



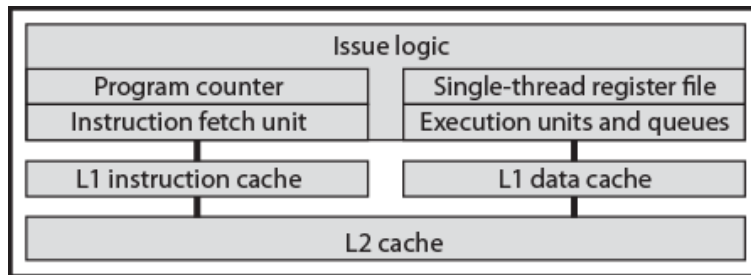
# Calcolatori Multicore



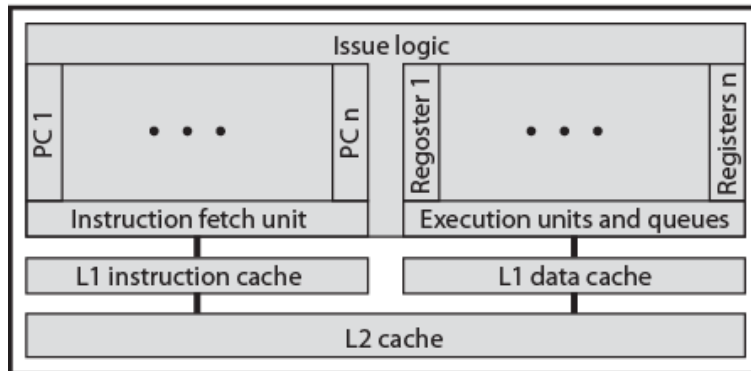
# Prestazioni hardware

- I microprocessori hanno visto una crescita esponenziale delle prestazioni
  - Miglioramento della organizzazione
  - Incremento della frequenza di clock
- Crescita del parallelismo
  - Pipeline
  - Pipeline parallele (superscalari)
  - superscalari + replicazione banco registri → Multithreading simultaneo (SMT)
- Problemi
  - Maggiore complessità richiede logica più complessa
  - Aumento dell'area del chip per supportare il parallelismo
    - Più difficile da progettare, realizzare e verificare (debug)

