Introduzione all'Informatica

Giuseppe Manco

Rappresentazione delle Informazioni

Lezione 2

09 Ottobre 2003

Codifica dati e istruzioni

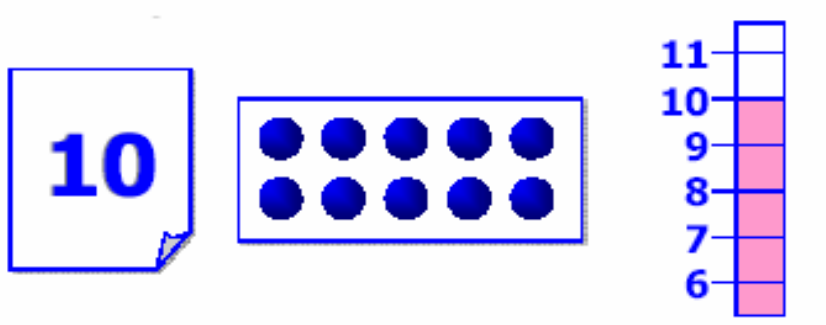
- Algoritmi
  - Istruzioni che operano su dati
- Per scrivere un programma è necessario rappresentare dati e istruzioni in un formato tale che l'esecutore automatico sia in grado di
  - Memorizzare istruzioni e dati
  - Manipolare istruzioni e dati

RAPPRESENTAZIONE DELLE INFORMAZIONI

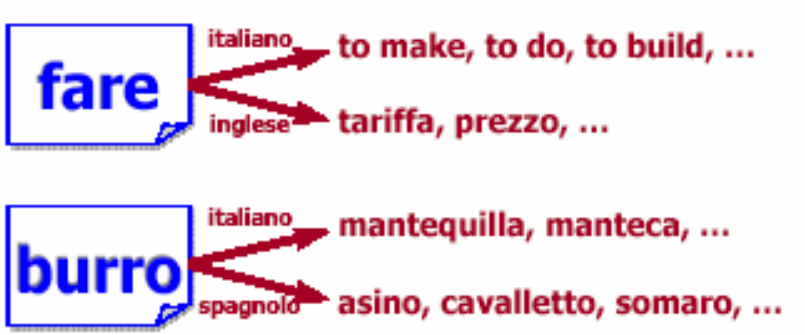
- Le informazioni gestite dai sistemi di elaborazione devono essere **codificate**
  - per poter essere memorizzate, elaborate, scambiate,…
- In un elaboratore bisogna codificare sia dati sia istruzioni

RAPPRESENTAZIONE DELLE INFORMAZIONI

- La stessa informazione si può rappresentare in modi differenti



- Stessa rappresentazione per informazioni differenti



SISTEMI DI CODIFICA

Sistema di codifica (o ***codifica***, o ***codice***)

- Usa un insieme di simboli di base (**alfabeto**)
- I simboli dell'alfabeto possono essere combinati ottenendo differenti **configurazioni** (o ***codici***, o ***stati***), distinguibili l'una dall'altra
- Associa ogni configurazione ad una particolare entità di informazione (la configurazione diventa un modo per rappresentarla)

Sistemi di Codifica

- Alfabeto
  - Cifre “0”, “1”, “2”, …, “9”
  - separatore decimale (“,”)
  - separatore delle migliaia (“.”)
  - Segni positivo (“+”) e negativo (“-”)
- Regole di composizione (sintassi)
  - Definiscono le combinazioni ben formate
    - 12.318,43
    - 12,318.43
- Codice (semantica)
  - Associano ad ogni configurazione un’entità di informazione
    - $12.318,43 = 1\times10^4+ 2\times10^3+ 3\times10^2+ 1\times10+ 1\times10^0+ 4\times10^{-1}+ 3\times10^{-2}$
- Lo stesso alfabeto può essere usato per codici diversi
  - $123,456 = 1\times10^2+ 2\times10+ 3\times10^0+ 4\times10^{-1}+ 5\times10^{-2} + 6\times10^{-3}$ [IT]
  - $123,456 = 1\times10^5+ 2\times10^4+ 3\times10^3+ 4\times10^2+ 5\times10^1 + 6\times10^0$ [UK]

