

# Il Microprocessore

## INTRODUZIONE

Il computer che stai usando per leggere questa pagina usa un microprocessore per farlo. Il microprocessore è il cuore di ogni personal computer, sia esso un [laptop](#), un desktop o un server. Ne esistono di vari tipi e marche (Pentium, K6, Sparc o Power PC) ma più o meno tutti fanno le stesse cose. In questo capitolo di [ComeFunziona.net](#) vedremo come sono fatti e cosa li differenzia.



## STORIA DEI MICROPROCESSORI

Il primo microprocessore della storia fu costruito nel 1971. Si trattava di un **Intel 4004**. Non poteva fare granchè, solamente addizioni e sottrazioni (4 bits alla volta). La vera novità era che tutti i componenti erano racchiusi in un singolo chip. Prima di allora infatti gli ingegneri erano costretti a costruire computers usando gruppi di chip e transistors.

Il primo microprocessore ad essere inserito in un Home Computer (nel 1974) fu l'**Intel 8080**, che poteva effettuare operazioni ad 8 [bits](#). Quello che però si impose veramente sul mercato fu il famosissimo **8088** che introdotto nel 1979 dalla Intel venne installato su milioni di PC IBM.

Negli anni successivi lo sviluppo dei microprocessori fu incredibile e seguirono quindi altri modelli sempre della Intel : 80286, 80386, 80486, Pentium, Pentium II, Pentium III e Pentium IV. Tutti questi però usano ancora oggi il disegno originale dell'8088 anche se esguono operazioni ad una velocità superiore di circa 5000 volte !

La tabella successiva aiuta a comprendere le differenze tra i vari modelli.

Nome	Data	Transistors	Microns	Velocità	Bits	MIPS
8080	1974	6.000	6	2 MHz	8	0.64
8088	1979	29.000	3	5 MHz	16	0.33
80286	1982	134.000	1.5	6 MHz	16	1
80386	1985	275.000	1.5	16 MHz	32	5
80486	1989	1.200.000	1	25 MHz	32	20
PENTIUM	1993	3.100.000	0.8	60 MHz	32	100
PENTIUM II	1997	7.500.000	0.35	233 MHz	32	~300
PENTIUM III	1999	9.500.000	0.25	450 MHz	32	~510
PENTIUM IV	2000	42.000.000	0.18	2.5 GHz	32	~1.700

I dati contenuti nella tabella sono i seguenti:

- **Data** - E' l'anno in cui il microprocessore fu lanciato sul mercato. C'è da dire che molti di essi furono reintrodotti con lo stesso nome ma con miglioramenti sulla velocità del clock.

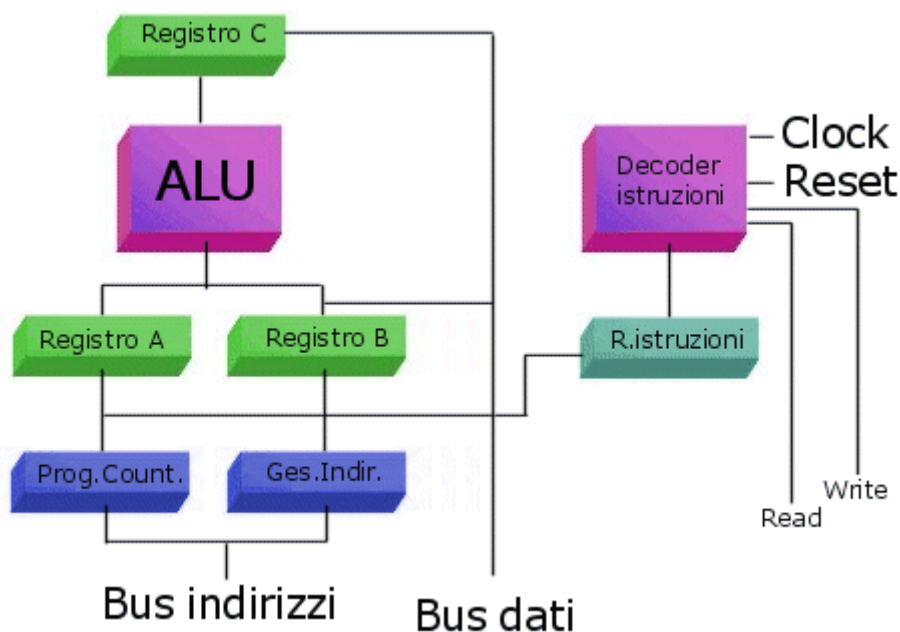
- **Transistors** - E' il numero di transistors presente nel singolo Microprocessore. E' evidente come nel corso degli anni il numero sia cresciuto incredibilmente.
- **Microns** - E' la larghezza di ogni singolo circuito (1 Micron = 1 milionesimo di metro)
- **Velocità** - E' l'espressione in Hertz della velocità del microprocessore.
- **Bits** - E' il numero di bits che la ALU (Arithmetic Logical Unit) può gestire. Ovviamente questo parametro influenza il numero di istruzioni necessarie per i calcoli. Se dobbiamo sommare o sottrarre 2 numeri a 32 bit, un 8080 ad 8 bit eseguirà 4 operazioni mentre un Pentium IV ne eseguirà solo 1.
- **MIPS** - E' l'acronimo di Million of Instruction Per Second, ovvero quante milioni di operazioni al secondo il microprocessore riesce ad eseguire.

