a to e). It is:

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

In a string of 4 bits, you can verify that the decimal numbers are expressed as:

Number 1: $0\times 2^3 + 0\times 2^2 + 0\times 2^1 + 1\times 2^0 = 0001$

Number 2: $0\times 2^3 + 0\times 2^2 + 1\times 2^1 + 0\times 2^0 = 0010$

So, what is the string 0101 equal to? Let us write it:

$$0101 = 0\times 2^3 + 1\times 2^2 + 0\times 2^1 + 1\times 2^0 = 5$$

The numbers 0 to 15 are expressed as binary digits in a string of 4 bits in Table 6.2. Essentially, we have converted the decimal numbers into binary numbers with the help of the method explained above.

Table 6.2: Equivalence of decimal and binary numbers using string of 4 bits

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

What you have also learnt from this exercise is that if we wish to express larger decimal numbers, we need longer strings containing more bits. **REMEMBER**: **A binary number may have any number of bits**. From Table 6.2, note that 4 binary digits are required for counting up to 15. We can generalise this result to see how many bits are required in a string to express a decimal number. **You should always remember**:

> If the number of bits is *n*, then we can go up to 2^n counts and the largest decimal number represented will be $2^n - 1$.

For example, in Table 6.2, $n = 4$, and, therefore, the largest decimal number represented is 15. To convert the next higher decimal number, i.e., 16, we need an additional column for the next power of the base in the binary system, namely, 2^4, as shown in Fig. 6.1a.

1	0	0	0	0
2^4	2^3	2^2	2^1	2^0

(a)

3	5	2	1	7
10^4	10^3	10^2	10^1	10^0

(b)

Fig. 6.1: a) Binary weights; b) decimal weights.

So, the binary number corresponding to the decimal number 16 (Fig. 6.1a) is:

$$1\times 2^4 + 0\times 2^3 + 0\times 2^2 + 0\times 2^1 + 0\times 2^0 = 16$$

Note that this is similar to how we write the weights of decimal numbers. For example, in Fig. 6.1b, we have shown the decimal number 35217 written as multiples of powers of 10 as per their place values or weights:

$$3\times 10^4 + 5\times 10^3 + 2\times 10^2 + 1\times 10^1 + 7\times 10^0 = 30000 + 5000 + 200 + 10 + 7$$
$$= 35217$$

You can use the following technique to convert a binary number to its decimal equivalent quickly:

1. Write the binary number.
2. Write the weights 1, 2, 4, 8, 16, etc. under each digit of the given binary number, starting from right.
3. Cross out the weight under each 0 in the binary number.
4. Add the remaining weights.

So, if you are given the binary number 11001, then you proceed as follows:

1. Write binary number: 1 1 0 0 1
2. Write weights: 16 8 4 2 1
3. Cross out weights under 0: 16 8 ~~0~~ ~~0~~ 1
4. Add weights: $16 + 8 + 0 + 0 + 1 = 25$

You can further process steps 2 to 4 in one step:

1 1 0 0 1

16 8 ~~4~~ ~~2~~ 1 ⇔ 25

We have given the positional values (weights) of powers of 2 from 0 to 10 and -1 to -3 in Table 6.3 for ready reference. Using Table 6.3, convert the binary numbers 10011, 11011, 110011 and 1100101 using the above technique. The answers are: 19, 27, 51 and 101.

Table 6.3: Weights of powers of 2

Power of 2	Weight
0	1
1	2
2	4
3	8
4	16
5	32
6	64
7	128
8	256
9	512
10	1024
-1	0.5
-2	0.25
-3	0.125

Let us now find out how to convert fractions in binary number system into decimal numbers.

Consider the number 11001.011. Note that the **point** in this number is called the **binary point** and is the counterpart of decimal point in decimal number system. The bit on the extreme right of a binary number is called the **least significant bit** (LSB) and the **bit on the extreme left is called most significant bit** (MSB). Each bit has its positional value as explained earlier and shown in Fig. 6.2. Let us write this number using the positional values of the bits:

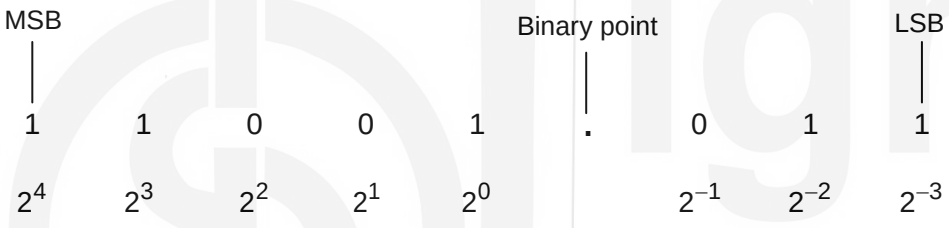


Fig. 6.2: **Binary number with positional value (weight) of each bit.**

Note from Fig. 6.2 that bits on the left of the binary point are expressed as positive powers of 2 and bits on the right of the binary point are expressed as negative powers of 2. Once again, we can find the decimal equivalent of this number by summing the products of each bit and its positional value/weight as follows:

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

$$= 16 + 8 + 0 + 0 + 1 + 0 + 0.25 + 0.125$$

$$= 25.375_{10}$$

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, for example, 10110_2 or 138.6_{10}, etc.

