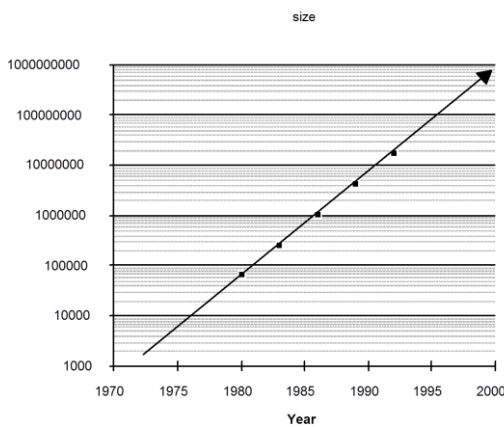


# Valutazione delle prestazioni

## Trend tecnologico: Capacità della Memoria

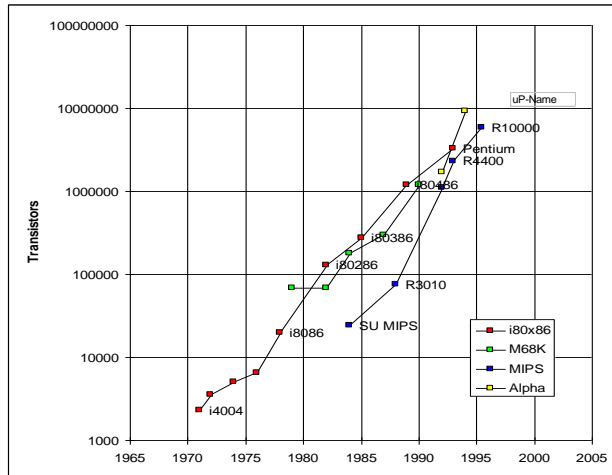


### Capacità chip DRAM

DRAM	
Year	Size
1980	64 Kb
1983	256 Kb
1986	1 Mb
1989	4 Mb
1992	16 Mb
1996	64 Mb
1999	256 Mb
2002	1 Gb

**Incremento 1,4 per anno  
4000X dal 1980**

# Trend tecnologico: Densità Microprocessori



**2X  
transistor/Chip  
ogni 1,5 anni**

## Trend tecnologici

### ▪ Processore

- Densità Logica: circa 30% per anno
- Frequenza Clock : circa 20% per anno

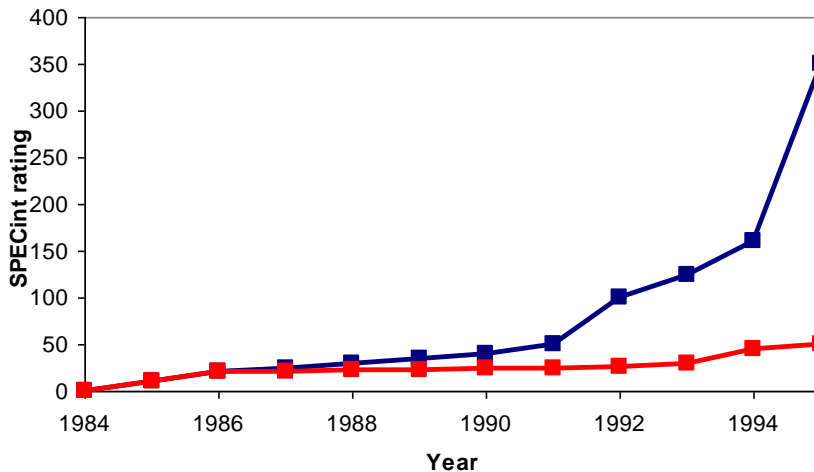
### ▪ Memoria

- Capacità DRAM capacity: circa 60% per anno
- Velocità Memoria: circa 10% per anno
- Costo per bit: riduzione di circa il 25% per anno

### ▪ Dischi

- Capacità: circa 60% per anno
- Larghezza di banda della rete:
  - Aumenta di più del 100% per anno!