## DENTRO AD UN MICROPROCESSORE

Un microprocessore, esegue una serie di istruzioni che dicono la processore cosa fare. A seconda delle istruzioni il processore compie fondamentalmente 3 operazioni:

- Usando la ALU (**Arithmetic Logical Unit**) può eseguire operazioni matematiche come somme, sottrazioni, divisioni e moltiplicazioni.
- Può spostare dati da una cella di memoria all'altra.
- Può prendere decisioni e spostare l'esecuzione del programma da un punto all'altro dello stesso.

Nella figura seguente viene schematizzato un semplice microprocessore che può eseguire le funzioni di base appena elencate.



Nei componenti riportati nella figura è possibile identificare:

- **Bus Indirizzi** - Può essere a 8, 16 o 32 bit e serve per inviare un indirizzo alla memoria.
- **Bus Dati** - Può essere a 8, 16 o 32 bit e serve per inviare e ricevere dati dalla memoria.
- **Clock Line** - Che stabilisce i cicli di clock del processore. Ad ogni ciclo viene eseguita una operazione.
- **Reset Line** - Che può resettare il program counter, ovvero far ripartire il programma dall'inizio.
- **Registri** - Che possono contenere informazioni a 8 bit. Sono le variabili del processore.

- **Registro istruzioni** - Contiene l'istruzione correntemente in esecuzione.
- **Decoder istruzioni** - Provvede a decodificare l'istruzione presente nell'apposito registro in veri e propri segnali da inviare ai componenti della CPU.
- **Linee Read e Write** - Provvedono a leggere e scrivere dei valori in memoria.

## ROM E RAM

I Bus dati e indirizzi sono collegati alla RAM e alla ROM del processore. Supponendo che i nostri bus siano a 8 bit, il processore potrà indirizzare  $2^8 = 256$  [bytes](#) di memoria e potrà leggere e scrivere 8 bytes alla volta.

[ROM](#) (Read Only Memory). Un chip ROM è programmato con dei dati permanenti, ovvero con dei bytes già settati che rimangono "vivi" anche a processore spento. Il processore ovviamente può solamente leggere questi bytes ma non scriverli.

[RAM](#) (Random Access Memory). La RAM contiene bytes di informazione che il microprocessore può leggere e scrivere. Il problema delle RAM è che non mantengono il valore inserito una volta che il processore viene spento.

Tutti i computer contengono sempre una ROM (grande o piccola) che viene chiamata BIOS ( **Basic Input/Output System** ). Appena il microprocessore viene acceso, esso comincia ad eseguire le istruzioni contenute nel BIOS che generalmente testano l'hardware della macchina e poi accedono all'Hard Disk per leggere il boot sector (Vedi [Come funziona un Hard Disk](#) per maggiori dettagli). Questo boot sector è un altro piccolo programma che viene letto dal BIOS e memorizzato nella RAM per essere eseguito. L'esecuzione non è altro che la lettura di altri piccoli programmi che formano l'intero [sistema operativo](#).

## LE ISTRUZIONI DEL MICROPROCESSORE

Anche il più piccolo e semplice microprocessore (come quello che abbiamo visto in precedenza) deve avere un set di istruzioni che può eseguire. Ogni istruzione è in pratica una serie di bits che vengono caricati di volta in volta nel registro istruzioni. Ovviamente è quasi impossibile memorizzare delle sequenze di 0 e 1 e ricordarle, quindi per facilitare la programmazione dei microprocessori vengono ridefiniti usando delle brevi parole (short words).

Questa collezione di brevi parole formano il linguaggio **assembler**. Attraverso un traduttore assembler è possibile successivamente trasformare ogni singola "parola" in sequenze di bits che vengono inserite in memoria per essere eseguite dal microprocessore.