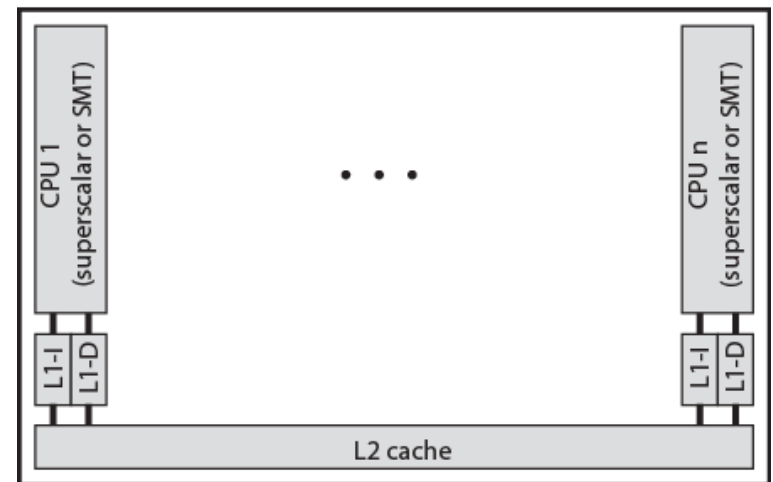
# Organizzazioni alternative del chip



(a) Superscalar



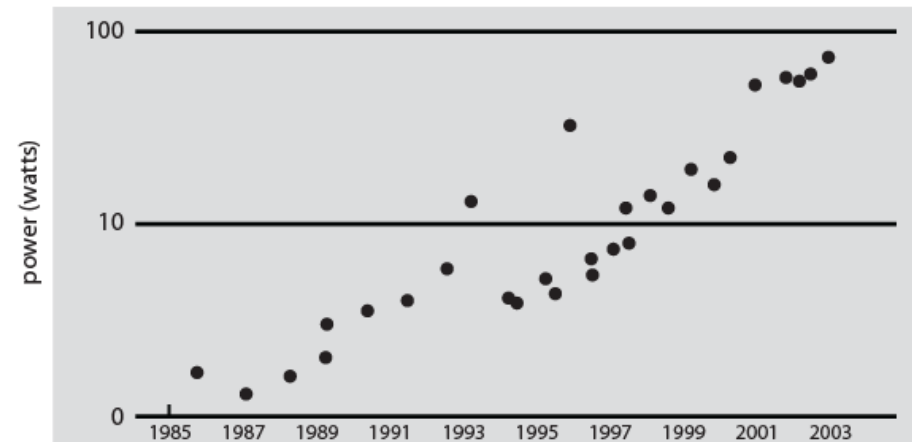
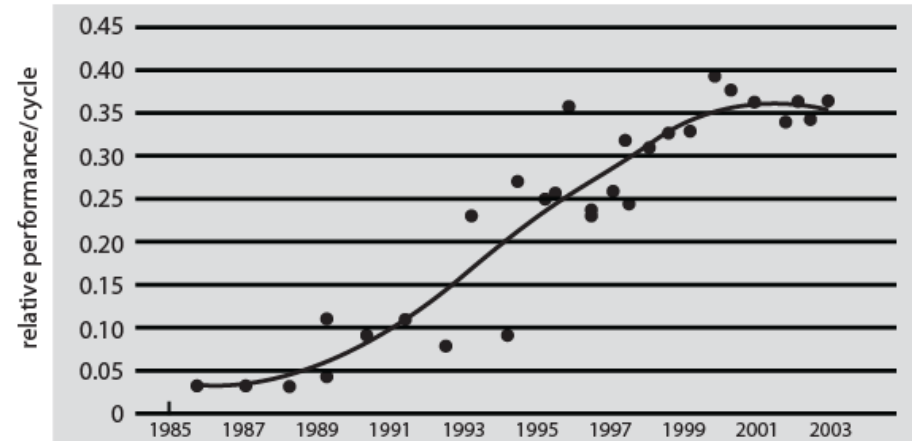
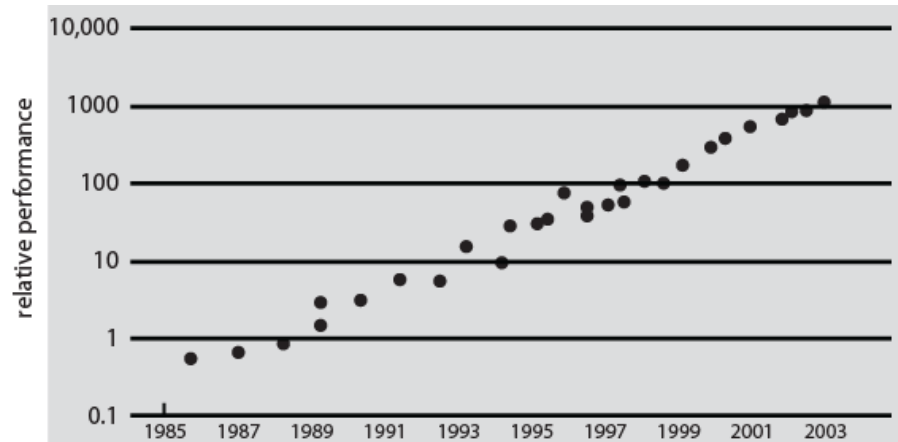
(b) Simultaneous multithreading