## Trend delle prestazioni



## Approccio quantitativo

- L'incremento delle prestazioni è superiore a quello tecnologico.
- Ciò è stato possibile per l'affermazione di un nuovo approccio nella progettazione:
  - **approccio quantitativo** ovvero basato su misure.
- **Principio da seguire nella progettazione:**
  - Rendere veloce il caso più frequente

## Approccio quantitativo

- Una diretta conseguenza di questo nuovo approccio è il passaggio da calcolatori con un numero elevato di istruzioni anche molto complesse (approccio CISC), a calcolatori con un ridotto insieme di istruzioni (approccio RISC)
- Da misure sul comportamento dei programmi si è visto che:
  - l'80% delle istruzioni eseguite corrispondeva al solo 20% del repertorio.
  - ⇒*conviene investire nella riduzione dei tempi di esecuzione di quel 20%, anziché aggiungere raffinate istruzioni, quasi mai usate, ma responsabili dell'allungamento del tempo di ciclo di macchina*
  - ⇒*conviene costruire processori molto veloci, necessariamente con repertori semplici, e contare sull'ottimizzazione del compilatore*

## Indici prestazionali

- Tempo di risposta (o tempo di esecuzione o latenza)
  - Tempo tra l'inizio e il completamento di un lavoro o compito elaborativo
  - ✓Durata dell'esecuzione del mio programma
  - ✓Attesa per l'accesso ad un sito web
- Throughput
  - Ammontare complessivo di lavoro svolto in un dato tempo
  - ✓Numero di programmi eseguiti nell'unità di tempo
  - ✓Numero di lavori (job, transazioni, interrogazioni a basi di dati) svolti nell'unità di tempo
  - ✓Numero di programmi eseguibili da una macchina contemporaneamente

## Il tempo di CPU

- Il **tempo** è la misura delle prestazioni di un computer
  - Il computer che svolge la stessa quantità di lavoro nel minore tempo è il più veloce
- **Tempo di risposta** rappresenta la latenza per il completamento di un lavoro
  - Includendo accessi a disco, accessi a memoria, attività di I/O
- **Tempo di CPU** rappresenta il tempo speso dalla CPU per eseguire il programma dato
  - **Non** include il tempo di attesa per I/O o per l'esecuzione di altri programmi
  - Comprende il **tempo utente di CPU** (tempo speso dalla CPU per eseguire le linee di codice che stanno nel nostro programma) + **tempo di CPU di sistema** (speso dal sistema operativo per eseguire i compiti richiesti dal programma)

## Tempo di CPU

Tempo di CPU = cicli di clock della CPU × durata periodo di clock

$$\text{Tempo di CPU} = \frac{\text{cicli di clock della CPU}}{\text{frequenza di clock}}$$

## Cicli di clock per Istruzione (CPI)

- In genere, istruzioni di tipo diverso richiedono quantità diverse di tempo
  - La moltiplicazione richiede più tempo dell'addizione
  - L'accesso alla memoria richiede più tempo dell'accesso ai registri
- Fissata la durata del ciclo di clock, varia il numero di cicli di clock richiesti dalle diverse istruzioni
- Si può calcolare il numero medio di cicli di clock per istruzione di un dato programma

$$\text{CPI} = \frac{\text{cicli di clock della CPU per eseguire il programma}}{\text{numero di istruzioni eseguite}}$$

## Tempo di CPU

- Indicati con:
  - $N_{ist}$ , il numero di istruzioni eseguite;
  - $CPI$ , il numero medio di cicli di clock per istruzione;
  - $T$ , il periodo del clock;
  - $F$ , la frequenza

$$T_{CPU} = N_{ist} * CPI / f = N_{ist} * CPI * T$$

## Tempo di CPU: $N_{ist}$

$$T_{CPU} = N_{ist} * CPI / f = N_{ist} * CPI * T$$

- $N_{ist}$  dipende dal repertorio di istruzioni e dal grado di ottimizzazione del compilatore.
  - Compilatori diversi possono dare luogo a  $N_{ist}$  diversi
  - Uno stesso compilatore che genera codice per due macchine diverse, darà  $N_{ist}$  diversi
  - Un repertorio CISC favorisce la riduzione del numero di istruzioni

## Tempo di CPU: la frequenza $f$

$$T_{CPU} = N_{ist} * CPI / f = N_{ist} * CPI * T$$

- $f$  (T) è legata alla tecnologia e all'organizzazione architeturale della CPU
  - Oggi 1500÷3000 MHz sono la norma
  - Istruzioni complesse richiedono di norma frequenze di più basse
  - Istruzioni semplici (RISC) permettono di diminuire i ritardi di propagazione nella logica di controllo e, quindi, di diminuire l'ampiezza del periodo di clock.

