

Multipli del byte

Prefissi SI

Prefissi binari

Nome	Simbolo	Multiplo	Nome	Simbolo	Multiplo
chilobyte	kB	10^3	kibibyte	KiB	2^{10}
megabyte	MB	10^6	mebibyte	MiB	2^{20}
gigabyte	GB	10^9	gibibyte	GiB	2^{30}
terabyte	TB	10^{12}	tebibyte	TiB	2^{40}
petabyte	PB	10^{15}	pebibyte	PiB	2^{50}
exabyte	EB	10^{18}	exbibyte	EiB	2^{60}
zettabyte	ZB	10^{21}	zebibyte	ZiB	2^{70}
yottabyte	YB	10^{24}	yobibyte	YiB	2^{80}

bit (b): 1 bit = 1/8 Byte

Byte (B): 8 bit

$$\mathbf{B = 8b}$$

$$\mathbf{b = B/8}$$

bit	(b)	1 cifra binaria	memorizza 0 oppure 1
byte	(B)	8 bit	memorizza un carattere
Kibibyte	(KiB)	1024 byte	circa mezza pagina di testo
Mebibyte	(MiB)	1024 KiB	un libro di 200 pagine
Gibibyte	(GiB)	1024 MiB	alcuni volumi
Tebibyte	(TiB)	1024 GiB	una biblioteca
Pebibyte	(PiB)	1024 TiB	molte biblioteche

Codifica binaria dei caratteri

- ▶ Quanti sono gli oggetti compresi nell'insieme?

- ▶ 26 lettere maiuscole + 26 minuscole \Rightarrow 52
- ▶ 10 cifre
- ▶ Circa 30 segni d'interpunzione
- ▶ Circa 30 caratteri di controllo (EOF, CR, LF, ...)

circa 120 oggetti complessivi $\Rightarrow k = \lceil \log_2 120 \rceil = 7$

- ▶ Codice ASCII: utilizza 7 bit e quindi può rappresentare al massimo $2^7 = 128$ caratteri

- ▶ Con 8 bit (= byte) rappresento 256 caratteri (ASCII esteso)
- ▶ Si stanno diffondendo codici più estesi (e.g. UNICODE) per rappresentare anche i caratteri delle lingue orientali

Codice ASCII a 7 bit

MSB \ LSB	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
010	spc	!	“	#	\$	%	&	'	()	*	+	,	-	.	/
011	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
100	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
101	P	Q	R	S	T	U	V	W	X	Y	Z	[\]	^	_
110	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
111	p	q	r	s	t	u	v	w	x	y	z	{		}	~	del

Esempio: il carattere parentesi graffa “ { “ ha codice ASCII decimale **123**, equivalente alla parola ASCII binaria di **7 bit: 111 1011** (si verifichi che la codifica di questo carattere è esatta).

Informazione in formato digitale

- ▶ il risultato? Una stringa di caratteri sarà rappresentata dal computer come una successione di gruppi di 8 bit (8 bit = 1 byte)

O	G	G	I		P	I	O	V	E
01001111	01000111	01000111	01001001	00100000	01010000	01001001	01001111	01010110	01000101

Il set di caratteri ISO-Latin-1

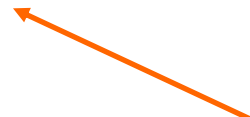
▶ ISO-Latin-1 (ISO-8859-1 o ASCII esteso)

- ▶ unica estensione standard di ASCII
- ▶ 1 byte = 8 bit = 2^8 punti di codice = 256 caratteri rappresentati
- ▶ sufficiente per lingue europee occidentali (italiano, francese, ecc.)

