

Due domande (cosa servono? Come sono fatti?)

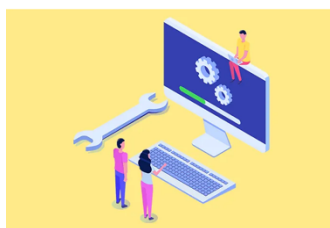
A seguito di questa breve rassegna dell'evoluzione del pensiero informatico, siamo di fronte allo **strumento dei nostri tempi**, tanto desiderato da Babbage e Lovelace: **un computer digitale!**

È una macchina con la quale noi lavoriamo, ci divertiamo, ci informiamo e comunichiamo tutti i giorni. È uno strumento importante e abbiamo visto quali sono state le scoperte teoriche e pratiche che hanno permesso la nascita del computer, ma.....

Ora le domande che possiamo porci sono due:

Cosa servono?

Come sono fatti?



Ognuno di noi ha avuto un inizio particolare e personale con il computer, il più comune avviene all'età delle scuole elementari e si tratta o di semplice video scrittura, o di un disegno con Paint. Per un ragazzino/a si tratta sempre di un primo approccio per prendere confidenza e imparare a destreggiarsi con il mouse per raggiungere una buona coordinazione oculo-manuale. I prodotti sono ovviamente semplici e banali, ma finalizzati all'auto addestramento.

Cosa servono?

La prima richiesta fatta ai computer, nei lontani anni 60 e precedenti, era quella di risolvere problemi matematici un po' alla Babbage, ma è sempre qualcosa di misterioso e affascinante! Imparare ad affrontare problemi e risolverli (Problem Solving) è stato uno degli obiettivi più affascinanti e importanti di questo periodo! Le linee guida sono indicate nella fig. 35.

Applicare le 5 azioni comporta modalità che devono adattarsi agli strumenti con i quali operiamo.

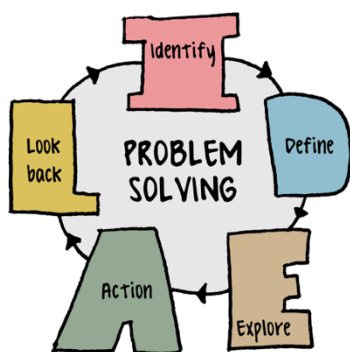


Figura 1 Problem Solving

In sintesi ecco i cinque punti che si affrontano in modo circolare e ricorsivo per risolvere un problema:

1. Identificazione
2. Definizione
3. Esplorazione
4. Azione
5. Guardare indietro, ricontrollare

DAL PROBLEMA AL PROGRAMMA

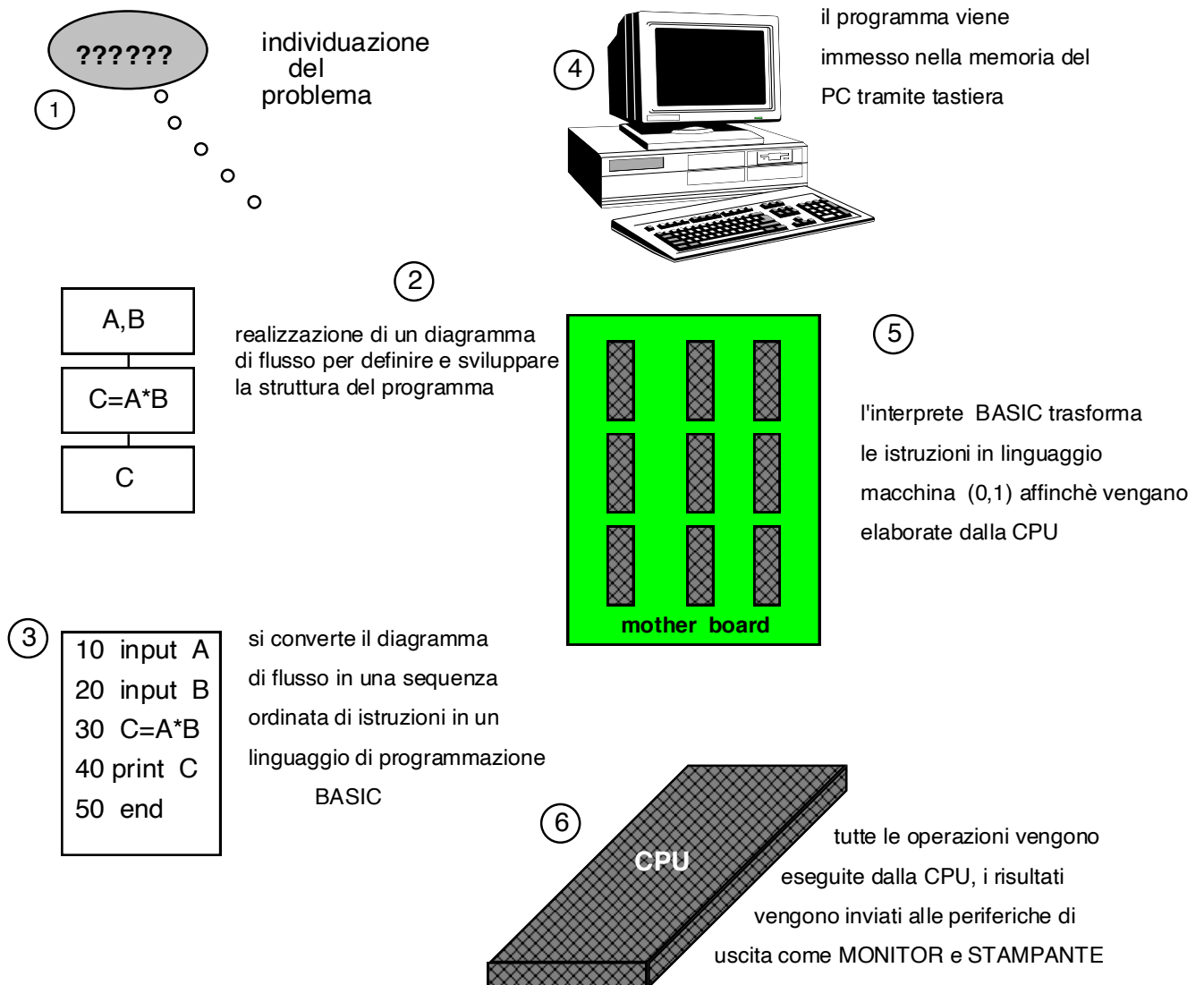
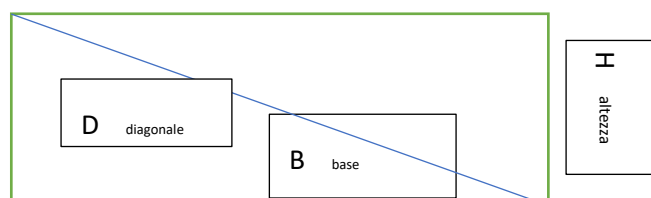


Figura 2 Dal Problema al Programma

Problema lineare

Si vuole calcolare la misura dell'area e del perimetro di un numero imprecisato, ma comunque alto di rettangoli. Per fare questo si utilizza il seguente programma:

```
10 INPUT B
20 INPUT H
30 A=B*H
40 P=2*(B+H)
50 PRINT A
60 PRINT P
70 END
```



Si osservi come tutte le istruzioni siano numerate (10;20;30;40;50;60;70) per assicurare la **giusta SEQUENZIALITA'** alle istruzioni del programma, appunto con i cosiddetti numeri di linea¹.

- Le istruzioni caratterizzate dai numeri di linea 10 e 20 servono ad introdurre i dati riguardanti la base e l'altezza del rettangolo.
- Le istruzioni caratterizzate dai numeri di linea 30 e 40 eseguono il calcolo dell'area e del perimetro del rettangolo con i dati precedentemente immessi.
- Le istruzioni caratterizzate dai numeri di linea 50 e 60 visualizzano sullo schermo l'area ed il perimetro.
- L'istruzione caratterizzata dal numero di linea 70 pone fine all'esecuzione del programma.

Una volta digitato il programma lo si LANCIA con un opportuno comando (RUN) che lo mette in esecuzione. Se volessimo calcolare anche la diagonale, tramite il teorema di Pitagora, basta modificare leggermente il programma inserendo altre due righe di programma e dare di nuovo il comando RUN.

45 D=SQR(B^2+H^2)

65 PRINT D

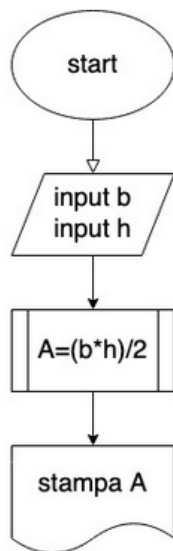
Sullo schermo appare un quadratino lampeggiante che ti chiede due numeri (due sono le istruzioni INPUT) e una volta soddisfatta la richiesta il programma calcola quanto richiesto. Questo programma può essere messo in esecuzione con il comando RUN quante volte ne abbia necessità.

Si calcoli la misura dell'area, del perimetro e la diagonale dei seguenti rettangoli (per brevità sono cinque):

	INPUT	INPUT	OUTPUT	OUTPUT	OUTPUT
	B	H	A	P	D
I rett.	4	7	28	22	8,06
II rett.	3,5	1,43	5,005	9,86	3,78
III rett.	12	6	72	36	13,42
IV rett.	39,4	10,1	397,94	99	40,67
V rett.	112	56	6272	336	125,22

Per meglio delineare le strutture di calcolo (algoritmi) ci si avvale dei flow chart detti diagrammi di flusso.

¹ In genere i numeri di linea si intervallano di 10, in modo da facilitare l'inserimento di altre linee di programma.



Questo è un semplicissimo flow chart che mi permette di visualizzare il calcolo dell'area di un triangolo. Ci sono diverse simbologie per la rappresentazione dei simboli grafici.

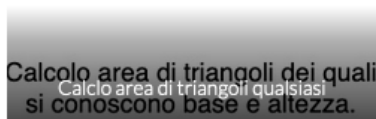


Figura 3 Calcolo dell'area di un triangolo

Il linguaggio adoperato è il BASIC. Quando apparvero sul mercato i primi HOME computer come il C64, il Texas TI90 o lo ZX Sinclair tutti si avvalevano del linguaggio BASIC e le prime applicazioni con le quali ci si cimentava erano quasi sempre di natura matematica.