(c) Multicore

# Tendenze Hardware

## Intel

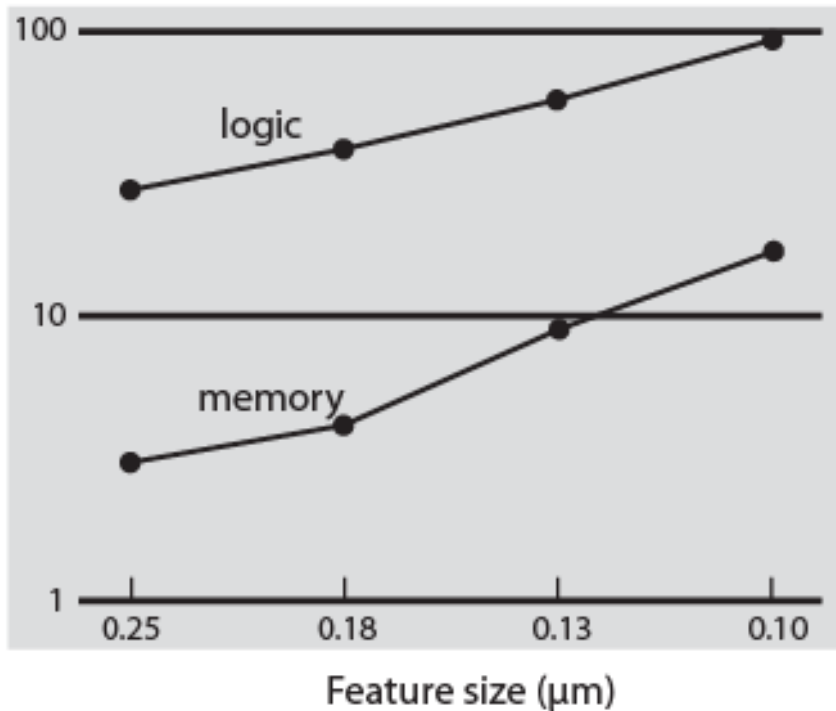


# Incremento in complessità

- La potenza cresce esponenzialmente con la densità del chip e la frequenza del clock
  - Rimedio: usare più spazio per la cache
    - Meno densa
    - Richiede molta meno potenza (ordini di magnitudine)
- Nel 2015
  - 100 miliardi di transistor in 300mm<sup>2</sup> sul “die” (chip)
    - Cache di 100MB
    - 1 miliardo di transistor per la logica
- Regola di Pollack:
  - Le prestazioni sono all'incirca proporzionali alla radice quadrata dell'incremento in complessità
    - Il raddoppio in complessità restituisce il 40% in più di prestazione
- Architetture multicore hanno il potenziale per ottenere un miglioramento quasi lineare
- Improbabile che un core possa utilizzare efficacemente tutta la memoria cache

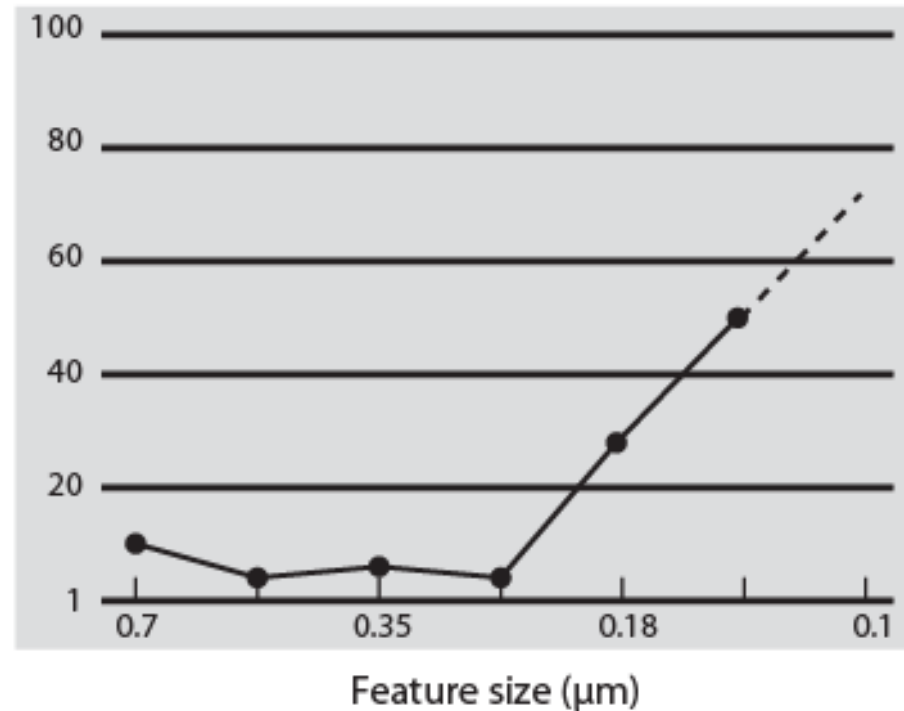
# Considerazioni sulla potenza e sulla memoria

