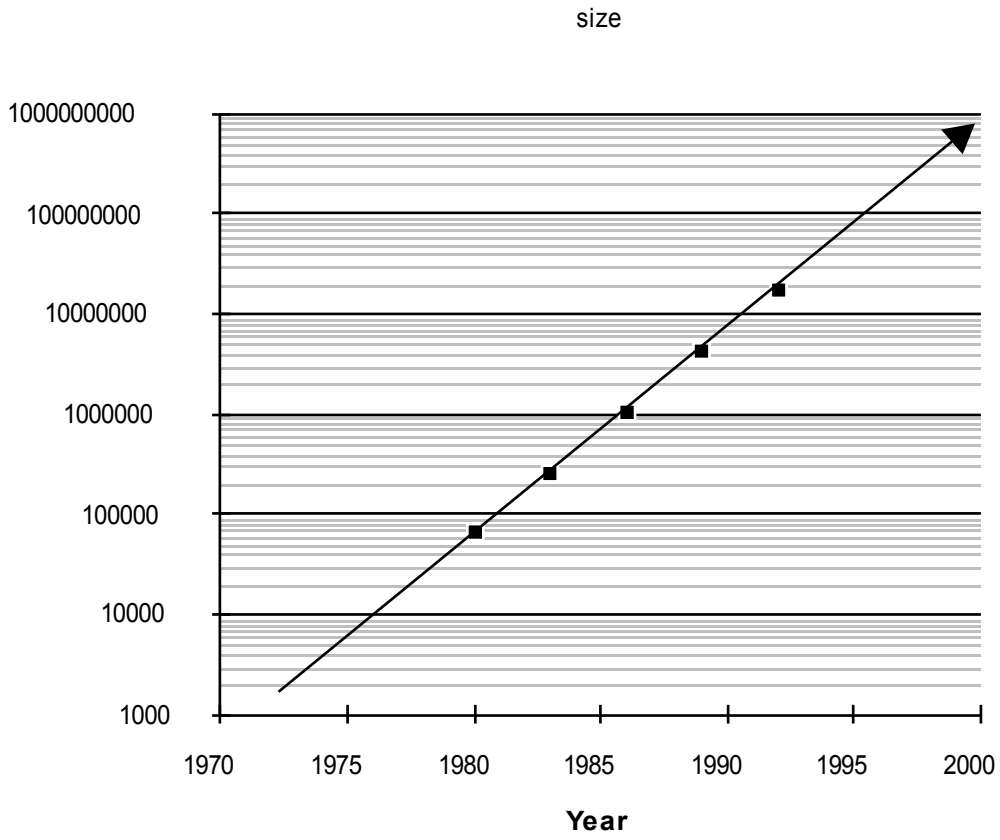
Valutazione delle prestazioni dei Calcolatori Elettronici

