Binary Arithmetic

Binary arithmetic includes the basic arithmetic operations of addition, subtraction, multiplication and division. The following sections present the rules that apply to these operations when they are performed on binary numbers.

Binary Addition

Binary addition is performed in the same way as addition in the decimal-system and is, in fact, much easier to master. Binary addition obeys the following four basic rules

0 +	0 +	1 +	1 +
0	1	0	1
0	1	1	10

The results of the last rule may seem somewhat strange, remember that these are binary numbers. Put into words, the last rule states that.

Example

```
01  +        10  +
10           00
----         ----
11           10
```

Example

```
1 1 1  +     1 0 1 0
1 0 1        1 0 0 1
------       1 1 0 1
1 1 0 0      -------
             1 0 0 0 0 0
```

Binary Subtraction:

Binary subtraction is just as simple as addition subtraction of one bit from another according the following four basic rules.

0 – 0 = 0

1 – 1 =0

1 – 0 = 1

0 – 1 = 1   with a transfer (borrow) of 1.

Example:

1001 -

101

1 0 0

10 0 0 0 -

0 0 101

1 0 0 1 1

Binary Multiplication

0 * 0 = 0

1 * 0=0

0 * 1= 1

1 * 1 = 1

Example

```
    101   *
     10
  ------
    0 0 0
  1 0 1
  ------
 1  0   1  0
```

Example:

```
        1100  *
        1010
       ------
        0 0 0 0
      1 1 0 0
    0 0 0 0
  1 1 0 0
  ------
1 1 1  1  0  0  0
```

Binary Division

0/ 1= 0

1 / 1= 1

Example:

```
        110
  11 | 10010
        11
        ---
         11
         11
         ---
          00
           0
          ---
           0
```

Example:

Coding System

- **Binary Coded Desimal (BCD)**
- **ASCII CODE**

- **Binary Coded Desimal (BCD)**

Is an encoding for decimal numbers in which each digit is represented by its own binary sequence. In computing and electronic systems, binary-coded decimal (BCD) is an encoding for decimal numbers in which each digit is represented by its own binary sequence. Its drawbacks are the increased complexity of circuits needed to implement mathematical operations and a relatively inefficient encoding – it occupies more space than a pure binary representation. Even though the importance of BCD has diminished, it is still widely used in financial, commercial, and industrial applications. In BCD, a digit is usually represented by four bits which, in general, represent the values/digits/characters 0-9. Other bit combinations are sometimes used for sign or other indications. To BCD-encode a decimal number using the common encoding, each decimal digit is stored in a four-bit nibble. Decimal: 0 1 2 3 4 5 6 7 8 9