Power density  
(watts/cm<sup>2</sup>)



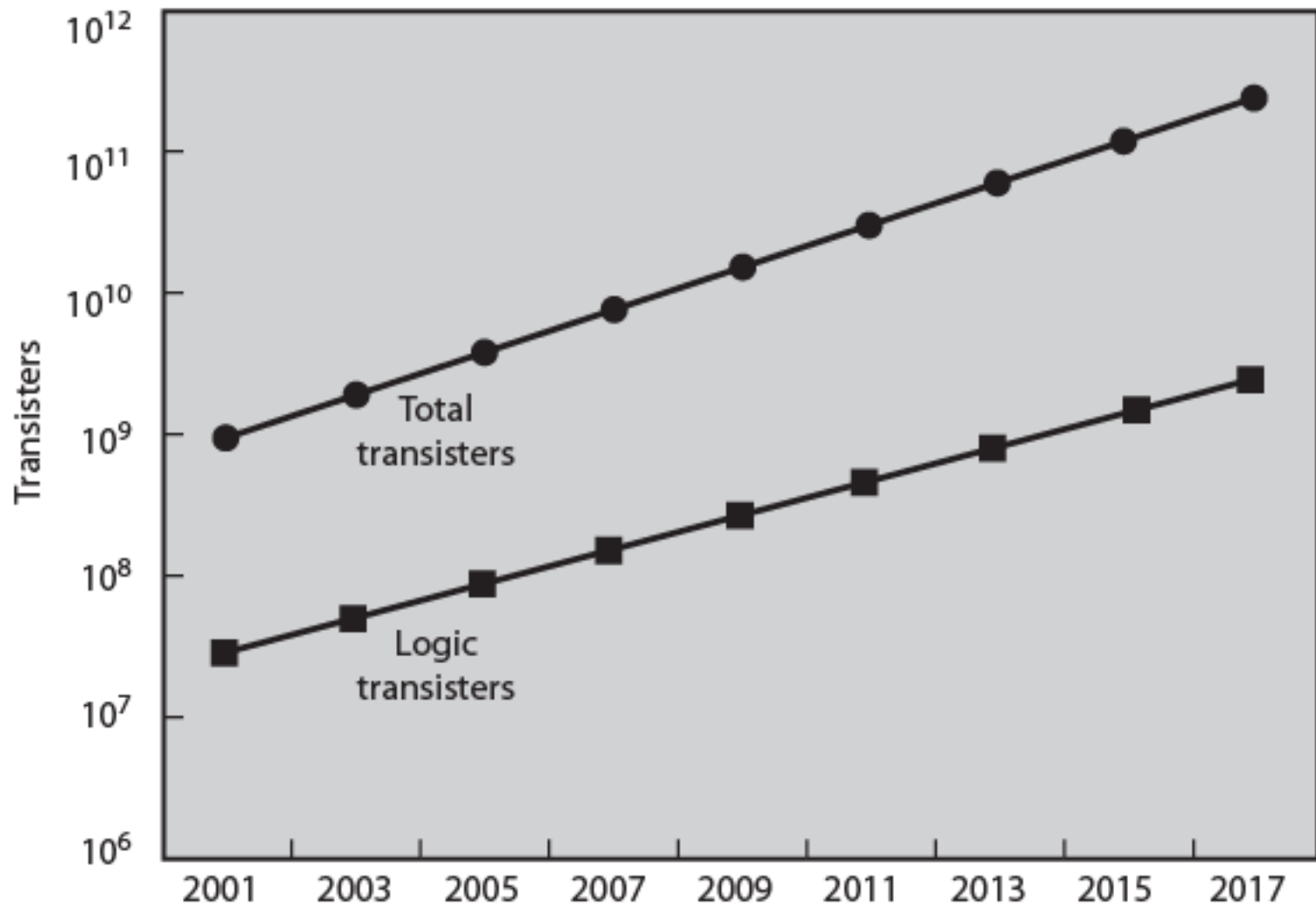
(a) Power density

cache percent  
of full chip area



(b) Chip area

# Utilizzo dei Transistor



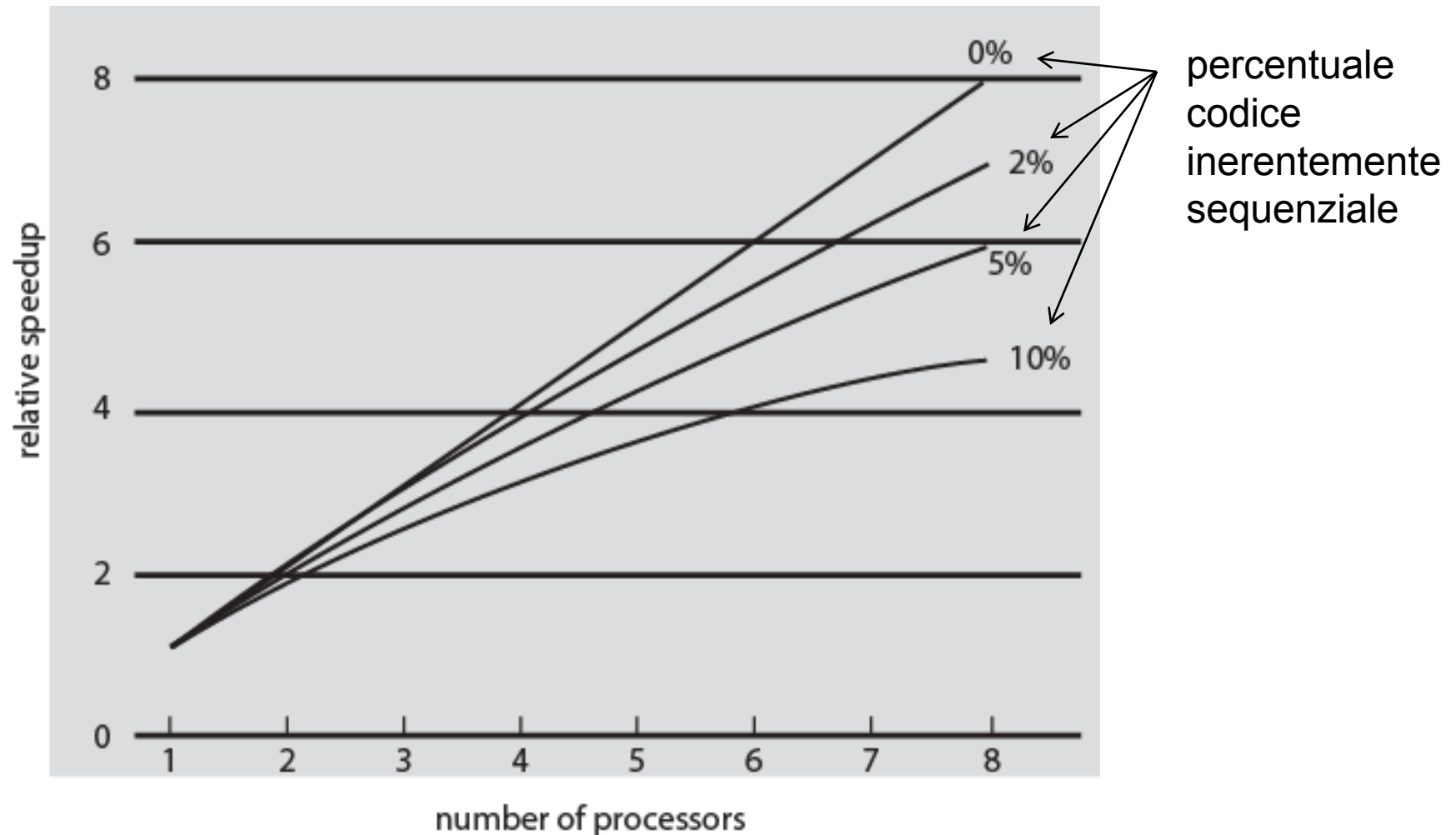


# Prestazioni del Software

- I vantaggi prestazionali dipendono dallo sfruttamento efficace delle risorse parallele
- Anche una piccola quantità di codice seriale ha un impatto significativo sulle prestazioni
  - Il 10% di codice intrinsecamente seriale eseguito su un sistema a 8 processori dà un incremento di prestazioni di solo 4,7 volte
- Overhead dovuto alla comunicazione, distribuzione del lavoro e mantenimento della coerenza della cache
- Alcune applicazioni effettivamente sfruttano i processori multicore



