**Processori**

La CPU è il cervello del computer, ed è composta dall’unità di controllo (CU), l’unità aritmetica e logica (ALU) e alcuni registri (piccole memorie ad altissima velocità).

Tra i registri sono molto importanti:  
- Program Counter (PC): punta alla prossima istruzione da prelevare per l’esecuzione;

- Instruction Register (IR): mantiene l’istruzione corrente in fase di esecuzione.

Le componenti di un computer sono collegate attraverso un bus: una collezione di cavi paralleli utilizzati per trasferire indirizzi, dati e segnali di controllo.

**Organizzazione della CPU**

Il percorso dati (data path) di una tipica CPU di von Neumann è composta dai registri (da 1 a 32), dalla ALU e da alcuni bus che connettono fra loro le diverse parti.

I registri alimentano due registri di input della ALU che mantengono i dati di ingresso della ALU mentre questa è occupata nell’esecuzione di alcune computazioni.

La ALU esegue alcune semplici operazioni sui suoi input, come addizioni, sottrazioni e altre semplici operazioni, generando un risultato che viene memorizzato in un suo apposito registro di output.

Questo valore può essere successivamente immagazzinato in uno dei registri della CPU che, volendo, può essere copiato in memoria in un secondo momento.

La maggior parte delle istruzioni può essere divisa in due categorie principali:

- istruzioni registro-memoria (necessita di una fase di caricamento delle parole[[1]](#footnote-1) della memoria nei registri): permettono di prelevare parole di memoria per portarle all’interno dei registri (o viceversa), dove sono utilizzabili, per esempio, come input della ALU per effettuare istruzioni successive.

- istruzioni registro-registro (gli operandi sono già pronti nei registri): preleva due operandi dai registri, li porta all’interno dei registri di input della ALU, esegue su di loro una qualche operazione e ne memorizza il risultato in uno dei registri.

**Esecuzione dell’istruzione**

La CPU esegue ogni istruzione compiendo una serie di passi:

1. prelevare la successiva istruzione dalla memoria per portarla nell’IR;

2. aggiornare il PC per farlo puntare all’istruzione seguente;

3. determinare il tipo dell’istruzione appena prelevata (decodifica dell’istruzione);

4. se l’istruzione usa una parola in memoria, determinare dove si trova;

5. se necessario, prelevare la parola per portarla in un registro della CPU;

6. eseguire l’istruzione;

7. tornare al punto 1 per iniziare l’esecuzione dell’istruzione successiva.

Spesso ci si riferisce a questa sequenza di passi con il termine di ciclo esecutivo delle istruzioni, o ciclo di prelievo-decodifica-esecuzione (fetch-decode-execute).

**Strategie di progettazione delle CPU**

- CISC (Complex Instruction Set Computer): la CPU è in grado di comprendere molte istruzioni complesse nativamente (è il più alto livello di astrazione riconosciuto dalla macchina);

- RISC (Reduced Instruction Set Computer): si basa sull’idea che se le istruzioni sono semplici e poche, esse possono essere eseguite rapidamente (è necessario un solo ciclo nel datapath);

- Ibrido: a partire dal x486, le CPU intel contengono un sottoinsieme di istruzioni RISC (quelle più comuni) che possono essere eseguite in un singolo ciclo nel datapath, mentre le altre complesse sono interpretate secondo la classica modalità CISC.

**Principi di progettazione dei calcolatori moderni**

Esiste un insieme di principi di progettazione, talvolta chiamati principi di progettazione RISC, che i progettisti delle CPU cercano di seguire il più possibile.

**- Tutte le istruzioni devono essere eseguite direttamente dall’hardware:**

Le istruzioni non devono essere interpretate. Per le architettura CISC, quelle istruzioni più complesse (e più rare) possono essere suddivise in parti ed eseguite, successivamente, come sequenze di microistruzioni.

**- Massimizzare la frequenza di emissione delle istruzioni:**

Il parallelismo gioca un ruolo essenziale nelle prestazioni di un computer. Infatti è possibile emettere un gran numero di lente istruzioni in un breve intervallo di tempo solo se si riescono a eseguire più istruzioni allo stesso tempo.

**- Le istruzioni devono essere facili da decodificare:**

Un limite critico sulla frequenza di emissione delle istruzioni è dato dal processo di decodifica, che deve essere effettuato per ogni singola istruzione allo scopo di determinare le risorse necessarie. Bisogna rendere le istruzioni regolari, di lunghezza fissa e con pochi campi. Meno formati di istruzioni ci sono, meglio è.

**- Solo le istruzioni Load e Store fanno riferimento alla memoria:**

L’operazione di spostamento degli operandi dalla memoria ai registri può essere compiuta separatamente mediante apposite istruzioni. Dato che l’accesso alla memoria può richiedere un tempo considerevole, queste operazioni possono essere efficientemente sovrapposte all’esecuzione di altre istruzioni. Tale osservazione porta alla conclusione che soltanto le istruzioni LOAD e STORE dovrebbero far riferimento alla memoria, mentre tutte le altre dovrebbero operare esclusivamente sui registri.

**- Molti registri disponibili:**

Dato che l’accesso alla memoria è relativamente lento occorre prevedere molti registri (almeno 32) di modo che, una volta prelevata la parola, possa essere mantenuta nel registro fintanto sia necessario. È particolarmente inefficiente trovarsi senza registri liberi, in quanto

obbliga a scaricare in memoria tutti i valori dei registri per poi ricaricarli.

**Parallelismo a livello di istruzione**

Poiché l’incremento del clock del processore ha raggiunto un limite fisico, i progettisti di CPU guardano al parallelismo (più istruzioni nello stesso tempo) per incrementare le prestazioni.

Il parallelismo si può ottenere in due diversi modi:  
- Parallelismo a livello di istruzione: il parallelismo è sfruttato all’interno delle istruzioni per ottenere un maggior numero di istruzioni al secondo;

- Parallelismo a livello di processore: più CPU collaborano per risolvere lo stesso problema.

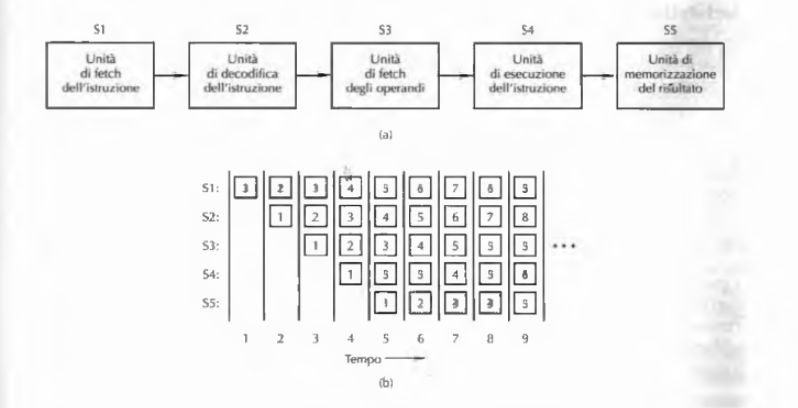
**Pipelining**

Una limitazione nella velocità di esecuzione delle istruzioni è rappresentato dal prelievo delle istruzioni dalla memoria.

Per alleviare questo problema, i calcolatori sono stati dotati della capacità di poter prelevare in anticipo le istruzioni dalla memoria, in modo da averle già a disposizione nel momento in cui dovessero rendersi necessarie. Le istruzioni venivano memorizzate in un insieme di registri chiamati buffer di prefetch, dai quali potevano essere prese nel momento in cui venivano richieste, senza dover attendere che si completasse una lettura della memoria.

In pratica la tecnica di prefetching divide l’esecuzione dell’istruzione in due parti: il prelievo dell’istruzione e la sua esecuzione effettiva.

Il pipeline divide l’esecuzione di un’istruzione in un numero maggiore di parti che possono essere eseguite in parallelo; ciascuna di queste parti è gestita da componenti hardware dedicati.

****

Durante il primo ciclo di clock lo stadio S1 sta lavorando sull’istruzione 1, prelevandola dalla memoria. Durante il ciclo di clock 2 lo stadio S2 decodifica l’istruzione 1, mentre lo stadio S1 preleva l’istruzione 2. Durante il ciclo 3 lo stadio S3 preleva gli operandi per l’istruzione 1, lo stadio S2 decodifica l’istruzione 2 e lo stadio S1 preleva la terza istruzione. Durante il quarto ciclo lo stadio S4 esegue l’istruzione 1, S3 preleva gli operandi per l’istruzione 2, S2 decodifica l’istruzione 3 e S1 preleva l’istruzione 4. Infine, durante l’ultimo ciclo S5 scrive il risultato dell’istruzione 1, mentre gli altri componenti lavorano sulle istruzioni successive.

Il pipelining permette di bilanciare la latenza (quanto dura l’esecuzione di una istruzione) con la banda del processore (quanti MIPS la CPU è in grado di emettere).

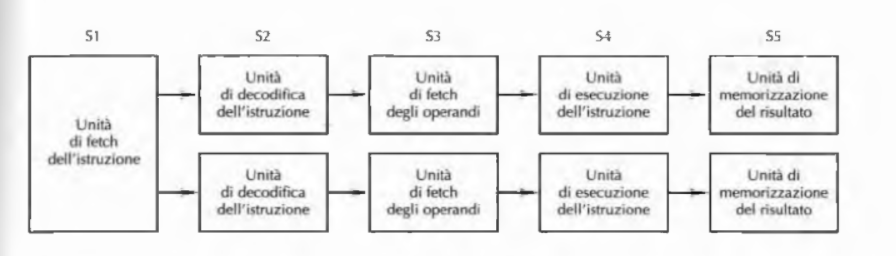
Una CPU, senza pipeline, che opera con clock di T ns ed emette una istruzione per ciclo di clock ha una banda di 103/T MIPS.

