

Fondamenti di Informatica

Accademia di Belle Arti di Verona

Università degli Studi di Verona

A.A. 2015-2016

Docente - Vincenzo Giannotti

CAPITOLO 4 – CODIFICA DELL'INFORMAZIONE

I Codici Binari

Tra tutti gli argomenti che affronteremo questo è quello un po' più tecnico e forse più complesso. Infatti ci viene chiesto di abbandonare per un attimo il nostro consueto modo di pensare i numeri per poterli rappresentare come vengono «visti» all'interno di una CPU.

Vedremo dunque come diversi elementi di informazione possono essere rappresentati all'interno di un computer affinché questi possa elaborarli.

Noi parleremo soprattutto di «numeri» e di come questi possano essere manipolati. I «numeri» infatti stanno alla base di tutte le tipologie di informazione che normalmente consideriamo siano elaborate dal nostro computer: immagini, testi, suoni etc..

I Codici Binari

Il problema che ci poniamo è il seguente: come possiamo rappresentare gli oggetti che vogliamo codificare, utilizzando i simboli a disposizione del computer?

Ora, il problema della codifica degli oggetti non è certamente oscuro: tutti i giorni noi ci confrontiamo con l'esigenza di codificare dei concetti e degli oggetti, tanto che non ci facciamo nemmeno caso. A ben guardare però tutto viene codificato nella nostra società:

1. i beni di una società hanno un numero di inventario;
2. una scatola di tonno o di piselli ha un suo proprio codice a barre;
3. noi stessi siamo codificati attraverso il nostro codice fiscale.

I Codici Binari

Ora però ci troviamo all'interno di un computer e quindi per poter codificare qualsiasi cosa dobbiamo utilizzare i simboli che abbiamo a disposizione e che sono, come abbiamo visto:

0 e 1

Dunque, per rappresentare un dato oggetto o una data istruzione dovremo utilizzare una opportuna sequenza di 0 e di 1 che ovviamente deve essere univoca per ciascun oggetto.

01001=



01001=



01011=



I Codici Binari

Ma come possono essere combinati tra loro questi BIT per poter rappresentare tutta l'informazione di cui abbiamo bisogno?

Come con l'alfabeto, che ci mette a disposizione 21 caratteri con cui l'uomo è stato capace finora di scrivere tutta la sua storia, così, allo stesso modo, pur con soli due BIT, noi possiamo fare altrettanto.

000 => 1



001 => 2



010 => 3



011 => 4



100 => 5



101 => 6



110 => 7



111 => 8



I Codici Binari

Con tre BIT possiamo dunque codificare otto diversi oggetti. La regola generale è che se abbiamo a disposizione un numero «n» di BIT possiamo rappresentare 2^n oggetti.

BINARIO	DECIMALE	ESADECIMALE	SIMBOLO	NOTE
0000000	0	0	NUL	Non ha alcuna azione
0000001	1	1	SOH	Start Of Heading (primo carattere di un'intestazione di un messaggio)
0000010	2	2	STX	Start Of Text (chiude l'intestazione e precede il testo)
0000011	3	3	ETX	End Of Text (fine testo)
0000100	4	4	EOT	End Of Transmission (fine trasmissione)
0000101	5	5	ENQ	Enquiry (richiesta identificazione di una stazione)
0000110	6	6	ACK	Acknowledge (conferma, è inviato dal ricevitore)
0000111	7	7	BEL	Bell (segnale sonoro, beep)
0001000	8	8	BS	BackSpace (ritorna indietro di uno spazio)
0001001	9	9	HT	Horizontal Tabulation (tabulazione orizzontale)
0001010	10	0A	LF	Line Feed (avanzamento di una linea)
0001011	11	0B	VT	Vertical Tabulation (tabulazione verticale)
0001100	12	0C	FF	Form Feed (avanzamento di una pagina)
0001101	13	0D	CR	Carriage Return (ritorno carrello)
0001110	14	0E	SO	Shift Out (shift disinserito)
0001111	15	0F	SI	Shift In (shift inserito)
0010000	16	10	DLE	Data Link Escape (modifica del collegamento dati)
0010001	17	11	DC1	Device Control 1 (controllo periferica 1)
0010010	18	12	DC2	Device Control 2 (controllo periferica 2)
0010011	19	13	DC3	Device Control 3 (controllo periferica 3)
0010100	20	14	DC4	Device Control 4 (controllo periferica 4)
0010101	21	15	NAK	Negative Acknowledge (conferma negativa, è inviato dal ricevitore)
0010110	22	16	SYN	Synchronous Idle (mantiene il sincronismo tra i computer)
0010111	23	17	ETB	End of Transmission Block (fine della trasmissione di un blocco del messaggio)
0011000	24	18	CAN	Cancel (annullamento)
0011001	25	19	EM	End of Medium (fine supporto)

