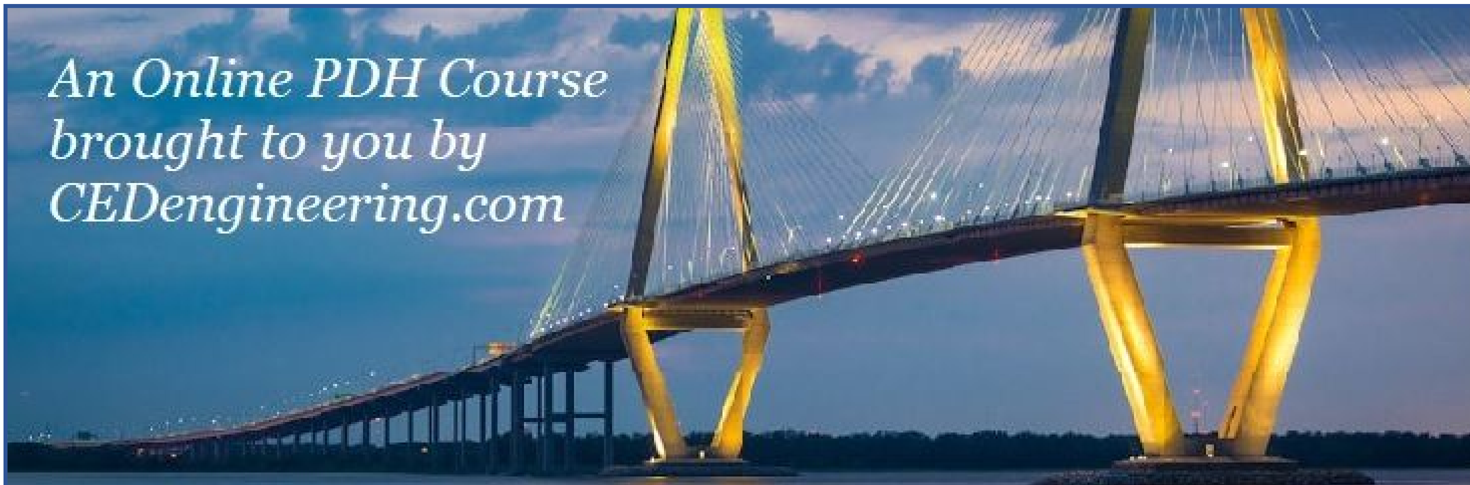


Number Systems

Course No: E04-029
Credit: 4 PDH

Jeffery Cwalinski, P. E.

Continuing Education and Development, Inc.

P: (877) 322-5800
info@cedengineering.com

ERRATA SHEET

There are several editorial corrections to NAVEDTRA 14185A, Chapter 1. They are:

1. On page 1-5, for 10-1, 10-2, and 10-3 the plus sign, “+”, should be a division sign, “÷”.

2. On page 1-8, in the first paragraph, the MSD shown in the bar graph, which is circled, is 0. It should be 1.

3. On page 1-18, the first example, the carry should not be in the right column. The only carry there should be is the one brought down to the MSD of the sum.

4. On page 1-28 the number used for the second method currently is 001011011002. It should be 100101101002.

5. In the answer key for question Q34 the answer is 758 not 1048.

6. On page 1-43 the heading for Table 1-5, right most column currently reads “OCTAL”. It should read “HEX”.

7. On page 1-45, in the powers of 16 table, the second equation should have 16 raised to the second power rather than to the third power.

8. On page 1-50, 7E5E16 – 47116 = 79ED16, not 374416.

9. In the answer key for A87 thru A90 the “2” at the end of the binary numbers is the subscript representing the base and should be subscripted.

10. In Section 1.10.1, 2nd paragraph the “24” should be “24”.

11. On page 1-63 in section 1.6.3, the 10 at the end of 6310 should be a subscript because 63 base 10 is intended.

12. On page 1-69, the 16 at the end of .716 should be a subscript as .7 base 16 is intended.

13. On page 1-80 the example to convert 24.3688 to decimal did not include the LSD, the “8”. Including this digit adds 8 * 8-3 = 0.015625 to the answer. The answer is now 20.48437510.

14. On page 1-80, in both occurrences (Example and Solution) the 24.368 should be 24.368.

15. On page 1-80, 20.4687510 (in Solution) should 20.4687510.

16. In the bar graph at the bottom of page 1-81, 65.536 should be 65,536 instead and 4.096 should be 4,096 instead.

1 NUMBER SYSTEMS

LEARNING OBJECTIVES

After you finish this chapter, you should be able to do the following:
1. Recognize different types of number systems as they relate to computers.
2. Identify and define unit, number, base/radix, positional notation, and most and least significant digits as they relate to decimal, binary, octal, and hexadecimal number systems.
3. Add and subtract in binary, octal, and hexadecimal number systems.
4. Convert values from decimal, binary, octal, hexadecimal, and binary-coded decimal number systems to each other and back to the other systems.
5. Add in binary-coded decimal.

1.1 INTRODUCTION

How many days' leave do you have on the books? How much money do you have to last until payday? It doesn't matter what the question is if the answer is in dollars or days or cows, it will be represented by numbers.

Just try to imagine going through one day without using numbers. Some things can be easily described without using numbers, but others prove to be difficult. Look at the following examples:

I am stationed on the aircraft carrier *Nimitz*.

He owns a green Chevrolet.

The use of numbers wasn't necessary in the preceding statements, but the following examples depend on the use of numbers:

I have $25 to last until payday.

I want to take 14 days' leave.

You can see by these statements that numbers play an important part in our lives.

1.2 BACKGROUND AND HISTORY

Man's earliest number or counting system was probably developed to help determine how many possessions a person had. As daily activities became more complex, numbers became more important in trade, time, distance, and all other phases of human life.

As you have seen already, numbers are extremely important in your military and personal life. You realize that you need more than your fingers and toes to keep track of the numbers in your daily routine.

Ever since people discovered that it was necessary to count objects, they have been looking for easier ways to count them. The abacus, developed by the Chinese, is one of the earliest known calculators. It is still in use in some parts of the world.

Blaise Pascal (French) invented the first adding machine in 1642. Twenty years later, an Englishman, Sir Samuel Moreland, developed a more compact device that could multiply, add, and subtract. About 1672, Gottfried Wilhelm von Leibniz (German) perfected a machine that could perform all the basic operations (add, subtract, multiply, divide), as well as extract the square root. Modern electronic digital computers still use von Liebniz's principles.

1.3 MODERN USE

Computers are now employed wherever repeated calculations or the processing of huge amounts of data is needed. The greatest applications are found in the military, scientific, and commercial fields. They have applications that range from mail sorting, through engineering design, to the identification and destruction of enemy targets. The advantages of digital computers include speed, accuracy, and manpower savings. Often computers are able to take over routine jobs and release personnel for more important work that cannot be handled by a computer.

People and computers do not normally speak the same language. Methods of translating information into forms that are understandable and usable to both are necessary. Humans generally speak in words and numbers expressed in the decimal number system, while computers only understand coded electronic pulses that represent digital information.

In this chapter you will learn about number systems in general and about binary, octal, and hexadecimal (which we will refer to as hex) number systems specifically. Methods for converting numbers in the binary, octal, and hex systems to equivalent numbers in the decimal system (and vice versa) will also be described. You will see that these number systems can be easily converted to the electronic signals necessary for digital equipment.

1.4 TYPESOF NUMBER SYSTEMS

Until now, you have probably used only one number system, the decimal system. You may also be familiar with the Roman numeral system, even though you seldom use it.

1.4.1 The Decimal Number System

In this module you will be studying modern number systems. You should realize that these systems have certain things in common. These common terms will be defined using the decimal system as our base. Each term will be related to each number system as that number system is introduced.

Each of the number systems you will study is built around the following components: the UNIT, NUMBER, and BASE (RADIX).

Unit and Number

The terms *unit* and *number* when used with the decimal system are almost self-explanatory. By definition the unit is a single object; that is, an apple, a dollar, a day. A number is a symbol representing a unit or a quantity. The figures 0, 1, 2, and 3 through 9 are the symbols used in the decimal system. These symbols are called Arabic numerals or figures. Other symbols may be used for different number systems. For example, the symbols used with the Roman numeral system are letters - V is the symbol for 5, X for 10, M for 1,000, and so forth. We will use Arabic numerals and letters in the number system discussions in this chapter.

Base (Radix)

The base, or radix, of a number system tells you the number of symbols used in that system. The base of any system is always expressed in decimal numbers. The base, or radix, of the decimal system is 10. This means there are 10 symbols - 0, 1, 2, 3, 4, 5, 6, 7, 8, and 9 - used in the system. A number system using three symbols - 0, 1, and 2 - would be base 3; four symbols would be base 4; and so forth. Remember to count the zero or the symbol used for zero when determining the number of symbols used in a number system.

The base of a number system is indicated by a subscript (decimal number) following the value of the number. The following are examples of numerical values in different bases with the subscript to indicate the base:

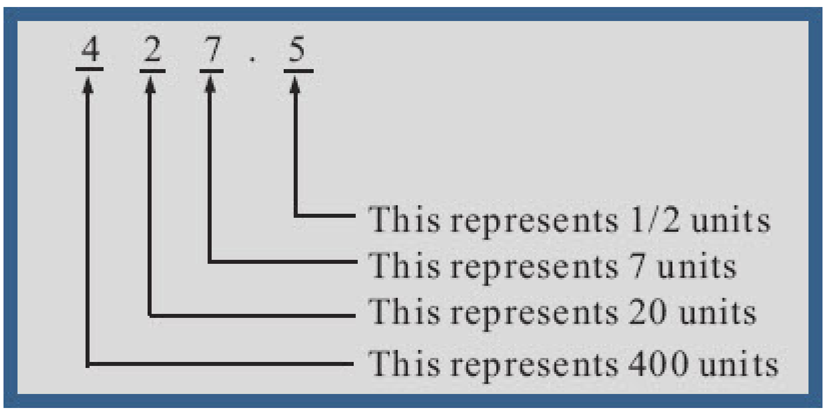
$$7592_{10} \quad 214_5 \quad 123_4 \quad 656_7$$

You should notice the highest value symbol used in a number system is always one less than the base of the system. In base 10 the largest value symbol possible is 9; in base 5 it is 4; in base 3 it is 2.

Positional Notation and Zero

You must observe two principles when counting or writing quantities or numerical values. They are the POSITIONAL NOTATION and the ZERO principles.

Positional notation is a system where the value of a number is defined not only by the symbol but by the symbol's position. Let's examine the decimal (base 10) value of 427.5. You know from experience that this value is four hundred twenty-seven and one-half. Now examine the position of each number:



If 427.5 is the quantity you wish to express, then each number must be in the position shown. If you exchange the positions of the 2 and the 7, then you change the value.

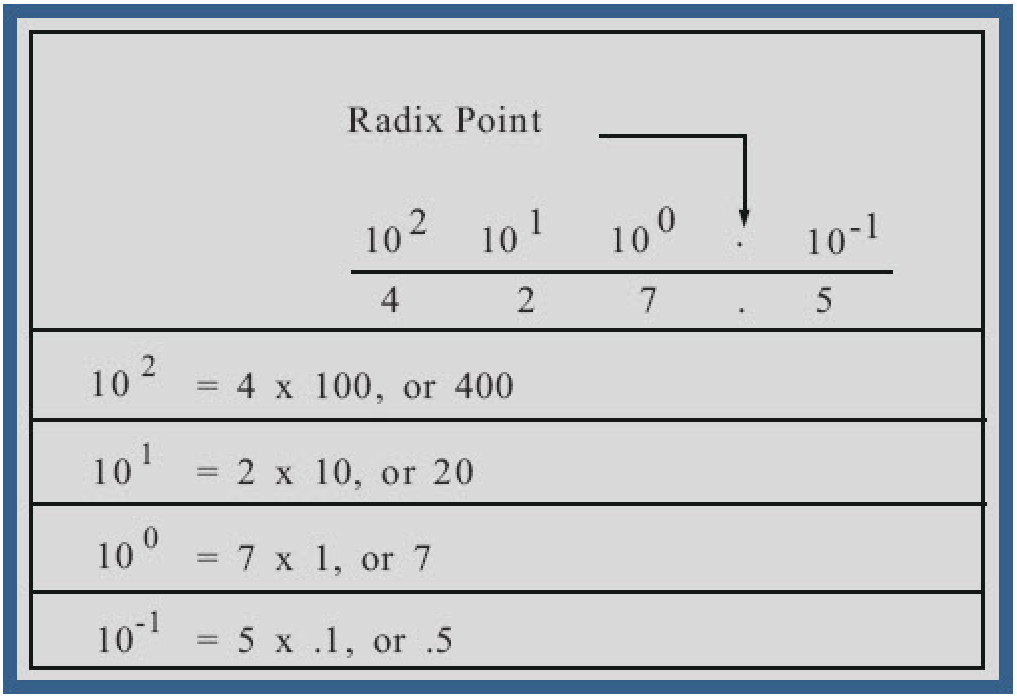
Each position in the positional notation system represents a power of the base, or radix. A POWER is the number of times a base is multiplied by itself. The power is written above and to the right of the base and is called an EXPONENT. Examine the following base 10 line graph:

Radix Point

10^3 10^2 10^1 10^0 . 10^{-1} 10^{-2} 10^{-3}

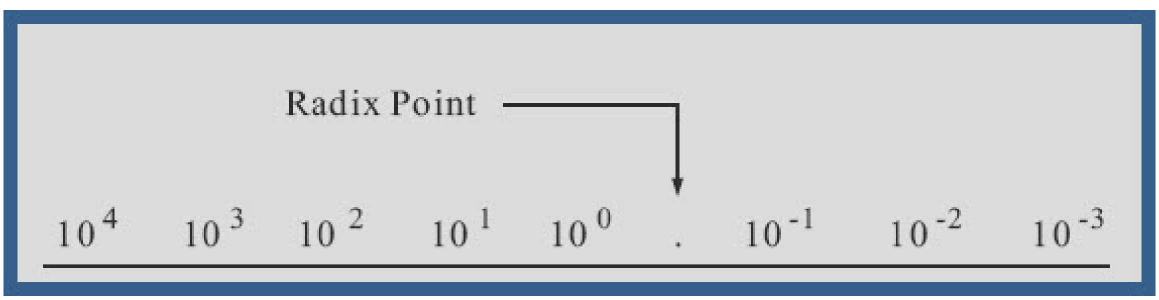
10^3	= 10 x 100, or 1000
10^2	= 10 x 10, or 100
10^1	= 10 x 1, or 10
10^0	= (any number raised to the power of 0 equals 1)
10^{-1}	= 1 + 10, or .1
10^{-2}	= 1 + 100. or .01
10^{-3}	= 1 + 1000, or .001

Now let's look at the value of the base 10 number 427.5 with the positional notation line graph:



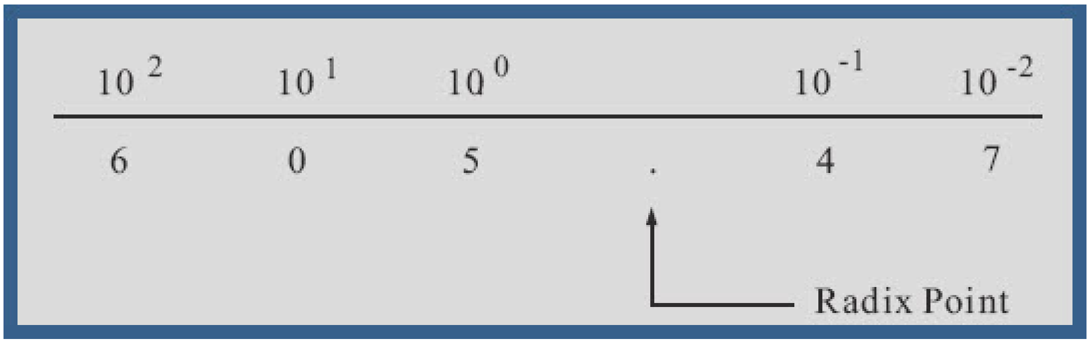
You can see that the power of the base is multiplied by the number in that position to determine the value for that position.

The following graph illustrates the progression of powers of 10:



All numbers to the left of the decimal point are whole numbers, and all numbers to the right of the decimal point are fractional numbers. A whole number is a symbol that represents one, or more, complete objects, such as one apple or $5. A fractional number is a symbol that represents a portion of an object, such as half of an apple (.5 apples) or a quarter of a dollar ($0.25). A mixed number represents one, or more, complete objects, and some portion of an object, such as one and a half apples (1.5 apples). When you use any base other than the decimal system, the division between whole numbers and fractional numbers is referred to as the RADIX POINT. The decimal point is actually the radix point of the decimal system, but the term radix point is normally not used with the base 10 number system.

Just as important as positional notation is the use of the zero. The placement of the zero in a number can have quite an effect on the value being represented. Sometimes a position in a number does not have a value between 1 and 9. Consider how this would affect your next paycheck. If you were expecting a check for $605.47, you wouldn't want it to be $65.47. Leaving out the zero in this case means a difference of $540.00. In the number 605.47, the zero indicates that there are no tens. If you place this value on a bar graph, you will see that there are no multiples of 10^1.



