

# Complete Description of Well-known Number Systems using Single Table

\*Shahid Latif, \*\*Junaid Qayyum, \*\*Muhammad Lal, \*\*Faheem Khan

\*Department of computer science & IT, \*\*Department of Telecommunication

\*Sarhad University of Science and Information Technology (SUIT), Peshawar Pakistan

\*\*Gandhara University of Sciences Peshawar Pakistan

[Shahid.csit@suit.edu.pk](mailto:Shahid.csit@suit.edu.pk)

[junaidkhan2311@ymail.com](mailto:junaidkhan2311@ymail.com)

[instructor4user@hotmail.com](mailto:instructor4user@hotmail.com)

[Kfaheem81@yahoo.com](mailto:Kfaheem81@yahoo.com)

## ABSTRACT

A number system is a set of rules and symbols used to represent a number, or any system used for naming or representing numbers is a number/numeral system. The modern society is familiar with decimal number system using ten digits. However digital devices and computers use binary number system instead of decimal, using only two digits i.e. 0 and 1 based on the fundamental concept of the decimal number system. Various other number systems also used this fundamental concept of decimal number system i.e. octal system and hexadecimal number system using eight and sixteen digits respectively. The awareness and concept of various number systems, their number representation, arithmetic operations, compliments and the inter conversion of numbers belongs to different number system is essential for understanding of computers. More over, successful programming for digital devices require the understanding of data formats (BCD, ASCII etc), number systems and their inter conversion (a process in which things are each converted into the other). Understanding all these number systems and particularly the inter conversion of numbers requires allot of time and techniques to expertise. In this paper the concepts of the four most common number systems, their representation, arithmetic, compliments and interconversion is taken under the consideration in tabulated form. It will provide an easy understanding and practising of these number systems to understand as well as memorise them. The four number systems are binary, octal, decimal and hexadecimal.

## General terms

Digital Electronics, Data Communication, Microprocessor, Digital Logic/Computer Design, Number systems Arithmetic

## Keywords

Number system, Binary, octal, hexadecimal, inter conversion, compliments

## 1. INTRODUCTION

As we do our computations since our childhood, using the numbers 0 - 9, the digits of the decimal number system. That's why we are so familiar to working with decimal numbers that we do not experience a need to think that how to use these numbers. However when we deals with computers we necessitate to be familiar that how a number will be used.

In the digital world when we deal with computer and information technology, normally we requires a working knowledge of various number systems, four of which are the most common such as binary, octal, decimal and hexadecimal. More specifically, the use of the microprocessor requires a working knowledge of binary, decimal and hexadecimal numbering system [4]. Computers communicate and operate in binary digits 0 and 1; on the other hand human beings generally use the decimal systems with ten digits 0-9. Other number systems are also used in digital systems, such as octal with eight digits i.e. 0-7 and hexadecimal system with digits from 0-15. In hexadecimal system, digits 10-15 are designated as A through F. Respectively to avoid confusion with the decimal numbers, 10 to 15

[5]. In computer science, sooner or later we will need to learn about number systems, and the involved mathematics. Commonly, the mathematics of number system refers to the simple properties when using the traditional operations of addition, subtraction, multiplication and division with smaller values of numbers, called arithmetic [2]. In other words, arithmetic is a branch of mathematics concerned with the addition, subtraction, multiplication, and division of the numbers.

In data communication a simple signal by itself does not carry information any more than a straight lines conveys words. The signal must be manipulated so that it contains identifiable changes that are recognizable to the sender and receiver as representing the information intended. First the information must be translates into agreed-upon patterns of 0s and 1s, for example, using ASCII. Also, data stored in the computer are in the form of 0s and 1s. To be carried from one place to another, data are usually converted to digital signals. Some times we need to convert an analog signal (such as voice in a telephone conversation) into a digital signal and vice versa [6]. So, in many applications we deal with the interconversion of number systems. There are various techniques that used for these inter conversions.

Remember, all number systems are interconvertable. But each conversion i.e. from one number system to another often takes place in a different way, using different techniques [1]. So it becomes very tedious for beginners to overcome this difficulty and understand these conversions in short time.

In case when data are stored in computer memory to represent negative numbers, compliments are used. Moreover, as Subtraction is the opposite of addition, so it also uses the compliments. Subtraction finds the difference between two numbers, the minuend minus the subtrahend. If the minuend is larger than the subtrahend, the difference is positive; if the minuend is smaller than the subtrahend, the difference is negative; if they are equal, the difference is zero. So for calculating these results, several methods are used, one of which is compliment technique particularly helpful to machine calculation, i.e. digital computers make use of the method of compliments (radix and radix-1 complements).

In this particular paper, we are taking under the consideration a tabulated format for all these four number systems. It covers each system's number representation, their arithmetic, complete interconversion of numbers from one to another system with in only three steps (while, in earlier approaches we have to use more than 20 steps to perform all these conversions), and their compliments. Using this table the whole concepts will be cover in approximately two or three contact hour of the lecture.