BASIC è una famiglia di linguaggi di programmazione generici e di alto livello progettati per facilità d'uso. La versione originale è stata creata da John G. Kemeny e Thomas E. Kurtz al Dartmouth College nel 1964. Volevano consentire agli studenti in campi non scientifici di utilizzare i computer.

A quel tempo, anni 70, l'home computer più diffuso e famoso era il C64. Quando lo accendevi si apriva con una finestra dove potevi comunicare direttamente in BASIC:



Si parla di un interprete BASIC, primo linguaggio a diffusione mondiale adoperato per strutturare problemi e dialogare con il PC. Questa parola è un acronimo il cui significato è:

Beginner 's	All-Purpose	Symbolic	Instruction	Code
Pincipianti	Tutte-Applicazioni	Simboliche	Istruzioni	Codice

In una espressione più accessibile significa: **Codifica a Istruzioni Simboliche, per Principianti, per Applicazioni di Carattere Generale.**

Il vocabolo **BASIC** in lingua inglese significa basilare, fondamentale, essenziale, semplice; questo è un esempio di come si sia voluto "forzare" l'acronimo in modo che assuma un doppio significato. Nel campo dell'informatica sono numerosi gli esempi di acronimi o di parole prese in prestito dal linguaggio comune e che hanno assunto poi un significato proprio.

In questi esempi lineari non sono state previste **scelte** e quindi l'esecuzione è stata lineare, ma non sempre è così! Ci sono problemi più complicati nei quali sono previste scelte. Il PC è in grado di farle? Direi proprio di sì: con la funzione IF THEN ELSE.

Il teorema di Böhm-Jacopini, enunciato nel 1966 dagli informatici Corrado Böhm e Giuseppe Jacopini, afferma che qualunque algoritmo può essere realizzato utilizzando le sole **tre** strutture di controllo fondamentali: **la sequenza, la selezione ed il ciclo.**

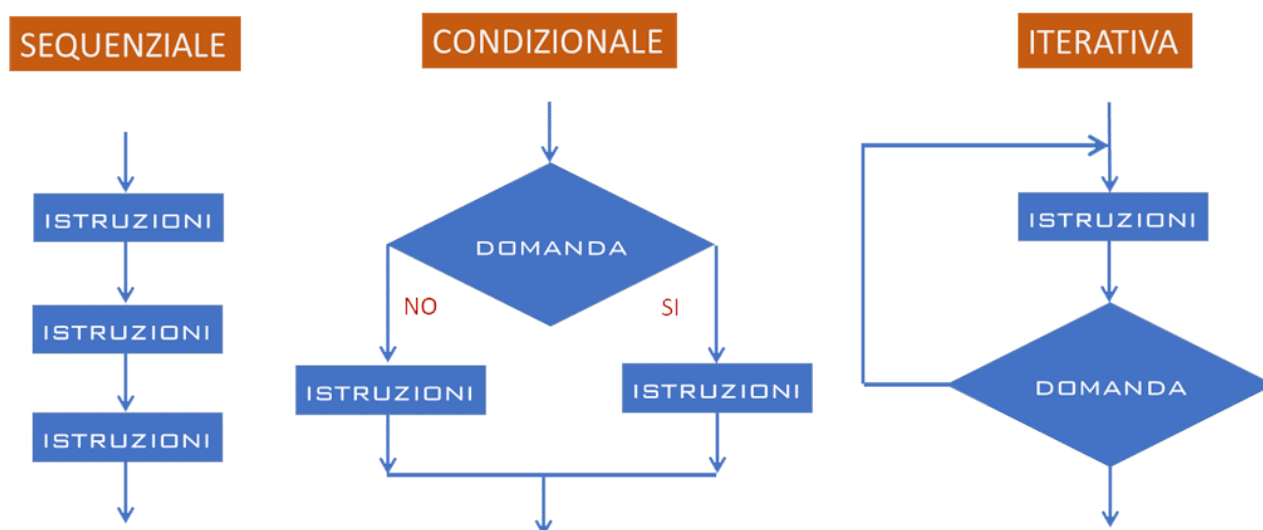


Figura 4 le tre strutture principali per definire e strutturare un algoritmo

Adesso definiamo cosa si intende per algoritmo.

L'algoritmo

Cosa sono gli algoritmi? Sono metodi di calcolo che risolvono i vari problemi e possono essere implementati, tramite apposito linguaggio, in un computer per essere calcolati.

I programmatori per risolvere i problemi con il computer, li devono esprimere sotto forma di *algoritmi* che verranno poi tradotti in programmi in un appropriato linguaggio di programmazione.

Il termine *algoritmo* deriva dal nome del matematico arabo **Al Khwarismi**² che pubblicò l'importante opera *Kitab al-jabr³ wal muqabala* (L'arte di numerare ed ordinare le parti di un tutto) da cui derivò il termine *algebra*. Quando questa opera fu introdotta in Europa, fu tradotta in latino e prese il titolo di *Algorismus*, la sua importanza è fondamentale per l'umanità perché grazie ad essa fu introdotta la numerazione posizionale, quella che usiamo oggi normalmente.

La prima definizione del termine *algoritmo*, nella sua accezione attuale, è stata data dal matematico Markov.

L'algoritmo è un qualsiasi insieme di regole precise che definisca un procedimento di calcolo destinato ad ottenere un determinato risultato partendo da determinati dati iniziali.

Può sembrare una definizione approssimata: la migliore definizione di *algoritmo* attualmente valida in campo informatico è stata data da Knuth, nel libro intitolato *The Art of computer programming*⁴.

L'algoritmo è un insieme di regole o istruzioni aventi le seguenti cinque caratteristiche:

1. deve essere finito, e concludersi dopo un numero finito di operazioni
2. deve essere definito e preciso: ogni istruzione deve essere definita in forma non ambigua
3. se ci sono dati in ingresso, il campo di applicazione deve essere precisato (ad esempio numeri interi, reali, caratteri ecc.)
4. deve fornire almeno un risultato (dato in uscita)
5. deve essere eseguibile: tutte le operazioni devono poter essere eseguite esattamente, e in tempo finito, da un uomo che utilizzi mezzi manuali

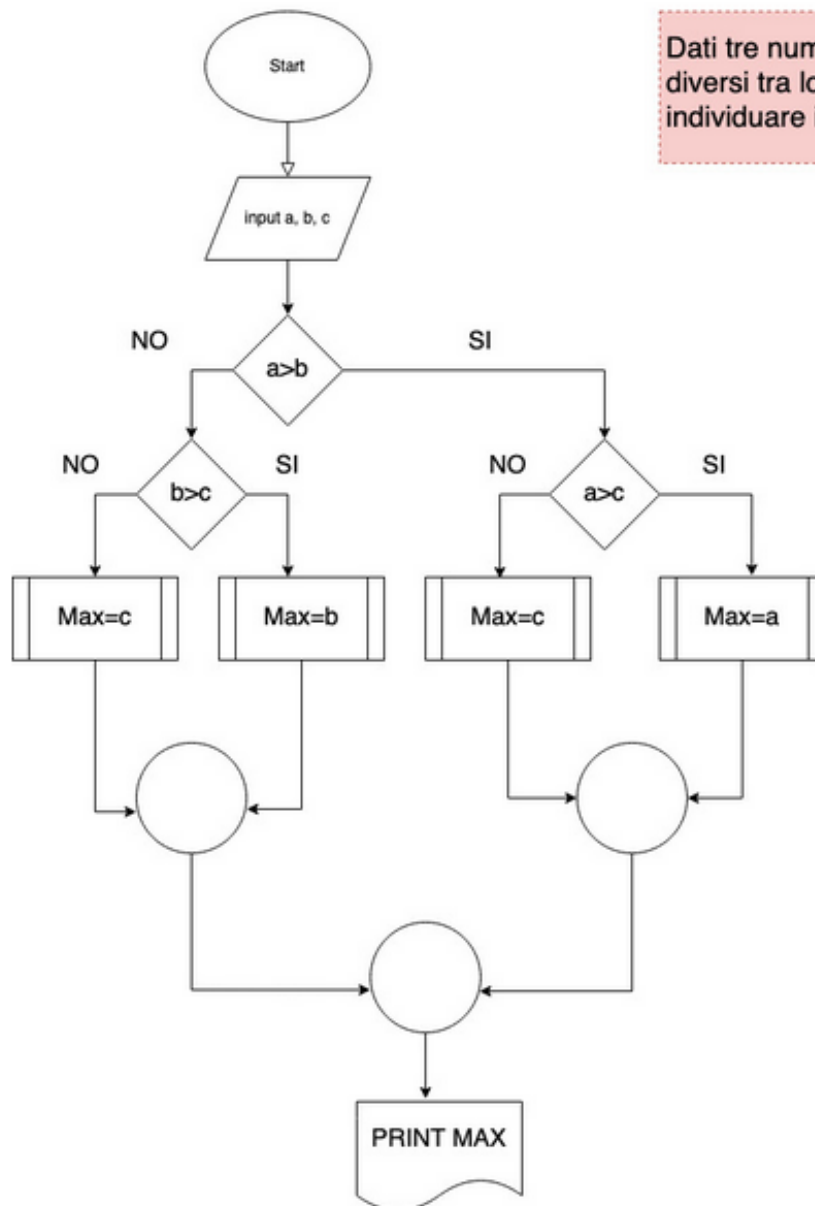
Individuato il problema bisogna quindi, come primo passo, creare l'algoritmo che ne permetta la soluzione.

Uno strumento utile per la definizione di un algoritmo è il diagramma di flusso o *flow-chart*. Nella figura seguente sono sintetizzate le principali fasi per risolvere un problema.

² **Al Khwarismi** dal nome di questo matematico deriva il termine **algoritmo**.

³ **Al-jabr**, da questa parola araba deriva il termine **algebra**!

⁴ Tratto da "Introduzione al Pascal" di Pierre Le Beux JACKSON.



Dati tre numeri reali e diversi tra loro, a, b, c individuare il più grande.

