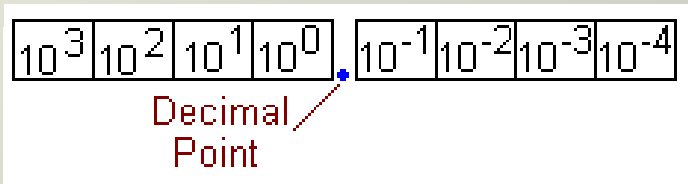
Sistemas de numeración, operaciones y códigos

Sistemas de numeración

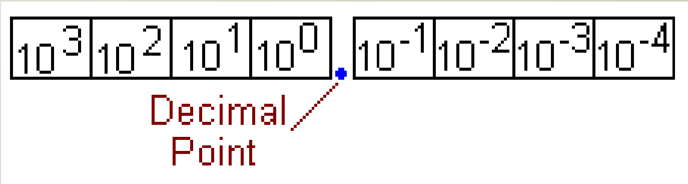
Números decimales

- El sistema de numeración decimal tiene diez dígitos: 0, 1, 2, 3, 4, 5, 6, 7, 8, y 9
- Es un sistema en base 10
- El valor de un dígito se determina por su posición dentro del número. Se le asigna un peso.



Números decimales

- Los pesos para los números enteros positivos son potencias de 10, que aumentan de derecha a izquierda, comenzando por 10^0 .
- Para números fraccionarios los pesos son potencias negativas de 10 que decrecen de izda a dcha empezando por 10^{-1}



Números binarios

- El sistema de numeración binario emplea dos dígitos (bits): 0 y 1
- Es un sistema en base 2
- El valor de un bit se determina por su posición dentro del número

POSITIVE POWERS OF TWO (WHOLE NUMBERS)									NEGATIVE POWERS OF TWO (FRACTIONAL NUMBER)					
2^8	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0	2^{-1}	2^{-2}	2^{-3}	2^{-4}	2^{-5}	2^{-6}
256	128	64	32	16	8	4	2	1	1/2	1/4	1/8	1/16	1/32	1/64
									0.5	0.25	0.125	0.0625	0.03125	0.015625

Números binarios

Número decimal	Número binario			
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1
10	1	0	1	0
11	1	0	1	1
12	1	1	0	0
13	1	1	0	1
14	1	1	1	0
15	1	1	1	1

Máximo nº decimal=
$2^n - 1$

Números binarios

- Un número binario es un número con peso
- Bit más a la derecha, es el menos significativo **LSB**
- Bit más a la izquierda, es el mas significativo **MSB**
- También hay números fraccionarios

Conversión binario a decimal

- Método de la suma de pesos

Sumar los pesos de todos los 1s de un número para obtener el correspondiente valor decimal.

Números binarios

Ejemplos

- 1101101=

26+25+23+22+20 = 64+32+8+4+1= 109

- 10010001=

$2^7+2^4+2^0$=128+16+1= 145

- 0,1011=

$2^{-1}+2^{-3}+2^{-4}$ = 0,5+0,125+0,0625= 0,6875

- 10,111=

$2^1+2^{-1}+2^{-2}+2^{-3}$= 2+0,5+0,25+0,125=2,875

Conversión decimal a binario (I)

- Método de la suma de pesos
- Método de la división sucesiva por 2
- Conversión de fracciones decimales a binario

Conversión decimal a binario (II)

- Suma de pesos

  Se hallan los pesos binarios que sumados darán el número decimal de partida.

$$9=8+1=$$

$$2^3+2^0 =1001$$

Conversión decimal a binario (III)

Ejemplos

- $25 = 16+8+1 = 2^4+2^3+2^0 = 11001$
- $58 = 32+16+8+2 = 2^5+2^4+2^3+2^1 = 111010$
- $82 = 64+16+2 = 2^6+2^4+2^1 = 1010010$
- $125 = 64+32+16+8+4+1 =$
  $2^6+2^5+2^4+2^3+2^2+2^0 = 1111101$

Números binarios

Conversión decimal a binario (IV)

- División sucesiva por 2

	(Resto)
19/2=9	1
9/2=4	1
4/2=2	0
2/2= *1*	0
1/2= *0*	*1*
	*1*0011

Conversión decimal a binario (V)

