

Il parallelismo nelle architetture degli attuali PC

Salvatore Venticinque

Modena – 16/09/2010

Superscalarità

- Scheduling statico:
 - parallelismo definito a priori dal programmatore o dal compilatore (IA-64)
- Scheduling dinamico:
 - Parallelismo deciso automaticamente dalla CPU. L'hardware cerca di individuare le istruzioni pronte a essere eseguite (MPIS, Pentium 4)
 - È possibile l'esecuzione in ordine diverso da quello dato
 - È possibile eseguire una stessa istruzione su più dati (SIMD)

- ***Parallelismo a livello di istruzione esplicito nelle istruzioni macchina piuttosto che determinato dal processore durante l'esecuzione.***
 - Questo tipo di parallelismo è noto come EPIC (Explicit Parallel Instruction Computing). In questo caso è il compilatore (e non il processore) a dover capire quali istruzioni possono essere eseguite in parallelo e a generare il codice macchina relativo.
 - I processori EPIC non necessitano di circuiti complessi per l'esecuzione fuori sequenza. sono previsti fino a 256 registri a 64 bit (128 interi + 128 in virgola mobile nell'Itanium).
- **Più pipeline parallele**
 - (si prevedono implementazioni con 8 o più unità esecutive). Il pipelining può essere gestito a livello software.
- **Il caricamento speculativo e l'esecuzione speculativa (lungo entrambi i rami di un salto).**
 - ***Sono previsti appositi formati (modificatori delle istruzioni) per la gestione efficiente dell'esecuzione speculativa.***

Schedulazione dinamica P4

- Fetch: prelievo e invio alla coda istruzioni
- Decodifica: Conversione in microistruzioni
- Esecuzione fuori ordine
 - Smistamento verso l'unità funzionale richiesta (stazione di prenotazione)
 - Inserimento del riferimento all'istruzione nel buffer di riordino
 - Esecuzione vera e propria
 - Eventuali criticità causano attesa
- Commit unit
 - Riordino istruzioni in modo che terminino sequenzialmente
 - Può fornire in uscita più istruzioni per clock
 - Si tiene qui traccia di istruzioni pendenti

Multithreading

- Per implementare il multithreading, la CPU deve poter gestire lo stato della computazione di ogni singolo thread.
- Ci devono quindi essere almeno un Program Counter e un set di registri separato per ciascun thread.
- Inoltre, il thread switch deve essere molto più efficiente del process switch, che richiede di solito (essendo effettuato almeno in parte a livello software) centinaia o migliaia di cicli di clock
- Esistono 3 tecniche di base per il multithreading:
 - fine-grained multithreading (uno switch ad ogni clock)
 - coarse-grained multithreading (uno switch in caso di stall)
 - medium--grained multithreading /switch quando si prevede uno stall)

Multithreading

Ma come si fa a sapere a quale thread appartiene una qualsiasi istruzione nella pipeline?

- Nel caso del fine-grained MT, l'unico modo è di attaccare un thread identifier ad ogni istruzione, ad esempio l'ID univoco associato a quel thread all'interno dell'insieme di peer threads a cui appartiene.
- Nel coarse-grained MT si può adottare la stessa soluzione, oppure si può anche svuotare la pipeline ad ogni thread switch: in questo modo le istruzioni di un solo thread alla volta sono nella pipeline, e quindi si sa sempre a quale thread appartengono.
- Ciò ha senso solo se lo switch avviene ad intervalli molto maggiori del tempo necessario a svuotare la pipeline.

SMT

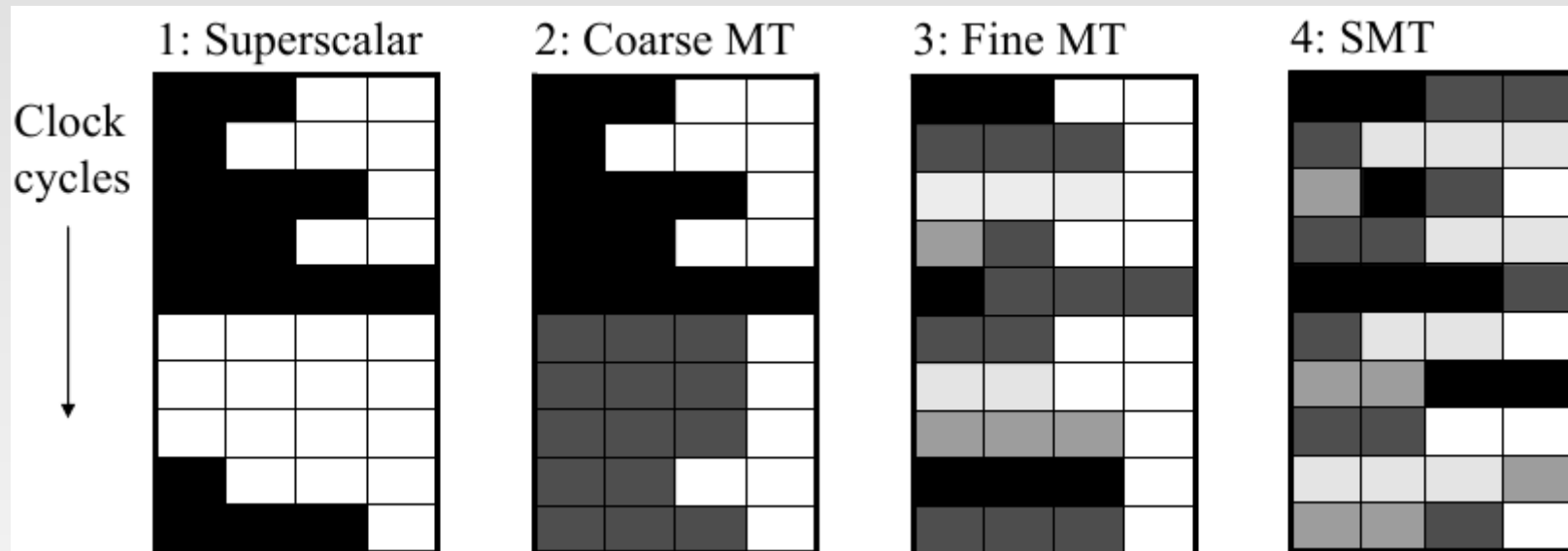
- Architetture superscalari, multiple issue e a scheduling dinamico della pipeline sfruttano contemporaneamente:
 - il parallelismo insito nelle istruzioni di un programma (ILP)
 - il thread level parallelism (TLP)

ILP + TLP = Simultaneous Multi-Threading (SMT)

La ragione per implementare l'SMT:

le moderne CPU multiple-issue hanno più unità funzionali di quante siano mediamente sfruttabili dal singolo thread in esecuzione.

Confronto



P4: Hyperthreading

Il multithreading è stato introdotto dalla Intel nello Xeon già nel 2002, e successivamente nel Pentium 4 nella versione a 3,06 Ghz con il nome di hyperthreading.

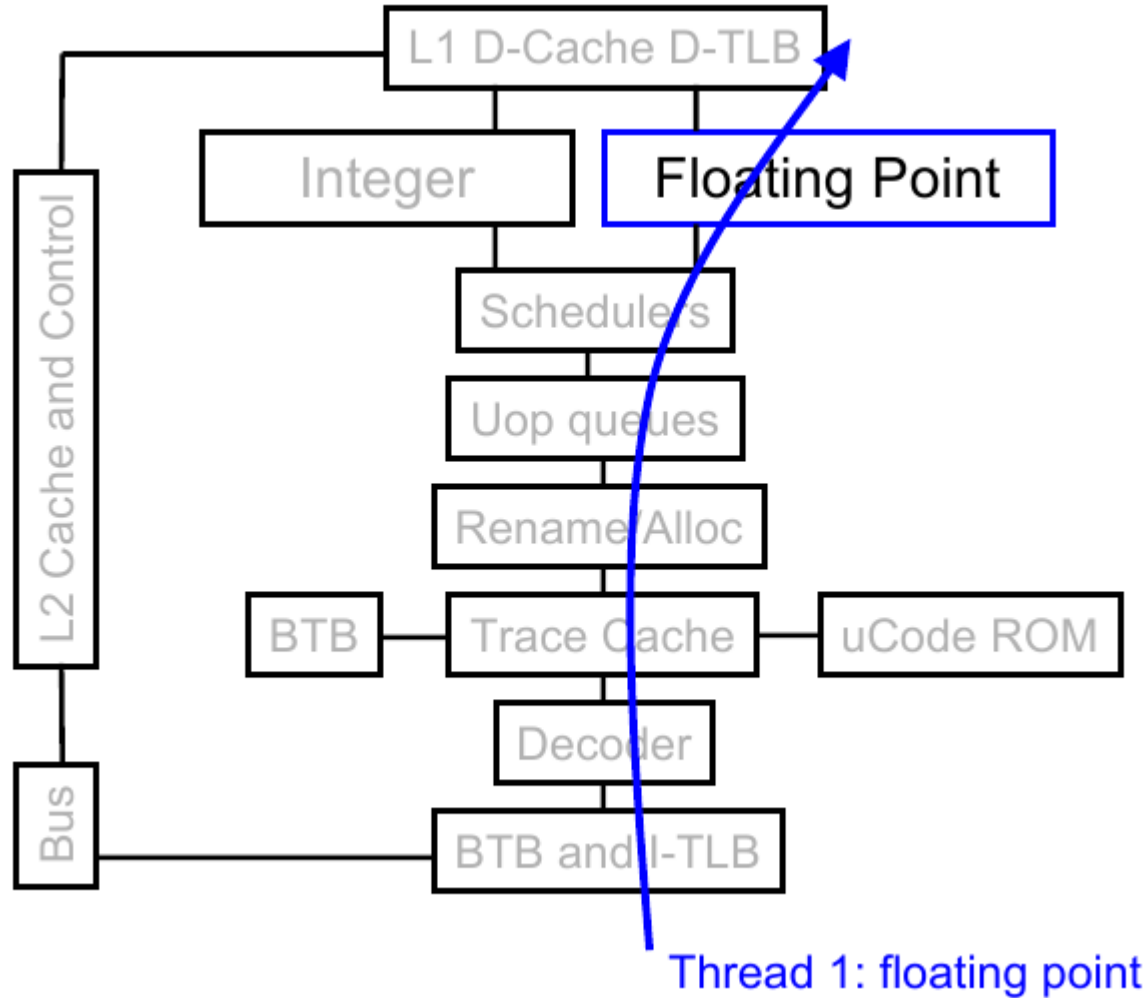
Il nome è altisonante, ma in realtà viene supportata l'esecuzione di due thread in modalità SMT.

Secondo quanto dichiarato dalla Intel, i progettisti hanno visto che per aumentare ulteriormente le prestazioni della CPU, il multithreading era la soluzione più semplice: per avere un secondo thread in esecuzione che sfruttasse le risorse della CPU che altrimenti rimanevano inutilizzate era sufficiente aumentare la dimensione dell'area della CPU del 5%.

Secondo i benchmark Intel, questo permette di aumentare le prestazioni della CPU di circa il 25% -- 30%.

