

## 2 Tipus de senyals

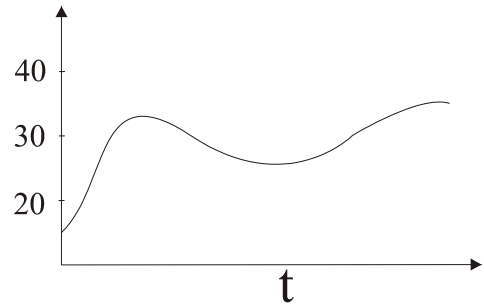
Un senyal és una informació que ens dóna un determinat element. Aquest element pot ser una magnitud (llum, tensió, freqüència, ...), un objecte (disc de trànsit, un nivell,...), una situació, etc. A partir d'aquest podrem actuar en conseqüència en un sentit o un altre.

Els que ens interessin ara es poden dividir en dos tipus: analògics i digitals.

### 2.1 Senyals analògics

Varien de forma contínua. És a dir, dins el seu rang poden prendre infinits valors diferents. Per exemple, si representam la velocitat a que hem vengut a l'institut, es pot semblar a la figura.

Entre el moment que hem anat a 30 i el que hem anat a 31, haurem passat per tots els valors intermitjos.



### 2.2 Senyals digitals

Només adopten valors concrets d'entre els disponibles. Per exemple el valor d'un dard en una diana. En particular ens interessarem pels senyals digitals binaris.

Els **senyals digitals binaris** només poden prendre dos estats:

- 1 → Hi ha senyal → Activat
- 0 → No hi ha senyal → Desactivat

Exercici: Descriu perquè un rellotge digital s'anomena així. Compara-ho amb un d'analògic.

### 3 Treball amb senyals digitals

Sabem que els ordinadors funcionen en sistema binari. Anem a veure ara com funciona aquest.

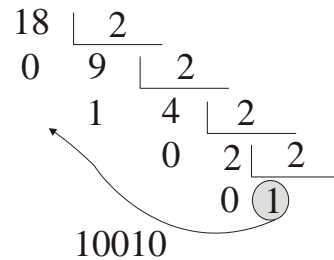
Nosaltres normalment calculam en sistema decimal (o en base 10). En aquest sistema tenim deu xifres (del 0 al 9). En binari només n'hi ha dues: 0 i 1. Amb aquestes es pot representar qualsevol nombre i, fins i tot, qualsevol caràcter.

#### 3.1 Representació de nombres

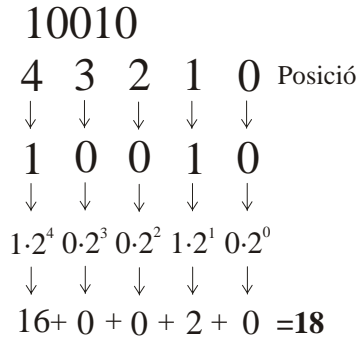
Anirem a veure l'equivalència entre el sistema binari i el decimal. I, de passada, ens començarem a habituar a aquest tipus de notació.

##### Pas de decimal a binari

Passar de decimal a binari és tan senzill com dividir per 2. Agafam el nombre a canviar de base i calculam quants de grups de 2 podem fer, és a dir, dividim per 2. Si el quocient encara es pot dividir es divideix. I així successivament fins que no es pugui dividir més. Arribats a aquest punt, agafam el darrer quocient i els residus i els col·locam ordenadament.



##### Pas de binari a decimal



Si abans anavem fent divisions, ara haurem de fer multiplicacions. Calcularem potències de dos i les multiplicarem pel dígit de la posició que ocupen. Per acabar les sumarem totes.

##### Operacions en binari

Totes les operacions que coneixem es poden realitzar en binari. Haurem de tenir en compte que 1+1 és 0 i que en duim un. I haurem "d'aprendre" les taules de l'u i del zero.

Exemples:

$$\begin{array}{r} 1\ 0\ 0\ 1\ 0 \\ +\ 1\ 0\ 1\ 1 \\ \hline 1\ 1\ 1\ 0\ 1 \end{array} \qquad \begin{array}{r} 1\ 0\ 1\ 1 \\ -\ 1\ 0\ 1 \\ \hline 0\ 1\ 1\ 0 \end{array}$$

$$\begin{array}{r} 1\ 0\ 0\ 1\ 0 \\ \times\ 1\ 0\ 1 \\ \hline 1\ 0\ 0\ 1\ 0 \\ 1\ 0\ 0\ 1\ 0 \\ \hline 1\ 0\ 0\ 1\ 0 \\ \hline 1\ 0\ 1\ 1\ 0\ 1\ 0 \end{array}$$

$$\begin{array}{r} 1\ 0\ 1\ 1\ 0\ 1\ 0 \\ \underline{1\ 0\ 1} \\ 0\ 1\ 0\ 1 \\ \underline{1\ 0\ 1} \\ 0\ 0 \end{array} \qquad \begin{array}{r} \underline{1\ 0\ 1} \\ 1\ 0\ 0\ 1\ 0 \end{array}$$

#### 3.2 Representació de caràcters

Per representar caràcters (lletres i símbols) haurem d'utilitzar un codi. Això és una sèrie d'equivalències entre valors binaris i lletres. N'hi ha uns quants i per a poder-nos entendre ens hem de posar d'acord en quin utilitzarem. El més comú és el codi ASCII (American Standard Code for Information Interchange).