This paper is organized in such a way that it consist of seven sections. Section one covers the brief introduction of the number systems, need of number systems, their representation, arithmetic, compliments and their interconversion and easy approach to it. Section two is the overview of the number systems and their representations. Section three covers the arithmetic of all four

number systems. Section four describes all the conversion techniques (both for integral and fractional part of the numbers) frequently used so far. Section five contains the compliment concepts for subtraction. Section six contains the proposed tabulated form as introduced earlier, while last one section concludes the paper.

**2. OVERVIEW OF NUMBER SYSTEMS**

When humans are speaking to one another, they speak in a particular language. This language is made of words and letters. Although we type words and letters in the computer, the computer does not understand the words and letters. Rather, those words and letters are translated into numbers. Computers “talk” and understand in numbers. Although many students know the decimal (base 10) system, and are very comfortable with performing operations using this system, it is important for students to understand that the decimal system is not the only system. By studying other number systems such as binary (base 2), octal (base 8), and hexadecimal (base 16), students will gain a better understanding of how number systems work in general.

**2.1 Digits**

Before the conversion concepts of numbers from one number system to another, the digit of a number system must be understood. The first digit in any numbering system is always a zero. For example, a base 2 (binary) numbers contains 2 digits: 0 and 1, a base 8 (octal) numbers contains 8 digits: 0 through 7 and so on. Note that a base 10 (decimal) numbers does not contain the digit 10, similarly base 8 numbers does not contain a digit 8. Same is the case for the other number systems. Once the digits of a number system are understood, larger numbers can be constructed using positional notation or place-value notation method. According to the positional notation method, in decimal numbers the first right most digit (integer) has a units position. Further, to the left of the units position is the tens position, the position to the left of the tens position is the hundreds position and so forth. Here, the units position has a weight of  $10^0$ , or 1; the tens position has a weight of  $10^1$ , or 10; and the hundreds position has a weight of  $10^2$ , or 100. The exponential powers of the positions are critical for understanding numbers in other numbering systems. Remember the position to the left of the radix point is always the unit’s position in any number system. For example the position to the left of the binary point is always  $2^0$ , or 1; the position to the left of the octal point is always  $8^0$ , or 1 and so on. The position to the left of the unit’s position is always the number whose base is raised to the first power; i.e.  $2^1$ ,  $8^1$  and so on.

**2.2 Number representation**

A number in any base system can be represented in a generalized format as follows:

$$N = A_n B^n + A_{n-1} B^{n-1} + \dots + A_1 B^1 + A_0 B^0, \text{ where}$$

$N = \text{Number}, B = \text{Base}, A = \text{any digit in that base}$

For example number 154 can be represented in various number systems as follows:

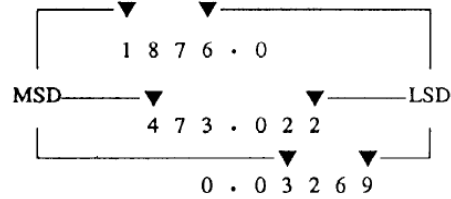
Table: 2.1 Number representations in various number systems

Decimal	154	$1 \times 10^2 + 5 \times 10^1 + 4 \times 10^0$ $= 100 + 50 + 4$	154
Binary	10011010	$1 \times 2^7 + 0 \times 2^6 + \dots + 0 \times 2^0$ $= 128 + 0 + 0 + 16 + 8 + 0 + 2 + 0$	154
Octal	232	$2 \times 8^2 + 3 \times 8^1 + 2 \times 8^0$ $= 128 + 24 + 2$	154
H-decimal	9A	$9 \times 16^1 + A \times 16^0 = 144 + 10$	154

**2.3 Most Significant Digit and Least Significant Digit**

The MSD in a number is the digit that has the greatest effect on that number, while The LSD in a number is the digit that has the least effect on that number.

Look at the following examples:

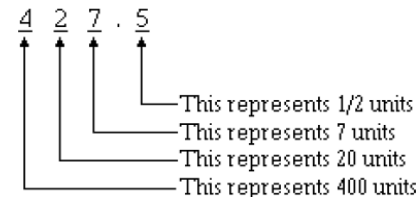


You can easily see that a change in the MSD will increase or decrease the value of the number in the greatest amount, while changes in the LSD will have the smallest effect on the value.

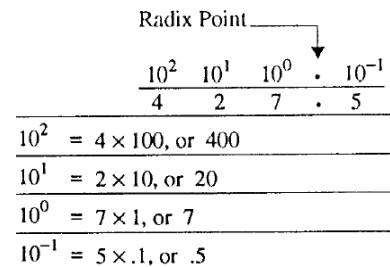
**2.4 Decimal Number System**

The decimal number system is known as international system of numbers [17]. It is also called base ten or occasionally denary number system. It has ten as its base. It is the numerical base most widely used by modern civilization [10].