SISTEMI DI CODIFICA

- Esistono standard internazionali per risolvere problemi di compatibilità
  - tra differenti sistemi software
  - tra calcolatori di tipo e marca diversi
- Vedremo brevemente sistemi di codifica per:
  - Codifica di numeri
  - Codifica di caratteri
  - Codifica di immagini
  - Codifica di sequenze sonore

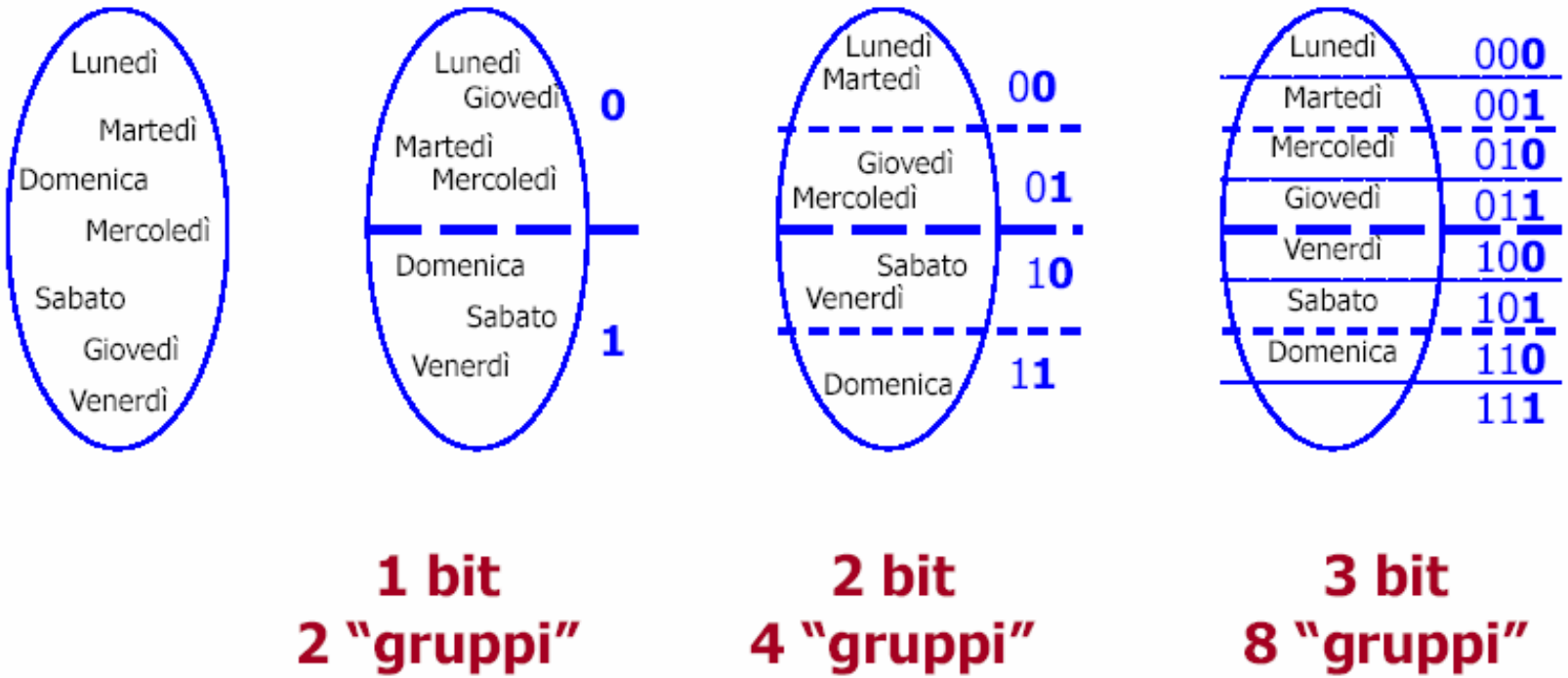
CODIFICA BINARIA

- Codifica binaria: usa un alfabeto di **2** simboli
- Utilizzata nei sistemi informatici
  - Si utilizza una grandezza fisica (luminosità, tensione elettrica, la corrente elettrica), per rappresentare informazione
  - Si divide in intervalli l'insieme dei valori che la grandezza può assumere: ogni intervallo corrisponde ad un simbolo
- Solo 2 simboli al fine di ridurre la probabilità di errore
  - Tanto più simboli si devono distinguere e tanto meno la rivelazione sarà affidabile (gli intervalli della grandezza fisica saranno meno ampi)