La primera versió del codi ASCII tenia 7 bits i l'actual en té 8.  
Ex: Quants caràcters tindrà el codi ASCII de 7 bits? I el de 8?

Ctrl	Dec	Hex	Char	Code	Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char
	0	00		NUL	32	20	sp	64	40	@	96	60	`
^A	1	01	☐	SOH	33	21	!	65	41	A	97	61	a
^B	2	02	☐	SIX	34	22	"	66	42	B	98	62	b
^C	3	03	♥	EIX	35	23	#	67	43	C	99	63	c
^D	4	04	♦	EDI	36	24	\$	68	44	D	100	64	d
^E	5	05	♣	ENQ	37	25	%	69	45	E	101	65	e
^F	6	06	♠	ACK	38	26	&	70	46	F	102	66	f
^G	7	07	•	BEL	39	27	'	71	47	G	103	67	g
^H	8	08	◼	BS	40	28	(	72	48	H	104	68	h
^I	9	09	○	HI	41	29	)	73	49	I	105	69	i
^J	10	0A	◻	LF	42	2A	*	74	4A	J	106	6A	j
^K	11	0B	♂	VI	43	2B	+	75	4B	K	107	6B	k
^L	12	0C	♀	FF	44	2C	,	76	4C	L	108	6C	l
^M	13	0D	♪	CR	45	2D	-	77	4D	M	109	6D	m
^N	14	0E	♫	SO	46	2E	.	78	4E	N	110	6E	n
^O	15	0F	✳	SI	47	2F	/	79	4F	O	111	6F	o
^P	16	10	▼	SLE	48	30	0	80	50	P	112	70	p
^Q	17	11	▲	CS1	49	31	1	81	51	Q	113	71	q
^R	18	12	↑	DC2	50	32	2	82	52	R	114	72	r
^S	19	13	!!	DC3	51	33	3	83	53	S	115	73	s
^T	20	14	¶	DC4	52	34	4	84	54	T	116	74	t
^U	21	15	⊗	NAK	53	35	5	85	55	U	117	75	u
^V	22	16	▬	SYN	54	36	6	86	56	V	118	76	v
^W	23	17	≡	EIB	55	37	7	87	57	W	119	77	w
^X	24	18	↑	CAN	56	38	8	88	58	X	120	78	x
^Y	25	19	↓	EM	57	39	9	89	59	Y	121	79	y
^Z	26	1A	→	SIB	58	3A	:	90	5A	Z	122	7A	z
^_	27	1B	←	ESC	59	3B	;	91	5B	[	123	7B	{
^X	28	1C	└	FS	60	3C	<	92	5C	\	124	7C	
^]	29	1D	⊕	GS	61	3D	=	93	5D	]	125	7D	}
^^	30	1E	▲	RS	62	3E	>	94	5E	^	126	7E	~
^-	31	1F	▼	US	63	3F	?	95	5F	_	127	7F	Δ†

Ex: Com seria en ASCII la paraula "Sis".

## 4 Mesures d'informació

Per mesurar la informació utilitzam unes unitats específiques. Anem a recordar-les.

- BIT (Binary digIT). Dígit binari. Representa un zero o un u.
- Byte. 8 bits. És la quantitat d'informació necessària per representar un caràcter en codi ASCII.
- Quilobyte (K, Kb). 1024 bytes. Equival, aproximadament, a la quantitat de caràcters que hi caben dins un full mecanografiat a doble espai.
- Megabyte (Mb). 1024 Kb.
- Gigabyte (Gb). 1024 Mb.
- Terabyte (Tb). 1024 Gb.

Segons l'element que es pretengui mesurar s'utilitzarà una unitat o una altra.

Ex. Quina unitat utilitzaries per mesurar els següents elements i perquè?

- a) Memòria
- b) Entrades d'un xip
- c) grandària videojoc
- d) espai lliure a disc
- e) espai ocupat a disc

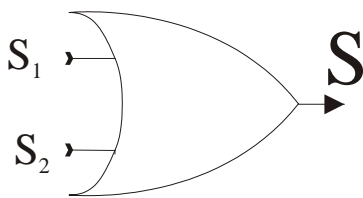
## 5 Portes lògiques

Hem comentat que un senyal binari pot representar dos estats (encès/apagat). Quan combinam estats per fer operacions més complexes s'utilitzen funcions lògiques. El funcionament d'aquestes fou estudiat per George Boole, per això també s'anomena àlgebra de Boole.

