Lesson 1 - Number Systems

Learning Outcomes

By the end of this lesson, students will be able to;

- recognize different types of number systems as they relate to computers.
- identify base/radix, positional notation, most and least significant digits as they relate to decimal, binary, octal and hexadecimal number systems.
- convert values from decimal, binary, octal and hexadecimal numbers systems to each other and back to the other systems.

1.1 Introduction

Number systems are simply ways to count things. The two number systems that we generally come across are *decimal system* and *Roman system*. But there are several other number systems such as *binary*, *octal*, *hexadecimal* etc. The binary number system, which contain only two digits 0 and 1, is widely used in electronic devices. The two states of an electronic device 'ON' or 'OFF' are denoted by 1 and 0 respectively. John Von Neumann suggested in 1946 that a computer should also use this kind of two way binary system. Such a system is known as John Von Neumann architecture and most of the computers use this architecture. Similar to the case of an electronic device when the switch is said to be 'ON' or 'OFF', a computer uses two signals; pulse and no pulse. Pulse is denoted by the digit '1' and no pulse is denoted by the digit '0'.

These systems are classified according to the values of the *base* of the number system. The number system having the value of the base as 10 is called a decimal number system, whereas that with a base of 2 is called a binary number system. Likewise, the number systems having base 8 and 16 are called octal and hexadecimal number systems respectively. The *decimal system is the most commonly* used number system.

1.2 Decimal System

The decimal system comprises of ten basic symbols or digits. They are 0, 1, 2, 3, 4, 5, 6, 7, 8 and 9. It is based on 'tens'. That is each of the digit is a number and has a value or importance ten times greater than the digit just to its right. For example, 9875 means reading from the right

$$\begin{aligned}
5 \text{ lots of } 1 \text{ (units)} &= 5 \\
7 \text{ lots of } 10 \text{ (tens)} &= 70 \\
8 \text{ lots of } 100 \text{ (hundreds)} &= 800 \\
9 \text{ lots of } 1000 \text{ (thousands)} &= 9000 \\
\text{Total} &= 9875
\end{aligned}$$

Each symbol in decimal system has its own absolute value and place value. In the above example, 9 have got two values. That is one its absolute value which is 9 and its place value which is 9000. As the decimal system has number 10 as its base or radix, the position of the digit in this system determines the magnitude of the numbers. For example the numbers 86 and 864 can be expressed in the base to system as

$$086 = 0 \times 10^2 + 8 \times 10^1 + 6 \times 10^0$$
$$864 = 8 \times 10^2 + 6 \times 10^1 + 4 \times 10^0$$

The base of number can be defined as the number which when raised to zero power assigns the lowest position value, raised to the power one assigns the second position value, raised to the power of two assigns third position value and so on.

A number in any number system can be expressed in general term as

$$\boxed{N = d_{n+1} \times r^n + \ldots \cdots + d_3 \times r^2 + d_2 \times r^1 + d_1 \times r^0,}$$

where N is the any number of any base system, d_n is a digit in the n^{th} position and r is the base of the number system. Here d is an integer in the range $0 \leq d < r$.

1.3 Binary Number System

In the binary system, there are only two digits. They are 0 and 1. In this number system, each of the digits in a number is twice the value of the digits on its right. For example 1001 means, reading from right to left

$$\begin{aligned} 1 \text{ lots of } 1 &= 1 \\ 0 \text{ lots of } 2 &= 0 \\ 0 \text{ lots of } 4 &= 0 \\ 1 \text{ lots of } 8 &= 8 \\ \text{Total} &= 9 \end{aligned}$$

As the binary system has number 2 as its base or radix, the value of the position in the binary system varies as powers of the two. That is as units, twos, fours, eights, sixteens and so on. For example, the binary number 1001 can be expressed in decimal system as

$$1001_2 = 1 \times 2^3 + 0 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 = 8 + 0 + 0 + 1 = 9_{10}.$$

The disadvantage of the binary number system against the decimal system is the increased number of digits required in the binary system to represent a number.

1.4 Octal Number System

The octal numeral system, is the base 8 number system. The octal number system uses 8 numbers from 0 to 7. They are 0, 1, 2, 3, 4, 5, 6 and 7. The main advantage of the octal number system compared to other number systems is that, it is more easy to write the number in octal number form than to write in binary number system, when we are working with computers. Especially, when we are working with large string of binary numbers, it is suggested to group them as set of three digits hence it has less chance to occur error.

1.5 Hexadecimal Number System

Hexadecimal is a base 16 number system. That means there are 16 possible digits used to represent numbers. Therefore, the 10 digits 0 to 9 and the 6 letters A to F are used. The letters A, B, C, D, E and F, which map out to values of 10, 11, 12, 13, 14 and 15.