CODIFICA BINARIA

- **BIT (BInary digiT)**
  - unità elementare di informazione rappresentabile con dispositivi elettronici
  - con 1 bit si possono rappresentare 2 stati
    - 0/1, on/off, si/no
- Combinando più bit si può codificare un numero maggiore di stati
  - con 2 bit possono rappresentare 4 stati
  - con **K** bit si possono rappresentare $\mathbf{2^K}$ stati
- Quanti bit sono necessari per codificare N oggetti?
  - $N \leq 2^K \rightarrow K \geq \log_2 N \rightarrow \mathbf{K = \lceil \log_2 N \rceil}$

I giorni della settimana in binario



CODIFICA BINARIA: UNITA' DERIVATE

- **BYTE** = 8 bit
  - può rappresentare $2^8 = 256$ stati
- **KiloByte** (KB) = 2^{10} byte = 1024 byte $\cong 10^3$ byte
- **MegaByte** (MB) = 2^{20} byte $\cong 10^6$ byte
- **GigaByte** (GB) = 2^{30} byte $\cong 10^9$ byte
- **TeraByte** (TB) = 2^{40} byte $\cong 10^{12}$ byte

CODIFICA DEI NUMERI NATURALI

- Sistema di numerazione posizionale con **base β**
  - β simboli (**cifre**) corrispondono ai numeri da 0 a β
  - i numeri naturali maggiori o uguali a β possono essere rappresentati da una sequenza di cifre
- Se un numero naturale *N* è rappresentato in base β dalla sequenza di *n* cifre

$$a_{n-2}\ a_{n-1}\ a_n \ldots a_1\ a_0$$

allora *N* può essere espresso come segue:

$$N = \sum_{i=0}^{n-1} a_i \beta^i = a_{n-1}\beta^{n-1} + a_{n-2}\beta^{n-2} + \ldots + a_2\beta^2 + a_1\beta + a_0$$

CODIFICA DEI NUMERI NATURALI

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

$$13 = \mathbf{1} \times 8 + \mathbf{1} \times 4 + \mathbf{0} \times 2 + \mathbf{1} \times 1$$