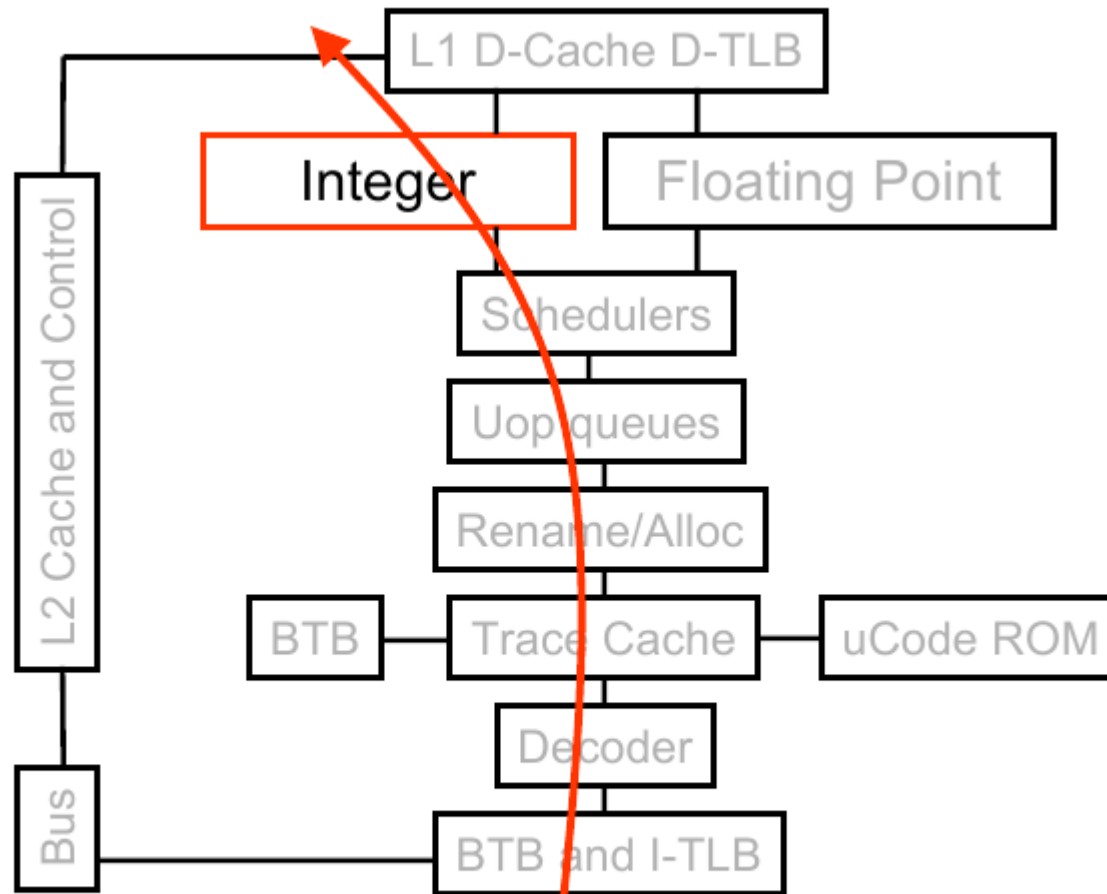
Implementazione senza SMT

T=0



Implementazione senza SMT

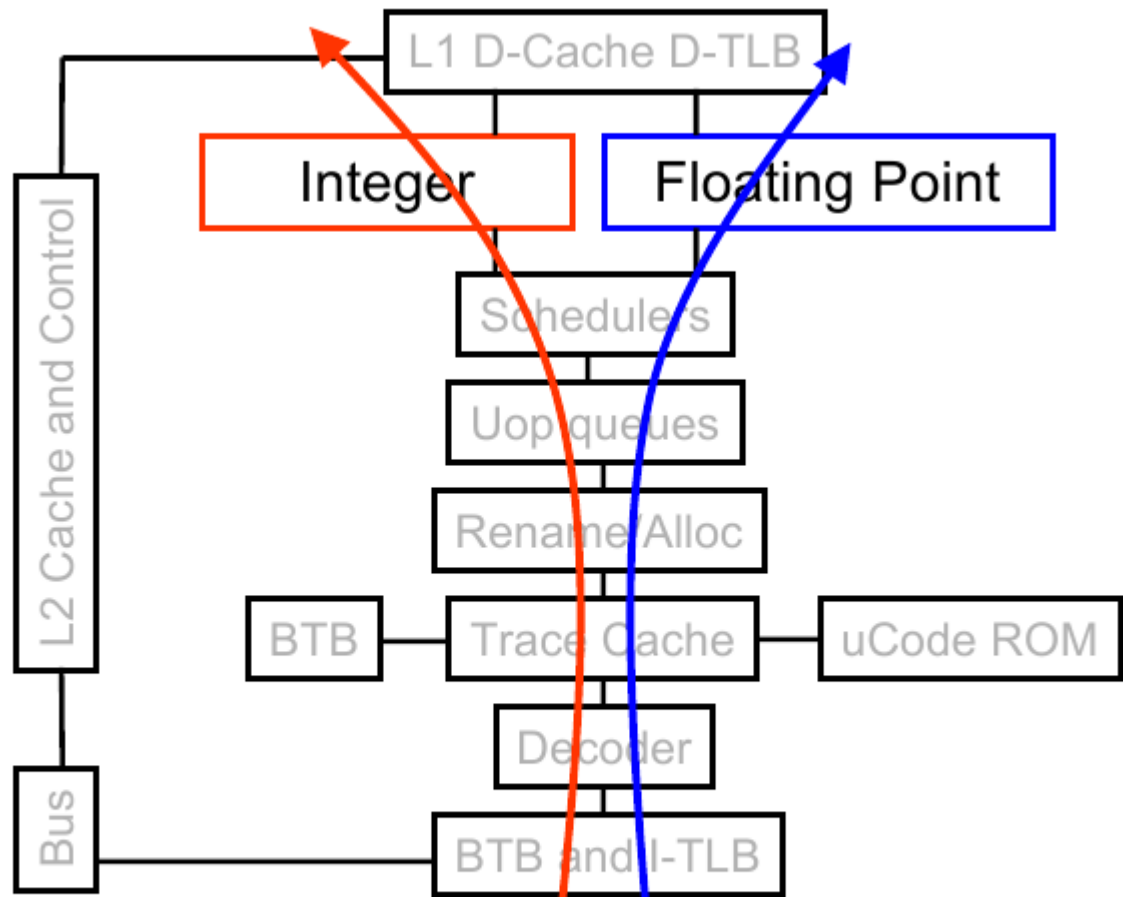
T=1



Thread 2:
integer operation

Implementazione con SMT

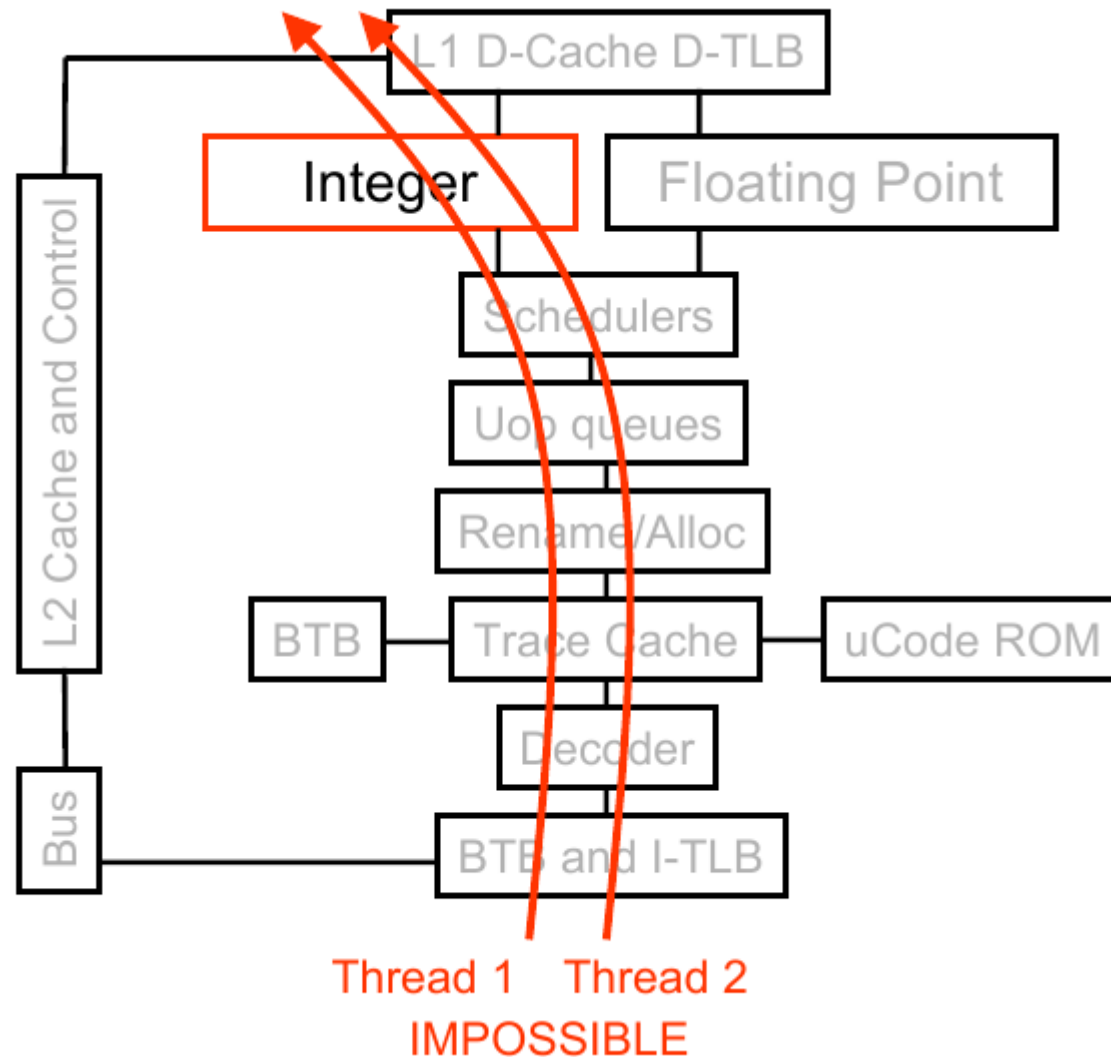
T=0



Thread 2:
integer operation

Thread 1: floating point

Implementazione SMT (con unico core)



