



**UNIVERSITA DEGLI STUDI DI FOGGIA**

**Dipartimento di Agraria**

Cdl in Ingegneria dei Sistemi Logistici per l'Agroalimentare

*Corso integrato di Sistemi di Elaborazione*

---

# Modulo I

**Prof. Crescenzo Gallo**

*crescenzo.gallo@unifg.it*

# La rappresentazione delle informazioni

# Il bit

*Si consideri un alfabeto di 2 simboli: '0' e '1'*



*Che tipo di informazione si può rappresentare con un bit?*

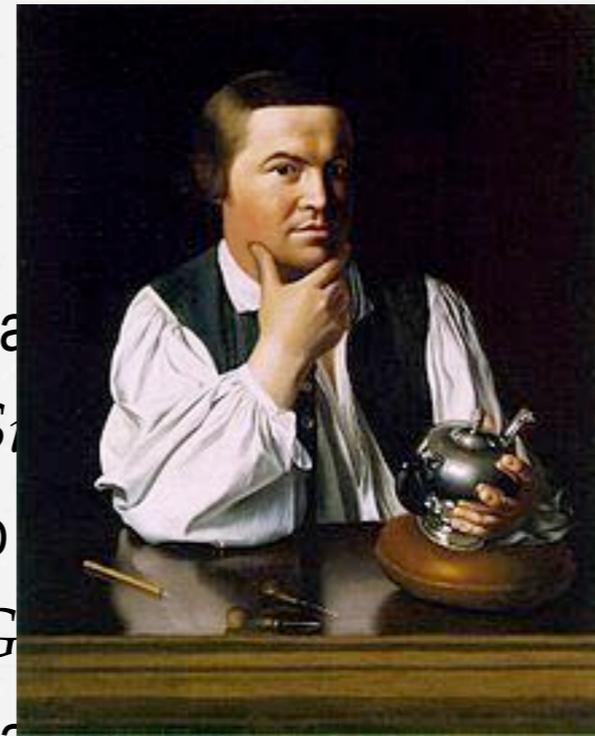
# Il bit

## La Cavalcata di Paul Révére

Paul Révére e la sua cavalcata di mezzanotte del 19 aprile 1775.

Egli disse al suo amico: *“Se gli inglesi giungeranno dalla terra o dal mare, questa notte appendi una lanterna sul campanile della Old North Church come segnale.*”

*Una lanterna se vengono da terra, due se vengono dal mare. Io sarò sulla riva opposta pronto a cavalcare e dare l'allarme generale. In ogni villaggio del Middlesex e in ogni fattoria, i contadini saranno pronti a battaglia.*”



Entrambe

S

Solo

G

Entrambe accese (II).

*Attacco dal mare*

*a terra*

# Il bit

Con un solo bit è possibile gestire un'informazione binaria, cioè un'informazione che può specificare uno tra due valori possibili (es. un punto di un'immagine bianco o nero).

Quanti stati possibili può assumere un insieme di bit ?

00	000	0000
01	001	0001
10	010	0010
11	011	0011
	100	0100
	101	0101
	110	0110
	111	0111
		1000
		1001
		1010
		1011
		1100
		1101
		1110
		1111

2 bit → 4 stati

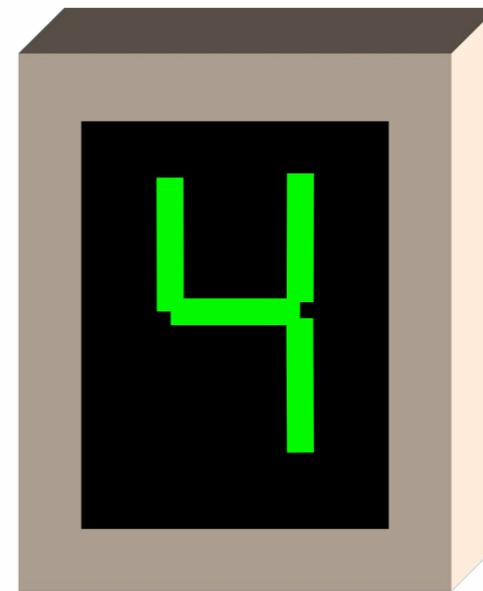
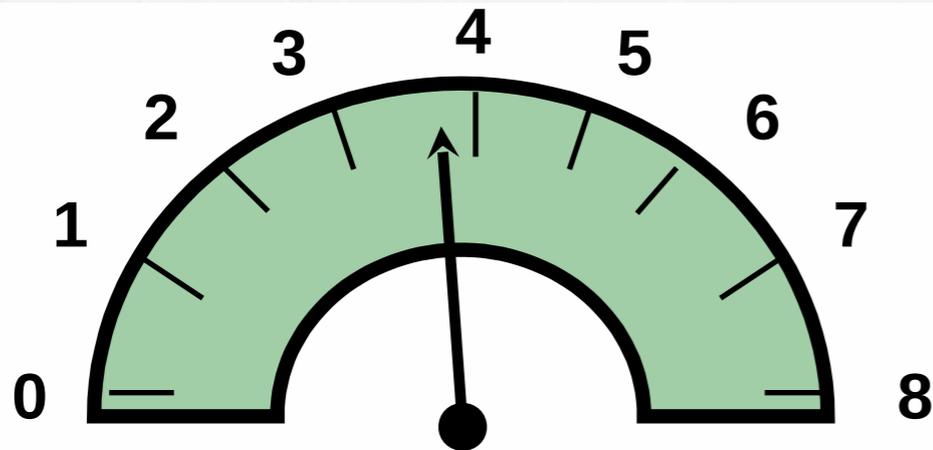
3 bit → 8 stati

4 bit → 16 stati

...