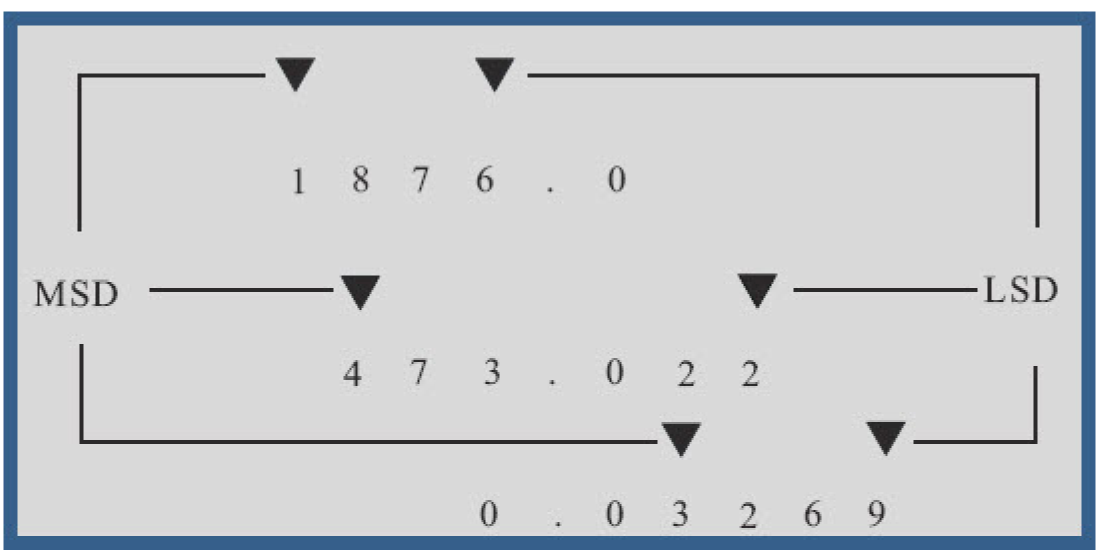
Most Significant Digit and Least Significant Digit (MSD and LSD)

Other important factors of number systems that you should recognize are the MOST SIGNIFICANT DIGIT (MSD) and the LEAST SIGNIFICANT DIGIT (LSD).

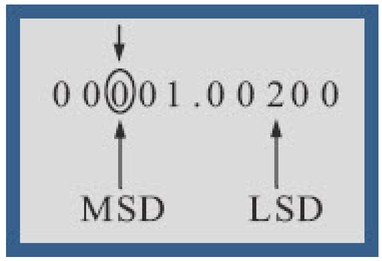
The MSD in a number is the digit that has the *greatest* effect on that number.

The LSD in a number is the digit that has the *least* effect on that number.

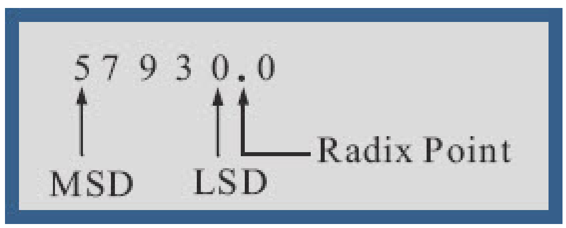
Look at the following examples:



You can easily see that a change in the MSD will increase or decrease the value of the number the greatest amount. Changes in the LSD will have the smallest effect on the value. The nonzero digit of a number that is the farthest LEFT is the MSD, and the nonzero digit farthest RIGHT is the LSD, as in the following example:



In a whole number the LSD will always be the digit immediately to the left of the radix point.



Q1. What term describes a single object?

Q2. A symbol that represents one or more objects is called a _________.

Q3. The symbols 0, 1, 2, and 3 through 9 are what type of numerals?

Q4. What does the base or radix of a number system tell you about the system?

Q5. How would you write one hundred seventy-three base 10?

Q6. What power of 10 is equal to 1,000? 100? 10? 1?

Q7. The decimal point of the base 10 number system is also known as the _________.

Q8. What is the MSD and LSD of the following numbers

> *(a) 420.*
> *(b) 1045.06*
> *(c) 0.0024*
> *(d) 247.0001*

Carry and Borrow Principles

Soon after you learned how to count, you were taught how to add and subtract. At that time, you learned some concepts that you use almost everyday. Those concepts will be reviewed using the decimal system. They will also be applied to the other number systems you will study.

ADDITION-Addition is a form of counting in which one quantity is added to another. The following definitions identify the basic terms of addition:

AUGEND-The quantity to which an addend is added

ADDEND-A number to be added to a preceding number

SUM -The result of an addition (the sum of 5 and 7 is 12)

CARRY -A carry is produced when the sum of two or more digits in a vertical column equals or exceeds the base of the number system in use

How do we handle the carry; that is, the two-digit number generated when a carry is produced? The lower order digit becomes the sum of the column being added; the higher order digit (the carry) is added to the next higher order column. For example, let's add 15 and 7 in the decimal system:

```
 1  Carry
 15 Augend
+7  Addend
---
 22 Sum
```

Starting with the first column, we find the sum of 5 and 7 is 12. The 2 becomes the sum of the lower order column and the 1 (the carry) is added to the upper order column. The sum of the upper order column is 2. The sum of 15 and 7 is, therefore, 22.

The rules for addition are basically the same regardless of the number system being used. Each number system, because it has a different number of digits, will have a unique digit addition table. These addition tables will be described during the discussion of the adding process for each number system.

A decimal addition table is shown in table 1-1. The numbers in row X and column Y may represent either the addend or the augend. If the numbers in X represent the augend, then the numbers in Y must represent the addend and vice versa. The sum of X + Y is located at the point in array Z where the selected X row and Y column intersect.

+	0	1	2	3	4	5	6	7	8	9
0	0	1	2	3	4	5	6	7	8	9
1	1	2	3	4	5	6	7	8	9	10
2	2	3	4	5	6	7	8	9	10	11
3	3	4	5	6	7	8	9	10	11	12
4	4	5	6	7	8	9	10	11	12	13
5	5	6	7	8	9	10	11	12	13	14
6	6	7	8	9	10	11	12	13	14	15
7	7	8	9	10	11	12	13	14	15	16
8	8	9	10	11	12	13	14	15	16	17
9	9	10	11	12	13	14	15	16	17	18

X

Z

Y

Table 1-1 Decimal Addition Table

To add 5 and 7 using the table, first locate one number in the X row and the other in the Y column. The point in field Z where the row and column intersect is the sum. In this case the sum is 12.

SUBTRACTION.-The following definitions identify the basic terms you will need to know to understand subtraction operations:

- SUBTRACT-To take away, as a part from the whole or one number from another
- MINUEND-The number from which another number is to be subtracted
- SUBTRAHEND-The quantity to be subtracted
- REMAINDER, or DIFFERENCE-That which is left after subtraction
- BORROW-To transfer a digit (equal to the base number) from the next higher order column for the purpose of subtraction.

Use the rules of subtraction and subtract 8 from 25. The form of this problem is probably familiar to you:

1 15 Carry
→ $\cancel{2}\cancel{5}$ Minuend
→ -8 Subtrahend
17 Difference

In addition,

$$X + Y = Z$$

In subtraction, the reverse is true; that is,

$$Z - Y = X$$

OR

$$Z - X = Y$$

Thus, in subtraction the minuend is always found in array Z and the subtrahend in either row X or column Y. If the subtrahend is in row X, then the remainder will be in column Y. Conversely, if the subtrahend is in column Y, then the difference will be in row X. For example, to subtract 8 from 15, find 8 in either the X row or Y column. Find where this row or column intersects with a value of 15 for Z; then move to the remaining row or column to find the difference.

1.4.2 The Binary Number System

The simplest possible number system is the BINARY, or base 2, system. You will be able to use the information just covered about the decimal system to easily relate the same terms to the binary system.

Unit and Number

The base, or radix-you should remember from our decimal section-is the number of symbols used in the number system. Since this is the base 2 system, only two symbols, 0 and 1, are used. The base is indicated by a subscript, as shown in the following example:

$$1_2$$

When you are working with the decimal system, you normally don't use the subscript. Now that you will be working with number systems other than the decimal system, it is important that you use the subscript so that you are sure of the system being referred to. Consider the following two numbers:

11	11

With no subscript you would assume both values were the same. If you add subscripts to indicate their base system, as shown below, then their values are quite different:

11_{10}	11_2

The base ten number 11_{10} is eleven, but the base two number 11_2 is only equal to three in base ten. There will be occasions when more than one number system will be discussed at the same time, so you MUST use the proper Subscript.

2^4 2^3 2^2 2^1 2^0 . 2^{-1} 2^{-2} 2^{-3}
2^4 is equal to 2 x 2 x 2 x 2, or 16_{10}
2^3 is equal to 2 x 2 x 2, or 8_{10}
2^2 is equal to 2 x 2, or 4_{10}
2^1 is equal to 2 x 1, or 2_{10}
2^0 is equal to 1_{10}
2^{-1} is equal to 1/2, or $.5_{10}$
2^{-2} is equal to 1/4, or $.25_{10}$
2^{-3} is equal to 1/8, or $.125_{10}$

All numbers or values to the left of the radix point are whole numbers, and all numbers to the right of the radix point are fractional numbers.

Let's look at the binary number 101.1 on a bar graph:

$$\begin{array}{cccccc} 2^2 & 2^1 & 2^0 & . & 2^{-2} \\ \hline 1 & 0 & 1 & . & 1 \end{array}$$

Working from the radix point to the right and left, you can determine the decimal equivalent:

$$\begin{array}{rcr} 1 \text{ x } 2^{-1} & = & .5_{10} \\ 1 \text{ x } 2^{0} & = & 1.0_{10} \\ 1 \text{ x } 2^{1} & = & 0.0_{10} \\ 1 \text{ x } 2^{2} & = & 4.0_{10} \\ \hline & & 5.5_{10} \end{array}$$

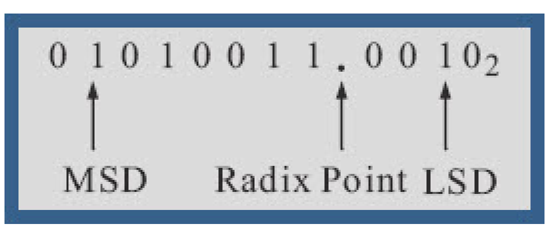
Table 1-2 provides a comparison of decimal and binary numbers. Notice that each time the total number of binary symbol positions increase, the binary number indicates the next higher power of 2. By this example, you can also see that more symbol positions are needed in the binary system to represent the equivalent value in the decimal system.

	DECIMAL	BINARY	
10^0	0	0	2^0
	1	1	
	2	10	2^1
	3	11	
	4	100	2^2
	5	101	
	6	110	
	7	111	
	8	1000	2^3
	9	1001	
10^1	10	1010	
	11	1011	
	12	1100	
	13	1101	
	14	1110	
	15	1111	
	16	10000	2^4
	17	10001	
	18	10010	
	19	10011	
	20	10100	

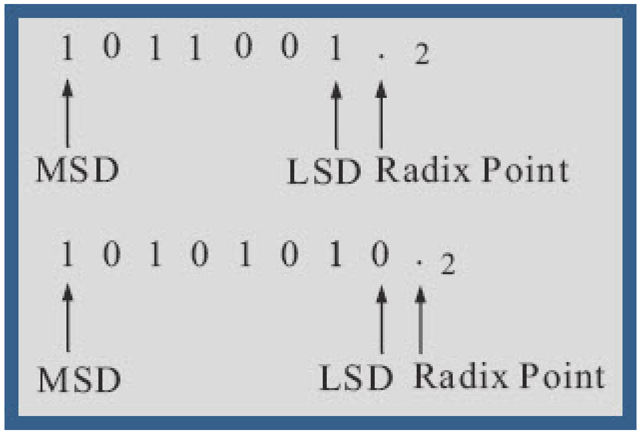
Table 1-2 Decimal and Binary Comparison

MSD and LSD

When you're determining the MSD and LSD for binary numbers, use the same guidelines you used with the decimal system. As you read from left to right, the first nonzero digit you encounter is the MSD, and the last nonzero digit is the LSD.



If the number is a whole number, then the first digit to the left of the radix point is the LSD.



Here, as in the decimal system, the MSD is the digit that will have the most effect on the number; the LSD is the digit that will have the least effect on the number.

The two numerals of the binary system (1 and 0) can easily be represented by many electrical or electronic devices. For example, 1_2 may be indicated when a device is active (on), and 0_2 may be indicated when a device is nonactive (off).

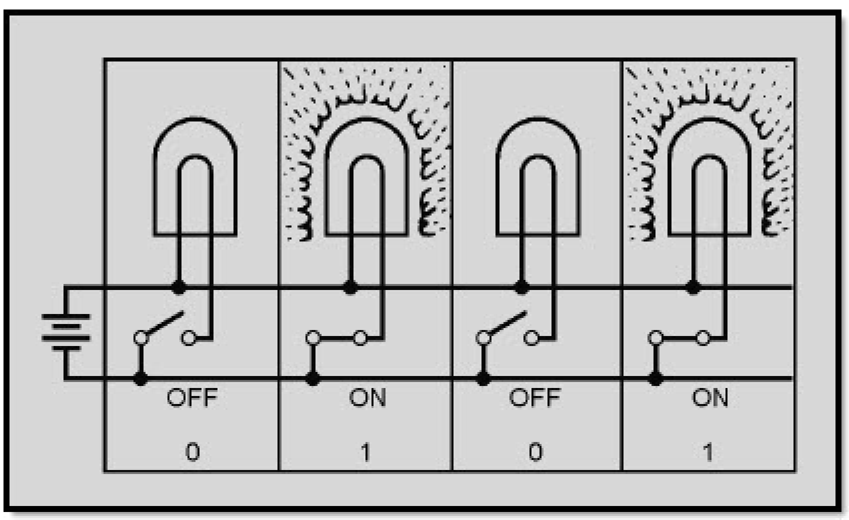


Figure 1-1 Binary Example

Look at the preceding figure. It illustrates a very simple binary counting device. Notice that 1_2 is indicated by a lighted lamp and 0_2 is indicated by an unlighted lamp. The reverse will work equally well. The unlighted state of the lamp can be used to represent a binary 1 condition, and the lighted state can represent the binary 0 condition. Both methods are used in digital computer applications. Many other devices are used to represent binary conditions. They include switches, relays, diodes, transistors, and integrated circuits (ICs).

Addition of Binary Numbers

Addition of binary numbers is basically the same as addition of decimal numbers. Each system has an augend, an addend, a sum, and carries. The following example will refresh your memory:

$$\begin{array}{rl} {}^{1} & \text{Carry} \\ 15 & \text{Augend} \\ \underline{+7} & \text{Addend} \\ 22 & \text{Sum} \end{array}$$

Since only two symbols, 0 and 1, are used with the binary system, only four combinations of addition are possible.

$$\begin{array}{c} 0 + 0 \\ 1 + 0 \\ 0 + 1 \\ 1 + 1 \end{array}$$

The sum of each of the first three combinations is obvious:

$$\begin{array}{c} 0 + 0 = 0_2 \\ 0 + 1 = 1_2 \\ 1 + 0 = 1_2 \end{array}$$

The fourth combination presents a different situation. The sum of 1 and 1 in any other number system is 2, but the numeral 2 does not exist in the binary system. Therefore, the sum of 1_2 and 1_2 is 10_2 (spoken as one zero base two), which is equal to 2_{10}.

$$\begin{array}{rl} {}^{1} & \text{Carry} \\ 1_2 & \text{Augend} \\ +\underline{1_2} & \text{Addend} \\ 10_2 & \text{Sum} \end{array}$$

Study the following examples using the four combinations mentioned above:

$$\begin{array}{rl} 101_2 & \text{Augend} \\ +\underline{010_2} & \text{Addend} \\ 111_2 & \text{Sum} \end{array}$$

$$\begin{array}{rl} {}^{1} & \text{Carry} \\ 101_2 & \text{Augend} \\ +\underline{101_2} & \text{Addend} \\ 1010_2 & \text{Sum} \end{array}$$

When a carry is produced, it is noted in the column of the next higher value or in the column immediately to the left of the one that produced the carry.

Example: Add 1011_2 and 1101_2.

Solution: Write out the problem as shown:

$$\begin{array}{rl} 1011_2 & \text{Augend} \\ +\underline{1101_2} & \text{Addend} \end{array}$$

As we noted previously, the sum of 1 and 1 is 2, which cannot be expressed as a single digit in the binary system. Therefore, the sum of 1 and 1 produces a carry:

$$\begin{array}{rrl} & 1 & \text{Carry} \\ & 1011_2 & \text{Augend} \\ + & \underline{1101_2} & \text{Addend} \\ & 0_2 & \text{Sum} \end{array}$$

The following steps, with the carry indicated, show the completion of the addition:

$$\begin{array}{rrl} & & \text{Previous Carry Used} \\ & 1\cancel{1} & \text{Carry} \\ & 1011_2 & \text{Augend} \\ + & \underline{1101_2} & \text{Addend} \\ & 00_2 & \end{array}$$

When the carry is added, it is marked through to prevent adding it twice.

$$\begin{array}{rrl} & & \text{Previous Carry Used} \\ & 1\cancel{1}\cancel{1} & \text{Carry} \\ & 1011_2 & \text{Augend} \\ + & \underline{1101_2} & \text{Addend} \\ & 000_2 & \end{array}$$

$$\begin{array}{rrl} & & \text{Previous Carry Used} \\ & 1\cancel{1}\cancel{1}\cancel{1} & \text{Carry} \\ & 1011_2 & \text{Augend} \\ + & \underline{1101_2} & \text{Addend} \\ & 11000_2 & \end{array}$$

In the final step the remaining carry is brought down to the sum.

In the following example you will see that more than one carry may be produced by a single column. This is something that does not occur in the decimal system.