Supponendo di disporre di una CPU con un clock di T ns e una pipeline a P stadi (in cui ogni stadio necessita di un ciclo di clock per essere completato), la latenza totale è pari a P **·** T ns mentre la banda è P **·** 103/T MIPS, cioè P volte rispetto alla banda di una CPU senza pipeline con medesima latenza.

**Processori con più pipeline**

Avere due pipeline è sicuramente meglio di averne una sola.

Questa architettura è stata utilizzata inizialmente dall’Intel x486.

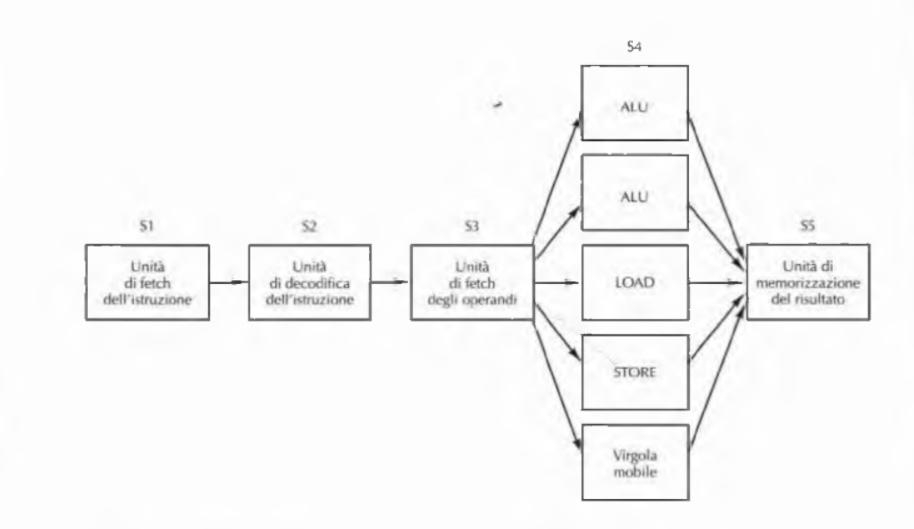


Affinché le due istruzioni possano essere eseguite in parallelo, non devono però esserci conflitti nell’uso delle risorse (cioè i registri) e nessuna delle due istruzioni deve dipendere dal risultato dell’altra.

Inoltre, non tutte le istruzioni possono essere svolte in parallelo (l’input di una istruzione può dipendere dal risultato della precedente) e sarebbero necessarie troppe componenti hardware per le varie unità che andrebbero poi sincronizzate.

**Architetture superscalari**

Viene utilizzata inizialmente da Intel Core. Il processore dispone di una sola pipeline con più unità funzionali in corrispondenza di alcuni stadi.



Affinché l’architettura abbia senso è necessario che la velocità di emissione della fase S3 sia più alta rispetto a quella della fase S4.

La fase S4 può avere delle unità ALU duplicate.

**Parallelismo a livello di processore**

Il parallelismo nel chip aiuta a migliorare le performance della CPU: con il pipelining e le architetture superscalari si può arrivare ad un fattore di miglioramento da 5 a 10.

Però per incrementare drasticamente le performance di un calcolatore occorre progettare sistemi con molte CPU, in questo caso si può arrivare ad ottenere un incremento di 50, 100, o anche più.

Esistono tre differenti approcci: computer con parallelismo sui dati, multiprocessori, multicomputer.

**Computer con parallelismo sui dati**

Ci sono due schemi differenti:

- Processori SIMD (Single Instruction-stream Multiple Data-stream): sono costituiti da un vasto numero di processori identici che eseguono la stessa sequenza di istruzioni su insieme differenti di dati;

- Processori vettoriali: un processore vettoriale esegue la stessa sequenza di operazioni su coppie di dati, ma tutte le addizioni sono svolte da un unico sommatore strutturato in pipeline.

Entrambe le architettura lavorano su array di dati, mentre il primo utilizza tanti sommatori quanti gli elementi del vettore, il secondo utilizza un solo sommatore e un unico registro vettoriale.

**Multiprocessori**

È un’architettura costituita da più CPU che condividono una memoria comune.

Poiché ciascuna CPU può leggere o scrivere qualsiasi zona della memoria comune, le CPU devono sincronizzarsi via software.

In questo caso le CPU hanno la necessità di interagire in modo così profondo che il sistema è detto fortemente accoppiato (tightly coupled).

**Multicomputer**

Multiprocessori con molte CPU sono difficili da realizzare, per via del problema delle connessioni di ciascuna CPU verso la memoria comune.

I progettisti hanno superato il problema abbandonando il concetto di memoria comune e realizzando un elevato numero di CPU interconnesse, ciascuna con la propria memoria privata.

Le CPU in un multicomputer sono accoppiate in modo lasco (loosely coupled) e comunicano attraverso scambi di messaggi.

In architetture grandi la completa interconnessione non è fattibile così sono utilizzate topologie differenti come la griglia, l’albero o l’anello.

**Memoria principale**

La memoria è quella parte del calcolatore in cui sono depositati programmi e dati.

**Bit**

L’unità base della memoria è il bit (BInary digiT): una variabile che assume solo due stati (0 o 1).

Un computer ragiona unicamente interpretando gruppi di bit, cioè comandi rappresentati da sequenze di “zero” e “uno”.

1 byte = 8 bit.

**Indirizzi di memoria**

La memoria è organizzata in celle (o locazioni) identificabili da una posizione (l’indirizzo della cella). La cella rappresenta l’unità più piccola indirizzabile.

In ogni cella c’è lo stesso quantitativo di informazione binaria (espresso in byte).

Se una memoria ha *n* celle, i suoi indirizzi varieranno da *0* a *n-1*.

Se una cella è costituita da *k* bit, essa può contenere una qualsiasi delle *2k*diverse combinazioni di bit.

I byte possono essere raggruppati in parole.

**Ordinamento dei byte**

I byte in una parola possono essere scritti da sinistra a destra (big endian) oppure da destra a sinistra (little endian).

Esempio: il numero 0A1FH può essere scritto come sequenza big endian F1A0H oppure come sequenza little endian così come è scritto.

**Codici correttori**

Occasionalmente le memorie dei calcolatori possono commettere degli errori per via di picchi di tensione sulle linee di alimentazione o per altre cause. Per proteggersi da tali errori, alcune memorie utilizzano dei codici di rilevazione e/o di correzione degli errori. Quando si impiegano questi codici vengono aggiunti dei bit extra a ogni parola di memoria secondo una modalità particolare. Quando una parola viene letta dalla memoria, si controllano questi bit aggiuntivi per vedere se si è verificato un errore.

Supponiamo che una parola di memoria consista di *m* bit di dati ai quali aggiungiamo *r* bit di controllo; sia quindi *n* = *m* + *r* la lunghezza totale. Un’unità di *n* bit contenente *m* bit di dati e *r* bit di controllo è spesso chiamata parola di codice (codeword) a *n* bit.

Date due parole di codice, diciamo 10001001 e 10110001, è possibile determinare il numero di bit corrispondenti rispetto ai quali differiscono. In questo caso la differenza è di 3 bit. Per determinare il numero di bit diversi, bisogna semplicemente calcolare l’OR ESCLUSIVO (XOR) tra i bit delle due parole e contare quanti bit valgono 1 nel risultato; questo valore è chiamato distanza di Hamming. Il suo significato principale è che, se fra due parole di codice vi è una distanza di Hamming pari a *d* allora saranno necessari *d* errori singoli per trasformare una parola nell’altra.

Per esempio le parole di codice 11110001 e 00110000 hanno distanza di Hamming 3 in quanto servono 3 errori singoli per convertire una parola nell’altra.

Con una parola di memoria a *m* bit tutte le 2m combinazioni di bit sono legali, ma, per via del modo in cui sono calcolati i bit di controllo, solo 2m delle 2n parole di codice sono valide. Se una lettura da memoria restituisce una parola di codice non valida, il calcolatore sa che è avvenuto un errore di memoria. Conoscendo l’algoritmo che calcola i bit di controllo è possibile costruire una lista completa delle parole di codice lecite e, da questa lista, trovare le due parole di codice la cui distanza di Hamming è minima; questa distanza è la distanza di Hamming dell’intero codice.

Le proprietà di rilevazione e di correzione degli errori di una parola di codice dipendono dalla sua distanza di Hamming. Per rilevare *d* errori singoli è necessaria una parola con distanza

*d + 1*, poiché con tale codice non esiste alcun modo in cui *d* errori singoli possano cambiare una parola valida in un’altra parola di codice valida. In modo analogo per correggere *d* errori singoli è necessario un codice con distanza *2d + 1*, poiché in questo modo le parole di codice legali sono sufficientemente lontane l’una dall’altra; anche con *d* cambiamenti la parola di codice originaria continua infatti a essere più vicina rispetto a tutte le altre parole di codice, potendola quindi determinare in modo univoco.

Come semplice esempio di codice a correzione di errore consideriamo un codice nel quale al dato si aggiunge un singolo bit di parità, scelto in modo che il numero di bit 1 nella parola di codice sia pari (oppure dispari). Un codice di questo tipo ha distanza 2, dato che ogni errore singolo genera una parola di codice la cui parità è errata. In altre parole servono due errori singoli per passare da una parola di codice valida a un’altra anch’essa valida.

Questo codice può quindi essere utilizzato per rilevare errori singoli; ogniqualvolta viene letta

dalla memoria una parola la cui parità è errata viene segnalata una condizione di errore. II

programma non può continuare, ma almeno non si calcolano alcun risultati errati.

