Il bit e il byte

Dal punto di vista fisico il calcolatore è un insieme di pezzi solidi assemblati in modo opportuno. Tuttavia possiamo già facilmente comprendere che, per poter svolgere dei servizi, il computer debba contenere informazioni. Ovviamente non possiamo usare il nostro linguaggio per "dare vita" a componenti inanimate. E' necessario parlare una lingua alla quale i metalli e i semimetalli da cui sono costituite le parti solide del calcolatore siano sensibili. E questa lingua è l'elettricità. Infatti i metalli possono essere percorsi da corrente ed esiste un particolare dispositivo, chiamato condensatore, in grado di presentare due stati diversi: può essere carico o scarico. Un condensatore piano è un dispositivo costituito da due lamine metalliche piane affacciate l'una all'altra a una certa distanza e tra le quali c'è il vuoto. Un cavo collega una lamina al polo positivo di un generatore di differenza di potenziale, l'altro al polo negativo. In tal modo il generatore inizia a trasportare elettroni da una lamina all'altra stabilendo una differenza di potenziale tra le armature (cioè le lamine) del condensatore.

L'apparecchiatura metallica del calcolatore si basa quindi su questo tipo di dispositivo, legando le informazioni allo stato di carica o di scarica dei condensatori. I due stati possibili (carico/scarico) sono anche interpretabili come: vero/falso, tensione/no tensione, chiuso/aperto, 3,6 V/0 V (che sono i valori usuali della differenza di potenziale tra le armature dei condensatori.

Abbiamo individuato un modo per raccogliere informazioni attraverso i condensatori. Ma un solo condensatore può contenere due sole informazioni: bianco o nero! Una coppia di informazioni di questo tipo, legate cioè al concetto carico/scarico, si dice BIT (BInary digiT). Il bit diviene dunque l'unità di misura delle informazioni del calcolatore.

La base binaria e il codice ASCII

Ora, come possiamo amplificare la capacità di informazioni del calcolatore e realizzare tutte le sfumature che i vari aspetti delle informazioni possono avere? Combinando più bit! Servono infatti più bit per dare al computer le informazioni che facciano funzionare correttamente il mouse e soddisfare il tatto umano. Sono necessari decine o centinaia di bit per i suoni e decine di milioni di bit per soddisfare la vista.

Matematicamente, il concetto di oscillazione tra due sole possibilità proprio del condensatore si esprime usando la base binaria o base 2. Partiamo dalla base 10: in base 10 abbiamo a disposizione dieci cifre (da 0 a 9) che possiamo combinare per formare i numeri superiori al nove stesso. Arrivati a 9 scattano le decine, cioè dobbiamo obbligatoriamente adottare una nuova potenza di dieci per procedere verso i numeri "più grandi". Arrivati a 99 scattano le centinaia. Adottiamo una nuova potenza di dieci.

Il procedimento è lo stesso in base 2, ma in questo caso abbiamo solo due cifre disponibili: zero (0) e uno (1). Se abbiamo zero oggetti usiamo 0, se abbiamo un oggetto usiamo 1. Ma se abbiamo 2 oggetti? Quando avevamo 10 unità in base dieci dovevamo aumentare di una potenza di dieci: mettevamo uno zero al posto del 9 a cui eravamo arrivati in un ipotetico conteggio e anteponevamo ad esso l'1 di riporto. Così nasceva in 10.

Analogamente aumentiamo la potenza di due di una unità. Passiamo da 2^0 a 2^1. Avendo 2 oggetti, trasformiamo l'1 a cui siamo arrivati nell'ipotetico conteggio in uno zero e gli anteponiamo l'1 di riporto. Perciò in base 2 la notazione "10" ha valore decimale 2. 3 in decimale equivale a 11 in binaria. A quattro si sale di un ordine di grandezza, come accade arrivati a 99 in decimale. In

binaria si passa da 2^1 a 2^2: bisogna quindi mettere uno zero in seconda posizione e riportare davanti l'1. 100 in binaria equivale al 4 decimale. E così via.
Se matematicamente è così facile aumentare il numero di oggetti rappresentabili in base binaria e se alla base binaria corrispondono esattamente i bit e i condensatori, altrettanto facile è implementare il numero di informazioni che si possono rappresentare a livello informatico. Basterà aumentare il numero di bit, cioè di condensatori. Dal discorso fatto si evince immediatamente l'equivalenza:

base 2 matematica = bit informatico = condensatore fisico

Per rappresentare un'informazione si usa allora una combinazione di 8 bit, che costituiscono un byte. 8 bit significano 2^8=256 valori diversi, in base decimale i numeri da 0 a 255. Infatti il numero 11111111 binario equivale proprio a 255 decimale:

$$11111111\ B = (1\times2^0 + 1\times2^1 + 1\times2^2 + 1\times2^3 + 1\times2^4 + 1\times2^5 + 1\times2^6 + 1\times2^7)\ D = 255\ D$$

Combinando tra loro due byte possiamo allora rappresentare $2^8\times2^8=2^{16}$=65536 informazioni. Solitamente però, arrivati a 2^{10} bit, il numero 1024 si approssima a 1 Kb = 10^3 bit. Ricorsivamente 2^{10} Kb si approssima a 1 Mb = 10^3 Kb = 10^6 bit, 2^{10} Mb si approssima a 1 Gb = 10^3 Mb = 10^6 kb = 10^9 bit, 2^{10} Gb si approssima a 1 Tb = 10^3 Gb = 10^6 Mb = 10^9 kb = 10^{12} bit.