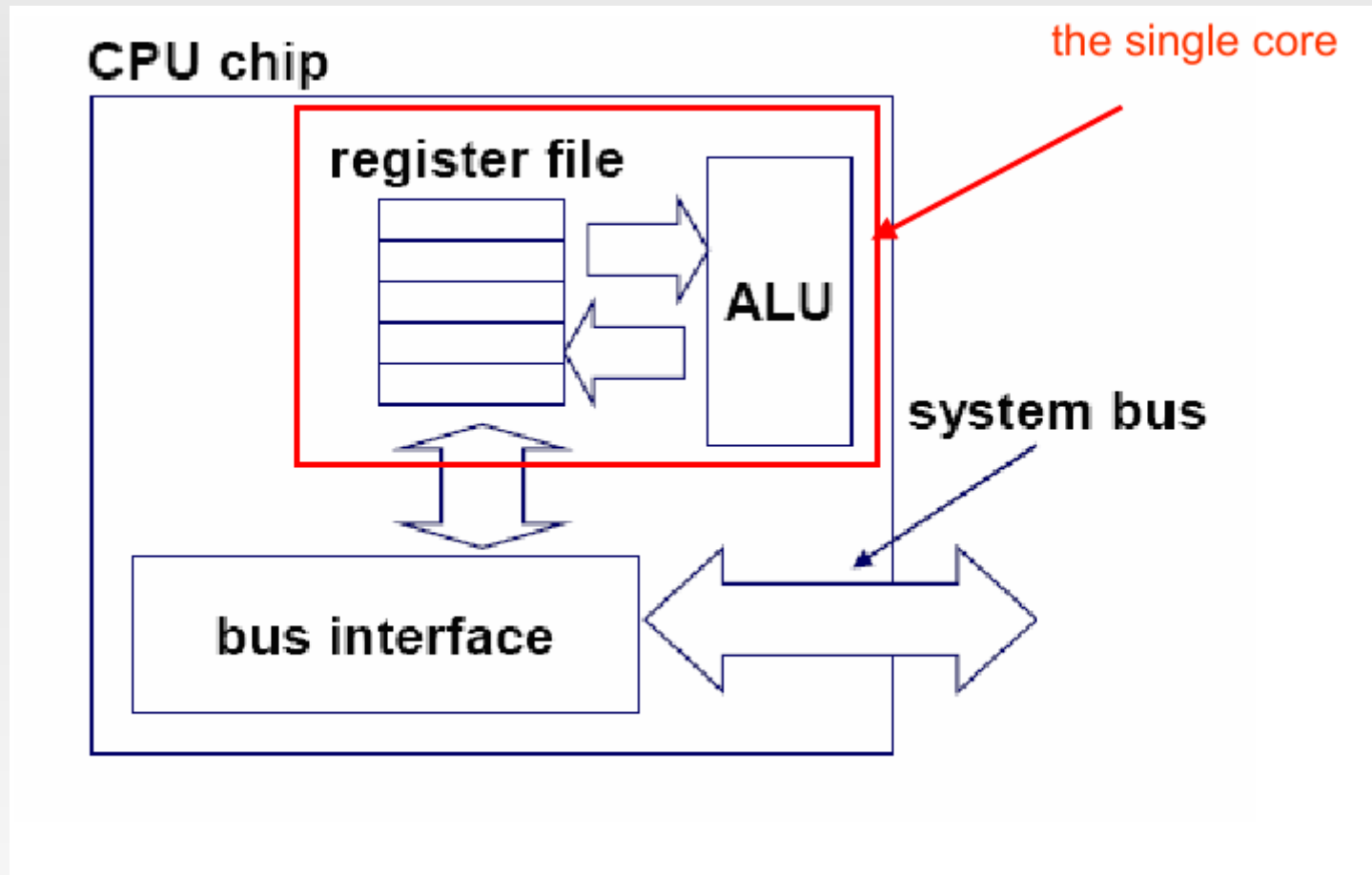
Realizzazioni Intel

	Freq. Core e Front Side Bus	Cache	Peculiarità architettura	Istruzioni speciali
Pentium (1993)	Core: 233 MHz FSB: 66 MHz	L1: 32 KB	2 Pipeline Superscalare	MMX
Pentium II (1997)	Core: 450 MHz FSB: 100 MHz	L1: 32 KB L2: 512 KB	2 Pipeline Superscalare	MMX
Pentium III (1999)	Core: 1,4 GHz FSB: 100 MHz	L1: 32 KB L2: 512 KB	2 Pipeline Superscalare	MMX, SSE
Pentium 4 (2000)	Core: 3.8 GHz FSB: 4×266 MHz (Quad Pumped)	L1: 32 KB L2: 512 KB L3: 2 MB	NetBurst: 2 Pipeline 20 stadi 5 unità esecuzione Migliore predizione Hyper Threading* NX-bit**	MMX, SSE, SSE2, SSE3, EM64T***

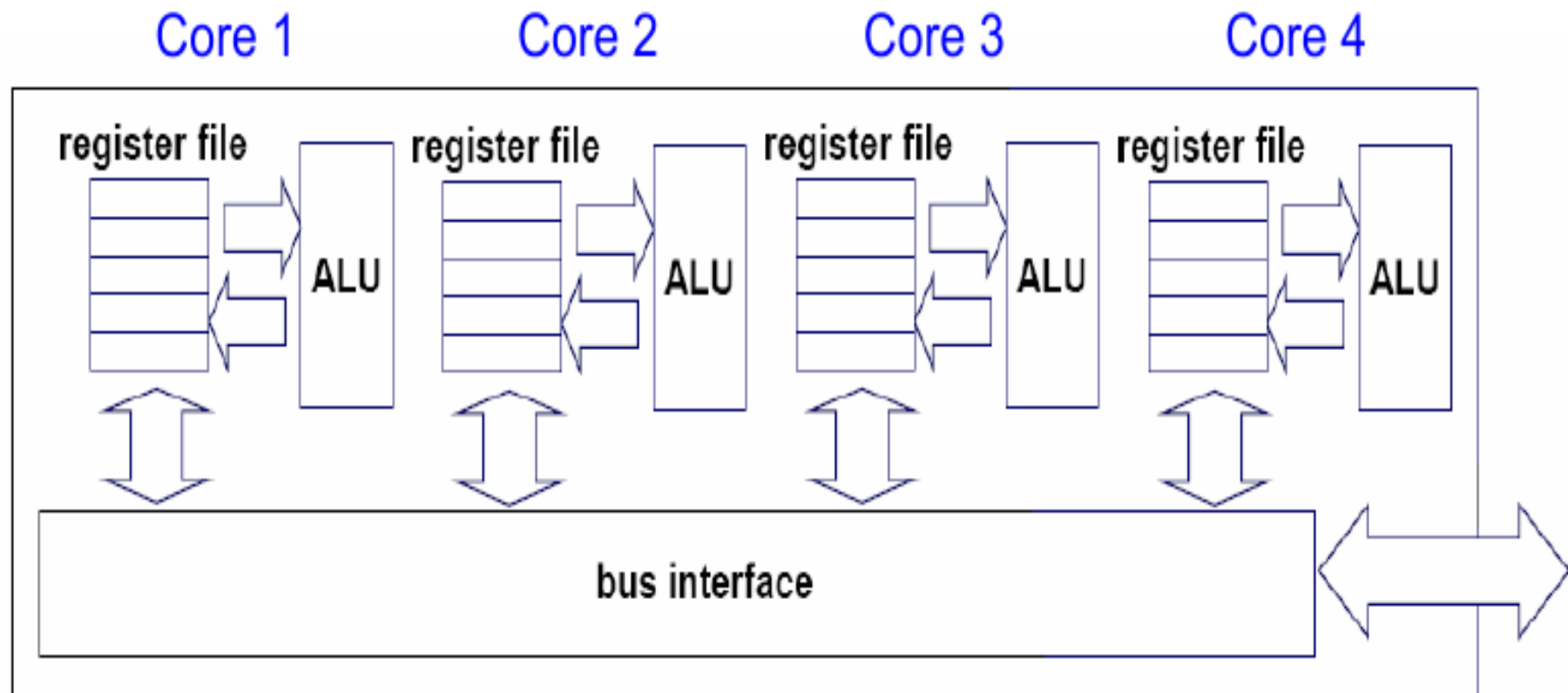
Multi-core

- Con l'avvento, a partire dalla metà del 2006, dei nuovi processori dual core basati sull'architettura "Core", Intel decise di abbandonare l'implementazione di tale tecnologia in quanto essa era poco efficace quando abbinata a processori dual core con pipeline corte come i Core 2 Duo.
- Agli inizi del 2008 al fine di non penalizzare troppo le applicazioni in grado di sfruttare il multi-threading, Intel ha deciso di riprendere la tecnologia Hyper-Threading che ha quindi ritrovato un proprio ruolo all'interno del mercato.
- Alla fine del 2008 sono arrivati i primi processori appartenenti alla nuova architettura Nehalem, successiva alla "Core", che porteranno con sé una tecnologia analoga al HT chiamata Simultaneous Multi-Threading.
- Al momento dell'annuncio da parte di Intel di re-inserire una tecnologia (sebbene si tratti di una sua evoluzione) che non ha mai goduto di particolare successo presso i clienti, sono emersi alcuni interrogativi sulle motivazioni di tale scelta. La risposta può essere trovata nello sviluppo del software; secondo Intel infatti, attraverso Windows Vista questa tecnologia potrà ottenere nuovi successi