Come semplice esempio consideriamo un codice con solo quattro parole di codice valide:

0000000000, 0000011111, 1111100000 e 1111111111

Questo codice ha una distanza 5, il che significa che può correggere errori doppi. Se arriva

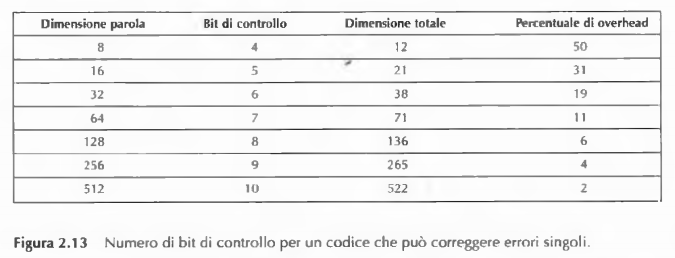
la parola di codice 0000000111 il ricevente sa che l’originale doveva essere 0000011111 (se

gli errori verificatisi erano al massimo doppi). Tuttavia un errore triplo che modifica per esempio la parola 0000000000 nella parola 0000000111 non può essere corretto.

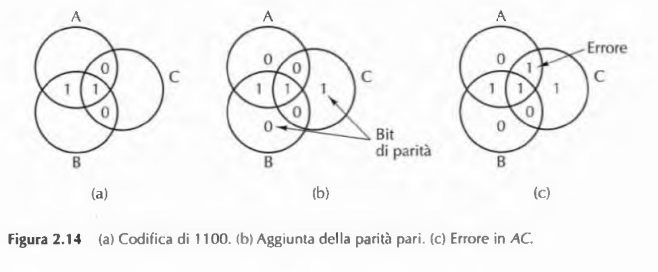
Immaginiamo di voler progettare un codice con *m* bit di dati e *r* bit di controllo, capace di correggere tutti gli errori singoli. Ciascuna delle 2n parole di memoria legali ha *n* parole di codice illegali a distanza 1 da essa. Queste sono formate invertendo sistematicamente ciascuno degli *n* bit nella parola di codice generata dalla parola di memoria considerata. Ciascuna delle 2n parole di memoria legali richiede quindi *n + 1* stringhe di bit a essa dedicate (*n* per i possibili errori e 1 per la combinazione corretta). Dato che il numero totale di combinazioni di

bit è 2n deve valere (n + 1) 2m 2n. Conoscendo *m*, essa fornisce un limite inferiore al numero di bit di controllo richiesti per correggere errori singoli.

La Figura 2.13 mostra il numero di bit di controllo richiesti per parole di memoria di varie lunghezze.

****

È possibile ottenere questo limite teorico utilizzando un metodo ideato da Richard Hamming. Prima di analizzare l’algoritmo di Hamming, guardiamo una semplice rappresentazione grafica che illustra chiaramente l’idea di un codice a correzione di errore per parole a 4 bit. Il diagramma di Venn della Figura 2.14(a) contiene tre cerchi. A, B e C, che insieme formano sette regioni. Come esempio codifichiamo la parola di memoria a 4 bit 1100 nelle regioni AB, ABC, AC e BC inserendo un bit in ogni regione (in ordine alfabetico). Questa codifica è mostrata nella Figura 2.14(a).



Aggiungiamo poi un bit di parità a ciascuna delle tre regioni vuote per ottenere al loro interno una parità pari, come illustrato nella Figura 2.14(b).

Per costruzione la somma dei bit in ciascuno dei tre cerchi, A, B e C, è ora un numero pari. Nel cerchio A abbiamo i quattro numeri 0, 0, 1 e 1, la cui somma fa 2, un numero pari. Nel cerchio B i numeri sono 1, 1, 0 e 0, che sommati fanno ancora 2, un numero pari. Infine nel cerchio C abbiamo nuovamente la stessa situazione. In questo esempio tutti i cerchi hanno la stessa somma, ma, in altre situazioni, è possibile ottenere somme diverse, come 0 e 4. Questa figura corrisponde a una parola di codice con 4 bit di dati e 3 bit di parità.

Supponiamo ora che il bit nella regione AC diventi errato, passando dal valore 0 al valore 1, come mostrato nella Figura 2.14(c). A questo punto il calcolatore può vedere che i cerchi A e C hanno la parità errata (dispari). L’unico cambiamento di un singolo bit che li può correggere corrisponde a riportare AC al valore 0; questa modifica corregge correttamente l’errore. Seguendo questa strategia il calcolatore può riparare automaticamente gli errori di memoria che coinvolgono un solo bit.

Vediamo ora come l’algoritmo di Hamming può essere usato per costruire codici a correzione di errore per parole di memoria dalla dimensione arbitraria. In un codice di Hamming gli *r* bit di parità sono aggiunti a una parola a *m* bit, formando una nuova parola di lunghezza *m + r* bit. I bit sono numerati a partire da 1, non 0, con il bit 1 nella posizione più a sinistra (più significativa). Tutti i bit la cui posizione è una potenza di 2 sono bit di parità; quelli restanti sono usati invece per i dati. Per esempio, con una parola a 16 bit, vengono aggiunti 5 bit di parità nelle posizioni 1, 2, 4, 8 e 16, mentre le altre contengono bit di dati.

In totale la parola di memoria diventa di 21 bit (16 di dati, 5 di parità). In questo esempio useremo (arbitrariamente) la parità pari.

Ciascun bit di parità controlla alcune posizioni dei bit specifiche ed è impostato in modo che sia pari il numero totale di bit che hanno valore 1 nelle posizioni controllate. Ogni bit di parità controlla le seguenti posizioni:

il bit 1 controlla i bit 1, 3, 5, 7, 9, 11, 13, 15, 17, 19, 21

il bit 2 controlla i bit 2, 3, 6, 7, 10, 11, 14, 15, 18, 19

il bit 4 controlla i bit 4, 5, 6, 7, 12, 13, 14, 15, 20, 21

il bit 8 controlla i bit 8, 9, 10, 11, 12, 13, 14, 15

il bit 16 controlla i bit 16, 17, 18, 19, 20, 21.

In generale il bit di posto *b* è controllato dai bit b1, b2, … bn tali che b1 + b2 + ... + bn = b.

Per esempio il bit 5 è controllato dai bit 1 e 4 dato che 1 + 4 = 5. Il bit 6 è controllato dai bit 2 e 4 in quanto 2 + 4 = 6, e così via.

La Figura 2.15 mostra la correzione di un codice di Hamming per la parola di memoria a 16 bit 1111000010101110. La parola di codice a 21 bit corrispondente è 001011100000101101110. Per vedere come funziona la correzione di errore consideriamo che cosa accadrebbe se il bit 5 fosse invertito da un sovraccarico di tensione sulla linea di alimentazione. La nuova parola di codice diventerebbe 001001100000101101110 (invece di 001011100000101101110). Controllando i 5 bit di parità si otterrebbero i seguenti risultati:

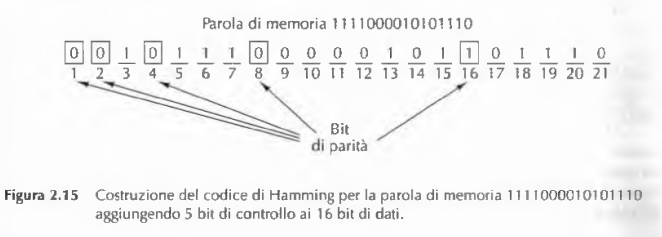
bit di parità 1 non corretto (1, 3, 5, 7, 9, 11, 13, 15, 17, 19, 21 contengono cinque 1)

bit di parità 2 corretto (2, 3, 6, 7, 10, 11, 14, 15, 18, 19 contengono sei 1)

bit di parità 4 non corretto (4, 5, 6, 7, 12, 13, 14, 15, 20, 21 contengono cinque 1)

bit di parità 8 corretto (8, 9, 10, 11, 12, 13, 14, 15 contengono due 1)

bit di parità 16 corretto (16, 17, 18, 19, 20, 21 contengono quattro 1).



Il numero totale di 1 nelle posizioni 1, 3, 5, 7, 9, 11, 13, 15, 17, 19 e 21 dovrebbe essere un numero pari dato che si sta usando la parità pari. Il bit non corretto deve essere uno dei bit controllati dal bit di parità 1, cioè uno fra i bit 1, 3, 5, 7, 9, 11, 13, 15, 17, 19 e 21.

Il bit di parità 4 è errato, il che significa che uno fra i bit 4, 5, 6, 7, 12, 13, 14, 15, 20 e 21 non è corretto. L’errore deve essere uno dei bit presenti in entrambe le liste, cioè il bit 5, 7, 13, 15 oppure 21. Tuttavia il bit 2 è corretto e quindi i bit 7 e 15 sono eliminati. Analogamente il bit 8 è corretto, eliminando così il bit 13. Infine, dato che anche il bit 16 è corretto, va eliminato pure il 21. Il solo bit rimasto è il bit 5, che è quindi quello in cui si è verificato l’errore; dato che è stato letto come 1, esso deve assumere il valore 0. Questa successione di considerazioni consente di correggere gli errori.

Un semplice metodo per trovare i bit errati consiste nel calcolare inizialmente tutti i bit di parità. Se sono tutti corretti allora non si è verificato alcun errore (oppure più di uno).

Successivamente si sommano tutti i bit di parità errati, contando 1 per il bit 1, 2 per il bit 2, 4 per il bit 4, e così via, e la somma risultante corrisponde alla posizione del bit errato.

Se per esempio i bit di parità 1 e 4 sono errati, ma 2, 8 e 16 sono corretti, significa che il

bit 5 (1 + 4) è stato invertito.