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

Thus, the BCD encoding for the number	1	2	7
would be:	0001	0010	0111

- **American Standard Code for Information Interchange (ASCII CODE)**

TABLA DE CARACTERES DEL CÓDIGO ASCII

1	☺	25	↓	49	1	73	I	97	a	121	y	145	æ	169	⌐	193	┴	217	┘	241	±
2	☻	26		50	2	74	J	98	b	122	z	146	Æ	170	¬	194	┬	218	┌	242	≥
3	♥	27		51	3	75	K	99	c	123	{	147	ô	171	½	195	├	219	█	243	≤
4	♦	28	∟	52	4	76	L	100	d	124	|	148	ö	172	¼	196	─	220	▄	244	⌠
5	♣	29	↔	53	5	77	M	101	e	125	}	149	ò	173	¡	197	┼	221	▌	245	⌡
6	♠	30	▲	54	6	78	N	102	f	126	~	150	û	174	«	198	╞	222	▐	246	÷
7		31	▼	55	7	79	O	103	g	127	▓	151	ù	175	»	199	╟	223	▀	247	≈
8		32		56	8	80	P	104	h	128	Ç	152	ÿ	176	░	200	╚	224	α	248	°
9		33	!	57	9	81	Q	105	i	129	ü	153	Ö	177	▒	201	╔	225	ß	249	∙
10		34	"	58	:	82	R	106	j	130	é	154	Ü	178	▓	202	╩	226	Γ	250	·
11		35	#	59	;	83	S	107	k	131	â	155	¢	179	│	203	╦	227	π	251	√
12		36	$	60	<	84	T	108	l	132	ä	156	£	180	┤	204	╠	228	Σ	252	ⁿ
13		37	%	61	=	85	U	109	m	133	à	157	¥	181	╡	205	═	229	σ	253	²
14		38	&	62	>	86	V	110	n	134	å	158	₧	182	╢	206	╬	230	µ	254	■
15		39	'	63	?	87	W	111	o	135	ç	159	ƒ	183	╖	207	╧	231	τ	255	
16	►	40	(	64	@	88	X	112	p	136	ê	160	á	184	╕	208	╨	232	Φ	PRESIONA LA TECLA	
17		41	)	65	A	89	Y	113	q	137	ë	161	í	185	╣	209	╤	233	Θ	Alt	
18	↕	42	*	66	B	90	Z	114	r	138	è	162	ó	186	║	210	╥	234	Ω	MÁS EL NÚMERO	
19	‼	43	+	67	C	91	[	115	s	139	ï	163	ú	187	╗	211	╙	235	δ		
20	¶	44	,	68	D	92	\	116	t	140	î	164	ñ	188	╝	212	╘	236	∞	CORTESÍA DE:	
21	§	45	-	69	E	93	]	117	u	141	ì	165	Ñ	189	╜	213	╒	237	φ		
22	▬	46	.	70	F	94	^	118	v	142	Ä	166	ª	190	╛	214	╓	238	ε		
23	↨	47	/	71	G	95	_	119	w	143	Å	167	º	191	┐	215	╫	239	∩		
24	↑	48	0	72	H	96	`	120	x	144	É	168	¿	192	└	216	╪	240	≡		

www.rey-dec.com

The binary-coded decimal scheme described in this article is the most common encoding, but there are many others. The method here can be referred to as Simple Binary-Coded Decimal (SBCD) or BCD 8421. In the headers to the table, the '8 4 2 1' indicates the four bit weights; note that in the 5th column two of the weights are negative. The following table represents decimal digits from 0 to 9 in various BCD systems:

Digit	BCD 8 4 2 1	Excess-3 or Stibitz Code	BCD 2 4 2 1 or Aiken Code	BCD 8 4 −2 −1	IBM 702 IBM 705 IBM 7080 IBM 1401 8 4 2 1
0	0000	0011	0000	0000	1010
1	0001	0100	0001	0111	0001
2	0010	0101	0010	0110	0010
3	0011	0110	0011	0101	0011
4	0100	0111	0100	0100	0100
5	0101	1000	1011	1011	0101
6	0110	1001	1100	1010	0110
7	0111	1010	1101	1001	0111
8	1000	1011	1110	1000	1000
9	1001	1100	1111	1111	1001