Following the faster technique of calculation, we can write:

1 1 0 0 1 . 0 1 1

16 8 ~~4~~ ~~2~~ 1 ~~1/2~~ 1/4 1/8 ⇔ 25.375

You have to learn to count with bits as digital circuits in computers work and count with binary numbers. Two state operation is the norm in digital electronics. The advantage of binary system is that it has made the job of

The code used in digital electronics:

State	Voltage	Bit
ON	HIGH	1
OFF	LOW	0

designing digital circuitry very easy because **only two distinct states or levels of voltages have to be handled**. For example, the '**ON**' state of a bulb may be represented by the bit '1' and its '**OFF**' state by '0'. In terms of voltages, 0 V or a '**LOW**' voltage may represent the bit '0' and 5 V or a '**HIGH**' voltage may represent the bit '1' (read the margin remark).

Actually, it is not necessary that precise voltages be assigned to each bit. In analog system the exact value of voltage is very important and it makes the design of accurate analog circuitry very difficult. However, in digital systems exact (5 V) value of voltage is not important because any voltage between 2.4 V and 5 V is treated to be high or '1' state; and any voltage between 0 to 0.8 V is taken as low or '0' state. 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. Let us consider a few more examples of binary to decimal conversion to explain this point better.

EXAMPLE 6.1: BINARY TO DECIMAL CONVERSION

Convert the following binary numbers into equivalent decimal numbers:

a) 1111. 00, 11110.0, 111100.0

b) 100011.101, 11100111.0101

c) 111.100, 11.1100, 1.11100

SOLUTION ■ Let us follow the procedure explained above.

a) We write the weights against the bits and add:

$$1111.00 = 1\times 2^3 + 1\times 2^2 + 1\times 2^1 + 1\times 2^0 + 0\times 2^{-1} + 0\times 2^{-2} = 15$$

We can also use the faster technique and write:

1	1	1	1	.	0	0	
8	4	2	1		~~1/2~~	~~1/4~~	$\Leftrightarrow$ 15

Similarly, we get:

$$11110.0 = 1\times 2^4 + 1\times 2^3 + 1\times 2^2 + 1\times 2^1 + 0\times 2^0 + 0\times 2^{-1} = 30$$

$$111100.0 = 1\times 2^5 + 1\times 2^4 + 1\times 2^3 + 1\times 2^2 + 0\times 2^1 + 0\times 2^0 + 0\times 2^{-1} = 60$$

Note from these examples that if the binary point is shifted towards the right side in a given number, then the value of the number is doubled.

b) We use the faster two-step technique: For 100011.101, we get:

1	0	0	0	1	1	.	1	0	1
32	~~16~~	~~8~~	~~4~~	2	1		1/2	~~1/4~~	1/8

Thus, $100011.101_2 = 35.625_{10}$

For 11100111.0101, we get:

1	1	1	0	0	1	1	1	.	0	1	0	1
128	64	32	~~16~~	~~8~~	~~4~~	2	1		~~1/2~~	1/4	~~1/8~~	1/16

Thus, $11100111.0101_2 = 231.3125_{10}$

c) By now, you should be able to calculate mentally:

$111.100 = 4 + 2 + 1 + 0.5 = 7.5,$

$11.1100 = 2 + 1 + 0.5 + 0.25 = 3.75,$

$1.11100 = 1 + 0.5 + 0.25 + 0.125 = 1.875$

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

You may now like to practice binary to decimal conversion in SAQ 2.

SAQ 2 – Conversion of binary numbers to decimal numbers

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

b) Convert 1011.101 into its decimal equivalent.

Let us now explain how to convert a decimal number into a binary number.

6.2.2 Decimal to Binary Conversion

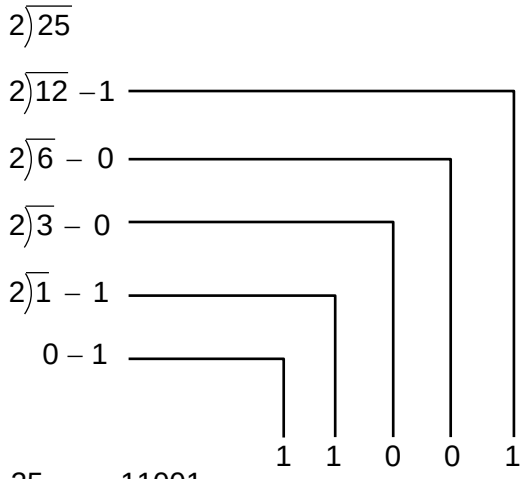
A decimal number is converted into its binary equivalent by dividing it repeatedly by 2. The division is continued till we get a quotient of 0. Then all the remainders are arranged sequentially with first remainder taking the position of LSB and the last one taking the position of MSB. Let us explain the conversion step by step.

Let us consider the conversion of 25_{10} into its binary equivalent. We divide the number and then the successive quotients by 2, keeping track of the remainders. We get the binary number by writing the first remainder in the position of LSB and the last remainder in the position of MSB.

Division by 2	Quotient	Remainder
25 ÷ 2	12	1 → LSB
12 ÷ 2	6	0
6 ÷ 2	3	0
3 ÷ 2	1	1
1 ÷ 2	0	1 → MSB

Hence, $25_{10} = 11001_2$

We present the same method pictorially as follows:



Thus, $25_{10} = 11001_2$

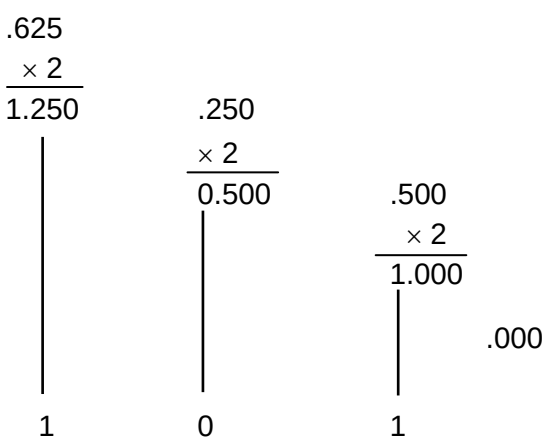
You may ask: If the decimal number also has some digits on the right of the decimal point, then how is it converted to a binary number? This part of the decimal number has to be treated separately. We **multiply** this part repeatedly by 2. After first multiplication by 2, either 1 or 0 will appear on the left of the decimal point. We keep this 1 or 0 separately and do not multiply it by 2 subsequently. This is followed for every multiplication. We continue multiplication by 2 till we get all 0s after the decimal point or up to the level of the accuracy desired. This will be clear from the following example.