Example: Add 1_2, 1_2, 1_2, and 1_2

$$\begin{array}{rl}
1_2 & \text{Augend} \\
12 & \text{1st Addend} \\
1^2 & \text{2nd Addend} \\
+\ \underline{1^2_2} & \text{3rd Addend}
\end{array}$$

The sum of the augend and the first addend is 0 with a carry. The sum of the second and third addends is also 0 with a carry. At this point the solution resembles the following example:

$$\begin{array}{rl}
1\ \ \ & \text{Carry} \\
1\ \ \ & \text{Carry} \\
1_2 & \text{Augend} \\
12 & \text{1st Addend} \\
1^2 & \text{2nd Addend} \\
+\ \underline{1^2_2} & \text{3rd Addend} \\
0_2 &
\end{array}$$

The sum of the carries is 0 with a carry, so the sum of the problem is as follows:

$$\begin{array}{rl}
1\ \ \ & \text{Carry} \\
11\ \ \ & \text{Carry} \\
1_2 & \text{Augend} \\
12 & \text{1st Addend} \\
1^2 & \text{2nd Addend} \\
+\ \underline{1^2_2} & \text{3rd Addend} \\
100_2 &
\end{array}$$

The same situation occurs in the following example:

Add 100_2, 101_2, and 111_2

$$\begin{array}{rl}
100_2 & \text{Augend} \\
101_2 & \text{Addend} \\
+\ 111_2 & \text{Addend} \\
\hline
\end{array}$$

$$\begin{array}{rl}
1 & \text{Carry} \\
100_2 & \text{Augend} \\
101_2 & \text{Addend} \\
+\ 111_2 & \text{Addend} \\
\hline
0_2 & \text{Sum}
\end{array}$$

$$\begin{array}{rl}
1\,1 & \text{Carry} \\
100_2 & \text{Augend} \\
101_2 & \text{Addend} \\
+\ 111_2 & \text{Addend} \\
\hline
00_2 & \text{Sum}
\end{array}$$

As in the previous example, the sum of the four 1s is 0 with two carries, and the sum of the two carries is 0 with one carry. The final solution will look like this:

$$\begin{array}{rl}
1 & \text{Carry} \\
1111 & \text{Carry} \\
100_2 & \text{Augend} \\
101_2 & \text{Addend} \\
+\ 111_2 & \text{Addend} \\
\hline
10000_2 & \text{Sum}
\end{array}$$

In the addition of binary numbers, you should remember the following binary addition rules:

Rule 1: $0_2 + 0_2 = 0_2$

Rule 2: $1_2 + 0_2 = 1_2$

Rule 3: $0_2 + 1_2 = 1_2$

Rule 4: $1_2 + 1_2 = 10_2$

Now practice what you've learned by solving the following problems:

Q9.

Add:

$$\begin{array}{r} 10101_2 \\ +\ 1010_2 \\ \hline \end{array}$$

Q10.

Add:

$$\begin{array}{r} 10011_2 \\ +\ 1010_2 \\ \hline \end{array}$$

Q11.

Add:

$$\begin{array}{r} 11101_2 \\ +\ 100_2 \\ \hline \end{array}$$

Q12.

Add:

$$\begin{array}{r} 10110_2 \\ +\ 11001_2 \\ \hline \end{array}$$

Q13.

Add:

$$\begin{array}{r} 111_2 \\ +\ 1_2 \\ \hline \end{array}$$

Q14.

Add:

$$\begin{array}{r} 1010010_2 \\ 1110111_2 \\ +\ \ 101012 \\ \hline \end{array}$$

Subtraction of Binary Numbers

Now that you are familiar with the addition of binary numbers, subtraction will be easy. The following are the four rules that you must observe when subtracting:

Rule 1: $0_2 - 0_2 = 0_2$

Rule 2: $1_2 - 0_2 = 1_2$

Rule 3: $1_2 - 1_2 = 0_2$

Rule 4: $0_2 - 1_2 = 1_2$ with a borrow

The following example ($10110_2 - 1100_2$) demonstrates the four rules of binary subtraction:

$$\begin{array}{rl} 10110_2 & \text{Minuend} \\ -\ 1100_2 & \text{Subtrahend} \\ \hline ?010_2 & \text{Difference} \end{array}$$

Rule 1: $0_2 - 0_2 = 0_2$

Rule 2: $1_2 - 0_2 = 1_2$

Rule 3: $1_2 - 1_2 = 0_2$

Rule 4: (Explained below)

Rule 4 presents a different situation because you cannot subtract 1 from 0. Since you cannot subtract 1 from 0 and have a positive difference, you must borrow the 1 from the next higher order column of the minuend. The borrow may be indicated as shown below:

10	Borrow
0	After borrow
$\cancel{1}0110_2$	Minuend
$-\ 1100_2$	Subtrahend
1010_2	Difference

Rule 4: $(0_2 - 1_2)$

Now observe the following method of borrowing across more than one column in the example, $1000_2 - 1_2$:

1 1	Borrow
$0\ \cancel{1}\cancel{0}\ \cancel{1}\cancel{0}\ \cancel{1}0$	After borrow (base 2)
$\cancel{1}000_2$	Minuend
$-\ 1_2$	Subtrahend
0111_2	Difference

Let's practice some subtraction by solving the following problems:

Q15. Subtract:

$$\begin{array}{r} 11001_2 \\ -\ \ 1001_2 \\ \hline \end{array}$$

Q16. Subtract:

$$\begin{array}{r} 10101_2 \\ -\ \ 1010_2 \\ \hline \end{array}$$

Q17. Subtract:

$$\begin{array}{r} 11111_2 \\ -\ \ \ \ 10_2 \\ \hline \end{array}$$

Q18. *Subtract:*

$$\begin{array}{r} 111_2 \\ -\quad 100_2 \\ \hline \end{array}$$

Q19. *Subtract:*

$$\begin{array}{r} 10001_2 \\ -\quad 11_2 \\ \hline \end{array}$$

Q20. *Subtract:*

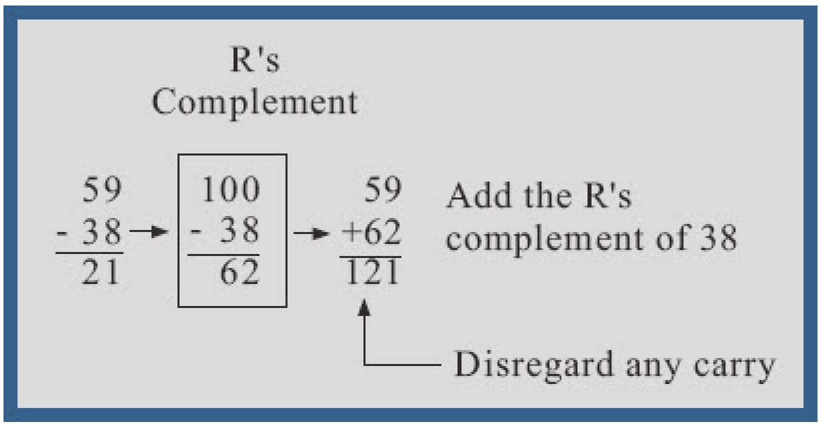
$$\begin{array}{r} 100000_2 \\ -\quad 1_2 \\ \hline \end{array}$$

Complementary Subtraction

If you do any work with computers, you will soon find out that most digital systems cannot subtract-they can only add. You are going to need a method of adding that gives the results of subtraction. Does that sound confusing? Really, it is quite simple. A COMPLEMENT is used for our subtractions. A complement is something used to complete something else.

In most number systems you will find two types of complements. The first is the amount necessary to complete a number up to the highest number in the number system. In the decimal system, this would be the difference between a given number and all 9s. This is called the nines complement or the radix-1 or R's-1 complement. As an example, the nines complement of 254 is 999 minus 254, or 745.

The second type of complement is the difference between a number and the next higher power of the number base. As an example, the next higher power of 10 above 999 is 1,000. The difference between 1,000 and 254 is 746. This is called the tens complement in the decimal number system. It is also called the radix or R's complement. We will use complements to subtract. Let's look at the *magic* of this process. There are three important points we should mention before we start: (1) Never complement the minuend in a problem, (2) always disregard any carry beyond the number of positions of the largest of the original numbers, and (3) add the R's complement of the original subtrahend to the original minuend. This will have the same effect as subtracting the original number. Let's look at a base ten example in which we subtract 38 from 59:



Now let's look at the number system that most computers use, the binary system. Just as the decimal system, had the nines (R's-1) and tens (R's) complement, the binary system has two types of complement methods. These two types are the ones (R's-1) complement and the twos (R's) complement. The binary system R's-1 complement is the difference between the binary number and all 1s. The R's complement is the difference between the binary number and the next higher power of 2.

Let's look at a quick and easy way to form the R's-1 complement. To do this, change each 1 in the original number to 0 and each 0 in the original number to 1 as has been done in the example below.

1011011_2

100100_2 R's-1 complement

There are two methods of achieving the R's complement. In the first method we perform the R's-1 complement and then add 1. This is much easier than subtracting the original number from the next higher power of 2. If you had subtracted, you would have had to borrow.

Saying it another way, to reach the R's complement of any binary number, change all 1s to 0s and all 0s to 1s, and then add 1.

As an example let's determine the R's complement of 10101101_2:

As an example let's determine the R's complement of 10101101_2:

Step 1 -- R's - 1 complement	01010010_2
Step 2 -- Add 1:	$+ \quad 1_2$
	01010011_2

The second method of obtaining the R's complement will be demonstrated on the binary number 0010110100_2.

Step 1—Start with the LSD, working to the MSD, writing the digits as they are up to and including the first one.

first one

0010110100_2

100_2

Step 2—Now R's-1 complement the remaining digits:

$$\underbrace{1101001}_{\text{R's-1 complement of remaining digits}}\overset{\overset{\text{first one}}{\downarrow}}{1}00_2$$

Now let's R's complement the same number using both methods:

Method 1

1001100_2	
0110011_2	R's-1 complement
$+ \quad 1_2$	Add 1
0110100_2	R's complement answer

Method 2

1001100_2	
$\underbrace{0110}_{\text{R's-1 complement of remaining digits}}\underbrace{100}_{\text{Unchanged digits}}{}_2$	R's-1 complement

Now let's do some subtracting by using the R's complement method. We will go through the subtraction of 3_{10} from 9_{10} (0011_2 from 1001_2):

9_{10}	1001_2	Minuend
$-\ 3_{10}$	$-\ 0011_2$	Subtrahend

Step 1-Leave the minuend alone:

$$1001_2 \text{ remains } 1001_2$$

Step 2—Using either method, R's complement the subtrahend:

$$1101_2 \text{ R's complement of subtrahend}$$

Step 3-Add the R's complement found in step 2 to the minuend of the original problem:

$$\begin{array}{rl} 1001_2 & \text{Original minuend} \\ +\ 1101_2 & \text{R's complement of subtrahend} \\ \hline 10110_2 & \text{Difference of original problem} \end{array}$$

Step 4—Remember to discard any carry beyond the size of the original number. Our original problem had four digits, so we discard the carry that expanded the difference to five digits. This carry we disregard is significant to the computer. It indicates that the difference is positive. Because we have a carry, we can read the difference directly without any further computations. Let's check our answer:

$$\begin{array}{rcr} 1001_2 & = & 9_{10} \\ -\ 0011_2 & = & -3_{10} \\ \hline 1\ 0110_2 & = & 6_{10} \end{array}$$

Discard

If we do *not* have a carry, it indicates the difference is a negative number. In that case, the difference must be R's complemented to produce the correct answer.

Let's look at an example that will explain this for you.

Subtract 9_{10} from 5_{10} (1001_2 from 0101_2):

Subtract 9_{10} from 5_{10} (1001_2 from 0101_2):

5_{10}	0101_2	Minuend
-9_{10}	-1001_2	Subtrahend
-4_{10}		

Step 1—Leave the minuend alone:

0101_2 remains 0101_2

Step 2—R's complement the subtrahend:

0111_2 R's complement of subtrahend

Step 3—Add the R's complement found in step 2 to the minuend of the original problem:

0101_2	Original minuend
$+0111_2$	Twos complement
1100_2	Difference of original problem

Step 4—We do *not* have a carry; and this tells us, and any computer, that our difference (answer) is negative. With no carry, we must R's complement the difference in step 3. We will then have arrived at the answer (difference) to our original problem. Let's do this R's complement step and then check our answer:

0100_2 R's complement of difference in step 3

Remember, we had no carry in step 3. That showed us our answer was going to be negative. Make sure you indicate the difference is negative. Let's check the answer to our problem:

$$\begin{array}{rcr} 0101_2 & = & 5_{10} \\ \underline{-\ 1001_2} & = & \underline{-9}_{10} \\ -\ 0100_2 & = & -4_{10} \end{array}$$

Try solving a few subtraction problems by using the complement method:

Q21. Subtract:

$$\begin{array}{r} 325_{10} \\ \underline{-\ 104}_{10} \end{array}$$

Q22. Subtract:

$$\begin{array}{r} 10010111_2 \\ \underline{-\ 00110100}_{\ 2} \end{array}$$

Q23. Subtract:

$$\begin{array}{r} 1011_2 \\ \underline{-\ 1100_2} \end{array}$$

1.4.3 Octal Number System

The octal, or base 8, number system is a common system used with computers. Because of its relationship with the binary system, it is useful in programming some types of computers.

Look closely at the comparison of binary and octal number systems in table 1-3. You can see that one octal digit is the equivalent value of three binary digits. The following examples of the conversion of octal 225_8 to binary and back again further illustrate this comparison:

Octal to binary

2	2	5_8
010	010	101_2

Binary to Octal

010	010	101_2
2	2	5_8

	BINARY	OCTAL	
2^0	0	0	8^0
	1	1	
2^1	10	2	
	11	3	
2^2	100	4	
	101	5	
	110	6	
	111	7	
2^3	1000	10	8^1
	1001	11	
	1010	12	
	1011	13	
	1100	14	
	1101	15	
	1110	16	
	1111	17	
2^4	10000	20	
	10001	21	
	10010	22	
	10011	23	
	10100	24	
	10101	25	
	10110	26	
	10111	27	
	11000	30	

Table 1-3 Binary and Octal Comparison

Unit and Number

The terms that you learned in the decimal and binary sections are also used with the octal system. The unit remains a single object, and the number is still a symbol used to represent one or more units.

Base (Radix)

As with the other systems, the radix, or base, is the number of symbols used in the system. The octal system uses eight symbols - 0 through 7. The base, or radix, is indicated by the subscript 8.

Positional Notation

The octal number system is a positional notation number system. Just as the decimal system uses powers of 10 and the binary system uses powers of 2, the octal system uses power of 8 to determine the value of a number's position. The following bar graph shows the positions and the power of the base:

$$8^3 \quad 8^2 \quad 8^1 \quad 8^0 \bullet 8^{-1} \quad 8^{-2} \quad 8^{-3}$$

Remember, that the power, or exponent, indicates the number of times the base is multiplied by itself. The value of this multiplication is expressed in base 10 as shown below:

$$8^3 = 8 \times 8 \times 8, \text{ or } 512_{10}$$

$$8^2 = 8 \times 8, \text{ or } 64_{10}$$

$$8^1 = 8_{10}$$

$$8^0 = 1_{10}$$

$$8^{-1} = \frac{1}{8}, \text{ or } .125_{10}$$

$$8^{-2} = \frac{1}{8 \times 8}, \text{ or } \frac{1}{64}, \text{ or } .015625_{10}$$

$$8^{-3} = \frac{1}{8 \times 8 \times 8}, \text{ or } \frac{1}{512}, \text{ or } .0019531_{10}$$

All numbers to the left of the radix point are whole numbers, and those to the right are fractional numbers.

MSD and LSD

When determining the most and least significant digits in an octal number, use the same rules that you used with the other number systems. The digit farthest to the left of the radix point is the MSD, and the one farthest right of the radix point is the LSD.

Example:

$$4\ 7\ 3\ 2 \bullet 2\ 6\ 1_8$$

↑ MSD	↑ Radix Point	↑ LSD

If the number is a whole number, the MSD is the nonzero digit farthest to the left of the radix point and the LSD is the digit immediately to the left of the radix point. Conversely, if the number is a fraction only, the nonzero digit closest to the radix point is the MSD and the LSD is the nonzero digit farthest to the right of the radix point.

Addition of Octal Numbers

The addition of octal numbers is not difficult provided you remember that anytime the sum of two digits exceeds 7, a carry is produced. Compare the two examples shown below:

$$\begin{array}{r} 4_8 \\ +2_8 \\ \hline 6_8 \end{array} \qquad \begin{array}{r} 4_8 \\ +4_8 \\ \hline 10_8 \end{array}$$

The octal addition table in table 1-4 will be of benefit to you until you are accustomed to adding octal numbers. To use the table, simply follow the directions used in this example:

Add: 6_8 and 5_8

+	0	1	2	3	4	5	6	7
0	0	1	2	3	4	5	6	7
1	1	2	3	4	5	6	7	10
2	2	3	4	5	6	7	10	11
3	3	4	5	6	7	10	11	12
4	4	5	6	7	10	11	12	13
5	5	6	7	10	11	12	13	14
6	6	7	10	11	12	13	14	15
7	7	10	11	12	13	14	15	16

X

Z

Y

Table 1-4 Octal Addition Table

Locate the 6 in the X column of the figure. Next locate the 5 in the Y column. The point in area Z where these two columns intersect is the sum.

$$\begin{array}{rl} 6_8 & \\ +5_8 & \\ \hline 13_8 & \text{(spoken, "one three, base eight")} \end{array}$$

If you use the concepts of addition you have already learned, you are ready to add octal numbers.