01100101	101	65	e	Lettera minuscola e
01100110	102	66	f	Lettera minuscola f
01100111	103	67	g	Lettera minuscola g
01101000	104	68	h	Lettera minuscola h
01101001	105	69	i	Lettera minuscola i
01101010	106	6A	j	Lettera minuscola j
01101011	107	6B	k	Lettera minuscola k
01101100	108	6C	l	Lettera minuscola l
01101101	109	6D	m	Lettera minuscola m
01101110	110	6E	n	Lettera minuscola n
01101111	111	6F	o	Lettera minuscola o
01110000	112	70	p	Lettera minuscola p
01110001	113	71	q	Lettera minuscola q
01110010	114	72	r	Lettera minuscola r
01110011	115	73	s	Lettera minuscola s
01110100	116	74	t	Lettera minuscola t
01110101	117	75	u	Lettera minuscola u
01110110	118	76	v	Lettera minuscola v
01110111	119	77	w	Lettera minuscola w
01111000	120	78	x	Lettera minuscola x
01111001	121	79	y	Lettera minuscola y
01111010	122	7A	z	Lettera minuscola z
01111011	123	7B	{	Parentesi graffa aperta
01111100	124	7C		Barra verticale
01111101	125	7D	}	Parentesi graffa chiusa
01111110	126	7E	~	Tilde
01111111	127	7F	DEL	Delete (cancellazione)

**Tabella dei
Caratteri ASCII**

I Codici Binari

Ora noi non siamo interessati all'applicazione di alcun criterio di associazione tra oggetti e sequenze di BIT, ma semplicemente vogliamo capire come trasformare una sequenza di BIT in un numero che siamo in grado di capire «al volo».

Prima di fare questo dobbiamo però fare una premessa su cosa sia un «Sistema di Numerazione».

Migliaia	Centinaia	Decine	Unita'
1000 = M	100 = C	10 = X	1 = I
2000 = MM	200 = CC	20 = XX	2 = II
3000 = MMM	300 = CCC	30 = XXX	3 = III
	400 = CD	40 = XL	4 = IV
	500 = D	50 = L	5 = V
	600 = DC	60 = LX	6 = VI
	700 = DCC	70 = LXX	7 = VII
	800 = DCCC	80 = LXXX	8 = VIII
	900 = CM	90 = XC	9 = IX



I Codici Binari – Sistemi di Numerazione

Un sistema di numerazione è costituito da:

1. Un insieme di «Simboli elementari» (le cifre);
2. Un insieme di «Regole» che definiscono come rappresentare un numero con le cifre a disposizione;
3. Un insieme di «Operazioni» possibili tra i numeri.

Nella nostra vita quotidiana noi siamo abituati ad utilizzare il «Sistema Decimale» e lo facciamo sostanzialmente perché da sempre, per far di conto, si sono utilizzate le dita, che sono dieci.

Il nostro sistema di numerazione però non è l'unico possibile e un esempio alternativo molto noto è quello del sistema di numerazione Romano.

I Codici Binari – Sistemi di Numerazione

I Romani avevano più difficoltà di noi a far di conto perché il loro sistema, che adottava dei simboli e non delle cifre per rappresentare i numeri, era di tipo «non posizionale».

Vediamo con un esempio:

9 => IX in questo caso la I cifra viene sottratta alla II cifra

11 => XI in questo caso la II cifra viene Sommata alla I cifra

Quindi per sapere cosa fare (sommare o sottrarre) noi dobbiamo conoscere come sono disposte le cifre.

La complicazione di questo metodo di rappresentazione è dato dal fatto che alle cifre non viene attribuito un «peso» dato dalla loro posizione.

I Codici Binari – Sistemi di Numerazione

Invece, in un sistema «Posizionale» ogni cifra ha un suo ben preciso peso. Per esempio, nel numero 324 il 3 rappresenta le centinaia, il 2 le decine e il 4 le unità.

