Multicore-Architekturen

Philipp Wendler

Seminar
Multicore-Programmierung
am
Lehrstuhl für Programmierung

30. April 2009

Inhalt

Ziele

Vorstellung von

- Architekturen
- Technologien
- konkreten Chips

Zielgruppe:
Programmierer / Software-Entwickler

Schwerpunkt:
aktuelle und lieferbare Chips für General-Purpose-Computing und Stream-Computing

Definitionen

Definition

Ein **Prozessor** ist eine nicht teilbare Einheit von Rechen- und Hilfseinheiten.

Definition

Ein **Kern** ist ein Teil eines Prozessors, der alles enthält um mindestens einen Thread **eigenständig** ausführen zu können.

Definition

Ein **Multicore**-Prozessor ist **ein** Prozessor auf dem **mehrere** Threads **gleichzeitig** und **unabhängig** voneinander ausgeführt werden können.

Parallelität in Hardware

In einem System können Operationen auf verschiedene Arten parallel ausgeführt werden:

- als SIMD-Operationen innerhalb einer Funktionseinheit (z.B. MMX, SSE)
- in SPMD-Threads innerhalb eines Kerns
- in MPMD-Threads
  - innerhalb eines Kerns
    ($\longrightarrow$ Hardware-Threads, Hyperthreading)
  - in verschiedenen Kernen eines Prozessors
  - in verschiedenen Prozessoren (Multiprozessor-Systeme)

Parallelität in Hardware

SxMD, Hardware-Threads und Multiprozessor-System sind keine Multicores.

Aber:

- für Anwendungsentwickler oft kein Unterschied feststellbar
- Die Entscheidung, welche Threads wie verteilt laufen, kann performance-entscheidend sein
  $\Rightarrow$ für Betriebssystem-Scheduler und HPC wichtig
- treten oft zusammen mit Multicores auf
- hohe (> 8) Hardware-Parallelität oft nur so erreichbar
- werden benutzt um Zahlen zu beschönigen

SPMD

- ein Thread wird gleichzeitig für verschiedene Daten ausgeführt
- Funktionseinheiten und Register mehrfach vorhanden, Instruction Counter nicht
- nicht für alle Aufgaben brauchbar
- Größenordnung der Parallelität: 2 bis 96

Hardware-Threads

- ein Kern führt mehrere Threads (fast) parallel aus
- Betriebssystem sieht mehr „Kerne“ als tatsächlich vorhanden
- führt zu besserer Auslastung der Funktionseinheiten, hilft Wartezeiten des Prozessors zu überbrücken
- nur Hilfseinheiten mehrfach (z.B. Register)
- bekannt geworden mit Pentium 4 (Hyper-Threading)
- billiger als Multicores, aber Performance-Gewinn kleiner
- Größenordnung der Parallelität: 2 bis 4

Multicores

- mehrere unabhängige Threads parallel
  ähnlich zu mehreren Prozessoren in einem Chip
- alle wichtigen Einheiten pro Kern vorhanden
  (ALUs, Instruction Counter, Register)
- evtl. gemeinsamer Cache
- gemeinsame Anbindung nach außen
- Größenordnung der Parallelität: 2 bis 64
- in Spezialgebieten (z.B. DSPs): mehrere hundert Kerne

Homogene Multicores

Alle Kerne eines Chips sind identisch

- heutzutage am weitesten verbreitet
- einfach für Hardware- und Software-Entwickler
- nicht immer das Performance- und Effizienz-Optimum

Heterogene Multicores

Unterschiedliche Kerne in einem Chip

Verschiedene Ansätze:

- ein herkömmlicher Kern für Betriebssystem und Anwendungen,
  mehrere spezialisierte Kerne für Berechnungen
- viele Kerne, davon einige für Spezialaufgaben (z.B. Verschlüsselung, Video-Decoding) optimiert

Noch selten, aber in Zukunft sicher aktuell.

Kommunikation

Verwendete Kommunikationsarten:

- keine direkte Kommunikation
  (nur über gemeinsamen Speicher)
- keine explizite Kommunikation
  (gemeinsamer Speicher mit getrenntem Cache und Kommunikation für die Cache-Synchronisation)
- durch DMA-Transfers zwischen lokalem Speicher
- theoretisch möglich: expliziter Nachrichtenversand

Topologien

- Verwendung bekannter Multiprozessor-Topologien:
  - Bus
  - Ring (oft bi-direktional)
  - Grid
- komplexere Topologien werden nicht verwendet, da 3D in Chips sehr teuer ist
- Anschluss an crossbar switch (n:m-Kommunikation mit n gleichzeitigen Punkt-zu-Punkt-Transfers)