Single core



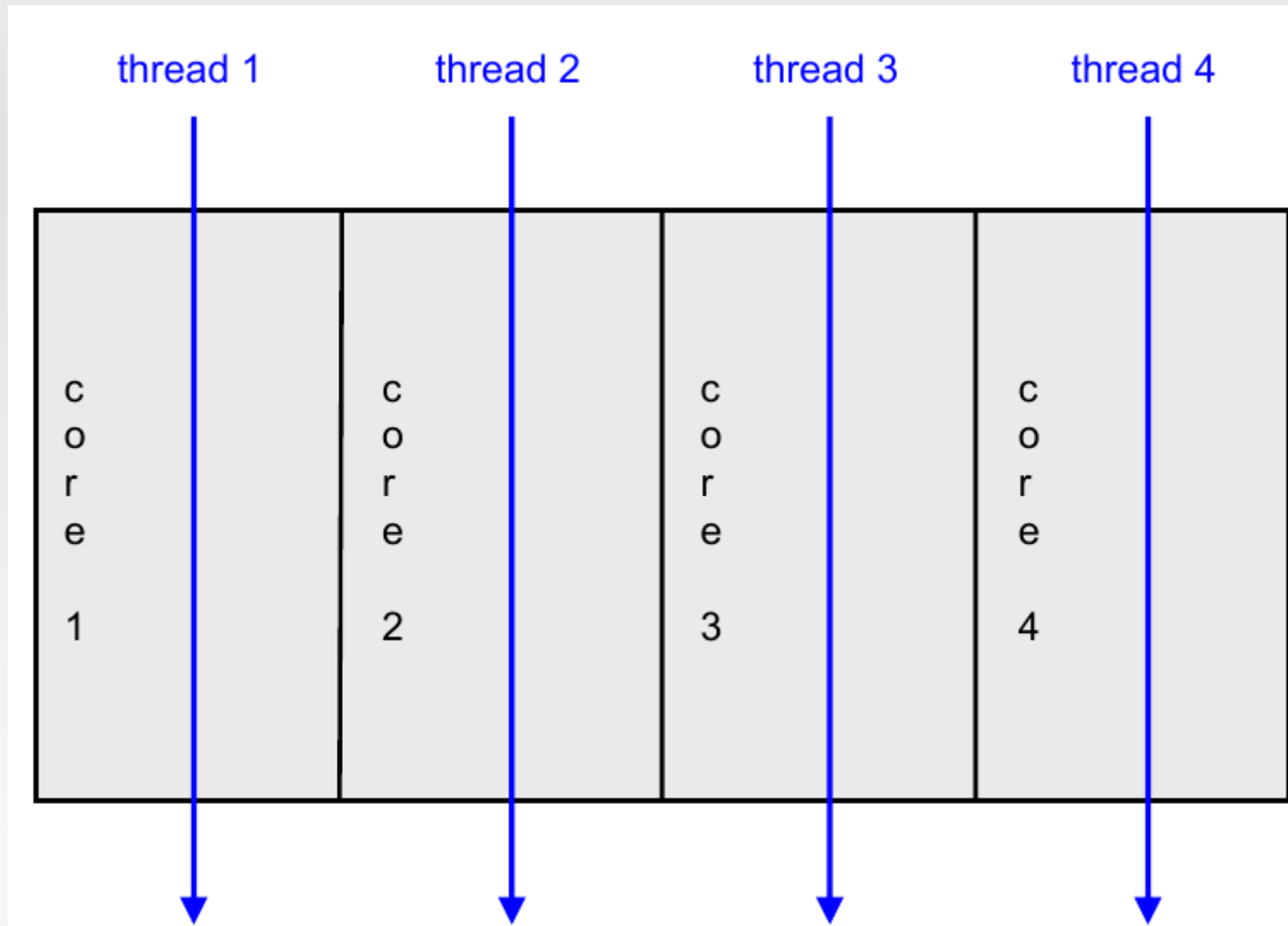
Sistemi multicore



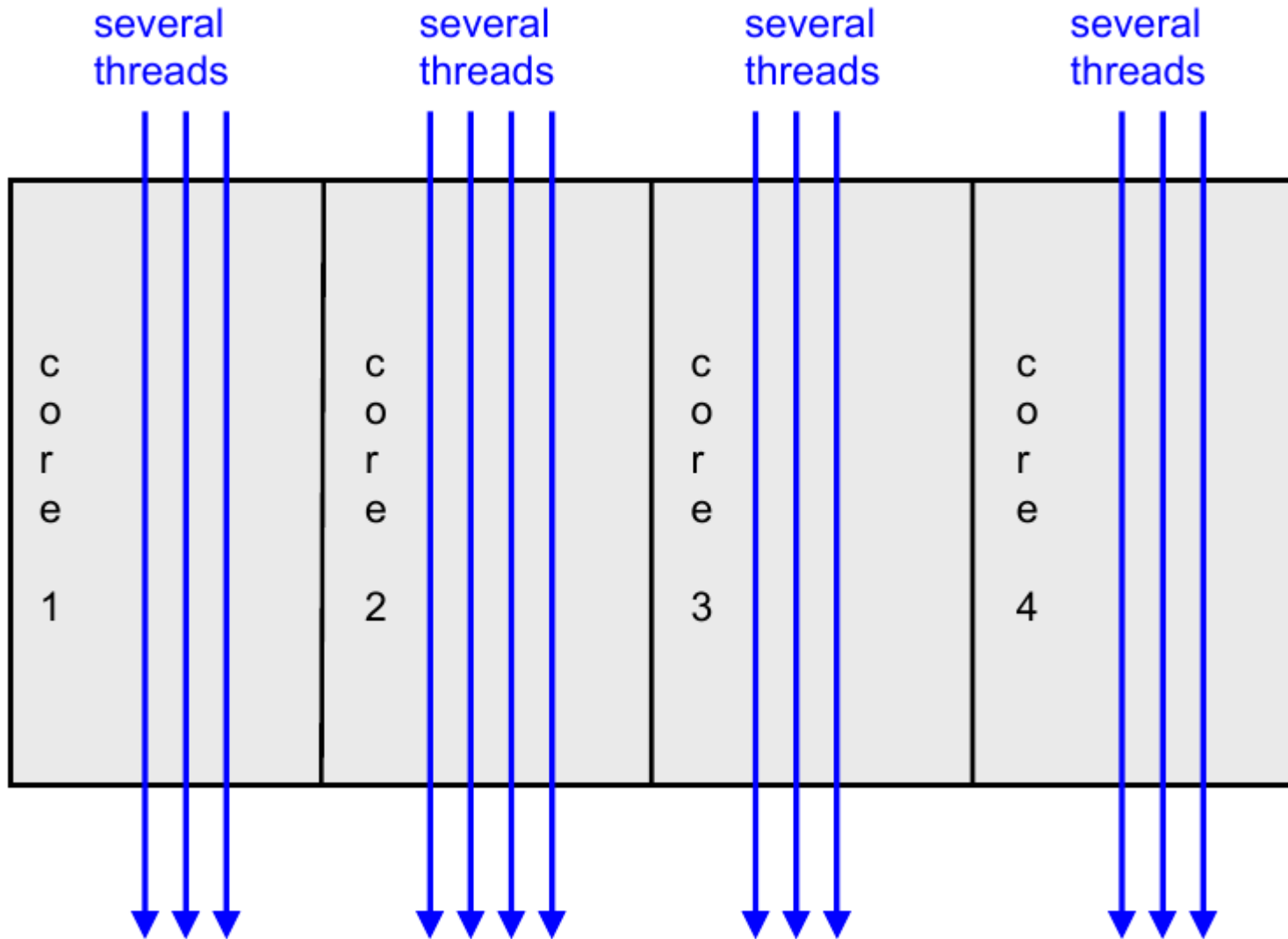
Multi-core

- Un processore Multi-core è un particolare tipo di multiprocessore:
 - Tutti i processori sono sullo stesso chip
 - I processori Multi-core sono MIMD
- Differenti cores eseguono differenti threads
 - (Multiple Instructions), operanti su parti diverse della memoria (Multiple Data).
- Il Multi-core è un processore shared memory
 - Tutti i core condividono la stessa memoria

I core eseguono in parallelo

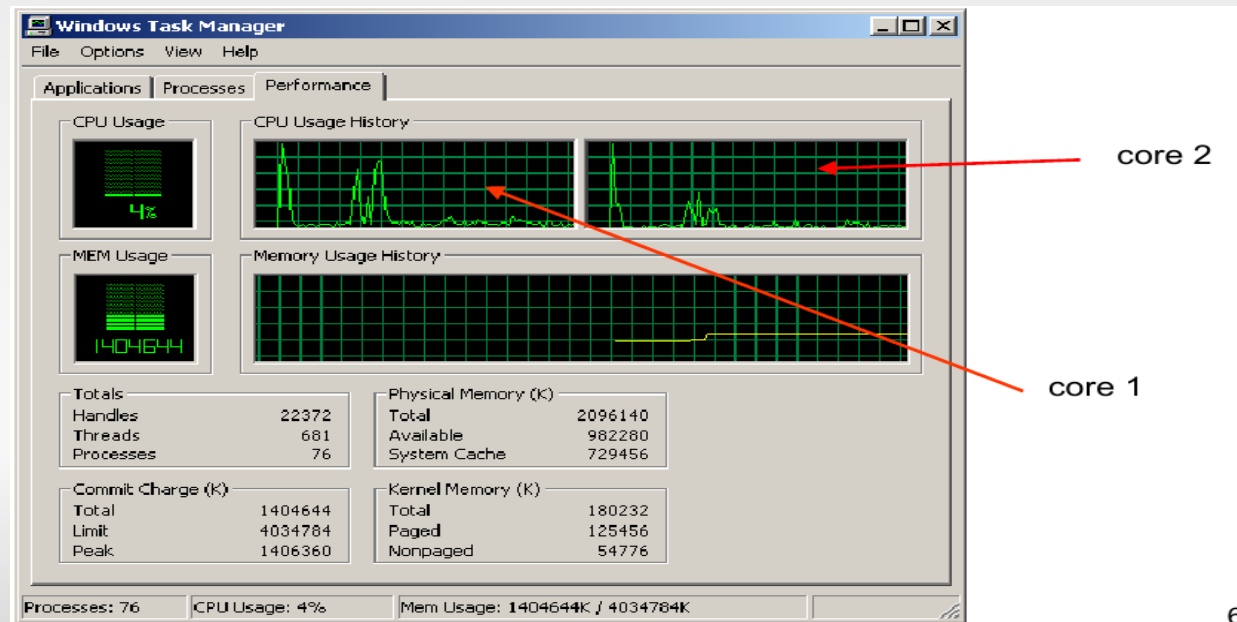


Multicore con supporto MT



Gesione cores dal SO

- Interazioni con il sistema operativo:
 - OS vede ogni core come un diverso processore
 - Lo scheduler di OS mappa i threads/processi sui diversi core
- Oggi la maggiorparte dei SO support i multicore



Perché i multi-core?

- Difficoltà nell'aumentare la frequenza del singolo core
- Alte frequenze causano:
 - Problemi di calore
 - Problemi legati alla velocità della luce
 - Difficoltà di progetto e verifica
 - Costi per i sistemi raffreddamento
- Molte applicazioni sono multithread

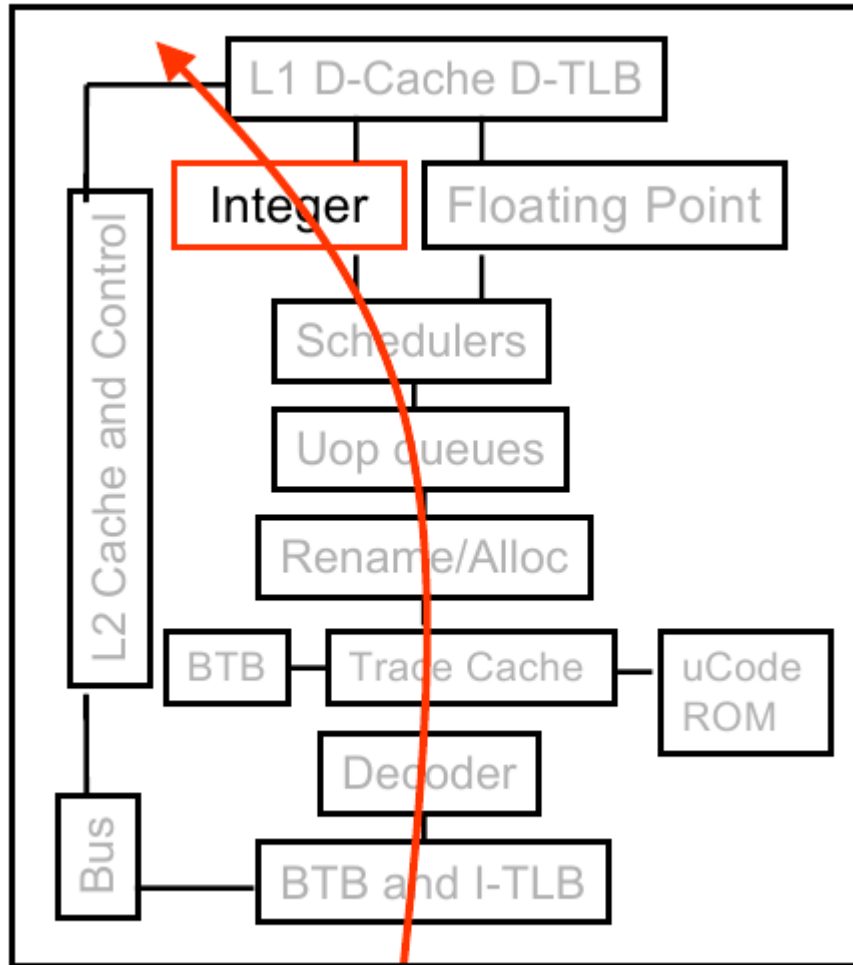
Rilevanza multi-core

- Thread-level parallelism (TLP) (granularità meno fine)
- I server possono gestire ogni richiesta in un thread diverso (Web server, database server)
- Un applicazione di gioco può eseguire AI, grafica, e fisica in 3 threads separati
- I processori single-core super-scalari non possono sfruttare al massimo la TLP

