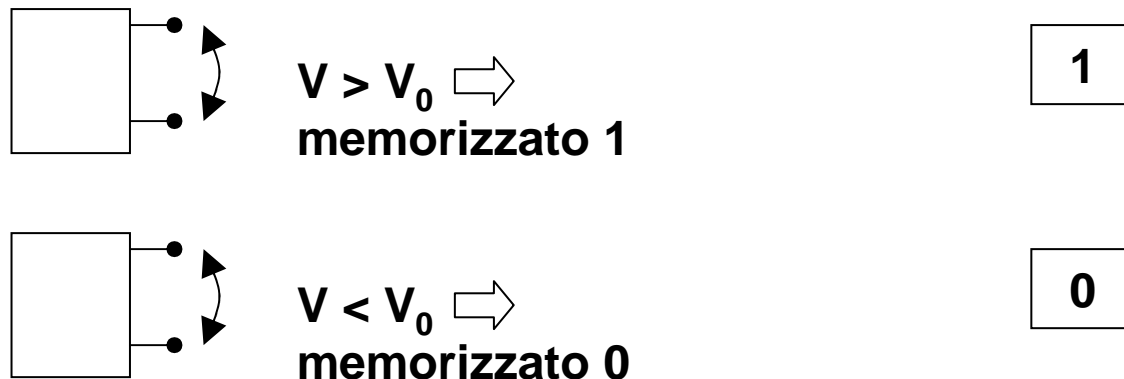


La codifica dei dati e delle istruzioni

La più piccola unità di informazione memorizzabile (e quindi utilizzabile) è il **bit**, che può assumere valore 0 o 1.

Il dispositivo utilizzato per memorizzare un bit è un **elemento bistabile**, cioè un dispositivo elettronico che può assumere uno tra due stati stabili (es. due livelli differenti di tensione), ognuno dei quali viene fatto corrispondere a 0 o a 1 (**cella di memoria**).



Operazioni possibili su una cella di memoria

Operazione di scrittura

La cella di memoria viene caricata con un determinato valore che permane memorizzato finchè:

- la cella viene alimentata elettricamente
- non si esegue un'altra operazione di scrittura che modifica il valore precedentemente memorizzato

Operazione di lettura

Si accede alla cella di memoria per consultarne il valore e copiarlo su un'altra cella di memoria.

Nota

Non su tutte le celle di memoria sono possibili entrambe le operazioni di lettura e scrittura.

Con un solo bit è possibile gestire un'informazione binaria, cioè un'informazione che può specificare uno tra due valori possibili (es. un punto di un'immagine bianco o nero).

Quanti stati possibili può assumere un insieme di bit ?

00	000	0000
01	001	0001
10	010	0010
11	011	0011
	100	0100
	101	0101
	110	0110
	111	0111
		1000
		1001
		1010
		1011
		1100
		1101
		1110
		1111

2 bit → 4 stati

3 bit → 8 stati

4 bit → 16 stati

...

Il registro di memoria

Un insieme di N celle elementari può assumere uno tra 2^N stati possibili.

Un tale insieme è organizzato in un **registro di memoria**.

Il registro costituisce un supporto per la memorizzazione di un'informazione che può assumere uno tra 2^N valori possibili. In particolare un insieme di 8 bit forma un **byte**.

Sul registro sono possibili operazioni di lettura e scrittura che interessano contemporaneamente tutte le celle di memoria contenute nel registro.

Il problema della codifica

Un calcolatore può trattare diversi tipi di dati: numeri (interi, reali), testo, immagini, suoni, ecc. che vanno comunque memorizzati su registri di memoria.

È quindi necessario adottare una **codifica** del tipo di dato considerato: occorre, cioè,

mettere in corrispondenza biunivoca i valori del tipo con gli stati che può assumere il registro.

Esempio

registro da un byte $\Rightarrow 2^8 = 256$ stati possibili.

Che cosa è possibile codificare ?

Numeri naturali [0,255]

0	\leftrightarrow	00000000
1	\leftrightarrow	00000001
....		
255	\leftrightarrow	11111111

Numeri interi [-128,127]

-128	\leftrightarrow	00000000
-127	\leftrightarrow	00000001
0	\leftrightarrow	10000000
+127	\leftrightarrow	11111111

Numeri reali [0,1[

0.0000	\leftrightarrow	00000000
0.0039	\leftrightarrow	00000001
0.0078	\leftrightarrow	00000010
....		
0.9961	\leftrightarrow	11111111

Caratteri

A	\leftrightarrow	01000001
a	\leftrightarrow	01100001
0	\leftrightarrow	00110000
1	\leftrightarrow	00110001

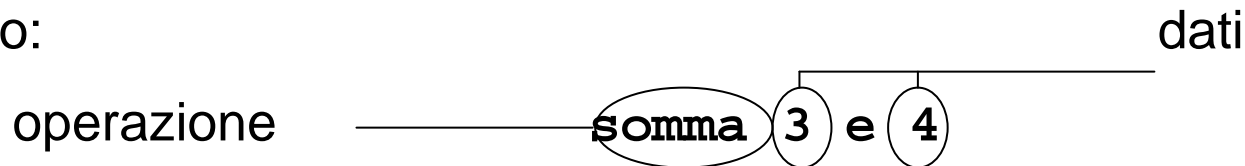
La codifica implica una rappresentazione dei dati limitata e discreta

Codifica delle istruzioni

Oltre ai dati, è necessario memorizzare anche le istruzioni, cioè le singole azioni elementari che l'unità centrale può eseguire.

Nello specificare un'istruzione, bisogna precisare l'operazione da compiere e i dati coinvolti nell'operazione.

Esempio:



Come rappresentare le operazioni ?

L'insieme delle diverse operazioni che l'unità centrale è in grado di eseguire è finito e quindi è possibile codificarlo con un certo numero di bit (**codice operativo**).

somma	0000
sottrai	0001
moltiplica	0010
dividi	0011

Una istruzione sarà quindi rappresentabile da una sequenza di bit divisa in due parti:

- un codice operativo
- un campo operandi (1, 2 o più operandi)

