Number System

Number System

The technique to represent and work with numbers is called **number system**. **Decimal number system** is the most common number system. Other popular number systems include **binary number system, octal number system, hexadecimal number system,** etc.

Decimal Number System

Decimal number system is a **base 10** number system having 10 digits from 0 to 9. This means that any numerical quantity can be represented using these 10 digits. Decimal number system is also a **positional value system**. This means that the value of digits will depend on its position. Let us take an example to understand this.

Say we have three numbers – 734, 971 and 207. The value of 7 in all three numbers is different–

- In 734, value of 7 is 7 hundreds or 700 or 7×100 or 7×10^2
- In 971, value of 7 is 7 tens or 70 or 7×10 or 7×10^1
- In 207, value 0f 7 is 7 units or 7 or 7×1 or 7×10^0

The weightage of each position can be represented as follows –

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

In digital systems, instructions are given through electric signals; variation is done by varying the voltage of the signal. Having 10 different voltages to implement decimal number system in digital equipment is difficult. So, many number systems that are easier to implement digitally have been developed. Let's look at them in detail.

Binary Number System

The easiest way to vary instructions through electric signals is two-state system – on and off. On is represented as 1 and off as 0, though 0 is not actually no signal but signal at a lower voltage. The number system having just these two digits – 0 and 1 – is called **binary number system**.

Each binary digit is also called a **bit**. Binary number system is also positional value system, where each digit has a value expressed in powers of 2, as displayed here.

2^5	2^4	2^3	2^2	2^1	2^0

In any binary number, the rightmost digit is called **least significant bit (LSB)** and leftmost digit is called **most significant bit (MSB)**.

1	1	0	1	0

MSB   LSB
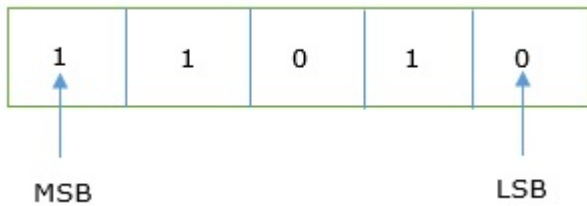
And decimal equivalent of this number is sum of product of each digit with its positional value.

$11010_2 = 1×2^4 + 1×2^3 + 0×2^2 + 1×2^1 + 0×2^0$

$= 16 + 8 + 0 + 2 + 0$

$= 26_{10}$

Computer memory is measured in terms of how many bits it can store. Here is a chart for memory capacity conversion.

- 1 byte (B) = 8 bits
- 1 Kilobytes (KB) = 1024 bytes
- 1 Megabyte (MB) = 1024 KB
- 1 Gigabyte (GB) = 1024 MB
- 1 Terabyte (TB) = 1024 GB
- 1 Exabyte (EB) = 1024 PB
- 1 Zettabyte = 1024 EB
- 1 Yottabyte (YB) = 1024 ZB

Octal Number System

Octal number system has eight digits – 0, 1, 2, 3, 4, 5, 6 and 7. Octal number system is also a positional value system with where each digit has its value expressed in powers of 8, as shown here –

8^5	8^4	8^3	8^2	8^1	8^0

Decimal equivalent of any octal number is sum of product of each digit with its positional value.

$726_8 = 7×8^2 + 2×8^1 + 6×8^0$

$= 448 + 16 + 6$

$= 470_{10}$

Hexadecimal Number System

Octal number system has 16 symbols – 0 to 9 and A to F where A is equal to 10, B is equal to 11 and so on till F. Hexadecimal number system is also a positional value system with where each digit has its value expressed in powers of 16, as shown here –

16^5	16^4	16^3	16^2	16^1	16^0

Decimal equivalent of any hexadecimal number is sum of product of each digit with its positional value.

$27FB_{16} = 2\times16^3 + 7\times16^2 + 15\times16^1 + 10\times16^0$

$= 8192 + 1792 + 240 + 10$

$= 10234_{10}$

Number System Relationship

The following table depicts the relationship between decimal, binary, octal and hexadecimal number systems.

HEXADECIMAL	CIMAL	CTAL	NARY
0	0	0	)000
1	1	1	)001
2	2	2	)010
3	3	3	)011
4	4	4	)100
5	5	5	)101
6	6	6	)110
7	7	7	)111
8	8	10	.000
9	9	11	.001
A	10	12	.010
B	11	13	.011
C	12	14	.100
D	13	15	.101
E	14	16	.110
F	15	17	.111

ASCII

Besides numerical data, computer must be able to handle alphabets, punctuation marks, mathematical operators, special symbols, etc. that form the complete character set of English language. The complete set of characters or symbols are called alphanumeric codes. The complete alphanumeric code typically includes −

- 26 upper case letters
- 26 lower case letters
- 10 digits
- 7 punctuation marks
- 20 to 40 special characters

Now a computer understands only numeric values, whatever the number system used. So all characters must have a numeric equivalent called the alphanumeric code. The most widely used alphanumeric code is American Standard Code for Information Interchange (ASCII). ASCII is a 7-bit code that has 128 (2^7) possible codes.

ASCII Code - Character to Binary

Char	Binary	Char	Binary	Char	Binary	Char	Binary
0	0011 0000	I	0100 1001	b	0110 0010	v	0111 0110
1	0011 0001	J	0100 1010	c	0110 0011	w	0111 0111
2	0011 0010	K	0100 1011	d	0110 0100	x	0111 1000
3	0011 0011	L	0100 1100	e	0110 0101	y	0111 1001
4	0011 0100	M	0100 1101	f	0110 0110	z	0111 1010
5	0011 0101	N	0100 1110	g	0110 0110		
6	0011 0110	O	0100 1111	h	0110 1000	:	0011 1010
7	0011 0110	P	0101 0000	i	0110 1001	;	0011 1011
8	0011 1000	Q	0101 0001	j	0110 1010	?	0011 1111
9	0011 1001	R	0101 0010	k	0110 1011	.	0010 1110
		S	0101 0011	l	0110 1100	,	0010 1111
		T	0101 0100	m	0110 1101	!	0010 0001
A	0100 0001	U	0101 0101	n	0110 1110	'	0010 1100
B	0100 0010	V	0101 0110	o	0110 1111	"	0010 0010
C	0100 0011	W	0101 0111	p	0111 0000	(	0010 1000
D	0100 0100	X	0101 1000	q	0111 0001	)	0010 1001
E	0100 0101	Y	0101 1001	r	0111 0010	space	0010 0000
F	0100 0110	Z	0101 1010	s	0111 0011		
G	0100 0111			t	0111 0100		
H	0100 1000	a	0110 0001	u	0111 0101		

ISCII

ISCII stands for **Indian Script Code for Information Interchange**. IISCII was developed to support Indian languages on computer. Language supported by IISCI include Devanagari, Tamil, Bangla, Gujarati, Gurmukhi, Tamil, Telugu, etc. IISCI is mostly used by government departments and before it could catch on, a new universal encoding standard called **Unicode** was introduced.

Unicode

Unicode is an international coding system designed to be used with different language scripts. Each character or symbol is assigned a unique numeric value, largely within the framework of ASCII. Earlier, each script had its own encoding system, which could conflict with each other.

In contrast, this is what Unicode officially aims to do – *Unicode provides a unique number for every character, no matter what the platform, no matter what the program, no matter what the language*.