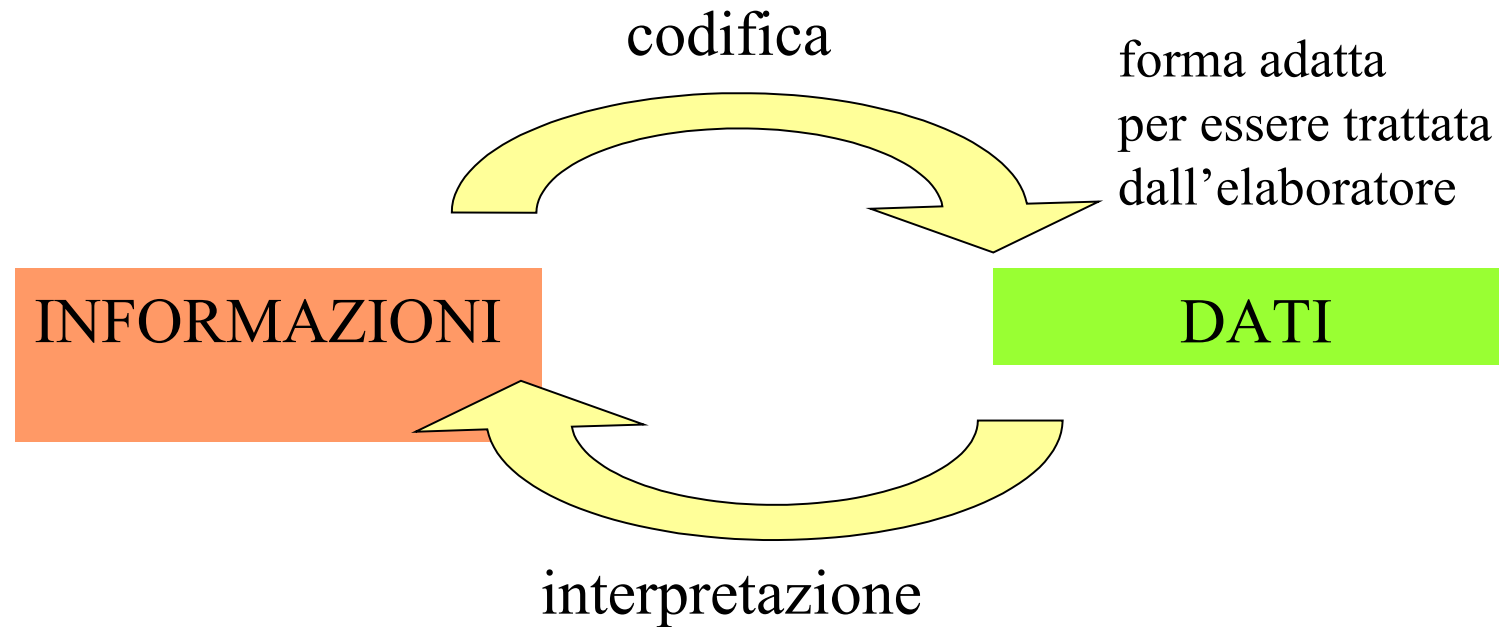


Rappresentazione dell'informazione

La codifica delle informazioni



Informazioni

- Numeri
 - Interi positivi
 - Positivi e negativi
 - Reali
- Testi
- Immagini fisse
 - Vettoriali
 - Bitmap
- Audio
- Video

Informazioni tradizionali

Informazioni multimediali

Rappresentazione dell'informazione

- Le informazioni vengono rappresentate mediante sequenze di simboli
- Nel caso dei simboli binari, le informazioni (numeri, oggetti, parole) sono rappresentate da sequenze dei due simboli (**0** e **1**)
- Servono regole di manipolazione dei simboli