Conversión de fracciones decimales a fracciones binarias

- Suma de pesos
- Multiplicación sucesiva por 2

Conversión decimal a binario (VI)

Método suma de pesos:

- 45,625

$=32+8+4+1+0,5+0,125= 2^5+2^3+2^2+2^0+2^{-1}=$

101101,101

Conversión decimal a binario (VII)

Divisiones/multiplicaciones sucesivas

- 0,375 x 2 = 0,75 parte entera: 0
  0,75 x 2 = 1,50 parte entera: 1
  0,50 x 2 = 1,0 parte entera: 0

0,375 = 0,011

Arimética binaria

Aritmética binaria

- Suma binaria
- Resta binaria
- Multiplicación binaria
- División binaria

Aritmética binaria

Suma binaria

Cuatro reglas básicas

- 0+0=0 Suma 0 con acarreo 0
- 0+1=1 Suma 1 con acarreo 0
- 1+0=1 Suma 1 con acarreo 0
- 1+1=10 Suma 0 con acarreo 1

Resta binaria

Cuatro reglas básicas

- 0-0=0
- 1-1=0
- 1-0=1
- 0-1=1

Se hace 10-1 (siendo 10-1=1, con acarreo negativo de 1)

Aritmética binaria

Resta binaria

- Ejemplos:

101-011= 010

10001 – 01010 = **00111**

$$17_{10} - 10_{10} = 7_{10}$$

11011001 – 10101011 = **00101110**

$$217_{10} - 171_{10} = 46_{10}$$

111101001 – 101101101 = **001111100**

$$489_{10} - 365_{10} = 124_{10}$$

Aritmética binaria

Multiplicación binaria

Cuatro reglas básicas

- 0x0=0
- 0x1=0
- 1x0=0
- 1x1=1

Numeros de varios bits, igual que en decimal

La multiplicación binaria de dos bits es igual que la multiplicación de los dígitos decimales 0 y 1

Aritmética binaria

División binaria

- Mismo procedimiento que la división decimal.

- Ejemplo:

1100/100

```
 1100 |100
-100    11
 0100
  -100
   000
```

Complemento de los números binarios

Complementos de los números binarios

- Complemento a 1
- Complemento a 2

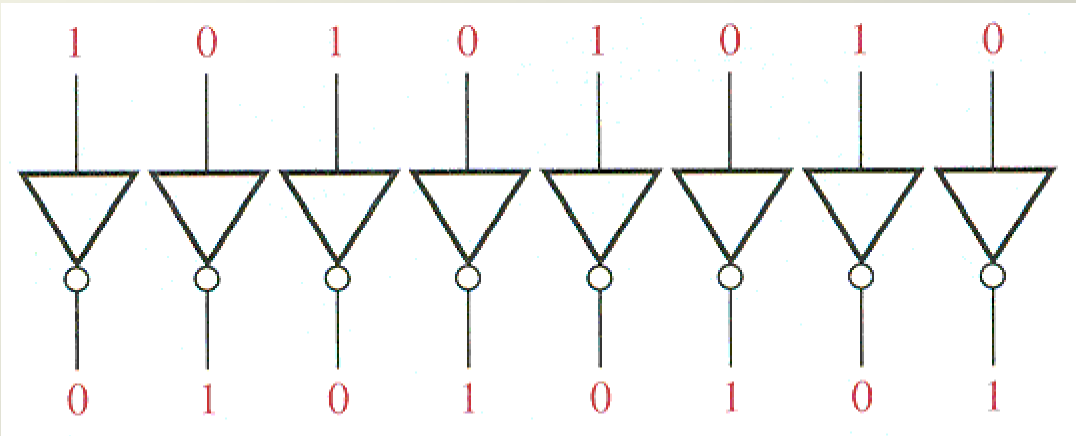
Sirven para representar los números negativos.

El complemento a 2 lo suelen usar las computadoras para manipular los números negativos.