Il codice ASCII (American Standard Code for Information Interchange) fa corrispondere a un byte di informazioni 256 caratteri alfanumerici differenti. Ad esempio:

- 00110000 è il carattere "0";
- 00110001 è il carattere "1";
- 01100001 è il carattere "a".

Pr rappresentare ciascun carattere servono quindi sette bit di informazione più uno detto di parità che impedisce che l'informazione si degradi durante la trasmissione.

La base esadecimale e gli indirizzi

Vogliamo a questo punto condensare gli otto caratteri che identificano i 255 valori assumibili dal byte. Al tempo stesso desideriamo condensarli usando una rappresentazione che rimandi immediatamente alla rappresentazione binaria. Potremmo usare i numeri da 1 a 255, ma in tal caso perderemmo ogni legame con la base binaria. Usiamo allora la base 16 o esadecimale. Essa è strettamente legata a quella binaria e si adatta perfettamente al byte: 2^4 = 16 e 16^2 = 256.
Come nella base 10 abbiamo dieci caratteri fondamentali (da 0 a 9) per rappresentare tutti i numeri, così ci servono 16 caratteri fondamentali per la base esadecimale. Usiamo i numeri da 1 a 9 per i primi dieci caratteri e le lettere A, B, C, D, E ed F per i caratteri che corrispondono ai numeri decimali 10, 11, 12, 13, 14 e 15.
Ne segue che basteranno solo due caratteri per rappresentare tutti i 256 valori possibili di un byte. Il primo rappresenterà la quantità di valore da 0 a 15 assunta dai primi quattro cifre che troviamo partendo da destra in un byte. Il secondo la quantità di valore da 0 a 15 assunta dal secondo gruppo di quattro cifre che troviamo a sinistra in un byte. Ad esempio prendiamo 10110101:

- dividiamo la stringa in due gruppi da quattro cifre: 1011 e 0101;
- trasformiamo i valori binari in decimale: 1011 = 11 e 0101 = 5;
- trasformiamo i valori binari esadecimale: 11 = B e 5 = 5.
- quindi 10110101 B = B5 H.

Byte, CPU e memoria

Abbiamo detto all'inizio che la memoria contiene dati e istruzioni, mentre la CPU è ciò che è in grado di utilizzare effettivamente il contenuto della memoria. Si rende dunque necessario che la CPU possa accedere alla memoria e viceversa. Esiste un cavo di collegamento tra memoria e CPU detto "Bus".

Per poter analizzare il suo funzionamento è necessario comprendere come sono organizzate le informazioni nella memoria. Le prime memorie potevano contenere $2^{16} \approx 64000$ bit. Esistevano cioè 64000 "cellette", ciascuna contenente una certa informazione. Ciascuna cella deve essere individuata con un nome univoco, cioè diverso da quello di tutte le altre. La soluzione più semplice è quella di assegnare ad ogni cella un numero da 0 a 63999. Si usa però la base esadecimale: la prima cella avrà nome 0000 H = 0 D, la seconda 0001 H = 1 D fino ad arrivare alla penultima FFFE H = 63998 D e all'ultima FFFF = 63999.

Ogni celletta possiede così un indirizzo univoco.

Nel bus servono allora 16 fili per comporre l'indirizzo di una qualsiasi delle 64000 cellette. Ogni celletta contiene poi un'informazione codificata mediante un numero in base binaria il cui valore è compreso tra o e 255. Servono dunque 8 ulteriori fili per catturare un'informazione. L'informazione viene poi portata in uno dei registri della CPU per essere elaborata secondo le istruzioni date dalla memoria. La CPU possiede solitamente quattro o cinque registri, denominati dalle lettere A, B, C, ecc.

Quali tipi di istruzioni possono trovarsi nella memoria? Tre tipi:

- istruzioni di spostamento;
- istruzioni per le operazioni aritmetiche e logiche;
- istruzioni di salto.

A titolo di esempio immaginiamo che nella cella 0000 vi sia l'istruzione di spostamento A7 H che dice "MOV A, 29 H", cioè dice di portare nel registro A l'informazione 29 H che supponiamo contenuta nella cella seguente, la 0001.