EXAMPLE 6.2 : DECIMAL TO BINARY CONVERSION

Convert the following decimal numbers into equivalent binary numbers:

25.625_{10}, 58.0625_{10}

SOLUTION ■ We have already converted 25 into its binary equivalent 11001_2. Let us now convert .625 for which we multiply it by 2 repeatedly as follows:



Thus, $25.625_{10} = 11001.101_2$

To convert 58.0625 into a binary number, we split it into two parts: 58 and .0625 and convert each one of them into binary number separately as described just now.

Division by 2	Quotient	Remainder
$2\overline{)58}$,	29	0
$2\overline{)29}$,	14	1
$2\overline{)14}$,	7	0
$2\overline{)7}$,	3	1
$2\overline{)3}$,	1	1
$2\overline{)1}$,	0	1

LSB

MSB

Thus, $58_{10} = 111010_2$

Now we take up the conversion of .0625:

$$\begin{array}{r} .0625 \\ \times\ 2 \\ \hline 0.1250 \end{array} \qquad \begin{array}{r} .1250 \\ \times\ 2 \\ \hline 0.2500 \end{array} \qquad \begin{array}{r} .2500 \\ \times\ 2 \\ \hline 0.5000 \end{array} \qquad \begin{array}{r} .5000 \\ \times\ 2 \\ \hline 1.000 \end{array}$$

0 0 0 1

Thus, $58.0625_{10} = 111010.0001_2$

SAQ 3 – Conversion of decimal numbers to binary numbers

Convert the decimal number 37.75 into its binary equivalent.

While doing the inter-conversions in the binary and decimal systems, were you not thinking how tedious the calculation was? This is particularly so since binary numbers often consist of a large string of 0's and 1's. Can you imagine the length of the binary equivalent of a 10 digit decimal number? Therefore, convenient shorthand forms for representing the binary numbers were developed such as the **octal system** and **hexadecimal system**. With these

You may be wondering why we have chosen base 8 here (octal means 8) since we have to eventually depict the numbers as 0 or 1 in computers. We choose the base 8 so that the numbers can be expressed in strings of '1'. So, we use bases like 8 (groups of 3 bits 000 to 111) or 16 (groups of bits 0000 to 1111).

number systems, long strings of 0's and 1's can be reduced to a manageable form. Let us explain what these systems are.

6.3 OCTAL NUMBER SYSTEM

The octal number system has base 8, that is, there are 8 digits in this system: 0, 1, 2, 3, 4, 5, 6 and 7. Octal number system does not include the decimal digits 8 and 9. The weight of each octal digit is some power of 8 depending upon the position of the digit. This is explained in Fig. 6.3 for the octal number 10627.453.

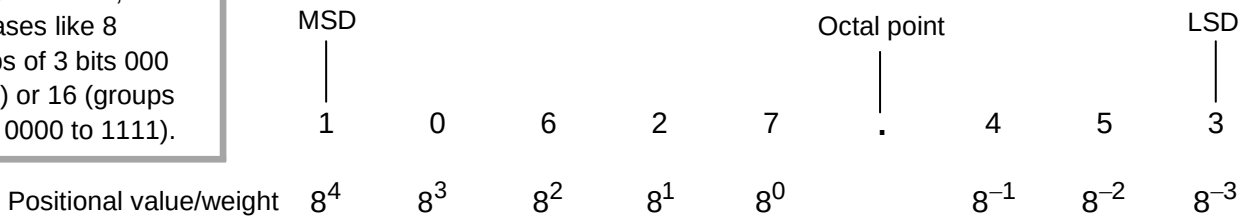


Fig. 6.3: **Octal number with positional value (weight) of each digit.**

Now let us see how counting is done in octal system. You are familiar with the counting in decimal system. In decimal system there are 10 digits from 0 to 9. Hence, the counting in such a system is done as in Table 6.4.

Table 6.4: Counting in decimal system

0	10	20
1	11	21
2	12	22
3	13	23
4	14	24
5	15	25
6	16	26
7	17	27
8	18	28
9	19	29…..

In octal system, we have 0 to 7. Hence, in octal counting, we jump to next digit after 7, instead of after 9 which is the case in the decimal system. Counting in the octal system is shown in Table 6.5.

Table 6.5: 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

You must have noticed in Table 6.5 that we have jumped from 7 to 10, and from 77 to 100, in octal counting. In 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 $4096 - 1 = 4095_{10}$.

Now answer SAQ 4 to check if you have understood this point.

SAQ 4 – Octal number system

a) Can the number 128.96 be an octal number?

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

Now we explain the conversion of octal to decimal numbers and vice-versa.

6.3.1 Octal to Decimal and Decimal to Octal Conversions

We can convert an octal number into its decimal equivalent in the same way as we convert a binary number. We **multiply the octal digit by its positional value**, i.e., **weight**. For example,

$$36.4_8 = (3 \times 8^1) + (6 \times 8^0) + (4 \times 8^{-1})$$

$$= 24 + 6 + 0.5 = 30.5_{10}$$

For converting a decimal number into an equivalent octal number, we divide it repeatedly by 8. This method is similar to the one for 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 the same as we followed for decimal to equivalent binary conversion. Let us explain these conversions with the help of an example.