Sistemi numerici

Sistemi numerici

- Per determinare un sistema numerico serve:
 - un insieme limitato di simboli (le **cifre**), che rappresentano quantità prestabilite (1, 2, V, X, M)
 - le **regole** per costruire i numeri:
 - sistemi numerici posizionali
 - sistemi numerici non posizionali

Sistemi numerici

- Sistemi numerici **non posizionali**:
 - valore delle cifre è indipendente dalla posizione
- Sistemi numerici **posizionali**:
 - il valore delle cifre dipende dalla loro posizione all'interno del numero (ogni posizione ha un **peso**)

Sistemi numerici posizionali

- Esempio:

$$N = d_3 d_2 d_1 d_0 ; V(N) = d_3 * p_3 + d_2 * p_2 + d_1 * p_1 + d_0 * p_0$$

- N → rappresentazione del numero
- $V(N)$ → valore del numero

- Sistemi a **base fissa**:

- $p_i = r^i$ dove:
 - r è la **base** del sistema
 - d_i rappresentano le **cifre**

Sistema decimale

- È un sistema numerico posizionale a base fissa
- Il sistema decimale utilizza:
 - $r = 10$
 - $d = 0, 1, 2, 3, 4, 5, 6, 7, 8, 9$

Sistema decimale

8427

=

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

Sistema binario

- Anche il sistema binario è un sistema numerico posizionale a base fissa
- Il sistema binario utilizza:
 - $r = 2$
 - $d = 0,1$
- Ogni cifra è detta **bit** (da **B**inary **dig**i**T**)

Sistema binario

$$1011 = 1 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 = 11_{10}$$

Somma binaria

- La tabella di definizione è:

→ $0 + 0 = 0$

→ $0 + 1 = 1$

→ $1 + 0 = 1$

→ $1 + 1 = 0$ con riporto di 1

→ $1 + 1 + 1 = 1$ con riporto di 1

- Esempi

Sottrazione binaria

- La tabella di definizione è:

→ $0 - 0 = 0$

→ $1 - 0 = 1$

→ $1 - 1 = 0$

→ $0 - 1 = 1$

con prestito di 1 dal
bit di peso superiore

- Esempi

Moltiplicazione e divisione

- Si utilizzano le stesse procedure:
 - per la moltiplicazione: somma e scorrimento
 - per la divisione: differenza e scorrimento
- Shift a sinistra di n -> moltiplico per 2^n
- Shift a destra di n -> divisione intera per 2^n

