Architettura di un elaboratore

Il termine informatica

Informazione + automatica

Informazione = dati + istruzioni

Computer ?

- Da un punto di vista logico il computer è un dispositivo che realizza la possibilità di scomporre processi complessi in lunghe sequenze di azioni molto semplici eseguibili in serie

- Il computer esegue queste operazioni e fornisce la risposta che descrive il processo

Computer ?

- Da un punto di vista fisico il computer è un dispositivo costituito da una serie di circuiti elettronici

- Poiché i calcolatori "capiscono" solo due condizioni ossia il passaggio o meno della corrente elettrica, possono essere immaginati come un insieme di interruttori che assumono due stati: "aperto", "chiuso"

- I due stati "aperto", "chiuso" vengono rappresentati con "0" e "1"

Rappresentazione delle informazioni all'interno degli elaboratori

- Il computer elaborano sequenze di "0" e "1". Quindi l'informazione all'interno di un calcolatore è rappresentata mediante sequenze binarie (es. 01101110010010O…)

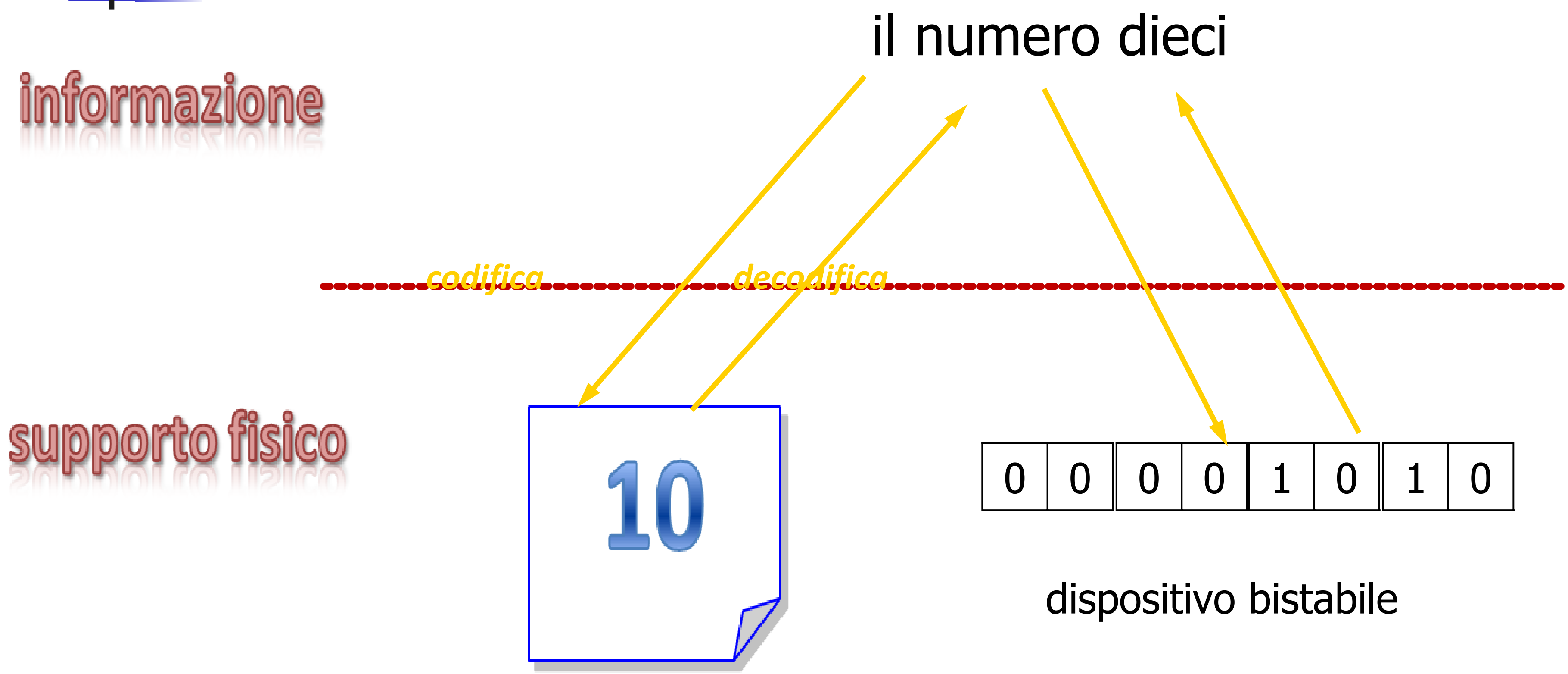
- L'entità minima di informazione all'interno di un elaboratore prende il nome di ***bit*** (*binary digit - cifra binaria*). Un byte è un raggruppamento di 8 bit.

- Per poter far elaborare l'informazione ad un calcolatore occorre codificarla nel linguaggio binario (*digitalizzarla*)

Lo standard IEC per i prefissi binari

Grandezza	Nome	Simbolo		Dimensione	SI	Diff. %
Kilo binario	**Kibi**	**Ki**	2^{10}	**1'024**	10^3	**2.40%**
Mega binario	**Mebi**	**Mi**	$(2^{10})^2$	**1'048'576**	$(10^3)^2$	**4.86%**
Giga binario	**Gibi**	**Gi**	$(2^{10})^3$	**1'073'741'824**	$(10^3)^3$	**7.37%**
Tera binario	**Tebi**	**Ti**	$(2^{10})^4$	**1'099'511'627'776**	$(10^3)^4$	**9.95%**
Peta binario	**Pebi**	**Pi**	$(2^{10})^5$	**1'125'899'906'842'624**	$(10^3)^5$	**12.59%**
Exa binario	**Exbi**	**Ei**	$(2^{10})^6$	**1'152'921'504'606'846'976**	$(10^3)^6$	**15.29%**
Zetta binario	**Zebi**	**Zi**	$(2^{10})^7$	**1'180'591'620'717'411'303'424**	$(10^3)^7$	**18.06%**
Yotta binario	**Yobi**	**Yi**	$(2^{10})^8$	**1'208'925'819'614'629'174'706'176**	$(10^3)^8$	**20.89%**