ASCII Standard



	0	1	2	3	4	5	6	7	8	9
0										
1										
2										
3				!	"	#	\$	%	&	'
4	()	*	+	,	-	.	/	0	1
5	2	3	4	5	6	7	8	9	:	;
6	<	=	>	?	@	A	B	C	D	E
7	F	G	H	I	J	K	L	M	N	O
8	P	Q	R	S	T	U	V	W	X	Y
9	Z	[\]	^	_	`	a	b	c
10	d	e	f	g	h	i	j	k	l	m
11	n	o	p	q	r	s	t	u	v	w
12	x	y	z	{		}	~	•	•	•
13	,	f	„	…	†	‡	~	‰	\$	<
14	œ	•	•	•	•	•	•	•	•	•
15	-	—	-	™	\$	>	œ	•	•	ÿ
16		ı	ç	£	□	¥	ı	\$	”	©
17	ˆ	«	¬	-	®	-	°	±	²	³
18	ˆ	µ	¶	.	,	1	°	»	¼	½
19	¾	¿	À	Á	Â	Ã	Ä	Å	Æ	Ç
20	È	É	Ê	Ë	Ì	Í	Î	Ï	Ð	Ñ
21	Ò	Ó	Ô	Õ	Ö	×	Ø	Ù	Ú	Û
22	Ü	Ý	ß	à	á	â	ã	ä	å	ä
23	æ	ç	è	é	ê	ë	ì	í	î	ï
24	ö	ñ	ò	ó	ô	õ	ö	÷	ø	ù
25	ú	û	ü	ý	þ	ÿ				



Caratteri di controllo

0-32



128-159

La famiglia di caratteri ISO-8859

- ▶ 14 set di caratteri standardizzati da ISO (International Standard Organization)
- ▶ Codifica: 1 byte = 256 caratteri rappresentati da ciascun set
- ▶ Soprainsiemi dei caratteri ASCII Standard
 - ▶ punti di codice 0 - 127 (parte comune) ASCII
 - ▶ punti di codice 128 - 159 codici di controllo (non corrispondono a caratteri grafici)
 - ▶ punti di codice 160 - 255 (parte variabile) caratteri aggiuntivi per greco, cirillico, lingue slave, arabo, ebraico, ecc.
- ▶ I set di ISO-8859 sono tutti reciprocamente incompatibili
 - ▶ Punto di codice 232
 - ▶ ISO-8859-1 (Latin-1) = “è”
 - ▶ ISO-8859-5 (Cyrillic) = “ш”
- ▶ ISO-8859 non copre lingue come giapponese, cinese, ecc.

La famiglia di caratteri ISO-8859

ISO-Latin-1

The parts of ISO 8859		
standard	name of alphabet	characterization
ISO 8859-1	Latin alphabet No. 1	"Western", "West European"
ISO 8859-2	Latin alphabet No. 2	"Central European", "East European"
ISO 8859-3	Latin alphabet No. 3	"South European"; "Maltese & Esperanto"
ISO 8859-4	Latin alphabet No. 4	"North European"
ISO 8859-5	Latin/Cyrillic alphabet	(for Slavic languages)
ISO 8859-6	Latin/Arabic alphabet	(for the Arabic language)
ISO 8859-7	Latin/Greek alphabet	(for modern Greek)
ISO 8859-8	Latin/Hebrew alphabet	(for Hebrew and Yiddish)
ISO 8859-9	Latin alphabet No. 5	"Turkish"
ISO 8859-10	Latin alphabet No. 6	"Nordic" (Sámi, Inuit, Icelandic)
ISO 8859-11	Latin/Thai alphabet	(for the Thai language)
<i>(Part 12 has not been defined.)</i>		
ISO 8859-13	Latin alphabet No. 7	Baltic Rim
ISO 8859-14	Latin alphabet No. 8	Celtic
ISO 8859-15	Latin alphabet No. 9	"euro"
ISO 8859-16	Latin alphabet No. 10	for a collection of languages (see below)

The Universal Character Set

UNICODE (ISO-10646)