Conversioni di base

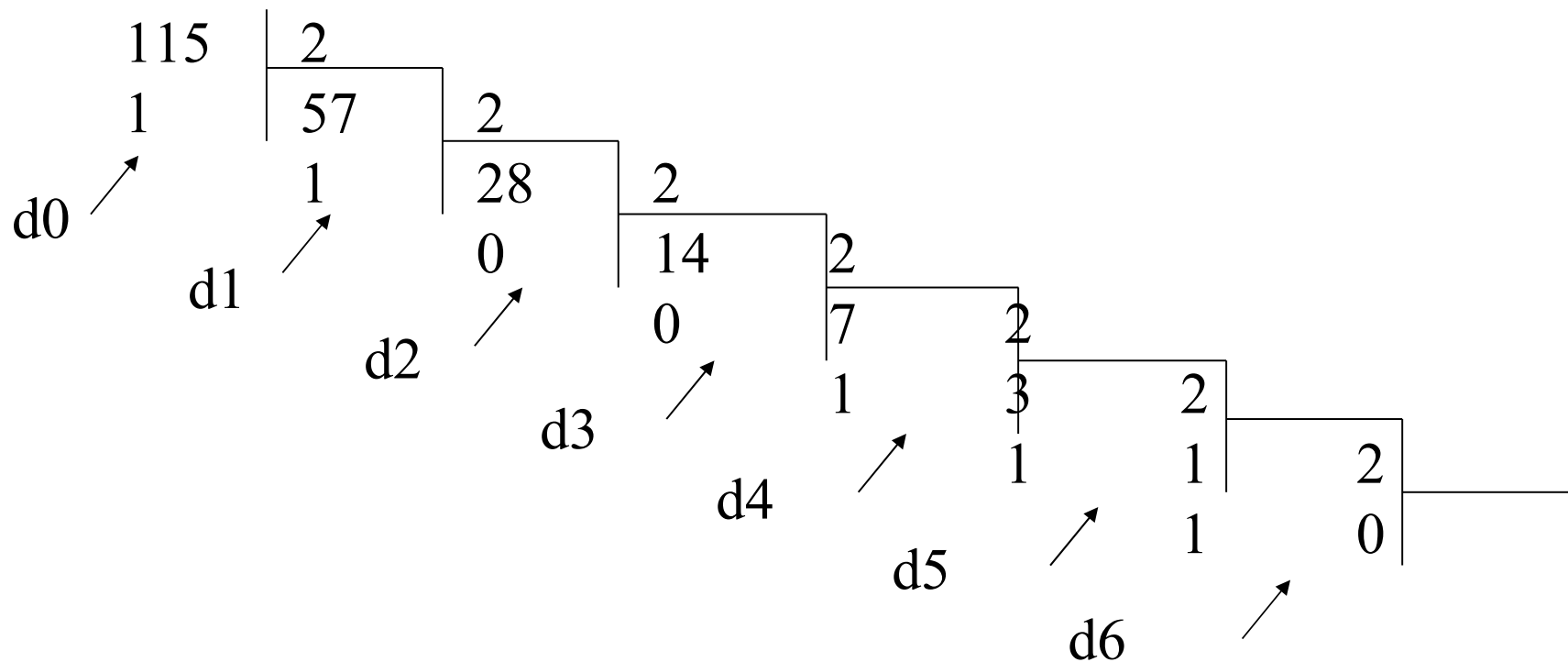
Dal sistema binario a quello decimale

- Utilizzando la definizione:
 - $1010_2 = (1*8 + 0*4 + 1*2 + 0*1)_{10} =$
 $= (8+2)_{10} = 10_{10}$
- Oppure si può utilizzare il seguente formato:
 - $N = ((d_{n-1}*r + d_{n-2})*r + d_{n-3}) \dots *r + d_0$

Conversioni di base

Dal sistema decimale a quello binario

- Esempio: $115_{10} = 1110011_2$:



Altri sistemi utilizzati

- Sistema **ottale**:
 - $r = 8$
 - $d = 0, 1, 2, 3, 4, 5, 6, 7$

- Sistema **esadecimale**:
 - $r = 16$
 - $d = 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F$

- I suddetti sistemi assumono una importanza particolare nel mondo dei calcolatori !

Conversione da binario ad esadecimale

- Esiste una corrispondenza diretta tra cifre esadecimali e il corrispondente binario

10001011₂

16 = 2⁴ !!!

1000 1011

↙ ↘
8B_h

Rappresentazione dei numeri nei calcolatori

Numero di cifre necessario

- Le macchine hanno vincoli spaziali:
 - è necessario conoscere il massimo valore rappresentabile:
 - con n bit si può rappresentare al massimo il numero $2^n - 1$
 - è facile determinare che per poter rappresentare fino ad X , sono necessari un numero n di bit pari a:
$$n = \text{INT}(\log_2 (X+1))$$

Rappresentazione dei numeri nei calcolatori

- Esiste un limite al numero di bit impiegati per rappresentare un numero
- Tale limite dipende da:
 - intervallo di variabilità
 - occupazione di memoria
- Dato che la rappresentazione è formata da un numero finito di bit, se si supera tale limite si ha errore (*overflow*)

Numeri negativi

- Esistono diverse possibilità di rappresentazione:
 - **modulo e segno**
 - **complemento a 2**

Modulo e segno

- Convenzione per il bit più significativo:
 - 0 : segno positivo
 - 1 : segno negativo
- esistono due rappresentazioni per lo '0'