Il sistema di numerazione in base 10 è stato introdotto in Europa dagli Arabi e ha determinato un drastico miglioramento delle capacità di fare di conto.

Le regole (l'algoritmo) di moltiplicazione dei numeri romani sono molto più complicate di quelle dei numeri arabi.

I Codici Binari – Sistemi di Numerazione

Per capire esattamente cosa sia un sistema posizionale e come questo poi possa essere esteso anche ai codici binari, diamo queste semplici regole.

Un sistema posizionale è definito da:

- Un numero «b» detto «Base» : 10 nel caso del sistema decimale;
- «b» cifre : 0,1,...9 nel caso del sistema decimale.

A ciascuna cifra viene dato un peso che dipende solo dalla sua posizione nella sequenza. Il valore del nostro Numero è dato da:

cifra in posizione 1 => unità (10^0) + cifra in posizione 2 => decine (10^1) + cifra in posizione 3 => centinaia (10^2) + cifra in posizione 4 => migliaia (10^3) e così via...

$$324 \Rightarrow 3 * 10^2 + 2 * 10^1 + 4 * 10^0$$

I Codici Binari – Sistemi di Numerazione

A questo punto abbiamo gli strumenti per riportare questo stesso ragionamento alla «base 2» che sappiamo essere quella adottata dal computer, potendo contare questo su soli due simboli (cifre). Le regole infatti sono le stesse.

Ci poniamo allora le seguenti domande:

Come possiamo convertire un numero rappresentato in «base 2» in uno rappresentato in «base 10»?

Come possiamo viceversa convertire un numero rappresentato in «base 10» in uno rappresentato in «base 2»?

I Codici Binari – Conversione di base

Prendiamo per esempio il seguente numero in base dieci:

$$108_{10} = ???_2$$

Il procedimento è il seguente: si divide $108/2$ e si mette da parte il resto; si prende il risultato e si ripete la stessa operazione fino alla fine.

$$108/2 = 54 \quad \text{resto} = 0$$

$$54/2 = 27 \quad \text{resto} = 0$$

$$27/2 = 13 \quad \text{resto} = 1$$

$$13/2 = 6 \quad \text{resto} = 1$$

$$6/2 = 3 \quad \text{resto} = 0$$

$$3/2 = 1 \quad \text{resto} = 1$$

$$1/2 = 0 \quad \text{resto} = 1$$

$$108_{10} = 1101100_2$$

I Codici Binari – Conversione di base

Prendiamo per esempio il seguente numero in base due:

1 1 0 1 1 0 0₂

Associamo a ciascuna cifra il suo peso, ricordando che siamo in base due:

$$\begin{array}{ccccccc} \mathbf{1} & \mathbf{1} & \mathbf{0} & \mathbf{1} & \mathbf{1} & \mathbf{0} & \mathbf{0} \\ \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow \\ 2^6 & 2^5 & 2^4 & 2^3 & 2^2 & 2^1 & 2^0 \\ \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow \\ 64 & 32 & 16 & 8 & 4 & 2 & 1 \\ \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow \\ 64 & +32 & +0 & +8 & +4 & +0 & +0 = \mathbf{108}_{10} \end{array}$$

I Codici Binari – La notazione esadecimale

Considerando l'attività interna di un computer vediamo che non viene utilizzato un bit alla volta, ma stringhe di bit che dipendono anche dalla capacità di trasporto dell'informazione.

Quando tra le specifiche del nostro computer leggiamo «sistema a 32 bit» o «sistema a 64 bit», significa che la capacità di trasporto (bus) e di elaborazione (processore) del nostro sistema è di 64 bit alla volta.

È molto difficile però leggere e comprendere velocemente stringhe di bit molto lunghe del tipo 1011010100110001 ed è anche molto facile commettere degli errori.

Per questo si preferisce utilizzare la «notazione esadecimale».