Complementos de los números binarios

• Complemento a 1

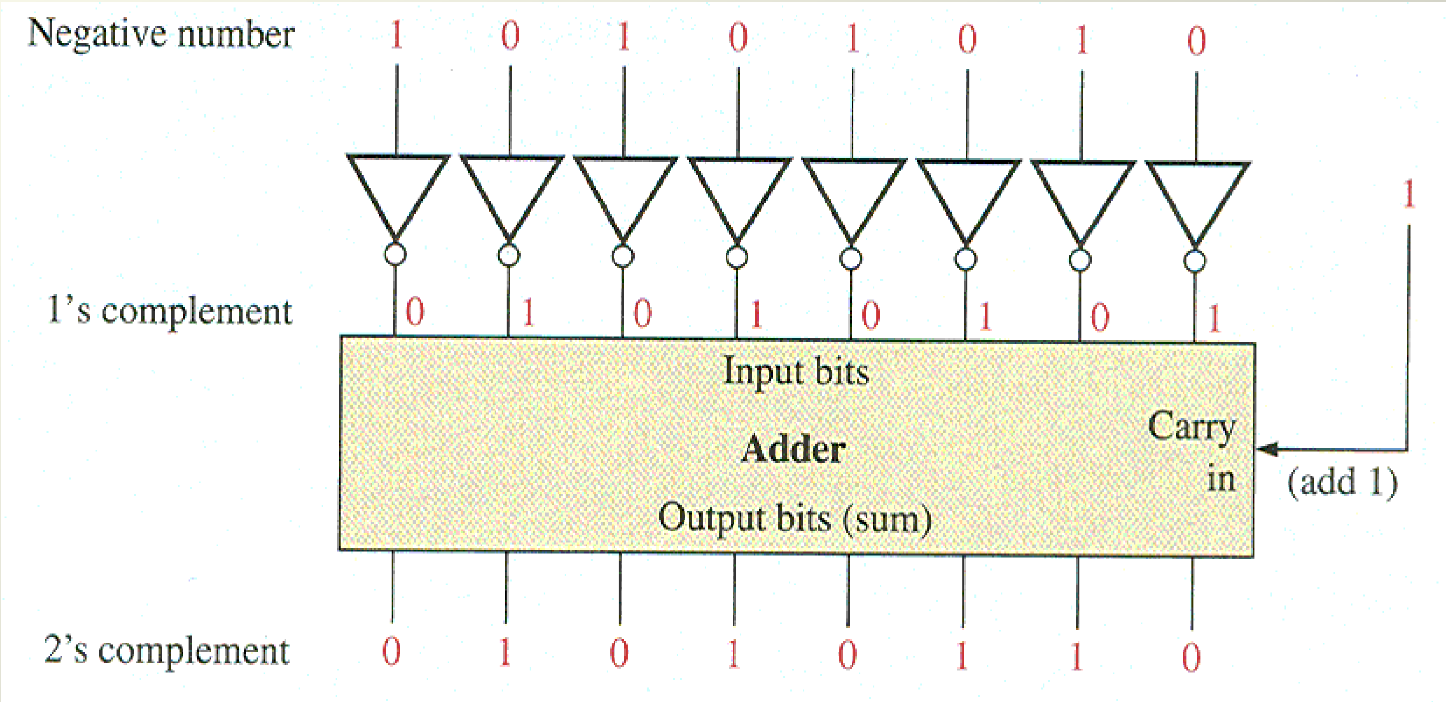
Cambio cada uno de los bits del número para obtener el complemento a 1



Complementos de los números binarios

- Complemento a 2

Sumar 1 al complemento a 1 para obtener el complemento a 2



Complementos de los números binarios

- Complemento a 2

Método alternativo

Cambiar todos los bits situados a la izquierda del 1 menos significativo para obtener el complemento a 2

10110010	Número binario
10110010	
01001110	
01001110	Complemento a 2

Números con signo

Números con signo

- Bit de signo
- Números binarios de 8 bits
- Formato signo-magnitud
- Formato complemento a 2
- Valor decimal de los números con signo
- Rango de valores de los números enteros

Números con signo

- Formato signo-magnitud
  - El bit más a la izquierda es el bit de signo y los restantes bits son los bits de magnitud
  - Un bit de signo 0 indica que es un número positivo
  - Un bit de signo 1 indica que es un número negativo
  - Ejemplo: 10011001 (-25)

Números con signo

- Formato del complemento a 2
  - Un número negativo es el complemento a 2 del correspondiente número positivo.
  - Ejemplo: 00011001 (25)

    11100111 (-25)