Con  $N$  bit:  $2^N$   
possibilità diverse

# Digitale vs Analogico

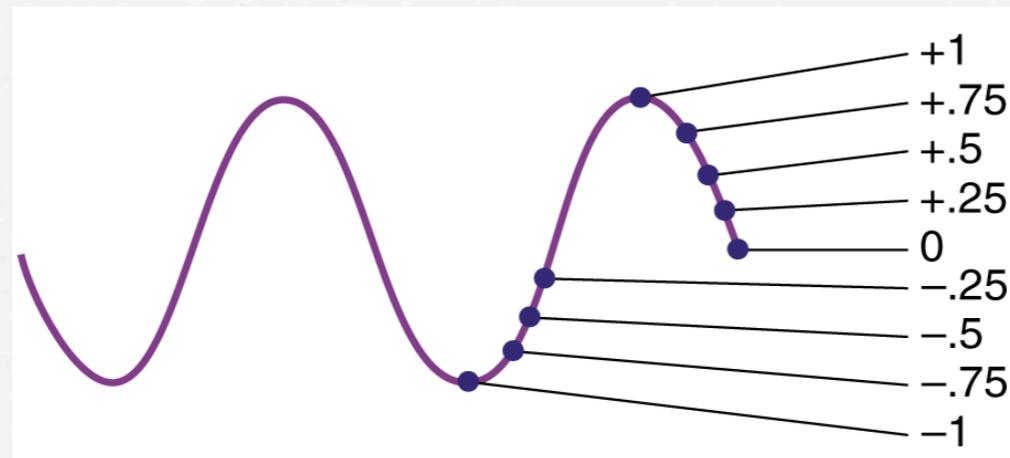


Nei sistemi **analogici**, le quantità vengono rappresentate in maniera **continua**.

Nei sistemi **digitali** le quantità vengono rappresentate in maniera **discreta**.

# Digitale vs Analogico

I **segnali analogici** sono molto sensibili alle interferenze (rumore)



I **segnali digitali** possono assumere solo due stati

- Per un dispositivo è semplice distinguere questi due stati, per cui vi è una maggiore immunità alle interferenze



# Digitale

Una buona foto in bianco e nero presa da un giornale avrà circa 256 sfumature di grigio.

## Rappresentazione analogica

256 gradi di luminosità con una lampada

Attenzione alle interferenze provocate dalla nebbia!

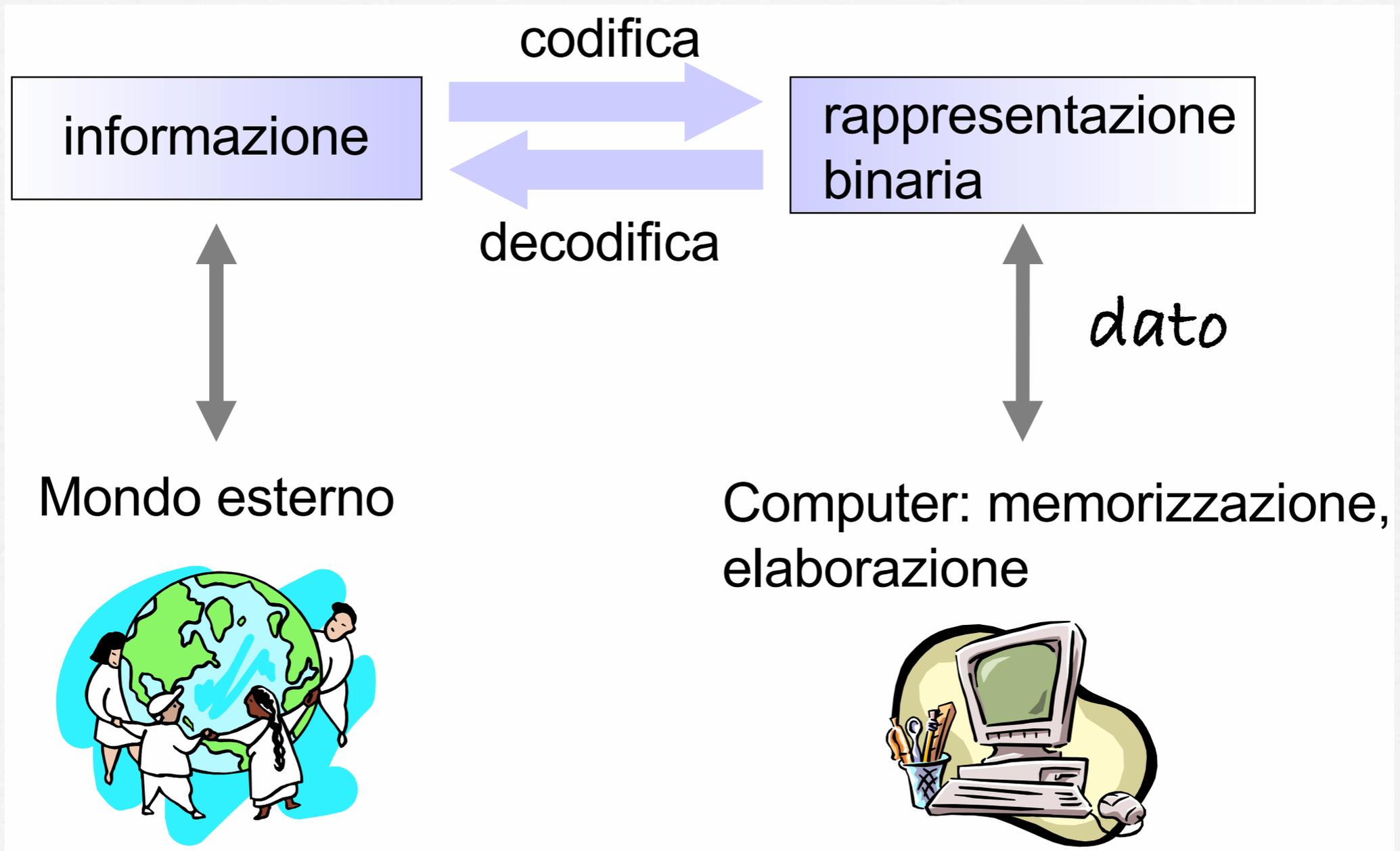
## Rappresentazione digitale

8 lampade (bit; 256 configurazioni diverse)

Ciascuna configurazione sarebbe più **sicura** anche in caso di nebbia!