Speicherverteilung

- globaler Hauptspeicher
  - Zugriff für alle Kerne gleich schnell
  - nicht immer mit Cache
- oft zusätzlich lokaler Speicher pro Kern („Scratchpad“)
  - typischerweise 16 bis 256 KB
  - muss explizit angesprochen werden
  - kein Zugriff von anderen Kernen
  - Anwendung ist selbst für Kohärenz verantwortlich

Cache-Topologien

Viele verschiedene Topologien möglich:

- kein Cache (dann existiert lokaler Speicher)
- gemeinsamer L2-Cache, getrennter L1-Cache
- gemeinsamer L2-Cache für Gruppen von Kernen
- getrennte L1- und L2-Caches,
  evtl. mit gemeinsamen L3-Cache

Caches haben hardware-seitig garantierte Kohärenz.

Zusätzliche Hilfsmittel in Hardware

Manche Chips bieten explizite Hilfsmittel zur Verwaltung der Parallelität:

- atomare Funktionen (Addition, compare-and-swap)
- (schnelle) Barrier-Synchronisation
- Hardware-Transaktionsspeicher
  (derzeit nur in Prototypen)

Sun UltraSPARC T2 (Niagara 2)

- seit 2007 auf dem Markt
- 8 Kerne
- 8 Hardware-Threads pro Kern um Wartezeiten auf Speicher zu minimieren
- einige Einheiten doppelt pro Kern vorhanden, meistens können 2 Threads parallel ausgeführt werden (billiger als doppelte Anzahl Kerne)
- gemeinsamer L2-Cache (aufgeteilt in 8 Bänke)
- Crossbar switch verbindet alle Kerne mit allen Cache-Banks und I/O

Sun UltraSPARC T2 (Niagara 2)