## Tempo di CPU: il CPI

$$T_{\text{CPU}} = N_{\text{ist}} * \text{CPI} / f = N_{\text{ist}} * \text{CPI} * T$$

•CPI dipende dall'architettura e dal repertorio delle istruzioni

–Istruzioni semplici richiedono un minor numero di cicli.

–Attraverso tecniche come la pipeline è possibile portare CPI ad un valore molto vicino ad 1.

–L'aggiunta di più unità di esecuzione in parallelo (macchine superscalari) permette di rendere CPI minore di 1.

## Tempo di CPU

Se indichiamo con:

– $N_i$ , il numero di volte in cui l'istruzione  $I_i$  viene eseguita in un programma,

– $\text{CPI}_i$ , il numero di cicli di clock richiesto della  $I_i$

– $n$ , numero di istruzioni diverse eseguite

avremo:

$$\text{cicli di clock della CPU} = \sum_{i=1}^n (\text{CPI}_i \times N_i)$$

•Questa formula può essere usata per esprimere il tempo di CPU come

$$T_{\text{CPU}} = \sum_{i=1}^n (\text{CPI}_i \times N_i) \times T = N_{\text{ist}} * \sum_{i=1}^n \left( \text{CPI}_i \times \frac{N_i}{N_{\text{ist}}} \right) \times T$$



# Tempo di CPU

•Indicato con

$$- f_i = N_i / N_{\text{ist}}$$

otterremo:

$$T_{\text{CPU}} = N_{\text{ist}} * \sum_{i=1}^n (\text{CPI}_i \times f_i) \times T$$

$$\text{CPI} = \sum_{i=1}^n (\text{CPI}_i \times f_i)$$

## Esempio

Si consideri un calcolatore in grado di eseguire le istruzioni riportate in tabella.

Calcolare CPI e il tempo di CPU per eseguire un programma composto da 100 istruzioni supponendo di usare una frequenza di clock pari a 500 MHz.

Tipo	Frequenza	CPI <sub>i</sub>
ALU	43%	1
Load	21%	4
Store	12%	4
Branch	12%	2
Jump	12%	2

**Soluzione**

$$\text{CPI} = 1 \cdot 0.43 + 4 \cdot 0.21 + 4 \cdot 0.12 + 2 \cdot 0.12 + 2 \cdot 0.12 = 2.23$$

$$T_{\text{CPU}} = IC * \text{CPI} / f_{\text{CK}} = 100 * 2.23 * 1 / (500 * 10^6) = 446 \text{ ns}$$

## Misura delle prestazioni

$$T_{\text{CPU}} = N_{\text{IST}} * \text{CPI} * T$$

- Processore1:  $N_{\text{IST1}} = N$ ;  $\text{CPI}_1 = 3$ ;  $T_1 = T$
- Processore2:  $N_{\text{IST2}} = 2,5N$ ;  $\text{CPI}_2 = 2$ ;  $T_2 = T/2$

- $T_{\text{CPU1}} = N_{\text{IST1}} * \text{CPI}_1 * T_1 = N * 3 * T$
- $T_{\text{CPU2}} = N_{\text{IST2}} * \text{CPI}_2 * T_2 = 2,5N * 2 * T/2 = 2,5 * N * T$

$$T_{\text{CPU1}} > T_{\text{CPU2}}$$

## Misura delle prestazioni

$$T_{\text{CPU}} = N_{\text{IST}} * \text{CPI} * T$$

- Processore1:  $N_{\text{IST1}} = N$ ;  $\text{CPI}_1 = 10$ ;  $T_1 = T/4$
- Processore2:  $N_{\text{IST2}} = 2N$ ;  $\text{CPI}_2 = 1$ ;  $T_2 = T$

- $T_{\text{CPU1}} = N_{\text{IST1}} * \text{CPI}_1 * T_1 = N * 10 * T/4 = 2,5 * N * T$
- $T_{\text{CPU2}} = N_{\text{IST2}} * \text{CPI}_2 * T_2 = 2N * 1 * T = 2 * N * T$

$$T_{\text{CPU1}} > T_{\text{CPU2}}$$