Trend tecnologico: Capacità della Memoria



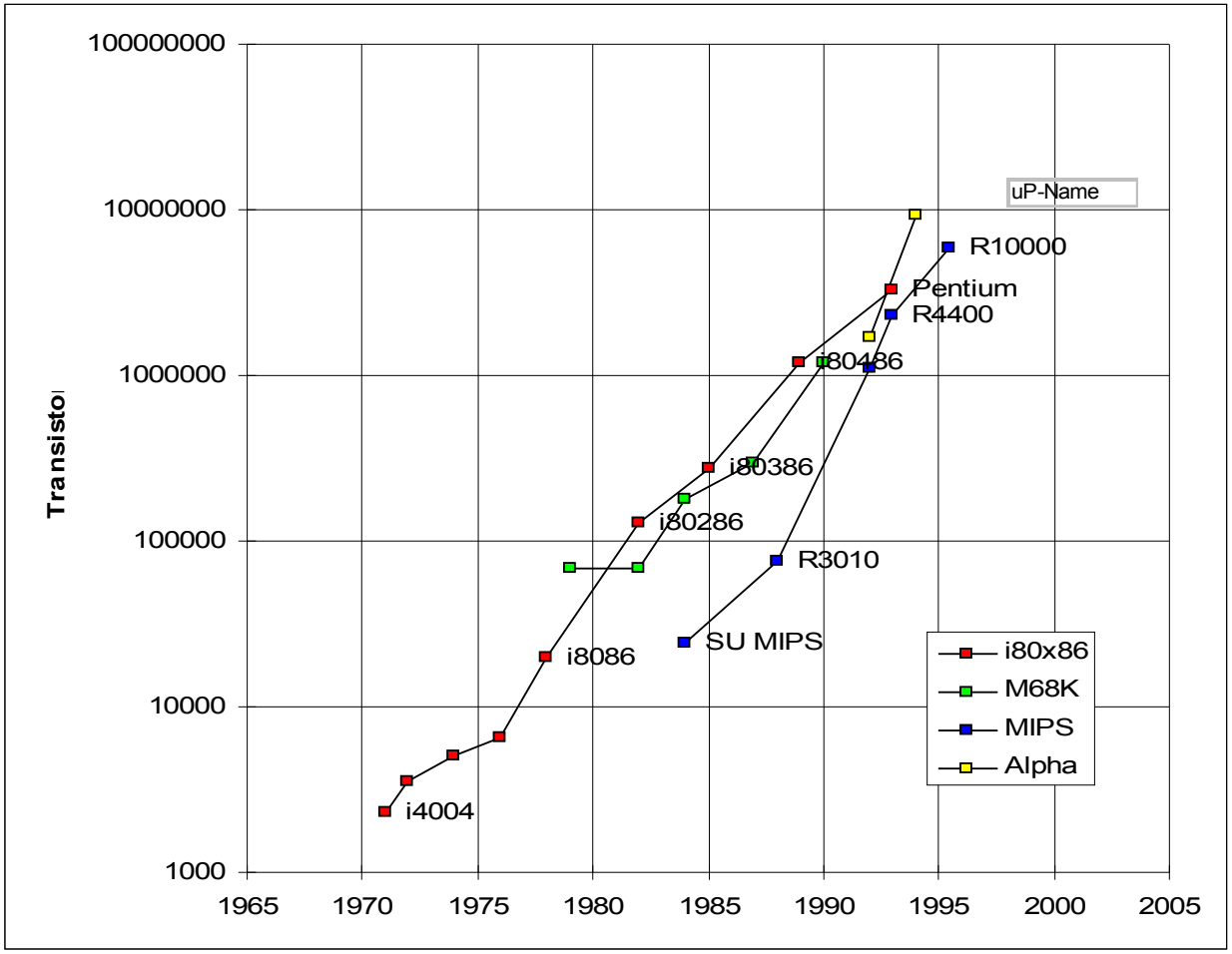
Capacità chip DRAM

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



Alpha 21264: 15 milion

Pentium Pro: 5.5 million

PowerPC 620: 6.9 million

Alpha 21164: 9.3 million

Sparc Ultra: 5.2 million

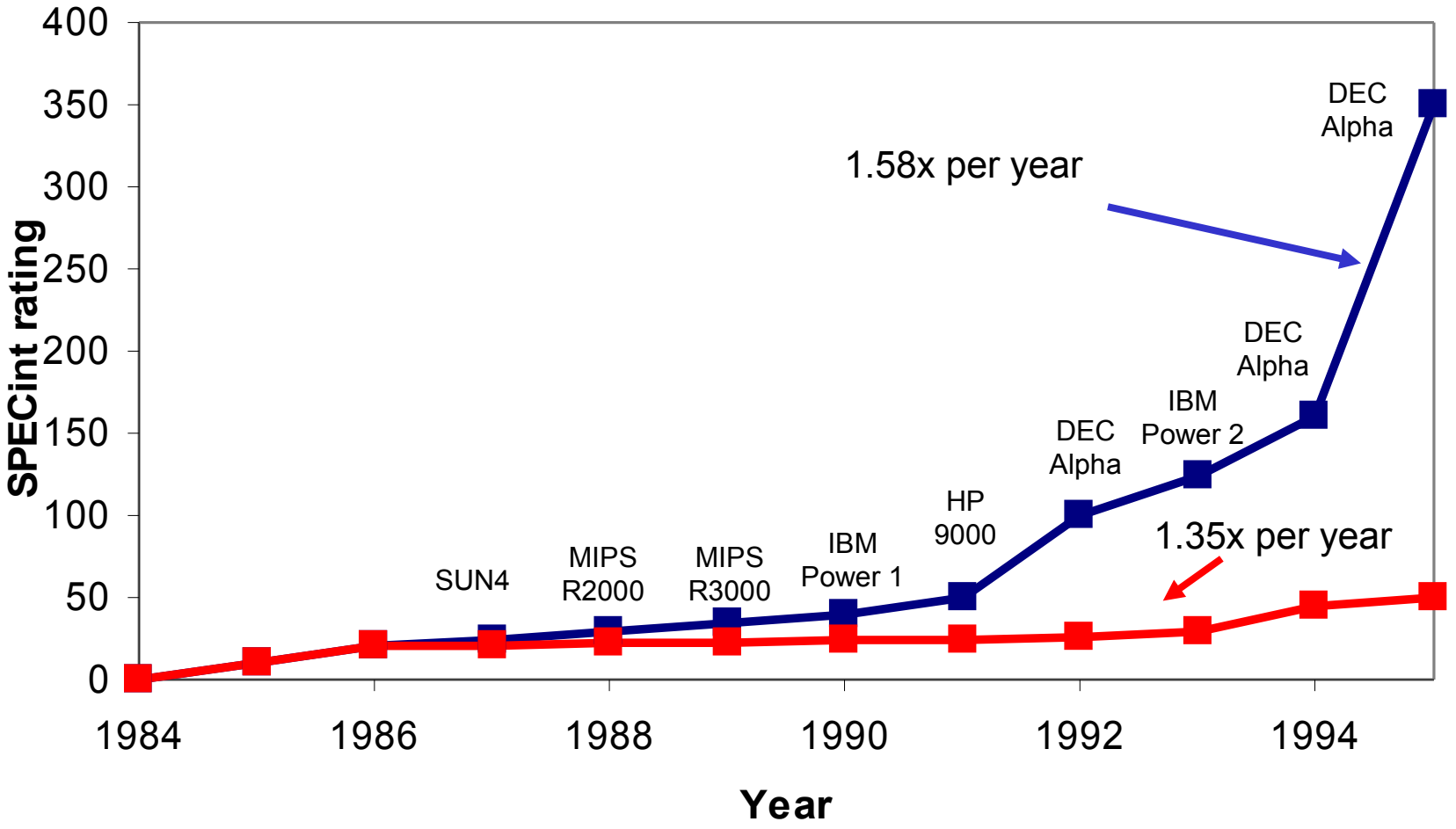
2X transistor/Chip

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: circa10% 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.

  $\Rightarrow$ *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*

  $\Rightarrow$ *conviene costruire processori molto veloci, necessariamente con repertori semplici, e contare sull'ottimizzazione del compilatore*

Motivazioni

- Misura/valutazione di un insieme di parametri quantitativi per:
  - Quantificare le caratteristiche di una macchina (velocità, ecc.)
  - Fare scelte intelligenti (es. miglioramento dell'hardware vs installazione nuovo software)
  - Orientarsi nell'acquisto del calcolatore più adatto per l'applicazione data

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)

Relazione tra le prestazioni di due macchine

La frase "*X* è più veloce di *Y*" è usata per indicare che il tempo di risposta o di esecuzione, per un dato lavoro è più basso in *X* che in *Y*

"X è n% più veloce di Y" -> $\frac{Tempo\ di\ Esecuzione\ di\ Y}{Tempo\ di\ esecuzione\ di\ X}$ = 1 + n/100