# Prestazioni del Software

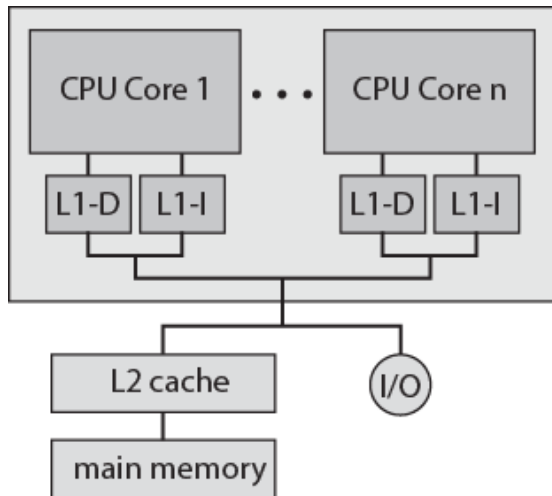




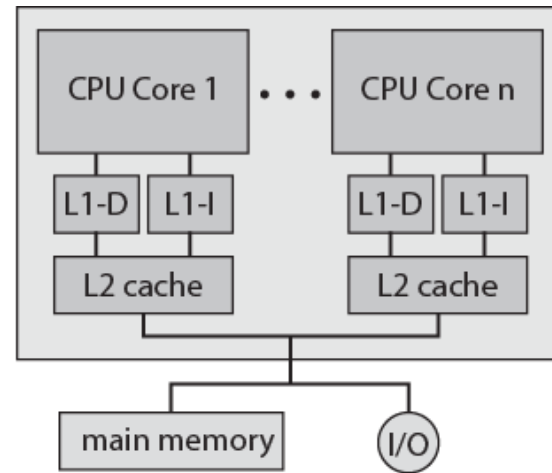
# Organizzazione Multicore

- Numero di core per chip
- Numero di livelli di cache per chip
- Quantità di cache condivisa