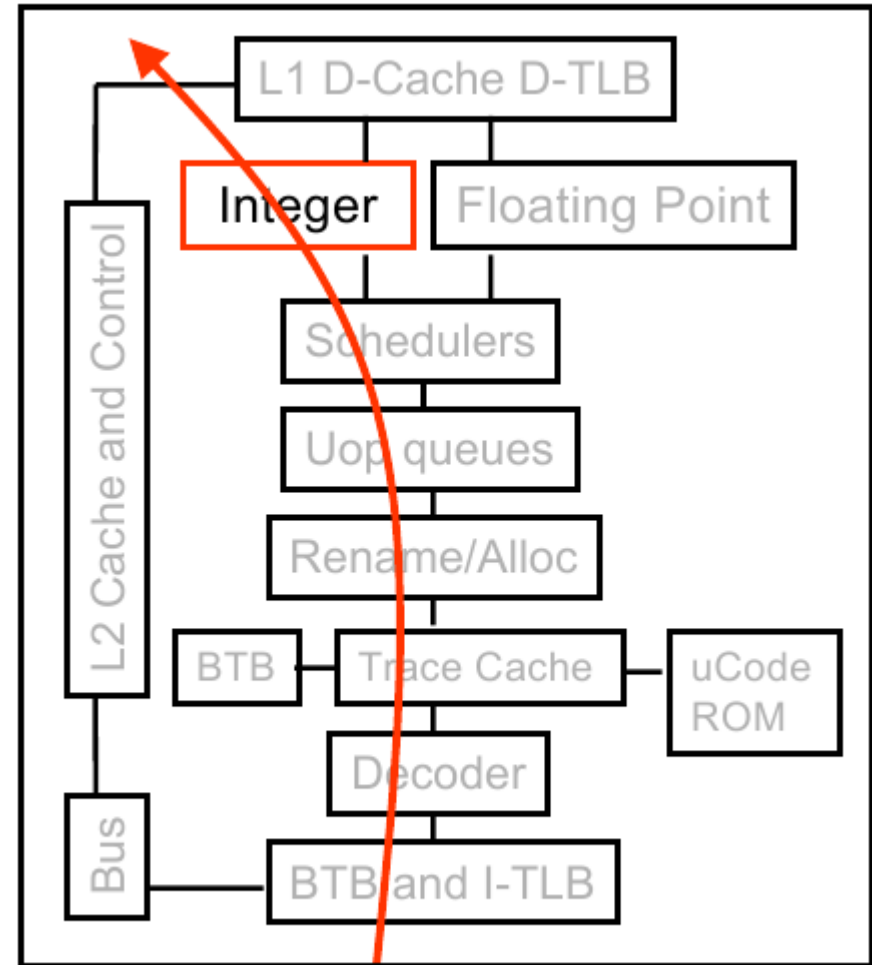
Applicazioni per multi-core

- Database server
- Web server
- Compilatori
- Applicazioni multimediali
- Applicazioni scientifiche (Es: CAD)
- In generale applicazioni TLP (in opposizione a ILP)