Il tempo di esecuzione è il reciproco della prestazione

$$1 + n/100 = \frac{\frac{1}{Prestazioni\ Y}}{\frac{1}{Prestazioni\ X}} = \frac{Prestazioni\ di\ X}{Prestazioni\ di\ Y}$$

$$n = \frac{(Prestazioni\ X - Prestazioni\ Y)}{Prestazioni\ Y}$$

Principi Quantitativi per la Progettazione

Rendere veloce il caso più comune

- Si deve favorire il caso più frequente a discapito del più raro
- Il caso più frequente è spesso il più semplice e può essere reso più veloce del caso infrequente

Legge di Amdahl

- *Il miglioramento di prestazione che può essere ottenuto usando alcune modalità di esecuzione più veloci è limitato dalla frazione di tempo nella quale tali modalità possono venire impiegate*

Speedup

$$\text{Speedup} = \frac{\text{Prestazione usando quando possibile il miglioramento}}{\text{Prestazione non usando il miglioramento}}$$

Ovvero

$$\text{Speedup} = \frac{\text{Tempo di esecuzione non usando il miglioramento}}{\text{Tempo di esecuzione usando quando possibile miglioramento}}$$

Legge di Amdahl

Frazione $_{migliorato}$ (≤ 1)

- Frazione del tempo di calcolo della macchina in cui è utilizzata la miglioria

Speedup $_{migliorato}$ (≥ 1)

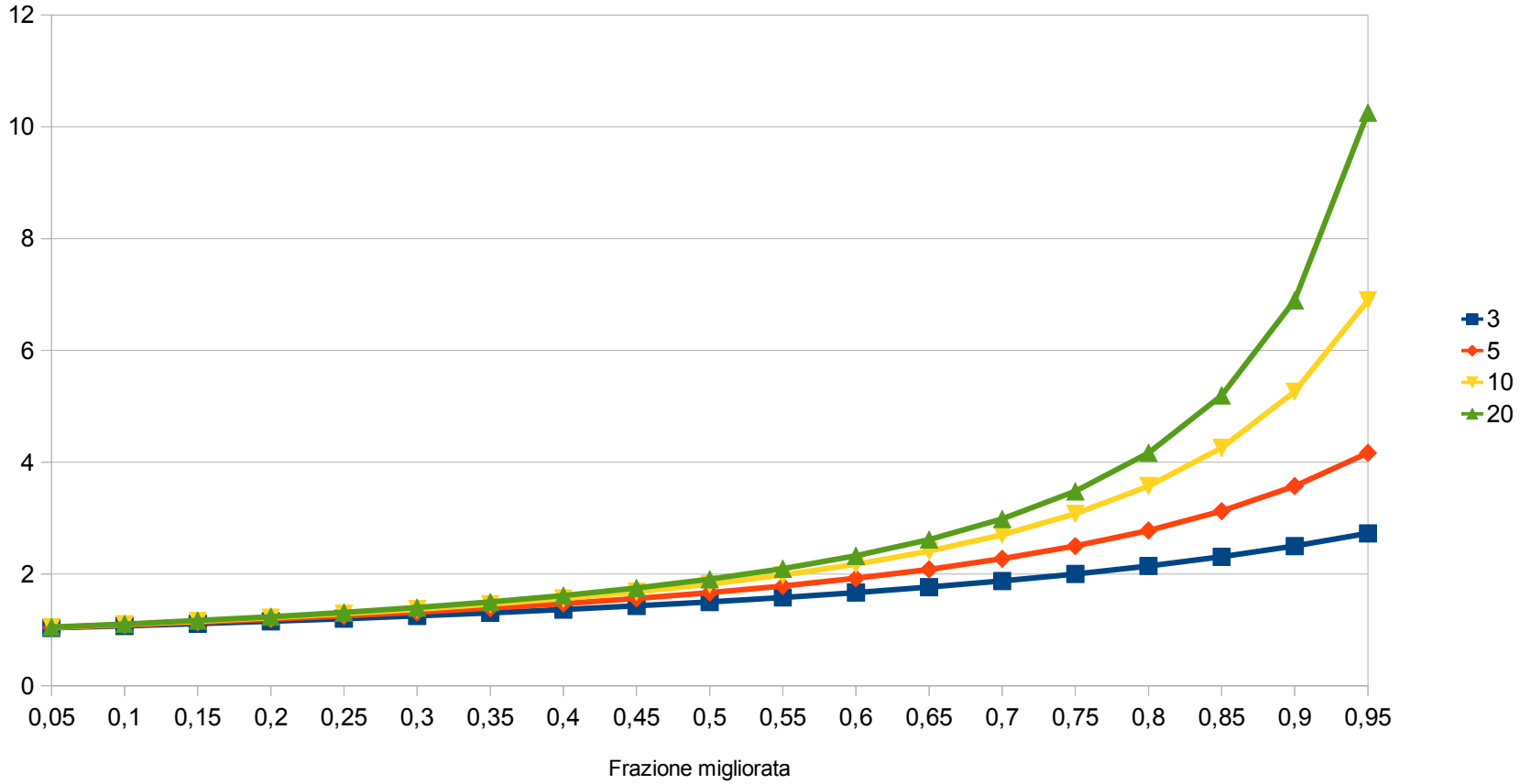
- Miglioramento ottenuto dal modo di esecuzione più veloce