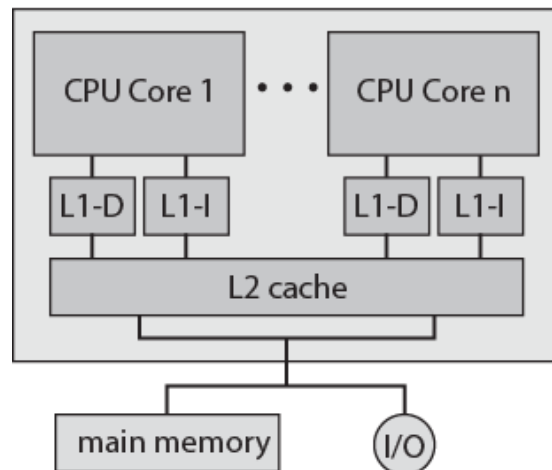
# Possibili alternative



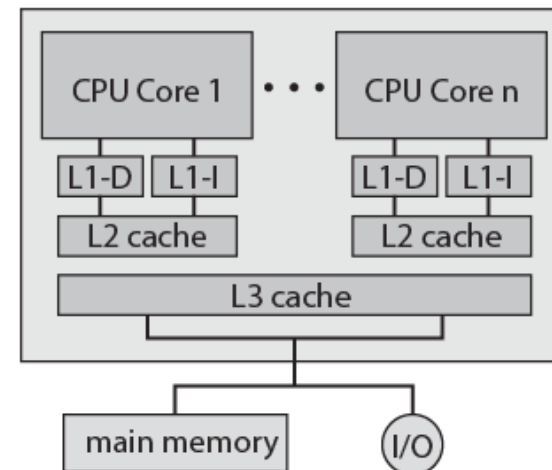
(a) Dedicated L1 cache



(b) Dedicated L2 cache



(c) Shared L2 cache



(d) Shared L3 cache

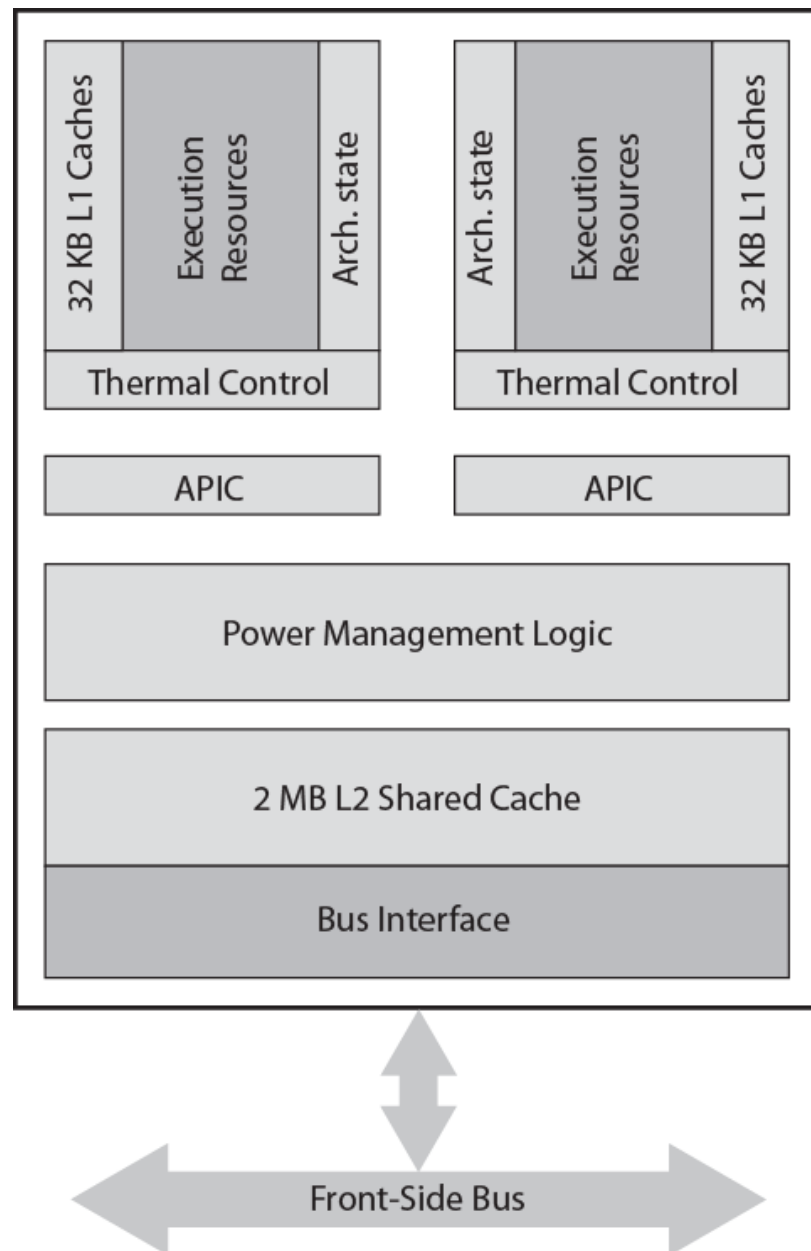


# Vantaggi di una Cache L2 condivisa

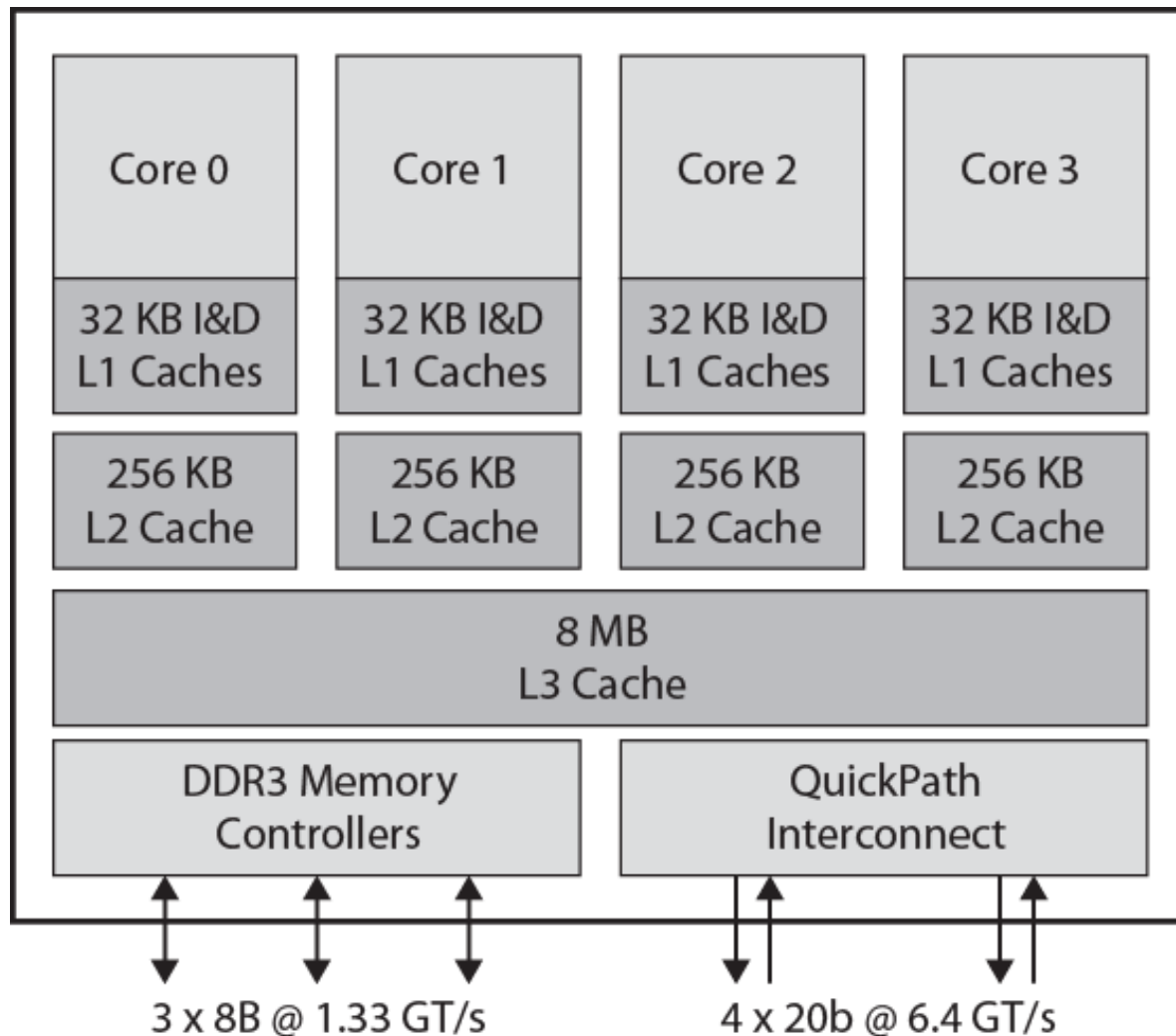
- Riduzione (accidentale) del numero di miss totali
- Dati condivisi da più core non sono replicati a livello di cache (a livello 2, ma possibile replicazione a livello 1)
- Con appropriati algoritmi di sostituzione dei blocchi, la quantità di cache dedicata ad ogni core è dinamica
  - Thread con minore località possono utilizzare più spazio di cache
- Comunicazione fra processi (anche in esecuzione su core diversi) facilitata dall'utilizzo della memoria condivisa
- Problema della coerenza della cache confinata al L1
- Cache L2 dedicate danno però un accesso alla memoria più rapido
  - Migliori prestazioni per thread con forte località
- Anche una cache L3 condivisa può migliorare le prestazioni

# Intel Core Duo

- 2006
- core superscalari
- cache L2 condivisa
- cache L1 dedicata
- unità di controllo termico
- controllori di interruzioni programmabili (APIC)
- logica di controllo della potenza
- Interfaccia bus



# Intel Core i7



- Novembre 2008
- core SMT
- cache L3 condivisa
- cache L1, L2 dedicate
- controllore memorie DDR3 sul chip
- Logica di connessione con controllo di coerenza della cache molto efficiente e veloce (banda totale di 25.6GB/s)

# Prestazioni dei processori multicore