Decimal notation often refers to a base-10 positional notation; however, it can also be used more generally to refer to non-positional systems. Positional decimal systems include a zero and use symbols (called digits) for the ten values (0, 1, 2, 3, 4, 5, 6, 7, 8, and 9) to represent any number, no matter how large or how small. Let’s examine the decimal (base 10) value of 427.5. You know that this value is four hundred twenty-seven and one-half. Now examine the position of each number:



Each digit has its own value (weight) as described in the above figure. 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. All numbers to the left of the decimal point are whole numbers or integers, and all numbers to the right of the decimal point are fractional numbers.

**2.5 Binary Number System**

The number system with base (or radix) 2, is known as the binary number system. Only two symbols are used to represent numbers in this system and these are 0 and 1, these are known as bits. It is a positional system i.e. every position is assigned a specific weight. Moreover, it has two parts the Integral part or integers and the fractional part or fractions, set a part by a radix point. For example (1101.101)<sub>2</sub> In binary number system the left–most bit is known as most significant bit (MSB) and the right–most bit is known as the least

significant bit (LSB), similar to decimal number system. The following graph shows the position and the power of the base (2 in this case):

$$2^3 \quad 2^2 \quad 2^1 \quad 2^0 \quad . \quad 2^{-1} \quad 2^{-2} \quad 2^{-3}$$

The arithmetic operations such as addition, subtraction, multiplication and division of decimal numbers can be also performed on binary numbers. Also binary arithmetic is much simpler than decimal arithmetic because here only two digits, 0 and 1 are involved.

### 2.6 Octal Number System

As its name reveal (octa = 8), the number system with base 8 is known as the octal number system. In this system eight symbols, 0, 1, 2, 3, 4, 5, 6, and 7 are used to represent the number. Hence, any octal number can not have any digit greater than 7 [3]. Similar to decimal and binary number systems, it is also a positional system; the octal number system uses power of 8 to determine the value of a number's position. The following graph shows the positions and the power of the base (8 in this case):

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

Octal number has also two parts: Integral and fractional, set a part (separated) by a radix point, for example (6327.4051)<sub>8</sub>

The main advantage of using the octal number system is that, in any digital transmission system it is highly tedious to handle long strings of binary numbers. It may also cause errors. Therefore, octal numbers are used for entering binary data and displaying certain information in short.

### 2.7 Hexadecimal Numbering System

Hexadecimal number system is very popular in computer uses. The base for hexadecimal number system is 16 which require 16 distinct symbols to represent the number. These are numerals 0 through 9 and alphabets A through F [18]. This is an alphanumeric number system because it uses both alphabets and numerical to represent a hexadecimal number. Hexadecimal number system use 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, and F. Any number in hexadecimal number system can be represented as (B52.AC3)<sub>16</sub>.

It has also two parts i.e. integral and fractional. Like the binary, octal, and decimal systems, the hexadecimal number system is a positional system. Powers of 16 are used for the positional values of a number. The following graph shows the positions and power of the base (16 in this case):

$$16^3 \quad 16^2 \quad 16^1 \quad 16^0 \quad . \quad 16^{-1} \quad 16^{-2} \quad 16^{-3}$$

The most significant and least significant digits will be determined in the same manner as the other number systems.

## 3 ARITHMETIC OF NUMBER SYSTEMS

The arithmetic is the most basic branch of mathematics, used by almost everyone from simple day-to-day counting to advanced science and business calculations. It simply refers to the basic mathematical operation such addition, subtraction, multiplication and division. As at the present, binary number system is the most common number system used by computer systems. However, long ago, there were such computer systems which were based on the decimal (base 10) number system rather than the binary number system. Systems designers have discovered that binary arithmetic is almost better than the decimal arithmetic for general calculations, but still decimal arithmetic is used in many software systems that specify the use of decimal arithmetic in their calculations. Therefore, despite the truth that decimal arithmetic is generally inferior to binary arithmetic; the need for decimal arithmetic still persists.

Remember that the arithmetic operations such as addition, subtraction, multiplication and division of decimal numbers can be

performed on binary, octal and hexadecimal numbers as well. Therefore, in this section of the paper only the decimal arithmetic is described. However, the binary arithmetic is much simpler than all the other number system's arithmetic, because here only two digits, 0 and 1 are involved.

### 3.1 Decimal arithmetic

**Addition:** In decimal numbers addition, a one quantity is added to another (for example 5+7=12). The basic terms of addition are:

**AUGEND:** The quantity to which an addend is added (first number i.e. 5 in this example)

**ADDEND:** A number to be added to a earlier number (second number i.e. 7 in this example)

**SUM:** The result of an addition (i.e. 12, the sum of 5 and 7)

**CARRY:** A carry is produced when the sum of two or more digits equals or exceeds the base of the number system in use.