**Memoria cache**

Le CPU sono sempre state più veloci delle memorie. Questa differenza di prestazioni si nota quando la CPU lancia una richiesta alla memoria, ma non otterrà la parola desiderata se non dopo molti cicli di CPU. Più lenta è la memoria, più cicli dovrà attendere la CPU .

Ci sono delle tecniche interessanti che permettono di combinare una piccola quantità di memoria veloce con una grande quantità di memoria lenta al fine di ottenere sia la velocità della memoria veloce (quasi) sia la capacità della memoria più grande. La piccola e veloce memoria è chiamata cache.

L’idea di base è semplice: le parole di memoria usate più di frequente sono mantenute all’interno della cache. Quando la CPU necessita di una parola, la cerca nella cache e, solo nel caso in cui essa non sia presente, la richiede alla memoria centrale. È possibile ridurre drasticamente il tempo medio di accesso se una frazione significativa delle parole è presente nella cache.

Il successo o il fallimento dipendono quindi da quali parole sono presenti nella cache.

Da anni si sa che i programmi non accedono alle loro memorie in modo completamente casuale.

Se ad un certo istante la memoria fa un riferimento all’indirizzo A è molto probabile che il successivo riferimento alla memoria si troverà nelle vicinanze di A (principio di località spaziale).

Analogamente, nell’arco temporale di esecuzione del programma, si accede molto spesso alle stesse zone del programma (principio di località temporale).

L’idea generale, quindi, prevede che quando una parola viene referenziata, la parola stessa e alcune parole vicine sono portate dalla grande e lenta memoria all’interno della cache, in modo che sia possibile accedervi velocemente in un secondo momento.

Per disporre sistemi di memoria più sofisticati, è possibile avere anche tre o più livelli di cache.

Il chip stesso della CPU contiene una piccola cache (L1) per le istruzioni e una piccola cache per i dati (L1), le cui dimensioni sono comprese tra i 16 e i 64 KB.

Dopo questa piccola memoria c’è una cache di secondo livello (L2), che non si trova nel chip del processore, ma che può essere inclusa all’interno del suo involucro; questa cache, di solito è unificata (contiene sia dati che istruzioni) e la sua dimensione può variare dai 512 KB a 1 MB.

La cache di terzo livello (L3) si trova sulla scheda del processore ed è costituita da alcuni MB.

Di solito le cache sono annidate, nel senso che l’intero contenuto della cache di primo livello è compreso nella cache di secondo livello e tutto il contenuto di quest’ultima è compreso in quella di terzo livello.

**Memoria secondaria**

**Gerarchie di memoria**

La soluzione che viene tradizionalmente adottata per memorizzare una gran mole di dati consiste nell’organizzare gerarchicamente la memoria.



Nella parte alta della gerarchia si trovano i registri della CPU, ai quali si può accedere alla stessa velocità della CPU.

Successivamente vi è la memoria cache, la cui dimensione può variare da 32 KB fino ad alcuni megabyte.

La memoria centrale è il passo successivo e la sua dimensione è compresa tra 16 MB per i sistemi più economici fino a decine di gigabyte per quelli professionali.

Successivamente troviamo i dischi magnetici, la vera forza lavoro per quanto riguarda la memorizzazione permanente.

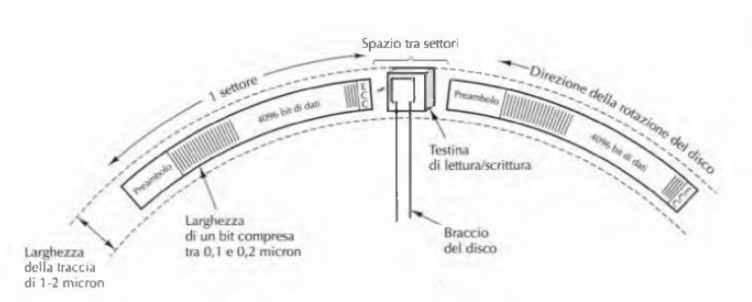
Infine ci sono i nastri magnetici e i dischi ottici utilizzati per l’archiviazione.

Muovendosi verso il basso della gerarchia aumentano tempo di accesso e capacità di memorizzazione ma i prezzi diminuiscono.

**Dischi magnetici**

Un disco magnetico consiste di uno o più piatti di alluminio rivestiti di materiale magnetico.

La testina del disco, contenente un solenoide, sfiora la superficie rimanendo sospesa su un cuscinetto d’aria. Una corrente che passa attraverso la testina magnetizza la superficie che si trova al di sotto, orientando le particelle magnetiche in direzione opposta a seconda del verso della corrente. Quando la testina passa sopra un’area magnetizzata, viene indotta nella testina una corrente positiva o negativa rendendo così possibile la lettura dei bit memorizzati. In questo modo, mentre il piatto ruota sotto la testina, è possibile scrivere e leggere un flusso di bit.



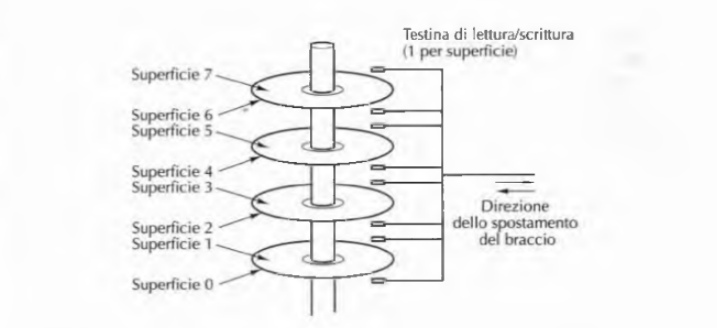
La sequenza circolare di bit scritti mentre il disco compie una rotazione completa è chiamata traccia.

Le tracce sono divise in un certo numero di settori di lunghezza fissa, contenenti in genere 512 byte di dati e preceduti da un preambolo che permette alla testina di sincronizzarsi prima della lettura o della scrittura. Dopo i dati segue un codice per la correzione di errore (ECC, Hamming o Reed-Solomon).

L’insieme di tracce alla stessa distanza dal centro è chiamato cilindro.

Tra due settori consecutivi vi è un piccola area chiamata spazio tra settori.

La maggior parte dei dischi consiste di più piatti impilati verticalmente in cui ciascuna superficie ha il proprio braccio e la propria testina. I dischi dei PC attuali hanno dai 6 ai 12 piatti, mettendo quindi a disposizione dalle 12 alle 24 superfici registrabili.



Le prestazioni dei dischi dipendono da un insieme di fattori. Per leggere o scrivere un settore il braccio deve inizialmente effettuare quella che viene chiamata seek (ricerca), cioè lo spostamento radiale sulla posizione corretta. Una volta che la testina si è posizionata radialmente c’è un ritardo, chiamato latenza rotazionale, nell’attesa che il settore desiderato ruoti sotto la testina. Il tempo di trasferimento dipende invece dalla densità lineare e dalla velocità rotazionale.

A ogni disco è associato un processore dedicato chiamato controllore del disco, il quale deve accertare i comandi dal software.

**Dischi IDE**

I dischi dei moderni PC sono un’evoluzione di un disco da 10 MB della Seagate gestito da un controllore collocato su una scheda aggiuntiva. Il sistema operativo leggeva dal disco e scriveva su di esso inserendo alcuni parametri nei registri della CPU e chiamando successivamente il BIOS (Basic Input Output System, “sistema elementare di input/output”). Il BIOS lanciava le istruzioni macchina necessarie per caricare i registri del controllore, che iniziava il trasferimento dei dati.

La tecnologia evolse rapidamente e si passò alle prime unità IDE (Integrated Drive Electronics, “memoria di massa con elettronica integrata”) in cui il controllo era strettamente integrato con l’unità.

Tuttavia, per ragioni di retrocompatibilità, le chiamate al BIOS non furono modificate una simile unità poteva avere una capacità massima di 504 MB.

Le unità IDE evolsero successivamente nelle unità EIDE (“Extended IDE”), che supportavano anche un secondo schema di indirizzamento chiamato LBA (“indirizzamento logico dei blocchi”). Ciò permette di superare il limite di 504 MB.

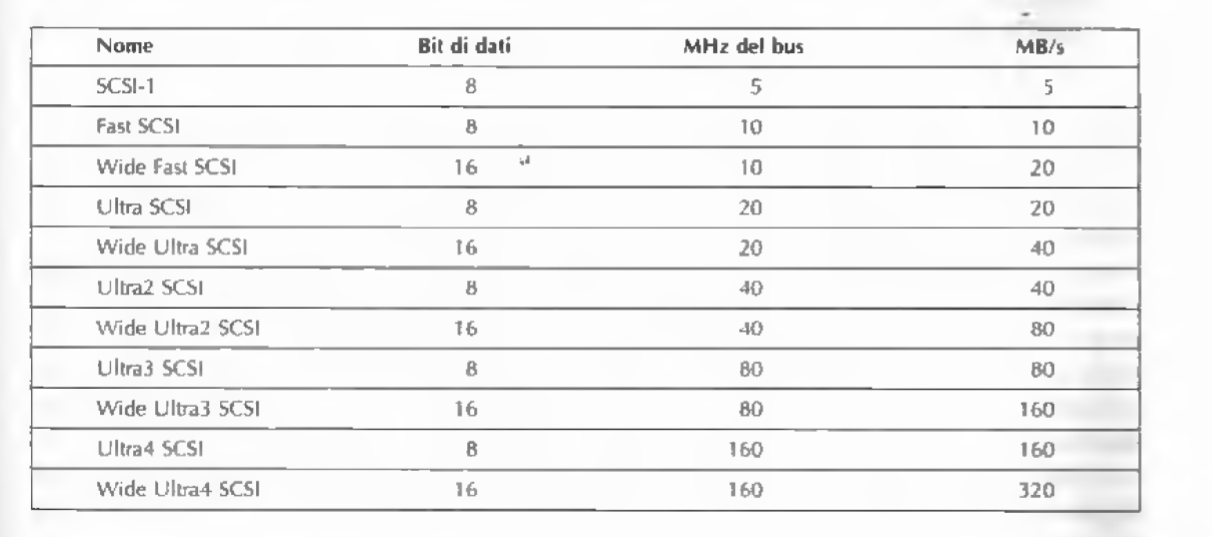
Il suo successore fu chiamato ATA-3 e ancora il successore chiamato ATAPI-4 con velocità aumentata a 33 MB/s (ATAPI-5 -> 66 MB/s).

