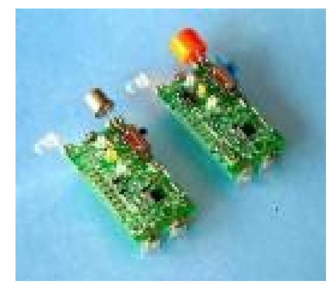
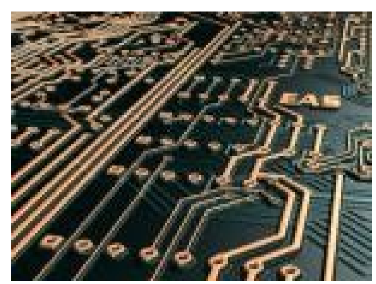
Universidad del País Vasco
Euskal Herriko Unibertsitatea
BILBOKO INGENIARITZA ESKOLA
ESCUELA DE INGENIERÍA DE BILBAO

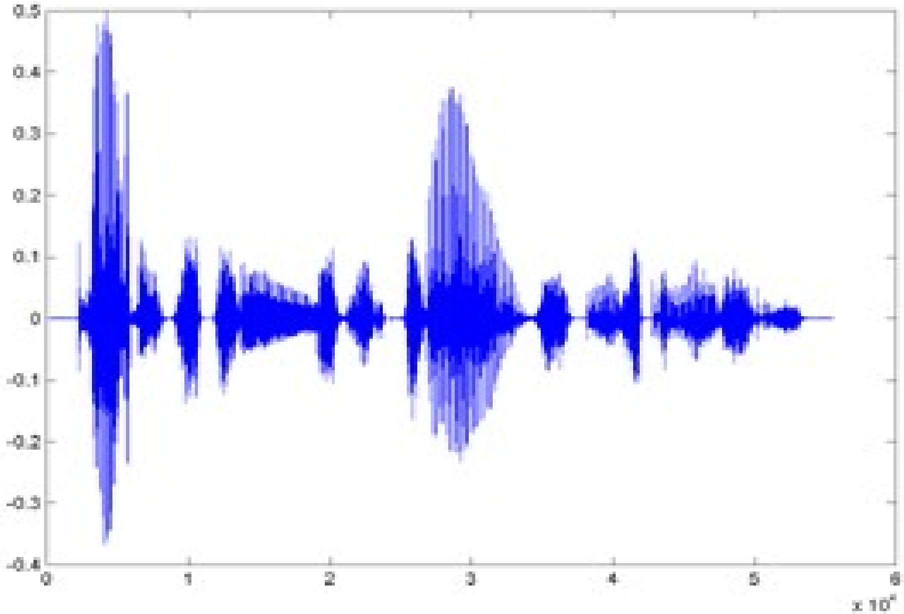
Tema 1: Representación de la información

Elementos de tecnología electrónica





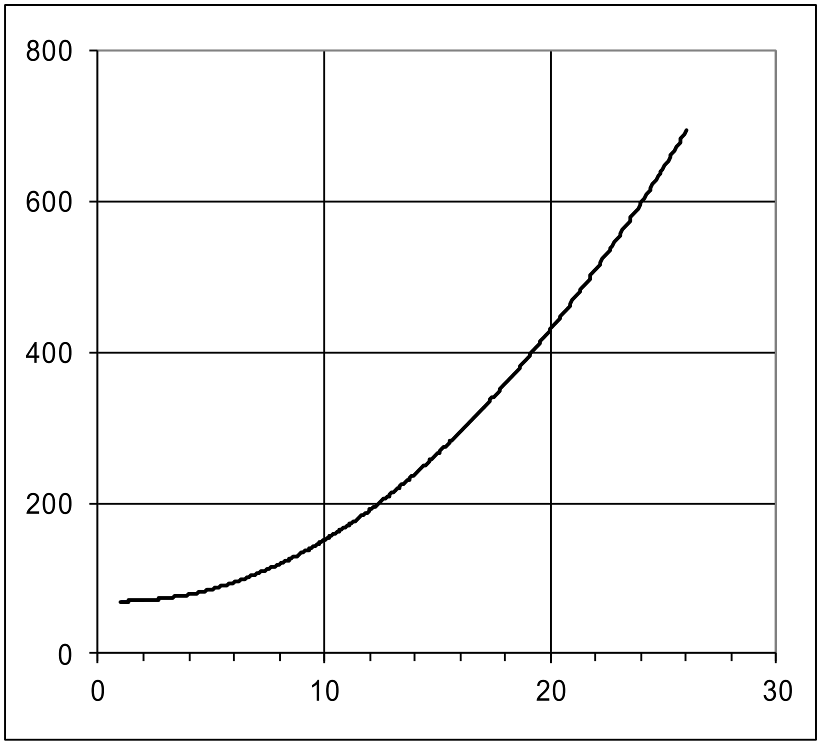
Los campos electromagnéticos son continuos: Electrónica analógica