+ 5 ---> 00101

- 10 ---> 11010

+ 0 ---> 00000

- 0 ---> 10000

Complemento a 2 (complemento alla base)

- Dato X in base 2 di n cifre:

$$2^n - X \quad (\text{complemento a 2 del numero } X)$$

- Se $X=01011$, e numero di cifre 5

$$2^5 - X = 100000 - 01011 = 10101$$

- **Regoletta pratica:** il complemento a 2 si trova analizzando i bit del numero a partire da destra: si riportano invariati tutti gli zeri fino al primo bit a 1, si riporta invariato questo stesso bit a 1, si complementano ($0 \rightarrow 1$, $1 \rightarrow 0$) tutti gli altri bit

Complemento a 2 (complemento alla base)

- complemento a 2:
 - per definizione il complemento a 2 di X è $2^n - X$
 - unica rappresentazione dello '0'
- I numeri positivi hanno il bit più significativo (segno) posto a zero.
- I numeri negativi sono rappresentati dal complemento a 2 del corrispondente numero positivo, segno compreso. Pertanto, i numeri negativi hanno il bit più significativo sempre a 1.

→ Esempio: +3 \Leftrightarrow 00011
 - 3 \Leftrightarrow 11101

Uso dei numeri negativi

- Usando modulo e segno:
 - la somma algebrica di numeri positivi e negativi può generare problemi
 - servono sistemi hardware specifici per la gestione corretta del formato
 - E' necessario riconoscere il segno dal primo bit

Usando il complemento a due:

- La sottrazione si esegue con una somma!

Numeri negativi: intervallo valori rappresentabili

Rappresentazione modulo e segno

$$-2^{n-1} + 1 \leq N \leq 2^{n-1} - 1$$

Rappresentazione in complemento a due

$$-2^{n-1} \leq N \leq 2^{n-1} - 1$$

Rappresentazione numeri reali

- I numeri reali sono nel range $[-\infty \div +\infty]$
- Talvolta è necessaria una rappresentazione estesa sulla retta dei reali
 - con 3 simboli $[+/-]$, $X, Y, Z \in \{0, 1, \dots, 9\}$ è possibile rappresentare $-999 \div +999$
 - oppure $9 * 10^{[+/-] 99}$
 - oppure $[+/-] 9 * 10^{[+/-] 99}$

Virgola mobile

- E' la risposta alla necessità di manipolare numeri di ordini di grandezza diversi
- **Notazione scientifica** - numeri espressi nella forma:

$$X.YYY * 10^{WW}$$

- X: parte intera
- Y: parte frazionaria
- W: esponente

Virgola mobile

- Nomenclatura:

$$A = M * B^E$$

- M: mantissa

- B: base

- E: esponente

- Necessita di un segno per la mantissa e uno per l'esponente

Virgola mobile

- Forma normalizzata:

$$\text{numero} = \pm 1.\text{XXXXXXXX} * 2^a$$

- .XXXXXXXX parte frazionaria
- a è detto esponente vero

Virgola mobile

- Rappresentazione standard (IEEE P754)
 - **Segno**: **1 bit** di segno (0 per i positivi, 1 per i negativi)
 - **Esponente**: l'esponente vero è rappresentato come numero senza segno su **8 bit** usando la rappresentazione **eccesso 127** (il valore che quindi si rappresenta è $a+127$; il valore dell'esponente vero dovrà essere in modulo minore di 127)
 - **Mantissa**: vengono rappresentati i primi **23 bit** della parte frazionaria della forma normalizzata (*hidden bit* : la parte intera di peso 2^0 viene sottintesa)

Virgola mobile

- Rappresentazione IEEE P754 (32 bit)
 - 1 bit per il segno
 - 8 bit per l'esponente (rappresentazione eccesso 127)
 - 23 per la mantissa (parte frazionaria - normalizzata)