$$\text{Tempo di esecuzione}_{nuovo} = (1 - \text{Frazione}_{migliorato}) \times \text{Tempo di esecuzione}_{vecchio} + \text{Frazione}_{migliorato} \times \text{Tempo di esecuzione}_{vecchio} / \text{Speedup}_{migliorato}$$

$$\text{Speedup}_{globale} = \frac{\text{Tempo di esecuzione}_{nuovo}}{\text{Tempo di esecuzione}_{vecchio}}$$

$$\text{Speedup}_{globale} = \frac{1}{(1 - \text{Frazione}_{migliorato}) + \frac{\text{Frazione}_{migliorato}}{\text{Speedup}_{migliorato}}}$$

Speedup Globale



Esempio 1

Si consideri un calcolatore (CALC1) che possiede un'unita aritmetico–logica intera (INT_ALU) e un'unita aritmetico-logica in virgola mobile (FP_ALU).

CALC1 esegue un'applicazioni software che tipicamente prevede il 90% di istruzioni in aritmetica intera e il 10% in virgola mobile.

Si consideri un nuovo calcolatore (CALC2) che possiede una INT_ALU ottimizzata che gli permette di raggiungere un miglioramento pari a **2x** nell'esecuzione delle istruzioni intere.

Si valuti il miglioramento globale ottenuto utilizzando CALC2 per l'applicazione software data.

Soluzione

$$\textbf{Speedup} = \frac{1}{(1 - 0{,}9) + \frac{0{,}9}{2}} = 1.81$$

Corollario

Se un miglioramento è utilizzabile solo per una frazione del lavoro complessivo, allora non è possibile accelerare il lavoro più del reciproco di uno meno tale frazione

$$\text{Speedup}_{\text{globale}} = \frac{1}{(1 - \text{Frazione}_{\text{migliorato}}) + \frac{\text{Frazione}_{\text{migliorato}}}{\text{Speedup}_{\text{migliorato}}}}$$

$$\leq \frac{1}{(1 - \text{Frazione}_{\text{migliorato}})}$$

Esempio 2

Si consideri un miglioramento che consente un funzionamento **10** volte più veloce rispetto alla macchina originaria, ma che sia utilizzabile solo per il **40%** del tempo. Qual è il guadagno complessivo che si ottiene incorporando detto miglioramento?

$\text{Frazione}_{\text{migliorato}} = 0.4$, $\text{Speedup}_{\text{migliorato}} = 10$

$$\text{Speedup}_{\text{migliorato}} = \frac{1}{(1-0.4)+\frac{0.4}{10}} = \frac{1}{0.6+0.04} = 1.5625$$

Esempio 3

Supponiamo di potere aumentare la velocità della CPU della nostra macchina di un fattore 5 (senza influenzare le prestazioni di I/O) con un costo 5 volte superiore.

