Materials for the Remaining Codes

- Byte
- Kilo Byte
- Mega Byte
- Giga Byte
- Tera Byte
- Peta Byte
- Exa Byte
- Zetta Byte
- Yotta Byte

Name	**Equal To**	**Size(In Bytes)**
Bit	1 Bit	1/8
Nibble	4 Bits	1/2 (rare)
Byte	8 Bits	1
Kilobyte	1024 Bytes	1024 [2]10
Megabyte	1, 024 Kilobytes	1, 048, 576 [2]20
Gigabyte	1, 024 Megabytes	1, 073, 741, 824 [2]30
Terrabyte	1, 024 Gigabytes	1, 099, 511, 627, 776 [2]40
Petabyte	1, 024 Terabytes	1, 125, 899, 906, 842, 624 [2]50
Exabyte	1, 024 Petabytes	1, 152, 921, 504, 606, 846, 976 [2]60
Zettabyte	1, 024 Exabytes	1, 180, 591, 620, 717, 411, 303, 424 [2]70
Yottabyte	1, 024 Zettabytes	1, 208, 925, 819, 614, 629, 174, 706, 176 [2]80

BCD

Short for **binary-coded decimal**, **BCD** is also known as **packet decimal** and is numbers 0 through 9 converted to four-digit binary. Below is a list of the decimal numbers 0 through 9 and the binary conversion.

Decimal	BCD

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

ASCII CODES

Short for **American Standard Code for Information Interexchange**, **ASCII** is a standard that assigns letters, numbers, and other characters in the 256 slots available in the 8-bit code. The ASCII decimal (Dec) number is created from binary, which is the language of all computers. As shown in the table below, the lowercase "h" character (Char) has a decimal value of 104, which is "01101000" in binary.

ASCII was first developed and published in 1963 by the X3 committee, a part of the ASA (American Standards Association). The ASCII standard was first published as ASA X3.4-1963, with ten revisions of the standard being published between 1967 and 1986.

ASCII sections

The ASCII table is divided into three different sections.

- **Non-printable**, system codes between 0 and 31.
- **Lower ASCII**, between 32 and 127. This table originates from the older, American systems, which worked on 7-bit character tables.
- **Higher ASCII**, between 128 and 255. This portion is programmable; characters are based on the language of your operating system or program you are using. Foreign letters are also placed in this section.
- **Standard or lower ASCII characters and codes**

Char	Dec	Binary	Char	Dec	Binary	Char	Dec	Binary
!	033	00100001	A	065	01000001	a	097	01100001
"	034	00100010	B	066	01000010	b	098	01100010
#	035	00100011	C	067	01000011	c	099	01100011
$	036	00100100	D	068	01000100	d	100	01100100
%	037	00100101	E	069	01000101	e	101	01100101
&	038	00100110	F	070	01000110	f	102	01100110
'	039	00100111	G	071	01000111	g	103	01100111
(	040	00101000	H	072	01001000	h	104	01101000
)	041	00101001	I	073	01001001	i	105	01101001

*	042	00101010	**J**	074	01001010	**j**	106	01101010
+	043	00101011	**K**	075	01001011	**k**	107	01101011
,	044	00101100	**L**	076	01001100	**l**	108	01101100
-	045	00101101	**M**	077	01001101	**m**	109	01101101
.	046	00101110	**N**	078	01001110	**n**	110	01101110
/	047	00101111	**O**	079	01001111	**o**	111	01101111
0	048	00110000	**P**	080	01010000	**p**	112	01110000
1	049	00110001	**Q**	081	01010001	**q**	113	01110001
2	050	00110010	**R**	082	01010010	**r**	114	01110010
3	051	00110011	**S**	083	01010011	**s**	115	01110011
4	052	00110100	**T**	084	01010100	**t**	116	01110100
5	053	00110101	**U**	085	01010101	**u**	117	01110101
6	054	00110110	**V**	086	01010110	**v**	118	01110110
7	055	00110111	**W**	087	01010111	**w**	119	01110111
8	056	00111000	**X**	088	01011000	**x**	120	01111000
9	057	00111001	**Y**	089	01011001	**y**	121	01111001
:	058	00111010	**Z**	090	01011010	**z**	122	01111010
;	059	00111011	[	091	01011011	{	123	01111011
<	060	00111100	\	092	01011100	|	124	01111100
=	061	00111101	]	093	01011101	}	125	01111101
>	062	00111110	^	094	01011110	~	126	01111110
?	063	00111111	_	095	01011111	_	127	01111111
@	064	01000000	`	096	01100000			

Definition - What does Extended Binary Coded Decimal Interchange Code (EBCDIC) mean?

Extended binary coded decimal interchange code (EBCDIC) is an 8-bit binary code for numeric and alphanumeric characters. It was developed and used by IBM.

It is a coding representation in which symbols, letters and numbers are presented in binary language.

Techopedia explains Extended Binary Coded Decimal Interchange Code (EBCDIC)
EBCDIC is an 8-bit character encoding widely used in IBM midrange and mainframe computers. This encoding was developed in 1963 and 1964. EBCDIC was developed to enhance the existing capabilities of binary-coded decimal code. This code is used in text files of S/390 servers and OS/390 operating systems of IBM.

universal character encoding standard

Unicode is a universal character encoding standard that assigns a code to every character and symbol in every language in the world. Since no other encoding standard supports all languages, **Unicode is** the only encoding standard that ensures that you can retrieve or combine data using any combination of languages.

Fundamentally, computers just deal with numbers. They store letters and other characters by assigning a number for each one. Before Unicode was invented, there were hundreds of different systems, called character encodings, for assigning these numbers. These early character encodings were limited and could not contain enough characters to cover all the world's languages. Even for a single language like English no single encoding was adequate for all the letters, punctuation, and technical symbols in common use.

Early character encodings also conflicted with one another. That is, two encodings could use the same number for two *different* characters, or use different numbers for the *same* character. Any given computer (especially servers) would need to support many different encodings. However, when data is passed through different computers or between different encodings, that data runs the risk of corruption.

What is Carry Bit

So when adding **binary** numbers, a **carry** out is generated when the “SUM” equals or is greater than two (1+1) and this becomes a “**CARRY**” **bit** for any subsequent addition being passed over to the next column for addition and so on.

Sign bit

In computer science, the sign bit is a bit in a signed number representation that indicates the sign of a number. Only signed numeric data types have a sign bit, and its place is usually the leftmost, where the most significant bit in unsigned numbers resides. Floating point numbers in IEEE format are always signed, with the sign bit in the leftmost position. Typically if the sign bit is 1 then the number is negative or non-positive, while 0 indicates a non-negative number

A parity bit is a check bit, which is added to a block of data for error detection purposes. It is used to validate the integrity of the data. The value of the parity bit is assigned either 0 or 1 that makes the number of 1s in the message block either even or odd depending upon the type of parity. Parity check is suitable for single bit error detection only.

The two types of parity checking are

- **Even Parity** − Here the total number of bits in the message is made even.
- **Odd Parity** − Here the total number of bits in the message is made odd.

Error Detection by Adding Parity Bit

Sender's End − While creating a frame, the sender counts the number of 1s in it and adds the parity bit in following way

- **In case of even parity** − If number of 1s is even, parity bit value is 0. If number of 1s is odd, parity bit value is 1.
- **In case of odd parity** − If number of 1s is odd, parity bit value is 0. If number of 1s is even, parity bit value is 1.