Calcolo di MAX tra tre numeri diversi

Figura 5 Flow Chart per il calcolo del MAX tra tre numeri

Proviamo a leggere questo flow chart.

- Si introducono i tre numeri a, b, c;
- a è maggiore di b? in questo caso si deve fare una scelta, SE no ALLORA si confronta b è maggiore di c?
- SE no ALLORA il MAX è c
-
-
-

Questo è l'inizio, acquisito il senso, poi è facile proseguire la lettura del flow chart.

Il problema poi è implementare con l'apposito linguaggio le istruzioni che codificano questo algoritmo.

Altri problemi

1 Come riconoscere con un apposito programma per computer, **se un numero è pari o dispari?**

Si individua l'algoritmo. È molto semplice si prende il numero da esaminare e lo si divide per due: **se** il resto della divisione è nullo **allora** il numero è pari **altrimenti** è dispari.

Abbiamo adoperato una implicazione logica che la macchina è in grado di riconoscere. In ambito informatico l'implicazione logica è indicata con l'asserzione IF THEN ELSE che può essere tradotta letteralmente in "se allora altrimenti". In alcuni linguaggi di programmazione, come il Basic, l'asserzione IF THEN coincide esattamente con la sintassi dell'istruzione condizionale.

Se volete seguire e vedere lo sviluppo e risoluzione di questo problema, la spiegazione completa e più dettagliata può essere reperita al seguente indirizzo:

<https://www.adalovelace.it/un-numero-e-pari-o-dispari/>

l'algoritmo è stato implementato in EXCEL (foglio elettronico di Microsoft).

2 Jenni

Fino ad ora abbiamo risolto dei semplici problemi numerici, l'esempio seguente parla di una bimba di 10 anni che ha difficoltà nella lingua (italiana):

Jenni era una bimba che aveva difficoltà nel capire la grammatica, non riesce a capire la differenza tra nome, verbo, avverbio: è un problema! Come si può risolverlo?

L'algoritmo e la sua risoluzione si possono reperire al seguente indirizzo:

<https://www.adalovelace.it/jenni/>

È un bel problema d'apprendimento, ma con apposito algoritmo si riesce a compilare una app che aiuta a risolvere questa difficoltà. (è sempre stato usato EXCEL)

3 Perché il periodo vittoriano e Ada Lovelace?

L'avventura moderna del calcolo così come la conosciamo si è sviluppata nel periodo vittoriano del primo 800 grazie a due personaggi già citati: Charles Babbage & Ada Byron contessa di Lovelace! Ed ecco a voi una presentazione che vi guida in questo magico mondo.

<https://www.adalovelace.it/presentazioni-in-pdf/>

sono presenti in formato pdf diverse presentazioni (scaricabili) quella che dovete cercare è la presentazione:

Ada & Charles

E' una monografia su questi due grandi personaggi della storia informatica. La presentazione è molto avvincente anche grazie all'uso dei disegni di Sydney Padua tratti dal libro "THE THRILLING ADVENTURES OF LOVELACE AND BABBAGE". Questo libro è un caposaldo di tutta la mia trattativa, il sottotitolo "**THE (MOSTLY) TRUE STORY OF THE FIRST COMPUTER" esprime il senso di tutto questo lavoro ((per lo più) LA VERA STORIA DEL PRIMO COMPUTER). Un vero e sentito ringraziamento all'autrice! Mia opinione personale è che questa opera è un **capolavoro**! Sono 55 slide anno di produzione 2019



Dentro al computer

L'avventura prosegue!

Come è fatto un computer, quali sono i suoi componenti e come sono collegati tra loro?

Guardando una Chiesa o un edificio si cerca di capirne l'architettura, lo stile e quale architetto l'ha creata.

Nel nostro caso l'architetto del computer è stato John von Neumann.

Neumann, John von (Budapest 1903 - Washington 1957), matematico e logico statunitense di origine ungherese. Studiò a Zurigo e presso le università di Berlino e di Budapest. Recatosi negli Stati Uniti nel 1933 per collaborare con l'università di Princeton, da quell'anno divenne membro permanente dell'Institute for Advanced Study di Princeton. Durante la seconda guerra mondiale operò come consulente per il progetto di costruzione della bomba atomica presso il Los Alamos National Laboratory. Nel 1955 divenne membro della Commissione statunitense per l'energia atomica.



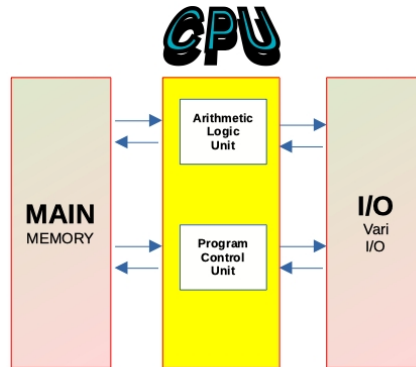
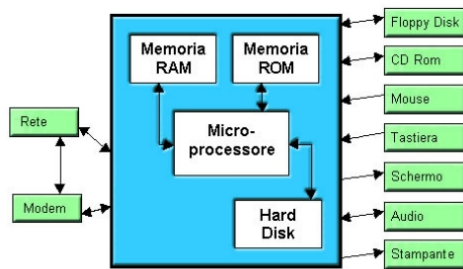
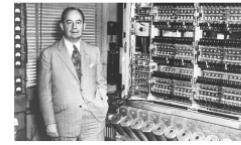
Figura 6 John Von Neumann

Brillante matematico, Von Neumann assiomatizzò la teoria degli insiemi, sviluppò insieme a Oskar Morgenstern il campo della teoria dei giochi e approfondì gli studi sulla teoria delle probabilità; inoltre, si distinse per il fondamentale contributo offerto alla teoria quantistica, in particolare per il concetto di "anelli di operatori" e per i suoi studi pionieristici di matematica applicata alla statistica e all'analisi numerica. In particolare comprese come l'impiego dei computer avrebbe dato un notevole impulso alla matematica applicata e consentito la risoluzione di specifici problemi. Fino a quel momento i computer erano stati utilizzati nella elaborazione di tabulati. Nel 1944 si interessò agli studi del fisico statunitense Howard Aiken, che aveva messo a punto il calcolatore ASCC, e del matematico George Stibitz, inventore di computer elettromeccanici. Negli anni seguenti, fu consulente di numerose commissioni di studio; in virtù di questo ruolo, avviò la collaborazione del Los Alamos National Laboratory con la Moore School of Electrical Engineering dell'università della Pennsylvania che si stava occupando della progettazione del computer ENIAC, e ideò il computer IAS per l'Institute for Advanced Studies. Nel 1945, descrisse l'architettura del computer elettronico digitale EDVAC (Electronic Discrete Variable Automatic Computer), capace di mantenere nella propria memoria i programmi da eseguire; l'apparecchio fu realizzato dal fisico John W. Mauchly e dall'ingegnere elettronico J. Presper Eckert. Operando con il gruppo di ricerca di Los Alamos, perfezionò il computer IAS, alla continua ricerca di adattare le potenzialità dei computer alla risoluzione di problemi scientifici. Nel 1952 progettò il primo computer utilizzando un programma flessibile, il MANIAC I. Nel 1956 la Commissione per l'energia atomica gli assegnò l'Enrico Fermi Award per il suo notevole contributo alla teoria e alla progettazione dei computer.

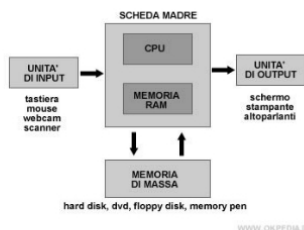
Microsoft ® Encarta ® Enciclopedia 2002. © 1993-2001 Microsoft Corporation. Tutti i diritti riservati.



Architettura di John von Neumann



John von Neumann
Il padre del computer



Ci sono diversi schemi, dal più semplice al più complesso o dettagliato.

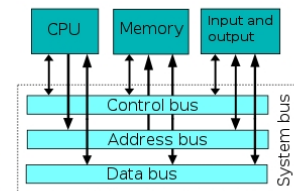


Figura 7 Vari schemi relativi all'architettura del computer

Gli schemi presentati nella fig. 41 sono tutti analoghi e rappresentano il funzionamento dei moderni computer. Tutti fanno riferimento al modello Neumann: cuore del modello è la CPU o la Central Processing Unit, il microprocessore dove vengono elaborati i programmi o i dati con l'ausilio della ALU o Arithmetic Logic Unit, è l'unità logica aritmetica che esegue tutte le operazioni avvalendosi dell'apporto delle Logic Gates. Queste operazioni una volta eseguite vengono poi rispediti alla CPU che li invia ai dispositivi di output.

Facciamo un esempio:

1. Inseriamo un programma di video scrittura, con un dispositivo di input viene passato alla CPU che lo invia in memoria;
2. Dalla memoria torna alla CPU per l'elaborazione;
3. Vengono inviati i dati sempre con dispositivo input;
4. La CPU li invia alla ALU che li elabora;
5. I dati elaborati tornano alla CPU che li invia in memoria o ad un dispositivo di output.

È una forte semplificazione, ma intanto diamo l'idea della complessità e numerosità delle operazioni che un computer deve eseguire. Sono molteplici e macchinose, ma la forza della macchina è la velocità con le quali riesce ad eseguirle. **LA VELOCITA'!!!**

Le specifiche più importanti che caratterizzano maggiormente un computer sono:

- La RAM (Random Access Memory) memoria ad accesso casuale, ovviamente più grande è più performante è il computer;
- La ROM (Read Only Memory) memoria di sola lettura dove sono contenute le istruzioni per l'avvio del computer. La memoria ROM ospita firmware e driver contenenti i set di istruzioni necessari all'avvio del processore e delle schede di espansione e periferiche ospitate dalla motherboard
- Il CLOCK Il termine clock, in elettronica, indica un segnale periodico, generalmente un'onda quadra, utilizzato per sincronizzare il funzionamento dei dispositivi elettronici digitali. Può essere generato da qualsiasi oscillatore, si usa generalmente il tipo a quarzo per la sua alta stabilità di oscillazione. Determina la velocità di esecuzione delle istruzioni.