cioè può essere rappresentato dalla sequenza di bit

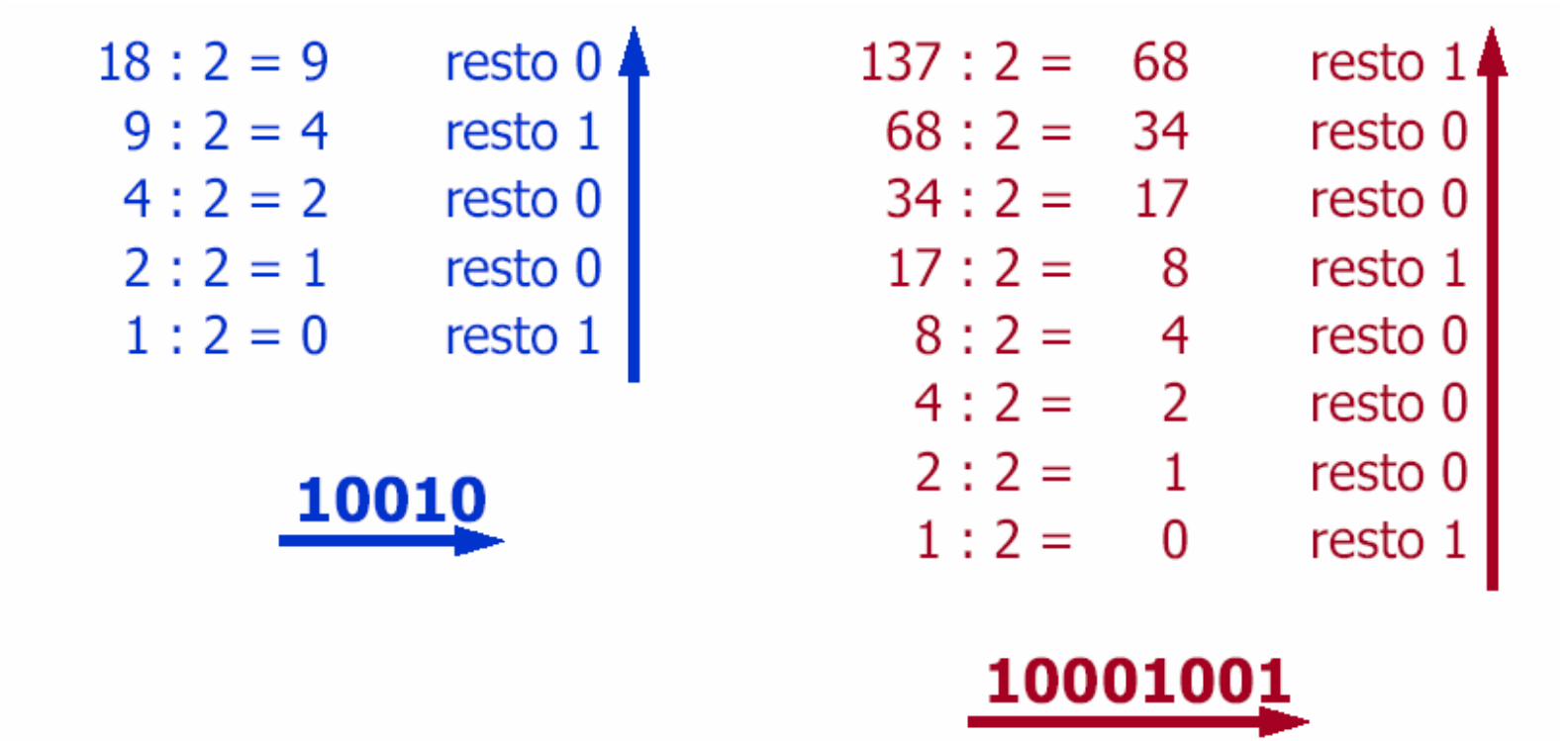
1 1 0 1

CODIFICA DEI NUMERI NATURALI

- Quindi
  - Numero = sequenza di bit (codifica in base 2)
  - Con **K** bit si rappresentano i numeri da 0 a $\mathbf{2^K-1}$
- Esempi:
  - 2 = sequenza 1 0
  - 3 = sequenza 1 1
  - 4 = sequenza 1 0 0
  - .....

CONVERSIONE DECIMALE-BINARIO

- Si calcolano i resti della divisione per 2



CODIFICA DEI NUMERI INTERI

- Si utilizzano codifiche quali
  - **Bit di segno**
    - Il bit piu` a sinistra rappresenta il segno del numero (0 = '+' , 1 = '-')
  - **Complemento a 1**
  - **Complemento a 2**

Il complemento a 2

- E' necessario indicare il numero k di bit che si vogliono utilizzare
- Numeri positivi
  - Stessa rappresentazione
  - 12, utilizzando un 8 bit

0	0	0	0	1	1	0	0

- Numeri negativi
  - Un numero x è appresentato come $2^k - x$
  - -12, utilizzando 8 bit
    - Rappresentiamo 2^8 -12 = 256 -12 = 244

1	1	1	1	0	1	0	0

Complemento a 2: osservazioni

- Unica rappresentazione dello 0
- La sottrazione viene calcolata come la somma (eventualmente troncando l'ultima cifra)
  - 4 – 6 = (+4) + (-6) = 0100 + 1010 = 1110 = -2
  - 9 – 6 = (+9) + (-6) = 01001 + 11010 = [1]00011 = +3
- Conseguenza: x + (-x) = 0
  - 12 -12 = (+12) + (-12) = 00001100 + 11110100 = [1]00000000
- Rappresenta i valori da $2^{k-1}-1$ a -2^{k-1}
  - con k=4, possiamo rappresentare i valori da 7 a -8
  - Con k=8, possiamo rappresentare i valori da 127 a -128