Codifica e decodifica



Digitalizzazione dei caratteri

- Dobbiamo rappresentare le lettere dell'alfabeto, incluse le cifre numeriche, lettere maiuscole e minuscole, simboli di punteggiatura, parentesi e operatori aritmetici, può essere codificato usando 7 bit ($2^7 = 128$) poi esteso a 8 bit

- Il metodo di codifica più diffuso tra i produttori di hardware e di software prende il nome di codice ASCII (American Standard Code for Information Interchange)

Dati: alfanumerici

Codice ASCII

	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

Digitalizzazione dei caratteri

Sebbene 7 bit siano sufficienti per codificare l'insieme di caratteri di uso comune, in genere il codice ASCII standard utilizza 8 bit, il primo dei quali è sempre 0

Codifica/decodifica della parola cane

01000011	01000001	01001110	01000101
c	a	n	e

…continua…

- ☞26 lettere maiuscole + 26 minuscole ⇒ 52
- ☞10 cifre
- ☞Circa 30 segni d’interpunzione
- ☞Circa 30 caratteri di controllo (EOF, CR, LF, …)

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

ASCII esteso: 8 bit

UNICODE: 16 bit

Passaggio Decimale-Binario

Dato un numero decimale, è possibile passare al corrispondente numero binario tramite una serie di divisioni successive per 2, nelle quali si considerano tutti i resti. Il primo resto ottenuto è il bit meno significativo della codifica binaria

Dati: numerici

Il sistema di numerazione decimale posizionale:

$$1110_{10} = 1\times10^3 + 1\times10^2 + 1\times10 + 0\times10^0$$

Codifica di un numero naturale

Cambio di base:

$$10_{10} = 1\times10 + 0\times10^0 =$$

$$=c_3\times2^3 + c_2\times2^2 + c_1\times2 + c_0\times2^0$$

Il resto della divisione intera per 2 è c_0

Il quoziente della divisione intera per 2 è $c_3\times2^2 + c_2\times2 + c_1\times2^0$

Ci siamo ricondotti ad una situazione analoga alla precedente

…continua…

esempio

10 : 2 = 5 resto 0
5 : 2 = 2 resto 1
2 : 2 = 1 resto 0
1 : 2 = 0 resto 1

1010
Su 1 byte 00001010

…continua…

esempio

18 : 2 = 9	resto 0	
9 : 2 = 4	resto 1	
4 : 2 = 2	resto 0	**10010**
2 : 2 = 1	resto 0	
1 : 2 = 0	resto 1	

…continua…

esempio

137 : 2 =68	resto 1
68 : 2 =34	resto 0
34 : 2 =17	resto 0
17 : 2 = 8	resto 1
8 : 2 =4	resto 0
4 : 2 =2	resto 0
2 : 2 = 1	resto 0
1 : 2 = 0	resto 1

10001001

Passaggio Binario-Decimale

Data una stringa binaria è possibile ottenere il corrispondente numero decimale moltiplicando ogni bit della stringa per la potenza di 2 corrispondente all'indice della cifra considerata e sommando tutti i risultati

Decodifica di un numero naturale

$$
\begin{aligned}
101100_{due} &= 1_{dieci}\times 2^{5}_{dieci} + 0_{dieci}\times 2^{4}_{dieci} + 1_{dieci}\times 2^{3}_{dieci} + 1_{dieci}\times 2^{2}_{dieci} + 0_{dieci}\times 2^{1}_{dieci} + \\
&\quad + 0_{dieci}\times 2^{0}_{dieci} = \\
&= 1_{dieci}\times 32_{dieci} + 0_{dieci}\times 16_{dieci} + 1_{dieci}\times 8_{dieci} + 1_{dieci}\times 4_{dieci} + 0_{dieci}\times 2_{dieci} \\
&\quad + 0_{dieci}\times 1_{dieci} = \\
&= 32_{dieci} + 8_{dieci} + 4_{dieci} = \\
&= 44_{dieci}
\end{aligned}
$$

…continua…

$$
\begin{aligned}
101110101_{due} = \; & 1_{dieci}\times 2^8_{dieci} + 0_{dieci}\times 2^7_{dieci} + 1_{dieci}\times 2^6_{dieci} + 1_{dieci}\times 2^5_{dieci} + 1_{dieci}\times 2^4_{dieci} + \\
& 0_{dieci}\times 2^3_{dieci} + 1_{dieci}\times 2^2_{dieci} + 0_{dieci}\times 2^1_{dieci} + 1_{dieci}\times 2^0_{dieci} = \\
= \; & 1_{dieci}\times 256_{dieci} + 0_{dieci}\times 128_{dieci} + 1_{dieci}\times 64_{dieci} + 1_{dieci}\times 32_{dieci} + \\
& 1_{dieci}\times 16_{dieci} + 0_{dieci}\times 8_{dieci} + 1_{dieci}\times 4_{dieci} + 0_{dieci}\times 2_{dieci} + 1_{dieci}\times 1_{dieci} = \\
= \; & 256_{dieci} + 64_{dieci} + 32_{dieci} + 16_{dieci} + 4_{dieci} + 1_{dieci} = \\
= \; & 373_{dieci}
\end{aligned}
$$

…continua…

Per la codifica dei numeri naturali si utilizzano abitualmente successioni di 32 bit