Questo schema è quello più completo e sufficientemente dettagliato in tutte le sue componenti. Notare le frecce e la loro direzione che indicano i flussi di informazione.

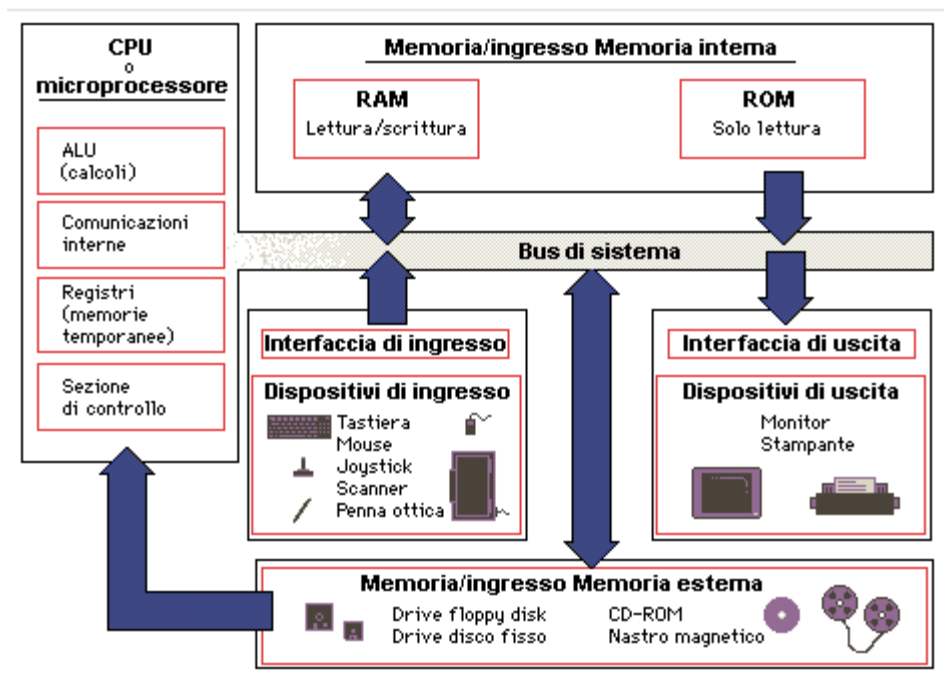
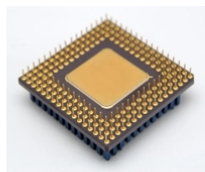


Figura 8 schema completo di un computer

Il computer tratta flussi di informazione. Quando accendi il computer prende le informazioni dalla ROM, tramite il bus si attivano tutte le periferiche o dispositivi I/O. Iniziamo a lavorare con la tastiera. Digittiamo le informazioni che vanno tramite il bus alla CPU e dopo essere elaborate, sempre con il bus alla RAM dove sono temporaneamente depositate. Sempre con il bus possono andare ai dispositivi di uscita più importanti come il monitor o la stampante. Se vogliamo che la nostra elaborazione rimanga permanente, sempre tramite il bus, la possiamo mandare ai dispositivi esterni come ad esempio l'hard disk. Spegniamo la macchina e quello che era presente nella RAM scompare, per questo è chiamata memoria volatile.

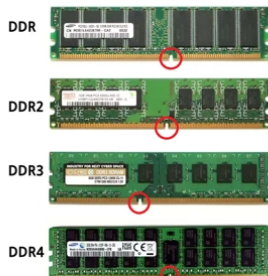


SPECIFICHE DI UN COMPUTER



CPU

prefisso			informatici		scientifici
KILO	K	2^{10}	1.024	10^3	1.000
MEGA	M	2^{20}	1.048.576	10^6	1.000.000
GIGA	G	2^{30}	1.073.741.824	10^9	1.000.000.000
TERA	T	2^{40}	1.099.511.627.776	10^{12}	1.000.000.000.000



Vari tipi di RAM

Simbolo	in Bit	in Byte	appross.
1 KB (kilobyte)	8.192	1.024	1.000
1 MB (megabyte)	8.388.608	1.048.576	1.000.000
1 GB (gigabyte)	8.589.934.592	1.073.741.824	1.000.000.000
1 TB (terabyte)	8.796.093.302.400	1.099.511.628.000	1.000.000.000.000



Hard Disk
HD

SSD
Solid
State
Drive



Clock, orologio, è un circuito oscillatore (basato su un cristallo di quarzo) che fornisce una temporizzazione a tutti i circuiti del computer e misura la velocità di esecuzione delle istruzioni. La frequenza degli impulsi è misurata in Mhz (MegaHertz).

Figura 9 dettagli tecnici dei componenti principali del computer

Struttura operativa di un elaboratore

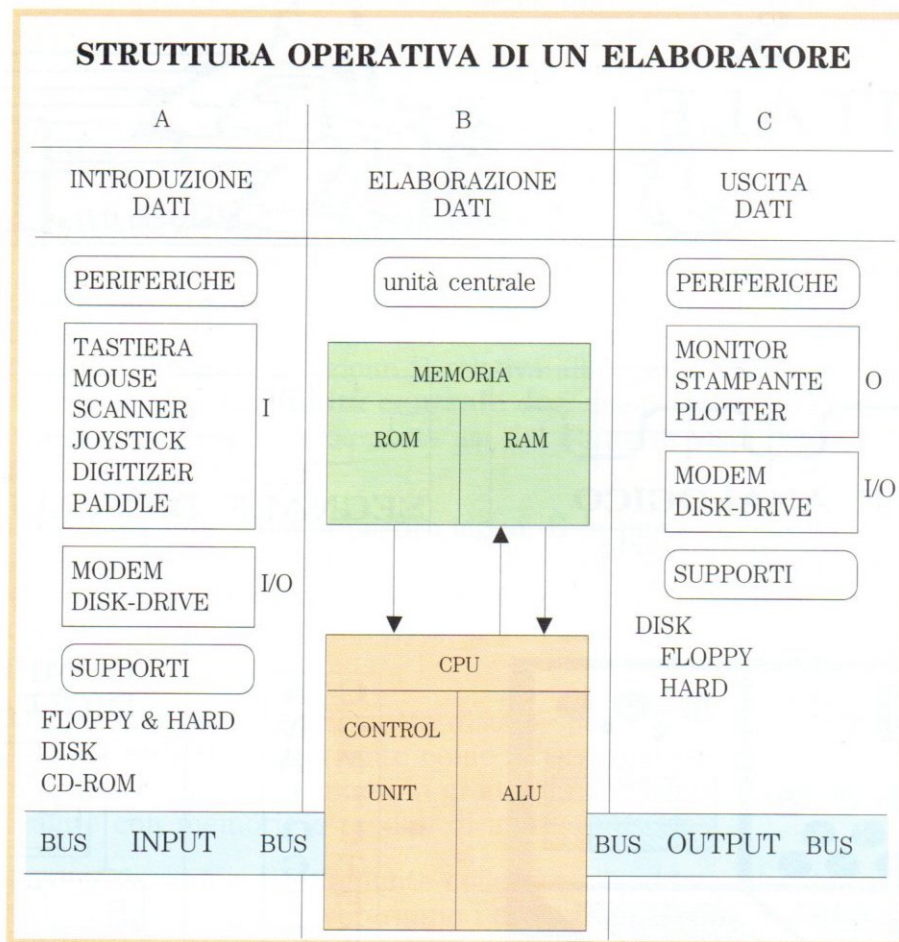


Figura 10 Struttura operativa di un computer

Lo schema riportato illustra la struttura e il funzionamento di un calcolatore numerico. Un calcolatore numerico lavora su **dati discreti** e **non continui**, difatti si parla di calcolatore numerico e non di calcolatore di tipo analogico. Questi dati discreti sono codificati per mezzo di numeri binari.

Analogico e Digitale

Nella letteratura informatica sono abbastanza ricorrenti due termini: *analogico* e *digitale*, (digitale deriva dall'inglese digit che significa *cifra* e digit deriva a sua volta dal latino *digitus* che significa dito).

L'esempio che segue e la relativa terminologia, possono sembrare ai più esperti non propriamente ortodossi, ma considerato lo scopo divulgativo del testo, è sorto il bisogno di non addentrarsi più del necessario in corrette, ma alquanto difficili, definizioni.

I termini *analogico* e *digitale* si riferiscono a due differenti modi di rappresentare la cosiddetta parte quantitativa di una informazione. Una informazione è infatti costituita, semplicisticamente parlando, da due parti: una qualitativa, l'altra quantitativa.

Supponiamo di fornire delle "informazioni" di natura climatica su di una giornata di agosto.

1. **è caldo**, è la parte "qualitativa" dell'informazione

2. **la temperatura è di 35 gradi Celsius**, è la parte "quantitativa" dell'informazione.

Il computer elabora la parte quantitativa dell'informazione.
--

Nelle apparecchiature analogiche le quantità numeriche sono rappresentate dallo stato fisico di una grandezza che può variare in modo continuo entro un certo intervallo di valori. Ad esempio il termometro a mercurio è uno strumento analogico, in quanto il livello del liquido presente nel capillare può assumere un numero infinito di posizioni possibili.

Un esempio elementare di apparecchiatura digitale è costituito da un interruttore di circuito per il quale non sono possibili variazioni continue di posizione, ma solo posizioni discrete, in questo caso solo **due posizioni**: quella corrispondente al "circuito aperto" e quella corrispondente al "circuito chiuso". Gli storici calcolatori a manovella sono esempi di dispositivi digitali (numerici), essendo costituiti da rotismi in cui ogni ruota dentata del meccanismo ha a disposizione solo dieci posizioni possibili.

I computer adoperati quotidianamente sono dei *dispositivi digitali*, perché il loro funzionamento è basato sulla logica del circuito aperto o chiuso, ma esistono anche computer di tipo analogico e sono adoperati in laboratori a carattere scientifico e tecnico.

Elaborazione dati

Lo schema esemplificativo è suddiviso in tre sezioni principali: l'introduzione dati, l'elaborazione dati e l'uscita dati. Vengono quindi citate le periferiche necessarie all'introduzione e all'uscita dati e i relativi supporti, queste due sezioni saranno trattate diffusamente in seguito, mentre un discorso necessariamente più approfondito deve essere fatto per quanto riguarda la sezione B cioè l'elaborazione dati.