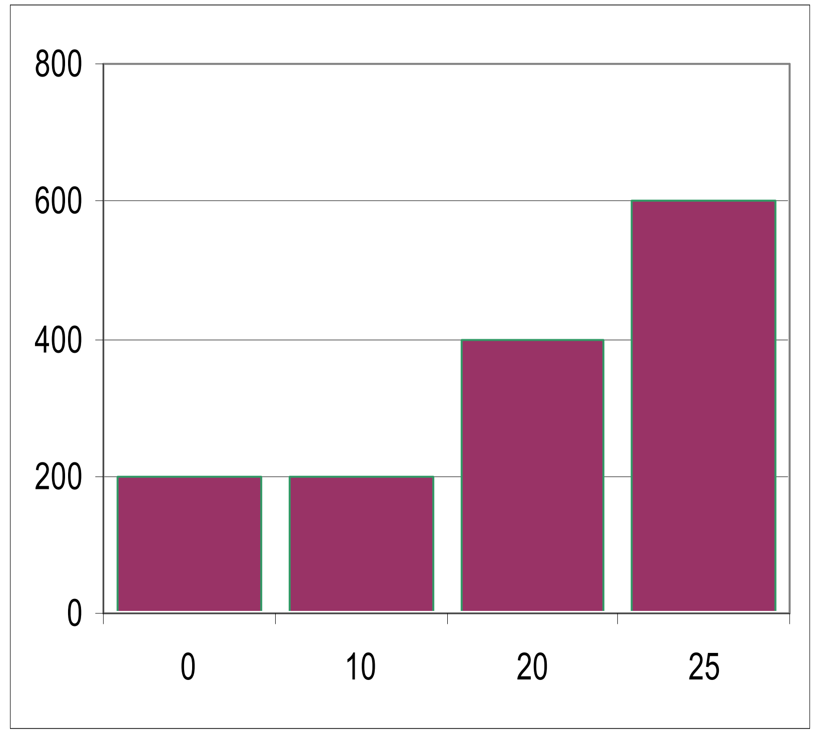
Para cada magnitud (tensión, intensidad) hay infinitos valores posibles:

$$v=f(t)$$
$$i=g(t)$$

Podemos tomar sólo algunos valores de una función continua: discretizar

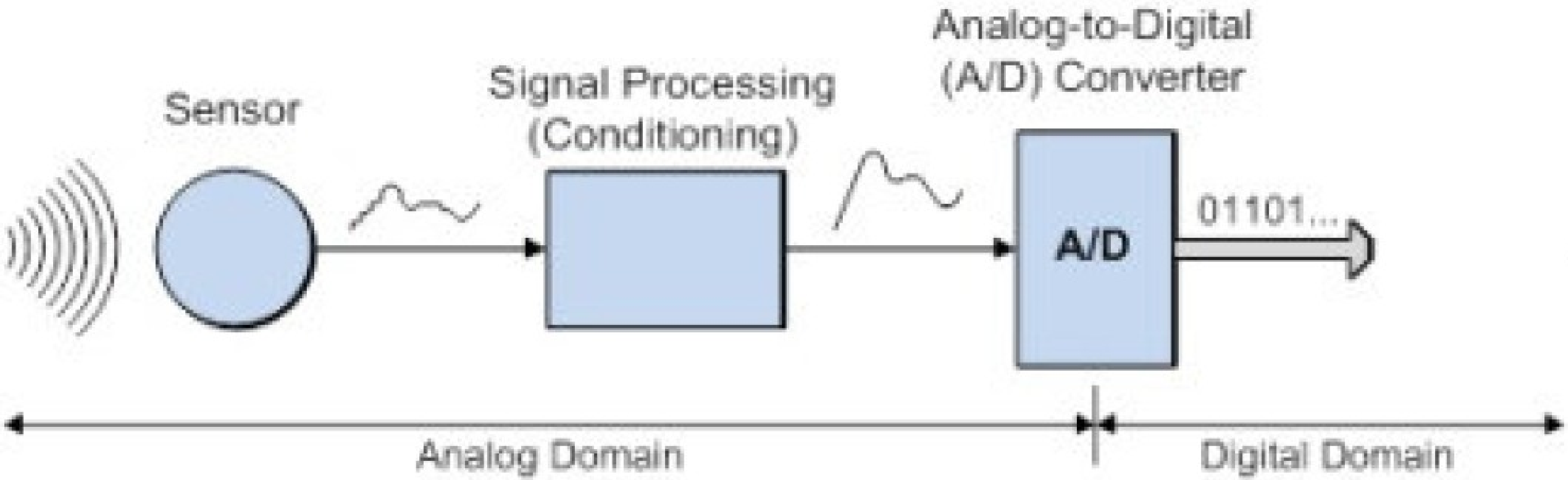


Función analógica: infinitos valores



Función discreta: número finito de valores

Procesado digital de señal

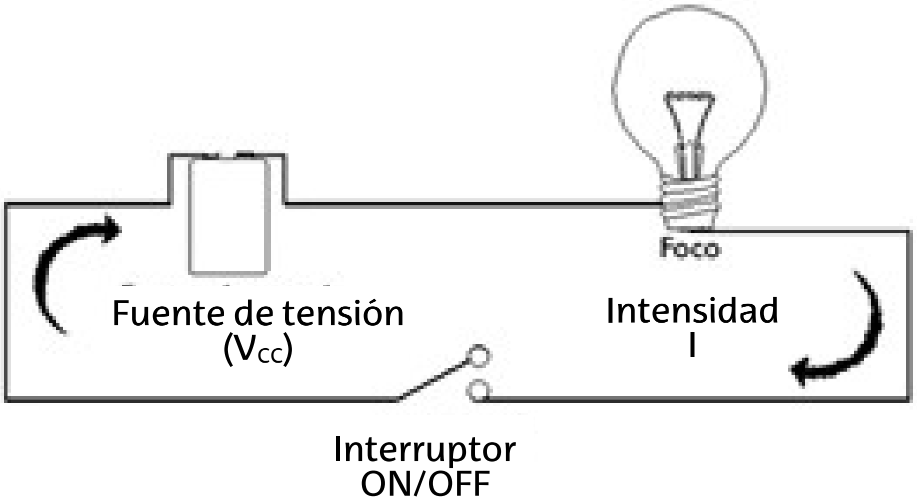


El interruptor reduce la tensión/intensidad a sólo dos valores: on/off

$V_{FOCO}=R_{FOCO}*I$

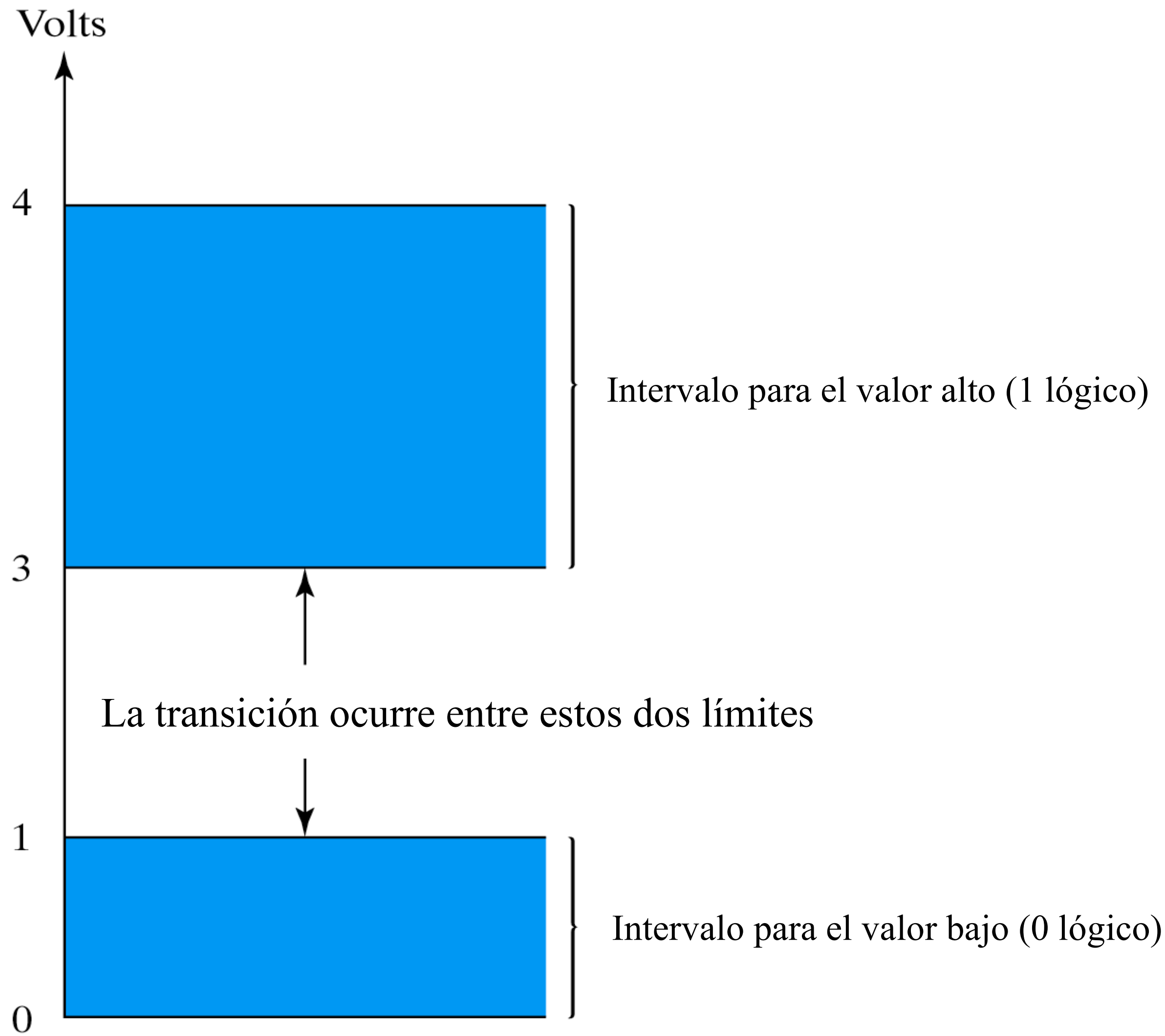
ON: $V_{ON/OFF}=0$; $V_{FOCO}=V_{CC}$

OFF: $I=0$; $V_{FOCO}=0$



Electrónica de dos valores ⇔ Electrónica digital

Tensión digital



Estos valores pueden cambiar según la familia de circuito integrado

Representación de la información

- La electrónica digital se basa en dos valores de tensión.
- Por ello, utilizaremos dos valores numéricos para representar las tensiones variables.
- Los valores serán el 0 (tensión baja: L) y el 1 (tensión alta: H).

Sistemas de numeración posicional

$$N = \sum_{i=-k}^{n-1} d_i \cdot b^i$$

d_i=i-ésima cifra, b=base

	b	d	$N = 2001_{10}$
Binario	2	0-1	11111010001
Octal	8	0-7	3721
Decimal	10	0-9	2001
Hexadecimal	16	0-9,A-F	7D1

Sistemas de numeración posicional

- En electrónica digital representamos las señales mediante dos valores.
- El sistema de numeración de base 2 (binario) sólo usa dos cifras.
- Por tanto, usaremos este sistema para representar información numérica en los sistemas de electrónica digital.