- ▶ Standard internazionale che si prefigge di rappresentare qualsiasi tipo di carattere appartenente ai sistemi grafici esistenti
 - ▶ Sistemi di scrittura di tutte le lingue europee, asiatiche, africane, ecc., sia antiche che moderne.
 - ▶ Sistemi di caratteri basati sui fonemi (p.es. italiano), sulle sillabe (p.es. Thai), su ideogrammi (p.es. Cinese), geroglifici, braille, ecc.
 - ▶ Sistemi di simboli tecnici e scientifici (p.es. matematica, logica)
 - ▶ Punteggiatura e segni diacritici (p.es. accenti)
- ▶ Risolve i problemi di incompatibilità dei sistemi ISO-8859
 - ▶ estende l'insieme dei caratteri supportati
 - ▶ permette la realizzazione di documenti multilingui
- ▶ <http://www.unicode.org>

The Universal Character Set UNICODE (ISO-10646)

- ▶ Circa 96.000 caratteri grafici rappresentati (Unicode v. 4.0)
... ma i punti di codice disponibili sono più di 1 milione (4 byte)!!

- Arabic
- Armenian
- Bengali
- Bopomofo
- Buhid
- Canadian Syllabics
- Cherokee
- Cyrillic
- Deseret
- Devanagari
- Ethiopic
- Georgian
- Gothic
- Greek
- Gujarati
- Gurmukhi
- Han
- Hangul
- Hanunóo
- Hebrew
- Hiragana
- Kannada
- Katakana
- Khmer
- Latin
- Lao
- Malayalam
- Mongolian
- Myanmar
- Ogham
- Old Italic (Etruscan)
- Oriya
- Runic
- Sinhala
- Syriac
- Tagalog
- Tagbanwa
- Tamil
- Telugu
- Thaana
- Thai
- Tibetan
- Yi

I primi 65536 caratteri (2^{16}) costituiscono il Basic Multilingual Plane (BMP), primo di 17 “piani” in cui è diviso UNICODE. Nel BMP, 6500 punti di codice sono riservati per usi privati (loghi, trademarks...)

The Universal Character Set

UNICODE (ISO-10646)

- ▶ I punti di codice sono rappresentati con “U+numero esadecimale”, ed hanno un nome standard:
 - ▶ “A” U+0041 = ”Latin Capital Letter A” (decimale 65)
 - ▶ “ω” = U+03C9 = ”Greek Letter Omega”
- ▶ I primi 256 caratteri sono identici al set di caratteri Latin-1, a sua volta un soprainsieme dell' ASCII.

UNICODE:

composizione dei caratteri

- ▶ Caratteri complessi (p.es. “u” con umlaut) possono essere rappresentati in due modi:
 - ▶ Come elementi *precostituiti* (codice U+00FC, “ü”)
 - ▶ Come elementi *composti*, formati da un carattere di base quale “u” (U+0075) ed uno o più caratteri che non introducono spaziatura (“non spacing”), che vengono quindi sovrascritti al precedente, in questo caso “̈” (U+0308)
- ▶ Problema dell’ordinamento alfabetico (come decomporre caratteri complessi)

Blocchi di codice

- ▶ I vari alfabeti sono divisi in gruppi detti “blocchi di codice” (code blocks)
- ▶ Si è tuttavia cercato di non duplicare i caratteri di lingue diverse. Non esiste p.es. una “A” italiana, una “A” norvegese.
- ▶ Anche le migliaia di ideogrammi comuni alla scrittura cinese, giapponese e coreana (che discendono tutti da una scrittura comune) sono unificati.

The Universal Character Set UNICODE (ISO-10646)

ASCII/8859-1 Text

A	0100 0001
S	0101 0011
C	0100 0011
I	0100 1001
I	0100 1001
/	0010 1111
8	0011 1000
8	0011 1000
5	0011 0101
9	0011 1001
-	0010 1101
l	0011 0001
	0010 0000
t	0111 0100
e	0110 0101
x	0111 1000
t	0111 0100