ATAPI-7 rappresenta una rottura con il passato in quanto usa, a differenza dei precedenti, un’interfaccia chiama serial ATA per trasferire 150 MB/s.

**Dischi SCSI**

Per quanto riguarda l’organizzazione di cilindri, tracce e settori, i dischi SCSI non sono differenti da quelli IDE, ma hanno una diversa interfaccia e una velocità di trasferimento dati molto più elevata.

Nel corso degli anni sono state sviluppate varie versioni, aumentando man mano in velocità (Fast, Ultra, Ultra2, Ultra3, Ultra4, ciascuna di esse aveva una versione Wide).



Per via della velocità di trasferimento più elevata, i dischi SCSI sono lo standard dei PC Intel di fascia alta, soprattutto nel caso dei server di rete.

SCSI non è soltanto un’interfaccia per hard disk, ma è anche un bus al quale possono essere collegati un controllore e sette dispositivi. Questi possono comprendere uno o più hard disk SCSI, CD-ROM, masterizzatori, scanner, unità a nastro e altre periferiche SCSI.

**RAID**

La differenza tra le prestazioni della CPU e quella dei dischi è aumentata con il tempo in modo considerevole.

Come già visto, spesso si utilizza l’elaborazione parallela per velocizzare le prestazioni delle CPU. Quindi anche l’I/O parallelo è sembrato ad alcuni una buona idea.

Questa idea fu adottata velocemente dal mercato e portò a una nuova classe di dispositivi di I/O chiamata RAID (Redundant Array of Inexpensive Disks, “insieme ridondante di dischi economici”).

L’idea su cui si basa RAID è quella di installare vicino al calcolatore un contenitore pieno di dischi, di sostituire la scheda contenente il controllore del disco con un controllore RAID, di copiare i dati sul RAID e di continuare quindi le normali operazioni.

In altre parole un RAID dovrebbe apparire agli occhi del sistema operativo come uno SLED (Single Large Expensive Disk, “singolo disco capiente e costoso”), seppur con prestazioni e affidabilità più elevate.

Tutti i RAID, oltre ad apparire al software come un singolo disco, hanno la proprietà di distribuire i dati sulle diverse unità per permetterne la gestione parallela.

Esistono diversi schemi RAID, conosciuti con nomi che vanno da RAID livello 0 a RAID livello 5 (6 diverse possibili organizzazioni).

Il RAID livello 0: Secondo questa organizzazione il disco virtuale simulato dal RAID è visto come se ognuno dei k settori fosse diviso in strip (“strisce”), con i settori da 0 a k - 1 che compongono la strip 0, i settori da k a 2k - 1 la strip 1 e così via; se k = 1 ogni strip è un settore, se k = 2 una strip è composta da due settori, ecc. L’organizzazione RAID livello 0 scrive sulle strip consecutive in modo ciclico (modalità round robin), come mostra la Figura 2.23(a) nel caso di un RAID con quattro unità. Il RAID livello 0 lavora meglio quando le richieste sono di grandi dimensioni.

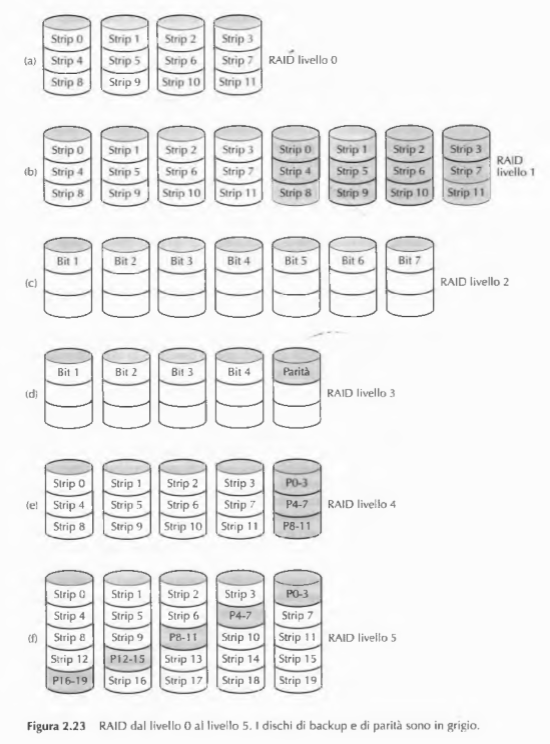
Il RAID livello 1, mostrato nella Figura 2.23(b), duplica tutti i dischi, risultando così composto da quattro dischi primari e da quattro di backup. Nel caso di una scrittura ogni strip viene scritta due volte, mentre nel caso di una lettura è possibile usare entrambe le copie, distribuendo il carico di lavoro su più unità. Ne consegue che le prestazioni in scrittura non sono migliori di quelle che si otterrebbero con una singola unità, mentre quelle in lettura possono essere fino a due volte migliori. La tolleranza ai guasti è eccellente: se un disco si rompe, si può semplicemente utilizzare la copia.

Diversamente dai livelli 0 e 1 che lavorano con strip di settori, il RAID livello 2 funziona sulla base di una parola o, in alcuni casi, anche sulla base di un byte. Questa organizzazione presenta però alcuni difetti: la rotazione dei dischi deve essere sincronizzata e lo schema ha senso soltanto se si utilizza un numero significativo di unità (anche con 32 dischi di dati e 6 dischi di parità l’overhead è del 19%).

Il RAID livello 3, mostrato nella Figura 2.23(d), è una versione semplificata del RAID livello 2. Il bit di parità viene calcolato per ogni parola di dati e poi scritto su un apposito disco. Dato che le parole di dati sono distribuite su più unità, anche in questo caso, come per il RAID livello 2, i dischi devono essere sincronizzati.

Il RAID livello 4, illustrato nella Figura 2.23(e), è come il RAID livello 0, con una parità strip-per-strip scritta su un disco aggiuntivo. Se un disco si guasta è possibile ricalcolare i byte persi grazie al disco di parità. Questo schema protegge dalla perdita di un disco, ma ha prestazioni scarse quando si aggiornano piccole quantità di dati. II disco di parità può inoltre diventare un collo di bottiglia, a causa del grande carico di lavoro che pesa su di esso.

Il RAID livello 5 elimina questo collo di bottiglia distribuendo uniformemente i bit di parità su tutti i dischi, in modalità round robin, com’è mostrato nella Figura 2.23(f). Tuttavia, quando si verifica un guasto a un disco, la ricostruzione del suo contenuto è un processo complesso.

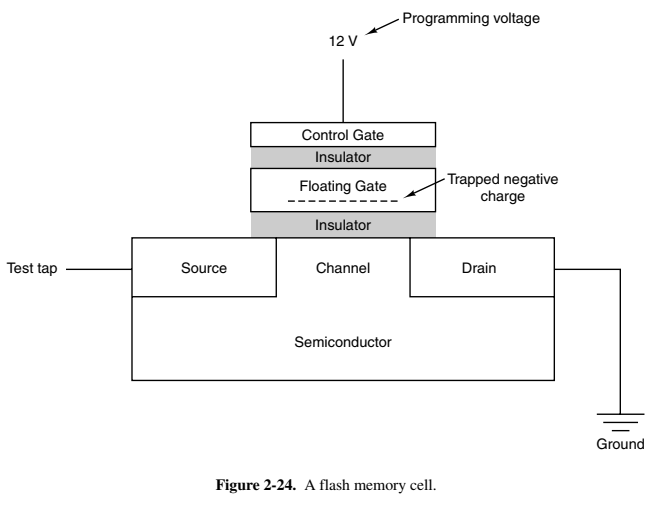


**Dischi a stato solido (SSD)**

Sono dischi basati su memoria flash non volatile e si sono diffusi come alternativa ad altà velocità ai tradizionali dischi magnetici.

I dischi flash sono fatti di celle di memoria flash a stato solido. Queste celle sono costituite da un singolo transistor flash speciale.

La figura 2.24 mostra una cella di memoria flash. Per programmare il bit flash, si applica alla porta di controllo una tensione elevata che accelera il processo di iniezione a caldo nella porta flottante. Gli elettroni vengono intrappolati nella porta flottante, portando così una carica negativa interna al transistor flash. La carica negativa aumenta la tensione necessaria ad accendere il transistor flash e, testando se il canale si accende con una tensione alta o bassa, è possibile determinare se porta floattante è carica oppure no, con conseguente valore di 0 o 1 della cella flash.



Poiché gli SSD sono essenzialmente memorie, hanno prestazioni superiori ai dischi a rotazione e il tempo di ricerca risulta pari a zero. Un SSD può operare due o tre volte più velocemente.

Il rovescio della medaglia è il maggiore costo e una maggiore percentuale di fallimento. Una tipica cella flash può essere scritta solo 100.000 volte prima di smettere definitivamente di funzionare.

**CD-ROM**

I dischi ottici, grazie alla loro grande capacità e al basso costo, sono ampiamente utilizzati per distribuire software, libri, film e dati di tutti i tipi, nonché per creare copie di backup degli hard disk.