**Subtraction:** Subtraction is the opposite of addition. Subtraction finds the difference between two numbers, the minuend minus the subtrahend. In other words, subtraction means to take away a part from the whole number or one number from another number. If the minuend is larger than the subtrahend, the difference is positive; if the minuend is smaller than the subtrahend, the difference is negative; if they are equal, the difference is zero. For example: 25-7 =18

The basic terms of subtraction are:

**MINUEND:** The number from which another number is to be subtracted (i.e. 25 in the above example)

**SUBTRAHEND:** The number to be subtracted (i.e. 7 here)

**REMAINDER or DIFFERENCE:** That number which is left after subtraction (i.e. 18 here)

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

**Multiplication:** Multiplication is also one of the basic operations of arithmetic. It also combines two numbers into a single number, called "product". In this arithmetic operation simply multiply the multiplicand by each digit of the multiplier and then add up all the properly shifted results.

For example: 32 × 8

The basic terms of multiplication are:

**MULTIPLIER:** The number by which another number is multiplied (i.e. 8 is the multiplier in above example)

**MULTIPLICAND:** The number that is to be multiplied by another. Here the multiplicand is 32

**PRODUCT:** The number or quantity obtained by multiplying two or more numbers together, i.e. 32 × 8 = 256

Binary, octal and hexadecimal multiplication is similar to decimal multiplication except that base and counting is changed accordingly. Each bit of the multiplier (2<sup>nd</sup> number), multiplies to the whole multiplicand number (1<sup>st</sup> number).

**Division:** Division is basically the opposite of multiplication. Division obtains the quotient of two numbers, when the dividend is divided by the divisor. Any dividend divided by zero is undefined. If the dividend is larger than the divisor, the quotient is greater than 1 otherwise it is less than 1. While in reverse if the quotient is multiplied by the divisor, it always yields the dividend. For example: 45 ÷ 3 = 15

The basic terms of division are:

**DIVIDER:** One number that divides another number (i.e. 3 here)

**DIVIDEND:** A number to be divided (i.e. 45 here)

**QUOTIENT:** The number obtained by dividing one number by another (i.e. 15)

**REMAINDER:** The number left over when one number is divided by another (in this example remainder is 0)

Binary, octal and hexadecimal division is obtained using the same procedure like decimal division except that base and counting is changed accordingly.

#### 4 CONVERSION BETWEEN NUMBER SYSTEMS

As we have discussed so far, the most common number systems are the decimal, binary, octal and hexadecimal. Now we have to check that how any number can be converted from one to another number system. Number systems are given in the ascending order as,

Binary  
Octal  
Decimal  
Hexadecimal

A given number in any of the above number systems may consist of two parts i.e. the integral part and the fractional part. Each part some times, required a different technique for conversion. In other words, in case of fractions the conversion process requires additional techniques. So as a whole, more than 20 various techniques are used for these inter conversions, which are enlisted below.

##### Integral part of a numbers

Binary – to – octal  
Octal – to – binary  
Binary – to – decimal  
Decimal – o – binary  
Binary- to – hexadecimal  
Hexadecimal – to – binary  
Octal – to – decimal  
Decimal –to – octal  
Octal –to – hexadecimal  
Hexadecimal –to- octal  
Decimal – to – hexadecimal  
Hexadecimal –to – decimal

##### Fractions

Binary fraction – to – octal  
Octal fraction– to – binary  
Binary fraction – to – decimal  
Decimal fraction – to – binary  
Binary fraction - to – hexadecimal  
Hexadecimal fraction – to – binary  
Octal fraction – to – decimal  
Decimal fraction –to – octal  
Octal fraction –to- hexadecimal  
Hexadecimal fraction –to- octal  
Decimal fraction – to – hexadecimal  
Hexadecimal fraction –to – decimal

It is obvious, that the beginners will be very dishearten and frustrated to use such a lot of techniques for the conversion in short time of one contact hour or so. To overcome this problem we present a very easy approach to the complete inter conversion of numbers as given under.

##### Complete inter conversion in 3-steps

We know that the decimal number system is the most common of the above mentioned number systems, because it is widely used in mathematics and our daily life calculations based on this number system, so we start the conversion from the decimal number system to the remaining systems. Here a word “other” is used for those number systems which are other than the consider one. We can perform the conversion between different number systems in three steps,

Step: 1 A) From Decimal number system  $\rightarrow$  to  $\rightarrow$  other number systems [Binary, Octal, Hexadecimal]  
B) From Other number systems [Binary, Octal, Hexadecimal]  $\rightarrow$  to  $\rightarrow$  Decimal number system

In step: 1 all the conversion processes related to decimal number system are covered. So we will not use the conversion from/to decimal number system to/from others, anymore.

Step: 2 A) From Binary number system  $\rightarrow$  to  $\rightarrow$  other [Octal, Hexadecimal] number systems  
B) From Other [Octal, Hexadecimal] number systems  $\rightarrow$  to  $\rightarrow$  Binary number system

In step: 2 all the conversion processes related to binary number system are covered. So we will not use the conversion from/to binary number system to/from others, anymore.