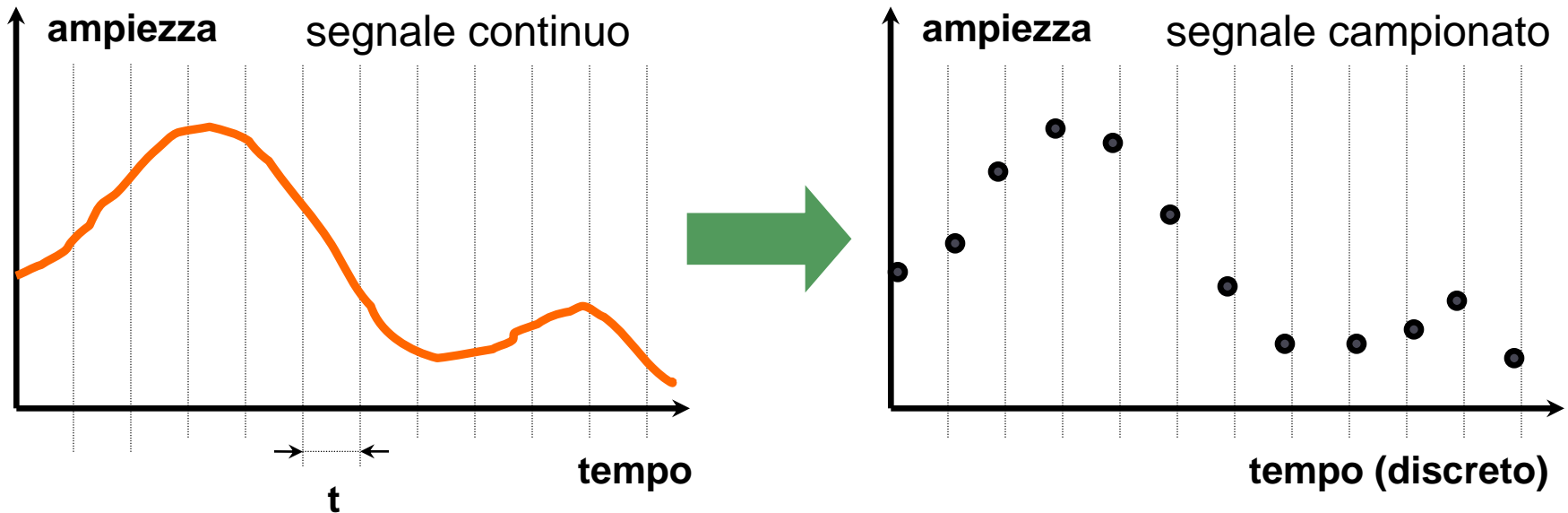
Unicode Text

A	0000 0000 0100 0001
S	0000 0000 0101 0011
C	0000 0000 0100 0011
I	0000 0000 0100 1001
I	0000 0000 0100 1001
	0000 0000 0010 0000
天	0101 1001 0010 1001
地	0101 0111 0011 0000
	0000 0000 0010 0000
س	0000 0110 0011 0011
ل	0000 0110 0100 0100
ا	0000 0110 0010 0111
م	0000 0110 0100 0101
	0000 0000 0010 0000
α	0000 0011 1011 0001
φ	0010 0010 0111 0000
γ	0000 0011 1011 0011