Work through the solutions to the following problems:

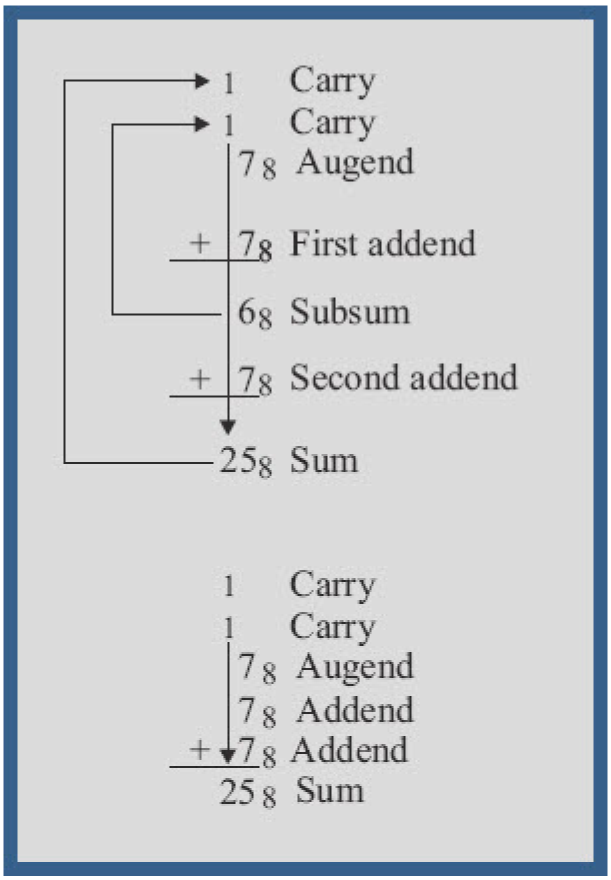
$$\begin{array}{rl} 11 & \text{Carry} \\ 456_8 & \text{Augend} \\ +\ 123_8 & \text{Addend} \\ \hline 601_8 & \text{Sum} \end{array}$$

$$\begin{array}{rl} 11111 & \text{Carry} \\ 77714_8 & \text{Augend} \\ +\quad 76_8 & \text{Addend} \\ \hline 100012_8 & \text{Sum} \end{array}$$

As was mentioned earlier in this section, each time the sum of a column of numbers exceeds 7, a carry is produced. More than one carry may be produced if there are three or more numbers to be added, as in this example:

7_8	Augend
7_8	Addend
$+\ 7_8$	Addend

The sum of the augend and the first addend is 6_8 with a carry. The sum of 6_8 and the second addend is 5_8 with a carry. You should write down the 5_8 and add the two carries and bring them down to the sum, as shown below:



Now let's try some practice problems:

Q24. Add:

$$\begin{array}{r} 3_8 \\ +\ 5_8 \\ \hline \end{array}$$

Q25. Add:

$$\begin{array}{r} 22_8 \\ +\ 36_8 \\ \hline \end{array}$$

Q26. Add:

$$\begin{array}{r} 621_8 \\ +\ 174_8 \\ \hline \end{array}$$

Q27. Add:

$$\begin{array}{r} 13255_8 \\ +\ 7031_8 \\ \hline \end{array}$$

Q28. Add

$$\begin{array}{r} 24_8 \\ 42_8 \\ +\ 63_8 \\ \hline \end{array}$$

Q29. Add:

$$\begin{array}{r} 3_8 \\ 5_8 \\ 2_8 \\ 6_8 \\ +\ 4_8 \\ \hline \end{array}$$

Subtraction of Octal Numbers

The subtraction of octal numbers follows the same rules as the subtraction of numbers in any other number system. The only variation is in the quantity of the borrow. In the decimal system, you had to borrow a group of 10_{10}. In the binary system, you borrowed a group of 2_{10}. In the octal system you will borrow a group of 8_{10}.

Consider the subtraction of 1 from 10 in decimal, binary, and octal number systems:

DECIMAL	BINARY	OCTAL
10_{10}	10_2	10_8
$-\ 1_{10}$	$-\ 1_2$	$-\ 1_8$
9_{10}	1_2	7_8

In each example, you cannot subtract 1 from 0 and have a positive difference. You must use a borrow from the next column of numbers. Let's examine the above problems and show the borrow as a *decimal* quantity for clarity:

^{10}	^{2}	^{8}	Borrow
$\cancel{1}0_{10}$	$\cancel{1}0_2$	$\cancel{1}0_8$	
$-\ 1_{10}$	$-\ 1_2$	$-\ 1_8$	
9_{10}	1_2	7_8	

When you use the borrow, the column you borrow from is reduced by 1, and the amount of the borrow is added to the column of the minuend being subtracted. The following examples show this procedure:

10	Borrow (Base 10)
2	After borrow
$\cancel{3}4_{10}$	Minuend
$-\ 9_{10}$	Subtrahend
25_{10}	Difference
10	Borrow (Base 8)
3	After borrow
$\cancel{4}6_8$	Minuend
$-\ 7_8$	Subtrahend
37_8	Difference

In the octal example 7_8 cannot be subtracted from 6_8, so you must borrow from the 4. Reduce the 4 by 1 and add 10_8 (the borrow) to the 6_8 in the minuend. By subtracting 7_8 from 16_8, you get a difference of 7_8. Write this number in the difference line and bring down the 3. You may need to refer to table 1-4, the octal addition table, until you are familiar with octal numbers. To use the table for subtraction, follow these directions. Locate the subtrahend in column Y. Now find where this line intersects with the minuend in area Z. The remainder, or difference, will be in row X directly above this point.

Do the following problems to practice your octal subtraction:

Q30. Subtract:

$$\begin{array}{r} 765_8 \\ -\ \underline{444}_8 \end{array}$$

Q31. Subtract:

$$\begin{array}{r} 44_8 \\ -\ \underline{\ \ 6}_8 \end{array}$$

Q32. Subtract:

$$\begin{array}{r} 532_8 \\ -\ \underline{174}_8 \end{array}$$

Q33. Subtract:

$$\begin{array}{r} 1023_8 \\ -\ \underline{424}_8 \end{array}$$

Q34. Subtract:

$$\begin{array}{r} 423_8 \\ -\ \underline{326}_8 \end{array}$$

Q35. Subtract:

$$\begin{array}{r} 7776_8 \\ -\ \underline{\ \ \ 7}_8 \end{array}$$

Check your answers by adding the subtrahend and difference for each problem.

1.4.4 Hexadecimal (HEX) Number System

The hex number system is a more complex system in use with computers. The name is derived from the fact the system uses 16 symbols. It is beneficial in computer programming because of its relationship to the binary system. Since 16 in the decimal system is the fourth power of 2 (or 2^4); one hex digit has a value equal to four binary digits. Table 1-5 shows the relationship between the two systems.

	BINARY	OCTAL	
2^0	0	0	16^0
	1	1	
2^1	10	2	
	11	3	
2^2	100	4	
	101	5	
	110	6	
	111	7	
2^3	1000	8	
	1001	9	
	1010	A	
	1011	B	
	1100	C	
	1101	D	
	1110	E	
	1111	F	
2^4	10000	10	16^1
	10001	11	
	10010	12	
	10011	13	
	10100	14	
	10101	15	
	10110	16	
	10111	17	
	11000	18	
	11001	19	
	11010	1A	
	11011	1B	
	11100	1C	

Table 1-5 Binary and Hexadecimal Comparison

Unit and Number

As in each of the previous number systems, a unit stands for a single object.
A number in the hex system is the symbol used to represent a unit or quantity. The Arabic numerals 0 through 9 are used along with the first six letters of the alphabet. You have probably used letters in math problems to represent unknown quantities, but in the hex system A, B, C, D, E, and F, each have a definite value as shown below:

$$A_{16} = 10_{10}$$

$$B_{16} = 11_{10}$$

$$C_{16} = 12_{10}$$

$$D_{16} = 13_{10}$$

$$E_{16} = 14_{10}$$

$$F_{16} = 15_{10}$$

Base (Radix)

The base, or radix, of this system is 16, which represents the number of symbols used in the system. A quantity expressed in hex will be annotated by the subscript 16, as shown below:

$$A3EF_{16}$$

Positional Notation

Like the binary, octal, and decimal systems, the hex system is a positional notation system. Powers of 16 are used for the positional values of a number. The following bar graph shows the positions:

$$16^3\ 16^2\ 16^1\ 16^0 \bullet 16^{-1}\ 16^{-2}\ 16^{-3}$$

Multiplying the base times itself the number of times indicated by the exponent will show the equivalent decimal value:

$$16^3 = 16 \text{ x } 16 \text{ x } 16\text{, or } 4096_{10}$$

$$16^3 = 16 \text{ x } 16\text{, or } 256_{10}$$

$$16^1 = 16_{10}$$

$$16^0 = 1_{10}$$

$$16^{-1} = \frac{1}{16}\text{, or } .0625_{10}$$

$$16^{-2} = \frac{1}{16 \text{ x } 16}\text{, or } .0039062_{10}$$

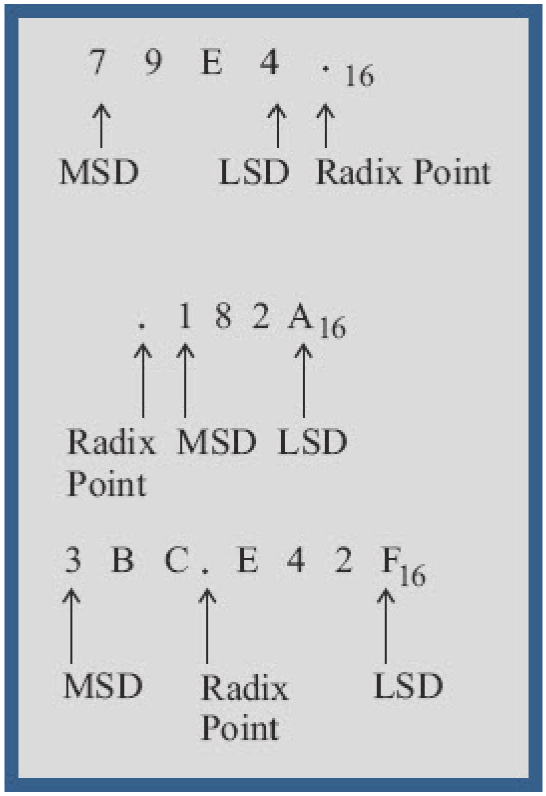
$$16^{-3} = \frac{1}{16 \text{ x } 16 \text{ x } 16}\text{, or } .0002441_{10}$$

You can see from the positional values that usually fewer symbol positions are required to express a number in hex than in decimal. The following example shows this comparison:

625_{16} is equal to 1573_{10}

MSD and LSD

The most significant and least significant digits will be determined in the same manner as the other number systems. The following examples show the MSD and LSD of whole, fractional, and mixed hex numbers:



Addition of Hex Numbers

The addition of hex numbers may seem intimidating at first glance, but it is no different than addition in any other number system. The same rules apply. Certain combinations of symbols produce a carry while others do not. Some numerals combine to produce a sum represented by a letter. After a little practice you will be as confident adding hex numbers as you are adding decimal numbers.

Study the hex addition table in table 1-6. Using the table, add 7 and 7. Locate the number 7 in both columns X and Y. The point in area Z where these two columns intersect is the sum; in this case 7 + 7 = E. As long as the sum of two numbers is 15_{10} or less, only one symbol is used for the sum. A carry will be produced when the sum of two numbers is 16_{10} or greater, as in the following examples:

$$\begin{array}{r} 8_{16} \\ +\ 8_{16} \\ \hline 10_{16} \end{array} \qquad \begin{array}{r} A_{16} \\ +\ D_{16} \\ \hline 17_{16} \end{array} \qquad \begin{array}{r} D_{16} \\ +\ 9_{16} \\ \hline 16_{16} \end{array}$$

+	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
1	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	10
2	2	3	4	5	6	7	8	9	A	B	C	D	E	F	10	11
3	3	4	5	6	7	8	9	A	B	C	D	E	F	10	11	12
4	4	5	6	7	8	9	A	B	C	D	E	F	10	11	12	13
5	5	6	7	8	9	A	B	C	D	E	F	10	11	12	13	14
6	6	7	8	9	A	B	C	D	E	F	10	11	12	13	14	15
7	7	8	9	A	B	C	D	E	F	10	11	12	13	14	15	16
8	8	9	A	B	C	D	E	F	10	11	12	13	14	15	16	17
9	9	A	B	C	D	E	F	10	11	12	13	14	15	16	17	18
A	A	B	C	D	E	F	10	11	12	13	14	15	16	17	18	19
B	B	C	D	E	F	10	11	12	13	14	15	16	17	18	19	1A
C	C	D	E	F	10	11	12	13	14	15	16	17	18	19	1A	1B
D	D	E	F	10	11	12	13	14	15	16	17	18	19	1A	1B	1C
E	E	F	10	11	12	13	14	15	16	17	18	19	1A	1B	1C	1D
F	F	10	11	12	13	14	15	16	17	18	19	1A	1B	1C	1D	1E

X

Z

Y

Table 1-6 Hexadecimal Addition Table

Use the addition table and follow the solution of the following problems:

	456_{16}	Augend
+	784_{16}	Addend
	BDA_{16}	Sum

In this example each column is straight addition with no carry.

Now add the addend (784_{16}) and the sum (BDA_{16}) of the previous problem:

	1 1	Carry
	784_{16}	Augend
+	BDA_{16}	Addend
	$1\,3\,5\,E_{16}$	Sum

Here the sum of 4 and A is E. Adding 8 and D is 15_{16}; write down 5 and carry a 1. Add the first carry to the 7 in the next column and add the sum, 8, to B. The result is 13_{16}; write down 3 and carry a 1. Since only the last carry is left to add, bring it down to complete the problem.

Now observe the procedures for a more complex addition problem. You may find it easier to add the Arabic numerals in each column first:

	1 1 1	Carry
	$C14_{16}$	Augend
	$19E_{16}$	Addend
	571_{16}	Addend
+	$BB3_{16}$	Addend
	$1ED6_{16}$	Sum

The sum of 4, E, 1, and 3 in the first column is 16_{16}. Write down the 6 and the carry. In the second column, 1, 1, 9, and 7 equals 12_{16}. Write the carry over the next column. Add B and 2 - the sum is D. Write this in the sum line. Now add the final column, 1, 1, 5, and C. The sum is 13_{16}. Write down the carry; then add 3 and B - the sum is E. Write down the E and bring down the final carry to complete the problem.

Now solve the following addition problems:

Q36. Add:

$$\begin{array}{r} 4A3C_{16} \\ +\ 9351_{16} \\ \hline \end{array}$$

Q37. Add:

$$\begin{array}{r} 4321_{16} \\ +\ DCBA_{16} \\ \hline \end{array}$$

Q38. Add:

$$\begin{array}{r} 274_{16} \\ +\ FEB_{16} \\ \hline \end{array}$$

Q39. Add:

$$\begin{array}{r} 79DF_{16} \\ +\ A641_{16} \\ \hline \end{array}$$

Q40. Add:

$$\begin{array}{r} ECFD_{16} \\ +\ A4AE_{16} \\ \hline \end{array}$$

Q41. Add:

$$\begin{array}{r} BC_{16} \\ A23_{16} \\ +\ FC9_{16} \\ \hline \end{array}$$

Subtraction of Hex Numbers

The subtraction of hex numbers looks more difficult than it really is. In the preceding sections you learned all the rules for subtraction. Now you need only to apply those rules to a new number system. The symbols may be different and the amount of the borrow is different, but the rules remain the same.

Use the hex addition table (table 1-6) to follow the solution of the following problems:

$$\begin{array}{rl} ABC_{16} & \text{Minuend} \\ \underline{-\quad 642_{16}} & \text{Subtrahend} \end{array}$$

Working from left to right, first locate the subtrahend (2) in column Y. Follow this line across area Z until you reach C. The difference is located in column X directly above the C - in this case A. Use this same procedure to reach the solution:

$$\begin{array}{rl} ABC_{16} & \text{Minuend} \\ \underline{-\quad 642_{16}} & \text{Subtrahend} \\ 47A_{16} & \text{Difference} \end{array}$$

Now examine the following solutions:

$$\begin{array}{rl} 7E5E_{16} & \text{Minuend} \\ \underline{-\quad 471_{16}} & \text{Subtrahend} \\ 3744_{16} & \text{Difference} \end{array}$$

$$\begin{array}{rl} 1E9C4_{16} & \text{Minuend} \\ \underline{-\quad F4A1_{16}} & \text{Subtrahend} \\ F523_{16} & \text{Difference} \end{array}$$

In the previous example, when F was subtracted from 1E, a borrow was used. Since you cannot subtract F from E and have a positive difference, a borrow of 10_{16} was taken from the next higher value column. The borrow was added to E, and the higher value column was reduced by 1.

The following example shows the use of the borrow in a more difficult problem:

				10_{16}	Borrow
			2		Minuend reduced by 1
	4	A	~~3~~	7_{16}	Minuend
-	2	C	4	B_{16}	Subtrahend
				C_{16}	Difference

In this first step, B cannot be subtracted from 7, so you take a borrow of 10_{16} from the next higher value column. Add the borrow to the 7 in the minuend; then subtract (17_{16} minus B_{16} equals C_{16}). Reduce the number from which the borrow was taken (3) by 1.

To subtract 4_{16} from 2_{16} also requires a borrow, as shown below:

			10_{16}	10_{16}	Borrow
		9	2		Minuend reduced by 1
	4	~~A~~	~~3~~	7_{16}	Minuend
-	2	C	4	B_{16}	Subtrahend
			E	C_{16}	Difference

Borrow 10_{16} from the A and reduce the minuend by 1. Add the borrow to the 2 and subtract 4_{16} from 12_{16}. The difference is E.

When solved the problem looks like this:

	10	10	10		Borrow (Base 16)
	3	9	2		Minuend reduced by 1
	~~4~~	~~A~~	~~3~~	7_{16}	Minuend
-	2	C	4	B_{16}	Subtrahend
	1	D	E	C_{16}	Difference

Remember that the borrow is 10_{16} not 10_{10}.