I Codici Binari – La notazione esadecimale

SEQUENZA BIT	RAPPRESENTAZIONE ESADECIMALE
0000	0
0001	1
0010	2
0011	3
0100	4
0101	5
0110	6
0111	7
1000	8
1001	9
1010	A
1011	B
1100	C
1101	D
1110	E
1111	F

Nella notazione esadecimale ad una cifra corrisponde una sequenza di 4 bit. Questo significa che una stringa come quella precedente:

1011010100110001

può essere rappresentata con 4 simboli esadecimali:

B531

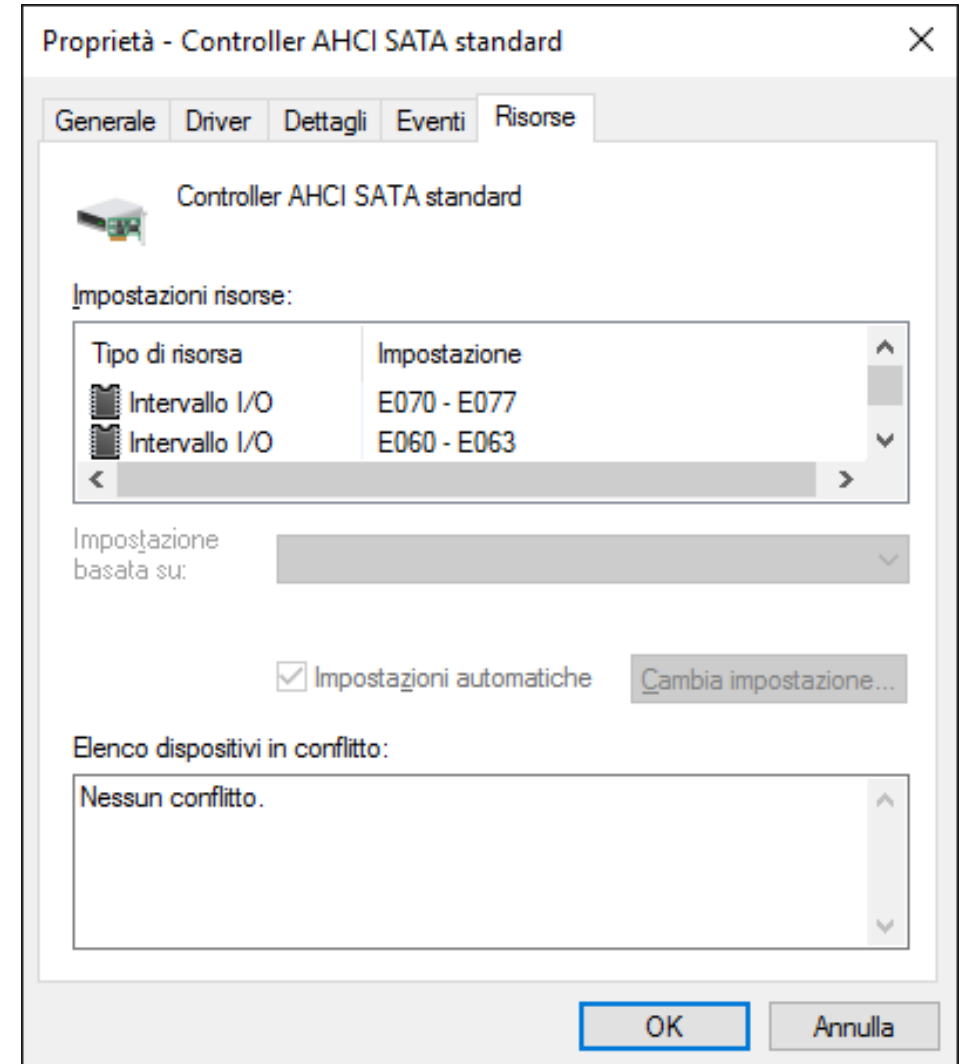
per ottenere questo risultato, la sequenza viene suddivisa in pacchetti di 4 bit e ciascun pacchetto viene convertito nella sua rappresentazione esadecimale:

1011 0101 0011 0001

B 5 3 1

I Codici Binari – La notazione esadecimale

Questa notazione viene utilizzata per rappresentare alcune informazioni di sistema del nostro computer, come per esempio quelle che riguardano le impostazioni dei dispositivi o dei registri.



L'organizzazione della Memoria

Per poter memorizzare le informazioni (dati e programmi) abbiamo visto che un computer può utilizzare diversi dispositivi:

La memoria principale RAM (Random Access Memory) – volatile

La memoria ROM (Read Only Memory) – per memorizzare il programma di avvio del computer

Registri e Cache memories – memorie di supporto alle attività di elaborazione

Memorie di massa (HD, penne, Cdrom etc..) – per memorizzare in maniera permanente grandi quantità di informazioni

La Memoria principale

Nella memoria principale ciascun bit viene memorizzato utilizzando un apposito circuito elettronico miniaturizzato. Tipicamente essa è organizzata in celle in grado di memorizzare 8 bit (un byte).