Campionamento e quantizzazione

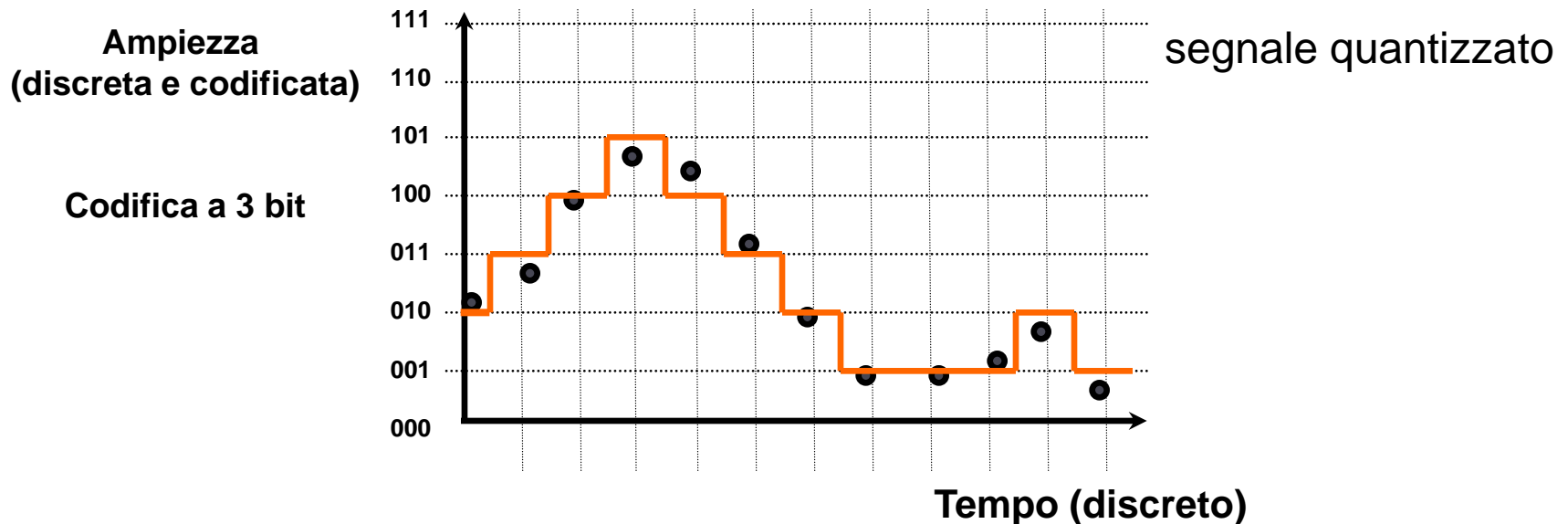
- ▶ Gli elaboratori elettronici hanno natura **discreta**, ovvero ogni grandezza in gioco può essere rappresentata soltanto da un numero finito di elementi.
- ▶ Per essere elaborati da un calcolatore, segnali intrinsecamente **continui** quali *suoni, immagini, video* ecc., devono essere discretizzati (*digitalizzati*) attraverso operazioni di **campionamento** e **quantizzazione**.

Campionamento



- ▶ Il segnale continuo viene campionato ad intervalli di tempo regolari t (t = intervallo di campionamento).
- ▶ Il segnale risultante è un insieme finito di punti equidistanti nel tempo. Tuttavia le ampiezze devono essere ancora approssimate ad intervalli discreti, ovvero **quantizzate**.
- ▶ Si noti che campionamento e quantizzazione comportano una perdita di informazione.

Quantizzazione



- ▶ La quantizzazione suddivide l'ampiezza in n intervalli uguali che vengono poi codificati in binario. Ogni valore di ampiezza del segnale campionato viene approssimato al più vicino valore discreto di ampiezza.
- ▶ Più valori (e quindi più bit) si utilizzano per suddividere le ampiezze, più il segnale risultante sarà preciso.

Le immagini digitali



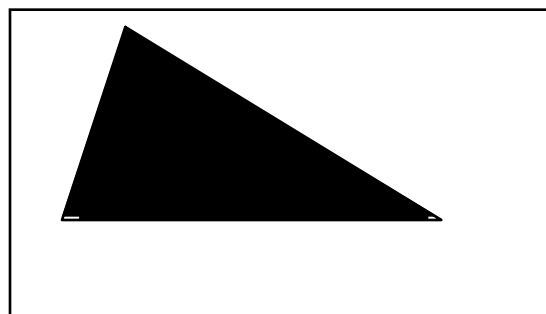
Le immagini digitali non hanno una struttura continua ma sono costituite da un numero finito di componenti monocromatiche (**pixel**) prodotte dal campionamento dell'immagine reale. I pixel assumono un numero finito di tonalità definite dalla quantizzazione dell'immagine campionata.