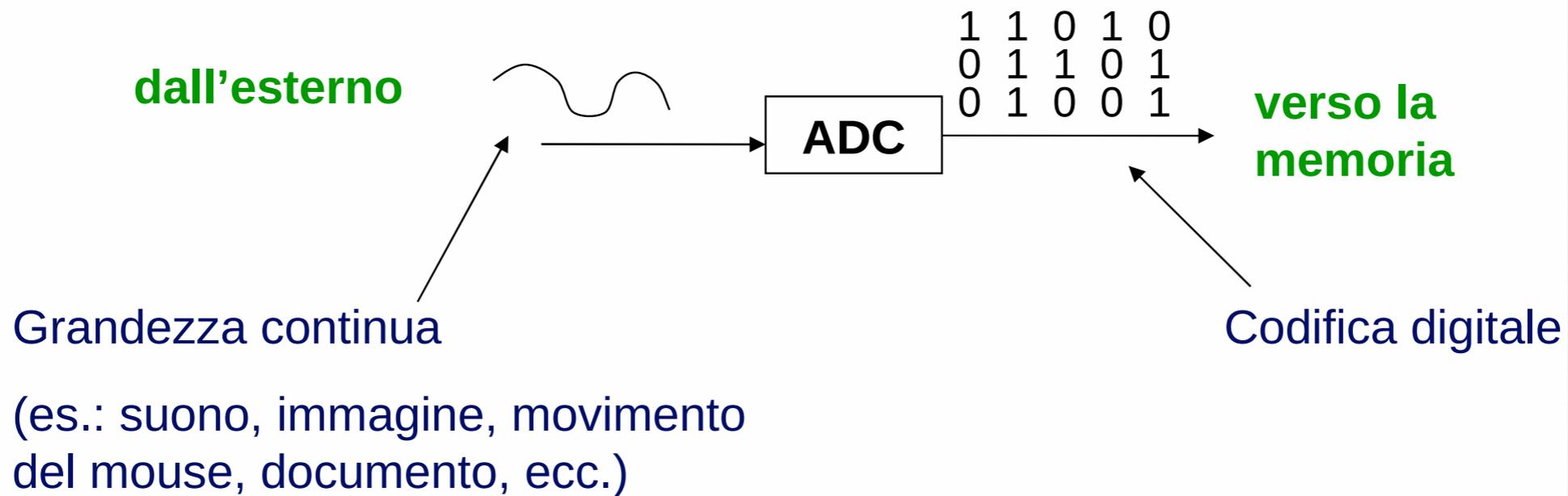


# Rappresentazione binaria



# Analogico → Digitale

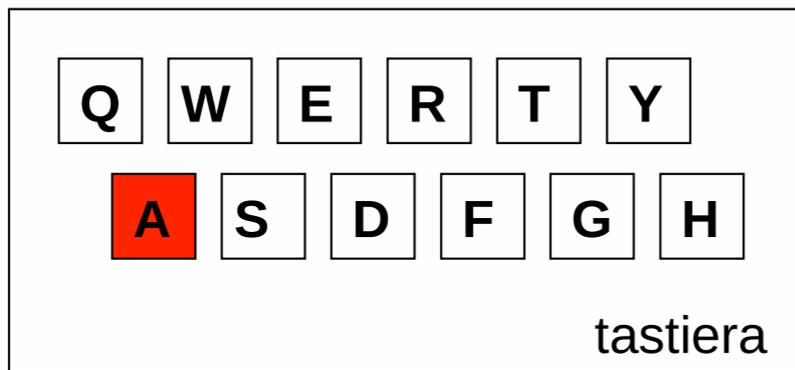
Compito delle periferiche di ingresso è quello di codificare una grandezza continua in ingresso tramite una rappresentazione digitale utilizzabile dal calcolatore.



**ADC:**  
Analog to Digital  
Conversion

# Analogico → Digitale

Il tipo di informazione rappresentata dipende dalla periferica impiegata



10000001

ne Algorithm  
Stochastic

Eugen



bit map

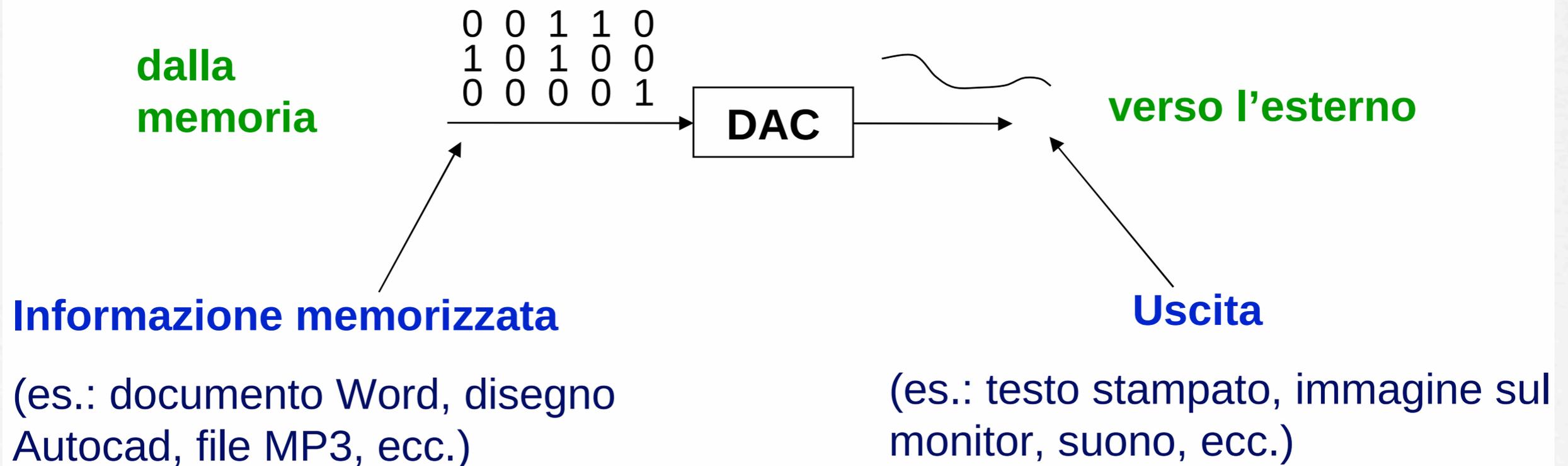
0	0	1	0	0
0	1	0	1	0
0	1	0	1	0
1	1	1	1	1
1	0	0	0	1
1	0	0	0	1

*scrimination is a general methodolog  
trary numbers of very weak compone  
very complex and accurate classifik  
neralize to new data. In fact, it is offer  
represente are added, even after see*



# Digitale → Analogico

Compito delle periferiche di uscita è quello di creare, delle informazioni codificate in digitale nella memoria del calcolatore, una rappresentazione direttamente comprensibile dall'utente umano.



