



Concetti Introduttivi

**Rappresentazione delle
informazioni**

Rappresentazione delle informazioni

- Per rappresentare i dati di ingresso o di uscita e la sequenza di operazioni che costituiscono un algoritmo, è necessario definire un linguaggio comprensibile da parte di un elaboratore elettronico.
- Poiché tutte le informazioni o istruzioni immesse in un elaboratore sono rappresentate sotto forma di segnali elettrici, si è introdotta la codifica binaria basata sull'utilizzo del **BIT**.
- Un **bit** rappresenta l'elemento minimo d'informazione che può essere trattata da un calcolatore elettronico, così come un carattere è l'elemento minimo d'informazione che può essere trattato nel testo scritto. Tuttavia, a differenza di un carattere, che può assumere moltissimi valori (alfanumerici), un bit può assumere soltanto due valori, convenzionalmente rappresentati dai simboli 0 o 1 (o vero e falso).

0



1



Codifica Binaria (1)

- Gli elaboratori utilizzano la codifica binaria (cioè con 0 e 1) dell'informazione
- Perché solo due simboli?
 - differenti tensioni elettriche, polarità magnetiche, ...
 - Riduce errori (ad es. causati da rumore nei segnali)
- Unità elementare di informazione:
bit interpretato come 0 o come 1
- Unità derivata:
byte = 8 bit

Codifica Binaria (2)

Combinando fra loro sequenze di BIT ed usando una opportuna codifica binaria, è possibile costruire *parole* di diversa lunghezza con cui rappresentare le istruzioni.

Esempio: 13 può essere espresso in funzione delle potenze di 2 come:

$$13 = 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0$$

$$13 = 1 \times 8 + 1 \times 4 + 0 \times 2 + 1 \times 1$$

Ovvero può essere rappresentato dalla sequenza di bit

1 1 0 1

Codifica Binaria (3)

- In particolare, utilizzando N simboli binari, ognuno dei quali quindi può assumere il valore 0 o 1, si possono costruire ben 2^N differenti *parole*.

N. di bit	N. di parole generabili
2	4
4	16
8	256
16	65536
32	4294967296

- Per costruire un carattere sono necessari un certo numero di bit. Per esempio, per *codificare* i numeri da 0 a 9 sarebbero necessari almeno 4 bit

Decimale	Binario	Decimale	Binario
'0'	0000	'5'	0101
'1'	0001	'6'	0110
'2'	0010	'7'	0111
'3'	0011	'8'	1000
'4'	0100	'9'	1001

Codifica Binaria (4)

- Poiché per rappresentare tutti i possibili caratteri è necessario sempre ricorrere all'utilizzo di un certo numero minimo di BIT, per convenzione in informatica si utilizza il termine **BYTE** per indicare l'aggregamento di 8 bit consecutivi.
- Per esempio le capacità dei dispositivi di memoria vengono generalmente espresse in termini di numero di bytes che sono in grado di contenere.
- Considerate le capacità attuali delle memorie, misurarne le capacità in termini di bytes equivarrebbe a misurare la distanza fra la terra e la luna in centimetri. Si utilizzano allora dei multipli del byte (tenendo conto che $2^{10}=1024$)
 - 8 bit formato un byte (B)
 - 2 byte formano una word
 - 1024 bytes formano un Kilobyte (KB)
 - 1024 KB formano un Megabyte (MB o "mega")
 - 1024 MB formano un Gigabyte (GB, o "giga")
 - 1024 GB formano un Terabyte (TB, o "tera").