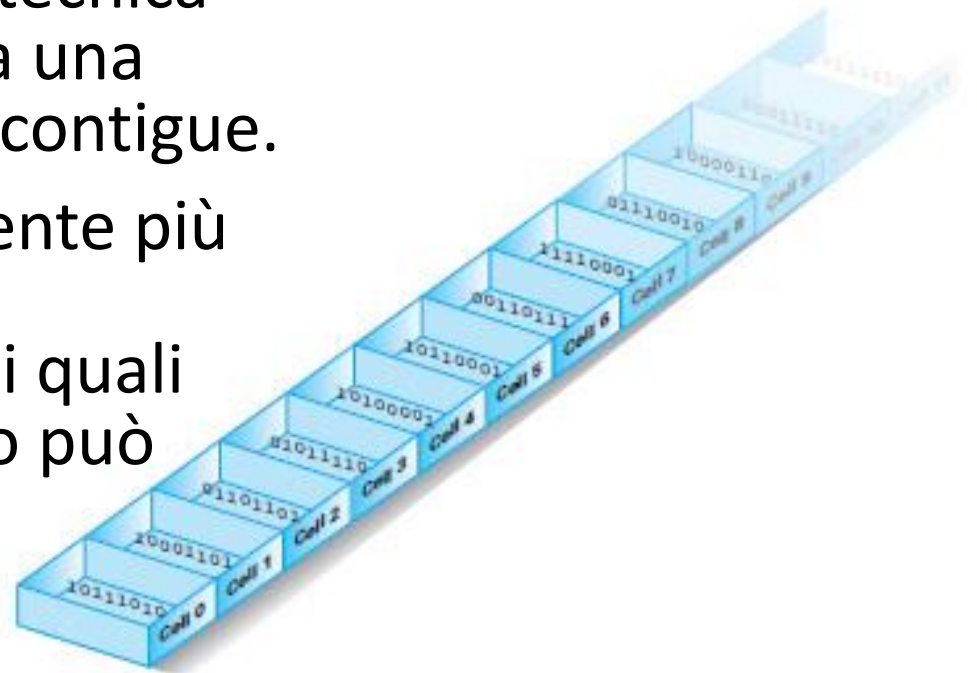
Sebbene il concetto di destra e sinistra non abbia alcun significato all'interno di un circuito elettronico, tuttavia per convenzione l'organizzazione dei bit all'interno della cella viene realizzata in maniera tale che a ciascun bit possa essere attribuito un «peso» cosicché sia possibile calcolare un valore numerico per ciascuna stringa di bit.

$$01101100_2 \Rightarrow 108_{10}$$

La Memoria principale

Ad ogni cella viene quindi attribuito un «indirizzo» numerico attraverso il quale noi siamo in grado di individuare univocamente qualsiasi posizione di memoria. Questa tecnica consente sia di saltare con precisione da una cella a un'altra, sia di muoversi tra celle contigue.

In questo modo non abbiamo teoricamente più limiti quando vogliamo, per esempio, rappresentare numeri molto grandi per i quali non basta un'unica cella (che ricordiamo può rappresentare numeri da 0 a 255).