Sistemas de numeración posicional

- Los números menores que 1 tienen que usar cifras a la izquierda de la coma, con exponentes negativos.

- De este modo se pueden expresar números reales:

$$14{,}75_{10} = 1\cdot10^{1}+4\cdot10^{0}+7\cdot10^{-1}+5\cdot10^{-2}$$

$$1110{,}11_{2} = 1\cdot2^{3}+1\cdot2^{2}+1\cdot2^{1}+0\cdot2^{0}+1\cdot2^{-1}+1\cdot2^{-2} =$$
$$8+4+2+0+1/2+1/4=14{,}75$$

Conversiones entre sistemas de numeración

Binario ⇔ Octal ⇔ Hexadecimal

Hexadecimal	7 B A 3 , B C 4
Binario	0111101110100011,101111000100
Octal	7 5 6 4 3 , 5 7 0 4

Conversiones entre sistemas de numeración

Decimal ⟺ **otros:**

- División entera del **número** que se quiere convertir : la **base** buscada ⟶ seguir dividiendo hasta que el cociente sea más pequeño que el divisor ⟶ los restos son el número en la nueva base, de derecha a izquierda.

- **Parte decimal** del número que se quiere convertir x **base** buscada ⟶ seguir multiplicando parte decimal mientras quede ⟶ las partes enteras son el número en la nueva base, de izquierda a derecha.

Precisión finita

- En los sistemas digitales el número de cifras es fijo, pues representa a señales de tensión.
- Por tanto, son finitos los números que pueden ser representados: coma fija, precisión finita.
- En base 2, con *n* cifras a la izquierda de la coma y *k* cifras a la derecha de la coma:

$$N_{máx}=2^{n}-1+(1-2^{-k})$$

$$N_{mín}=2^{-k}$$

Sistema decimal codificado en binario: BCD

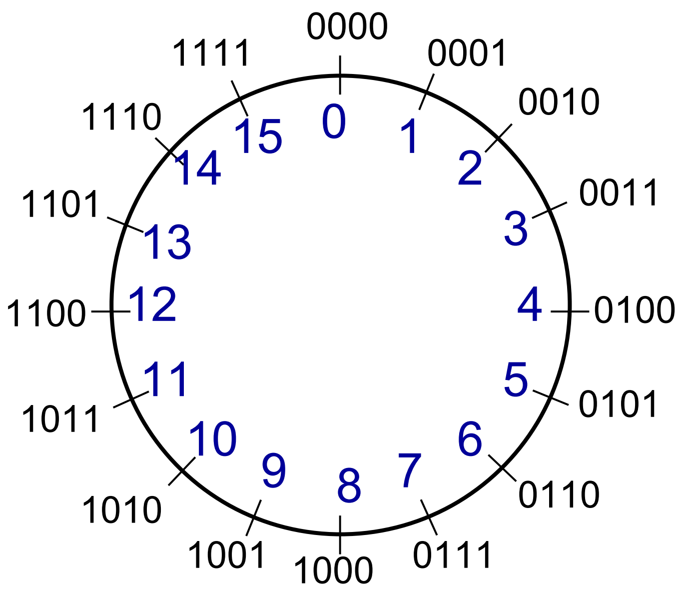
Decimal	Número BCD
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001
No usadas	1010 1011 1100 1101 1110 1111

- Las personas estamos más acostumbradas al código decimal→El más usado para codificar con 0 y 1: BCD (*Binary Coded Decimal*).
- 396_{10}= 0011 1001 0110 (16 combinaciones, 6 no se usan)
- Las operaciones aritméticas no son válidas→Reglas especiales

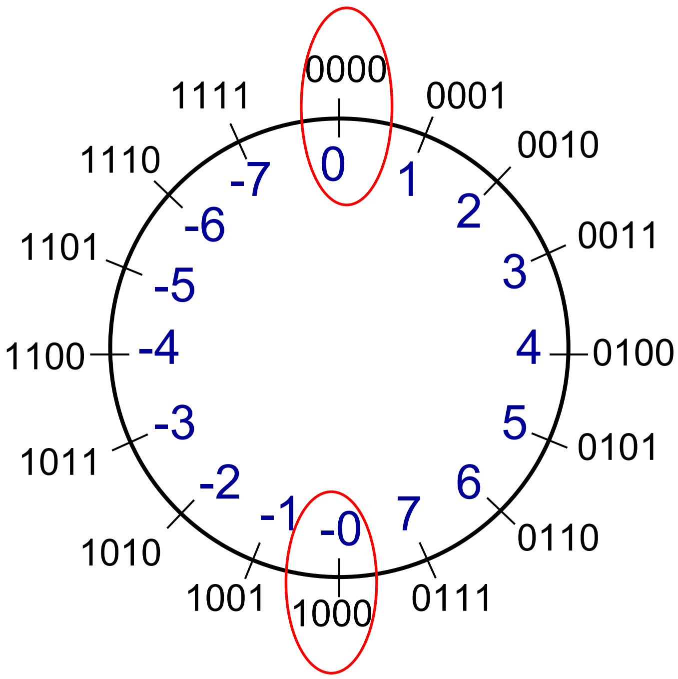
Representación de números con signo

- Con n bits se pueden representar 2^n números enteros.
- Para representar números enteros positivos y negativos, tendremos que dividir entre dos los 2^n números: 2^{n-1} enteros positivos y otro tanto negativos.
- El cero se considera positivo.
- El método más empleado es el complemento a 2.