CODIFICA DEI NUMERI RAZIONALI

- Rappresentazione in **virgola fissa**
  - $0.1011_{due} = 1\times2^{-1} + 0\times2^{-2} + 1\times2^{-3} + 1\times2^{-4}$
    $= 0.5 + 0.125 + 0.0625 = 0.6875_{dieci}$
  - $11.101_{due} = 1\times2^{1} + 1\times2^{0} + 1\times2^{-1} + 0\times2^{-2} + 1\times2^{-3}$
    $= 2 + 1 + 0.5 + 0.125 = 3.625_{dieci}$
  - Il numero di cifre prima e dopo la "virgola" è **fisso**!!
- Rappresentazione in **virgola mobile** (float)
  - usata spesso anche in decimale per rappresentare numeri molto grandi o molto piccoli: 0.1357×10^{64}
    - **mantissa** – parte frazionaria compresa tra 0 e 1 [0.1357]
    - **esponente** – numero **intero** ($\gtrless 0$)
  - utilizza 1 bit per il segno (s), h bit per l'esponente (e) e k bit per la mantissa (m): $\mathbf{R = s \times m \times 2^e}$

CODIFICA DI CARATTERI

- Codifica **ASCII**: 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'

Ascii su 7 bit

	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
010	sp	!	"	#	$	%	&	'	(	)	*	+	,	-	.	/
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	{	|	}	~	canc

CODIFICA DI DATI MULTIMEDIALI

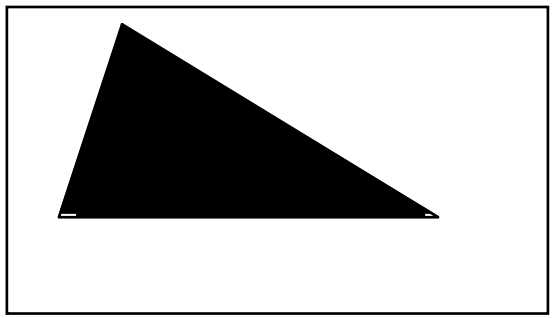
- Lettere e numeri non costituiscono le uniche informazioni utilizzate dagli elaboratori ma si diffondono sempre di più applicazioni che usano ed elaborano anche altri tipi di informazione:
  - diagrammi
  - immagini
  - suoni
- Spesso in questi casi si parla di applicazioni di tipo multimediale

CODIFICA DI IMMAGINI

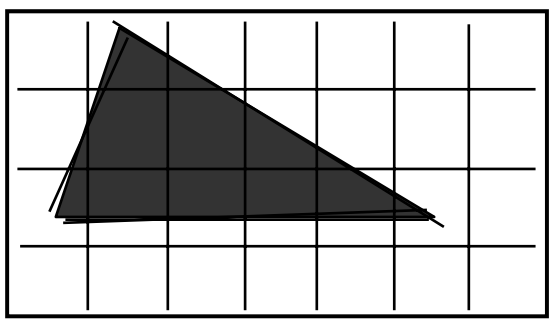
- Esistono numerose tecniche per la memorizzazione digitale e l'elaborazione di un'immagine
- Immagini = sequenze di bit!
- L'immagine viene digitalizzata cioè rappresentata con sequenze di pixel
- Ogni pixel ha associato un numero che descrive un particolare colore (o tonalità di grigio)
- Inoltre si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

CODIFICA 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

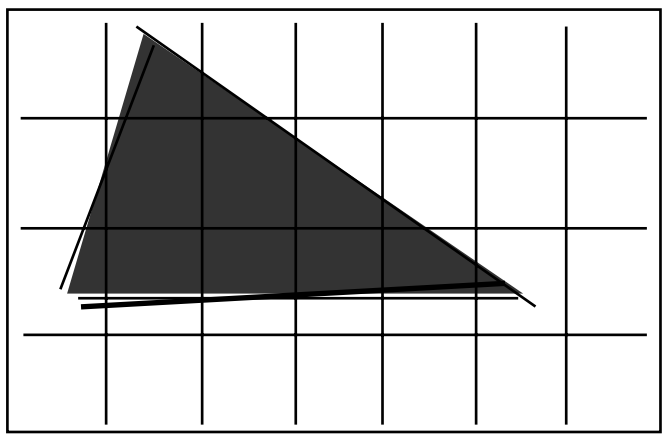


CODIFICA DI IMMAGINI

- 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
- **ipotesi:** assumiamo che i pixel siano ordinati dal basso verso l'alto e da sinistra verso destra