Multi-core senza SMT

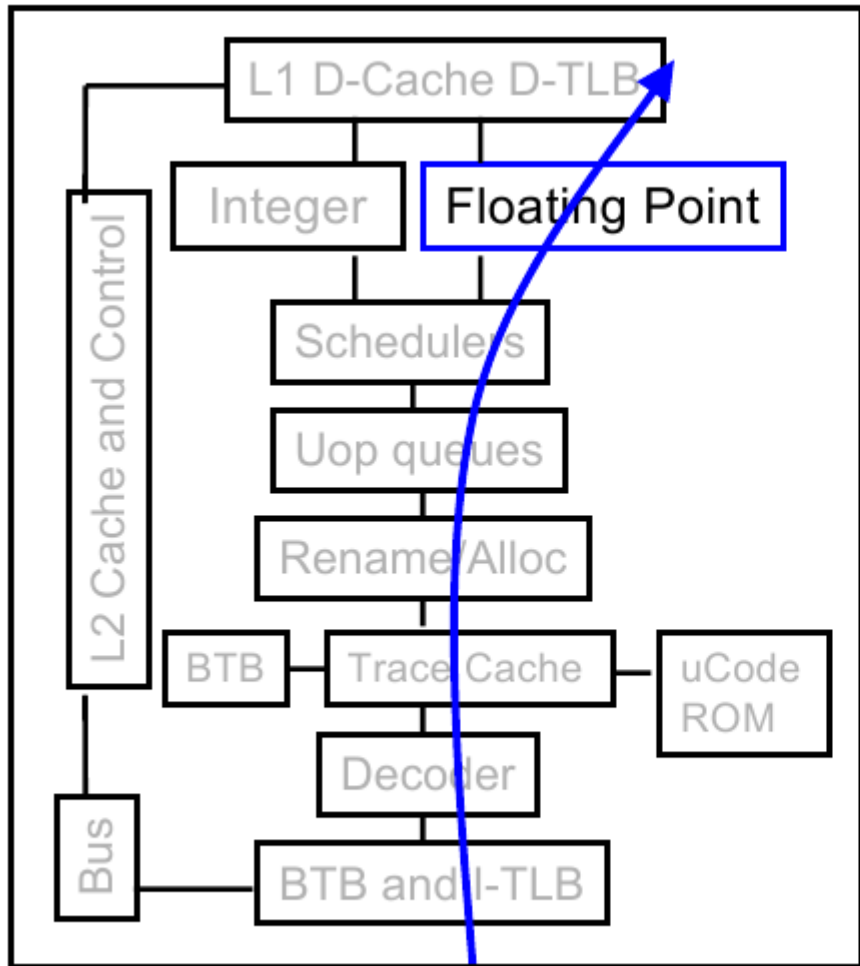


Thread 1

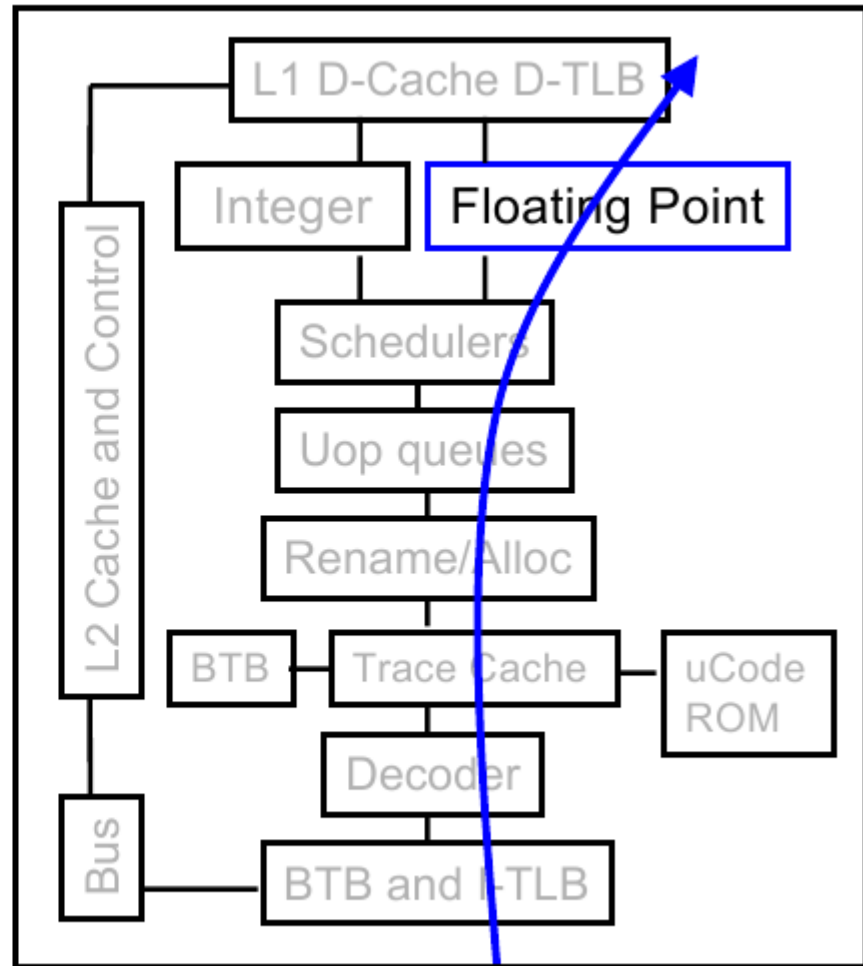


Thread 3

Multi-core senza SMT

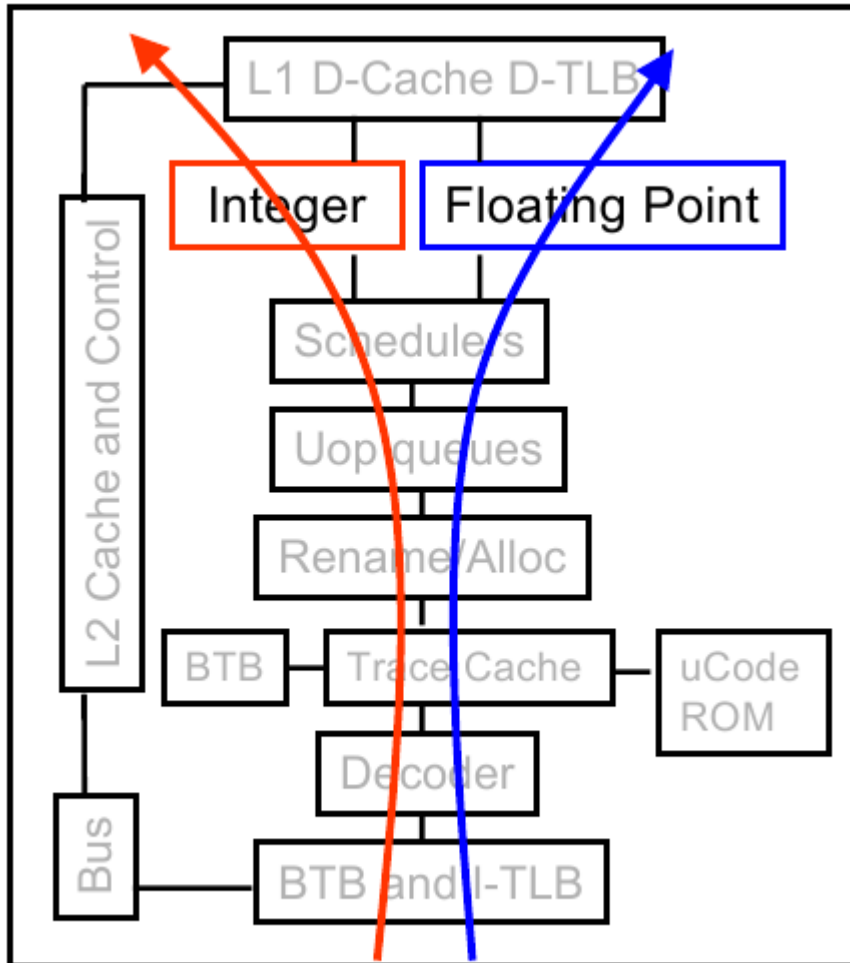


Thread 2

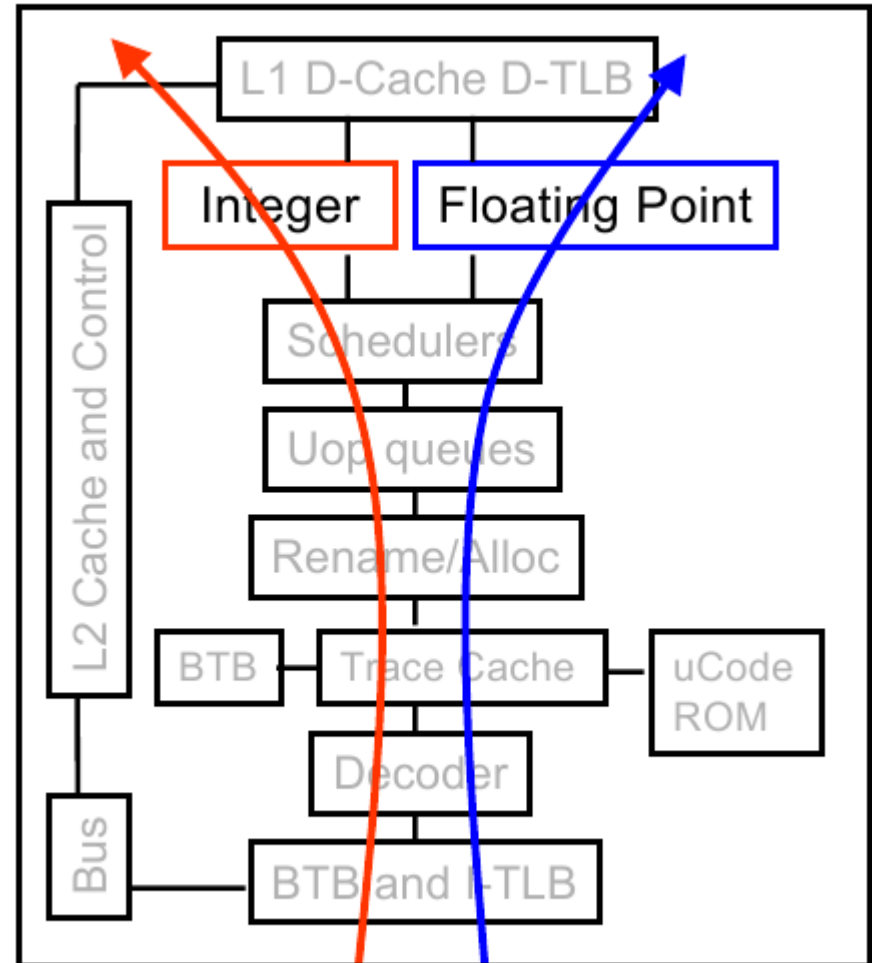


Thread 4

Multi-core con SMT




Thread 1 Thread 2



Thread 3 Thread 4

Intel core 2 duo

Essentials	
Status	Launched
Launch Date	Q1'09
Processor Number	T6400
# of Cores	2
# of Threads	2
Clock Speed	2 GHz
Bus/Core Ratio	10
FSD Speed	600 MHz
Instruction Set	64-bit
Embedded Options Available	 No
Supplemental SKU	No
Lithography	45 nm
Max TDP	35 W
VID Voltage Range	1.000V 1.250V
Package Specifications	
TJUNCTION	105°C
Package Size	35mm x 35mm
Sockets Supported	PGA478

Intel core2 duo

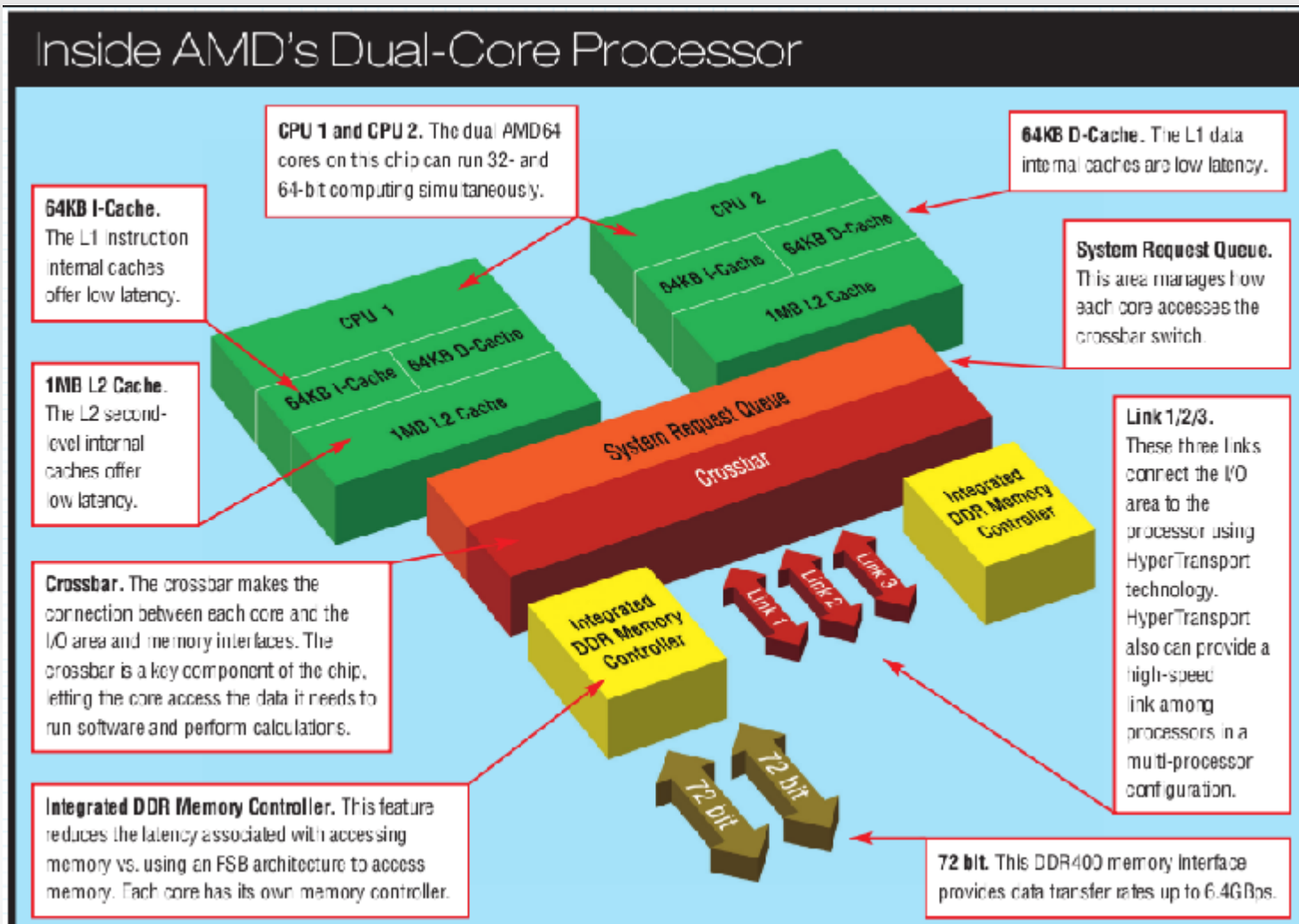
Advanced Technologies

Intel® Turbo Boost Technology		No
Intel® Hyper-Threading Technology		No
Intel® Virtualization Technology (VT-x)		No
Intel® Trusted Execution Technology		No
Intel® 64		Yes
Enhanced Intel SpeedStep® Technology		Yes
Intel® Demand Based Switching		No
Execute Disable Bit		Yes

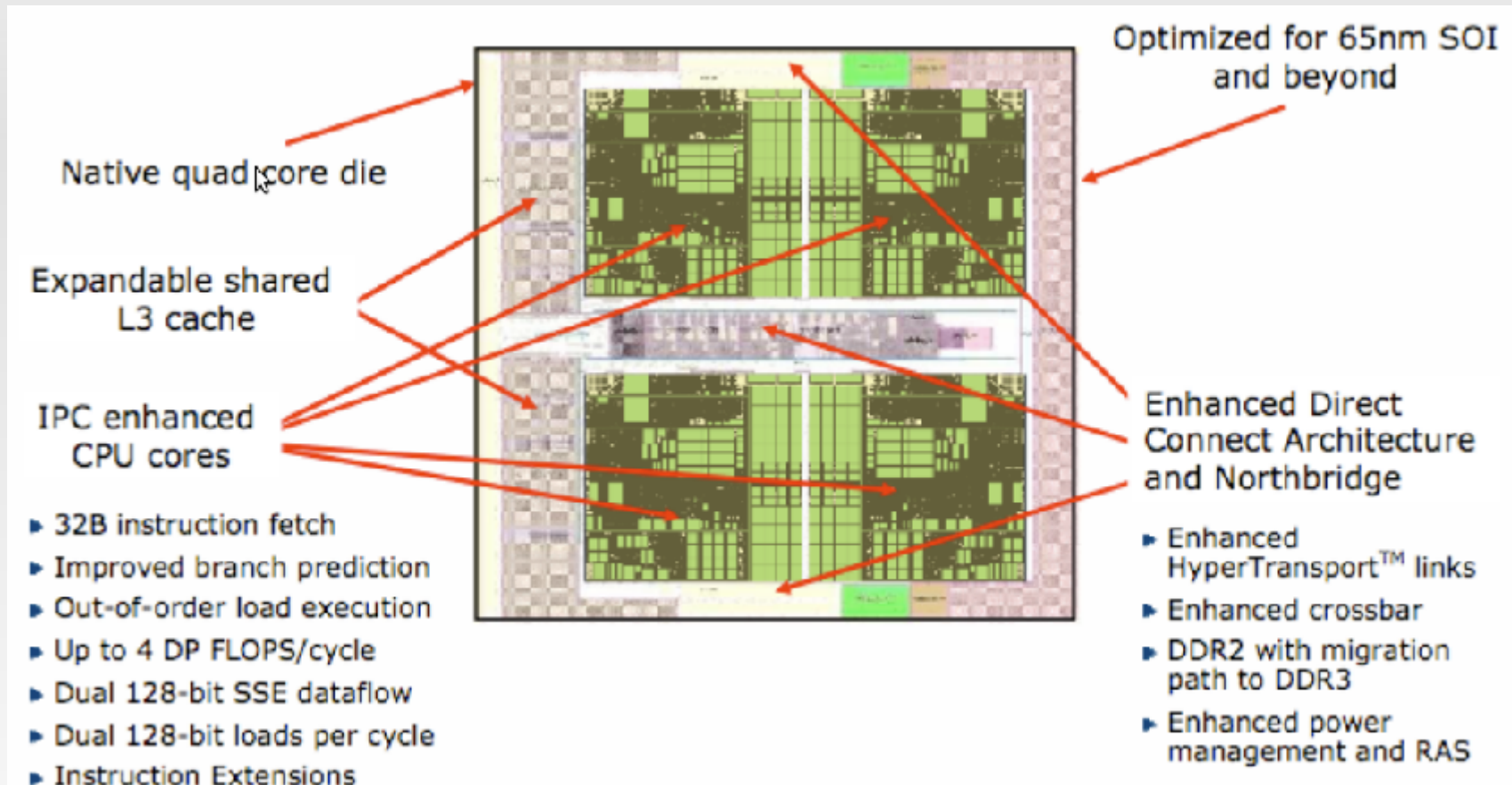
Attuali processori intel

Model	Core/threads	HyperThreading
P4	1C/2T	Y
Core 2 duo	2C/2T	N
i7-840	4C/8T	Y
i7-6xx	2C/8T	Y
i5-7xx	4C/4T	N
i5-6xx	2C/4T	Y