EXAMPLE 6.3 : CONVERSIONS – OCTAL AND DECIMAL

a) Convert the octal number 126.25_8 into its equivalent decimal number.

b) Convert the decimal number 126.38_{10} into its equivalent octal number.

SOLUTION ■ We follow the method outlined just before this example:

a) $126.25_8 = (1 \times 8^2) + (2 \times 8^1) + (6 \times 8^0) + (2 \times 8^{-1}) + (5 \times 8^{-2})$

$$= 64 + 16 + 6 + 0.25 + 0.078 = 86.328_{10}$$

b) We split the number 126.38 into two parts:126 and 0.38, and proceed as we did for decimal to binary conversion:

Division by 8	Quotient	Remainder
$8\overline{)126}$	15	6 LSB
$8\overline{)15}$	1	7
$8\overline{)1}$	0	1 MSB

Alternatively, we can write:

$8\overline{)126}$

$8\overline{)15} \rightarrow 6$

$8\overline{)1} \rightarrow 7$

$8\overline{)0} \rightarrow 1$

1 7 6

Thus, $126_{10} = 176_8$

Next, we convert .38 into octal equivalent as follows:

$$.38 \times 8 = 3.04$$

$$.04 \times 8 = 0.32$$

$$.32 \times 8 = 2.56$$

$$.56 \times 8 = 4.48$$

.48

3 0 2 4

We need not keep multiplying and can stop the process at the desired accuracy like we have at the 4th digit.

Thus, $126.38_{10} = 176.3024_8$

Notice the similarity in the methods of conversion of decimal numbers to binary and octal numbers. But there is an important difference: the base is 2 for the binary system, and 8 for the octal system. So, for the binary system, divisions and multiplications involve 2, whereas for the octal system, 8 is

involved. If you practice, you will be able to convert these numbers in no time. Try SAQ 5.

SAQ 5 – Inter-conversions of octal and decimal numbers

a) What is the decimal equivalent of 37.2_8?

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

6.3.2 Octal to Binary and Binary to Octal Conversions

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 3-bit binary numbers**. The binary equivalent of each octal digit is shown in Table 6.6. The main advantage of the octal number system is the ease with which any octal number can be converted into its binary equivalent.

Table 6.6: 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 Table 6.6, we convert any octal number into its binary equivalent by simply replacing each octal digit by a 3-bit binary number. For example, the conversion of 567_8 is:

$$567_8 = 101\ 110\ 111$$

Thus, $567_8 = 101110111_2$

A binary number can be converted into its octal equivalent by first making groups of 3-bits starting from the LSB side. If the group on MSB side does not have 3 bits, then we add 0s on the left to make the last group of 3 bits. Next, we replace each group of 3 bits by its octal equivalent, and convert the binary number into octal equivalent. Let us consider an example to show these inter-conversions.

EXAMPLE 6.4 : CONVERSIONS: OCTAL AND BINARY

a) Convert the octal number 627.27_8 into its binary equivalent.

b) Convert the binary number 1100011001_2 into its octal equivalent.

SOLUTION ■ Following the procedure explained before this example, we use Table 6.5 and write:

a) $627.27_8 = 110\ 010\ 111\ .010\ 111 = 110010111.010111_2$

b) We first make groups of 3 bits starting from LSB:

$1100011001_2 = 1\ 100\ 011\ 001$

Since the MSB side does not have 3 bits, we add two 0's to the left of the digit to make the last group of 3 bits and use Table 6.5:

$1100011001_2 = 001\ 100\ 011\ 001$

$= 1\quad 4\quad 3\quad 1$

Thus, $1100011001_2 = 1431_8$

Try SAQ 6 for practice.

SAQ 6 – Inter-conversions of octal and binary numbers

a) Write the binary equivalent of 10027.12_8.

b) Write the octal equivalent of 10010_2.

Do you see how much easier it is to convert octal numbers to binary numbers and vice-versa? Let us now explain the hexadecimal or hex number system.

6.4 HEXADECIMAL (HEX) 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. Hexadecimal is popularly known as hex number. Each hex has a positional value that is some power of 16 depending upon its position in the number. This is illustrated in Fig. 6.4 for 124AE.B9.

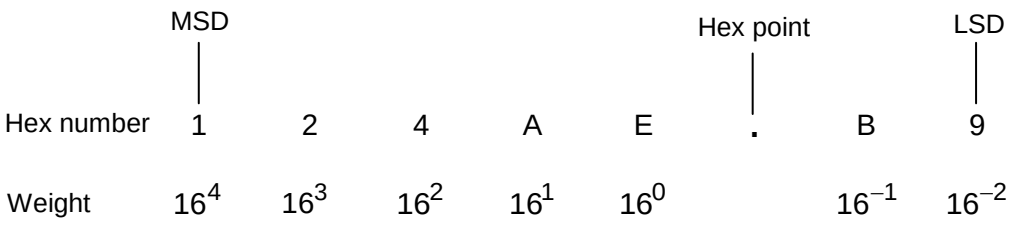


Fig. 6.4: Hexadecimal number showing positional values (weightS) of each digit.

The relationship of hex digits with decimal and binary numbers is given in Table 6.7. Note that to represent the largest hex digit 15, we require four binary bits. Therefore, **the binary equivalents of all the hex digits have to be written in 4-bit numbers**. A hexadecimal number is written with a suffix 16 or H, e.g., $F012_{16}$ or $F012_H$.