Assumiamo inoltre che la CPU sia utilizzata per il 50% del tempo ed il rimanente sia destinato ad attesa per operazioni di I/O.

Se la CPU è un terzo del costo totale del computer è un buon investimento da un punto di vista costo/prestazioni, aumentare di un fattore cinque la velocità della CPU?

$$\text{Speedup}_{\text{migliorato}} = \frac{1}{(1-0.5)+\frac{0.5}{5}} = 1.67$$

Incremento costo=1/3*5+2/3*1 = 2.33

L'incremento del costo è superiore a quello delle prestazioni.

Il rapporto costo prestazioni peggiora

Il Tempo di CPU

$$\text{Tempo di CPU} = \text{cicli di clock della CPU} \times \text{durata periodo di clock}$$

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

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

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

Il 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

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

Il 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

Il 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 architetturale 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.

Il Tempo di CPU: il CPI

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

- 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 parallello (macchine superscalari) permette di rendere CPI minore di 1.

Il Tempo di CPU

Se indichiamo con:

- N_i, il numero di volte in cui l'istruzione I_i viene eseguita in un programma,
- CPI_i, il numero di cicli di clock richiesto della I_i
- n, numero di istruzioni diverse eseguite

avremo:

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

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

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

Il Tempo di CPU

- Indicato con
  - $f_i = N_i / N_{ist}$

otterremo:

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

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

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_i
ALU	43%	1
Load	21%	4
Store	12%	4
Branch	12%	2
Jump	12%	2

Soluzione

$CPI = 1*0.43 + 4*0.21 + 4*0.12 + 2*0.12 + 2*0.12 = 2.23$

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

Misura delle prestazioni: esempio

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

- Processore1: $N_{IST1} = N$; $CPI_1=3$; $T_1=T$
- Processore2: $N_{IST2} = 2,5N$; $CPI_2=2$; $T_2=T/2$

- $T_{CPU1}=N_{IST1}*CPI_1*T_1=N*3*T$
- $T_{CPU2}=N_{IST2}*CPI_2*T_2=2,5N*2*T/2=2,5*N*T$

$$T_{CPU1} > T_{CPU2}$$

Misura delle prestazioni: esempio

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

- Processore1: $N_{IST1}=N$; $CPI_1=10$; $T_1=T/4$
- Processore2: $N_{IST1}=2N$; $CPI_2=1$; $T_2=T$

- $T_{CPU1}=N_{IST1}*CPI_1*T_1=N*10*T/4=2,5*N*T$
- $T_{CPU2}=N_{IST2}*CPI_2*T_2=2N*1*T=2*N*T$

$$T_{CPU1} > T_{CPU2}$$

MIPS(Milioni di istruzioni al secondo)

$$MIPS = \frac{\text{numero di istruzioni eseguite}}{\text{tempo di esecuzione} \times 10^6} = \frac{IC}{T_{CPU} \times 10^6}$$

$$\text{Ricordando che } T_{CPU} = \frac{IC \times CPI}{f_{CK}}$$

$$MIPS = \frac{f_{CK}}{CPI \times 10^6} = \frac{1}{CPI \times T \times 10^6}$$

MIPS(Milioni di istruzioni al secondo)

- Processore1: $N_{IST1}=2N$; $CPI_1=2$; $T_1=T$
- Processore2: $N_{IST1}=N$; $CPI_2=1,5$; $T_2=2T$

- $T_{CPU1}=N_{IST1}*CPI_1*T_1=2N*2*T=4*N*T$
- $T_{CPU2}=N_{IST2}*CPI_2*T_2=N*1,5*2T=3*N*T$

$T_{CPU1} > T_{CPU2}$ CPU_2 migliore di CPU_1

$MIPS_1=1/(CPI_1*T_1*10^6)=1/(2T*10^6)$

$MIPS_2=1/(CPI_2*T_2*10^6)=1/(3T*10^6)$

$MIPS_1 > MIPS_2$