There may be times when you need to borrow from a column that has a 0 in the minuend. In that case, you borrow from the next highest value column, which will provide you with a value in the 0 column that you can borrow from.

		F		Borrow reduced by 1
		~~10~~	10	Borrow (Base 16)
	1			Minuend reduced by 1
	~~2~~	0	7_{16}	Minuend
-			A_{16}	Subtrahend
	1	F	D_{16}	Difference

To subtract A from 7, you must borrow. To borrow you must first borrow from the 2. The 0 becomes 10_{16}, which can give up a borrow. Reduce the 10_{16} by 1 to provide a borrow for the 7. Reducing 10_{16} by 1 equals F. Subtracting A_{16} from 17_{16} gives you D_{16}. Bring down the 1 and F for a difference of $1FD_{16}$.

Now let's practice what we've learned by solving the following hex subtraction problems:

Q42. *Subtract:*

$$\begin{array}{r} 758_{16} \\ - \underline{\;423}_{16} \end{array}$$

Q43. *Subtract:*

$$\begin{array}{r} D9F_{16} \\ - \underline{\;46A}_{16} \end{array}$$

Q44. *Subtract:*

$$\begin{array}{r} A1C6_{16} \\ + \underline{\;C95_{16}} \end{array}$$

Q45. *Subtract:*

$$\begin{array}{r} 4057_{16} \\ - \underline{\;9A4}_{16} \end{array}$$

Q46. *Subtract:*

$$\begin{array}{r} 13579_{16} \\ - \underline{\;2ABD}_{16} \end{array}$$

Q47. *Subtract:*

$$\begin{array}{r} EFACD_{16} \\ - \underline{\;ACBBE}_{16} \end{array}$$

1.5 CONVERSION OF BASES

We mentioned in the introduction to this chapter that digital computers operate on electrical pulses. These pulses or the absence of, are easily represented by binary numbers. A pulse can represent a binary 1, and the lack of a pulse can represent a binary 0 or vice versa.

The sections of this chapter that discussed octal and hex numbers both mentioned that their number systems were beneficial to programmers. You will see later in this section that octal and hex numbers are easily converted to binary numbers and vice versa..

If you are going to work with computers, there will be many times when it will be necessary to convert decimal numbers to binary, octal, and hex numbers. You will also have to be able to convert binary, octal, and hex numbers to decimal numbers. Converting each number system to each of the others will be explained. This will prepare you for converting from any base to any other base when needed.

1.6 DECIMAL CONVERSION

Some computer systems have the capability to convert decimal numbers to binary numbers. They do this by using additional circuitry. Many of these systems require that the decimal numbers be converted to another form before entry.

1.6.1 Decimal to Binary

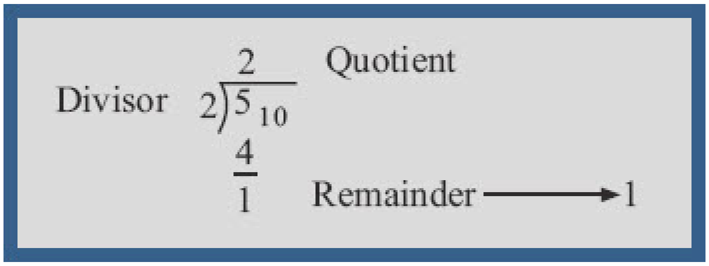
Conversion of a decimal number to any other base is accomplished by dividing the decimal number by the radix of the system you are converting to. The following definitions identify the basic terms used in division:

- DIVIDEND-The number to be divided
- DIVISOR-The number by which a dividend is divided
- QUOTIENT-The number resulting from the division of one number by another
- REMAINDER-The final undivided part after division that is less or of a lower degree than the divisor.

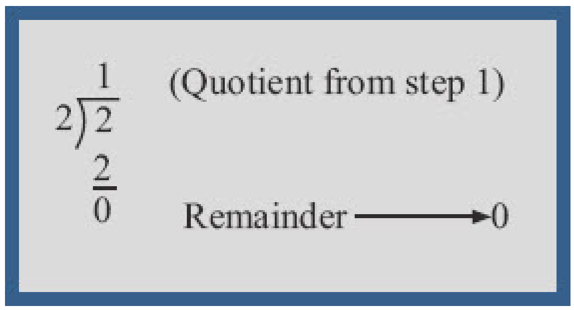
To convert a base 10 whole number to its binary equivalent, first set up the problem for division:

$$2\overline{)5_{10}}$$

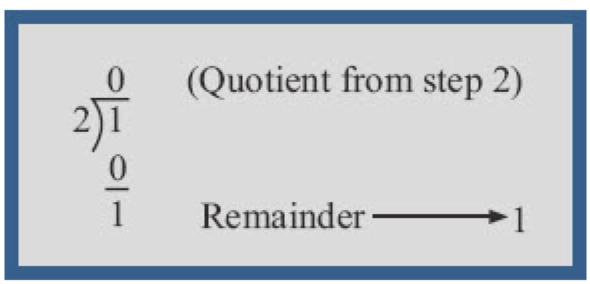
Step 1—Divide the base 10 number by the radix (2) of the binary system and extract the remainder (this becomes the binary number's LSD).



Step 2—Continue the division by dividing the quotient of step 1 by the radix (2 x 2).

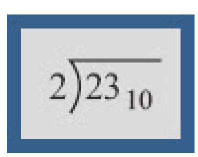


Step 3—Continue dividing quotients by the radix until the quotient becomes smaller that the divisor; then do one more division. The remainder is our MSD.



The remainder in step 1 is our LSD. Now rewrite the solution, and you will see that 5_{10} equals 101_2. Now follow the conversion of 23_{10} to binary:

Step 1—Set up the problem for division:



Step 2—Divide the number and extract the remainder:

$$\begin{array}{rl} 11 & \\ 2\overline{)23} & \\ \underline{2} & \\ 03 & \\ \underline{2} & \\ 1 & \text{Remainder} \longrightarrow 1 \ \text{(LSD)} \end{array}$$

$$\begin{array}{rl} 5 & \\ 2\overline{)11} & \text{(Quotient from previous step)} \\ \underline{10} & \\ 1 & \text{Remainder} \longrightarrow 1 \end{array}$$

$$\begin{array}{rl} 2 & \\ 2\overline{)5} & \text{(Quotient from previous step)} \\ \underline{4} & \\ 1 & \text{Remainder} \longrightarrow 1 \end{array}$$

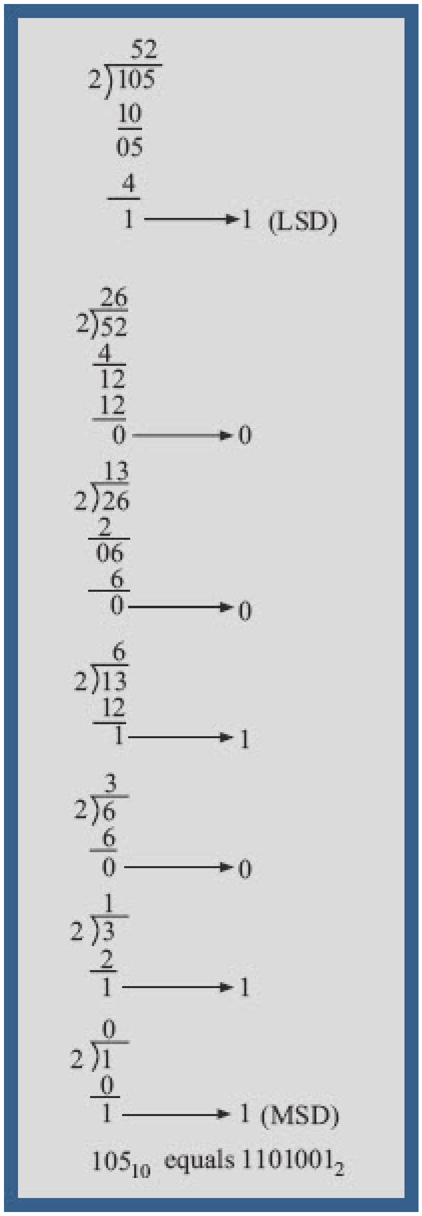
$$\begin{array}{rl} 1 & \\ 2\overline{)2} & \text{(Quotient from previous step)} \\ \underline{2} & \\ 0 & \text{Remainder} \longrightarrow 0 \end{array}$$

$$\begin{array}{rl} 0 & \\ 2\overline{)1} & \text{(Quotient from previous step)} \\ \underline{0} & \\ 1 & \text{Remainder} \longrightarrow 1 \ \text{(MSD)} \end{array}$$

Step 3—Rewrite the solution from MSD to LSD:

$$10111_2$$

No matter how large the decimal number may be, we use the same procedure. Let's try the problem below. It has a larger dividend:



We can convert fractional decimal numbers by multiplying the fraction by the radix and extracting the portion of the product to the *left* of the radix point. Continue to multiply the fractional portion of the previous product until the desired degree of accuracy is attained.

Let's go through this process and convert 0.25_{10} to its binary equivalent:

$$
\begin{array}{llr}
 & & .25_{10} \\
 & & \underline{\times \quad 2} \\
\text{MSD} \leftarrow & 0 \leftarrow & 0.50 \\
 & & \underline{\times \quad 2} \\
\text{LSD} \leftarrow & 1 \leftarrow & 1.00
\end{array}
$$

The *first* figure to the left of the radix point is the MSD, and the last figure of the computation is the LSD. Rewrite the solution from MSD to LSD preceded by the radix point as shown:

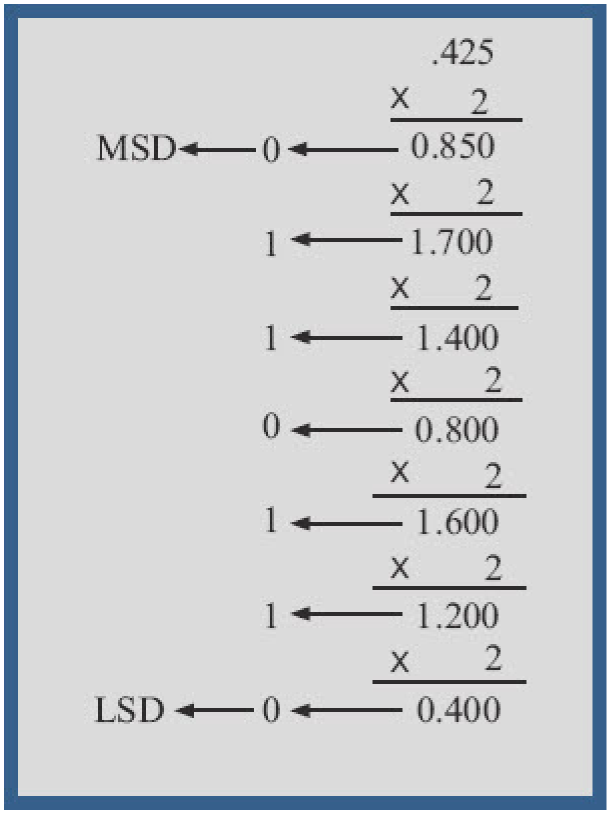
$$\boxed{.01_2}$$

Now try converting $.625_{10}$ to binary:

$$
\begin{array}{llr}
 & & .625 \\
 & & \underline{\times \quad 2} \\
\text{MSD} \leftarrow & 1 \leftarrow & 1.250 \\
 & & \underline{\times \quad 2} \\
 & 0 \leftarrow & 0.500 \\
 & & \underline{\times \quad 2} \\
\text{LSD} \leftarrow & 1 \leftarrow & 1.000 \\
 & & \underline{\times \quad 2} \\
 & & 0.000
\end{array}
$$

$.625_{10}$ equals $.101_2$

As we mentioned before, you should continue the operations until you reach the desired accuracy. For example, convert $.425_{10}$ to five places in the binary system:



Although the multiplication was carried out for seven places, you would only use what is required. Write out the solution as shown:

$.01101_2$

To convert a mixed number such as 37.625_{10} to binary, split the number into its whole and fractional components and solve each one separately. In this problem carry the fractional part to four places. When the conversion of each is completed, recombine it with the radix point as shown below:

$$37_{10} = 100101_2$$

$$.625_{10} = .1010_2$$

$$37.625_{10} = 100101.1010_2$$

Convert the following decimal numbers to binary:

Q48. 72_{10}.

Q49. 97_{10}.

Q50. 243_{10}.

Q51. 0.875_{10} *(four places).*

Q52. 0.33_{10} *(four places).*

Q53. 17.42_{10} *(five places)*

1.6.2 Decimal to Octal

The conversion of a decimal number to its base 8 equivalent is done by the repeated division method. You simply divide the base 10 number by 8 and extract the remainders. The first remainder will be the LSD, and the last remainder will be the MSD.

Look at the following example. To convert 15_{10} to octal, set up the problem for division:

$$8\overline{)15_{10}}$$

Since 8 goes into 15 one time with a 7 remainder, 7 then is the LSD. Next divide 8 into the quotient (1). The result is a 0 quotient with a 1 remainder. The 1 is the MSD:

$$\begin{array}{l} \quad\ \ 1 \\ 8\overline{)15_{10}} \\ \quad\ \ \underline{8} \\ \quad\ \ 7 \longrightarrow 7\ (\text{LSD}) \\ \\ \quad\ 0 \\ 8\overline{)1} \\ \quad\ \underline{0} \\ \quad\ 1 \longrightarrow 1\ (\text{MSD}) \end{array}$$

Now write out the number from MSD to LSD as shown:

$$17_8$$

The same process is used regardless of the size of the decimal number. Naturally, more divisions are needed for larger numbers, as in the following example:

Convert 264_{10} to octal:

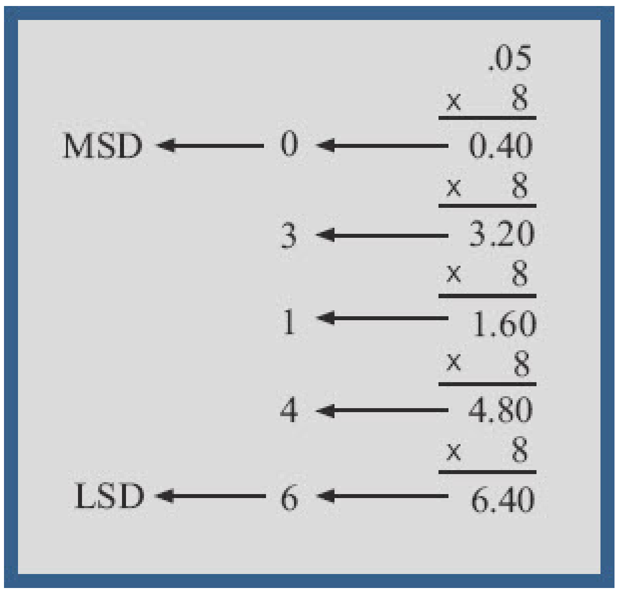
$$\begin{array}{l} \qquad\ 33 \\ 8\overline{)264_{10}} \\ \quad\ \underline{24} \\ \qquad\ 24 \\ \qquad\ \underline{24} \\ \qquad\ \ 0 \longrightarrow 0\ (\text{LSD}) \\ \\ \qquad\ 4 \\ 8\overline{)33} \\ \quad\ \underline{32} \\ \qquad 1 \longrightarrow 1 \\ \\ \quad\ \ 0 \\ 8\overline{)4} \\ \quad\ \underline{0} \\ \quad\ 4 \longrightarrow 4\ (\text{MSD}) \end{array}$$

By rewriting the solution, you find that the octal equivalent of 264_{10} is as follows:

410_8

To convert a decimal fraction to octal, *multiply* the fraction by 8. Extract everything that appears to the left of the radix point. The first number extracted will be the MSD and will follow the radix point. The last number extracted will be the LSD.

Convert 0.05_{10} to octal:



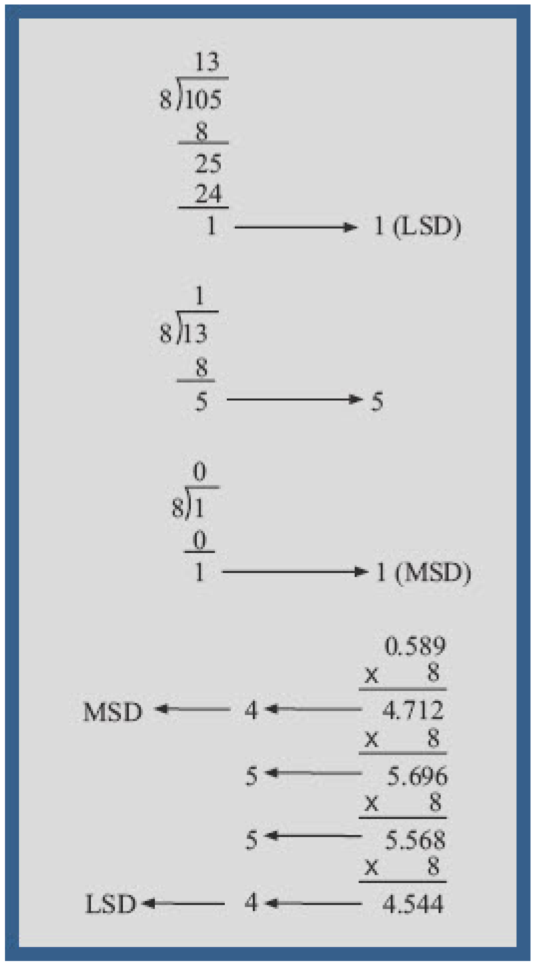
Write the solution from MSD to LSD:

$.03146_8$

You can carry the conversion out to as many places as needed, but usually four or five places are enough.