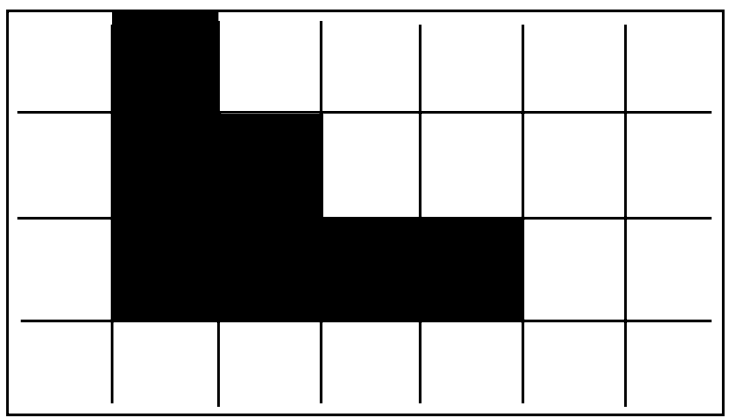
0_{22}	1_{23}	0_{24}	0_{25}	0_{26}	0_{27}	0_{28}
0_{15}	1_{16}	1_{17}	0_{18}	0_{19}	0_{20}	0_{21}
0_{8}	1_{9}	1_{10}	1_{11}	1_{12}	0_{13}	0_{14}
0_{1}	0_{2}	0_{3}	0_{4}	0_{5}	0_{6}	0_{7}

La rappresentazione della figura è data dalla stringa binaria

0000000 0111100 0110000 0100000

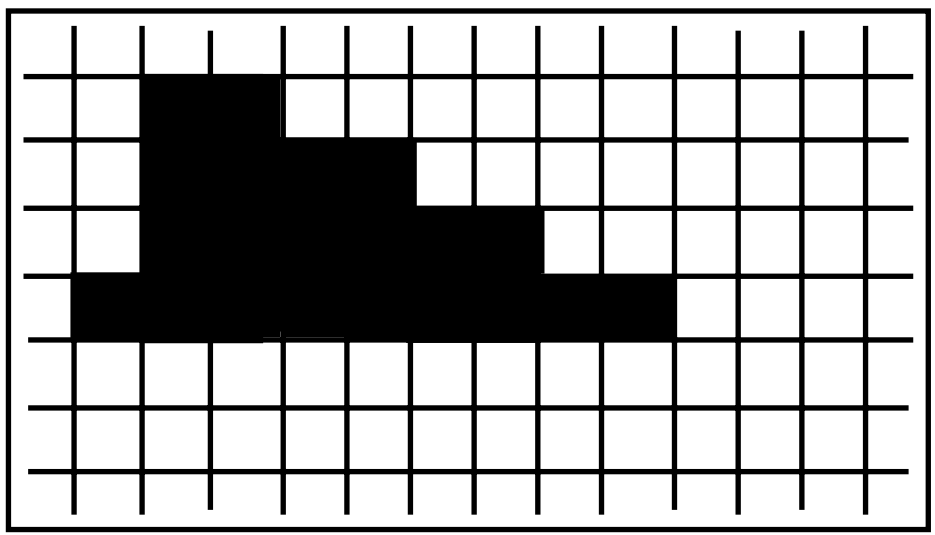
CODIFICA DI IMMAGINI

- 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 **000000001111000110000010000**0 in immagine otteniamo



CODIFICA DI IMMAGINI

- La rappresentazione sarà più fedele all'aumentare del numero di pixel
  - ossia al diminuire delle dimensioni dei quadratini della griglia in cui è suddivisa l'immagine



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, codificate mediante una opportuna sequenza di bit

- La rappresentazione di un'immagine mediante la codifica dei pixel, viene chiamata **codifica bitmap**

CODIFICA DI IMMAGINI A COLORI

- 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

- 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 piu' 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 e l'intervallo di campionamento e maggiore e 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