Segno S	Esponente E	Mantissa F
------------	----------------	---------------

$$N = (-1)^S * 2^{E-127} * 1.F$$

Virgola mobile - precisione

S (1 bit)	E (8 bit)	F (23 bit)
--------------	--------------	---------------

32 bit (singola precisione), $M=127$

S (1 bit)	E (11 bit)	F (52 bit)
--------------	---------------	---------------

64 bit (doppia precisione), $M=1023$

S (1 bit)	E (15 bit)	F (112 bit)
--------------	---------------	----------------

128 bit (quadrupla precisione), $M=16383$

Virgola mobile

- Esempi usando: $B=10$, 2 cifre all'esponente e 8 alla mantissa:

→ +1 0 01 10000000

→ -63517,8 1 05 63517800

→ -0,000635178 1 97 63517800

→ $-8,75 * 10^{-13}$ 1 88 87500000

Virgola mobile

- Moltiplicazione e divisione:
 - si moltiplica o si dividono le mantisse in modo consueto
 - si sommano o si sottraggono gli esponenti
 - si normalizza
 - Esempio: $10,4 * 200 =$
 $0\ 02\ 10400000 * 0\ 03\ 20000000 =$
 $0\ 05\ 20800000$

Virgola mobile

- Somma e sottrazione:
 - si uguagliano gli esponenti
 - le mantisse vengono sommate
 - aggiustamento in caso di traboccamento
 - Esempio: $10,4 + 2 =$
 $0\ 02\ 10400000 + 0\ 01\ 20000000 =$
 $0\ 02\ 10400000 + 0\ 02\ 02000000 =$
 $0\ 02\ 12400000 = 12,4$

Virgola mobile

- Approssimazioni:

- $34,56 + 0,005 =$
 $0\ 02\ 3456 + 0\ 98\ 5000 =$
 $0\ 02\ 3456 + 0\ 02\ 0000 =$
 $0\ 02\ 3456 = 34,56$

- La precisione è data dal numero di cifre della mantissa:
 - Doppia precisione: doppia lunghezza della mantissa (range invariato, precisione raddoppiata)

Memorizzazione su calcolatore e codici

Memorizzazione su calcolatore

- L'unità atomica è il **bit** (Binary Digit)
- L'insieme di 8 bit è detto **byte**
- **Word**: (tipicamente 16, 32 o 64bit): insieme di bit la cui dimensione è una importante caratteristica del calcolatore considerato. Infatti essa influenza:
 - La larghezza degli indirizzi
 - La dimensione dei registri del processore
 - Larghezza dei bus (word o multipli di essa)
- **Double-word**: il doppio di una word

Intervalli di variabilità

- **bit:** Numero di configurazioni: 2
intervallo di variabilità: $[0-1]$
- **byte:** Numero di configurazioni: 256
intervallo di variabilità:
dipende dal tipo di memorizzazione

Tipi di memorizzazione

- Modulo: 256 configurazioni,
[0, 255]
- Modulo e segno: 256 configurazioni,
[-127, +127]
- Complemento a 2: 256 configurazioni,
[-128, +127]

Intervalli di variabilità

- Word (32 bit): $[0, 2^{32}-1]$
- Double-word (64 bit): $[0, 2^{64}-1]$

Codifica dei testi

- Si utilizza una tabella (arbitraria)
- Standard oggi (quasi) universalmente riconosciuto

Codice **ASCII**

American Standard Code for Information Interchange

Codice ASCII

- Si utilizzano **7 bit** quindi **128 simboli diversi**
- ASCII esteso (**8bit**)
 - diverse estensioni in dipendenza dal paese
 - oppure aggiunge la parità

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.LookupTables.com

Codice ASCII - note

- I caratteri alfabetici sono consecutivi e in ordine (alfabetico)
- La distanza tra una lettera minuscola e la corrispondente maiuscola è costante
- Le cifre decimali sono consecutive e in ordine (da 0 a 9)