In questa sezione viene descritta **l'unità centrale** dove sono svolte tutte le operazioni richieste al PC. L'unità centrale può essere scomposta sia dal punto di vista fisico sia funzionale in tre unità principali:

ALU: Arithmetic and Logic Unit, unità aritmetico logico. È la parte esecutiva del computer, difatti esegue le operazioni matematiche (addizione, sottrazione, moltiplicazione, divisione etc.) logiche (confronti del tipo maggiore, minore etc e del tipo vero falso); inoltre esegue la manipolazione dei bit all'interno dei byte, il suo funzionamento è controllato dall'unità di controllo.

UNIT CONTROL: unità di controllo. Si tratta di uno speciale circuito, progettato in modo tale che la CPU si comporti esattamente come le istruzioni ricevute in memoria. Questo importante blocco funzionale è quindi in grado di controllare l'esecuzione di una istruzione e di interagire con memorie e moduli di ingresso-uscita.

MEMORIA: La memoria, che è direttamente collegata ai suddetti blocchi funzionali, contiene i programmi e i dati sui quali lavoreranno i programmi. Le memorie in genere sono caratterizzate dalle seguenti grandezze: supporto fisico, modalità e tempo d'accesso, capacità, tipo di operazioni consentite.

Entrando in maggior dettaglio, la memoria che interagisce nell'unità centrale è del tipo a semiconduttori (tecnologia MOS da Metal Oxide Semi-conductor). L'elemento base di questo tipo di memoria è un BISTABILE che contiene l'informazione codificata come 0 o 1. I bistabili sono costituiti da due transistori opportunamente collegati e sono caratterizzati da due stati stabili. In uno stato un transistor è saturo e l'altro è interdetto, mentre nell'altro stato vi è una situazione opposta. Ad uno di questi stati è associato 1 e all'altro 0.

La modalità d'accesso è del tipo ad accesso diretto, ed il tempo che indica il tempo necessario alla lettura di un quantum di informazione è dell'ordine dei nanosecondi.

Come tipo di operazioni consentite si possono avere **memorie a lettura/scrittura** e **memorie a sola lettura**.

RAM: Random Access Memory. È la memoria ad accesso casuale (o diretto), ed è quella dove vengono collocati i programmi e i dati che dovranno essere elaborati dal processore. È una memoria volatile, non ritiene permanentemente le informazioni, pertanto quando si spegne il computer, si perde tutto!!!!

ROM: Read Only Memory. Questa è la memoria di sola lettura, in essa risiede un programma elementare di inizializzazione del computer, necessario quando si avvia la macchina per andare a recuperare il sistema operativo residente in genere su memorie di massa.

È venuto il momento di parlare del componente fondamentale dell'unità centrale: il MICROPROCESSORE.

Viene anche chiamato CPU (Central Processing Unit), è un chip dalle dimensioni molto ridotte dell'ordine delle decine di μm , in esso vi sono racchiuse mediante tecniche di microfilmatura gli schemi e i componenti dei circuiti elettrici. In pratica si tratta di un circuito ad alta integrazione di componenti.

È la cosiddetta UNITA' CENTRALE DI ELABORAZIONE ed è preposta al controllo, al coordinamento di tutte le funzioni del PC oltre all'elaborazione dei dati. Le sue dimensioni sono molto ridotte (dell'ordine di pochi cm^2) ed insieme ad altri chip come ad esempio quelli costituenti la memoria, è alloggiata sulla MOTHER BOARD o scheda madre. La MOTHER BOARD assieme ad altre schede preposte a diverse funzioni specifiche è alloggiata all'interno della UNITA' DI SISTEMA.

A questo scopo viene proposto un disegno che illustra in modo più approfondito la struttura dell'unità di sistema, alloggiata in una scatola metallica detta *cabinet*. Le varie retinature sottolineano le connessioni tra le varie parti. Un rilievo particolare riveste la MOTHER BOARD (scheda madre); è una scheda dove sono alloggiati i chips più importanti come la CPU, la memoria e altri.

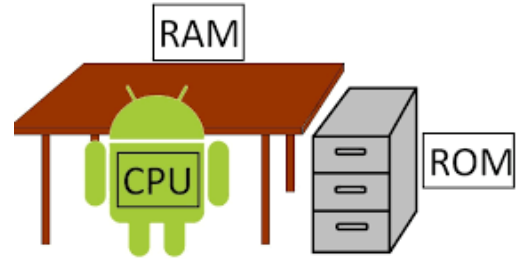
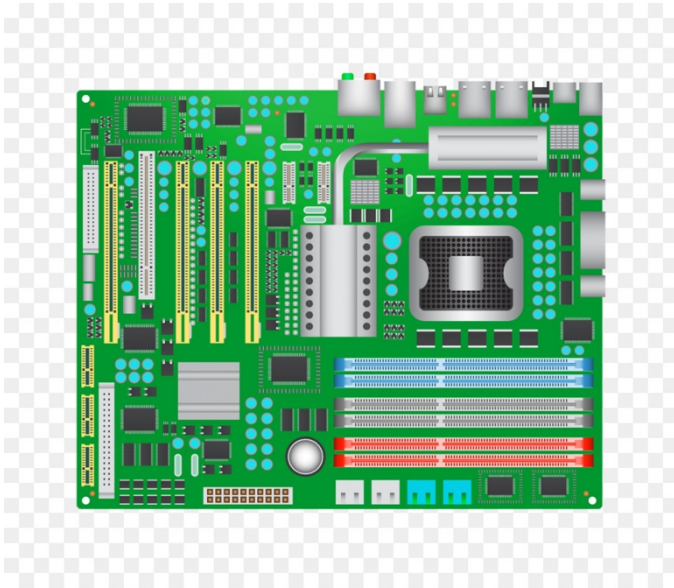


Figura 11 Le memorie RAM e ROM, notare la ROM disegnata come archivio e la RAM disegnata come un tavolo libero.

Figura 12 Mother Board cartoon

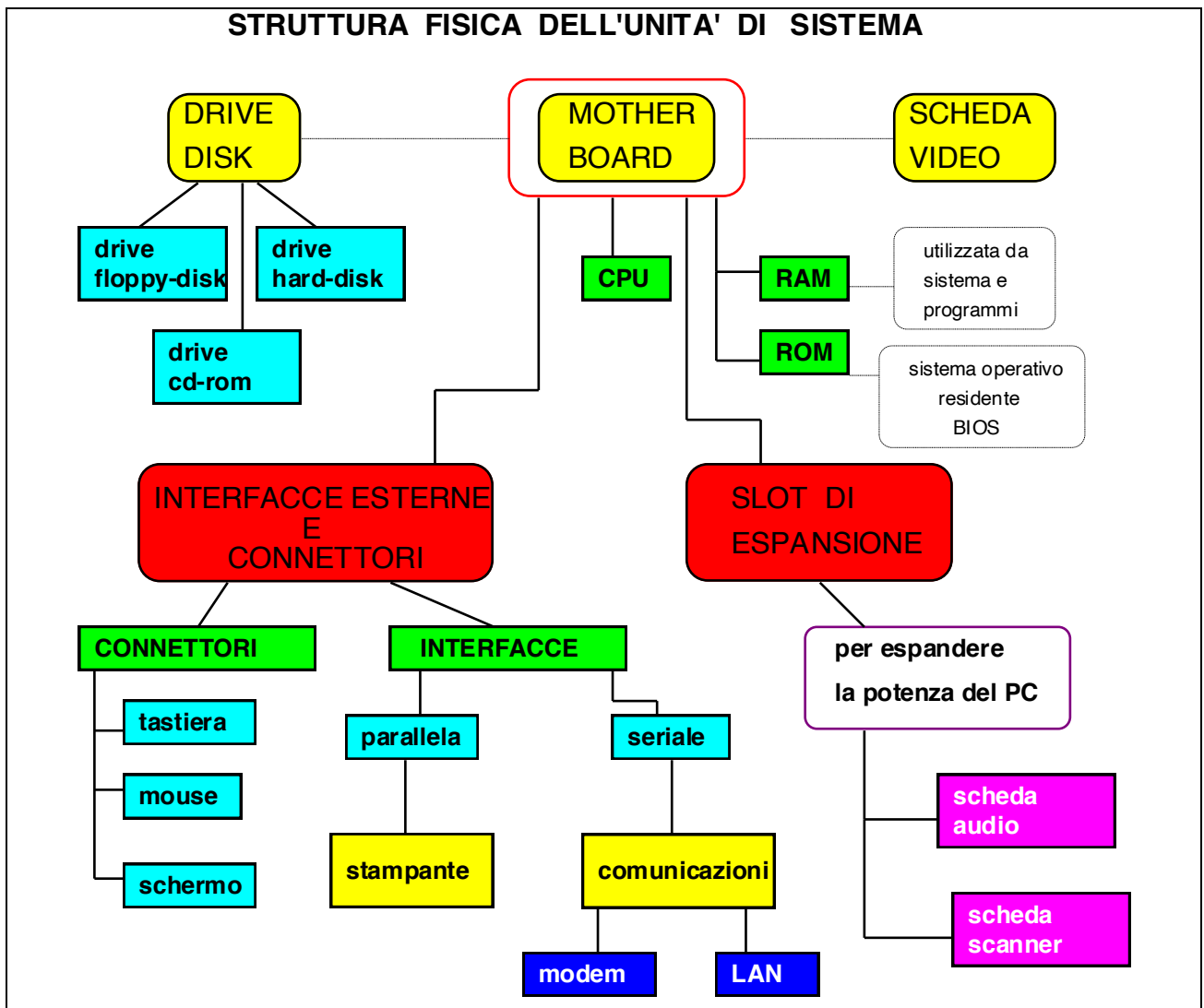


Figura 13 Struttura fisica dell'unità di sistema

BUS

Le informazioni che transitano fra le varie parti del sistema, la memoria e le unità di ingresso e uscita viaggiano per mezzo di un insieme di fili elettrici detti **bus**. È l'abbreviazione di busbar che letteralmente significa fascio di fili formanti il tracciato per gli impulsi elettrici in modo coordinato. Un bus è un mezzo fisico che trasporta un'informazione da una parte all'altra di un sistema formato da tanti circuiti digitali.