Table 6.7: Binary and decimal equivalents 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

In the hex number system, if n is the number of hex digits then counting can be done up to 16^n counts and the largest decimal number represented by a hex number is $16^n - 1$. The hex counting is shown in Table 6.8.

Table 6.8: 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	101
2	12	22	32	42	….	92	A2	B2	C2	D2	E2	F2	102
3	13	23	33	43	….	93	A3	B3	C3	D3	E3	F3	103
.	.	.	.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.	.	.	.
9	19	29	39	49	….	99	A9	B9	C9	D9	E9	F9	109
A	1A	2A	3A	4A	….	9A	AA	BA	CA	DA	EA	FA	10A
B	1B	2B	3B	4B	….	9B	AB	BB	CB	DB	EB	FB	10B
C	1C	2C	3C	4C	….	9C	AC	BC	CC	DC	EC	FC	10C
D	1D	2D	3D	4D	….	9D	AD	BD	CD	DD	ED	FD	10D
E	1E	2E	3E	4E	….	9E	AE	BE	CE	DE	EE	FE	10E
F	1F	2F	3F	4F	….	9F	AF	BF	CF	DF	EF	FF	10F

SAQ 7 – Hexadecimal numbers

In Table 6.8, what will the number next to $835F_{16}$ be? What is the largest decimal number represented by a 3-digit hex number?

6.4.1 Hex to Decimal and Decimal to Hex Conversions

Hex to decimal conversion is done in the same way as the 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.

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 on the right of the decimal point is to be repeatedly multiplied by 16. Let us demonstrate these conversions with the help of an example.

EXAMPLE 6.5 : INTERCONVERSIONS – HEX AND DECIMAL

a) Convert the hex numbers $514.AF_{16}$ and $3BE.1A_{16}$ to equivalent decimal numbers.

b) Convert the decimal number 579.26 to equivalent hex number.

SOLUTION ■ For hex to decimal conversion of part (a), we follow the method of binary/octal to decimal conversion. For decimal to hex conversion of part (b), we follow the procedure for decimal to binary/octal conversion with base 16.

a) $514.AF_{16} = (5 \times 16^2) + (1 \times 16^1) + (4 \times 16^0) + (10 \times 16^{-1}) + (15 \times 16^{-2})$

$= 1280 + 16 + 4 + 0.625 + 0.0586 = 1300.6836_{10}$

Similarly,

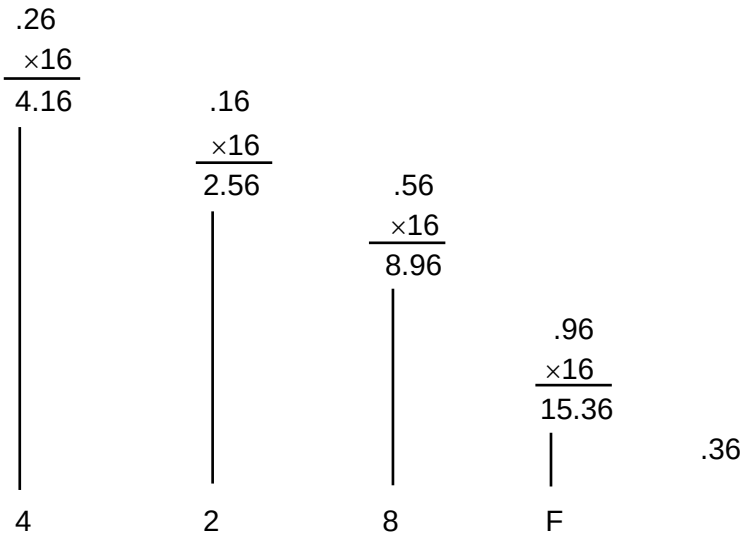
$3BE.1A_{16} = (3 \times 16^2) + (11 \times 16^1) + (14 \times 16^0) + (1 \times 16^{-1}) + (10 \times 16^{-2})$

$= 768 + 176 + 14 + 0.0625 + 0.0391 = 958.1016_{10}$

b) First we split the number 579.26 into two numbers: 579 and .26. Then we follow the procedure of Example 6.3 for hex numbers: 579 is converted into hex number as follows:

Division by 16	Quotient	Remainder
$16\overline{)579}$	36	3
$16\overline{)36}$	2	4
$16\overline{)2}$	0	2

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



We need not keep multiplying and can stop the process at the desired accuracy like we have at the 4th digit.

Thus, $579.26_{10} = 243.428F_{16}$

SAQ 8 – Inter-conversions of hex and decimal numbers

a) What is decimal equivalent of the hex number $1BE2_{16}$?

b) What is the hex equivalent of the decimal number 37_{10}?

6.4.2 Hex to Binary and Binary to Hex Conversions

A hex number is converted into its binary equivalent by replacing each hex digit by its equivalent 4-bit binary number.

For converting a binary number to its hex equivalent, we follow the reverse of the process for hex to binary conversion.

So, starting from the LSB side, we group the binary number bits into **groups of four bits**. If towards the MSB side, the number of bits is less than four, then we add zeros on the left of the MSB so that the group of four bits is complete. Finally, we replace each group by its equivalent hex digit. We show these conversions in the following example.

EXAMPLE 6.6 : INTERCONVERSIONS – HEX AND BINARY

a) Convert the hex number $BA6_{16}$ into its binary equivalent.

b) Convert the binary number 1001101110_2 into its hex equivalent.