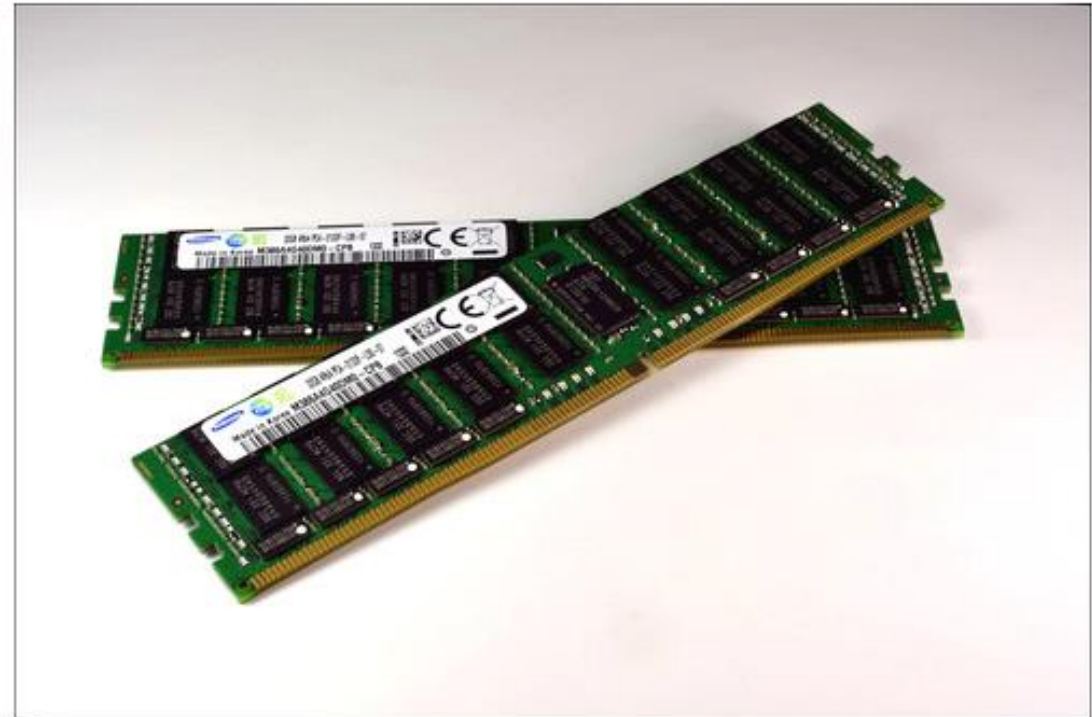
La Memoria principale

Poiché la memoria principale del nostro computer è composta da celle indirizzate e accessibili indipendentemente l'una dall'altra, in maniera casuale, essa viene chiamata «**memoria RAM**» ossia Random Access Memory (memoria ad accesso casuale).

Talvolta si possono trovare altre definizioni per questo tipo di memoria come per esempio:

- DRAM (Dynamic RAM)
- SDRAM (Synchronous DRAM)

Queste si riferiscono a differenti tecnologie utilizzate per realizzare gli elementi di memoria.



Memorie di Massa

A causa del fatto che la memoria principale è volatile ed ha una capacità limitata di memorizzazione (e anche un costo elevato), la maggior parte dei computer utilizza dei dispositivi aggiuntivi per lo *storage* di grandi quantità di dati.

Alcuni di questi dispositivi, a noi ben noti, sono gli Hard Disk, i CD, i DVD, le memorie flash.

Il maggiore svantaggio di questi dispositivi rispetto alle memorie RAM risiede nel fatto che normalmente essi utilizzano degli elementi meccanici (per posizionarsi ad un dato indirizzo) che comporta prestazioni molto inferiori in termini di «tempi di accesso» all'informazione.

Dischi Magnetici

I dischi magnetici (**Hard Disk**) sono da molti anni il dispositivo più utilizzato per memorizzare e leggere, in maniera efficiente, grandi moli di dati. Delle testine poste sopra la superficie magnetica del disco sono in grado di leggere o scrivere dei dati.

Mentre il disco gira velocemente, la testina posta trasversalmente può posizionarsi da una «traccia» all'altra del disco. Normalmente un hard disk è composto da più dischi impilati uno sull'altro, distanziati tra loro per far passare le testine di lettura/scrittura che si muovono all'unisono.