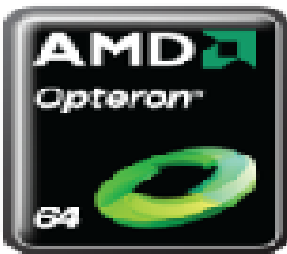
Architettura AMD



Quad core solution



Offerta AMD

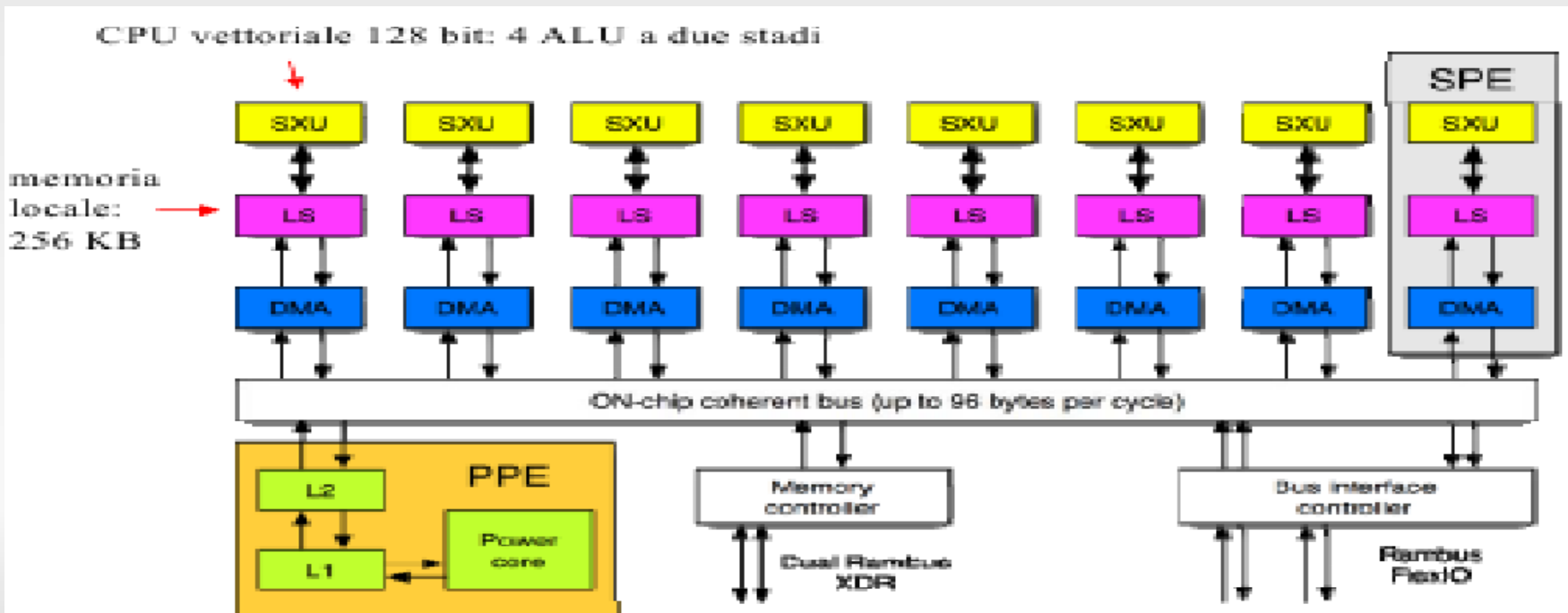
AMD Opteron processor family				
Logo	Server			
	Code-named	Core	Date released	Remarks
	Sledgehammer	130 nm	Jun 2003	Solo core
	Venus	90 nm	Aug 2005	
	Troy	90 nm	Jan 2006	
	Athens	90 nm	Jan 2006	
	Denmark	90 nm	Mar 2006	Dual-core
	Italy	90 nm	May 2006	
Egypt	90 nm	Jun 2006		
Santa Ana Santa Rosa	90 nm 90 nm	Aug 2006 Aug 2006		
	Barcelona	65 nm	Sep 2007	Quad-core
	Budapest	65 nm	Apr 2008	
	Shanghai	45 nm	Nov 2008	
	Istanbul	45 nm	Jun 2009	Six-core
	Magny-Cours	45 nm	Mar 2010	Eight-core
	Magny-Cours	45 nm	Mar 2010	Twelve-core

[List of AMD Opteron microprocessors](#)

Bulldozer core 3 o 4 moduli (6 o 8 threads) supporterà l'SMT

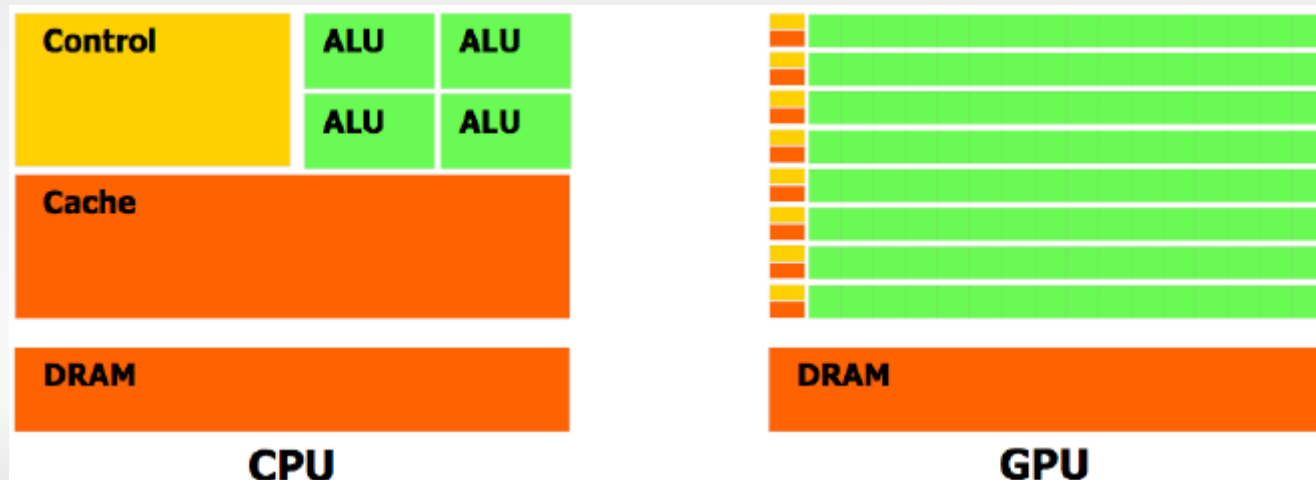
Cell (Playstation)

- Cell è un processore multi-core con 9 core, dei quali uno è (il PowerProcessingElement) è il coordinatore e gli altri (SynergisticPE) eseguono parallelamente i calcoli assegnati
- Livello teorico:
 - 204 Gflops in precisione singola
 - 25 precisione doppia
 - 64 e 8 volte la velocità di un P4 con pari clock



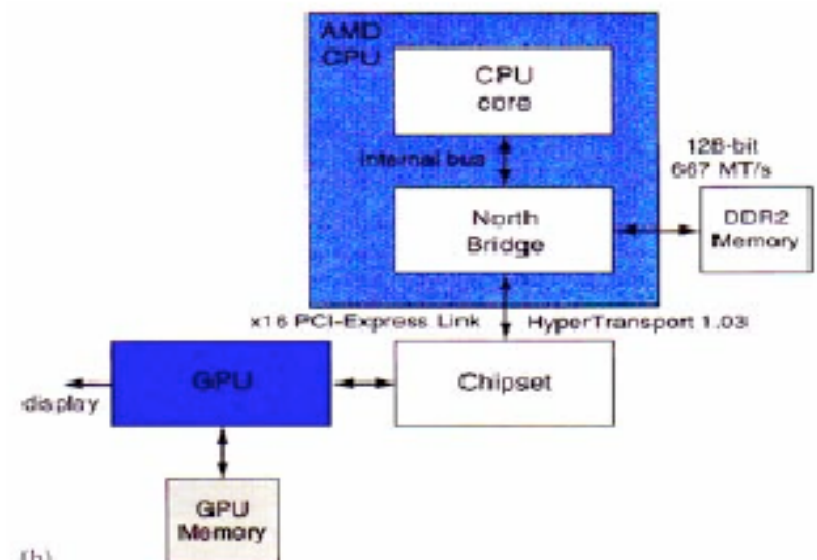
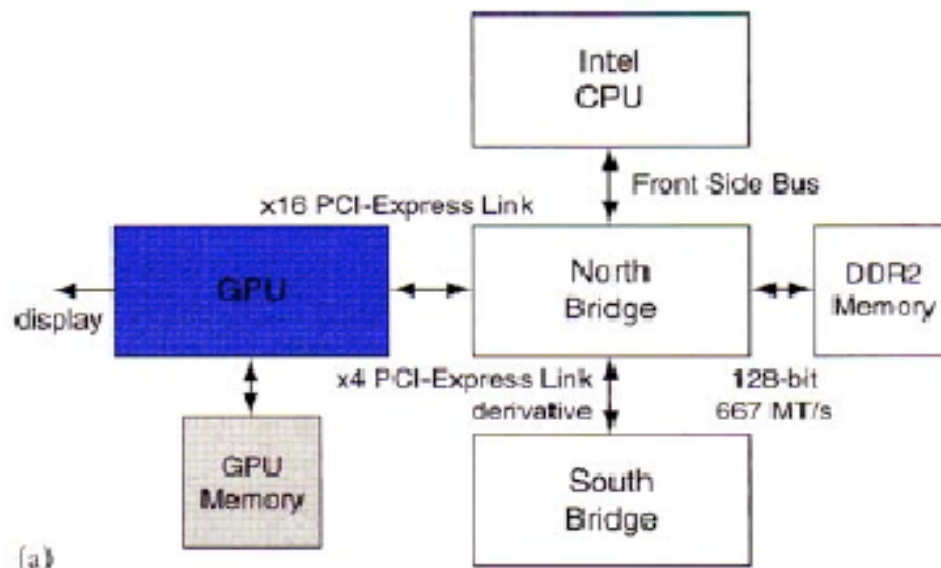
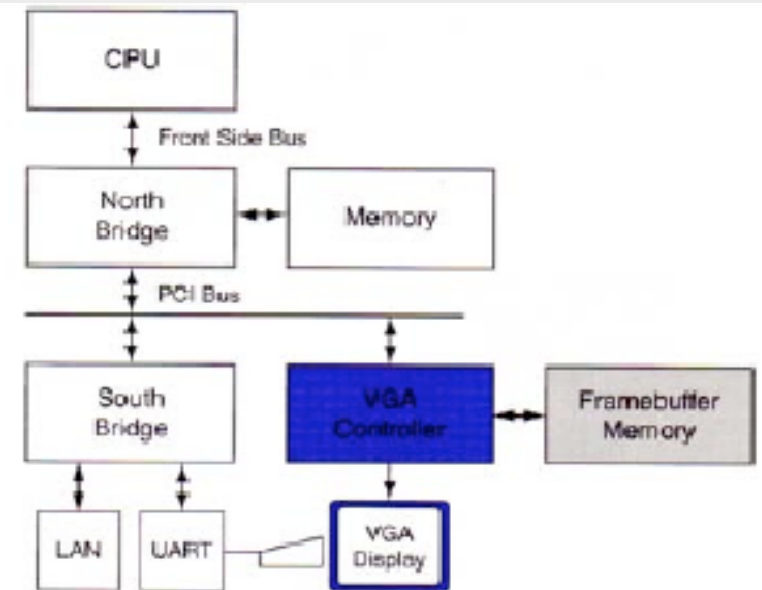
GPU

- Le GPU sono processori specializzati per impieghi fortemente paralleli e di calcolo intensivo
- Stesso algoritmo su molti dati (SIMD)
- Memorie semplici a bassa latenza