To convert a mixed decimal number to its octal equivalent, split the number into whole and fractional portions and solve as shown below:

Convert 105.589_{10} to octal:



Combine the portions into a mixed number:

$$151.4554_8$$

Convert the following decimal numbers to octal:

Q54. 7_{10}

Q55. 43_{10}

Q56. 499_{10}

Q57. 0.951_{10} *(four places).*

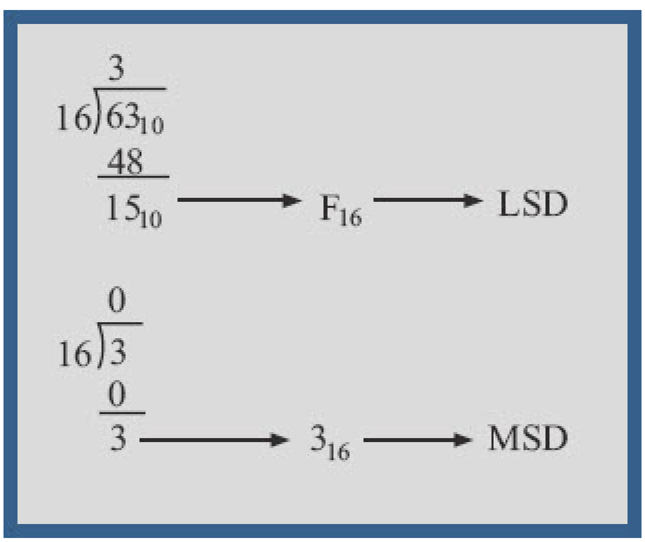
Q58. 0.004_{10} *(five places).*

Q59. 252.17_{10} *(three places).*

1.6.3 Decimal to Hex

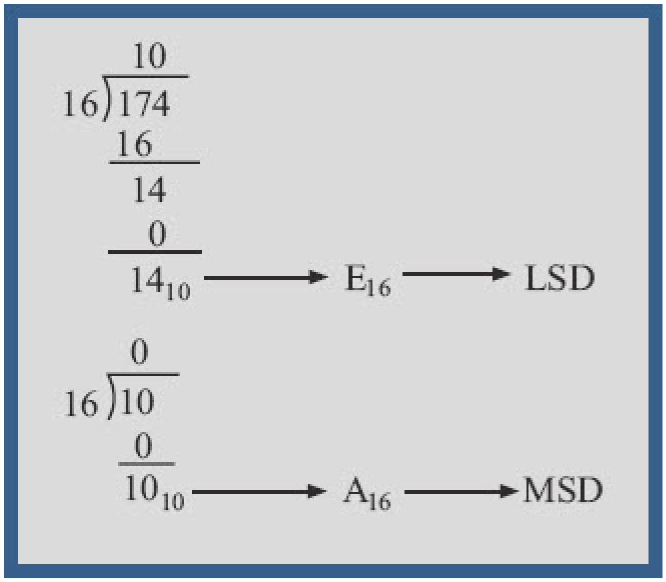
To convert a decimal number to base 16, follow the repeated division procedures you used to convert to binary and octal, only divide by 16. Let's look at an example:

Convert 6310 to hex:



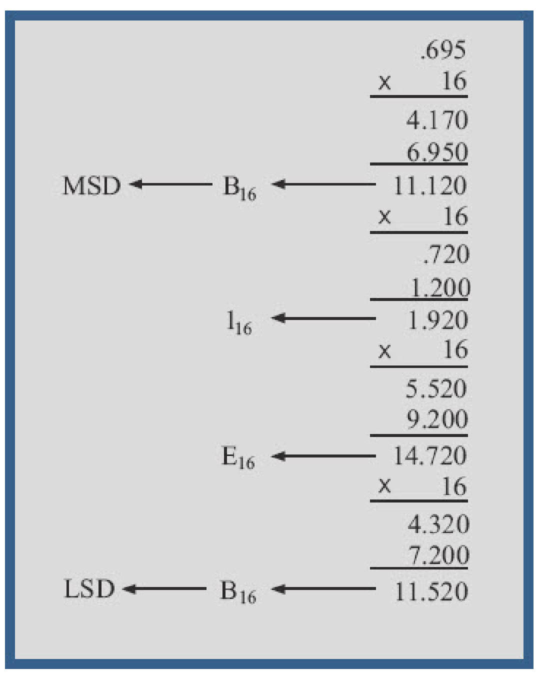
Therefore, the hex equivalent of 63_{10} is $3F_{16}$.

You have to remember that the remainder is in base 10 and must be converted to hex if it exceeds 9. Let's work through another example:



Write the solution from MSD to LSD:

AE_{16}



The solution: .$B1EB_{16}$

Should you have the need to convert a decimal mixed number to hex, convert the whole number and the fraction separately; then recombine for the solution.

Convert the following decimal numbers to hex:

Q60. 42_{10}.

Q61. 83_{10}.

Q62. 176_{10}.

Q63. 491_{10}.

Q64. 0.721_{10} *(four places).*

The converting of binary, octal, and hex numbers to their decimal equivalents is covered as a group later in this section.

1.7 BINARY CONVERSION

Earlier in this chapter, we mentioned that the octal and hex number systems are useful to computer programmers. It is much easier to provide data to a computer in one or the other of these systems. Likewise, it is important to be able to convert data from the computer into one or the other number systems for ease of understanding the data.

1.7.1 Binary to Octal

Look at the following numbers:

$$10111001001101_2$$

$$27115_8$$

You can easily see that the octal number is much easier to say. Although the two numbers look completely different, they are equal.

Since 8 is equal to 2^3, then one octal digit can represent three binary digits, as shown below:

$$
\begin{aligned}
0_8 &= 000_2 \\
1_8 &= 001_2 \\
2_8 &= 010_2 \\
3_8 &= 011_2 \\
4_8 &= 100_2 \\
5_8 &= 101_2 \\
6_8 &= 110_2 \\
7_8 &= 111_2
\end{aligned}
$$

With the use of this principle, the conversion of a binary number is quite simple. As an example, follow the conversion of the binary number at the beginning of this section.

Write out the binary number to be converted. Starting at the radix point and moving left, break the binary number into groups of three as shown. This grouping of binary numbers into groups of three is called binary-coded octal (BCO). Add 0s to the left of any MSD that will fill a group of three:

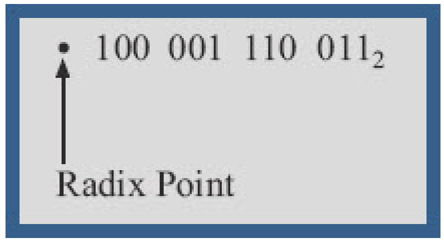
$$010\ 111\ 001\ 001\ 101\,\bullet_{2}$$

Radix Point

Next, write down the octal equivalent of each group:

010	111	001	001	$101._2$
2	7	1	1	$5._8$

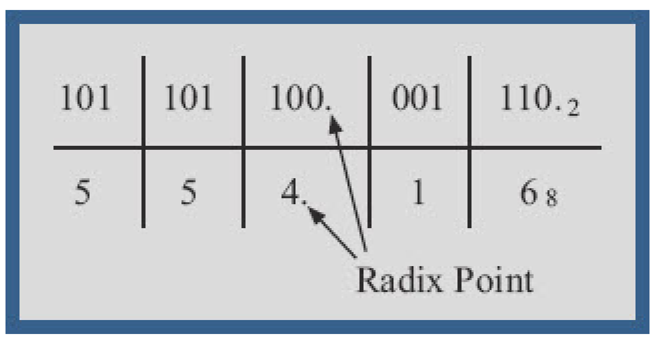
To convert a binary fraction to its octal equivalent, starting at the radix point and moving right, expand each digit into a group of three:



Add 0s to the right of the LSD if necessary to form a group of three. Now write the octal digit for each group of three, as shown below:

.100	001	110	$011._2$
.4	1	6	3_8

To convert a mixed binary number, starting at the radix point, form groups of three both right and left:



Convert the following binary numbers to octal:

Q65.	10_2.
Q66.	1010_2.
Q67.	101111_2.
Q68.	0.0011_2.
Q69.	0.110011_2.
Q70.	110111.010101_2.

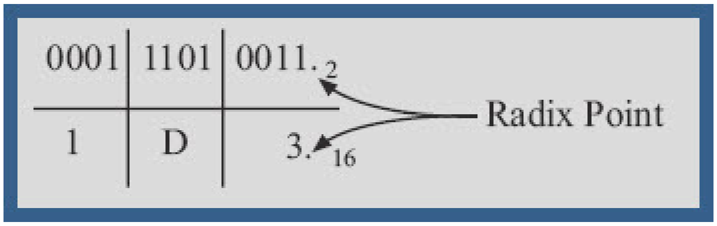
1.7.2 Binary to Hex

The table below shows the relationship between binary and hex numbers. You can see that four binary digits may be represented by one hex digit. This is because 16 is equal to 2^4.

HEX		BINARY
0	=	0000
1	=	0001
2	=	0010
3	=	0011
4	=	0100
5	=	0101
6	=	0110
7	=	0111
8	=	1000
9	=	1001
A	=	1010
B	=	1011
C	=	1100
D	=	1101
E	=	1110
F	=	1111

Using this relationship, you can easily convert binary numbers to hex. Starting at the radix point and moving either right or left, break the number into groups of four. The grouping of binary into four bit groups is called binary-coded hexadecimal (BCH).

Convert 111010011_2 to hex:



Add 0s to the left of the MSD of the whole portion of the number and to the right of the LSD of the fractional part to form a group of four.

Convert $.111_2$ to hex:

$$\frac{.1110_2}{.E_{16}}$$

In this case, if a 0 had not been added, the conversion would have been .716, which is incorrect.

Convert the following binary numbers to hex:

Q71. 10_2.

Q72. 1011_2.

Q73. 101111_2.

Q74. 0.0011_2.

Q75. 0.110011_2.

Q76. 110111.010101_2.

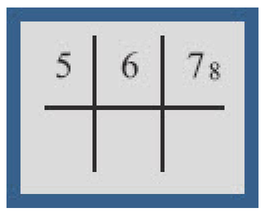
1.8 OCTAL CONVERSION

The conversion of one number system to another, as we explained earlier, is done to simplify computer programming or interpreting of data.

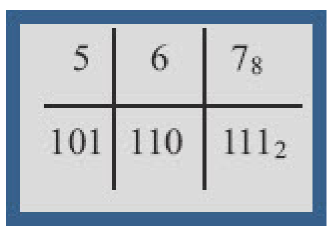
1.8.1 Octal to Binary

For some computers to accept octal data, the octal digits must be converted to binary. This process is the reverse of binary to octal conversion.

To convert a given octal number to binary, write out the octal number in the following format. We will convert octal 567_8:



Next, below each octal digit write the corresponding three-digit binary-coded octal equivalent:



Solution: 567_8 equals $101\ 110\ 111_2$

Remove the conversion from the format:

$$101110111_2$$

As you gain experience, it may not be necessary to use the block format.

An octal fraction ($.123_8$) is converted in the same manner, as shown below:

.1	2	3
.001	010	011_2

Solution: $.123_8$ equals $.001010011_2$

Apply these principles to convert mixed numbers as well.

Convert 32.25_8 to binary:

3	2.	2	5_8
011	010.	010	101_2

Solution: 32.25_8 equals 011010.010101_2

Convert the following numbers to binary:

Q77. 73_8

Q78. 512_8

Q79. 403_8

Q80. 0.456_8

Q81. 0.73_8

Q82. 36.5_8

1.8.2 Octal to Hex

You will probably not run into many occasions that call for the conversion of octal numbers to hex. Should the need arise, conversion is a two-step procedure. Convert the octal number to binary; then convert the binary number to hex. The steps to convert 53.7_8 to hex are shown below:

5	3 .	7_8
101	011.	111_2

Regroup the binary digits into groups of four and add zeros where needed to complete groups; then convert the binary to hex.

0010	1011.	1110_2
2	B .	E_{16}

Solution: 53.7_8 equals $2B.E_{16}$

Convert the following numbers to hex:

Q83.	74_8
Q84.	512_8
Q85.	0.03_8
Q86.	14.42_8

1.9 HEX CONVERSION

The procedures for converting hex numbers to binary and octal are the reverse of the binary and octal conversions to hex.

1.9.1 Hex to Binary

To convert a hex number to binary, set up the number in the block format you used in earlier conversions. Below each hex digit, write the four-digit binary equivalent. Observe the following example:

Convert ABC_{16} to binary:

A	B	C_{16}
1010	1011	1100_2

Solution: $ABC_{16} = 101010111100_2$

1.9.2 Hex to Octal

Just like the conversion of octal to hex, conversion of hex to octal is a two-step procedure. First, convert the hex number to binary; and second, convert the binary number to octal. Let's use the same example we used above in the hex to binary conversion and convert it to octal:

A	B	C_{16}
1010	1011	1100_2

101	010	111	100_2
5	2	7	4_8

Convert these base 16 numbers to their equivalent base 2 and base 8 numbers:

Q87. 23_{16}

Q88. $1B_{16}$

Q89. $0.E4_{16}$

Q90. $45.A_{16}$

1.10 CONVERSION TO DECIMAL

Computer data will have little meaning to you if you are not familiar with the various number systems. It is often necessary to convert those binary, octal, or hex numbers to decimal numbers. The need for understanding is better illustrated by showing you a paycheck printed in binary. A check in the amount of $\$10,010,101.00_2$ looks impressive but in reality only amounts to $\$149.00_{10}$

1.10.1 Binary to Decimal

The computer that calculates your pay probably operates with binary numbers, so a conversion takes place in the computer before the amount is printed on your check. Some computers, however, don't automatically convert from binary to decimal. There may be times when you must convert mathematically.

To convert a base 2 number to base 10, you must know the decimal equivalent of each power of 2. The decimal value of a power of 2 is obtained by multiplying 2 by itself the number of times indicated by the exponent for whole numbers; for example, $2_4 = 2 \cdot 2 \cdot 2 \cdot 2$ or 16_{10}.

For fractional numbers, the decimal value is equal to 1 divided by 2 multiplied by itself the number of times indicated by the exponent. Look at this example:

$$2^{-3} = \frac{1}{2 \times 2 \times 2} \text{ or } .125_{10}$$

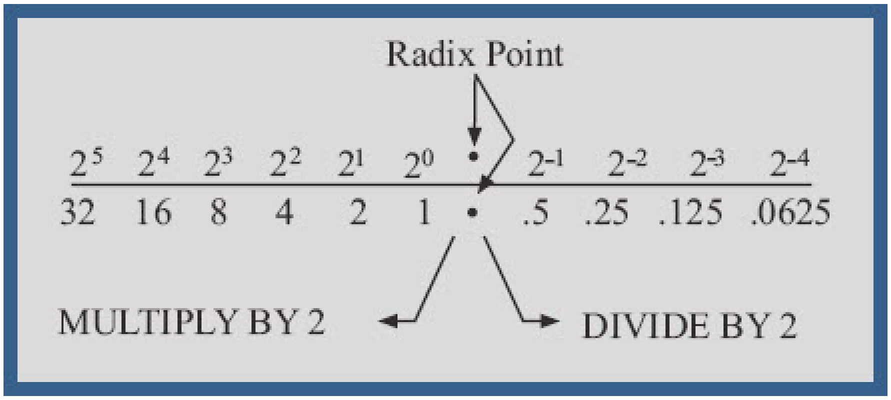
The table below shows a portion of the positions and decimal values of the binary system:

Radix Point

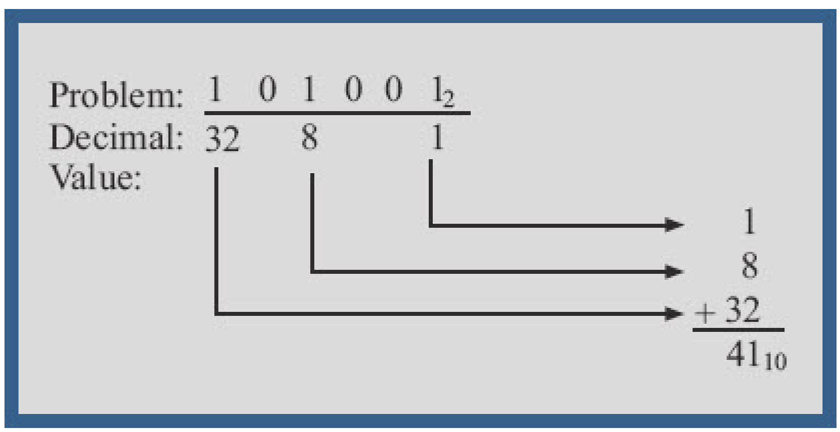
2^5	2^4	2^3	2^2	2^1	2^0	.	2^{-1}	2^{-2}	2^{-3}
32	16	8	4	2	1	.	.5	.25	.125

Remember, earlier in this chapter you learned that any number to the 0 power is equal to 1_{10}.

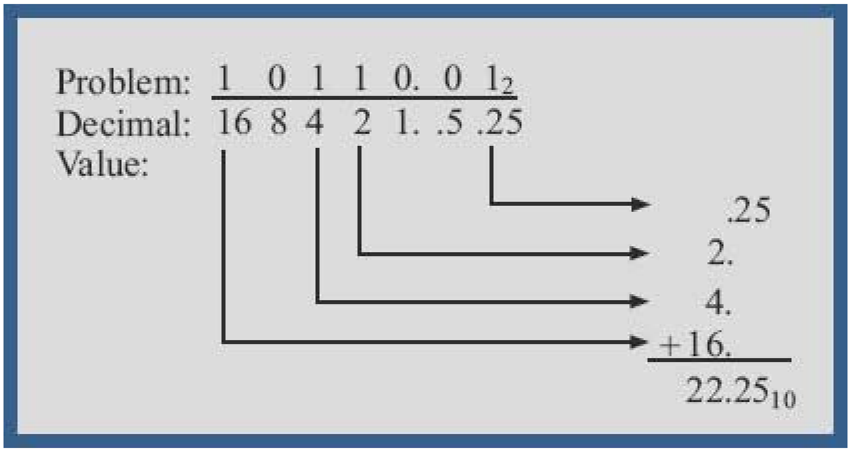
Another method of determining the decimal value of a position is to multiply the preceding value by 2 for whole numbers and to divide the preceding value by 2 for fractional numbers, as shown below:



Let's convert a binary number to decimal by using the positional notation method. First, write out the number to be converted; then, write in the decimal equivalent for each position with a 1 indicated. Add these values to determine the decimal equivalent of the binary number. Look at our example:



You may want to write the decimal equivalent for each position as we did in the following example. Add only the values indicated by a 1.