1.6 Most Significant Bit and Least Significant Bit

In the weighted number systems we define the bit that carries the most weight as the *most significant bit* (MSB) and the bit that carries the least weight as the *least significant bit* (LSB). Conventionally the MSB is the leftmost bit and the LSB the rightmost bit.

1.7 Conversion Between Number Systems

It is often required to convert a number in a particular number system to any other number system, e.g., it may be required to convert a decimal number to binary or octal or hexadecimal. The reverse is also true, i.e., a binary number may be converted into decimal and so on.

1.7.1 Decimal-to-Binary Conversion

Conversion of Integer Decimal-to-Binary

To convert a decimal integer to binary divide successively by 2, until the quotient of zero is obtained. The remainders are noted down for each of the division steps. Then the column of the remainder is read in reverse order i.e., from bottom to top order.

Example 1.1

Convert 53_{10} in to binary.

Solution.

Division	Quotient	Remainder
$53 \div 2$	26	1
$26 \div 2$	13	0
$13 \div 2$	6	1
$6 \div 2$	3	0
$3 \div 2$	1	1
$1 \div 2$	0	1

Hence the converted binary number is 110101_2.

Conversion of Fractional Decimal-to-Binary

To convert a decimal fraction to binary the given number is multiplied by 2 to give an integer and a fraction. Then the new fraction is multiplied by 2 to give a new integer and a new fraction. The process is continued until the fraction become 0 or until the number of digits have sufficient accuracy. The digits of the binary number are obtained by reading the integer numbers from top to bottom order.

Example 1.2

Convert 0.6875_{10} in to binary.

Solution.

Multiplication	Integer	Fraction
$0.6875 \times 2 = 1.3750$	1	0.3750
$0.3750 \times 2 = 0.7500$	0	0.7500
$0.7500 \times 2 = 1.5000$	1	0.5000
$0.5000 \times 2 = 1.0000$	1	0.0000

Hence the converted binary number is 0.1011_2.

Remark 1.1. *The conversion of decimal numbers with both integer and fraction parts is done by converting the integer and the fraction separately and then combining the two answers. Using the results of Examples* 1.1 *and* 1.2, *we obtain*

$$53.6875_{10} = 110101.1011_2.$$

Remark 1.2. *The conversion of decimal integer to any base R system is similar to the procedure explained in section 1.7.1 except the division is done by R instead of* 2.

Remark 1.3. *The conversion of decimal fraction to a number expressed in base R, a similar procedure which is explained in section 1.7.1 can be used. Multiplication is done by R instead of* 2. *Therefore, integers part may range in value from* 0 *to $R-1$, instead of* 0 *and* 1.

1.7.2 Decimal-to-Octal Conversion

Conversion of Integer Decimal-to-Octal

Example 1.3

Convert 426_{10} in to an octal number.

Solution.

Division	Quotient	Remainder
$426 \div 8$	53	2
$53 \div 8$	6	5
$6 \div 8$	0	6

Hence the converted octal number is 652_8.

Conversion of Fractional Decimal-to-Octal

Example 1.4

Convert 0.513_{10} to octal.

Solution.

Multiplication	Integer	Fraction
$0.513 \times 8 = 4.104$	4	0.104
$0.104 \times 8 = 0.832$	0	0.832
$0.832 \times 8 = 6.656$	6	0.656
$0.656 \times 8 = 5.248$	5	0.248
$0.248 \times 8 = 1.984$	1	0.984
$\vdots$	$\vdots$	$\vdots$

The answer to five significant digits is 0.40651_8.

1.7.3 Decimal-to-Hexadecimal Conversion

Conversion of Integer Decimal-to-Hexadecimal

Example 1.5

Convert 348_{10} in to a hexadecimal number.

Solution.

Division	Quotient	Remainder
$348 \div 16$	21	12
$21 \div 16$	1	5
$1 \div 16$	0	1

Hence the converted hexadecimal number is $15C_{16}$.

Conversion of Fractional Decimal-to-Hexadecimal

Example 1.6

Convert 0.85_{10} in to a hexadecimal number.

Solution.

Multiplication	Integer	Fraction
$0.85 \times 16 = 13.6$	13	0.6
$0.6 \times 16 = 9.6$	9	0.6
$0.6 \times 16 = 9.6$	9	0.6
$\vdots$	$\vdots$	$\vdots$

The answer to three significant digits is $0.D99_{16}$.

1.7.4 Binary-to-Decimal Conversion

Conversion of Integer Binary-to-Decimal

The binary, octal or hexadecimal number system is a positional number system, i.e., each of the digits in the number systems discussed above has a positional weight as in the case of the decimal system. Therefore a number in the binary system can be converted into decimal system by multiplying binary positional weight with a binary digit.