Nel 1980 la Philips sviluppò insieme alla Sony il CD (Compact Disc), che sostituì rapidamente il disco in vinile per la registrazione della musica.

La preparazione di un CD avviene per mezzo di un laser infrarosso ad alta potenza che crea sulla superficie fotosensibile di un disco di vetro (glass master) dei buchi dal diametro di 0,8 micron.

A partire da questo master viene creato uno stampo che presenta rilievi in corrispondenza delle scanalature prodotte dal laser e all’interno del quale si inserisce del policarbonato liquido. Il risultato è la creazione di un CD con le scanalature disposte esattamente come quelle del glass master. Successivamente si deposita sul policarbonato un sottile strato di alluminio riflettente, lo si ricopre con una vernice protettiva e infine con un’etichetta.

Le scanalature nel sottostrato di policarbonato sono chiamate pit, mentre le aree non incise tra i pit sono chiamate land.

Per la lettura, un diodo laser a bassa potenza invia sui pit e sui land luce infrarossa. Dato che il laser si trova sul lato del policarbonato i pit sporgono nella sua direzione come rilievi sulla superficie altrimenti piatta. Dato che i pit hanno un’altezza pari a un quarto della lunghezza d’onda della luce del laser, la luce riflessa da un pit viene sfasata di mezza lunghezza d’onda rispetto alla luce riflessa sulla superficie circostante. Il risultato è che le due parti interferiscono in modo distruttivo e restituiscono al fotorilevatore del lettore meno luce di quanta viene riflessa in corrispondenza dei land. In questo modo il lettore può distinguere un pit da un land. Si codificano i valori 1 e 0 come la presenza o l’assenza di una transizione pit/land o land/pit.

Nel 1984 Philips e Sony compresero il potenziale dell’uso dei CD per memorizzare i dati dei calcolatori e pubblicarono così un preciso standard per quelli che ora vengono chiamati CD-ROM (Compact Disc-Read Only Memory).

**CD-Registrabili**

Dal punto di vista fisico i CD-R sono simili ai CD-ROM, tranne per il fatto che contengono una scanalatura larga 0,6 mm che serve a guidare il laser nella fase di scrittura.

Diversamente dai CD-ROM, sui quali vi sono delle vere scanalature fisiche, le diverse proprietà riflettenti dei pit e dei land devono essere simulate.

Per scrivere sul CD-R la potenza del laser viene portata a un valore alto.

Quando il fascio colpisce una regione del pigmento, esso lo scalda al punto da rompere un legame chimico e questo cambiamento della struttura molecolare crea una regione scura.

In fase di lettura il fotorilevatore vede una differenza tra le regioni scure in cui il pigmento è stato colpito e le aree trasparenti dove è ancora intatto.

Quando il disco è letto da un normale lettore CD-ROM questa differenza è interpretata allo stesso modo di quella che vi è tra i pit e i land.

**CD-Riscrivibili**

Anche se abitualmente vengono usati supporti che permettono un’unica scrittura, come la carta o la pellicola fotografica, c’è una grande richiesta di CD-ROM riscrivibili.

Una tecnologia oggi disponibile è quella dei CD-RW (CD-ReWritable, CD -Riscrivibili), che, impiegano una lega con due stati stabili, quello cristallino e quello amorfo, con diverse proprietà riflettenti.

I lettori CD-RW utilizzano laser che funzionano a tre potenze distinte.

Alla potenza più elevata il laser scioglie la lega portandola dallo stato cristallino altamente riflettente a quello amorfo, dotato di una riflettività minore e che rappresenta un pit.

Alla potenza media la lega si scioglie e si ricompone nello stato cristallino naturale per

tornare a rappresentare un land.

Alla potenza più bassa è possibile rilevare lo stato del materiale (per operazioni di lettura), senza però indurre alcuna trasformazione.

I CD-RW vergini sono molto più costosi dei CD-R vergini e per questo non li hanno sostituiti completamente. Inoltre il fatto che, una volta scritto, un CD-R non possa essere cancellato accidentalmente, rappresenta un gran vantaggio per il backup dei dischi.

**DVD**

Le aziende di elettronica stanno cercando un nuovo prodotto per visualizzare i successi cinematografici e molte società informatiche vorrebbero aggiungere funzionalità multimediali al proprio software.

Questa combinazione tra tecnologia e domanda ha portato al DVD, che originalmente era l’acronimo di Digital Video Disk (disco video digitale), mentre ora significa ufficialmente Digital Versatile Disk (disco digitale versatile).

I DVD sono progettati in modo simile ai CD: si inietta policarbonato in uno stampo e sulla superficie si possono distinguere pit e land che vengono illuminati da un diodo laser e letti da un fotorilevatore. Le novità riguardano l’uso di:

1) pit più piccoli;

2) una spirale più stretta;

3) un laser rosso.

Insieme, questi miglioramenti permettono una capacità sette volte maggiore, che arriva fino a 4,7 GB. Un DVD 1x ha una velocità di 1,4 M B/s (contro i 150 KB/s dei CD).

Sfortunatamente il passaggio ai laser rossi obbliga i lettori DVD ad avere un secondo laser o una sofisticata ottica di conversione per poter leggere i CD e i CD-ROM esistenti, cosa che non tutti i lettori permettono. Inoltre la lettura di CD-R e CD-RW può non essere possibile sui lettori DVD.

Per ciò che concerne la capacità sono stati definiti quattro formati:

1) singolo lato, singolo strato (4,7 GB);

2) singolo lato, doppio strato (8,5 GB);

3) doppio lato, singolo strato (9,4 GB);

4) doppio lato, doppio strato (17 GB).

La tecnologia a doppio strato ha uno strato riflettente nella parte bassa, coperto da uno semiriflettente. A seconda del punto in cui il laser è messo a fuoco, esso rimbalza su un livello oppure sull’altro. Il livello più basso richiede che i pit e i land siano leggermente più grandi per essere leggibili in modo affidabile, e per questo motivo la sua capacità è leggermente inferiore rispetto a quella dello strato superiore.

I dischi a doppio lato sono creati prendendo due dischi a singolo lato e incollandoli l’uno contro l’altro, cioè “schiena contro schiena”.

**Input/Output**

Un calcolatore è composto da tre componenti principali: la CPU, le memorie e i dispositivi di I/O.

**Bus**

Sarà trattato nel prossimo capitolo.

**Terminali**

I terminali sono spesso composti da una tastiera e dal monitor.

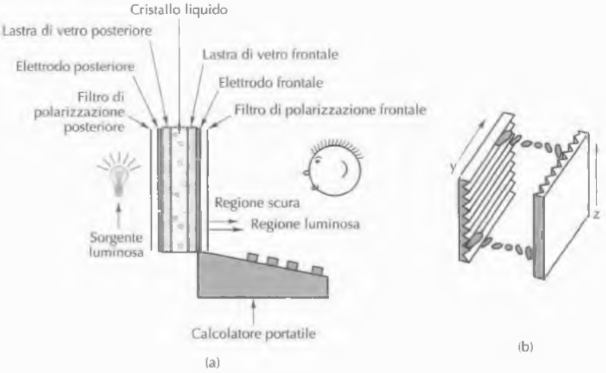
**Tastiere**

La pressione di un tasto di un PC genera un interrupt che stimola una parte del sistema operativo, chiamata gestore dell’interrupt della tastiera. Questa routine, leggendo un registro hardware della tastiera, ricava il numero (da 1uh a 102) associato al tasto premuto. Anche quando si rilascia un tasto viene generato un interrupt. In questo modo se l’utente preme il tasto SHIFT, poi preme e rilascia il tasto M e infine rilascia il tasto SHIFT, il sistema operativo può capire che l’utente voleva digirare una “M” maiuscola invece che una “m” minuscola.

**Monitor piatti**

Data la dimensione dei calcolatori portatili, la tecnologia più comune è quella dello schermo a cristalli liquidi, LCD (Liquid Crystal Display).

I cristalli liquidi sono molecole organiche viscose che si muovono come un liquido, ma che hanno anche una struttura spaziale simile a quella di un cristallo. Quando tutte le molecole sono allineate nella stessa direzione le proprietà ottiche del cristallo dipendono dalla direzione e dalla polarizzazione della luce incidente. E possibile modificare l’allineamento delle molecole, e quindi le proprietà ottiche, tramite l’applicazione di un campo elettrico. In particolare è possibile controllare elettricamente l’intensità della luce uscente facendola passare attraverso un cristallo liquido. Questa proprietà può essere sfruttata per costruire schermi piatti.



**Ram della scheda video**

Entrambi i tipi di schermo vengono ridisegnati dalle 60 alle 100 volte al secondo, accedendo alla RAM della scheda video (Video RAM), situata sulla scheda del controllore dello schermo. Questa memoria ha una o più bitmap che rappresentano lo schermo; su uno schermo di 1600 x 1200 pixel (*picture* *element*, “elemento d’immagine”) la RAM della scheda video deve contenere 1600 x 1200 valori, uno per ogni pixel.

Su uno schermo di fascia alta ogni pixel dovrebbe essere rappresentato da un valore RGB (da Red, Green e Blue) composto da 3 byte, uno per l’intensità di ciascun componente.

Una RAM della scheda video con 1600 x 1200 pixel e 3 byte per pixel richiede circa 5.5 MB per memorizzare l’immagine e una quantità non indifferente di tempo di CPU per operare su di essa.