# I Sistemi di Numerazione

# Generalità

Per determinare un sistema di numerazione occorre:

- un insieme limitato di simboli (le **cifre**), che rappresentano quantità prestabilite (0,1,2,V,X,M)
- le **regole** per costruire i numeri
  - sistemi di numerazione **posizionali**
  - sistemi numerici **non posizionali**

# Sistemi di numerazione

## Sistemi numerici **non posizionali**

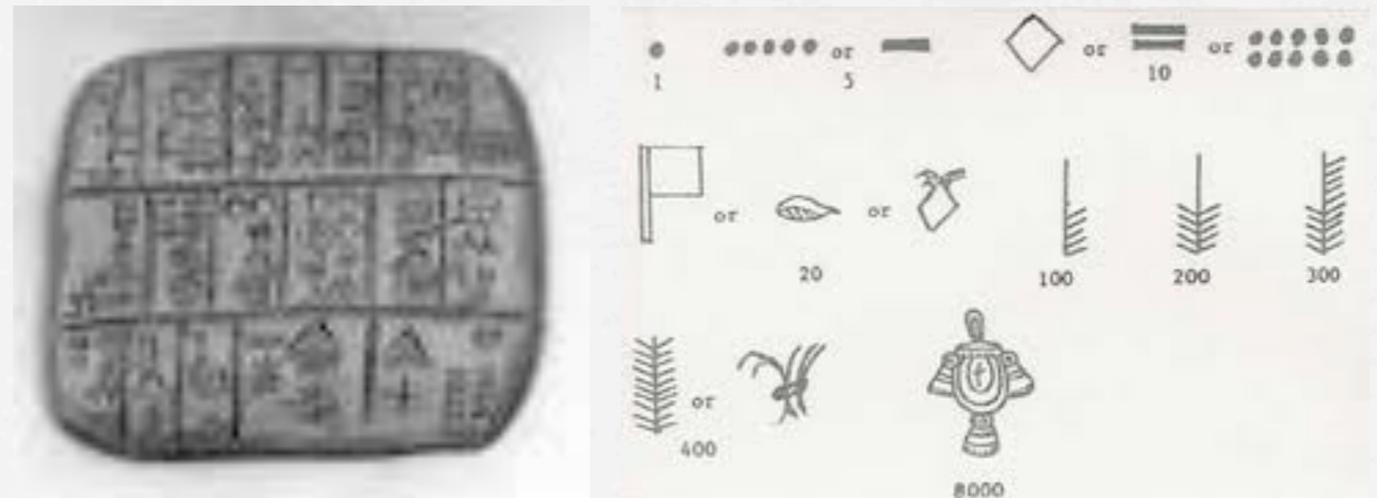
Il valore delle cifre è **indipendente** dalla posizione

- Es. Numeri romani

## Sistemi numerici **posizionali**

Il valore delle cifre **dipende** dalla loro posizione all'interno del numero

- Ogni posizione ha un **peso**



sistemi di numerazione sumero e azteco

# Sistemi posizionali

## Esempio

$$N = c_3c_2c_1c_0$$

$$V(N) = c_3 \times p_3 + c_2 \times p_2 + c_1 \times p_1 + c_0 \times p_0$$

$N$  = Rappresentazione del numero

$V(N)$  = Valore del numero

## Sistemi a base fissa

$$p_i = r^i$$

$r$  è la base del sistema

# Il Sistema Decimale

È un sistema numerico posizionale a base fissa.

Il sistema decimale utilizza:

- $r = 10$  (*base*)
- $c = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$  (*cifre*)

# Sistema Decimale ESEMPIO

Cifra più  
significativa

Cifra meno  
significativa

8427

=

$$8 \times 10^3 + 4 \times 10^2 + 2 \times 10^1 + 7 \times 10^0$$

# Sistema Binario

Anche il sistema binario è un sistema numerico posizionale a base fissa.

Il sistema binario utilizza:

- $r = 2$  (*base*)
- $c = \{0, 1\}$  (*cifre*)

Ogni cifra è detta **bit** (da **BI**nary digi**T**)

# Sistema Binario

## ESEMPIO

Bit più  
significativo (MSB)

Bit meno  
significativo (LSB)