Ecco una sequenza di possibili parole che potrebbero essere create per il nostro microprocessore di esempio:

- **LOADA mem** - Carica il registro A con un valore di memoria
- **LOADB mem** - Carica il registro B con un valore di memoria
- **CONB con** - Carica il registro B con una costante
- **SAVEB mem** - Salva il registro B in un indirizzo di memoria
- **SAVEC mem** - Salva il registro C in un indirizzo di memoria
- **ADD** - Somma A e B e mette il risultato in C
- **SUBD** - Sottrae B da A e mette il risultato in C
- **MUL** - Moltiplica A e B e mette il risultato in C
- **DIV** - Divide A con B e mette il risultato in C

- **COMP** - Confronta A e B e mette il risultato in C
- **JUMP addr** - Salta ad un indirizzo
- **JEQ addr** - Salta ad un indirizzo se uguale
- **JNEQ addr** - Salta ad un indirizzo se non uguale
- **JG addr** - Salta ad un indirizzo se maggiore
- **JL addr** - Salta ad un indirizzo se minore
- **STOP** - Ferma l'esecuzione

Ecco un esempio di programma in Assembler:

```
// Assume a is at address 128
// Assume F is at address 129
0  CONB 1      // a=1;
1  SAVEB 128
2  CONB 1      // f=1;
3  SAVEB 129
4  LOADA 128   // if a > 5 the jump to 17
5  CONB 5
6  COM
7  JG 17
8  LOADA 129   // f=f*a;
9  LOADB 128
10 MUL
11 SAVEC 129
12 LOADA 128   // a=a+1;
13 CONB 1
14 ADD
15 SAVEC 128
16 JUMP 4      // loop back to if
17 STOP
```

Come abbiamo già detto ogni singola parola deve essere definita nella ROM con il corrispondente valore numerico (detto **opcode**). Ecco un esempio di codifica:

```
LOADA - 1
LOADB - 2
CONB - 3
SAVEB - 4
SAVEC mem - 5
ADD - 6
SUB - 7
MUL - 8
DIV - 9
COM - 10
JUMP addr - 11
JEQ addr - 12
JNEQ addr - 13
JG addr - 14
JGE addr - 15
JL addr - 16
JLE addr - 17
```

STOP - 18

Il nostro programma, codificato opportunamente sostituendo ad ogni parola il corrispondente opcode diventerà così:

```
// Assume a is at address 128
// Assume F is at address 129
Addr opcode/value
0 3 // CONB 1
1 1
2 4 // SAVEB 128
3 128
4 3 // CONB 1
5 1
6 4 // SAVEB 129
7 129
8 1 // LOADA 128
9 128
10 3 // CONB 5
11 5
12 10 // COM
13 14 // JG 17
14 31
15 1 // LOADA 129
16 129
17 2 // LOADB 128
18 128
19 8 // MUL
20 5 // SAVEC 129
21 129
22 1 // LOADA 128
23 128
24 3 // CONB 1
25 1
26 6 // ADD
27 5 // SAVEC 128
28 128
29 11 // JUMP 4
30 8
31 18 // STOP
```

Il programma che abbiamo appena visto è in realtà una semplice routine che calcola il fattoriale di un numero. In altri linguaggi (Visual Basic o C) l'avremmo scritta con 10 o al massimo 15 righe. In assembler come visto ne occorrono 17 che poi diventano 31 bytes in ROM/RAM.

Il componente del microprocessore che poi legge e traduce questi opcode in una serie di segnali che muovono i differenti componenti del processore è l' **instruction decoder**.

## FEDERICO FAGGIN

Chi è costui vi chiederete? E cosa centra con i microprocessori? Ecco la sua storia: Faggin è nato a Vicenza il primo dicembre 1941. Diplomatosi perito industriale nel 1960, si è

laureato in fisica a Padova cinque anni dopo. Nel 1968 fu assunto dalla Fairchild Camera Instruments come ricercatore e messo a capo di un team che lavorava sui semiconduttori a Palo Alto (California).

Nel 1970 passò alla Intel e con le sue intuizioni iniziò un cambiamento epocale nel mondo. È infatti nel 1971 che ha inizio la 'seconda rivoluzione industriale'.

Diversamente da quello a vapore della prima rivoluzione, il suo 'motore' è costituito dalla straordinaria invenzione del microprocessore o MPU (MicroProcessing Unit), ad opera di tre ingegneri della Intel di Santa Clara, **Federico Faggin** e gli americani **Marcian Edward Hoff Jr.** e **Stanley Mazer**, che riuscirono a concentrare su una piastrina di 4 millimetri per 3 un 'supercircuito integrato' (che venne soprannominato 'miracle chip') contenente ben 2.250 transistor che costituivano tutti i componenti di una unità di elaborazione: 'cervello', memoria d'entrata e di uscita.