L'informazione è codificata in binario. Un bus è composto da tanti conduttori elettrici, lungo i quali è trasportata l'informazione sotto forma di segnali elettrici binari. Ogni calcolatore contiene molti tipi di bus, essi contribuiscono a determinare la velocità e la potenza di calcolo.

Lo schema riassume l'intera e numerosa famiglia dei bus.

1. BUS INTERNI sono costituiti da piste conduttrici di dimensioni microscopiche presenti su **singolo chip**; figura A
1. 2. BUS INTERNI sono anche quelli che costituiscono i collegamenti interni ad un **microprocessore**; figura A
2. 3. BUS DI COMPONENTI costituiscono il collegamento tra i vari microprocessori, installati su di una singola **scheda**; figura B
3. BACKPLANE BUS ogni scheda ha un connettore di collegamento, per backplane bus si intendono le varie linee di bus collocate sulla **mother board** dove sono installate le varie schede. Le schede così installate formano un **cassetto**; figura C
4. BUS DI SISTEMA sono i collegamenti tra i vari cassette. L'insieme di cassette costituisce il **rack**, detto anche **cabinet** che contiene fisicamente il computer.
5. BUS DI INPUT/OUTPUT sono le **prese di interconnessione** per le periferiche.

Poiché il compito di ogni bus è quello di trasferire informazioni, la misura fondamentale di questa capacità è la velocità di trasferimento. L'unità di misura usata è il **baud** (impulsi al secondo) oppure più semplicemente in **bit/s** (bit al secondo)

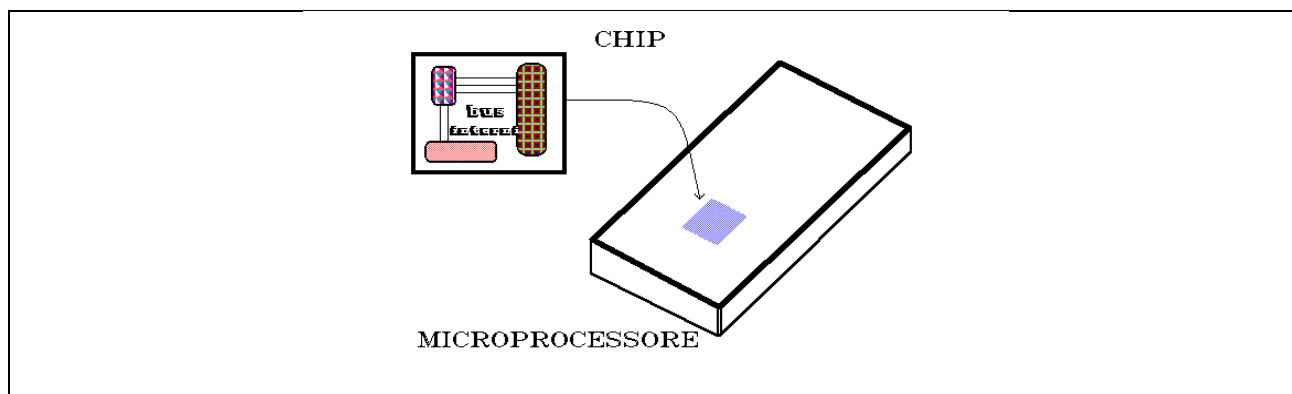


Figura A: visualizza i bus interni.

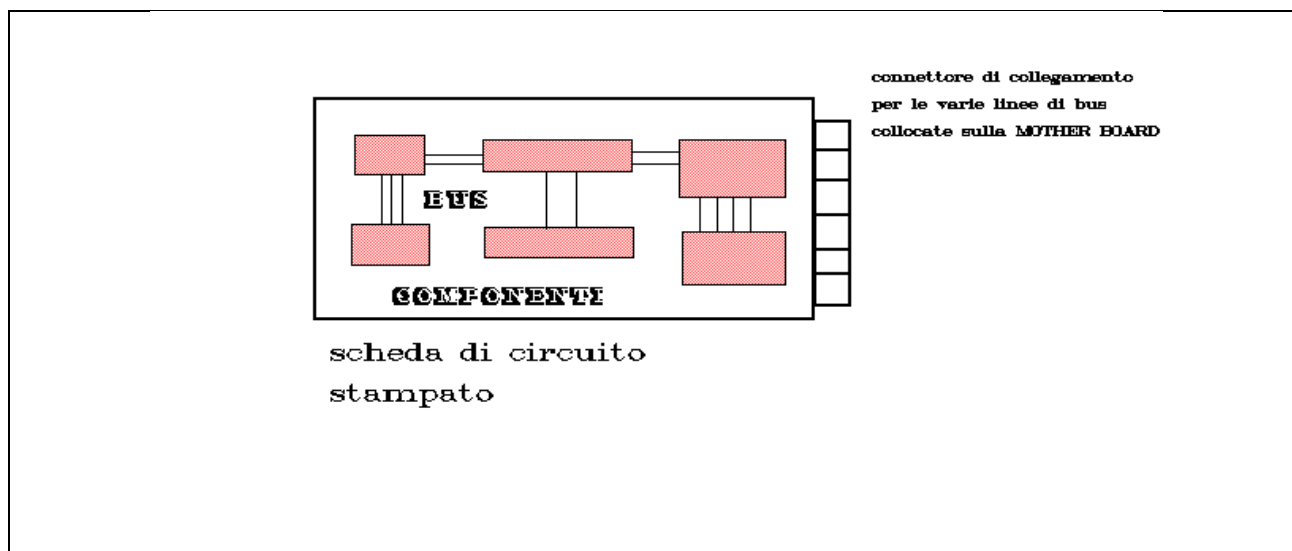


Figura B: inerente ai bus di componenti.

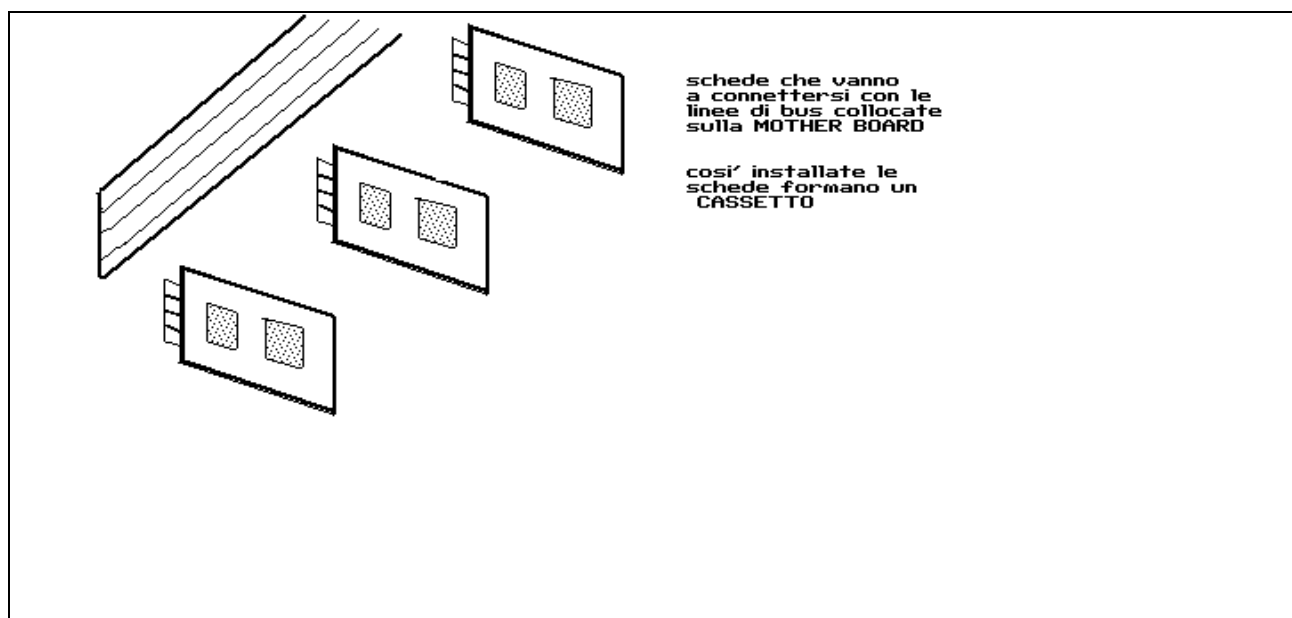


Figura C: le schede, tramite il loro connettore, si collegano alla mother board.

Configurazione di un Personal Computer

Per lavorare con un computer è altresì importante conoscere la configurazione disponibile e le sue possibili estensioni.

La configurazione base di un PC è la seguente:

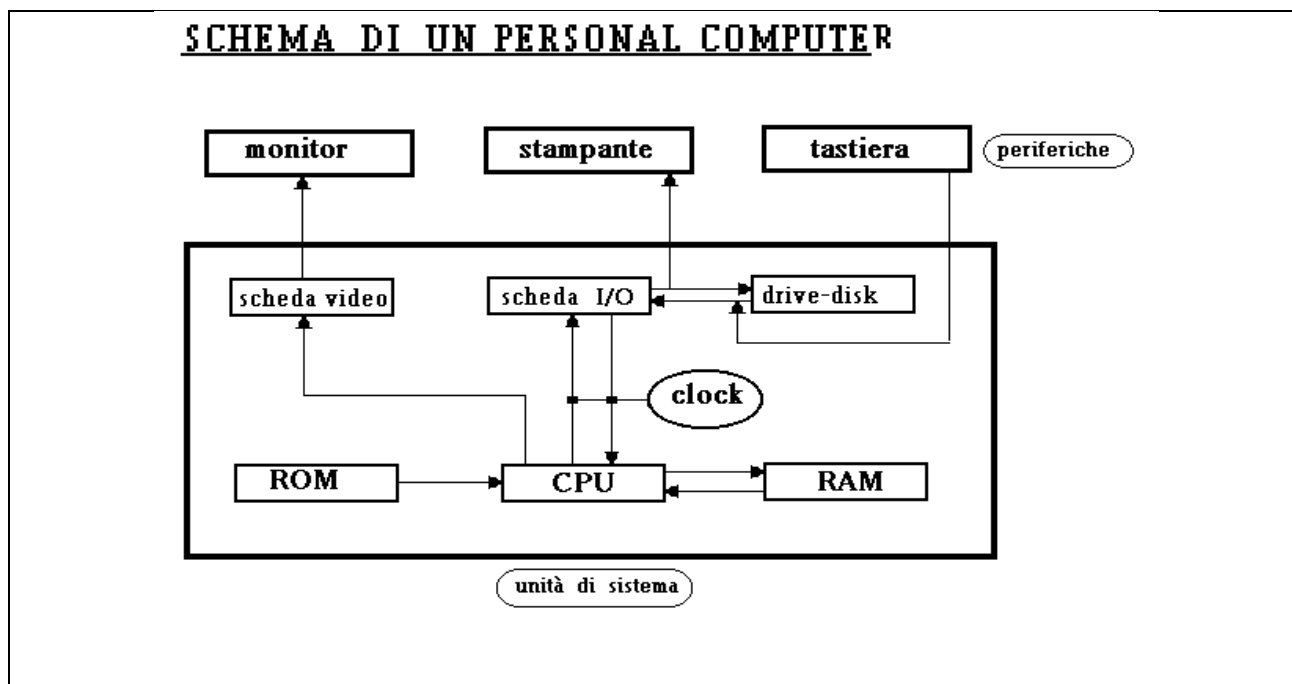


Figura 14 ulteriore schema di un pc

In figura si è operata una separazione fra *unità di sistema* e *periferiche*.