Step: 3 A) From octal number system  $\rightarrow$  to  $\rightarrow$  hexadecimal number systems  
B) From hexadecimal number systems  $\rightarrow$  to  $\rightarrow$  octal number system

##### Step: 1

A) Conversion from Decimal number system  $\rightarrow$  to  $\rightarrow$  other number systems [Binary, Octal, Hexadecimal]

To convert a given number from decimal number system to any other number system, follow these steps:

1. Divide the decimal number by r i.e. base of the other system (2, 8, or 16). Remember the quotient and the remainder of this division.
2. After that, divide the quotient (from the first division) by r, again remembering the quotient and the remainder.
3. Keep dividing your new quotient by r until you get a quotient of 0. After each division, keep track of the remainder.
4. When you reach a quotient of 0, the remainders of all the divisions (written in reverse order) will be the equivalent number in base r number system.

[Reverse order mean that, the first remainder that you got in step-1 will be the least significant digit (LSD) of the number in base r number system].

In case of the fractions or the fractional part in a given number the repeated multiplication method is used. In this method the fractional part of the number is multiplied by the base. Here is given a simple, step-by-step technique for computing the expansion on the right-hand side of the radix point.

1. Start with the decimal fraction given in a number (say .625) and multiply by base (2, 8 or 16). The whole number part of the result is the first binary digit to the right of the point. I.e.  $0.625 \times 2 = 1.25$ , So now we have .625 = .1--- (in base 2).

2) Next without involving the whole number part of the previous result (the 1 in this case) and multiply by 2 once again. The whole number part of this new result is the second digit to the right of the point.

3) Continue this process until we get a zero as our decimal part or up to required number beyond the radix point. Hence the representation of .625 = .101 (in base 2)

B) Conversion from Other number systems [binary, octal, hexadecimal]  $\rightarrow$  to  $\rightarrow$  decimal number system

The conversion process from other number systems [i.e. binary, octal, and hexadecimal] to decimal number system has the same procedure. Here any given number can be converted into its equivalent decimal number using the weights assigned to each bit position. Incase of binary the weights are  $2^0$  (Units),  $2^1$  (twos),  $2^2$  (fours),  $2^3$  (eights),  $2^4$  (sixteen) and so on. Similarly in case of octal the weights are  $8^0$ ,  $8^1$ ,  $8^2$ ,  $8^3$ ,  $8^4$  and so on. For hexadecimal the weights are  $16^0$ ,  $16^1$ ,  $16^2$ ,  $16^3$ ,  $16^4$  and so on. Here few steps are given which are helpful in faster and easy conversion of other systems to decimal number system.

1. Write the given (i.e. 2, 8, or 16) base number
2. Write the corresponding weight  $x^0$ ,  $x^1$ ,  $x^2$ ,  $x^3$ , ..., under each digit.

3. Cross out any weight under a 0 (means that any 0 involve in given number).  
4. Add the remaining weights.

In case of converting the fractions or the fractional part in a given number to the decimal representation, same procedure is used as mentioned above. The only difference is that the negative weights are assigned to each bit position instead of positive weights. For example in case of binary the weights are  $2^{-1}$ ,  $2^{-2}$ ,  $2^{-3}$ ,  $2^{-4}$ , and so on. Similarly in case of octal the weights are  $8^{-1}$ ,  $8^{-2}$ ,  $8^{-3}$ ,  $8^{-4}$ , and so on.

### Step: 2

A) Conversion from Binary number system  $\rightarrow$  to  $\rightarrow$  other [Octal, Hexadecimal] number systems

A binary number can be converted to octal number and hexadecimal number by replacing method. The binary digits are grouped by threes (in case of converting to octal) and fours (in case of converting to hexadecimal) respectively by starting from the decimal point and proceeding to the left and to the right. Add leading 0s (or trailing zeros to the right of decimal point) to fill out the last group of three or four if needed. Then replace group of three or group of four with the equivalent octal or hexadecimal digit accordingly.

B) Conversion from Other [Octal, Hexadecimal] number systems  $\rightarrow$  to  $\rightarrow$  Binary number system

For some computers to accept octal or hexadecimal data, the octal or hexadecimal digits must be converted to binary. This process is the reverse of binary to octal and hexadecimal conversion. To convert a given (octal or hexadecimal) number to binary, write out the number and then write below each digit the corresponding three-digit binary-coded octal equivalent (in case of converting from octal) or four-digit binary-coded hexadecimal equivalent (in case of converting from hexadecimal). A fraction in both octal and hexadecimal is converted to binary in the same manner. The below table shows the equivalent in other number system of first 16 (0 through 15) decimal numbers:

Table: 4.1 conversion table- Decimal, Hexadecimal, Octal, Binary

Dec	Hex	Oct	Bin
0	0	000	0000
1	1	001	0001
2	2	002	0010
3	3	003	0011
4	4	004	0100
5	5	005	0101
6	6	006	0110
7	7	007	0111
8	8	010	1000
9	9	011	1001
10	A	012	1010
11	B	013	1011
12	C	014	1100
13	D	015	1101
14	E	016	1110
15	F	017	1111

### Step: 3

A) Conversion from Octal number system  $\rightarrow$  to  $\rightarrow$  Hexadecimal number systems

The conversion from octal number system to hexadecimal number system is a two-step procedure using binary as an intermediate base. Octal is converted to binary and then binary to hexadecimal. When the conversion from octal done, then grouping the binary digits into groups of fours and add zeros where needed to complete groups, which correspond each to a hexadecimal digit. In case of converting

the fractions or fractional part of the given octal number to hexadecimal number system the same procedure i.e. used for the integral part will be also used here.