Números con signo

- El valor decimal de los números con signo.
- Decimal → Binario en tres formatos
  - – Signo-magnitud
  - – Complemento a 2

Números con signo

- Binario a Decimal en formato signo-magnitud

Sumar los pesos de todas las posiciones de los bits de magnitud cuando son 1 e ignorando las posiciones en las que hay 0s

10010101

0010101= $2^4+2^2+2^0= 21$

-21

Números con signo

- Binario a decimal en formato complemento a 2

Números positivos- Sumar los pesos de todas las posiciones de los bits de donde haya 1 e ignorando las posiciones en las que hay 0s

Números negativos- Se da valor negativo al peso del bit de signo, se suman los pesos donde hay 1

01010110

$2^6+2^4+2^2+2^1=$ **+86**

10101010

$-2^7+2^5+2^3+2^1=$ **-86**

Números con signo

- Rango de valores de los números naturales y enteros

  – Con n bits tenemos *2^n* combinaciones diferentes

  – Naturales (sin signo):
    - 0 hasta $2^n - 1$
    - Ejemplo: n=4, rango 0 hasta 15

  – Enteros en formato signo y magnitud:
    - $-(2^{n-1} - 1)$ hasta $+(2^{n-1} - 1)$.
    - Un bit se emplea en el signo, y el cero tiene doble representación
    - Ejemplo: n=4, rango -7 hasta +7

  – Enteros en formato complemento a 2:
    - $-(2^{n-1})$ hasta $+(2^{n-1} - 1)$.
    - El cero no tiene doble representación y el rango es asimétrico
    - Ejemplo: n=4, rango= $-(2^3)=-8$ hasta $2^3-1=+7$

Operaciones aritméticas de números con signo

Operaciones aritméticas de números con signo

- Suma
- Resta
- Multiplicación
- División

Operaciones aritméticas de números con signo

- Cuando hay números negativos, sumamos empleando el formato complemento a dos
- SIEMPRE debemos asegurarnos de que el número de bits es suficiente para contener el resultado
- Descartamos los acarreos
- Ejemplos
  - 00001111+11111010 = 00001001   15+(-6)=9
  - 00010000+11101000 = 11111000   16+(-24)= -8
  - 11111011+11110111 = 11110010   (-5)+(-9)=-14

Operaciones aritméticas de números con signo

Resta

- Tomar el complemento a dos del sustraendo y sumar
- Ejemplo

00001000-00000011=

00001000+11111101= 00000101

Operaciones aritméticas de números con signo

Multiplicación y división

- Se realizarán siempre con números positivos y al resultado se le aplicará el signo correspondiente
- No hacemos multiplicaciones y divisiones en formato complemento a dos

Códigos

Números hexadecimales

- Números decimales, binarios y hexadecimales

DECIMAL	BINARY	HEXADECIMAL
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7
8	1000	8
9	1001	9
10	1010	A
11	1011	B
12	1100	C
13	1101	D
14	1110	E
15	1111	F

Números hexadecimales

- Conversión binario-hexadecimal
- Conversión hexadecimal-binario
- Conversión hexadecimal-decimal
- Conversión decimal-hexadecimal

Números hexadecimales

- Conversión binario a hexadecimal
    1. Se parte el número binario en grupos de 4 bits comenzando por el bit más a la derecha
    2. Se reemplaza cada grupo de 4 bits por su símbolo hexadecimal equivalente.

Números hexadecimales

- Conversión hexadecimal a binario

  Se reemplaza cada símbolo hexadecimal por el grupo de cuatro bits adecuado

Números hexadecimales

- Conversión hexadecimal a decimal

Dos métodos:

- Convertir el número hexadecimal a binario y convertir el binario a decimal
- Suma de pesos: Multiplicar el valor decimal de cada dígito hexadecimal por su peso y sumar los productos

Números hexadecimales

- Conversión decimal a hexadecimal
  - Suma de pesos inversa
  - División sucesiva por 16

Código Decimal Binario (BCD)

Código Decimal Binario (BCD)

Dígitos decimales y BCD

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

Decimal 382 - BCD: 0011 1000 0010

Códigos Gray

- El código Gray
- Código para comunicaciones
- Es muy fiable porque dos símbolos consecutivos solo cambian un bit

DECIMAL	BINARY	GRAY CODE
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
11	1011	1110
12	1100	1010
13	1101	1011
14	1110	1001