Architettura PC/GPU

- Architettura PC tradizionale (1990)
 - Video=Interfaccia
 - Bus PCI
- Architettura PC Moderno
 - GPU
 - Memoria dedicata
 - Bus veloce PCIE (16GB/s)



Approccio CUDA

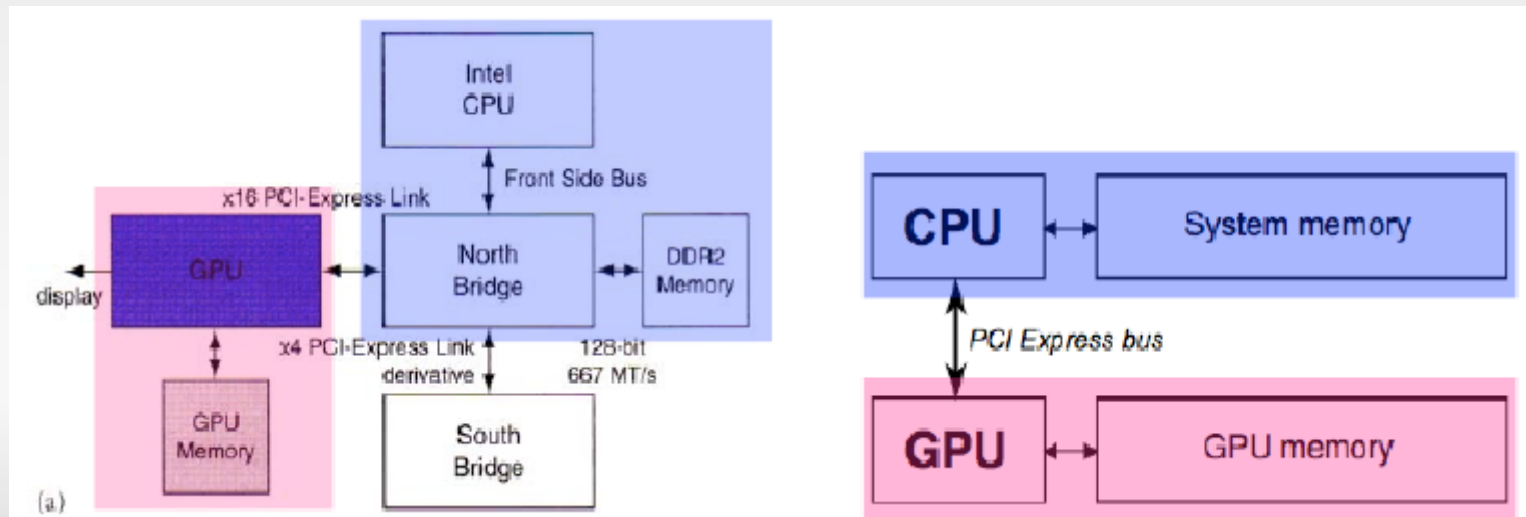
1. CPU invia dati alla GPU

(CPU memory \rightarrow GPU memory)

2. GPU elabora i dati e elabora i risultati

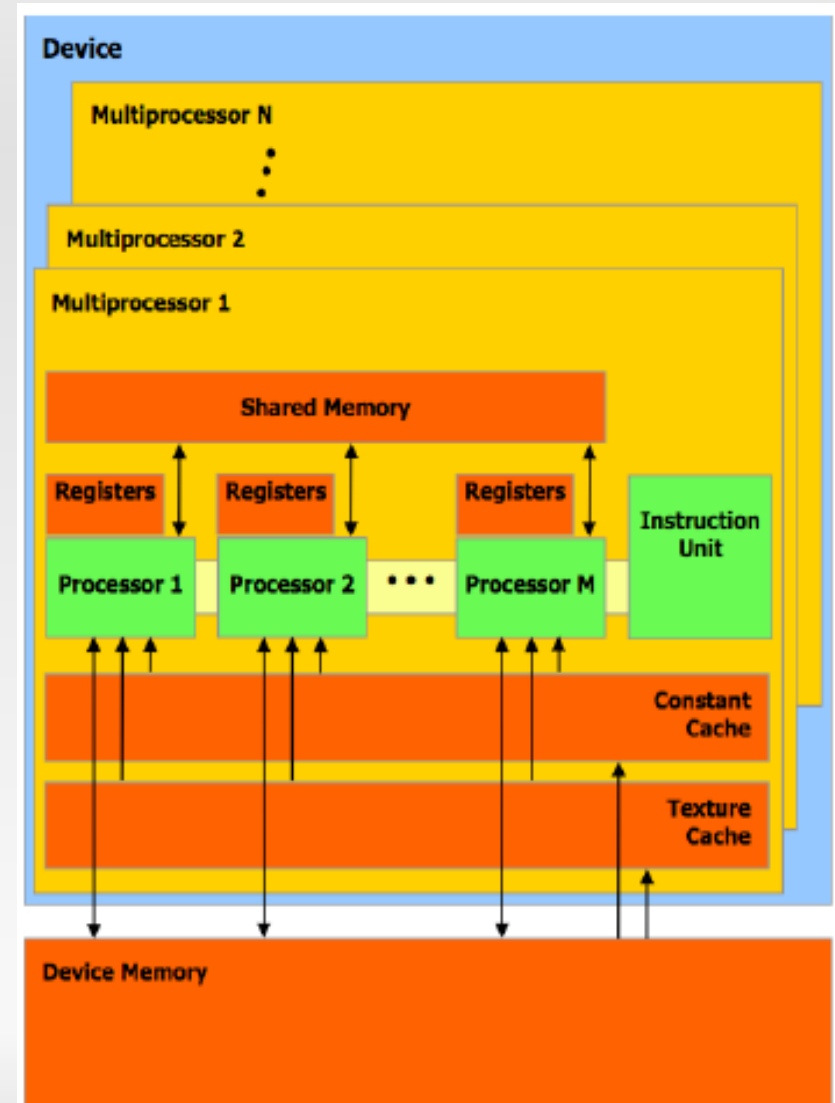
3. CPU recupera i risultati

(GPU memory \rightarrow CPU memory)



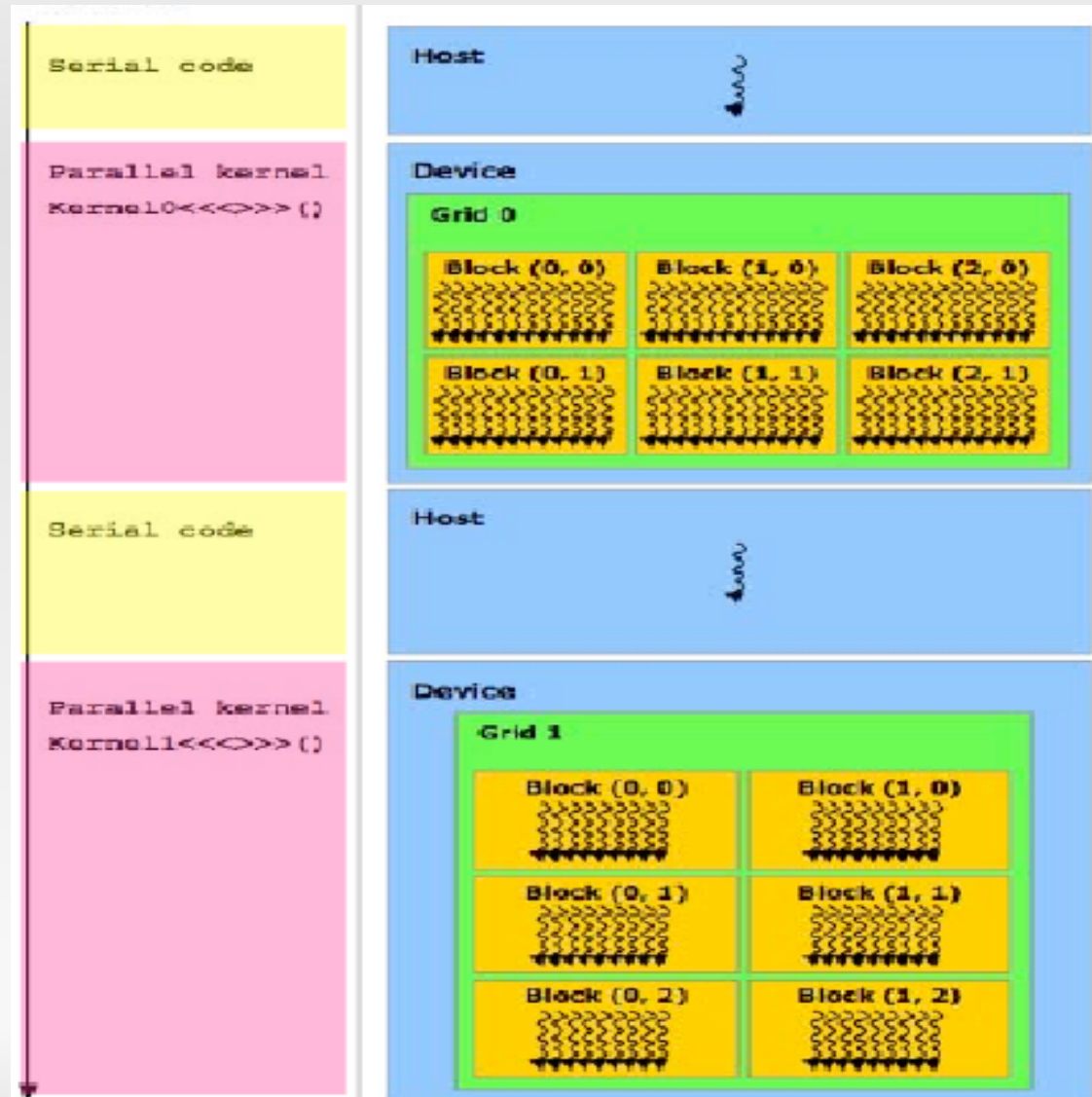
CUDA: architettura hardware

- Device CUDA: array di Multiprocessori
- Streaming Multiprocessor (SM)
 - Uno SM contiene M processori, in grado di eseguire threads in parallelo in modalità SIMD
 - Uno SM è in grado di eseguire 32 threads
- Organizzazione della memoria
 - Register files: spazio privato di ogni singolo processore
 - Shared memory: condiviso tra i processori di uno stesso SM
 - Read-only constant cache: per la lettura accelerata di costanti
 - Texture cache: per la lettura di texture
 - Device memory: spazio condiviso tra tutti gli SM



Modello di esecuzione

- Codice seriale (CPU) alternato a codice parallelo (GPU)
- Codice parallelo lanciato sulla GPU come kernel
 - GPU esegue un kernel alla volta
- Ogni block viene eseguito da un solo multiprocessore
 - Non può essere spezzato
 - Più block possono essere eseguiti sullo stesso SM
 - Tutti i thread di un block condividono la shared memory
 - Il numero max di threads dipende dalle dimensioni del SM
 - Il register file e la shared memory vengono suddivisi tra tutti i threads residenti



Riferimenti

- Libri di testo
- *Computer Organization & Design* di D.A.Patterson e J.L.Hannessy - ed. Morgan Kaufmann Publishers
- *Architettura e Organizzazione dei Calcolatori* di W.Stallings - ed. PEARSON Addison Wesley - 6° edizione
- **Architettura e organizzazione dei Calcolatori Elettronici**
– **Strutture Avanzate** di Giacomo Bucci - ed. McGraw-Hill