All'interno dell'unità di sistema sono evidenziati i principali gruppi funzionali con i rispettivi flussi di informazioni, all'esterno sono evidenziate invece le tre periferiche essenziali per un corretto e produttivo funzionamento del computer.

Le configurazioni possibili possono essere numerose e rispondere a particolari esigenze operative.

Le periferiche possono essere suddivise in tre categorie:

INPUT	OUTPUT	INPUT/OUTPUT
-------	--------	--------------

INPUT: sono le periferiche di sola entrata come *la tastiera, il mouse, il joystick, la tavoletta grafica, lo scanner per digitalizzare immagini e testo, il CD-ROM, il digitizer, la penna ottica*. Il joystick, la penna ottica, il mouse e la tavoletta grafica con il software applicativo richiesto per farli funzionare, sono strumenti di INPUT in grado di fornire una alternativa alla tastiera. In vario modo permettono lo spostamento del cursore sul video e tramite la pressione di un apposito tasto sono in grado di opzionare una determinata funzione rappresentata in questo modo da una *icona* (immagine simbolica). Un altro strumento che immette nel computer informazioni è lo scanner; esso permette di digitalizzare immagini, testi e fotografie da introdurre poi nel computer e memorizzare per successive elaborazioni.

OUTPUT: sono periferiche di sola uscita come *il monitor, la stampante, il plotter, macchine da cutting*; se si vuole stampare su carta le informazioni elaborate si collega al PC una stampante per i documenti, per i disegni il plotter che è una stampante in possesso di pennini a vari colori; è utilizzato per riprodurre grafici, disegni e progetti e viene adoperato in genere in quegli studi in cui occorre una progettazione grafica. La copia su carta di quanto viene elaborato è chiamata hard-copy in contrapposizione ai dati che compaiono sul monitor che costituiscono la soft-copy. Le macchine da cutting sono delle macchine da taglio, in pratica incidono su materiali di diversa natura come fogli adesivi, plastica ecc. ad esempio per scritte di tipo pubblicitario dove testo e figure sono elaborate con appositi programmi.

INPUT/OUTPUT: sono periferiche di entrata e uscita come i drive che agiscono su *supporti di tipo magnetico e ottico come l'hard-disk, il floppy-disk, e il CD-ROM*. Un ruolo particolare lo svolge il modem che permette il collegamento in rete dei computer, come modem si può anche intendere il fax che può essere inserito come scheda aggiuntiva nell'*unità di sistema*.



Come si comunica con il computer?

La comunicazione con il computer è un bel problema, sia nell'immettere dati e sia nella risposta che viene fornita dalla macchina. In ogni caso si tratta di numeri, noi immettiamo numeri e lui risponde con numeri! I numeri sono espressi nel sistema binario 0 e 1. Come si può immaginare per noi è complicato comunicare in binario e leggere le risposte in binario. Per questo motivo è stato necessario inventare un codice che permettesse l'implementazione dei dati in una forma a noi più intellegibile così come le risposte.

Codice ASCII

Nel 1963 si è dovuto accettare l'idea che serviva un alfabeto comune per tutti i computer con i quali comunicare. Si trovò l'accordo con la creazione del ASCII di seguito:

CODICE ASCII ASCII - American Standard Code for Information Interchange è un codice standard a 7-bit che fu proposto dall'ANSI nel 1963 e diventò definitivo nel 1968. ASCII (si pronuncia "askii") è il codice standard per i microcomputer e consiste di 128 numeri decimali che vanno da 0 a 127. I numeri che vanno da 128 a 255 costituiscono il set di caratteri estesi che comprendono caratteri speciali, matematici, grafici e di lingue straniere.											
Byte	Cod.	Char	Byte	Cod.	Char	Byte	Cod.	Char	Byte	Cod.	Char
00000000	0	Null	00100000	32	Spc	01000000	64	@	01100000	96	-
00000001	1	Start of heading	00100001	33	!	01000001	65	A	01100001	97	a
00000010	2	Start of text	00100010	34	"	01000010	66	B	01100010	98	b
00000011	3	End of text	00100011	35	#	01000011	67	C	01100011	99	c
00000100	4	End of transmit	00100100	36	\$	01000100	68	D	01100100	100	d
00000101	5	Enquiry	00100101	37	%	01000101	69	E	01100101	101	e
00000110	6	Acknowledge	00100110	38	&	01000110	70	F	01100110	102	f
00000111	7	Audible bell	00100111	39	'	01000111	71	G	01100111	103	g
00001000	8	Backspace	00101000	40	(	01001000	72	H	01101000	104	h
00001001	9	Horizontal tab	00101001	41	)	01001001	73	I	01101001	105	i
00001010	10	Line feed	00101010	42	*	01001010	74	J	01101010	106	j
00001011	11	Vertical tab	00101011	43	+	01001011	75	K	01101011	107	k
00001100	12	Form Feed	00101100	44	,	01001100	76	L	01101100	108	l
00001101	13	Carriage return	00101101	45	-	01001101	77	M	01101101	109	m
00001110	14	Shift out	00101110	46	.	01001110	78	N	01101110	110	n
00001111	15	Shift in	00101111	47	/	01001111	79	O	01101111	111	o
00010000	16	Data link escape	00110000	48	0	01010000	80	P	01110000	112	p
00010001	17	Device control 1	00110001	49	1	01010001	81	Q	01110001	113	q
00010010	18	Device control 2	00110010	50	2	01010010	82	R	01110010	114	r
00010011	19	Device control 3	00110011	51	3	01010011	83	S	01110011	115	s
00010100	20	Device control 4	00110100	52	4	01010100	84	T	01110100	116	t
00010101	21	Neg. acknowledge	00110101	53	5	01010101	85	U	01110101	117	u
00010110	22	Synchronous idle	00110110	54	6	01010110	86	V	01110110	118	v
00010111	23	End trans. block	00110111	55	7	01010111	87	W	01110111	119	w
00011000	24	Cancel	00111000	56	8	01011000	88	X	01111000	120	x
00011001	25	End of medium	00111001	57	9	01011001	89	Y	01111001	121	y
00011010	26	Substitution	00111010	58	:	01011010	90	Z	01111010	122	z
00011011	27	Escape	00111011	59	;	01011011	91	[	01111011	123	{
00011100	28	File separator	00111100	60	<	01011100	92	\	01111100	124	
00011101	29	Group separator	00111101	61	=	01011101	93	]	01111101	125	}
00011110	30	Record Separator	00111110	62	>	01011110	94	^	01111110	126	~
00011111	31	Unit separator	00111111	63	?	01011111	95	_	01111111	127	Del

Figura 15 Codice ASCII

In questo codice ogni sequenza di otto bit (un **byte**) corrisponde ad una lettera o un numero o a un controllo di operazioni macchina. Nella versione originale sono 128. Ad ogni byte corrisponde il corrispettivo numero decimale, difatti abbiamo quattro zone formate da tre colonne con intestazione Byte, Cod., Char e corrispondono a:

1. Caratteri di comando;

2. Caratteri speciali;
3. Caratteri dell'alfabeto maiuscoli;
4. Caratteri dell'alfabeto minuscoli.

Sistemi di numerazione ottale e esadecimale

Nel campo informatico per evitare la ridondanza⁵ delle cifre 0 1 del sistema binario si ricorre a due altri sistemi di numerazione, quello ottale e quello esadecimale che “comprimono” le lunghe serie di 0 e 1.

Un sistema di numerazione ottale si basa solo su otto cifre:

0 1 2 3 4 5 6 7

Il principio di costruzione di questi sistemi di numerazione è sempre lo stesso: quello del sistema decimale o del sistema binario. Per il sistema ottale si ha che al numero otto corrisponde 10, al numero nove 11, al numero dieci 12 e così via.