Codifica delle immagini

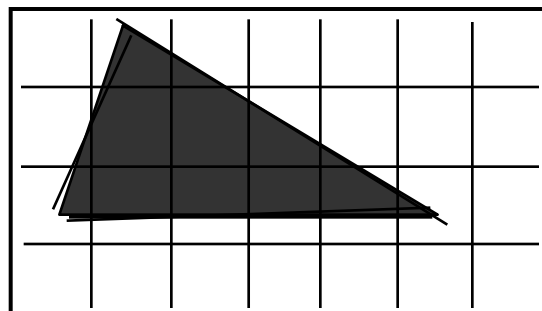
- ▶ L'immagine è suddivisa in punti (*pixel*) e ciascun punto è codificato con un numero che corrisponde
 - ▶ A un particolare colore
 - ▶ A un particolare tono di grigio nelle immagini b/n
- ▶ In genere si utilizza un numero di colori o di sfumature di grigio che sia potenza di 2 per rappresentare un'immagine come sequenza di byte.
- ▶ Deve essere memorizzata anche la dimensione dell'immagine e la risoluzione (*dpi*, "dot per inch")

Codifica di immagini

- ▶ 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



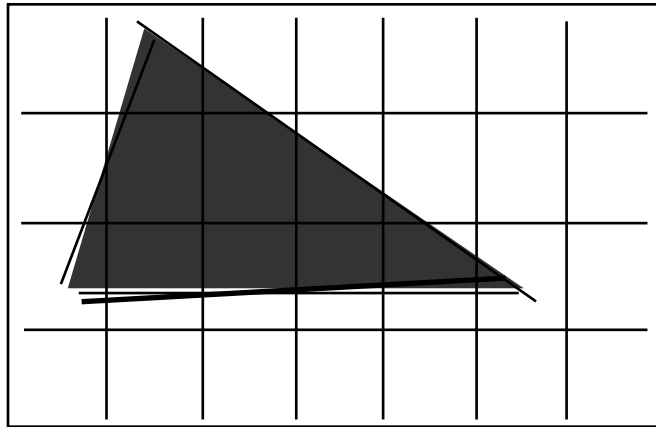
Codifica di immagini

- ▶ 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)

Codifica di immagini

Poiché una sequenza di bit è lineare, si deve definire una convenzione per **ordinare** i pixel della griglia

Assumiamo che i pixel siano ordinati dal basso verso l'alto e da sinistra verso destra



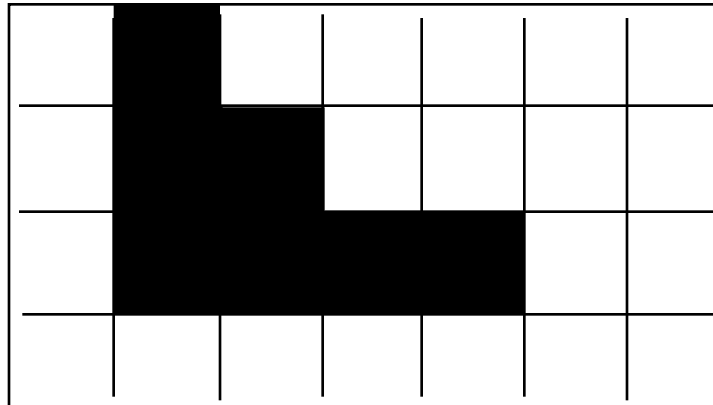
0 ₂₂	1 ₂₃	0 ₂₄	0 ₂₅	0 ₂₆	0 ₂₇	0 ₂₈
0 ₁₅	1 ₁₆	1 ₁₇	0 ₁₈	0 ₁₉	0 ₂₀	0 ₂₁
0 ₈	1 ₉	1 ₁₀	1 ₁₁	1 ₁₂	0 ₁₃	0 ₁₄
0 ₁	0 ₂	0 ₃	0 ₄	0 ₅	0 ₆	0 ₇