SOLUTION ■ We use Table 6.7 and follow the procedure explained in the text above this example for both parts.

a) $BA6_{16}$ = B A 6
= 1011 1010 0110
= 101110100110_2

b) 1001101110_2 = 0010 0110 1110
= 2 6 E = $26E_{16}$

SAQ 9 – Inter-conversions of hex and binary numbers

a) What is binary equivalent of the hex number $6F10_{16}$?

b) What is the hex equivalent of the binary number 11001010100111_2?

6.4.3 Hex to Octal and Octal to Hex Conversions

Follow the steps given below to **convert a hex number to an octal number**:

1. Convert each digit of the hex number to its equivalent four bit binary number.
2. Then group the bits of the equivalent binary number into groups of three bits.
3. Then replace each group by its equivalent octal digit to get the octal number.

For converting a hex number to an octal number, just reverse the process described above. Let us show these conversions with the help of an example.

EXAMPLE 6.7 : CONVERSIONS – HEX AND OCTAL

a) Convert the hex number $5AF_{16}$ into its equivalent octal number.

b) Convert the octal number 5457_8 into its equivalent hex number.

c) Show that the hex number $3C_{16}$, the binary number 111100_2 and the decimal number 60_{10} are equivalent.

SOLUTION ■ We use the procedure explained above and Tables 6.6 and 6.7 for the conversions.

a) $5AF_{16}$ = 0101 1010 1111 = 010110101111
= 010 110 101 111 = 2 6 5 7

Thus, $5AF_{16}$ = 2657_8

b) 5457_8 = 101 100 101 111 = 1011 0010 1111
= B 2 F

Thus, 5457_8 = $B2F_{16}$

c) $3C_{16}$ is a hex number. Let us convert it to a decimal number using Table 6.7.

$$3C_{16} = 3 \times 16^1 + C \times 16^0$$

$$= 3 \times 16^1 + 12 \times 16^0 = 48 + 12 = 60_{10}$$

Let us convert it to a binary number:

$$3C_{16} = 0011 \quad 1100 = 111100_2$$

Let us convert this binary number to an equivalent decimal number:

$$111100_2 = 1 \times 2^5 + 1 \times 2^4 + 1 \times 2^3 + 1 \times 2^2 = 60_{10}$$

Hence, $3C_{16} = 60_{10} = 111100_2$ and these numbers are equivalent.

SAQ 10 – Inter-conversions of hex and octal numbers

a) What is the octal equivalent of the hex number $5A9_{16}$?

b) What is the hex equivalent of the octal number 327_8?

So far, you have learnt about binary, octal and hexadecimal number systems used in digital electronics. We need specific codes to input and retrieve information in any electronic device. You will learn about two such codes in Sec. 6.5: BCD code and ASCII code.

6.5 CODES

What have you learnt so far? Let us express it in a concise form. You have learnt that 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:

$$N_{10} = N_m \times B^m + N_{m-1} \times B^{m-1} + ... + N_3 \times B^3 + N_2 \times B^2 + N_1 \times B^1 + N_0 \times B^0 \quad (6.3)$$

You have also observed that a number in any system can be written in the binary form. By definition, 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. (6.3). But there are other relationships or codes that relate decimal numbers and groups of binary that do not obey Eq. (6.3). These relationships are called **codes**. We will now discuss two important codes used in digital technology.

6.5.1 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. A string of 4 bits is also called a **nibble**. For example, we use Table 6.2 and write the number 951_{10} in the decimal system in BCD code as:

951_{10} = 1001 0101 0001

= 100101010001_{BCD}

Let us write the BCD code for another decimal number, 2378:

2378_{10} = 0010 0011 0111 1000 = 0010001101111000_{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 0101 1000 = 000101011000_{BCD}

= 10011110_2 (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 up to 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 0101 0111 0010

= 1 5 7 2

= 1572_{10}

BCD codes are useful when we need to transfer information to or from a digital system. For example, when we enter a decimal number in the calculator, it is taken in as a BCD number. Then it is converted into a binary number and processed. The result is converted from binary number to its equivalent BCD code and displayed as the answer on the display screen. The same happens in digital clocks, digital meters, electronic counters, etc. However, BCD numbers are of little use in computers.

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, other symbols, and data of other kinds. Computers operate by coding letters of the alphabet, other symbols, and data into binary form. This needs an **alphanumeric code** for the input-output units of a computer rather than a number code. The code used for this purpose is the ASCII code about which you will study now. But before that, try SAQ 11.

SAQ 11 – BCD Code

What is the BCD code of 187_{10}?

6.5.2 ASCII Code

The word ASCII is the short form of **American Standard Code for Information Interchange**. This is the most widely used alphanumeric code in

computers. The alphanumeric code is one that represents alphabets, numerical numbers, punctuation marks and other special characters recognized by a computer. The ASCII code is a 7-bit code representing 26 English alphabets, digits 0 through 9, 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 6.9. You should read the table as you would read a graph.

Table 6.9: A few ASCII codes for numbers, alphabets and other 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

From Table 6.9, the ASCII code for any character is expressed as: $A_6A_5A_4A_3A_2A_1A_0$. For example, A has $A_6A_5A_4$ of 100 and an $A_3A_2A_1A_0$ of 0001. Therefore, its ASCII code is

$$100 \quad 0001 = \text{A}$$

The ASCII code for a is 1100001. You may like to read Table 6.9 yourself and practice writing the ASCII code.

SAQ 12 – ASCII Code

What is the ASCII code of SHARMA?

Digital computers can perform arithmetic operations using only binary numbers. Therefore, you also need to learn **binary arithmetic**. This means that you should know how to add, subtract, multiply or divide binary numbers.

6.6 BINARY ARITHMETIC

In this section, you 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 the binary system.