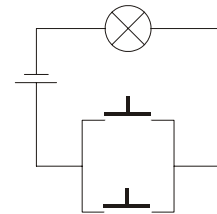
Les funcions més simples s'anomenen portes lògiques. Anem a veure quines són i com es representen.

### 5.1 Porta O (or)

Produeix un senyal de sortida quan hi ha algun senyal d'entrada.



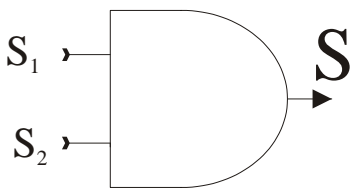
$S_1$	$S_2$	$S$
0	0	0
0	1	1
1	0	1
1	1	1



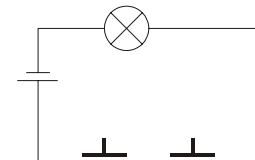
Exemple: Volem que es pugui encendre un llum des de dos llocs diferents. És a dir, s'ha de poder encendre des d'un lloc o des de l'altre.

### 5.2 Porta I (and)

Produeix un senyal de sortida quan s'activen tots els senyals d'entrada.



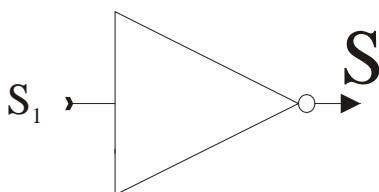
$S_1$	$S_2$	$S$
0	0	0
0	1	0
1	0	0
1	1	1



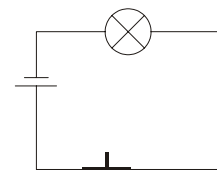
Exemple: Per veure la televisió l'hem d'endollar i pitjar el botó d'endegada. Si només feim una d'aquestes accions no funcionarà.

### 5.3 Porta NO (not)

Produeix un senyal de sortida oposat al de l'entrada.



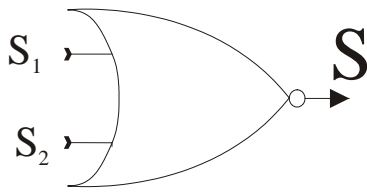
$S_1$	$S$
0	1
1	0



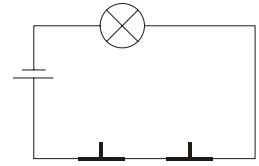
Exemple: Les faroles del carrer s'encenen quan no hi ha llum i s'apaguen quan n'hi ha.

### 5.4 Porta NOR

Produeix un senyal de sortida mentre no s'activi un dels senyals d'entrada.



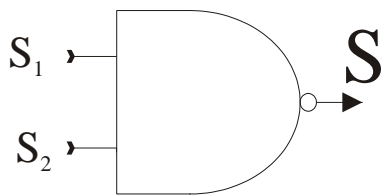
$S_1$	$S_2$	$S$
0	0	1
0	1	0
1	0	0
1	1	0



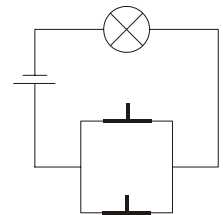
Exemple: Una màquina que està en funcionament mentre no es pitgi un dels botons de seguretat.

### 5.5 Porta NAND

Produeix un senyal de sortida mentre no s'activin tots els senyals d'entrada.

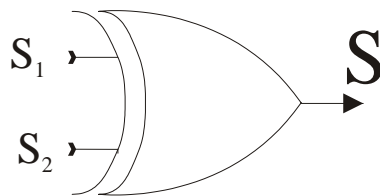


$S_1$	$S_2$	$S$
0	0	1
0	1	1
1	0	1
1	1	0



### 5.6 Porta XOR

Produeix un senyal de sortida quan s'activa un, i només un, dels senyals d'entrada.

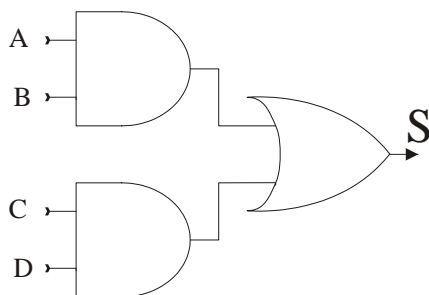


$S_1$	$S_2$	$S$
0	0	0
0	1	1
1	0	1
1	1	0

## 6 Funcions lògiques

Tot el funcionament d'un ordinador es pot descriure a partir de les portes lògiques. Nosaltres no veurem funcions tan complexes. Però en veurem algunes amb uns exemples.

Ex: Sabries fer el circuit de la funció que representa el diagrama? I la taula de veritat?



Ex: Fes el diagrama de portes lògiques. I la taula de veritat.