COMPRESSIONE DEI DATI

- Vantaggi: memorizzazione e trasmissione
- Esempio:
  - {A, C, G, T}
  - A=00, C=01, G=10, T=11
  - ATTACCGAAAACTTCTCTCGGGTG… → 1 milione caratteri = 2 milioni di bit
  - 00111100010110…
  - fr(A)=50%, fr(C)=25%, fr(G)=12.5%, fr(T)=12.5%
  - A=0, C=10, G=110, T=111
  - 01111110101010110…
  - 1 x 50% + 2 x 25% + 2 x 3 x 12.5% x 10^6 = 1.75 milioni di bit → risparmio di 250000 bit!
  - La nuova successione binaria deve essere decodificabile

COMPRESSIONE DEI DATI

- Lossless
  - Senza perdita di informazione
  - Programmi, documenti

- Lossy
  - Con perdita di informazione
  - Rapporto di compressione variabile dall'utente
  - Immagini: GIF, JPEG (elimina lievi cambiamenti di colore)
  - Animazioni: MPEG (memorizza solo differenze tra fotogrammi)
  - Audio: MP3 (elimina suoni a basso volume sovrapposti con suoni ad alto volume)

JPEG: Fattore qualità 90/100 (253KB)

800x600

16,8mln colori 24 bit

Bitmap: 1440000 byte

JPEG: 258971 byte

JPEG: Fattore qualità 50/100 (30KB)

JPEG: Fattore qualità 25/100 (20KB)

JPEG: Fattore qualità 10/100 (12KB)

JPEG: Fattore qualità 1/100 (9KB)

ELABORAZIONE DEI DATI

- **Dati** = sequenze di bit
- **Operazioni e funzioni complesse** = ?
- **Hardware**
  - **da cosa è costituito ?**
  - fornisce operazioni primitive che vengono utilizzate per risolvere problemi attraverso programmi

OPERAZIONI TRA NUMERI

- Ad esempio: somma

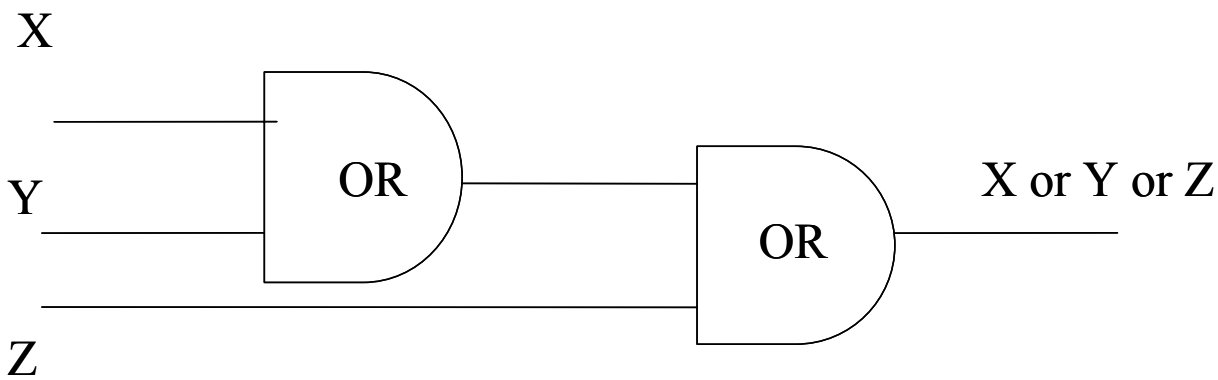
$$5 = 1 \times 4 + 0 \times 2 + 1 \times 1$$
$$2 = 0 \times 4 + 1 \times 2 + 0 \times 1$$
$$7 = 1 \times 4 + 1 \times 2 + 1 \times 1$$

- Esprimibile mediante operazioni su bit

OPERAZIONI LOGICHE

- **Valori di verità** (boolean):
  - ***FALSO*** = 0
  - ***VERO*** = 1
- **Operatori logici**
  - X **AND** Y = *VERO* sse X = *VERO* e Y = *VERO*
  - X **OR** Y = *VERO* sse X = *VERO* oppure Y = *VERO*
  - **NOT** X = *VERO* sse X = *FALSO*
- Gli operatori logici permettono di esprimere **operazioni bit a bit**