Nel caso di video in movimento (full-motion) sono necessari almeno 25 fotogrammi al secondo, che corrispondono a una velocità totale di trasferimento dati di 155MB/s. A partire dal Pentium II Intel ha cominciato a impiegare un nuovo bus verso la RAM video, chiamato bus AGP (Accelerated Graphics Port, “porta per la grafica veloce”), che può trasferire 32 bit a una frequenza di 66 MHz fornendo così una velocità di trasferimento dati di 252 MB/s. Sono state prodotte versioni successive del bus (fino a 8 volte più veloce) per mettere a disposizione una larghezza di banda adatta alla grafica altamente interattiva senza sovraccaricare il bus PCI principale.

**Mouse**

Il mouse è dispositivo I/O posto a fianco della tastiera, che al singolo movimento un puntatore sullo schermo si muove in corrispondenza, permettendo agli utenti di interfacciarsi più facilmente con l’aiuto dei due o tre pulsanti presenti su di esso.

Esistono tre tipi di mouse: meccanici, ottici e opto-meccanici.

Il primo mouse aveva sul fondo due rotelle di gomma sporgenti con degli assi disposti perpendicolarmente, in tal modo quando il mouse veniva spostato parallelamente a uno degli assi principali, ruotava soltanto una rotella.

Il secondo tipo di mouse è quello ottico che ha sul fondo un LED (Light Emitting Diode, “diodo luminescente”) e un fotorilevatore. Viene utilizzato su uno speciale tappetino sul quale c’è una griglia di linee molto vicine fra loro; quando il mouse si muove il fotorilevatore riconosce l’incrocio tra due linee rilevando un cambiamento nella quantità di luce generata dal LED che viene riflessa.

Il terzo tipo di mouse è quello opto-meccanico, anch’esso dotato di una pallina che fa ruotare due cilindretti perpendicolari tra loro. Questi sono collegati a codificatori che hanno una serie di fori attraverso i quali può passare la luce; quando il mouse si sposta, i cilindretti ruotano e la luce colpisce il rilevatore ogni volta che un foro si trova allineato con il LED e il suo fotorilevatore. Il numero di impulsi che vengono rilevati è proporzionale allo spostamento effettuato.

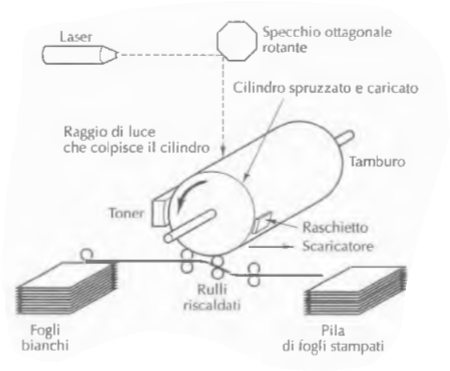
In genere un mouse spedisce al calcolatore una sequenza di 3 byte, chiamata in alcuni casi mickey. Il primo byte contiene la distanza della *x* effettuata dall’ultimo movimento, mentre il secondo byte fornisce la stessa informazione per la *y* e il terzo byte contiene lo stato dei pulsanti del mouse.

**Stampanti**

**Stampanti laser**

Il cuore della stampante a laser è un tamburo rotante molto preciso. All’inizio di ciascun ciclo di pagina il tamburo è caricato fino a circa 1000 volt e rivestito con un materiale fotosensibile. Successivamente la luce generata dal laser passa lungo tutta lunghezza del tamburo e per ottenere la deviazione orizzontale si utilizza uno specchio ottagonale rotante. Il fascio di luce viene modulato per produrre regioni luminose e regioni scure: le parti del tamburo colpite dal raggio perdono la loro carica elettrica.

Il tamburo, dopo aver disegnato una linea di punti, ruota di una frazione di grado per permettere il disegno della linea successiva. A questo punto la prima linea di punti raggiunge il toner, un contenitore di polvere nera elettrostaticamente sensibile. II toner è attirato dai punti che hanno ancora una carica elettrica, formando così un’immagine visiva sulla linea del tamburo. Un istante dopo, ruotando, il tamburo ricoperto di toner viene premuto contro il foglio di carta, trasferendo su questo la polvere nera. Il foglio passa quindi attraverso dei rulli riscaldati per fissare l’immagine, fondendo in modo permanente il toner sulla carta. Nel seguito della rotazione il cilindro viene scaricato e ripulito di ogni residuo del toner, per poter essere nuovamente caricato e ricoperto di materiale fotosensibile, pronto per la stampa della pagina successiva.



Una stampante laser può stampare in bianco e nero ma non può riprodurne le sfumature. Per riprodurre la scala dei grigi la tecnica più comune è quella dei mezzitoni. A seconda dei valori del grigio diverse celle vengono riempite di nero in maniera tale che l’occhio le percepisca come fosse del grigio.

**Stampanti a colori**

Le immagini a colori vengono viste in due modi distinti:

1) le immagini per luce trasmessa sono create mediante la sovrapposizione di tre colori primari additivi, il rosso, il verde e il blu;

2) le immagini per luce riflessa assorbono alcune lunghezze d’onda di luce e riflettono le restanti. Queste immagini sono create dalla sovrapposizione di tre colori primari sottrattivi, il ciano (assorbito dal rosso), il giallo (assorbito dal blu) e il magenta (assorbito dal verde).

Le cinque tecnologie di stampa attualmente in uso si basano sul sistema CMYK (Cyan, Magenta, Yellow, blacK).

**Stampanti speciali**

1) stampanti a colori a getto d’inchiostro: funzionano come quelle monocromatiche ma con quattro cartucce (C, M, Y e K); a un basso prezzo forniscono risultati di buona qualità ma le cartucce non sono molto economiche.

2) stampanti a inchiostro solido:contengono quattro speciali inchiostri a cera solidificati in blocchi che vengono poi sciolti all’interno di serbatoi riscaldati.

3) stampanti a getto di cera:hanno un ampio nastro rivestito di quattro inchiostri a cera e suddiviso in settori lunghi quanto la larghezza della pagina. Al passaggio della carta migliaia di elementi riscaldanti sciolgono la cera, che si fonde con la carta.

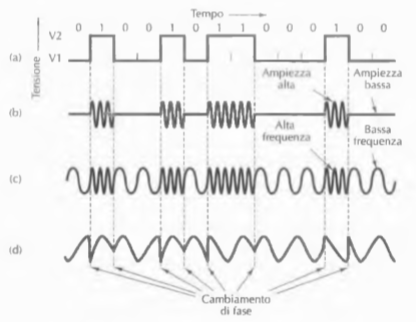
4) stampanti a sublimazione: (passaggio dallo stato solido a quello gassoso senza passare per il liquido, esempi: ghiaccio secco e zolfo) un elemento mobile contenente i coloranti CMYK passa sopra una testina di stampa in cui vi sono migliaia di elementi riscaldanti programmabili; i coloranti vengono vaporizzati e assorbiti da una carta speciale collocata vicino alla testina. Maggiore è la temperatura, maggiore è l’intensità del colore.

5) stampante termica: contiene una piccola testina di stampa su cui vi sono un certo numero di piccoli aghi. Quando una corrente passa attraverso un ago, vengono disegnati su della carta speciale termosensibile dei punti in corrispondenza degli aghi caldi.

**Apparecchiature per le telecomunicazioni**

**Modem**

Una linea telefonica permette la comunicazione fisica tra due sistemi ma non è adatta a trasmettere i segnali di un calcolatore, che quindi vengono convertiti.



Una volta trasmessi subiscono una distorsione, provocando errori di trasmissione, ma relativamente piccola. L’uso di questo segnale, chiamato portante, è alla base della maggior parte dei sistemi di telecomunicazione.

Variando l’ampiezza, la frequenza o la fase di una sinusoide è possibile trasmettere una sequenza di 1 o di 0. Questo processo è conosciuto come modulazione.

Se i dati da trasmettere consistono in una serie di caratteri a 8 bit questi vengono trasmessi un bit alla volta in quanto non esistono connessione telefoniche con 8 coppie di fili. Il modem è un dispositivo che riceve un segnale sotto forma di segnali a due livelli e trasmette i bit in gruppi di uno o due. Ogni sequenza viene di solito limitata con un bit di inizio e uno di fine.

Un modem spedisce i bit a intervalli di tempo regolari. Dati i tempi uniformi di trasmissione dei bit, il modem determina l’inizio del carattere e sincronizza il suo orologio per sapere quando deve campionare la linea per leggere i bit in entrata.

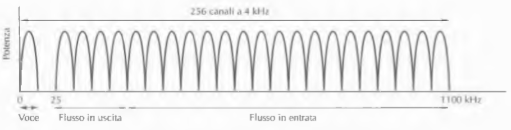
Quasi tutti questi modem sono full-duplex, il che significa che possono trasmettere allo stesso tempo in entrambe le direzioni. Al contrario vengono chiamati half-duplex. Le linee che possono trasmettere in una sola direzione sono invece chiamate simplex.

**Digital subscriber line**

Data la necessità di una connessione Internet più veloce, molte compagnie telefoniche cominciarono a offrire un nuovo servizio chiamato a banda larga.

Le principali offerte erano sottocategorie della xDSL (Digital Subscriber Line) e oggi la più diffusa è la ADSL (Asymmetric DSL).

La lentezza dei modem deriva dal fatto che le linee telefoniche sono state create appositamento per la voce umana e non per il trasferimento di dati. Così il cavo che collega un abbonato alla compagnia telefonica (ciclo locale) viene limitato a 3000 Hz da un filtro imposto dalla stessa compagnia. Il ciclo locale viene suddiviso in 256 canali.



La maggior parte dei fornitori di servizi alloca l’80%-90% della larghezza di banda al canale in entrata. A questa scelta si deve la lettera “A” nell’acronimo ADSL.

**Internet via cavo**

Sparse per la città esistono delle stazioni di testa connesse alla sede principale da fibre ottiche o da cavi con un’alta larghezza di banda.