Código ASCII

- El código ASCII (Caracteres de control)

NAME	DEC	BINARY	HEX	NAME	DEC	BINARY	HEX
NUL	0	0000000	00	DLE	16	0010000	10
SOH	1	0000001	01	DC1	17	0010001	11
STX	2	0000010	02	DC2	18	0010010	12
ETX	3	0000011	03	DC3	19	0010011	13
EOT	4	0000100	04	DC4	20	0010100	14
ENQ	5	0000101	05	NAK	21	0010101	15
ACK	6	0000110	06	SYN	22	0010110	16
BEL	7	0000111	07	ETB	23	0010111	17
BS	8	0001000	08	CAN	24	0011000	18
HT	9	0001001	09	EM	25	0011001	19
LF	10	0001010	0A	SUB	26	0011010	1A
VT	11	0001011	0B	ESC	27	0011011	1B
FF	12	0001100	0C	FS	28	0011100	1C
CR	13	0001101	0D	GS	29	0011101	1D
SO	14	0001110	0E	RS	30	0011110	1E
SI	15	0001111	0F	US	31	0011111	1F

Códigos ASCII

- El código ASCII (símbolos gráficos 20h – 3Fh)

SYMBOL	DEC	BINARY	HEX	SYMBOL	DEC	BINARY	HEX
space	32	0100000	20	0	48	0110000	30
!	33	0100001	21	1	49	0110001	31
"	34	0100010	22	2	50	0110010	32
#	35	0100011	23	3	51	0110011	33
$	36	0100100	24	4	52	0110100	34
%	37	0100101	25	5	53	0110101	35
&	38	0100110	26	6	54	0110110	36
'	39	0100111	27	7	55	0110111	37
(	40	0101000	28	8	56	0111000	38
)	41	0101001	29	9	57	0111001	39
*	42	0101010	2A	:	58	0111010	3A
+	43	0101011	2B	;	59	0111011	3B
,	44	0101100	2C	<	60	0111100	3C
–	45	0101101	2D	=	61	0111101	3D
.	46	0101110	2E	>	62	0111110	3E
/	47	0101111	2F	?	63	0111111	3F

Códigos ASCII

- El código ASCII (símbolos gráficos 40h – 5Fh)

SYMBOL	DEC	BINARY	HEX	SYMBOL	DEC	BINARY	HEX
@	64	1000000	40	P	80	1010000	50
A	65	1000001	41	Q	81	1010001	51
B	66	1000010	42	R	82	1010010	52
C	67	1000011	43	S	83	1010011	53
D	68	1000100	44	T	84	1010100	54
E	69	1000101	45	U	85	1010101	55
F	70	1000110	46	V	86	1010110	56
G	71	1000111	47	W	87	1010111	57
H	72	1001000	48	X	88	1011000	58
I	73	1001001	49	Y	89	1011001	59
J	74	1001010	4A	Z	90	1011010	5A
K	75	1001011	4B	[	91	1011011	5B
L	76	1001100	4C	\	92	1011100	5C
M	77	1001101	4D	]	93	1011101	5D
N	78	1001110	4E	^	94	1011110	5E
O	79	1001111	4F	_	95	1011111	5F

Códigos Digitales

- El código ASCII (símbolos gráficos 60h – 7Fh)

SYMBOL	DEC	BINARY	HEX	SYMBOL	DEC	BINARY	HEX
`	96	1100000	60	p	112	1110000	70
a	97	1100001	61	q	113	1110001	71
b	98	1100010	62	r	114	1110010	72
c	99	1100011	63	s	115	1110011	73
d	100	1100100	64	t	116	1110100	74
e	101	1100101	65	u	117	1110101	75
f	102	1100110	66	v	118	1110110	76
g	103	1100111	67	w	119	1110111	77
h	104	1101000	68	x	120	1111000	78
i	105	1101001	69	y	121	1111001	79
j	106	1101010	6A	z	122	1111010	7A
k	107	1101011	6B	{	123	1111011	7B
l	108	1101100	6C	|	124	1111100	7C
m	109	1101101	6D	}	125	1111101	7D
n	110	1101110	6E	~	126	1111110	7E
o	111	1101111	6F	Del	127	1111111	7F