Representación de números con signo



Números binarios positivos



Magnitud con signo

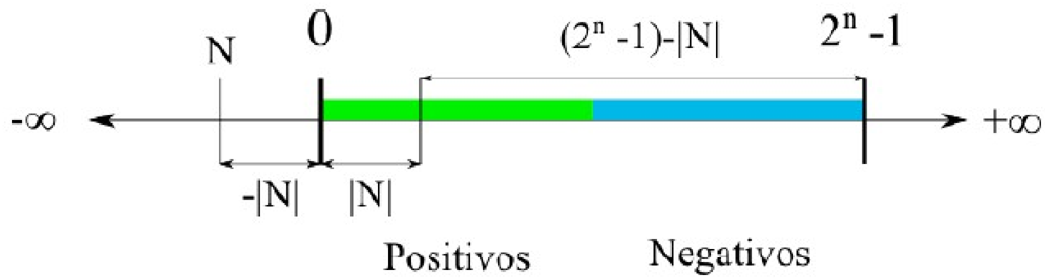
Representación de números con signo

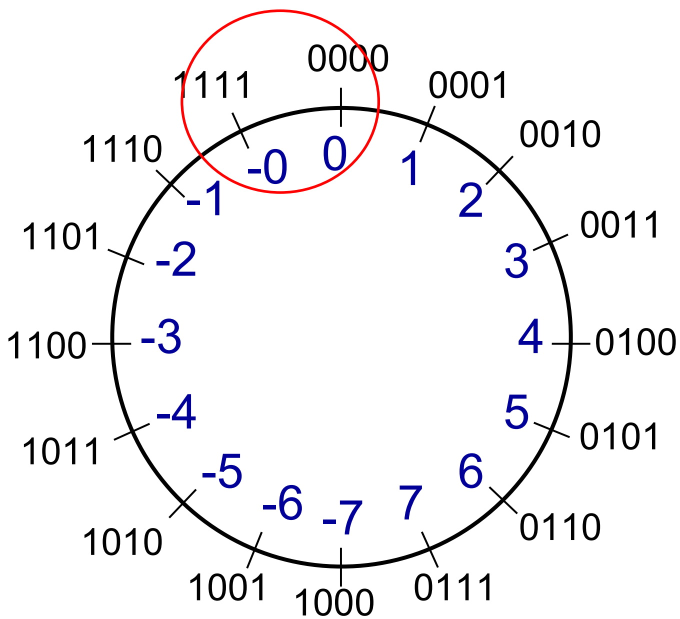
- El sistema de magnitud con signo es el más sencillo, pero tiene desventajas:
  - El cero tiene dos símbolos: +0 y -0
  - Las operaciones aritméticas no son válidas con números de este sistema

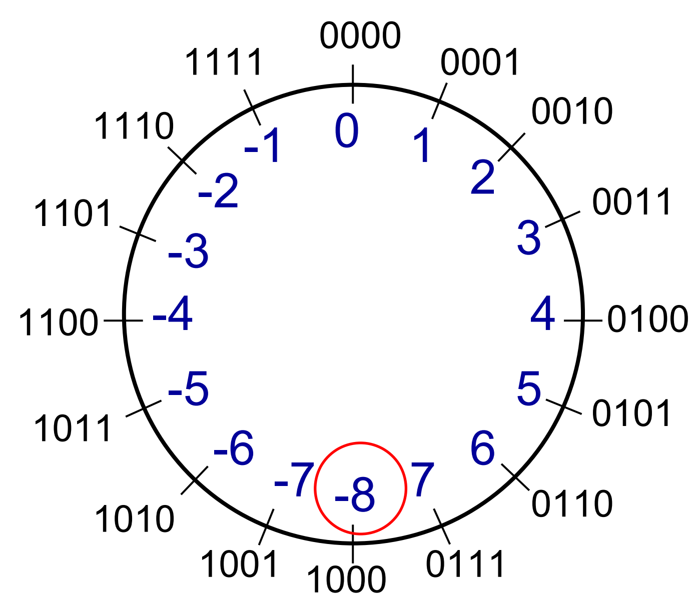
- El número más positivo es: $N_{máx}=2^{n-1}-1$

- El número más negativo es : $N_{mín}=-(2^{n-1}-1)$

Representación de números con signo







Complemento a 1

$A^{(1)} = 2^4 - 1 - |A|$

Complemento a 2

$A^{(2)} = 2^4 - |A| = A^{(1)} + 1$

Representación de números con signo

- El complemento a 1 se puede implementar invirtiendo 0s y 1s al número en valor absoluto.
- Sirve para implementar el complemento a 2, sumándole 1.
- El complemento a 2 sólo asigna un código al cero, por lo que hay un número negativo más: -2^{n-1} .
  - El número más positivo es: $N_{máx}=2^{n-1}-1$.
  - El número más negativo es : $N_{mín}=-2^{n-1}$.

Representación de números con signo

Sumandos	0	0	1	1
	+0	+1	+0	+1
Suma	0	1	1	0
Acarreo	0	0	0	1

Decimal	Complemento a 1	Complemento a 2
10	00001010	00001010
+ (–3)	11111100	11111101
+7	1 00000110	1 00000111
	Acarreo 1	Descartado
	00000111	

Suma binaria con números negativos

Representación de números con signo

- Complemento a 2 → descartar acarreo.
- Complemento a 1 → acarreo cíclico.
- Al sumar números de distinto signo, no aparece desbordamiento (overflow).
- Si son del mismo signo → bit de signo del resultado diferente → hay desbordamiento.

Representación de números con signo

- El sistema de exceso a 2^{n-1} suma a todos los números (positivos y negativos) 2^{n-1}, de modo que el resultado siempre es positivo.
- Para saber el valor de un número tenemos que restarle 2^{n-1}.
- Este sistema no vale para la suma, pues siempre aparece un 2^{n-1}de más.

$$A+2^{n-1}+B+2^{n-1}=\underbrace{((A+B)+2^{n-1})}_{\text{CORRECTO}}+2^{n-1}$$

Representación de números con signo

Decimal	Magnitud con signo	Complemento a 1	Complemento a 2	Exceso a 8
-8	------	------	1000	0000
-7	1111	1000	1001	0001
-6	1110	1001	1010	0010
-5	1101	1010	1011	0011
-4	1100	1011	1100	0100
-3	1011	1100	1101	0101
-2	1010	1101	1110	0110
-1	1001	1110	1111	0111
-0	1000	1111	------	------
0	0000	0000	0000	1000
1	0001	0001	0001	1001
2	0010	0010	0010	1010
3	0011	0011	0011	1011
4	0100	0100	0100	1100
5	0101	0101	0101	1101
6	0110	0110	0110	1110
7	0111	0111	0111	1111

Notación científica: números en coma flotante

$$\boxed{N = f \cdot 10^{e}}$$

f: fracción o mantisa → precisión
e: exponente → rango

Coma flotante: Versión binaria.