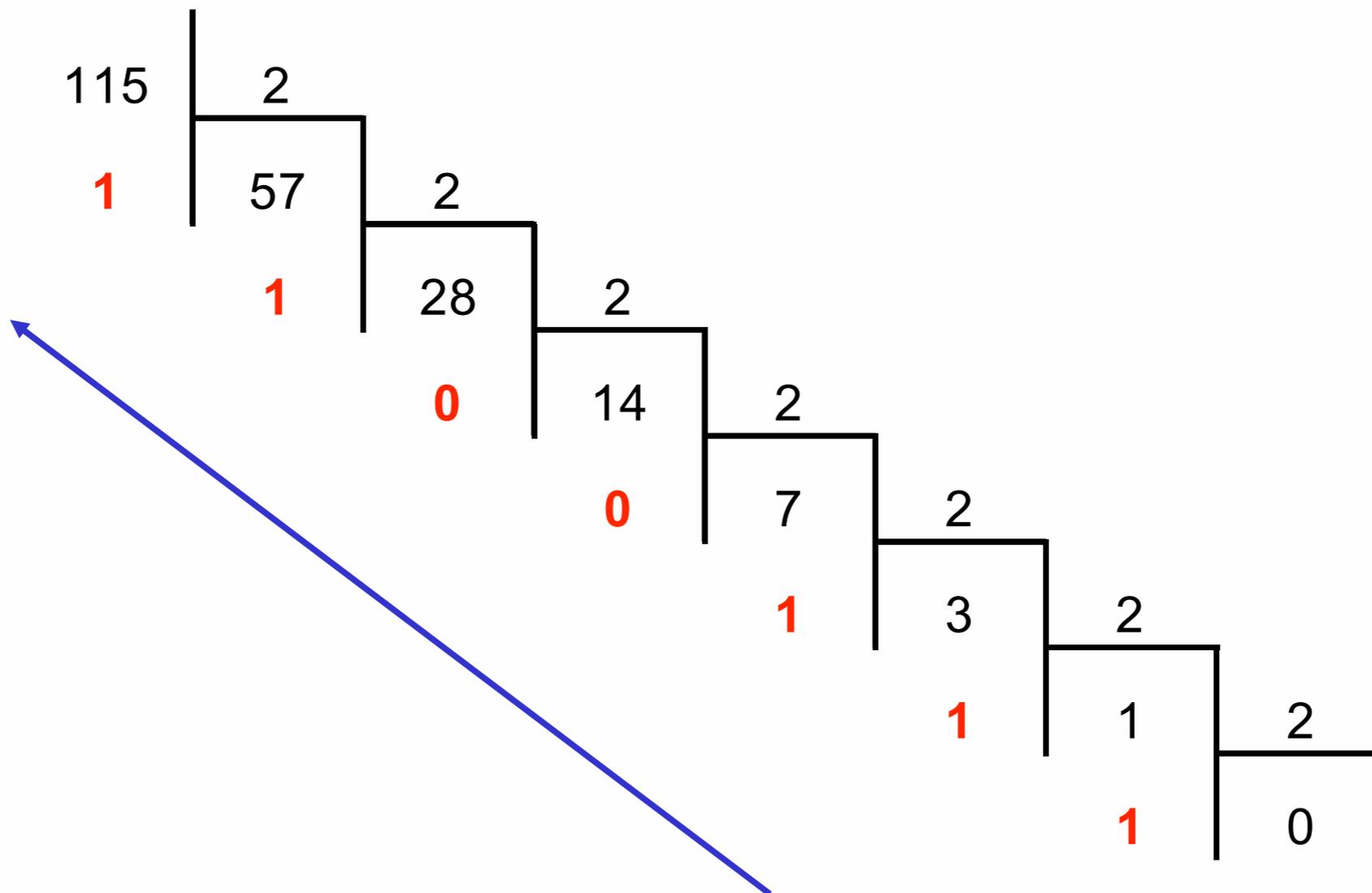
**1011**<sub>2</sub>

=

$$1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 = 11_{10}$$

# Da Decimale a Binario

Es.:  $115_{10} = 1110011_2$



# Altre basi di numerazione

## Sistema **ottale**

- $r = 8$  (*base*)
- $c = \{0, 1, 2, 3, 4, 5, 6, 7\}$  (*cifre*)

## Sistema **esadecimale**

- $r = 16$  (*base*)
- $c = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F\}$  (*cifre*)

# Base 2, 8 e 16

Esiste una corrispondenza diretta tra cifre ottali, esadecimali e il corrispondente binario.

**Ottale:** 8 cifre

- 3 bit per rappresentare una cifra ottale

$$\begin{array}{c} 11000110_2 \\ \hline 3 \quad 0 \quad 6 \end{array}$$

**Esadecimale:** 16 cifre

- 4 bit per rappresentare una cifra esadecimale

$$\begin{array}{c} 11000110_2 \\ \hline C \quad 6 \end{array}$$

$$198_{10} = 11000110_2 = 306_8 = C6_{16}$$

# Bit necessari

## Le macchine hanno vincoli spaziali

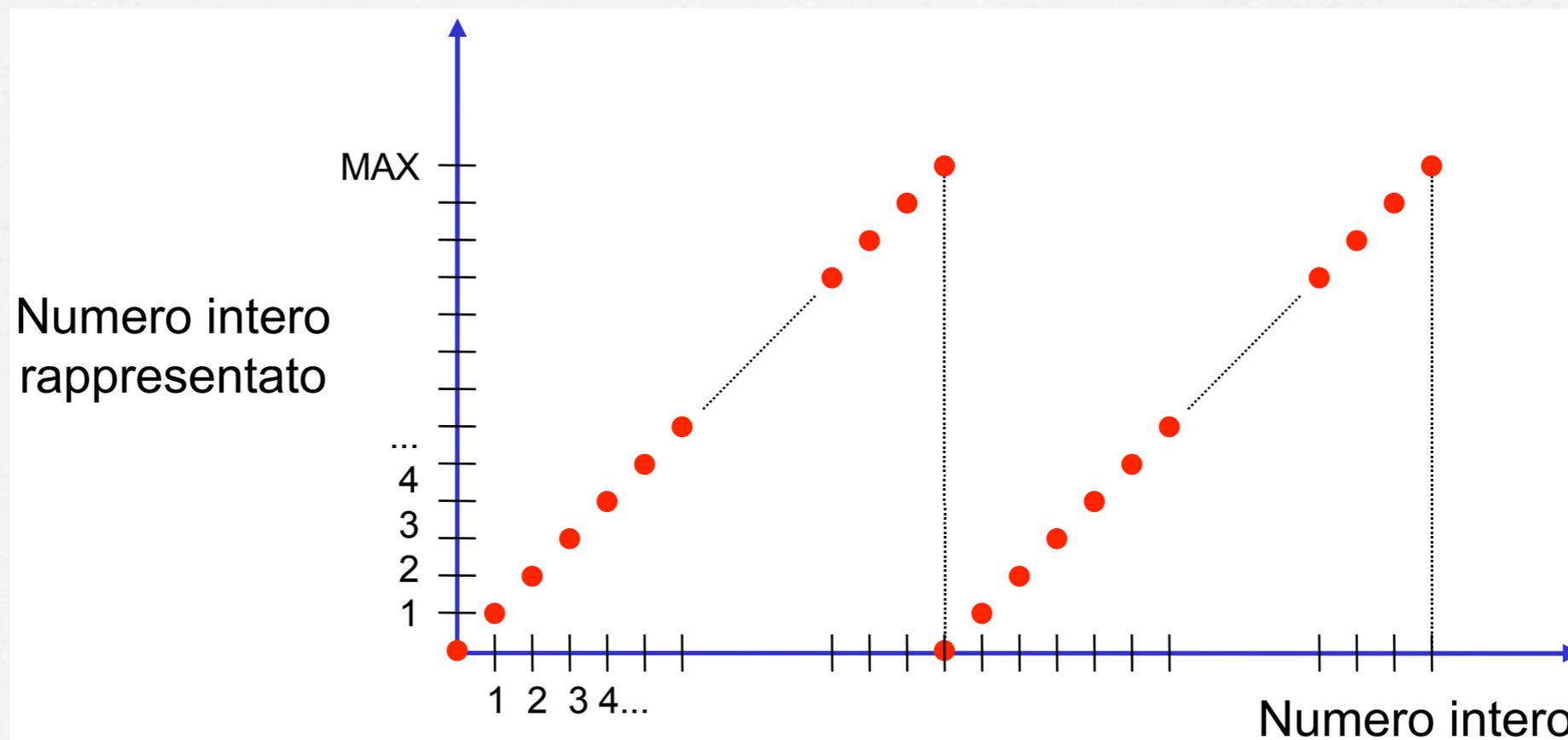
- E' necessario conoscere il massimo valore rappresentabile
  - Con  $n$  bit si può rappresentare al massimo il numero  $2^n - 1$
- E' facile determinare che per poter rappresentare al massimo il valore  $X$ , sono necessari un numero  $n$  di bit pari a:

$$\lceil \log_2 X \rceil$$

dove  $\lceil y \rceil$  restituisce il più piccolo numero intero maggiore o uguale a  $y$ .

# Overflow

Esiste un limite al numero di bit impiegati per rappresentare un numero. Dato che la rappresentazione è formata da un numero finito di bit, se si supera tale limite si ha errore (**overflow**).