Da queste scatole partono dei cavi, con una larghezza di banda di circa 750 MHz, ai quali si connettono i clienti che vivono vicino al suo passaggio.

La condivisione del cavo pone il problema di determinare chi può spedire dati, quando e a quale frequenza. Per capirne il funzionamento occorre descriverne quello della TV. In Europa la banda parte da 65 MHz e ogni canale occupa tra i 6 e gli 8 MHz. La parte inferiore della banda non è utilizzata per la trasmissione televisiva.

I cavi moderni funzionano bene anche oltre i 750 MHz. I canali in uscita (cioè dall’utente alla stazione di testa) sono collocati nella banda compresa tra 5 e 42 MHz (leggermente più alta in Europa), mentre il traffico in entrata (cioè dalla stazione di testa all’utente) utilizza la parte alta della banda totale.

****

I modem via cavo, come quelli ADSL, sono sempre attivi: stabiliscono una connessione non appena la macchina viene accesa e la mantengono finché non viene spenta.

**Codifica dei caratteri**

Ogni calcolatore ha un insieme di caratteri che, come minimo indispensabile, comprende

le 26 lettere maiuscole, le 26 lettere minuscole, le cifre da 0 a 9 e un insieme di simboli speciali, come spazio, punto, virgola, segno meno e ritorno a capo.

Per poter utilizzare questi caratteri nel calcolatore occorre assegnare loro un numero:

per esempio a = 1, b = 2, ..., z = 26, + = 27, - = 28. La corrispondenza tra caratteri e numeri naturali costituisce un codice di caratteri. È necessario che due calcolatori che comunicano tra loro utilizzino lo stesso codice, altrimenti non saranno in grado di capirsi. Per questa ragione sono stati definiti alcuni standard.

**ASCII**

(American Standard Code for Information Interchange, “standard americano per lo scambio d’informazioni”).

I caratteri ASCII sono definiti da 7 bit, permettendo così un totale di 128 caratteri distinti.

I codici compresi tra 0 e 1F (in esadecimale) sono caratteri di controllo e non vengono stampati. Molti dei caratteri di controllo ASCII sono pensati per la trasmissione di dati. Un messaggio potrebbe per esempio consistere di un carattere SOH (Start of Header, “inizio intestazione”), un’intestazione, un carattere STX (Start of Text, “inizio testo”), il testo stesso, un ETX (End of Text, “fine testo”) e infine un carattere EOT (End of Transmission, “fine trasmissione”).

I caratteri ASCII stampabili comprendono lettere maiuscole e minuscole, cifre, simboli di punteggiatura e alcuni simboli matematici.

**UNICODE**

Lo standard ASCII è adatto alla lingua inglese, ma meno per altre lingue. I francesi hanno bisogno degli accenti (per esempio, système), i tedeschi dei segni diacritici (per esempio, für), e così via. Altre lingue hanno alfabeti completamente differenti (per esempio, il russo o l’arabo) e alcune lingue addirittura non hanno alfabeto (per esempio, il cinese).

II primo tentativo di espandere il codice ASCII fu il codice IS 646, che aggiungeva ulteriori 128 caratteri, trasformandolo così in un codice a 8 bit chiamato Latin-1.

I caratteri aggiuntivi erano per lo più lettere latine accentate o con segni diacritici.

Il tentativo successivo fu il codice IS 8859 che introdusse il concetto di code page, un insieme di 256 caratteri specifico di una particolare lingua o di un gruppo di lingue. Il problema principale del code page è che il software deve tener traccia di quale pagina è attiva, rendendo tra l’altro impossibile mischiare lingue diverse in uno stesso documento; inoltre questo schema non copre né il giapponese né il cinese.

Un gruppo di produttori di calcolatori ha deciso di risolvere il problema formando un

consorzio per creare un nuovo sistema, chiamato UNICODE, che successivamente è diventato uno Standard Internazionale (IS 10646).

Oggi UNICODE è supportato da alcuni linguaggi di programmazione (per esempio, Java), da alcuni sistemi operativi (per esempio, Windows XP) e da molte applicazioni. L’idea alla base di UNICODE consiste nell’assegnare a ogni carattere un valore a 16 bit, chiamato code point; non esistono quindi caratteri o sequenze speciali composte da più byte. Il fatto che ogni simbolo sia composto da 16 bit rende più semplice la scrittura di programmi.

Utilizzando simboli a 16 bit, UNICODE ha 65.536 code point. Dato che le lingue di tutto il mondo usano complessivamente circa 200.000 simboli, i code point sono una risorsa scarsa che deve essere allocata con molta attenzione. Circa metà dei code point sono già stati assegnati e il consorzio UNICODE valuta in continuazione nuove proposte per i rimanenti. Per far sì che il codice UNICODE fosse accettato più velocemente, il consorzio ha deciso intelligentemente di utilizzare Latin-1 per i code point compresi tra 0 e 255, rendendo più facile la conversione era ASCII e UNICODE. Inoltre, per evitare uno spreco eccessivo di code point, ogni segno diacritico ha il suo proprio code point e spetta al software combinarlo con la lettera vicina per formare un nuovo carattere.

A ciascun alfabeto principale è assegnata una sequenza di zone consecutive.

I code point, oltre alle lettere di questi alfabeti, sono stati assegnati anche ai segni diacritici (112), ai simboli di punteggiatura (112), ai caratteri soprascritti e sottoscritti (48), ai simboli matematici (256), alle forme geometriche (96) e ai simboli ornamentali (192).

Dopo questi code point ci sono i simboli richiesti dal cinese, dal giapponese e dal coreano.

6400 code point sono inoltre stati allocati per uso locale, di modo che gli utenti possano definire caratteri speciali per usi particolari.

Anche se UNICODE risolve molti problemi legati all’internazionalizzazione, esso non risolve (e non cerca di risolvere) tutti i problemi globali. Per esempio, mentre le lettere dell’alfabeto latino sono nel loro ordine corretto, gli ideogrammi Hn non sono ordinati come nel dizionario. Da ciò ne consegue che un programma inglese può ordinare alfabeticamente le parole *cat* e *dog* controllando i valori UNICODE delle loro iniziali, mentre lo stesso non può però essere fatto con un programma giapponese, che necessita di tabelle esterne per capire quale simbolo preceda l’altro.

Un altro problema è che con il tempo continuano ad apparire nuove parole. 50 anni fa nessuno parlava di applet, cyberspazio, gigabyte, laser, modem, smiley o videocassette. Se in inglese l’aggiunta di nuove parole non richiede nuovi code point, in giapponese sì.

Oltre a nuove parole tecniche, c’è la richiesta di aggiungere almeno 20.000 nuovi nomi personali (soprattutto cinesi) e di luoghi. I non vedenti ritengono che il linguaggio Braille dovrebbe far parte di UNICODE, così come altri gruppi con particolari interessi (di tutti i tipi) vorrebbero che ciò che percepiscono fosse codificato, di loro diritto, in code point. Il consorzio UNICODE analizza e valuta tutte le nuove proposte.

UNICODE utilizza lo stesso code point per i caratteri che, in giapponese e cinese, hanno un aspetto simile, ma un significato differente o una scrittura leggermente diversa (come se gli elaboratori di testo inglesi scrivessero "blue” come “blew" solo perché suonano allo stesso modo). Alcune persone interpretano questo fatto come un’ottimizzazione dovuta alla scarsità di code point, altri come un esempio di imperialismo culturale anglosassone.

Per rendere la cosa ancora peggiore si pensi che un dizionario giapponese completo contiene 50.000 kanji (nomi esclusi); avendo a disposizione soltanto 20.992 code point per gli ideogrammi Han, è necessario compiere delle scelte. Non tutti i giapponesi ritengono che un consorzio di produttori di calcolatori, anche se alcuni sono giapponesi, sia il foro ideale per prendere queste decisioni.

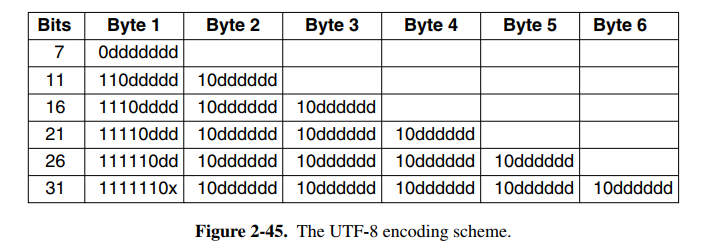
**UTF-8**

Alla fine anche UNICODE ha esaurito i code point e inoltre utilizza 16 bit per carattere per rappresentare testo ASCII puro, il che è uno spreco.

Per rispondere a questi problemi, è stato introdotto il sistema UTF-8 UCS Transformation Format. I codici UTF-8 sono di lunghezza variabile, da 1 a 4 byte, e possono codificare circa due miliardi di caratteri.

Uno dei vantaggi di UTF-8 è che i codici da 0 a 127 corrispondono ai caratteri ASCII, che possono essere espressi in 1 byte.

Sono utilizzati in tutto sei formati differenti, come illustrato nella Figura 2.45. I bit contrassegnati con “d” sono bit di dati.



Un altro vantaggio di UTF-8 è il fatto che il primo byte di ogni carattere UTF-8 determina univocamente il numero di byte nel carattere.

Inoltre, i byte successivi al primo in un carattere UTF-8 iniziano sempre con 10, cosa mai vera per il byte iniziale, rendendo il codice auto sincronizzante. Cioè, in caso di errore di comunicazione o di memoria, è sempre possibile andare avanti e trovare l’inizio del carattere successivo.

1. unità di dati che vengono spostate tra la memoria e i registri [↑](#footnote-ref-1)