You should make sure that the decimal values for each position are properly aligned before adding. For practice let's convert these binary numbers to decimal:

Q91. 10010_2

Q92. 1111100_2

Q93. 1010101_2

Q94. 0.0101_2

Q95. 0.1010_2

Q96. 1101101.1111_2

1.10.2 Octal to Decimal

Conversion of octal numbers to decimal is best done by the positional notation method. This process is the one we used to convert binary numbers to decimal.

First, determine the decimal equivalent for each position by multiplying 8 by itself the number of times indicated by the exponent. Set up a bar graph of the positions and values as shown below:

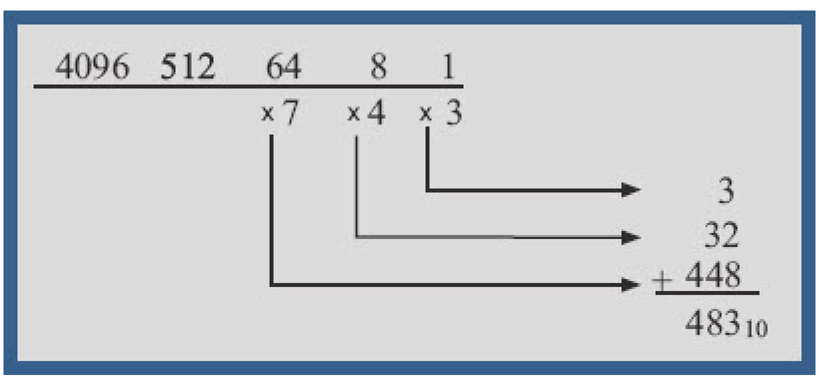
Positional Notation:	8^4	8^3	8^2	8^1	8^0	8^{-1}	8^{-2}	8^{-3}
Decimal Equivalent:	4096	512	64	8	1.	.125	.015625	.0019531

To convert an octal number to decimal, write out the number to be converted, placing each digit under the proper position.

Example:

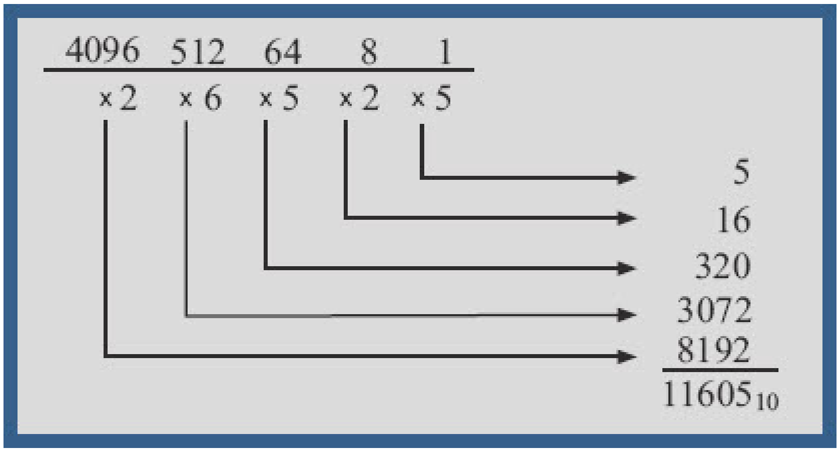
4096	512	64	8	1
		7	4	3_8

Next, multiply the decimal equivalent by the corresponding digit of the octal number; then, add this column of figures for the final solution:



Solution: 743_8 is equal to 483_{10}

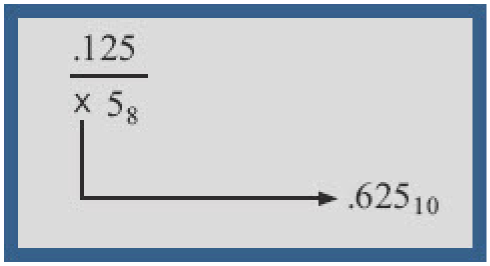
Now follow the conversion of 26525_8 to decimal:



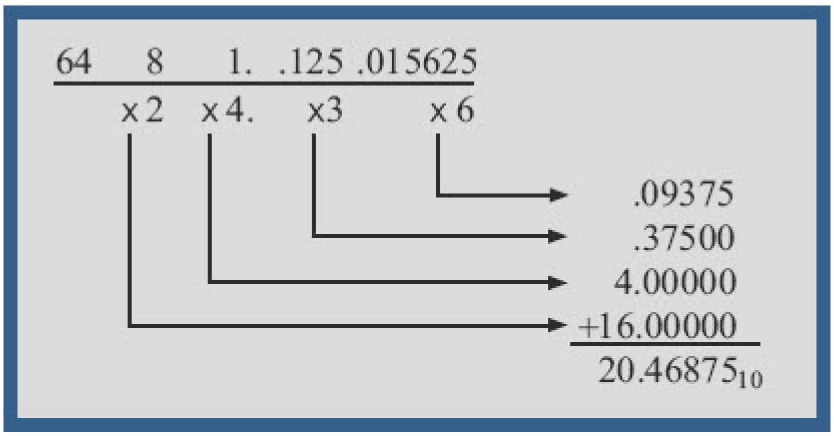
Solution: 11605_{10} is the decimal equivalent of 26525_8

To convert a fraction or a mixed number, simply use the same procedure.

Example: Change $.5_8$ to decimal:



Example: Convert 24.368 to decimal:



Solution: 24.368 equals 20.4687510

If your prefer or find it easier, you may want to convert the octal number to binary and then to decimal.

Convert the following numbers to decimal:

Q97. 17_8

Q98. 64_8

Q99. 375_8

Q100. 0.4_8

Q101. 0.61_8

Q102. 10.22_8

1.10.3 Hex to Decimal

It is difficult to comprehend the magnitude of a base 16 number until it is presented in base 10; for instance, $E0_{16}$ is equal to 224_{10}. You must remember that usually fewer digits are necessary to represent a decimal value in base 16.

When you convert from base 16 to decimal, you may use the positional notation system for the powers of 16 (a bar graph). You can also convert the base 16 number to binary and then convert to base 10.

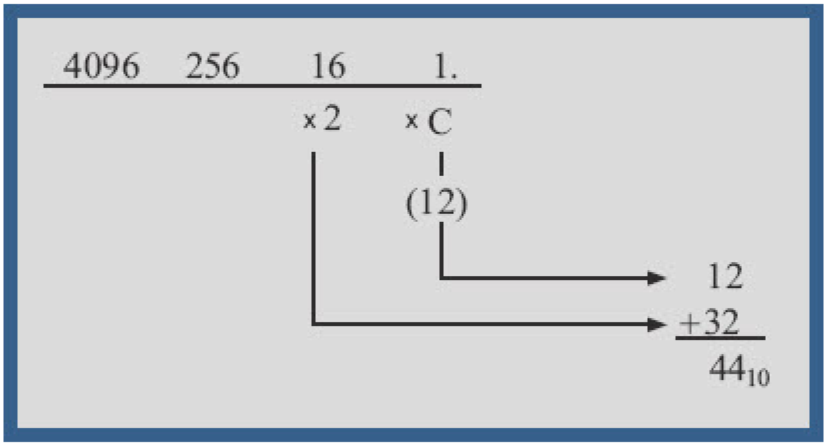
Note in the bar graph below that each power of 16 results in a tremendous increase in the decimal equivalent. Only one negative power (16^{-1}) is shown for demonstration purposes:

Radix Point

Positional Notation:	16^4	16^3	16^2	16^1	16^0.	16^{-1}
Decimal Equivalent:	65.536	4.096	256	16	1.	.0625

Just as you did with octal conversion, write out the hex number, placing each digit under the appropriate decimal value for that position. Multiply the decimal value by the base 16 digit and add the values. (Convert A through F to their decimal equivalent before multiplying). Let's take a look at an example.

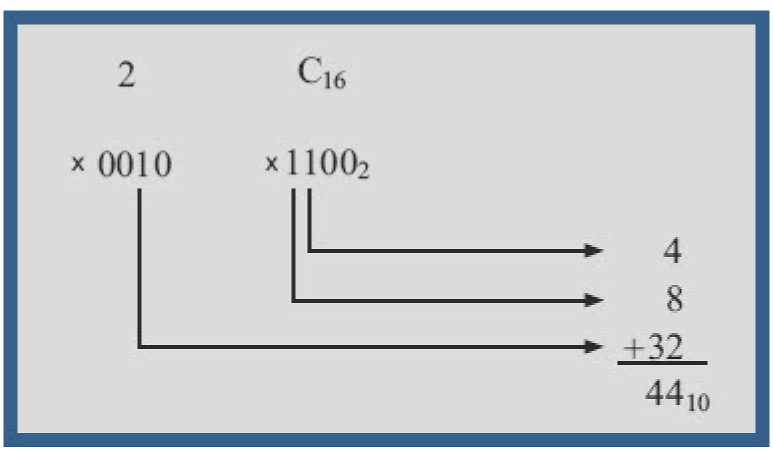
Convert $2C_{16}$ to decimal:



The decimal equivalent of $2C_{16}$ is 44_{10}.

Use the same procedure we used with binary and octal to convert base 16 fractions to decimal.

If you choose to convert the hex number to binary and then to decimal, the solution will look like this:



Convert these base 16 numbers to base 10:

Q103.	24_{16}
Q104.	$A5_{16}$
Q105.	DB_{16}
Q106.	$3E6.5_{16}$

1.11 BINARY-CODED DECIMAL

In today's technology, you hear a great deal about microprocessors. A microprocessor is an integrated circuit designed for two purposes: data processing and control.

Computers and microprocessors both operate on a series of electrical pulses called words. A word can be represented by a binary number such as 101100011_2. The word length is described by the number of digits or BITS in the series. A series of four digits would be called a 4-bit word and so forth. The most common are 4-, 8-, and 16-bit words. Quite often, these words must use binary-coded decimal inputs.

Binary-coded decimal, or BCD, is a method of using binary digits to represent the decimal digits 0 through 9. A decimal digit is represented by four binary digits, as shown below:

BCD		Decimal
0000	=	0
0001	=	1
0010	=	2
0011	=	3
0100	=	4
0101	=	5
0110	=	6
0111	=	7
1000	=	8
1001	=	9

You should note in the table above that the BCD coding is the binary equivalent of the decimal digit.

Since many devices use BCD, knowing how to handle this system is important. You must realize that BCD and binary are not the same. For example, 49_{10} in binary is 110001_2, but 49_{10} in BCD is 01001001_{BCD}. Each decimal digit is converted to its binary equivalent.

1.11.1 BCD Conversion

You can see by the above table, conversion of decimal to BCD or BCD to decimal is similar to the conversion of hexadecimal to binary and vice versa.

For example, let's go through the conversion of 264_{10} to BCD. We'll use the block format that you used in earlier conversions. First, write out the decimal number to be converted; then, below each digit write the BCD equivalent of that digit:

2	6	4_{10}
0010	0110	0100_{BCD}

The BCD equivalent of 264_{10} is 001001100100_{BCD}. To convert from BCD to decimal, simply reverse the process as shown:

1001	1000	0011_{BCD}
9	8	3_{10}

1.11.2 BCD Addition

The procedures followed in adding BCD are the same as those used in binary. There is, however, the possibility that addition of BCD values will result in invalid totals. The following example shows this:

Add 9 and 6 in BCD:

	1001_{BCD}	=	9_{10}
	$+\,0110_{BCD}$	=	6_{10}
INVALID BCD →	1111		15_{10}

The sum 1111_2 is the binary equivalent of 15_{10}; however, 1111 is not a valid BCD number. You cannot exceed 1001 in BCD, so a correction factor must be made. To do this, you add 6_{10} (0110_{BCD}) to the sum of the two numbers. The "add 6" correction factor is added to any BCD group larger than 1001_2. Remember, there is no 1010_2, 1011_2, 1100_2, 1101_2, 1110_2, or 1111_2 in BCD:

1111	← INVALID BCD
0110_{BCD}	Add 6_{10}
$0001\ 0101$	← New BCD

The sum plus the add 6 correction factor can then be converted back to decimal to check the answer. Put any carries that were developed in the add 6 process into a new 4-bit word:

0001	0101_{BCD}
1	5_{10}

Now observe the addition of 60_{10} and 55_{10} in BCD:

$$\begin{array}{lcl} 60_{10} & = & 0110\ 0000_{BCD} \\ 55_{10} & = & \underline{0101\ 0101_{BCD}} \\ & & 1011\ 0101 \longleftarrow \text{INVALID BCD} \end{array}$$

In this case, the higher order group is invalid, but the lower order group is valid. Therefore, the correction factor is added only to the higher order group as shown:

$$\begin{array}{rl} 1011\ 0101 & \\ \underline{+\ 0110\ 0000} & \text{Add } 6_{10} \\ 0001\ 0001\ 0101_{BCD} & \end{array}$$

Convert this total to decimal to check your answer:

$$\begin{array}{ccc} 0001 & 0001 & 0101_{BCD} \\ \hline 1 & 1 & 5_{10} \end{array}$$

Remember that the correction factor is added only to groups that exceed 9_{10} (1001_{BCD}).

Convert the following numbers to BCD and add:

Q107.

$$\begin{array}{r} 3_{10} \\ +\ 5_{10} \\ \hline \end{array}$$

Q108.

$$\begin{array}{r} 1_{10} \\ +\ 8_{10} \\ \hline \end{array}$$

Q109.

$$\begin{array}{r} 7_{10} \\ +\ 4_{10} \\ \hline \end{array}$$

Q110.

$$\begin{array}{r} 14_{10} \\ +\ 8_{10} \\ \hline \end{array}$$

1.12 SUMMARY

Now that you've completed this chapter, you should have a basic understanding of number systems. The number systems that were dealt with are used extensively in the microprocessor and computer fields. The following is a summary of the emphasized terms and points found in the "Number Systems" chapter.

The **UNIT** represents a single object.

A **NUMBER** is a symbol used to represent one or more units.

The **RADIX** is the base of a positional number system. It is equal to the number of symbols used in that number system.

A **POSITIONAL NOTATION** is a system in which the value or magnitude of a number is defined not only by its digits or symbol value, but also by its position. Each position represents a power of the radix, or base, and is ranked in ascending or descending order.

The **MOST SIGNIFICANT DIGIT (MSD)** is a digit within a number (whole or fractional) that has the largest effect (weighing power) on that number.

The **LEAST SIGNIFICANT DIGIT (LSD)** is a digit within a number (whole or fractional) that has the least effect (weighting power) on that number.

The **BINARY NUMBER SYSTEM** is a base 2 system. The symbols 1 and 0 can be used to represent the state of electrical/electronic devices. A binary 1 may indicate the device is active; a 0 may indicate the device is inactive.

The **OCTAL NUMBER SYSTEM** is a base 8 system and is quite useful as a tool in the conversion of binary numbers. This system works because 8 is an integral power of 2; that is, $2_3 = 8$. The use of octal numbers reduces the number of digits required to represent the binary equivalent of a decimal number.

The **HEX NUMBER SYSTEM** is a base 16 system and is sometimes used in computer systems. A binary number can be converted directly to a base 16 number if the binary number is first broken into groups of four digits.

The basic rules of **ADDITION** apply to each of the number systems. Each system becomes unique when carries are produced.

SUBTRACTION in each system is based on certain rules of that number system. The borrow varies in magnitude according to the number system in use. In most computers, subtraction is accomplished by using the complement (R's or R's-1) of the subtrahend and adding it to the minuend.

To **CONVERT A WHOLE BASE 10 NUMBER** to another system, divide the decimal number by the base of the number system to which you are converting. Continue dividing the quotient of the previous division until it can no longer be done. Extract the remainders -the remainder from the first computation will yield the LSD; the last will provide the MSD.

To **CONVERT DECIMAL FRACTIONS**, multiply the fraction by the base of the desired number system. Extract those digits that move to the left of the radix point. Continue to multiply the fractional product for as many places as needed. The first digit left of the radix point will be the MSD, and the last will be the LSD. The example to the right shows the process of converting 248.32_{10} to the octal equivalent (370.243_8).

BINARY numbers are converted to **OCTAL** and **HEX** by the grouping method. Three binary digits equal one octal digit; four binary digits equal one hex digit.

To **CONVERT** binary, octal, and hex numbers to DECIMAL use the POWERS of the base being converted.

BINARY-CODED DECIMAL (BCD) is a coding system used with some microprocessors. A correction factor is needed to correct invalid numbers

ANSWERS TO QUESTIONS Q1. THROUGH Q110.

A1. Unit

A2. Number

A3. Arabic

A4. The number of symbols used in the system

A5. 173_{10}

A6. 10^3, 10^2, 10^1, 10^0,

A7. Radix point

A8.