# La Codifica

# Informazioni

## Numeri

- *Naturali*
- *Relativi*
- *Reali*

## Testi



Informazioni  
tradizionali

## Immagini

- *Bitmap*
- *Vettoriali*

## Audio

## Video



Informazioni  
multimediali

# Bit, Byte e Word

- L'unità atomica è il **bit** (**B**inary **D**igi**T**)
- L'insieme di 8 bit è detto **byte**
- **Word**
  - ✓ Tipicamente 16, 32 o 64 bit
  - ✓ Insieme di bit la cui dimensione è una importante caratteristica del calcolatore considerato. Essa influenza:
    - La larghezza degli indirizzi
    - La dimensione dei registri del processore
    - Larghezza dei bus (word o multipli di essa)

# Il problema della codifica

- Un calcolatore può trattare **diversi tipi di dati**: numeri (interi, reali), testo, immagini, suoni, etc. che vanno comunque memorizzati in registri di memoria.
- È quindi necessario adottare una **codifica** del tipo di dato considerato: occorre, cioè, mettere in corrispondenza biunivoca i valori del tipo con gli stati che può assumere il registro.

Registro da un byte  $\Rightarrow 2^8 = 256$  stati possibili. Che cosa è possibile codificare ?

<p><b>Numeri naturali [0,255]</b></p> <p>0    <math>\leftrightarrow</math> 00000000          1    <math>\leftrightarrow</math> 00000001          ....          255 <math>\leftrightarrow</math> 11111111</p>	<p><b>Numeri interi [-128,127]</b></p> <p>-128 <math>\leftrightarrow</math> 00000000          -127 <math>\leftrightarrow</math> 00000001          0    <math>\leftrightarrow</math> 10000000          +127 <math>\leftrightarrow</math> 11111111</p>	<p><b>Numeri reali [0,1[</b></p> <p>0.0000    <math>\leftrightarrow</math> 00000000          0.0039    <math>\leftrightarrow</math> 00000001          0.0078    <math>\leftrightarrow</math> 00000010          ....          0.9961    <math>\leftrightarrow</math> 11111111</p>	<p><b>Caratteri</b></p> <p>A    <math>\leftrightarrow</math> 01000001          a    <math>\leftrightarrow</math> 01100001            0    <math>\leftrightarrow</math> 00110000          1    <math>\leftrightarrow</math> 00110001</p>
--	--	--	---

**La codifica implica una rappresentazione dei dati limitata e discreta**

# Codifica binaria

- Esiste una particolare aggregazione di bit che è costituita da **8 bit** ( $2^8 = 256$  informazioni) e prende il nome di **byte (B)**
- Di solito si usano i multipli del byte

Kilo	KB	$2^{10}$ (~ un migliaio, 1024 byte)
Mega	MB	$2^{20}$ (~ un milione, 1KB x 1024)
Giga	GB	$2^{30}$ (~ un miliardo, 1MB x 1024)
Tera	TB	$2^{40}$ (~ mille miliardi, 1GB x 1024)
Peta	PB	$2^{50}$ (~ miliardo miliardi, 1TB x 1024)

# Codifica binaria

- ▶ Oltre ai dati, è necessario memorizzare anche le **istruzioni**, cioè le singole azioni elementari che l'unità centrale può eseguire.
- ▶ Nello specificare un'istruzione, bisogna precisare l'**operazione** da compiere e i **dati** coinvolti nell'operazione.

*Esempio: somma 3 e 4*  
*operazione dati*

- ▶ Come rappresentare le operazioni? L'insieme delle diverse operazioni che l'unità centrale è in grado di eseguire è **finito** e quindi è possibile codificarlo con un certo numero di bit (**codice operativo**):

**0000 somma**  
**0001 sottrai**  
**0010 moltiplica**  
**0011 dividi**  
...

# Codifica binaria

Una istruzione sarà quindi rappresentabile da una sequenza di bit divisa in due parti:

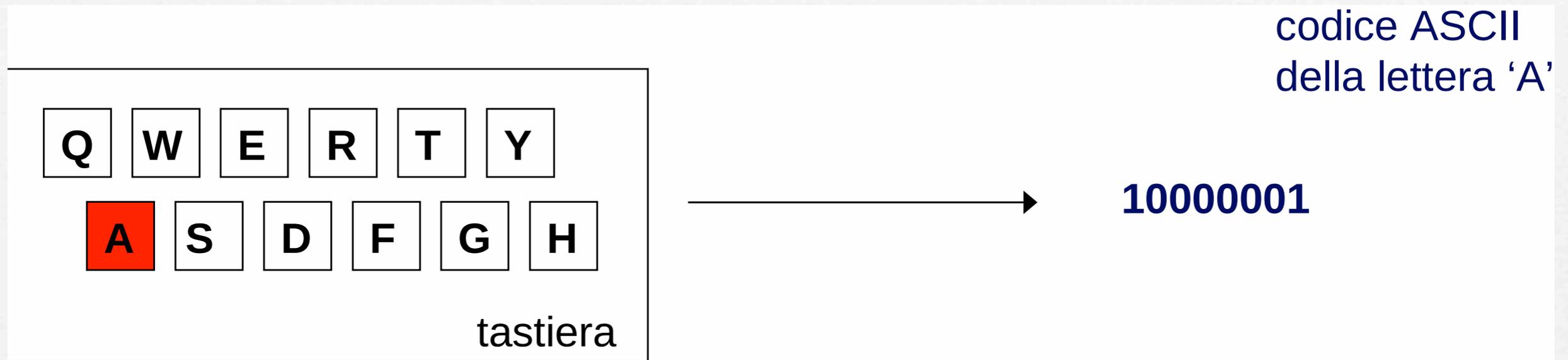
- un **codice operativo**
- un campo **operandi** (1, 2 o più operandi)



# Codifica dei testi

- ➔ Si utilizza una tabella (arbitraria)
- ➔ Standard oggi (quasi) universalmente riconosciuto il codice ASCII (**A**merican **S**tandard **C**ode for **I**nformation **I**nterchange)
  - ✓ Nella versione base ogni carattere (simbolo) è codificato con 7 bit
    - 128 simboli diversi
  - ✓ Versione estesa di 8 bit
    - 256 simboli diversi

# La codifica ASCII



La Codifica ASCII serve a codificare i caratteri alfanumerici.

Il Formato RTF (Rich Text Format) memorizza alcune caratteristiche aggiuntive dei caratteri.