CIRCUITI LOGICI



- **Circuito logico**
  - composizione di operatori che trasforma ingressi (input) in uscite (output), e definisce una funzione

    f: input → output
  - input, output = sequenze di bit
  - *Es*: somma fra due numeri (codificati con sequenze di bit)

ELABORAZIONE DEI DATI

- **Dati** = sequenze di bit
- **Operazioni** = **operazioni logiche bit a bit**
- **Funzioni complesse** = **circuiti**
- **Hardware** = **implementazione fisica dei circuiti logici**

CODIFICA DELLE ISTRUZIONI

Istruzioni aritmetico-logiche	
Codice	Istruzione
01100000	ADD
01100100	SUB
01111110	AND
… … …	… … …

- Linguaggio macchina
  - A ogni istruzione è assegnato un codice univoco, detto codice operativo
  - E' necessario specificare dove leggere gli operandi (dati) dell'istruzione e dove scrivere il risultato
  - Il numero di dati che ogni istruzione manipola è variabile in funzione dell'istruzione stessa

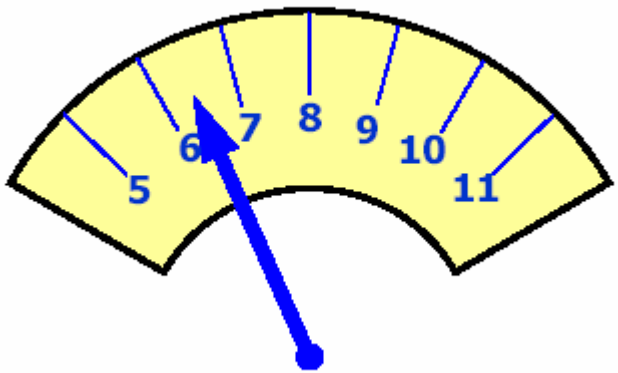
Codifica analogica e codifica digitale

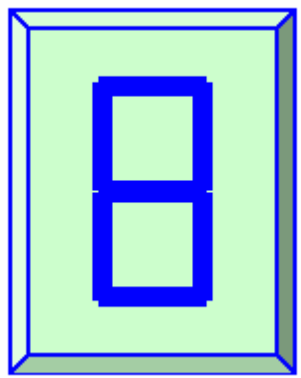
- Le codifiche viste finora non esaminano l'informazione dal punto di vista della "struttura"
  - L'ordine tra gli elementi, la metrica che permette di valutare la distanza tra due elementi…
- L'informazione sulla struttura dell'informazione è chiamata meta-informazione
  - Informazione sull'informazione
- Come possiamo rappresentare informazione e meta-informazione nel supporto?

Due alternative

- Meta-informazione esplicita nel supporto
  - Le relazioni tra le varie configurazioni presenti nel supporto permettono di esprimere relazioni tra le informazioni rappresentate
    - Esempio: i voti degli studenti rappresentati come rettangoli di lunghezza variabile
    - Il confronto della lunghezza permette direttamente il confronto dei voti
- Meta-informazione implicita nella regola di codifica
  - Esempio: voti espressi tramite una serie di simboli
    - A, B, C, D, E, F
  - La regola di codifica (una convenzione che adottiamo) ci dice che A è il voto massimo, e F è il voto minimo
- Cosa succede se vogliamo aggiungere nuove configurazioni?
  - Ad esempio, aggiungendo i mezzi punti

Codifica analogica e digitale





- Analogico
  - Analogia tra struttura delle configurazioni e struttura dell'informazione rappresentata
  - Supporta un insieme potenzialmente infinito di configurazioni
- Digitale
  - Struttura dell'informazione implicita nella regola di codifica
  - Supporta un insieme finito di configurazioni

Perché il digitale è più attrattivo dell'analogico?

- Nessun supporto fisico è immune a perturbazioni esterne
  - L'esistenza di rumore può altera le configurazioni possibili, e di conseguenza l'informazione rappresentata
- Nella codifica analogica ogni configurazione è lecita
  - Il rumore non è distinguibile dal segnale
- Nella codifica digitale sono un insieme prestabilito di configurazioni è lecito
  - Possibilità di distinguere tra informazione e rumore