1 KB = 2^{10} bytes	= 1024 bytes	= 8192 bit
1 MB = 2^{10} KB	= 1048576 bytes	= 8388608 bit
1 GB = 2^{10} MB	= 1073741824 bytes	= 8589934592 bit
1 TB = 2^{10} GB	= 1099511627776 bytes	= 8796093022208 bit

Operazioni “binarie”

Operazioni aritmetiche

Ad esempio somma di numeri

$$5 = 1 \times 4 + 0 \times 2 + 1 \times 1$$

$$1 = 0 \times 4 + 0 \times 2 + 1 \times 1$$

$$6 = 1 \times 4 + 1 \times 2 + 0 \times 1$$

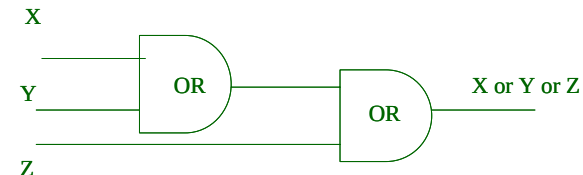
Esprimibile come operazione tra bit

Operazioni logiche: AND, OR, NOT, ...

0 = falso, 1=vero

X AND Y = 1	sse X=1 e Y=1
X OR Y = 1	sse X=1 oppure Y=1
NOT X = 1	sse X = 0

Implementate mediante circuiti logici



Codifica di caratteri

- Associando un simbolo dell'alfabeto ad ogni numero possiamo codificare tutte le lettere

a-z A-Z 0-9

usando 7 bit (cioè in un byte)!!

- Esempio: **00000101** rappresenta la lettera 'c'

Codice ASCII

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

Source: www.asciitable.com

Codifica di dati multimediali

- Lettere e numeri non costituiscono le uniche informazioni utilizzate dagli elaboratori ma si diffondono sempre di più applicazioni che usano ed elaborano anche altri tipi di informazione:
 - diagrammi
 - immagini
 - suoni
- Spesso in questi casi si parla di applicazioni di tipo **multimediale**

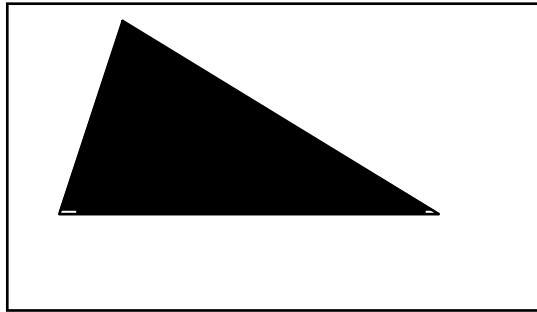


Codifica immagini (1)

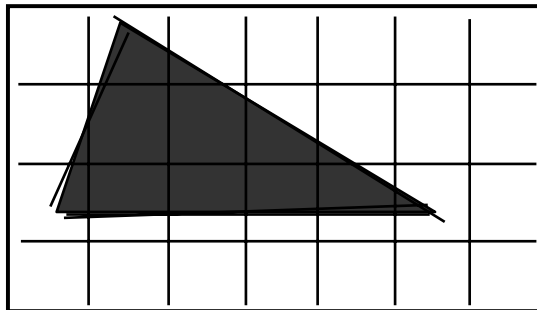
- Esistono numerose tecniche per la memorizzazione digitale e l'elaborazione di un'immagine, che comunque sono sequenze di bit.
- L'immagine viene digitalizzata cioè rappresentata con sequenze di **pixel** (picture element)
- Ogni pixel ha associato un numero che descrive un particolare colore (o tonalità di grigio)
- Inoltre si mantengono la dimensione, la risoluzione (numero di punti per pollice), e il numero di colori utilizzati

Codifica immagini (2)

Consideriamo un'immagine in bianco e nero, senza ombreggiature o livelli di chiaroscuro

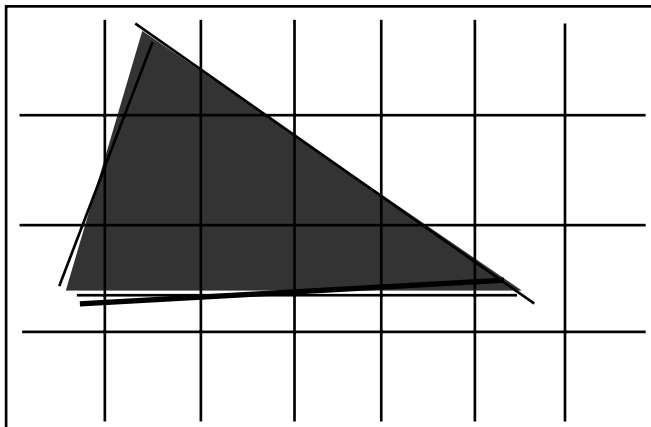


Suddividiamo l'immagine mediante una griglia formata da righe orizzontali e verticali a distanza costante