La rappresentazione della figura è data dalla stringa binaria

000000 0111100 0110000 0100000

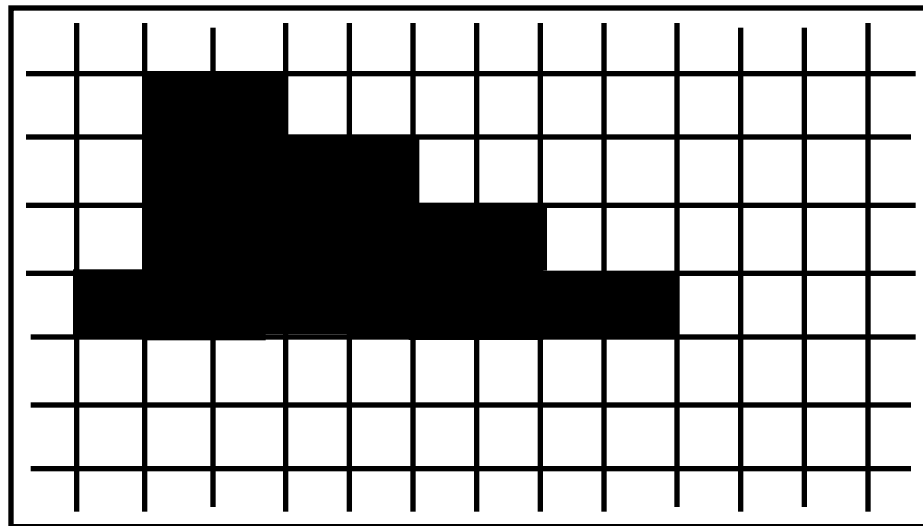
Codifica di immagini

- ▶ Non sempre il contorno della figura coincide con le linee della griglia: nella codifica si ottiene un'*approssimazione* della figura originaria
- ▶ Se riconvertiamo la stringa **00000000 | | | | 000 | | 00000 | 00000** in immagine otteniamo



Codifica di immagini

- ▶ La rappresentazione dell'immagine - e quindi la sua qualità - sarà tanto più fedele all'originale quanto più aumenterà il numero di pixel; ovviamente se i pixel aumentano in numero, diminuiscono di dimensione.



Codifica di immagini



Codifica di immagini con toni di grigio

- ▶ Le immagini in bianco e nero hanno delle sfumature, o **livelli di intensità di grigio**
- ▶ Per codificare immagini con sfumature:
 - ▶ si fissa un insieme di livelli (*toni*) di grigio, cui si assegna convenzionalmente una rappresentazione binaria
 - ▶ per ogni pixel si stabilisce il livello medio di grigio e si memorizza la codifica corrispondente a tale livello
- ▶ Per memorizzare un pixel non è più sufficiente 1 bit.
 - ▶ con 4 bit si possono rappresentare $2^4=16$ livelli di grigio
 - ▶ con 8 bit ne possiamo distinguere $2^8=256$,
 - ▶ con **K** bit ne possiamo distinguere 2^K

Codifica di immagini a colori

- ▶ Analogamente possono essere codificate le immagini a colori:
 - bisogna definire un insieme di sfumature di colore differenti (es. RGB = Red Green Blue), codificate mediante una opportuna sequenza di bit
- ▶ La rappresentazione di un'immagine mediante la codifica dei pixel viene chiamata **codifica bitmap**
- ▶ Il numero di byte richiesti dipende dalla **risoluzione** e dal **numero di colori** che ogni pixel può assumere
- ▶ *Es:* 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)**
- ▶ I monitor tipici utilizzano
 - ▶ risoluzione: 640×480 , 1024×768 , 1280×1024
 - ▶ numero di colori per pixel: da **256** fino a **16 milioni**
- ▶ Tecniche di **compressione** consentono di ridurre notevolmente lo spazio occupato dalle immagini