(a) MSD -4, LSD -0

(b) MSD -1, LSD -6

(c) MSD -2, LSD -4

(d) MSD -2, LSD -1

A9. 11111_2

A10. 11101_2

A11. 100001_2

A12. 101111_2

A13. 1000_2

A14. 11011110_2

A15. 10000_2

A16. 1011_2

A17. 11101_2

A18. 11_2

A19. 1110_2

A20. 11111_2

A21. 221_{10}

A22. 01100011_2

A23. -0001_2

A24. 10_8

A25. 60_8

A26. 1015_8

A27. 22306_8

A28. 151_8

A29. 24_8

A30. 321_8

A31. 36_8

A32. 336_8

A33. 377_8

A34. 104_8

A35. 7767_8

A36. $DD8D_{16}$

A37. $11FDB_{16}$

A38. $125F_{16}$

A39. 12020_{16}

A40. $191AB_{16}$

A41. $1AA8_{16}$

A42. 335_{16}

A43. 935_{16}

A44. 9531_{16}

A45. $36B3_{16}$

A46. $10ABC_{16}$

A47. $42F0F_{16}$

A48. 1001000_2

A49. 1100001_2

A50. 11110011_2

A51. 0.1110_2

A52. 0.0101_2

A53. 10001.01101_2

A54. 7_8

A55. 53_8

A56. 763_8

A57. 0.7467_8

A58. 0.00203_8

A59. 374.127_8

A60. $2A_{16}$

A61. 53_{16}

A62. $B0_{16}$

A63. $1EB_{16}$

A64. $0.B893_{16}$

A65. 2_8

A66. 12_8

A67. 57_8

A68. 0.14_8

A69. 0.63_8

A70. 67.25_8

A71. 2_{16}

A72. B_{16}

A73. $2F_{16}$

A74. 0.3_{16}

A75. $0.CC_{16}$

A76. 37.54_{16}

A77. 111011_2

A78. 101001010_2

A79. 100000011_2

A80. 0.100101110_2

A81. 0.111011_2

A82. 11110.101_2

A83. $3C_{16}$

A84. $14A_{16}$

A85. $0.0C_{16}$

A86. $C.88_{16}$

A87. 1000112; 43_8

A88. 110112; 33_8

A89. 0.1110012; 0.71_8

A90. 1000101.1012; 105.5_8

A91. 18_{10}

A92. 124_{10}

A93. 85_{10}

A94. 0.3125_{10}

A95. 0.625_{10}

A96. 109.9375_{10}

A97. 15_{10}

A98. 52_{10}

A99. 253_{10}

A100. 0.5_{10}

A101. 0.765625_{10}

A102. 8.28125_{10}

A103. 36_{10}

A104. 165_{10}

A105. 219_{10}

A106. 998.3125_{10}

A107. 1000_{BCD}

A108. 1001_{BCD}

A109. $0001\ 0001_{BCD}$

A110. $0010\ 0010_{BCD}$

APPENDIX A

GLOSSARY

ADDEND - A number to be added to an augend.

ADDITION - A form of counting where on quantity is added to another.

AND GATE - A logic circuit in which all inputs must be HIGH to produce a HIGH output.

ASSOCIATIVE LAW - A simple equality statement A(BC) = ABC or A+(B+C) = A+B+C.

AUGEND - A number to which another number is to be added.

BASE - The number of symbols used in the particular number system.

BCD (BINARY CODED DECIMAL) - A method of using binary digits to represent the decimal digits 0 through 9.

BINARY SYSTEM - The base 2 number system using 0 and 1 as the symbols.

BOOLEAN ALGEBRA - A mathematical concept based on the assumption that most quantities have two possible conditions - TRUE and FALSE.

BOOLEAN EXPRESSION - A description of the input or output conditions of a logic gate.

BORROW - To transfer a digit (equal to the base of the number system) from the next higher order column for the purpose of subtraction.

CARRY - A carry is produced when the sum of two or more numbers in a vertical column equals or exceeds the base of the number system in use.

CLOCK - A circuit that generates timing control signals in a computer or other type of digital equipment.

COMMUTATIVE LAW - The order in which terms are written does not affect their value; AB = BA, A+B =B+A.

COMPATIBILITY - The feature of logic families that allows interconnection of circuits without the need for additional circuitry.

COMPLEMENT - Something used to complete something else.

COMPLEMENTARY LAW - A term ANDed with its complement is 0, and a term ORed with its complement is 1; $A\overline{A} = 0$, $A + \overline{A} = 1$.

CONVERSION - To change a number in one base to its equivalent in another base.

COUNTER -A device that counts.

D FLIP-FLOP - Stores the data bit (D) in conjunction with the clock input.

DECADE COUNTER - Counter from 0 to 10_{10} in base 2, then resets.

DECIMAL POINT - The radix point for the decimal system.

DECIMAL SYSTEM - A number system with a base or radix of 10.

DEMORGAN'S THEOREM - This theorem has two parts: the first states that $\overline{AB} = \overline{A} + \overline{B}$ the second states that $\overline{A + B} = \overline{A}\,\overline{B}$.

DIFFERENCE - That which is left after subtraction.

DISTRIBUTIVE LAW - (1) a term (A) ANDed with a parenthetical expression (B+C) equals that term ANDed with each term within the parenthesis: A(B+C) = AB+AC; (2) a term (A) ORed with a parenthetical expression (BC) equals that term ORed with each term within the parenthesis: A+(BC) = (A+B)(A+C).

DIVIDEND - A number to be divided.

DIVISOR - A number by which a dividend is divided.

DOUBLE NEGATIVE LAW - A term that is inverted twice is equal to the term; $\overline{\overline{A}} = A$.

DOWN COUNTER - A circuit that counts from a predetermined number down to 0.
EXCLUSIVE-NOR (X-NOR) - A logic circuit that produces a HIGH output when all inputs are LOW or all inputs are HIGH.

EXCLUSIVE-OR (X-OR) GATE - A logic circuit that produces a HIGH output when one and only one input is HIGH.

EXPONENT - A number above and to the right of a base indicating the number of time the base is multiplied by itself; $2^4 = 2 \cdot 2 \cdot 2 \cdot 2$.

FLIP-FLOP - A bistable multivibrator.

FRACTIONAL NUMBER - A symbol to the right of the radix point that represents a portion of a complete object.

HEXADECIMAL (HEX) SYSTEM - The base 16 number system using 0 through 9 and A, B, C, D, E, and F as symbols.

IDEMPOTENT LAW - States that a term ANDed with itself or ORed with itself is equal to the term; AA = A, A+A = A.

INVERTER - A logic gate that outputs the complement of its input.

J-K FLIP-FLOP - Can perform the functions of the RS, T, and D flip-flops.

LAW OF ABSORPTION - This law is the result of the application of several other laws. It states that A(A+B) = A or A+(AB) = A.

LAW OF COMMON IDENTITIES - The two statements $A(\overline{A}+B) = AB$ and $A+\overline{A}B = A+B$ are based on the complementary law.

LAW OF IDENTITY - States that a term TRUE in one part of an expression will be TRUE in all parts of the expression; $A = A$, $\overline{A} = \overline{A}$.

LAW OF INTERSECTION - A term ANDed with 1 equals that term, and a term ANDed with 0 equals 0; A·1 = A, A· 0 = 0.

LAW OF UNION - A term ORed with 1 equals 1; a term ORed with 0 equals that term; A+1 = 1, A + 0 = 0.

LEAST SIGNIFICANT (LSD) - The digit which has the least effect on the value of a number.

LOGIC - The science of reasoning; the development of a reasonable or logical conclusion based on known information.

LOGIC FAMILY - A group of logic circuits based on specific types of circuit elements (DTL, TTL, CMOS, and so forth).

LOGIC GATES - Decision-making circuits in computers and other types of equipment.

LOGIC POLARITY - The polarity of a voltage used to represent the logic 1 state.

LOGIC SYMBOL - Standard symbol used to indicate a particular logic function.

MINUEND - The number from which another number is subtracted.

MIXED NUMBER - Represents one or more complete units and a portion of a single unit.

MODULUS - The number of different values that a counter can contain or display.

MOST SIGNIFICANT DIGIT (MSD) - The digit which if changed will have the greatest effect on the value of a number.

NAND GATE - An AND gate with an inverted output. The output is LOW when all inputs are HIGH, and HIGH when any or all inputs are LOW.

NEGATIVE LOGIC - The voltage representing logic state 1 is more negative that the voltage representing a logic state 0.

NEGATOR - See inverter.

NOR GATE - An OR gate with an inverted output. The output is LOW when any or all inputs are HIGH, and HIGH when all inputs are LOW.

NOT CIRCUIT - See inverter.

NUMBER - A symbol used to represent a unit or a quantity.

OCTAL SYSTEM - The base 8 number system using 0 through 7 as the symbols.

OR GATE - A logic circuit which produces a HIGH output when one or more inputs is/are HIGH.

PARALLEL DATA - Each bit of data has a separate line and all bits are moved simultaneously.

PARALLEL REGISTER - A register that receives, stores, and transfers data in a parallel mode.

POSITIONAL NOTATION - A method where the value of the number is defined by the symbol and the symbol's position.

POSITIVE LOGIC - The voltage representing logic state 1 is more positive than the voltage representing a logic state 0.

POWER OF A NUMBER - The number of times a base is multiplied by itself. The power of a base is indicated by the exponent; that is, $10^3 = 10 \cdot 10 \cdot 10$.

QUOTIENT - The result in division.

RADIX POINT - The symbol that separates whole numbers and fractional numbers.

RADIX - The total number of symbols used in a particular number system.

REGISTER - A circuit of flip-flops designed to receive, store, and transfer data.

REMAINDER - The final undivided part that is less than the divisor.

RING COUNTER - A loop in which only one flip-flop will be set at any given time; used in timing.

RIPPLE (ASYNCHRONOUS) COUNTER - A circuit that counts from 0 to a specified value. Subject to error at high frequency.

R's (RADIX) COMPLEMENT - The difference between a given number and the next higher power of the number system (1000_8 minus 254_8 equals 524_8).

R's-1 (RADIX-1) COMPLEMENT - The difference between a given number and the highest value symbol in the number system (777_8 minus 254_8 equals 524_8).

R-S FLIP-FLOP - A flip-flop with two inputs - S (set) and R (reset). The Q output is HIGH in the set mode and LOW in the reset mode.

SERIAL DATA - All data bits are transferred one bit at a time along a single conductor.

SHIFT REGISTER - A register capable of serial-to-parallel and parallel-to-serial conversion and scaling.

SHIFTING - Moving the contents of a register right or left to scale the number or to input or output serial data.

SUBSCRIPT - A number written below and to the right of a value indicating the base or radix of the number system in use (35_8).

SUBTRACTION - Taking away one number from another.

SUBTRAHEND - The quantity to be subtracted from the minuend.

SUM - The result in addition.

SYNCHRONOUS COUNTER - Performs the same function as a ripple counter but error free at high frequency.

T FLIP-FLOP - A single input flip-flop that changes state with each positive pulse or each negative pulse. Divides input frequency by two.

TRUTH TABLE - A chart showing all possible input combinations and the resultant outputs.

UNIT - A single object.

UP/DOWN COUNTER - A counter circuit that can count up or down on command.

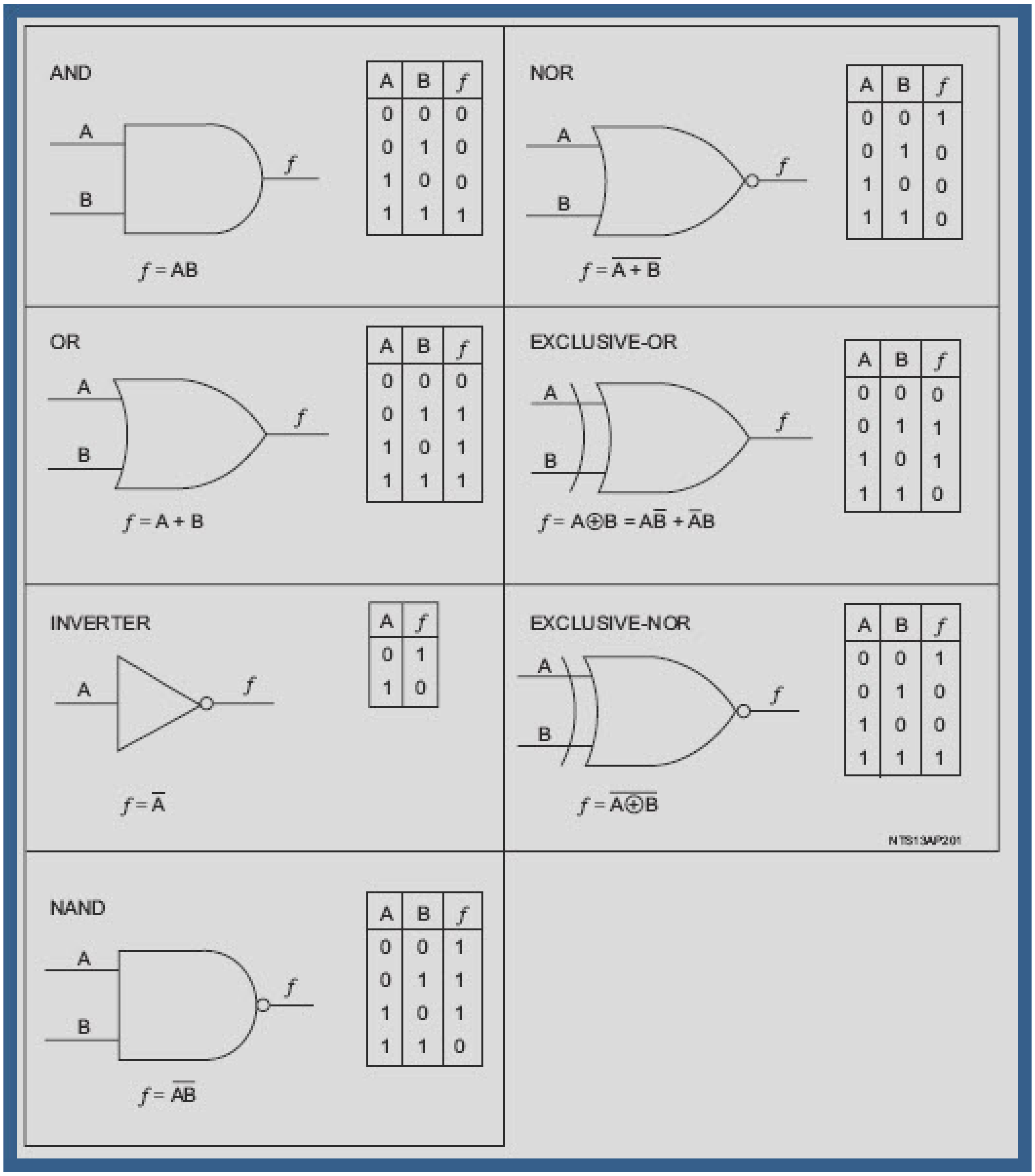
VINCULUM - A bar over a logic statement indicating the FALSE condition of the statement.

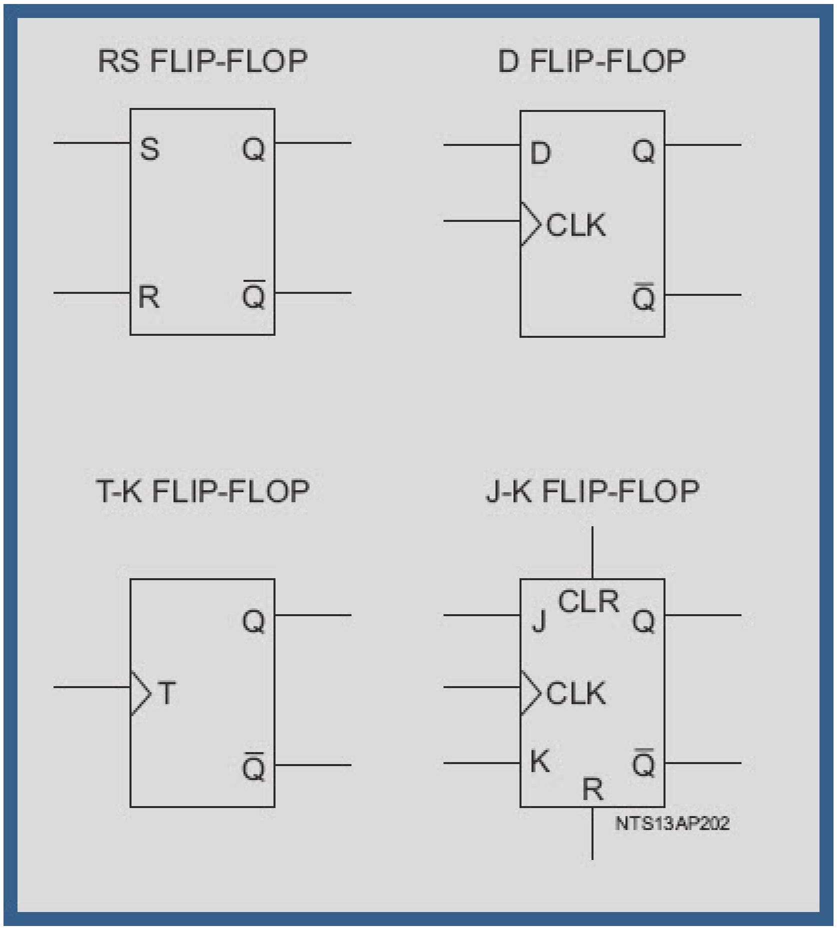
WHOLE NUMBER - A symbol that represents one or more complete objects.

ZERO - A symbol that indicates no numerical value for a position in positional notation.

APPENDIX B

LOGIC SYMBOLS




RS FLIP-FLOP
S
Q
R
$\overline{Q}$
D FLIP-FLOP
D
Q
CLK
$\overline{Q}$
T-K FLIP-FLOP
Q
T
$\overline{Q}$
J-K FLIP-FLOP
CLR
J
Q
CLK
K
$\overline{Q}$
R
NTS13AP202