6.6.1 Addition

Let us first quickly recall how we add decimal numbers to establish a method for binary numbers. When we add any two numbers, say, 563 and 146, we start by adding the digits in the least significant column, i.e., the ones, as follows:

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

Next, we add the digits of the second column (the tens) and get,

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

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

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

We add binary numbers column-wise using carry in the same way. But before we do this, we need to discuss four simple rules.

We know in the decimal number system, 3 + 6 = 9 symbolizes the combining of 3 with 6 to get a total of 9. What happens with binary numbers?

Let us now state the **four rules for binary addition**:

Rule 1: 0 + 0 = 0

Rule 2: 0 + 1 = 1

Rule 3: 1 + 0 = 1

Rule 4: 1 + 1 = 10

You must practice these rules thoroughly because these are different from the rules for addition of decimal numbers. To add large binary numbers, we carry into higher-order columns as is done with decimal numbers.

Let us illustrate binary addition in Example 6.8.

EXAMPLE 6.8 : BINARY ADDITION

Add the following binary numbers:

a) 10 and 10,

b) 1, 1 and 1,

c) 101 and 110,

d) 110 and 111,

e) 101.011 and 111.110

SOLUTION ■ We use the 4 rules of binary addition explained in this section to add these binary numbers.

a) To add 10 to 10 we proceed as shown below:

```
  10
+ 10
 100
```

In the first column, 0 plus 0 is 0. In the second column, 1 plus 1 is 10. So, we write 0 with a carry of 1, which is 10.

b) To calculate 1 + 1 + 1, we first add two of the 1' s to get 10 and write 1 + 1 + 1 = 10 + 1. Next, we add the digits 0 and 1 on the right (least significant digits) to get 1, Thus, we get:

```
 10
+ 1
 11
```

We calculate 101 + 110 as follows:

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

c) To add 110 to 111 we proceed as shown below:

```
  110     first column:      0 + 1 = 1   (no carry)
+ 111     second column: 1 + 1 = 10 (zero, carry 1)
 1101     third column:      1 + 1 + 1 = 11
                                     \
                                     carry
```

d) We calculate 101.011 + 111.110 as follows:

$$\begin{array}{r} 101.011 \\ +\ 111.110 \\ \hline 1101.001 \end{array}$$

first column: 1 + 0 = 1

second column: 1 + 1 = 10 (zero, carry 1)

third column: carry 1 + 1 = 10 (zero, carry 1)

fourth column: 1 + 1 + 1 = 11 (1, carry 1)

fifth column: carry 1 + 1 =0 (zero, carry 1)

sixth column: 1 + 1 + 1 = 11

carry

Thus, the sum of 101.011 and 111.110 is 1101.001.

In all digital networks or computers, only two numbers are added at a time. To add more than two numbers, first two numbers are added, then to their sum the third number is added, and so on. Therefore, we should not worry about the addition of more than two numbers. The computer can add numbers in a few microseconds or even less. Multiplication, division and subtraction are actually done by the computers by the process of addition. In the next section, we discuss subtraction of binary numbers. In Unit 9, you will learn another method of subtracting binary numbers called the **2's complement method**.

SAQ 13 – Binary addition

Add the binary numbers a) 1010 and 1101; b) 1011 and 1010.

6.6.2 Subtraction

Binary subtraction is done in the same way as in the decimal system. Let us recall decimal subtraction. Consider, for example, 56 − 49. You know that

$$\begin{array}{r} 56 \\ -\ 49 \\ \hline 7 \end{array}$$

As you know, in this example, a 1 is borrowed from the ten's position giving 16 in the least significant position. Then 16 − 9 = 7. Borrowing a 1 from the ten's position leaves 4 in place of 5. Then 4 − 4 = 0. We perform binary subtraction in the same way.

Let us now set out the rules for subtracting binary numbers. There are 4 rules stated ahead, which you must remember.

Rule 1: $0 - 0 = 0$

Rule 2: $1 - 0 = 1$

Rule 3: $1 - 1 = 0$

Rule 4: $10 - 1 = 1$

To subtract large binary numbers, we subtract column by column, borrowing from the adjacent column when necessary. Let us illustrate binary subtraction with the help of an example.

EXAMPLE 6.9 : BINARY SUBTRACTION

Subtract a) 101 from 111, b) 1010 from 1101.

SOLUTION ■ We use the 4 rules of binary subtraction explained in this section.

a) To subtract 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
```

b) To calculate 1101 – 1010, we proceed as follows:

```
 1101      first column:    1 – 0 = 1
– 1010     second column:   10 (after borrow) – 1 = 1
  0011     third column:    0 (after borrow) – 0 = 0
           fourth column:   1 – 1 = 0
```

Practice binary subtraction before studying the next section.

SAQ 14 – **Binary subtraction**

Subtract the binary number 100011 from 110011.

6.6.3 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 numbers has only four rules.

Rule 1: $0 \times 0 = 0$

Rule 2: $0 \times 1 = 0$

Rule 3: $1 \times 0 = 0$

Rule 4: $1 \times 1 = 1$

Let us illustrate binary multiplication with the help of an example.

EXAMPLE 6.10 : BINARY MULTIPLICATION

Multiply 1101 by 1001.

SOLUTION ■ We use the rules of binary multiplication explained in this section.

To multiply 1101 by 1001, we proceed as follows:

$$\begin{array}{r} 1101 \\ \times\ 1001 \\ \hline 1101 \\ 0000 \\ 0000 \\ 1101 \\ \hline 1110101 \end{array}$$

Note that in the beginning, we write the first partial product of 1101 with 1. Subsequently we write each partial product below the previous one and shift it one place towards left relative to the previous place. For example, 1101 by 0 in the second and third rows, and lastly 1101 by 1. Note that each partial product is shifted to the left by one place as we do in decimal multiplication. Then we sum the partial products.

However, **REMEMBER** that the digital circuits or **computers add only two binary numbers at a time**.

Therefore, to the sum of first two partial products is added the third partial product. To this sum is added the fourth partial product to give the final sum.

SAQ 15 – Binary multiplication

Multiply 10110 by 110.

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:

$$\begin{array}{r} 110 \\ 10\overline{)\ 1100} \\ -10 \\ \hline 10 \\ -10 \\ \hline 00 \end{array}$$

With this we end the discussion on number systems used in digital electronics, their interconversions and arithmetic operations on binary numbers. You should practice these for a better grasp.

Let us summarise what you have learnt in this unit.

6.7 SUMMARY

Concept	Description
Number systems	■ There are mainly four number systems, namely, **decimal**, **binary**, **octal** and **hexadecimal** which have 10, 2, 8, and 16 digits, respectively. In digital electronics, it is the ease in applications that decides which number system is used. Every computer uses two or more of these number systems simultaneously. A number belonging to any one of these number systems can be converted into its equivalent in the other number systems.. • The **binary number system** has only two digits: 0 and 1. A binary digit is called **bit**. • The octal number system has 8 digits: 0 through 7. • The hex number system has 16 digits: 0 through 9, A (10) through F (15).
Codes	■ It is possible to arrange sets of binary digits to represent numbers, letters of the alphabet or other information by using a given code. Two 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. It is used for display purposes 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 sufficient to describe the capital and small letters of the alphabet, digits, punctuation marks, and other symbols.
Binary arithmetic	■ The four fundamental binary arithmetic operations of addition, subtraction, multiplication and division are similar to the corresponding operations in the decimal system. The following rules apply: • **Binary addition** 1. 0 + 0 = 0 2. 0 + 1 = 1 3. 1 + 0 = 1 4. 1 + 1 = 10 and 1 must be carried over to next higher (more significant) bit. • **Binary subtraction** 1. 0 – 0 = 0 2. 1 – 0 = 1 with borrow 1 from the next more significant bit 3. 1 – 1 = 0 4. 10 – 1 = 1

- **Binary multiplication**

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

6.8 TERMINAL QUESTIONS

1. In the binary sequence, what number follows 10111?
2. What is the largest decimal number that can be expressed by 6 bits?
3. Convert 1100101_2 into its decimal equivalent.
4. Convert 11011011010.1101_2 into its decimal equivalent.
5. Convert 372.125_{10} into its binary equivalent.
6. Convert 9.875_{10} into its binary equivalent.
7. What is the largest decimal number represented by a five digit octal number?
8. Convert 7777_8 into its decimal equivalent.
9. Convert 6789_{10} into its octal equivalent.
10. Convert 23401_8 into its binary equivalent.
11. Convert 1100110111001010_2 into its octal equivalent.
12. Add the binary numbers 1110001 and 1010101.
13. Multiply the binary numbers 101.1 and 11.01.
14. Divide the binary number 11011 by the binary number 100.

6.9 SOLUTIONS AND ANSWERS

Self-Assessment Questions

1. $11 = 1 \times 2^1 + 1 \times 2^0 = 3$

   $101 = 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 = 5$

   $110 = 1 \times 2^2 + 1 \times 2^1 + 0 \times 2^0 = 6$

   $111 = 1 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 = 7$

   $1000 = 1 \times 2^3 + 0 \times 2^2 + 0 \times 2^1 + 0 \times 2^0 = 8$

2. a) Largest decimal number is $2^n - 1$.

With $n = 10$, $2^{10} - 1 = 1024 - 1 = 1023_{10}$

b) 11.625_{10}

3. 100101.11

4. a) No. Octal numbers do not have digits 8 and 9.

b) The largest decimal number is $8^3 - 1 = 512 - 1 = 511_{10}$

5. a) 31.25_{10}

b) 17.2_8

6. a) 001000000010111.001010_2

b) 22_8

7. a) 8360_{16}

b) Largest decimal number is $16^3 - 1 = 4096 - 1 = 4095_{10}$

8. a) $1BE2_{16} = 1 \times 16^3 + 11 \times 16^2 + 14 \times 16^1 + 2 \times 16^0$

$= 4096 + 2816 + 224 + 2 = 7138_{10}$

b) 25_{16}

9. a) $6F10_{16}$ = 0110 1111 0001 0000 = 0110111100010000_2

b) 11001010100111_2 = 0011 0010 1010 0111

= $32A7_{16}$

10. a) $5A9_{16}$ = 0101 1010 1001

= 010 110 101 001

= 2651_8

b) 327_8 = 011 010 111 = 0 1101 0111

= $D7_{16}$

11. BCD code of 187_{10} :

187_{10} = 0001 1000 0111 = 000110000111_{BCD}

12. ASCII code of SHARMA:

SHARMA = 1010011 1001000 1000001 1010010 1001101 1000001

13. a) 10111 b) 10101

14. 10000

15. 10000100

Terminal Questions

1. 11000_2
2. 63_{10}
3. 101_{10}
4. 1754.8125_{10}
5. 101110100.001_2
6. 1001.111_2
7. 32767_{10}
8. 4095_{10}
9. 15205_8
10. 100111000000001_2
11. 1100110111001010_2 = 001 100 110 111 001 010
    = 1 4 6 7 1 2
    = 146712_8
12. 11000110
13. 10001.111
14. 110.11