Codifica di filmati

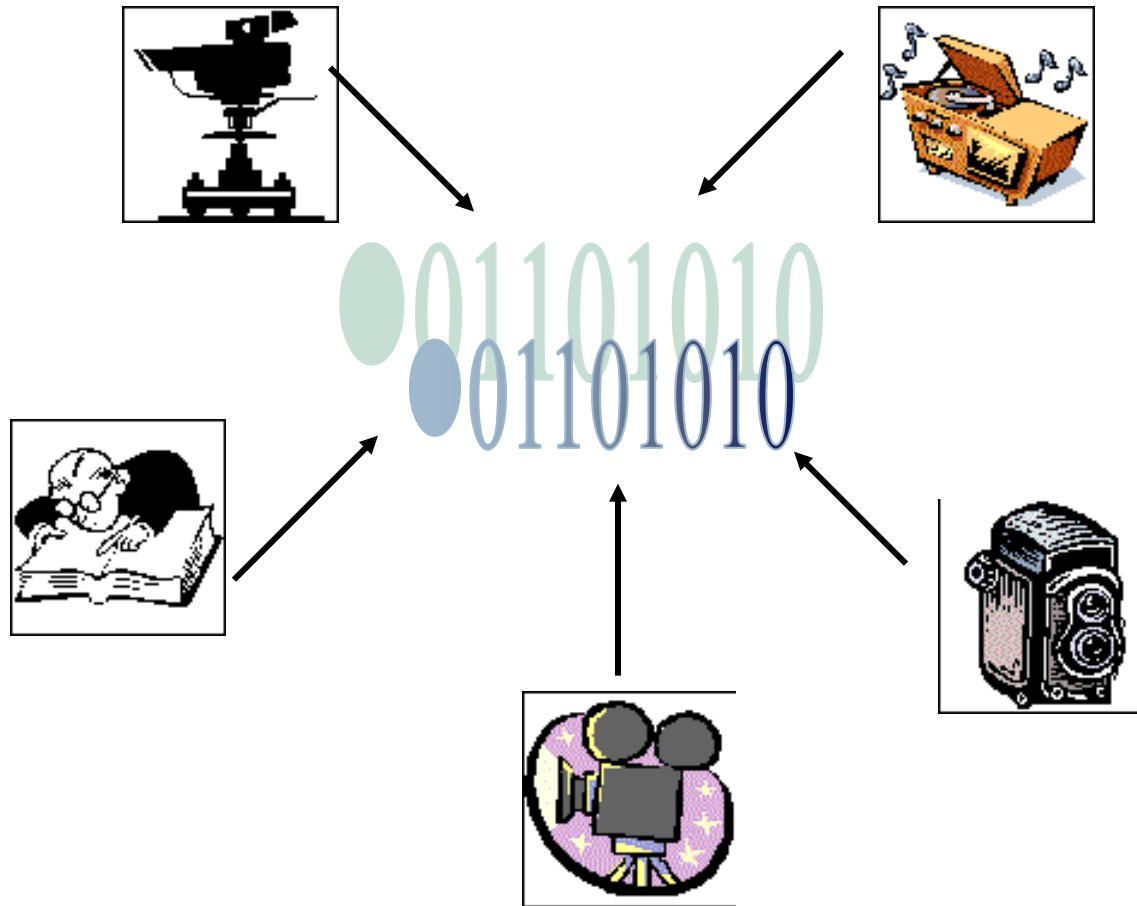
- ▶ Un filmato non è altro che una successione di fotogrammi (frame) accompagnata da una colonna sonora!
- ▶ Immagini in movimento sono memorizzate come sequenze di fotogrammi
- ▶ In genere si tratta di sequenze compresse di immagini
 - ▶ ad esempio si possono registrare solo le variazioni tra un fotogramma e l'altro
- ▶ Esistono vari formati (comprendente il sonoro):
 - ▶ *mpeg* (il più usato)
 - ▶ *avi* (microsoft)
 - ▶ *quicktime* (apple)
 - ▶ *mov*
- ▶ E' possibile ritoccare i singoli fotogrammi



Codifica di 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*
 - ▶ **.midi** usato per l'elaborazione della musica al PC

La convergenza e i suoi strumenti



La convergenza e i suoi strumenti

con la convergenza al digitale:

- ▶ tendono a unificarsi i supporti (memorie di massa, rete)
- ▶ tendono a unificarsi le tecnologie di produzione
- ▶ tendono a unificarsi le tecnologie di riproduzione
- ▶ tendono a unificarsi i mercati
- ▶ i dati passano con facilità da un dispositivo all'altro