B) Conversion from Hexadecimal number systems  $\rightarrow$  to  $\rightarrow$  Octal number system

The conversion from hexadecimal number system to octal number system is the reversal of the same algorithm as declared in first part of step-3. Reverse the previous algorithm to achieve the conversion.

The following table reveals the conversion between the four number systems in three steps along the methods of conversion (for integers and fractions).

Remember, in step 2 and 3 conversion technique (method) for integral and fractional part is not mentioned in the table. It is so, because in both steps, same technique is used for both integers and fractions conversion.

Table: 4.2 Conversion between Decimal, Binary, Octal and Hexadecimal along the conversion techniques

Step No:	Part-A	Part-B
Step 1	Decimal to others [binary, octal, hexadecimal] $(==)_{10} \rightarrow (==)_{2,8,16}$ <b>Integer:</b> repeated division method <b>Fraction:</b> repeated multiplication method	Others [binary, octal, hexadecimal] to decimal $(==)_{2,8,16} \rightarrow (==)_{10}$ <b>Integer:</b> sum of [(+ve weights) $\times$ (integer)] <b>Fraction:</b> sum of [(-ve weights) $\times$ (fraction)]
Step 2	binary to other [octal, hexadecimal] $(==)_2 \rightarrow (==)_{8,16}$ <b>To octal:</b> replace group of 3-binary bits by octal digit <b>To hex:</b> replace group of 4-binary bits by hexadecimal digit (same method for both integral and fractional part)	Other [octal, hexadecimal] to binary $(==)_{8,16} \rightarrow (==)_2$ <b>From octal:</b> replace each octal digit by 3-bit binary <b>From hex:</b> replace each hexadecimal digit by 4-bit binary (same method for both integral and fractional part)
Step 3	octal to hexadecimal $(==)_8 \rightarrow (==)_{16}$ Direct conversion not applicable Octal $\rightarrow$ Binary $\rightarrow$ Hexadecimal	hexadecimal to octal $(==)_{16} \rightarrow (==)_8$ Direct conversion not applicable Hexadecimal $\rightarrow$ Binary $\rightarrow$ Octal

## 5 COMPLIMENTS

In computer science the complement is a technique used to subtract one number from another using addition of positive numbers. This method was commonly used in mechanical calculators and is still used in modern computers. For example to subtract a number B (the subtrahend) from another number A (the minuend), the radix complement of B is added to A and the initial '1' (i.e. MSB in case of binary or MSD = most significant digit in case of other number systems) of the result is discarded or added back to the result. Moreover, a number of times we store data in complemented form to represent negative numbers. Indeed, two's complement is used in most modern computers to represent signed numbers.

**Decimal compliments:** To subtract a decimal number B from another decimal number A, two compliment methods may be use i.e. ten's (radix) compliment method and nine's (radix-1) compliment method.

To understand the decimal compliments, let try an example by subtracting 218 from 873 (i.e.  $A-B=873-218$ ). The nines' complement of B is first obtained by determining the complement of each digit. The complement of a decimal digit in the nines' complement system is the number that must be added to it to produce 9, for example, the complement of 3 is 6, and the complement of 7 is 2, and so on. So nine's compliment of 218 is 781 (or  $999-218=781$ ). Next, sum the A and the nines' complement of B, i.e.  $873+781$ . If carry is generated then add it with result. I.e.  $873+781=1654$

Adding the carry to the result gives  $654+1=655$ .

For ten's compliment, first obtain the nine's compliment of B and then add 1 with the nine's compliment of B. this becomes the ten's compliment of B. Next sum the A with ten's compliment of B. If carry is generated then simply discard it. For example,  $A-B=873-218$  So ten's compliment of 218 is  $781+1=782$

Sum of A and complimented B is  $873+782=1655$

Now discard the carry and get the result i.e. 655

In case of binary, octal and hexadecimal numbers, both the radix and radix-1 compliments can be calculated in analogous to decimal compliments. These compliments can be realized with the help of the examples given in table 5.1.