Nella cella 0002 i trova invece l'istruzione di spostamento A5 H, che dice "MOV B, 35 H", cioè dice di portare nel registro B l'informazione 35 H che supponiamo contenuta nella cella 0003.

Infine nella cella 0004 immaginiamo vi sia l'operazione aritmetica di somma rappresentata dal valore 32 H che impone "ADD B" cioè di sommare il contenuto del registro A con quello del registro B e di mettere il risultato nel registro A.

Eseguiamo la somma:

29 H =	0	0	1	0	1	0	0	1 B	= 41 D	+
32 H =	0	0	1	1	0	0	1	0 B	= 50 D	=
5B H =	0	1	0	1	1	0	1	1 B	= 91 D	

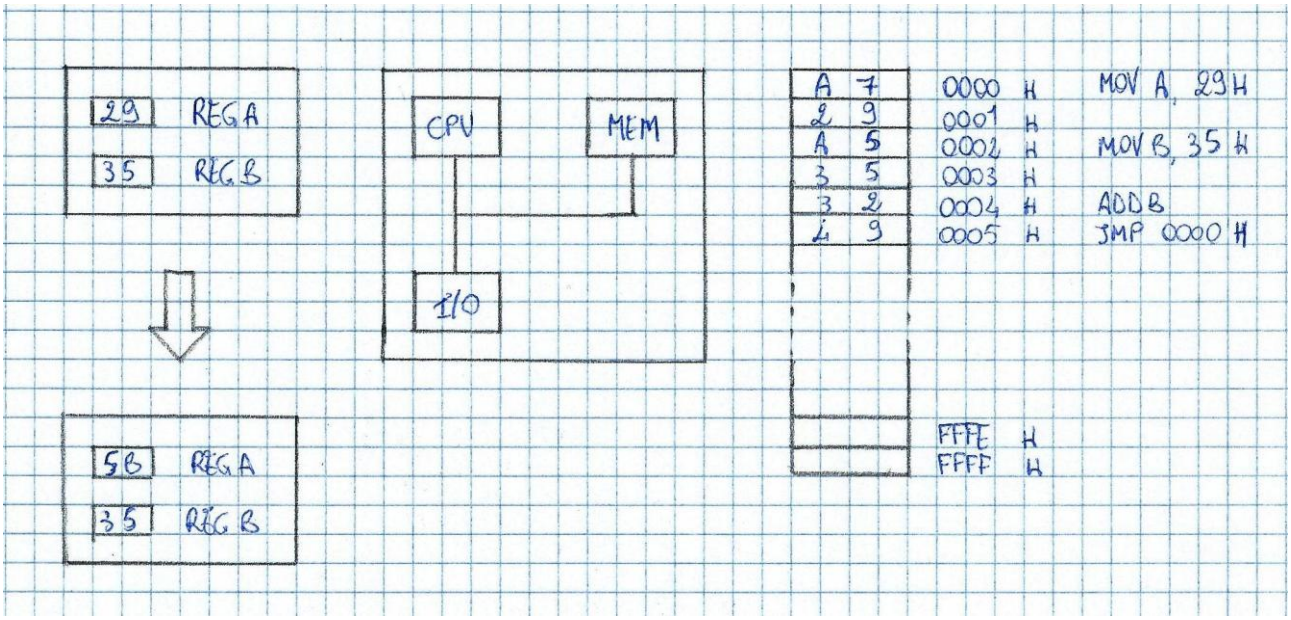
Un'operazione di salto può essere ad esempio "JMP 0000 H" (ad esempio l'informazione 49 H nella cella 0005) che fa ricominciare il ciclo di operazioni a partire dalla cella 0000.

Esistono due tipi di operazioni di salto:

- le operazioni di salto condizionato, in cui si ha il salto solo se è verificata una certa condizione;

- le operazioni di salto incondizionato, in cui il salto avviene comunque, indipendentemente da condizioni.

Ecco uno schema di quanto abbiamo descritto finora:



Potremmo a questo punto fare un'obiezione sull'utilizzo della base binaria: se un computer deve contenere miliardi di informazioni, perché non usare la base dieci, molto più immediata? Per rispondere ipotizziamo di usare la base tre, cioè di assegnare tre valori alla carica del condensatore: totalmente scarico, totalmente carico o carico a metà. In tal caso accade che sia più facile la presenza di errori nelle informazioni contenute. E ciò è facilmente comprensibile, dato che ho tre possibilità anziché due. Infatti i condensatori risento dei campi magnetici esterni e la loro carica è soggetta a variazioni. Ma se il condensatore può assumere solo due valori, che per di più sono massimo e minimo della funzione che descrive la loro carica, è molto meno probabile che ci si sbagli. O è carico o è scarico: "tertium non datur", non abbiamo altra scelta. Se già si crea confusione con la base tre, figuriamoci usare la base dieci!