La spinta alla realizzazione del primo microprocessore al mondo fu la richiesta della società giapponese Busicom di sviluppare la parte elettronica di una calcolatrice da tavolo. Hoff riprogettò l'intero il circuito e invece di 12 chip ne utilizzò solo uno che conteneva tutta l'unità centrale di elaborazione (CPU, Central Processing Unit), oltre ai due per la memoria **RAM** (Random Access Memory) e per quella **ROM** (Read-Only Memory). Lo schema di base del primo microprocessore venne messo a punto da Hoff e da Stan Mazer, mentre il compito di tradurre questa intuizione in una macchina funzionante fu di Faggin, anche lui, come gli altri due, transfuga dalla Fairchild. La realizzazione elettronica dello schema eseguita da Faggin portò alla realizzazione del primo microprocessore: l'Intel 4004. Per la loro invenzione, Faggin, Hoff e Mazer avranno un posto d'onore nella National Inventor's Hall of Fame degli Usa.

Il microprocessore riunisce in un circuito integrato relativamente semplice, ma dall'architettura già perfettamente matura e chiaramente predisposta per ulteriori sviluppi, tutti gli elementi (aritmetici, logici e di controllo) indispensabili per un elaboratore. Già al momento della sua nascita, la capacità di elaborazione, 60 mila operazioni al secondo, è superiore al gigantesco ENIAC a valvole del 1946 o ad un computer IBM dei primi anni '60 con una unità centrale grande come un tavolino.

Per lo sviluppo del microprocessore 4004, la Intel - fondata nel 1968 da un gruppo di entusiasti giovani ricercatori e di docenti con a capo Robert Noyce e Gordon Moore - spese solo 150 mila dollari. Oggi la Intel è il maggiore produttore al mondo e ciò conferma che l'innovazione non è solo il prodotto di ingenti investimenti, ma il risultato di applicazione e creatività di ricercatori ben preparati.

Nel 1972 Faggin realizzò il microprocessore 8008, il primo chip da 8 bit di uso universale. L'8008, con la prima memoria statica, è in grado di conservare i dati sino a quando non viene interrotta l'alimentazione elettrica. Su questo chip gli ingegneri Nat Wadsworth e Robert Findley realizzarono il primo microcomputer, che fu prodotto in serie in scatola di montaggio dalla Scelbi e venduto per corrispondenza a 440 dollari. Anche la Digital realizzò nel 1974 con lo stesso microprocessore un microcomputer su un'unica scheda, ma a livello industriale non intuì il formidabile avvenire dei piccoli calcolatori, per continuare a dedicarsi ai minicalcolatori aziendali.

Il microprocessore 8008 trovò immediatamente applicazioni nei più disparati settori, dal controllo dei semafori stradali a quello delle emissioni di gas di scappamento delle auto, dagli strumenti scientifici ai giochi elettronici e alle macchine 'intelligenti' di tutti i tipi. Nel '74 Faggin lasciò la Intel e si mise in proprio fondando a Cupertino la Zilog, dove mise a punto lo Z-80,

Provenienza: <http://www.comefunziona.net/articolo.asp?Ogg=BitsBytes>

uno dei chip più popolari mai realizzati. Anche il nome Zilog fu inventato dallo stesso Faggin: la lettera zeta, ultima dell'alfabeto, stava ad indicare l'ultimo grido del campo dei microcircuiti, la 'i' per integrated, e 'log' per logico.

## OVERCLOCKING

Overclockare vuol dire in breve aumentare la frequenza di funzionamento di un componente elettronico, ovvero farlo lavorare più velocemente di quanto sia stato progettato.

Se riusciamo a far lavorare ad esempio un processore a **166MHz** nonostante la sua frequenza di default sia **133MHz** avremo ottenuto un overclocking e disporremo di una macchina leggermente più performante.

A differenza di quanto succede per il software in cui non c'è possibilità legale di modificare i codici sorgenti, per quanto riguarda i processori, l'overclocking è un'operazione del tutto legale visto che una volta acquistato il chip ne possiamo disporre a piacimento. Ovviamente modificare il chip significa perdere i diritti di garanzia su di esso!

L'overclocking, aumenta in pratica la frequenza del clock, e questa operazione provoca solitamente un innalzamento della temperatura interna che rende necessario un adeguato sistema di raffreddamento aggiuntivo. Una volta dimensionato il raffreddamento è necessario sottoporre il chip modificato ad una serie di stress test su differenti applicazioni per determinarne la stabilità operativa.

Se tutto funziona correttamente avremo guadagnato in termini di velocità. In caso contrario potremmo avere un microprocessore instabile oppure addirittura un microprocessore bruciato dall'alta temperatura.