I sistemi impiegati in informatica sono genericamente tre, il sistema binario, ottale ed esadecimale.

sistema decimale	sistema binario	sistema ottale	sistema esadecimale
0	0	0	0
1	1	1	1
2	10	2	2
3	11	3	3
4	100	4	4
5	101	5	5
6	110	6	6
7	111	7	7
8	1000	10	8
9	1001	11	9
10	1010	12	A
11	1011	13	B
12	1100	14	C
13	1101	15	D
14	1110	16	E
15	1111	17	F
16	10000	20	10
17	10001	21	11
18	10010	22	12
19	10011	23	13
20	10100	24	14
21	10101	25	15
22	10110	26	16
23	10111	27	17
24	11000	30	18
25	11001	31	19
26	11010	32	1A
27	11011	33	1B
28	11100	34	1C
29	11101	35	1D
30	11110	36	1E
31	11111	37	1F
32	100000	40	20

Il sistema binario ha due cifre, il sistema ottale otto cifre, l'esadecimale sedici cifre e per la prima volta compaiono anche le lettere dell'alfabeto (maiuscole) A B C D E F. Esempio di conversione tra i vari sistemi. Prendiamo il numero decimale 28 e lo scriviamo in forma polinomiale con potenze di dieci che indicano la posizione della cifra.

$$28 (dec) = 2 \cdot 10^1 + 8 \cdot 10^0 = 28 (dec)$$

$$34 (ott) = 3 \cdot 8^1 + 4 \cdot 8^0 = 28 (dec)$$

$$11100 (bin) = 1 \cdot 2^4 + 1 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 0 \cdot 2^0 = 28 (dec)$$

$$1C (esa) = 1 \cdot 16^1 + 12 \cdot 16^0 = 28 (dec)$$

Scriviamo il numero decimale 28 nel sistema ottale che corrisponde a 34. La forma polinomiale è scritta con potenze di otto per indicare la posizione delle cifre. Nel sistema binario il numero decimale 28 è 11100, in forma polinomiale abbiamo il numero espresso con potenze di due.

Il numero dec 28 in esadecimale corrisponde a 1C nella forma polinomiale si capisce $1 \cdot 16^1 + 12 \cdot 16^0$ la prima parte è comprensibile, mentre si resta perplessi sul dodici per sedici alla 0. Il C corrisponde al 12 in dec.

Figura 16 I tre sistemi numerici adoperati in informatica

⁵ La ridondanza è avere una serie di zero e uno, facilmente riconducibili ad un numero o istruzione.

L'ultima versione della tabella ASCII aggiornata possiede più di 128 caratteri e li si può suddividere in stampabili e non stampabili.



Figura 17 ASCII

La figura 51 rappresenta benissimo, in modo intuitivo ed immediato, cosa rappresenta il codice ASCII e la sua stretta correlazione con i numeri binari.

L'utente digita l'istruzione che vuole assegnare alla macchina tramite la tastiera e quando preme un tasto, ad esempio, la lettera S (maiuscola) il computer la tramuta nel byte corrispondente nella tabella ASCII e lo invia all'elaborazione. La risposta elaborata del byte inviata se operiamo con un word processor sarà la comparsa di **S** sullo schermo.



La Pescheria a Treviso

Binario Ottale Decimale Esadecimale

000 0000	000	0	00	NUL NUL ^@ \0 Null character
000 0001	001	1	01	SOF SOH ^A ☺
000 0010	002	2	02	STX STX ^B ☹
000 0011	003	3	03	ETX ETX ^C ♥
000 0100	004	4	04	EOT EOT ^D ♦
000 0101	005	5	05	ENQ ENQ ^E ♣
000 0110	006	6	06	ACK ACK ^F ♠
000 0111	007	7	07	BEL BEL ^G \a •
000 1000	010	8	08	BS BS ^H \b ▣
000 1001	011	9	09	HT HT ^I \t ○
000 1010	012	10	0A	LF LF ^J \n ■
000 1011	013	11	0B	VT VT ^K \v ♂
000 1100	014	12	0C	FF FF ^L \f ♀
000 1101	015	13	0D	CR CR ^M \r ♪
000 1110	016	14	0E	SO SO ^N 🎵
000 1111	017	15	0F	SI SI ^O ☼
001 0000	020	16	10	DLE DLE ^P ►
001 0001	021	17	11	DC1 DC1 ^Q ◀
001 0010	022	18	12	DC2 DC2 ^R ↑
001 0011	023	19	13	DC3 DC3 ^S !!
001 0100	024	20	14	DC4 DC4 ^T ¶
001 0101	025	21	15	NAK NAK ^U §
001 0110	026	22	16	SYN SYN ^V —
001 0111	027	23	17	ETB ETB ^W ⚡
001 1000	030	24	18	CAN CAN ^X ↑
001 1001	031	25	19	EM EM ^Y ↓
001 1010	032	26	1A	SUB SUB ^Z →
001 1011	033	27	1B	ESC ESC ^[←
001 1100	034	28	1C	FS FS ^\ ⌞
001 1101	035	29	1D	GS GS ^] ↔
001 1110	036	30	1E	RS RS ^^ ▲
001 1111	037	31	1F	US US ^_ ▼
111 1111	177	127	7F	DEL DEL ^? Delete (tasto elimina-bakspace)

Figura 18 Tabella ASCII aggiornata al 1986 riguardante i comandi

Binario	Oct	Dec	Hex	Glifo	Binario	Oct	Dec	Hex	Glifo	Binario	Oct	Dec	Hex	Glifo
010 0000	040	32	20	Spazio	100 0000	100	64	40	@	110 0000	140	96	60	`
010 0001	041	33	21	!	100 0001	101	65	41	A	110 0001	141	97	61	a
010 0010	042	34	22	"	100 0010	102	66	42	B	110 0010	142	98	62	b
010 0011	043	35	23	#	100 0011	103	67	43	C	110 0011	143	99	63	c
010 0100	044	36	24	\$	100 0100	104	68	44	D	110 0100	144	100	64	d
010 0101	045	37	25	%	100 0101	105	69	45	E	110 0101	145	101	65	e
010 0110	046	38	26	&	100 0110	106	70	46	F	110 0110	146	102	66	f
010 0111	047	39	27	'	100 0111	107	71	47	G	110 0111	147	103	67	g
010 1000	050	40	28	(	100 1000	110	72	48	H	110 1000	150	104	68	h
010 1001	051	41	29	)	100 1001	111	73	49	I	110 1001	151	105	69	i
010 1010	052	42	2A	*	100 1010	112	74	4A	J	110 1010	152	106	6A	j
010 1011	053	43	2B	+	100 1011	113	75	4B	K	110 1011	153	107	6B	k
010 1100	054	44	2C	,	100 1100	114	76	4C	L	110 1100	154	108	6C	l
010 1101	055	45	2D	-	100 1101	115	77	4D	M	110 1101	155	109	6D	m
010 1110	056	46	2E	.	100 1110	116	78	4E	N	110 1110	156	110	6E	n
010 1111	057	47	2F	/	100 1111	117	79	4F	O	110 1111	157	111	6F	o
011 0000	060	48	30	0	101 0000	120	80	50	P	111 0000	160	112	70	p
011 0001	061	49	31	1	101 0001	121	81	51	Q	111 0001	161	113	71	q
011 0010	062	50	32	2	101 0010	122	82	52	R	111 0010	162	114	72	r
011 0011	063	51	33	3	101 0011	123	83	53	S	111 0011	163	115	73	s
011 0100	064	52	34	4	101 0100	124	84	54	T	111 0100	164	116	74	t
011 0101	065	53	35	5	101 0101	125	85	55	U	111 0101	165	117	75	u
011 0110	066	54	36	6	101 0110	126	86	56	V	111 0110	166	118	76	v
011 0111	067	55	37	7	101 0111	127	87	57	W	111 0111	167	119	77	w
011 1000	070	56	38	8	101 1000	130	88	58	X	111 1000	170	120	78	x
011 1001	071	57	39	9	101 1001	131	89	59	Y	111 1001	171	121	79	y
011 1010	072	58	3A	:	101 1010	132	90	5A	Z	111 1010	172	122	7A	z
011 1011	073	59	3B	;	101 1011	133	91	5B	[	111 1011	173	123	7B	{
011 1100	074	60	3C	<	101 1100	134	92	5C	\	111 1100	174	124	7C	
011 1101	075	61	3D	=	101 1101	135	93	5D	]	111 1101	175	125	7D	}
011 1110	076	62	3E	>	101 1110	136	94	5E	^	111 1110	176	126	7E	~
011 1111	077	63	3F	?	101 1111	137	95	5F	_					

Figura 19 Tabella ASCII aggiornata al 1986 riguardante i caratteri

La tastiera estesa del Personal Computer

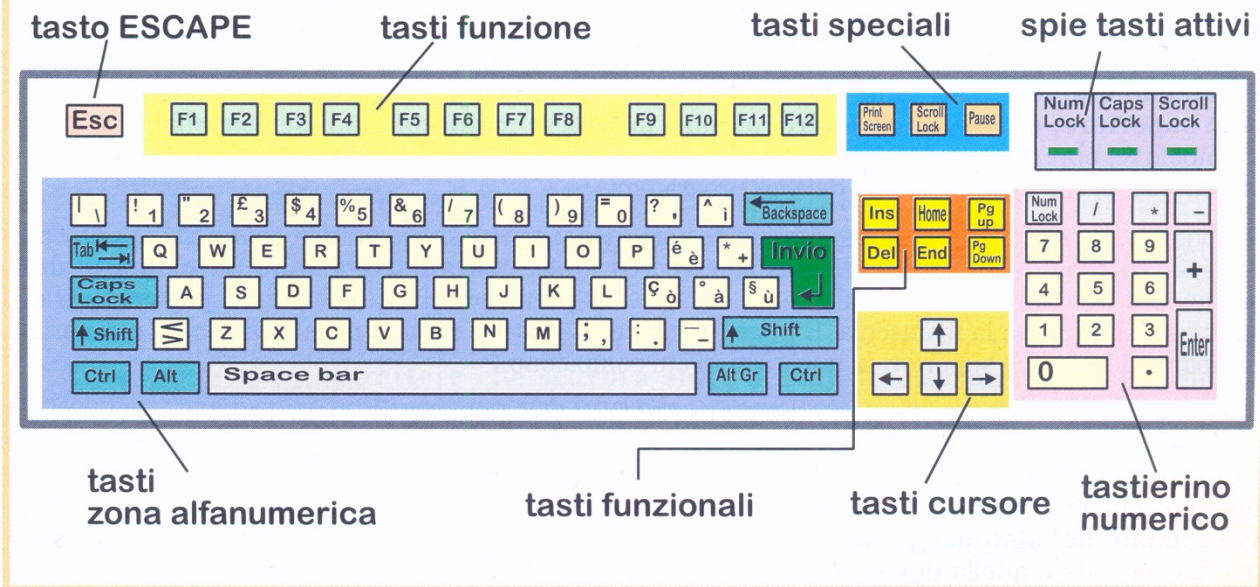


Figura 20 Tastiera PC (SO Windows)