# Tabella codici ASCII base

Dec	Hx	Oct	Char	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr
0	0	000	<b>NUL</b> (null)	32	20	040	&#32;	Space	64	40	100	&#64;	@	96	60	140	&#96;	`
1	1	001	<b>SOH</b> (start of heading)	33	21	041	&#33;	!	65	41	101	&#65;	A	97	61	141	&#97;	a
2	2	002	<b>STX</b> (start of text)	34	22	042	&#34;	"	66	42	102	&#66;	B	98	62	142	&#98;	b
3	3	003	<b>ETX</b> (end of text)	35	23	043	&#35;	#	67	43	103	&#67;	C	99	63	143	&#99;	c
4	4	004	<b>EOT</b> (end of transmission)	36	24	044	&#36;	\$	68	44	104	&#68;	D	100	64	144	&#100;	d
5	5	005	<b>ENQ</b> (enquiry)	37	25	045	&#37;	%	69	45	105	&#69;	E	101	65	145	&#101;	e
6	6	006	<b>ACK</b> (acknowledge)	38	26	046	&#38;	&	70	46	106	&#70;	F	102	66	146	&#102;	f
7	7	007	<b>BEL</b> (bell)	39	27	047	&#39;	'	71	47	107	&#71;	G	103	67	147	&#103;	g
8	8	010	<b>BS</b> (backspace)	40	28	050	&#40;	(	72	48	110	&#72;	H	104	68	150	&#104;	h
9	9	011	<b>TAB</b> (horizontal tab)	41	29	051	&#41;	)	73	49	111	&#73;	I	105	69	151	&#105;	i
10	A	012	<b>LF</b> (NL line feed, new line)	42	2A	052	&#42;	*	74	4A	112	&#74;	J	106	6A	152	&#106;	j
11	B	013	<b>VT</b> (vertical tab)	43	2B	053	&#43;	+	75	4B	113	&#75;	K	107	6B	153	&#107;	k
12	C	014	<b>FF</b> (NP form feed, new page)	44	2C	054	&#44;	,	76	4C	114	&#76;	L	108	6C	154	&#108;	l
13	D	015	<b>CR</b> (carriage return)	45	2D	055	&#45;	-	77	4D	115	&#77;	M	109	6D	155	&#109;	m
14	E	016	<b>SO</b> (shift out)	46	2E	056	&#46;	.	78	4E	116	&#78;	N	110	6E	156	&#110;	n
15	F	017	<b>SI</b> (shift in)	47	2F	057	&#47;	/	79	4F	117	&#79;	O	111	6F	157	&#111;	o
16	10	020	<b>DLE</b> (data link escape)	48	30	060	&#48;	0	80	50	120	&#80;	P	112	70	160	&#112;	p
17	11	021	<b>DC1</b> (device control 1)	49	31	061	&#49;	1	81	51	121	&#81;	Q	113	71	161	&#113;	q
18	12	022	<b>DC2</b> (device control 2)	50	32	062	&#50;	2	82	52	122	&#82;	R	114	72	162	&#114;	r
19	13	023	<b>DC3</b> (device control 3)	51	33	063	&#51;	3	83	53	123	&#83;	S	115	73	163	&#115;	s
20	14	024	<b>DC4</b> (device control 4)	52	34	064	&#52;	4	84	54	124	&#84;	T	116	74	164	&#116;	t
21	15	025	<b>NAK</b> (negative acknowledge)	53	35	065	&#53;	5	85	55	125	&#85;	U	117	75	165	&#117;	u
22	16	026	<b>SYN</b> (synchronous idle)	54	36	066	&#54;	6	86	56	126	&#86;	V	118	76	166	&#118;	v
23	17	027	<b>ETB</b> (end of trans. block)	55	37	067	&#55;	7	87	57	127	&#87;	W	119	77	167	&#119;	w
24	18	030	<b>CAN</b> (cancel)	56	38	070	&#56;	8	88	58	130	&#88;	X	120	78	170	&#120;	x
25	19	031	<b>EM</b> (end of medium)	57	39	071	&#57;	9	89	59	131	&#89;	Y	121	79	171	&#121;	y
26	1A	032	<b>SUB</b> (substitute)	58	3A	072	&#58;	:	90	5A	132	&#90;	Z	122	7A	172	&#122;	z
27	1B	033	<b>ESC</b> (escape)	59	3B	073	&#59;	;	91	5B	133	&#91;	[	123	7B	173	&#123;	{
28	1C	034	<b>FS</b> (file separator)	60	3C	074	&#60;	<	92	5C	134	&#92;	\	124	7C	174	&#124;	
29	1D	035	<b>GS</b> (group separator)	61	3D	075	&#61;	=	93	5D	135	&#93;	]	125	7D	175	&#125;	}
30	1E	036	<b>RS</b> (record separator)	62	3E	076	&#62;	>	94	5E	136	&#94;	^	126	7E	176	&#126;	~
31	1F	037	<b>US</b> (unit separator)	63	3F	077	&#63;	?	95	5F	137	&#95;	_	127	7F	177	&#127;	DEL