Table: 5.1 Examples of the binary, octal and hexadecimal compliments

Number system	Subtraction Example	(Radix-1) Complement	Radix Compliment
Binary	$\begin{array}{r} 1001 \\ 0110 - \\ \hline 0011 \end{array}$	$\begin{array}{r} 1001 \\ 1001 + (1's) \\ \hline ①0010 \\ \text{↳} +1 \\ \hline 0011 \end{array}$	$\begin{array}{r} 1001 \\ 1010 + (2's) \\ \hline ①0011 \\ \text{Discard the MSB} = 1, \\ \text{so the result is } 0011 \end{array}$
Octal	$\begin{array}{r} 753 \\ 23 - \\ \hline 730 \end{array}$	$\begin{array}{r} 753 \\ 754 + (7's) \\ \hline ①727 \\ \text{↳} +1 \\ \hline 730 \end{array}$	$\begin{array}{r} 753 \\ 755 + (8's) \\ \hline ①730 \\ \text{Discard the MSD} = 1, \\ \text{so the result is } 730 \end{array}$
Hexa-decimal	$\begin{array}{r} B52 \\ 97 - \\ \hline ABB \end{array}$	$\begin{array}{r} B52 \\ F68 + (15,s) \\ \hline ①ABA \\ \text{↳} +1 \\ \hline ABB \end{array}$	$\begin{array}{r} B52 \\ F69 + (16's) \\ \hline ①ABB \\ \text{Discard the MSD} = 1, \\ \text{so the result is } ABB \end{array}$

## 6 TABULATED FORMATE

So far we have studied that how a number can be represented in various number systems, how mathematical operation can be perform on numbers of various systems, how a compliment of a number can be calculated and so on.

Now we are presenting the tabulated form, which covers these entire concepts in a single table. This table consist of four sections. Section one describes that how much digits a number system will use and how the number in any number system may be represented. Section two describes the arithmetic operations of four number systems with the help of simple examples. Section three covers the complete inter conversion of numbers along the techniques used for these conversions, and the last section of the table presents the compliment techniques for each number system. The table 6.1 is shown on the last page of the paper.

## CONCLUSION

In this particular paper we suggest a simple, short and easy approach (using a single table) to the entire number systems used in the digital technology specifically computing devices. This proposed table cover almost every thing associated to the four most common number systems. It envelops the number representation, allowed digits (symbols) in each number system, arithmetic of each number system; inter conversion of numbers, and the possible compliment

techniques in each number system. Remember that, these four number systems are not the only number systems used in digital world, but are the very common and frequently used in most of the digital technologies and devices.

The concept of these number systems and especially the complete inter conversion takes a lot of time to understand and memorize all the processes and techniques involved. From this paper we conclude that this is simply shorthand to the famous number systems and their interconversion, used in digital technology providing a rapid practice to the understanding, memorizing these systems, their arithmetic, interconversion between number systems and their compliments.

It will be very help full for those people who are new in the field of computer science or digital electronics. Moreover, this table can be used in the appendix of the digital related article, magazine and book as well. As a future work, the conversion table proposed in this paper may be enhanced by including more number systems. Also, as all the techniques used for conversion and compliments mentioned in this paper are almost old techniques, so newer conversion techniques can be added in it.

## ACKNOWLEDGMENT

Bundle of thanks to the International Journal of Engineering and Sciences (IJECS-IJENS) to give us the chance to discover and share our ideas with each and every one. I also pay appreciation to my family who support me and pray for me specially my brothers Zahid Latif (scholar of IR) and Khalid Latif (scholar of Geology).

## REFERENCES

- [1] Shahid Latif, Rahat Ullah, Hamid Jan "A Step towards an Easy Interconversion of Various Number Systems" International Journal of Engineering and Sciences (IJECS-IJENS), Volume: 11, No: 02
- [2] Davenport, Harold, *The Higher Arithmetic: An Introduction to the Theory of Numbers* (7th ed.), Cambridge University Press, Cambridge, UK, 1999, ISBN 0-521-63446-6
- [3] A. Saha and N. Manna "Digital Principals and Logic Design" ISBN: 978-0-7637-7373-1 Printed in United States of America
- [4] BARRY B, BREY *The Intel Microprocessors*, Sixth edition, prentice hall of India private limited, New Delhi – 110 001, 2002
- [5] Ramesh S. Gaonkar *Microprocessor Architecture, Programming, and Applications with the 8085/8080A* Published by H.S. Poplai for Wiley Eastern Limited, New Delhi ISBN: 0 85226 297 3
- [6] Behrouz A. Forouzan *Data Communication And Networking 2<sup>nd</sup> edition* Updated
- [7] Albert Paul Malvino, Jernald A. Brown *Digital Computer Electronics* third edition, GLENCOE Macmillan/ McGraw- Hill
- [8] Thomas L. Floyd *Digital Fundamentals*, 7<sup>th</sup> edition, 2000, 1997 Prentice-Hall, Inc. New jersey 07458
- [9] Number Representation and Computer Arithmetic, Article to appear in Encyclopedia of Information Systems, Academic Press, 2001.
- [10] *Histoire universelle des chiffres*, Georges Ifrah, Robert Laffont, 1994 (Also: *The Universal History of Numbers: From prehistory to the invention of the computer*, Georges Ifrah, ISBN 0471393401, John Wiley and Sons Inc., New York, 2000. Translated from the French by David Bellos, E.F. Harding, Sophie Wood and Ian Monk)
- [11] M. MORRIS MANO *Digital Logic and Computer Design 2<sup>nd</sup> edition*
- [12] RICHARD, R. K. *Arithmetic Operation in Digital Computers*, New York; Van Nost land Co., 1955.
- [13] FLORES, I., *The Logic of Computer Arithmetic*, Englewood Chffs, N. J.: Prentice-Hall, Inc. 1963.
- [14] Chu, Y., *Digital Computer Design Fundamentals*. New York: McGraw-Hill Book Co., 1962.
- [15] Kostopoulos, G. K., *Digital Engineering*. New York: John Wiley & sons, Inc., 1975.
- [16] Rhyne, V.T., *Fundamentals of Digital Systems Design*. Englewood Cliffs, N.J.: Prentice-Hall, Inc., 1973.
- [17] D. Nasib S. Gill, J.B. Dixit "Digital Design and Computer Organisation"
- [18] R. P. Jain "Modren Digital Electronics" 4<sup>th</sup> edition, Published by Tata McGraw Hill Education Private Limited, 7 West Patel Nagar, New Delhi 110 008