- **ANSI/IEEE Std. 754 (1985)**
  - Exponente :
    - Exceso a $2^{n-1}-1$ en números normalizados.
    - Todo ‘0’y todo ‘1’ reservados.
  - Fracción normalizada con el primer 1 a la izquierda de la coma.

Notación científica: números en coma flotante

$$\boxed{N = f \cdot 10^{e}}$$

f: fracción o mantisa → precisión.

e: exponente → rango.

NORMALIZADA

$$15.000 = 0{,}15 \cdot 10^{5} = 15 \cdot 10^{3} = 0{,}0015 \cdot 10^{7} \rightarrow \begin{cases} \boxed{f = 0{,}15 \qquad e = 5} \\ f = 15 \qquad e = 3 \\ f = 0{,}0015 \quad e = 7 \end{cases}$$

Notación científica: números en coma flotante

- Variante para el computador de este sistema en base 2.
- Si el número a la derecha de la coma de la fracción es ‘1’, se dice que está *normalizada*.
- Si hay ceros a la derecha de la coma →desplazamos el número a la izquierda decrementando el exponente (la fracción queda normalizada sin cambiar su valor).

Notación científica: números en coma flotante

Ejemplo:

23	22, 21 ... 16	15, 14, 13, 12 .... 2, 1, 0
+ -	exponente	fracción

Base=2, exponente en exceso a 64.

No normalizado:

$$01010100{,}0000000000011011=2^{20}\cdot(2^{-12}+2^{-13}+2^{-15}+2^{-16})=432$$

Normalizado:

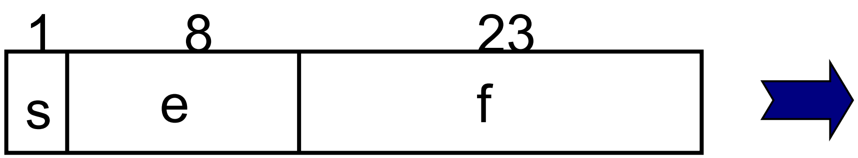
$$01001001{,}1101100000000000=2^{9}\cdot(2^{-1}+2^{-2}+2^{-4}+2^{-5})=432$$

Notación científica: IEEE 754

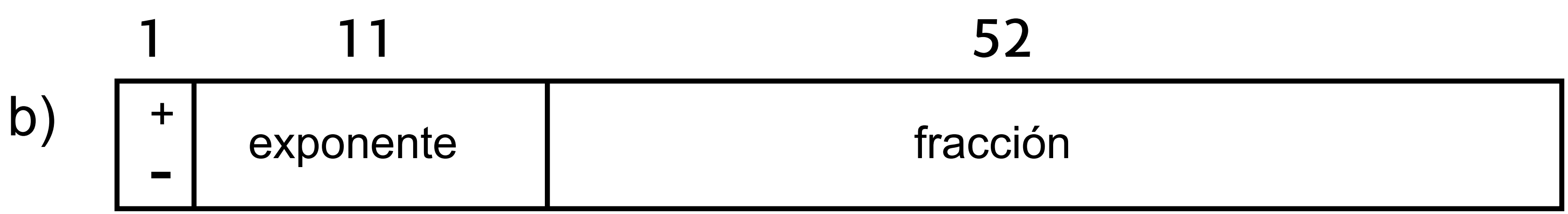
ANSI/IEEE Std. 754 (1985)

Modo normalizado: El primer 1 a la izquierda de la coma, 1 implícito en la fracción.