Example 1.7

Convert 110011_2 in to a decimal number.

Solution.

The given binary number is	1	1	0	0	1	1
Positional weights	5	4	3	2	1	0

The positional weights for each of the digits are written below each digit. Hence

$$\begin{aligned} 110011_2 &= 1 \times 2^5 + 1 \times 2^4 + 0 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 \\ &= 32 + 16 + 0 + 0 + 2 + 1 \\ &= 51_{10} \end{aligned}$$

Conversion of fractional Binary-to-Decimal

Example 1.8

Convert 0.110101_2 in to a decimal number.

Solution.

The given binary number is	.1	1	0	1	0	1
Positional weights	-1	-2	-3	-4	-5	-6

The positional weights for each of the digits are written below each digit. Hence

$$\begin{aligned} 0.110101_2 &= 1 \times 2^{-1} + 1 \times 2^{-2} + 0 \times 2^{-3} + 1 \times 2^{-4} + 0 \times 2^{-5} + 1 \times 2^{-6} \\ &= 0.5 + 0.25 + 0 + 0.0625 + 0 + 0.015625 \\ &= 0.828125_{10} \end{aligned}$$

1.7.5 Octal-to-Decimal Conversion

Conversion of Integer Octal-to-Decimal

Example 1.9

Convert 3462_8 in to a decimal number.

Solution.

The given octal number is	3	4	6	2
Positional weights	3	2	1	0

The positional weights for each of the digits are written below each digit. Hence

$$\begin{aligned} 3462_8 &= 3 \times 8^3 + 4 \times 8^2 + 6 \times 8^1 + 2 \times 8^0 \\ &= 1536 + 256 + 48 + 2 \\ &= 1842_{10} \end{aligned}$$

Conversion of Fractional Octal-to-Decimal

Example 1.10

Convert 0.752_8 in to a decimal number.

Solution.

The given octal number is	.7	5	2
Positional weights	-1	-2	-3

The positional weights for each of the digits are written below each digit. Hence

$$\begin{aligned} 0.752_8 &= 7 \times 8^{-1} + 5 \times 8^{-2} + 2 \times 8^{-3} \\ &= 0.875 + 0.078125 + 0.00390625 \\ &= 0.95703125_{10} \end{aligned}$$

1.7.6 Hexadecimal-to-Decimal Conversion

Conversion of Integer Hexadecimal-to-Decimal

Example 1.11

Convert $42AD_{16}$ in to a decimal number.

Solution.

The given hexadecimal number is	4	2	A	D
Positional weights	3	2	1	0

The positional weights for each of the digits are written below each digit. Hence

$$\begin{aligned} 42AD_{16} &= 4 \times 16^3 + 2 \times 16^2 + 10 \times 16^1 + 13 \times 16^0 \\ &= 16384 + 512 + 160 + 13 \\ &= 17069_{10} \end{aligned}$$

Conversion of Fractional Hexadecimal-to-Decimal

Example 1.12

Convert 0.12_{16} in to a decimal number.

Solution.

The given hexadecimal number is	.1	2
Positional weights	-1	-2

The positional weights for each of the digits are written below each digit. Hence

$$\begin{aligned} 0.12_{16} &= 1 \times 16^{-1} + 2 \times 16^{-2} \\ &= 0.0625 + 0.00390625 \\ &= 0.06640625_{10} \end{aligned}$$

1.7.7 Conversion of Octal-to-Binary

For obtaining the binary equivalent of an octal number, each significant digit in the given number is replaced by its *3-bit binary* equivalent. Combining all those digits we can get the final binary equivalent.

Example 1.13

Convert 376_8 in to binary.

Solution.

The given octal number is	3	7	6
3-bit binary equivalent	011	111	110

Hence the converted binary number is 11111110_2

The same procedure is to be followed to convert fractional parts also.

Example 1.14

Convert 43.0276_8 in to binary.

Solution.

The given octal number is	4	3	.	0	2	7	6
3-bit binary equivalent	100	011	.	000	010	111	110

Hence the converted binary number is 100011.000010111110_2

1.7.8 Conversion of Binary-to-Octal

To convert from binary to octal, write the bits of the binary number in *groups of three*, starting from the LSB. Then write the decimal equivalent of those groups and we get the final octal number. If necessary, add binary digit "0" in the MSB side to complete a group of three digits. This is called left padding of the number with 0.

Example 1.15

Convert 1011110_2 into an equivalent octal number.

Solution.