Gray Code

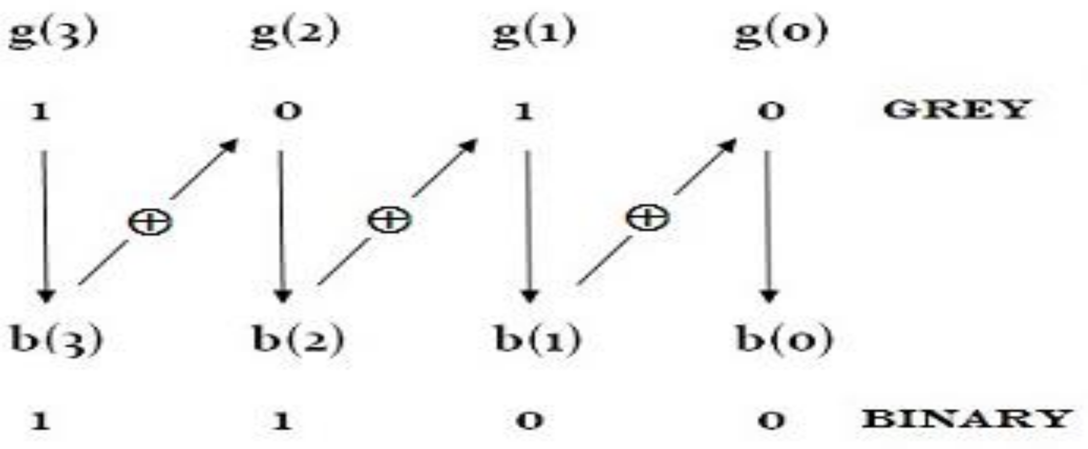
A Gray Code represents numbers using a binary encoding scheme that groups a sequence of bits so that only one bit in the group changes from the number before and after. It is named for Bell Labs researcher Frank Gray, who described it in his 1947 patent submittal on Pulse Code Communication. A Gray Code is not weighted, the columns of bits do not reflect an implicit base weight as the Binary number system does.

A comparison of the first ten numbers in Decimal, Binary and Gray Code is shown in Table 1.

Table 1. Decimal, Binary, Gray Code Numbers

Decimal (base 10)	Binary (base 2)	Binary-Reflected (no base)
0	0000	0000
1	0001	0001
2	0010	0011
3	0011	0010
4	0100	0110
5	0101	0111
6	0110	0101
7	0111	0100
8	1000	1100
9	1001	1101
10	1010	1111

To convert from gray to binary



From the above operation, finally we can get the binary values like

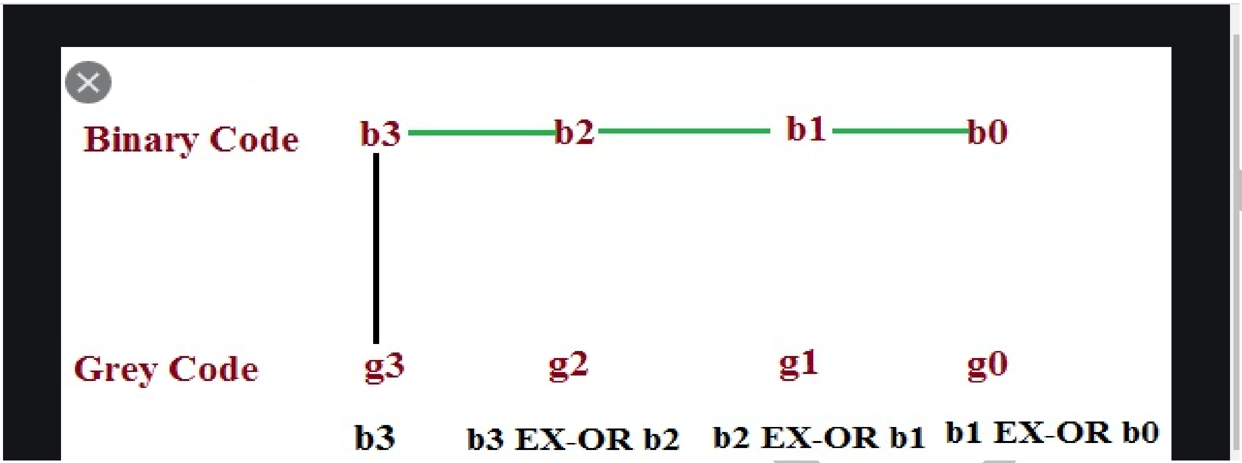
b3 = g3,

b2 = b3 XOR g2,

b1= b2 XOR g1,

b0 = b1 XOR g0.

To Convert from binary to Gray



From the above operation, finally we can get the binary values like

g3=b3

g2 = b3 XOR b2,

g1= b2 XOR b1,

g0 = b1 XOR b0.