Exponente en exceso a $2^{n-1}-1$.



$$(-1)^s \times 2^{(e-127)} \times (1+f)$$



a) Precisión sencilla *b) Doble precisión*

Notación científica: IEEE 754

Excepción del significado de la fracción y el exponente (todo ‘0’ y todo ‘1’ reservado para valores especiales):

Normalizado	±	0 < Exp < Max	Cualquier grupo de bits
No normalizado	±	0	Cualquier grupo de bits distinto de cero
Cero	±	0	0
Infinito	±	1 1 1…1	0
No es número	±	1 1 1…1	Cualquier grupo de bits distinto de cero

Bit de signo

Códigos alfanuméricos: ASCII

ASCII: *American Standard Code for Infomation Interchange*

- 7bit → 128 caracteres.
- 1byte :

0	código ASCII

- MSB=1 → otras 128 combinaciones, para caracteres acentuados o letras griegas (Latin-1).

$$\boxed{0} \quad \underbrace{B7\ B6\ B5}_{\text{Columna}}\ \underbrace{B4\ B3\ B2\ B1\ B0}_{\text{Fila}}$$

Códigos alfanuméricos: ASCII

	$B_7B_6B_5$							
$B_4B_3B_2B_1$	000	001	010	011	100	101	110	111
0000	NULL	DLE	SP	0	@	P	`	p
0001	SOH	DC1	!	1	A	Q	a	q
0010	STX	DC2	"	2	B	R	b	r
0011	ETX	DC3	#	3	C	S	c	s
0100	EOT	DC4	$	4	D	T	d	t
0101	ENQ	NAK	%	5	E	U	e	u
0110	ACK	SYN	&	6	F	V	f	v
0111	BEL	ETB	'	7	G	W	g	w
1000	BS	CAN	(	8	H	X	h	x
1001	HT	EM	)	9	I	Y	i	y
1010	LF	SUB	*	:	J	Z	j	z
1011	VT	ESC	+	;	K	[	k	{
1100	FF	FS	,	<	L	\	l	|
1101	CR	GS	-	=	M	]	m	}
1110	SO	RS	.	>	N	^	n	~
1111	SI	US	/	?	O	_	o	DEL

Códigos alfanuméricos: ASCII

- La tabla anterior es para B_8=0, con B_8=1 aparecen caracteres acentuados y los correspondientes a varias lenguas europeas.
- ¿Pero qué pasa con otras lenguas?→Tabla de código, una tabla para cada lengua.
- Los caracteres chinos son muchos más que 256, no sirve el sistema de tabla de código.

Códigos alfanuméricos: UNICODE

- El sistema UNICODE asigna a cada carácter un único número entero (punto de código).
- El sistema se inició con 16 bits (UTF-16), pero actualmente utiliza 32 bits (UTF-32) para cada carácter.
- Los $2^{32} \approx 4 \times 10^9$ caracteres son asignados por un consorcio creado por Apple, Microsoft y Sun, entre otras empresas, desde 1991.

Códigos alfanuméricos: UNICODE

	Control		ASCII						Control		Latin 1					
	000	001	002	003	004	005	006	007	008	009	00A	00B	00C	00D	00E	00F
0	CTRL	CTRL	SPACE	0	@	P	`	p	CTRL	CTRL	NB SP	°	À	Ð	à	D
1	CTRL	CTRL	!	1	A	Q	a	q	CTRL	CTRL	¡	±	Á	Ñ	á	ñ
2	CTRL	CTRL	"	2	B	R	b	r	CTRL	CTRL	¢	²	Â	Ò	â	ò
3	CTRL	CTRL	#	3	C	S	c	s	CTRL	CTRL	£	³	Ã	Ó	ã	ó
4	CTRL	CTRL	$	4	D	T	d	t	CTRL	CTRL	¤	´	Ä	Ô	ä	ô
5	CTRL	CTRL	%	5	E	U	e	u	CTRL	CTRL	¥	µ	Å	Õ	å	õ
6	CTRL	CTRL	&	6	F	V	f	v	CTRL	CTRL	¦	¶	Æ	Ö	æ	ö
7	CTRL	CTRL	'	7	G	W	g	w	CTRL	CTRL	§	·	Ç	×	ç	÷
8	CTRL	CTRL	(	8	H	X	h	x	CTRL	CTRL	¨	¸	È	Ø	è	ø
9	CTRL	CTRL	)	9	I	Y	i	y	CTRL	CTRL	©		É	Ù	é	ù
A	CTRL	CTRL	*	:	J	Z	j	z	CTRL	CTRL	ª	º	Ê	Ú	ê	ú
B	CTRL	CTRL	+	;	K	[	k	{	CTRL	CTRL	«	»	Ë	Û	ë	û
C	CTRL	CTRL	,	<	L	\	l	|	CTRL	CTRL	¬	¼	Ì	Ü	ì	ü
D	CTRL	CTRL	-	=	M	]	m	}	CTRL	CTRL	-	½	Í	Ý	í	ý
E	CTRL	CTRL	.	>	N	^	n	~	CTRL	CTRL	®	¾	Î	?b	î	'b
F	CTRL	CTRL	/	?	O	_	o	CTRL	CTRL	CTRL	¯	¿	Ï	ß	ï	ÿ

Los primeros 256 caracteres de UNICODE