Codifica immagini (3)

- Ogni quadratino derivante da tale suddivisione prende il nome di **pixel** (picture element) e può essere codificato in binario secondo la seguente convenzione:
 - Il simbolo “0” viene utilizzato per la codifica di un pixel corrispondente ad un quadratino bianco (in cui il bianco è predominante)
 - Il simbolo “1” viene utilizzato per la codifica di un pixel corrispondente ad un quadratino nero (in cui il nero è predominante)
- Poiché una sequenza di bit è lineare, si deve definire una convenzione per ordinare la griglia dei pixel in una sequenza
 - Assumiamo per ipotesi che i pixel siano ordinati dal basso verso l'alto e da sinistra verso destra



0 ₂₂	1 ₂₃	0 ₂₄	0 ₂₅	0 ₂₆	0 ₂₇	0 ₂₈
0 ₁₅	1 ₁₆	1 ₁₇	0 ₁₈	0 ₁₉	0 ₂₀	0 ₂₁
0 ₈	1 ₉	1 ₁₀	1 ₁₁	1 ₁₂	0 ₁₃	0 ₁₄
0 ₁	0 ₂	0 ₃	0 ₄	0 ₅	0 ₆	0 ₇

La rappresentazione della figura sarà data dalla stringa binaria

0000000 0111100 0110000 0100000

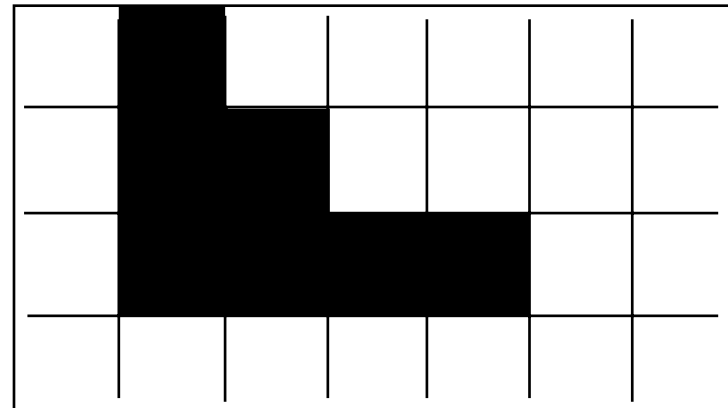
Codifica immagini (4)

Non sempre il contorno della figura coincide con le linee della griglia, per cui nella codifica si ottiene un'approssimazione della figura originaria