The given binary number is	1	0	1	1	1	1	0				
Starting with LSB and grouping 3 bits	**0**	**0**	1	|	0	1	1	|	1	1	0
Octal equivalent		1				3				6	

Hence the octal equivalent number is 136_8

If the number has a fractional part then there will be two different classes of groups; one for the integer part starting from the left of the decimal point and proceeding toward the left and the second one starting from the right of the decimal point and proceeding toward the right. If, for the second class, any 1 is left out, we complete the group by adding two 0s on the right side. This is called right padding.

Example 1.16

Convert 1101.0111_2 into an equivalent octal number.

Solution.

The given binary number is	1	1	0	1	.	0	1	1	1					
Grouping 3 bits	**0**	**0**	1	|	1	0	1	.	0	1	1	|	1	**0**	**0**
Octal equivalent		1				5		.		3				4	

Hence the octal number is 15.34_8.

1.7.9 Conversion of Hexadecimal-to-Binary

For obtaining the binary equivalent of an hexadecimal number, each significant digit in the given number is replaced by its *4-bit binary* equivalent. Combining all those digits we can get the final binary equivalent.

Example 1.17

Convert hexadecimal number $29C_{16}$ to binary.

Solution.

The given octal number is	2	9	C
4-bit binary equivalent	0010	1001	1100

Hence the equivalent binary number is 001010011100_2

The same procedure is to be followed to convert fractional parts also.

Example 1.18

Convert hexadecimal number $A4.E_{16}$ to binary.

Solution.

The given octal number is	A	4	.	E
4-bit binary equivalent	1010	0100	.	1110

Hence the equivalent binary number is 10100100.1110_2

1.7.10 Conversion of Binary-to-Hexadecimal

To convert from binary to hexadecimal, write the bits of the binary number in *groups of four*, starting from the LSB. Then write the decimal equivalent of those groups and get the final hexadecimal number. If necessary, add binary digit "0" in the MSB side to complete a group of four digits.

Example 1.19

Convert 110011110_2 into an equivalent hexadecimal number.

Solution.

The given binary number is	1	1	0	0	1	1	1	1	0					
Grouping 4 bits	**0**	**0**	**0**	1	|	1	0	0	1	|	1	1	1	0
Hexadecimal equivalent		1					9					E		

Hence the hexadecimal number is $19E_{16}$.

If the number has a fractional part, as in the case of octal numbers, then there will be two different classes of groups; one for the integer part starting from the left of the decimal point and proceeding toward the left and the second one starting from the right of the decimal point and proceeding toward the right. If, for the second class, any uncompleted group is left out, we complete the group by adding 0s on the right side.

Example 1.20

Convert 111011.011_2 into an equivalent hexadecimal number.

Solution.

The given binary number is	1	1	1	0	1	1	.	0	1	1			
Grouping 4 bits	**0**	**0**	1	1	|	1	0	1	1	.	0	1	1	**0**
Hexadecimal equivalent		3					B			.		6	

Hence the hexadecimal equivalent number is $3B.6_{16}$.

1.7.11 Conversion of Octal-to-Hexadecimal

To convert a octal number to hexadecimal, the following steps can be applied.

1. Convert the given octal number to its binary equivalent.
2. Form group of 4-bit, starting from the least significant bit.
3. Write the equivalent hexadecimal number for each group of 4-bits.

Example 1.21

Convert 32_8 into an equivalent hexadecimal number.

Solution.

The given octal number is	3	2								
Binary equivalent is	011	010								
Forming groups of 4 bits from the LSB	**0**	**0**	0	1	|	1	0	1	0
Hexadecimal equivalent			1					A	

Hence the hexadecimal equivalent of 32_8 is $1A_{16}$.

1.7.12 Conversion of Hexadecimal-to-Octal

To convert a hexadecimal number to octal, the following steps can be applied.

1. Convert the given hexadecimal number to its binary equivalent.
2. Form group of 3-bit, starting from the least significant bit.
3. Write the equivalent octal number for each group of 3-bits.

Example 1.22

Convert $(47)_{16}$ into an equivalent octal number.

Solution.

The given hexadecimal number is	4	7								
Binary equivalent is	0100	0111								
Forming groups of 3 bits from the LSB	**0** 0	1	|	0	0	0	|	1	1	1
Hexadecimal equivalent		1			0				7	

Hence the octal equivalent of $(47)_{16}$ is 107_8.

Self-Assessment Exercises

1. Convert the decimal number 635 to octal number.
2. Convert the octal number 617025 to binary number.
3. Convert the hexadecimal number $ABC.DE$ to octal number.
4. Convert the binary number 1100011.111 to decimal number.
5. Convert the decimal number 0.456743 to binary number.

Suggested Reading

Chapter 19: William Stallings, (2010) Computer Organization and Architecture: Designing for Performance, 9$^{\text{th}}$ Edition, Prentice Hall.