www.lookuptables.com



# Codice ASCII esteso

0		32		64	@	96	`	128	Ç	160	á	192	Ł	224	Ó
1	☺	33	!	65	A	97	a	129	ü	161	í	193	ł	225	ó
2	☹	34	"	66	B	98	b	130	é	162	ó	194	Ł	226	Ô
3	♥	35	#	67	C	99	c	131	â	163	ú	195	ł	227	Ò
4	♦	36	\$	68	D	100	d	132	ä	164	ñ	196	—	228	ô
5	♣	37	%	69	E	101	e	133	à	165	Ñ	197	+	229	Õ
6	♠	38	&	70	F	102	f	134	á	166	ª	198	â	230	µ
7	·	39	'	71	G	103	g	135	ç	167	º	199	Ã	231	þ
8	▣	40	(	72	H	104	h	136	ê	168	¿	200	Ł	232	ƒ
9	○	41	)	73	I	105	i	137	ë	169	©	201	ł	233	Ú
10	◼	42	*	74	J	106	j	138	è	170	¬	202	ł	234	Û
11	♂	43	+	75	K	107	k	139	ï	171	½	203	ł	235	Ü
12	♀	44	,	76	L	108	l	140	î	172	¼	204	ł	236	ý
13	♪	45	-	77	M	109	m	141	í	173	¡	205	=	237	Ý
14	♫	46	.	78	N	110	n	142	Ä	174	«	206	≠	238	—
15	☼	47	/	79	O	111	o	143	Å	175	»	207	◻	239	´
16	▶	48	0	80	P	112	p	144	É	176	◻	208	◻	240	-
17	◀	49	1	81	Q	113	q	145	æ	177	◻	209	Đ	241	±
18	↑	50	2	82	R	114	r	146	Æ	178	◻	210	Ê	242	—
19	!!	51	3	83	S	115	s	147	ø	179		211	Ë	243	¾
20	¶	52	4	84	T	116	t	148	ö	180	└	212	È	244	¶
21	§	53	5	85	U	117	u	149	ò	181	Á	213	ı	245	§
22	—	54	6	86	V	118	v	150	û	182	Â	214	í	246	÷
23	↑	55	7	87	W	119	w	151	ù	183	Ã	215	î	247	,
24	↑	56	8	88	X	120	x	152	ÿ	184	©	216	ï	248	°
25	↓	57	9	89	Y	121	y	153	ÿ	185	¶	217	┘	249	¨
26	→	58	:	90	Z	122	z	154	Ü	186		218	┘	250	·
27	←	59	;	91	[	123	{	155	ø	187	¶	219	◼	251	¹
28	└	60	<	92	\	124		156	£	188	┘	220	◼	252	º
29	↔	61	=	93	]	125	}	157	Ø	189	φ	221	ı	253	²
30	▲	62	>	94	^	126	~	158	×	190	¥	222	ı	254	▣
31	▼	63	?	95	_	127	△	159	f	191	┘	223	◼	255	



# Codice EBCDIC

In informatica, la sigla **EBCDIC** (dall'inglese Extended Binary Coded Decimal Interchange Code) indica un sistema di codifica dell'informazione a 8 bit usato in numerosi sistemi operativi di produzione IBM, sia per elaboratori di classe mainframe che per minicomputer.

Viene inoltre utilizzato da varie piattaforme di altri produttori.

Deriva dalla codifica a 6 bit binary-coded decimal, utilizzata nelle schede perforate e nella maggior parte delle periferiche IBM della fine degli anni 1950 e dell'inizio degli anni 1960.

	-0	-1	-2	-3	-4	-5	-6	-7	-8	-9	-A	-B	-C	-D	-E	-F
0-	NUL 00	SOH 01	STX 02	ETX 03	SEL 04	HT 09	RNL 0A	DEL 0B	GE 0C	SPS 0D	RPT 0E	VT 0F	FF 10	CR 11	SO 12	SI 13
1-	DLE 10	DC1 11	DC2 12	DC3 13	RES 14	NL 15	BS 16	POC 17	CAN 18	EM 19	UBS 1A	CU1 1B	IFS 1C	IGS 1D	IRS 1E	IUS 1F
2-	DS 20	SOS 21	FS 22	WUS 23	BYP 24	LF 25	ETB 26	ESC 27	SA 28	SFE 29	SM 2A	SW 2B	CSP 2C	MFA 2D	ENQ 2E	ACK 2F
3-			SYN 30	IR 31	PP 32	TRN 33	NBS 34	EOT 35	SBS 36	IT 37	RFF 38	CU3 39	DC4 3A	NAK 3B		SUB 3C
4-	SP 40	RSP 41	à 42	ã 43	ä 44	á 45	â 46	ç 47	ñ 48	[ 49	. 4A	< 4B	( 4C	+ 4D	! 4E	
5-	& 50	é 51	ê 52	ë 53	è 54	í 55	î 56	ï 57	ì 58	β 59	] 5A	\$ 5B	* 5C	) 5D	; 5E	^ 5F
6-	- 60	/ 61	À 62	Ã 63	Ä 64	Á 65	Ã 66	Ç 67	Ñ 68	! 69	, 6A	% 6B	_ 6C	> 6D	? 6E	
7-	ø 70	É 71	Ê 72	Ë 73	È 74	Ì 75	Î 76	Ï 77	 78	· 79	: 7A	# 7B	@ 7C	' 7D	= 7E	" 7F
8-	Ø 80	a 81	b 82	c 83	d 84	e 85	f 86	g 87	h 88	i 89	« 8A	» 8B	ò 8C	ó 8D	þ 8E	± 8F
9-	° 90	j 91	k 92	l 93	m 94	n 95	o 96	p 97	q 98	r 99	ª 9A	º 9B	æ 9C	, 9D	Æ 9E	¤ 9F
A-	µ A0	~ A1	s A2	t A3	u A4	v A5	w A6	x A7	y A8	z A9	i AA	¿ AB	Ð AC	Ý AD	Þ AE	@ AF
B-	€ B0	£ B1	¥ B2	· B3	© B4	§ B5	¶ B6	¼ B7	½ B8	¾ B9	¬ BA	 BB	- BC	- BD	' BE	x BF
C-	{ C0	A C1	B C2	C C3	D C4	E C5	F C6	G C7	H C8	I C9	SHY CA	ó CB	ô CC	õ CD	ö CE	õ CF
D-	} D0	J D1	K D2	L D3	M D4	N D5	O D6	P D7	Q D8	R D9	¹ DA	û DB	ü DC	ù DD	ú DE	ÿ DF
E-	\ E0	+ E1	S E2	T E3	U E4	V E5	W E6	X E7	Y E8	Z E9	² EA	Û EB	Ö EC	Ò ED	Ó EE	Õ EF
F-	0 F0	1 F1	2 F2	3 F3	4 F4	5 F5	6 F6	7 F7	8 F8	9 F9	³ FA	Û FB	Ü FC	Ù FD	Ú FE	EO FF



# UNICODE

Un altro codice molto diffuso oggi è lo UNICODE, che impiega 16 bit per carattere (Extended ASCII + caratteri etnici):

$2^{16} = 65.536$  simboli diversi

Derivazioni: UTF-8,16,32 per il web

## Da Wikipedia

Unicode era stato originariamente pensato come una codifica a 16 bit (quattro cifre esadecimali) che dava la possibilità di codificare 65.536 caratteri. Tanto si riteneva essere sufficiente per rappresentare i caratteri impiegati in tutte le lingue scritte del mondo.

Ora invece lo standard Unicode, che tendenzialmente è perfettamente allineato con la norma [ISO/IEC 10646](#), prevede una codifica fino 21 bit e supporta un repertorio di codici numerici che possono rappresentare circa un milione di caratteri. Ciò appare sufficiente a coprire anche i fabbisogni di codifica di scritti del patrimonio storico dell'umanità, nelle diverse lingue e negli svariati sistemi di segni utilizzati.