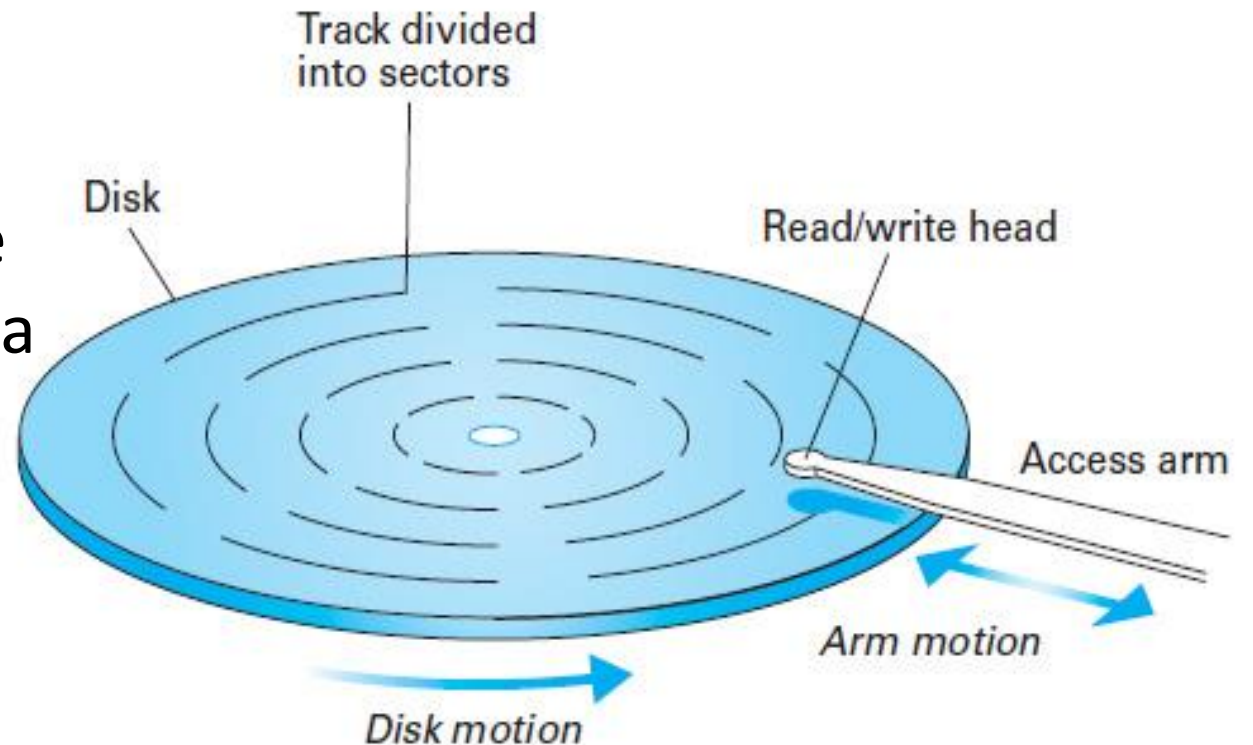


Dischi Magnetici

Ciascuna traccia a sua volta è suddivisa in «settori» dove le informazioni sono memorizzate come sequenze continue di bit (un valore tipico è 512).

Il processo di inizializzazione di un disco, nel quale vengono allocate tracce e settori, è detto «**formattazione**».

Tra i parametri utilizzati per specificare le prestazioni di un disco ci sono il tempo di accesso e la velocità di rotazione: un disco da 7200 rpm (che è un tipico valore dei dischi commerciali) ha un tempo di accesso di circa 9ms.



Dischi Ottici

I dischi ottici (**CD e DVD**) sono realizzati utilizzando un materiale riflettente ricoperto da una superficie protettiva. Le informazioni sono memorizzate creando una variazione sulla superficie riflettente che può essere letta con un laser che rileva le irregolarità sulla superficie del disco mentre questo ruota. I tradizionali CD hanno una capacità di memorizzazione di circa 600-700 MB. Il DVD (Digital Versatile Disk) che è composto di più strati sovrapposti può consentire di memorizzare diversi GB (il laser può venire focalizzato sui diversi strati).

Infine abbiamo il **Blu-ray Disk**; questo può memorizzare fino a 5 volte i dati memorizzabili dal DVD utilizzando dei laser più precisi in luce blu.

Memorie Flash

Sia i dischi magnetici, sia i dischi ottici hanno la limitazione di dover utilizzare degli elementi meccanici per leggere/scrivere i dati e questo riduce notevolmente le prestazioni di questi dispositivi.

La recente tecnologia delle Memorie Flash ha la possibilità di ridurre questo problema.

Nelle Memorie Flash i bit sono memorizzati inviando dei segnali elettrici ad un supporto dove gli elettroni sono intrappolati in piccole camere di diossido di silicio. Questa alterazione può essere mantenuta per diversi anni.

Questo tipo di memoria ha però una limitazione data dal fatto che la ripetuta cancellazione e sovrascrittura di dati causa dei micro-danneggiamenti che ne impediscono l'utilizzazione come memorie principali.



Prossimo Capitolo – Basi di dati e Reti di computer

Nel prossimo capitolo vedremo brevemente cosa è un data base, come funziona una rete di calcolatori e come funzionano Internet e il World Wide Web.