Table: 6.1  
Number representation,  
arithmetic, interconversion  
and compliments of four  
number systems

Number Representation				
Decimal	0,1,2,3,4,5,6,7,8,9		1798.65 or (1798.65) <sub>10</sub>	
Binary	0,1		1011.01b or (1011.01) <sub>2</sub>	
Octal	0,1,2,3,4,5,6,7		1354.40o or (135.40) <sub>8</sub>	
Hexa-decimal	0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F		B521.F9h or (B521.F9) <sub>16</sub>	
Arithmetic				
	Addition	Subtraction	Multiplication	Division
Decimal	5 + 7 = 12	25 - 7 = 18	6 × 3 = 18	33 ÷ 3 = 11
Binary	101 <sub>2</sub> + 111 <sub>2</sub> = 1100 <sub>2</sub>	11001 <sub>2</sub> - 111 <sub>2</sub> = 10010 <sub>2</sub>	110 <sub>2</sub> × 11 <sub>2</sub> = 10010 <sub>2</sub>	100001 <sub>2</sub> ÷ 11 <sub>2</sub> = 1011 <sub>2</sub>
Octal	58 + 78 = 148	258 - 78 = 168	68 × 38 = 228	338 ÷ 38 = 118
Hexa-decimal	516 + 716 = C16	2516 - 716 = 1E16	616 × 316 = 1216	3316 ÷ 316 = 1116
Inter conversion				
	Part-A		Part-B	
Step1	Decimal to others [binary, octal, hexadecimal] ( $\Rightarrow$ ) <sub>10</sub> $\rightarrow$ ( $\Rightarrow$ ) <sub>2,8,16</sub> <b>Integer:</b> repeated division method <b>Fraction:</b> repeated multiplication method		Others [binary, octal, hexadecimal] to decimal ( $\Rightarrow$ ) <sub>2,8,16</sub> $\rightarrow$ ( $\Rightarrow$ ) <sub>10</sub> <b>Integer:</b> sum of [(+ve weights) × (integer)] <b>Fraction:</b> sum of [(-ve weights) × (fraction)]	
Step2	binary to other [octal, hexadecimal] ( $\Rightarrow$ ) <sub>2</sub> $\rightarrow$ ( $\Rightarrow$ ) <sub>8,16</sub> <b>To octal:</b> replace group of 3-binary bits by octal digit <b>To hex:</b> replace group of 4-binary bits hexadecimal digit (same method for both integral and fractional part)		Other [octal, hexadecimal] to binary ( $\Rightarrow$ ) <sub>8,16</sub> $\rightarrow$ ( $\Rightarrow$ ) <sub>2</sub> <b>From octal:</b> replace each octal digit by 3-bit binary <b>From hex:</b> replace each hexadecimal digit by 4-bit binary (same method for both integral and fractional part)	
Step3	octal to hexadecimal ( $\Rightarrow$ ) <sub>8</sub> $\rightarrow$ ( $\Rightarrow$ ) <sub>16</sub> Direct conversion not applicable Octal $\rightarrow$ Binary $\rightarrow$ Hexadecimal		hexadecimal to octal ( $\Rightarrow$ ) <sub>16</sub> $\rightarrow$ ( $\Rightarrow$ ) <sub>8</sub> Direct conversion not applicable Hexadecimal $\rightarrow$ Binary $\rightarrow$ Octal	
Compliments				
	Radix compliment		(radix-1) compliment	
	Minuend + subtrahend's radix compliment = result (Discard the carry or MSD of the result)		Minuend + subtrahend's (radix-1) compliment = result Add carry (if generated) to the result	
Decimal	Ten's Compliment		Nine's Compliment	
Binary	Two's Compliment		One's Compliment	
Octal	Eight's Compliment		Seven's Compliment	
H-decimal	Sixteen's Compliment		Fifteen's Compliment	