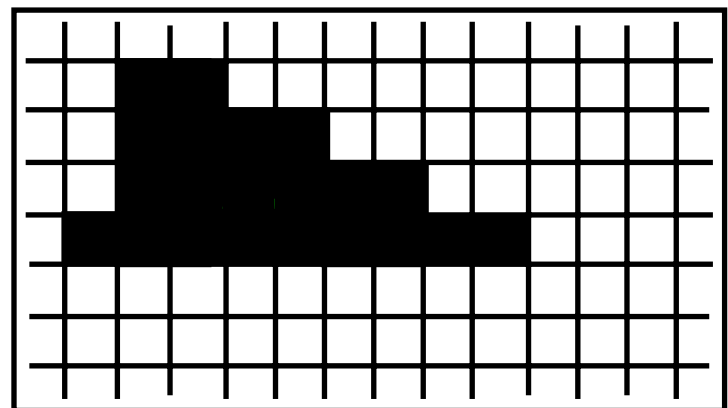
Riconvertendo la stringa

0000000011110001100000100000

in immagine si ottiene



La rappresentazione sarà più fedele all'aumentare del numero di pixel



Codifica immagini in bianco e nero

- Le immagini in bianco e nero hanno delle sfumature (diversi livelli di intensità di grigio), ma assegnando un bit ad ogni pixel è possibile codificare solo immagini senza livelli di chiaroscuro
- Per codificare le immagini con diversi livelli di grigio si usa la stessa tecnica: per ogni pixel si stabilisce il livello medio di grigio cui viene assegnata convenzionalmente una rappresentazione binaria
- Per memorizzare un pixel non è più sufficiente un solo bit.
 - ad esempio, con 4 bit possiamo rappresentare $2^4=16$ livelli di grigio, mentre con 8 bit ne possiamo distinguere $2^8=256$, ecc.
- La rappresentazione di un'immagine mediante la codifica dei pixel, viene chiamata **codifica bitmap**

Codifica immagini a colori

- Analogamente possiamo codificare le immagini a colori. In questo caso si tratta di individuare un certo numero di sfumature di colore differenti e di codificare ogni sfumatura mediante un'opportuna sequenza di bit
- Il numero di byte richiesti dipende dalla risoluzione e dal numero di colori che ogni pixel può assumere
 - i monitor utilizzano risoluzioni di 640×480 , 1024×768 , oppure 1280×1024 ed un numero di colori per pixel che va da 256 fino a 16 milioni
- Per distinguere 256 colori sono necessari 8 bit per la codifica di ciascun pixel:
 - la codifica di un'immagine formata da 640×480 pixel richiederà 2457600 bit (307200 byte)

Formato immagini (1)

- Gli aspetti fondamentali di una immagine digitalizzata sono:
 - **risoluzione**: dimensioni e qualità dell'immagine. Comprende due aspetti:
 - **quantità** di punti che compongono l'immagine, misurata in pixel (es. $640 \times 480 = 640$ pixel orizzontali $\times$ 480 pixel verticali)
 - **densità** di punti nella visualizzazione o stampa e si misura in dpi (dot per inch) ad es. 72dpi, 300dpi, ecc.
 - **profondità** di colore ovvero numero massimo di colori possibili; si misura in bit (es. 8 bit = 256 colori, 16 bit = 65mila colori, ecc).
 - **formato/compressione** ovvero il tipo di salvataggio o codifica del file. Alcuni formati permettono di comprimere i dati, ottenendo una riduzione della dimensione del file. Da tenere presente però che la compressione in alcuni casi può influire sulla qualità dell'immagine stessa.

Formato immagini (2)

Formato	Compressione	Profondità di colore	Applicazione Ottimale	Note
GIF Graphics Interchange Format	SI Automatica	256 colori	Disegni, icone, schermate computer	supporta sfondo trasparente ed animazioni
JPG, JPEG Joint Photograph Expert Group	SI Regolabile	milioni di colori	Foto	Ottima qualità visiva ma file di dimensioni enormi
PNG Portable Network Graphics	SI Automatica	milioni di colori	Foto	Supporta sfondo trasparente

Codifica filmati

- Sono immagini in movimento
- Memorizzazione mediante sequenze di fotogrammi (sono necessarie delle tecniche per ottimizzare tale memorizzazione)
- Sequenze di immagini compresse:
 - ad esempio si possono registrare solo le variazioni tra un fotogramma e l'altro
- Esistono vari formati (compresi i suoni):
 - *mpeg* (il piu' usato)
 - *avi* (microsoft)
 - *quicktime* (apple)
 - *mov*
- E' possibile ritoccare i singoli fotogrammi con appositi programmi

Codifica suoni

- L'onda sonora viene misurata (campionata) ad intervalli regolari
- Minore è l'intervallo di campionamento e maggiore è la qualità del suono
- CD musicali: 44000 campionamenti al secondo, 16 bit per campione
- Alcuni formati:
 - .mov, .wav, .mpeg, .avi,
 - formato **midi** usato per l'elaborazione della musica al PC