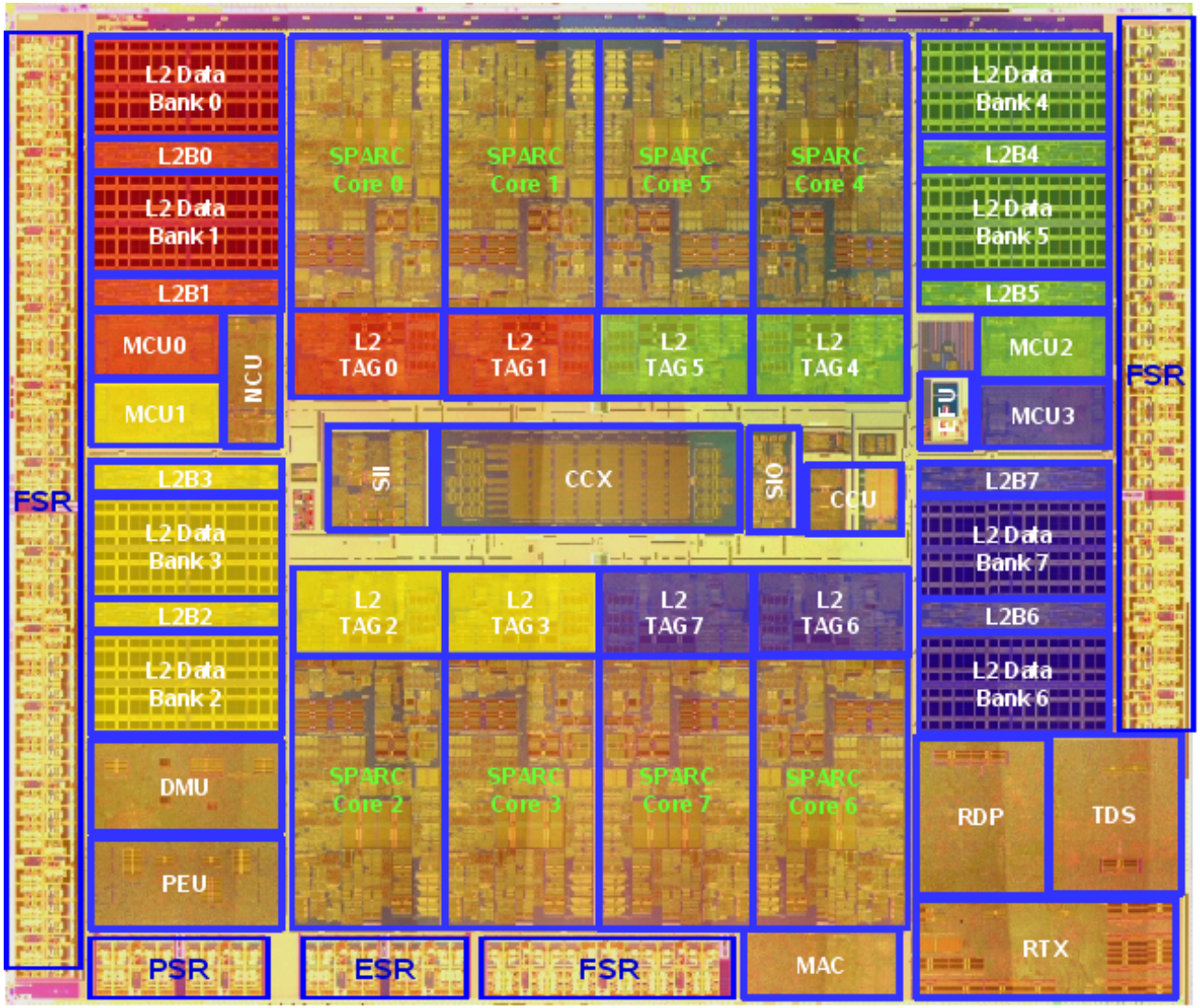


Foto: `http://www.opensparc.net`

GPGPU Chips

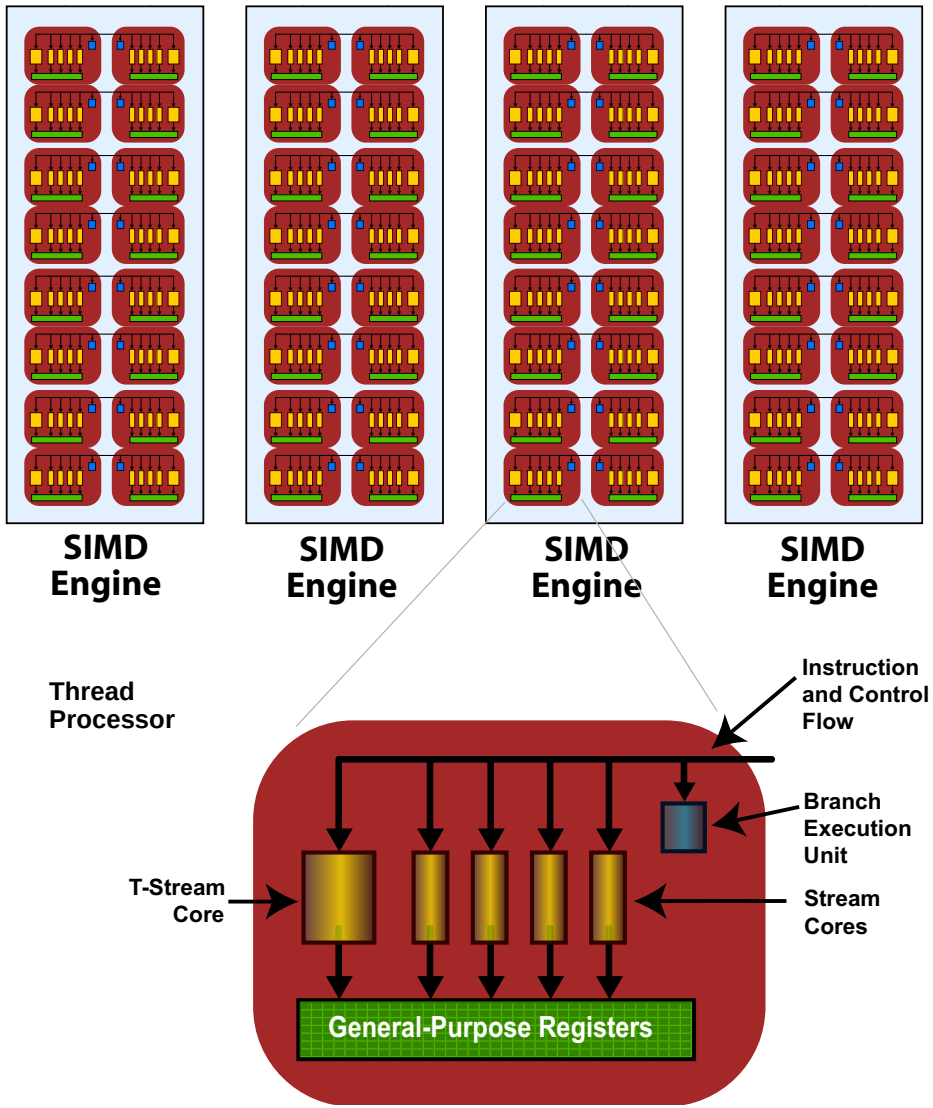
GPGPU bedeutet „General Purpose GPU“

- aktuelle Chips von AMD / ATI und Nvidia (seit ca. 2007)
- bis zu 30 SPMD-Einheiten mit je bis zu 16 Threads
- zusätzlich SIMD

$\Rightarrow$ 800(!) parallele Berechnungen möglich

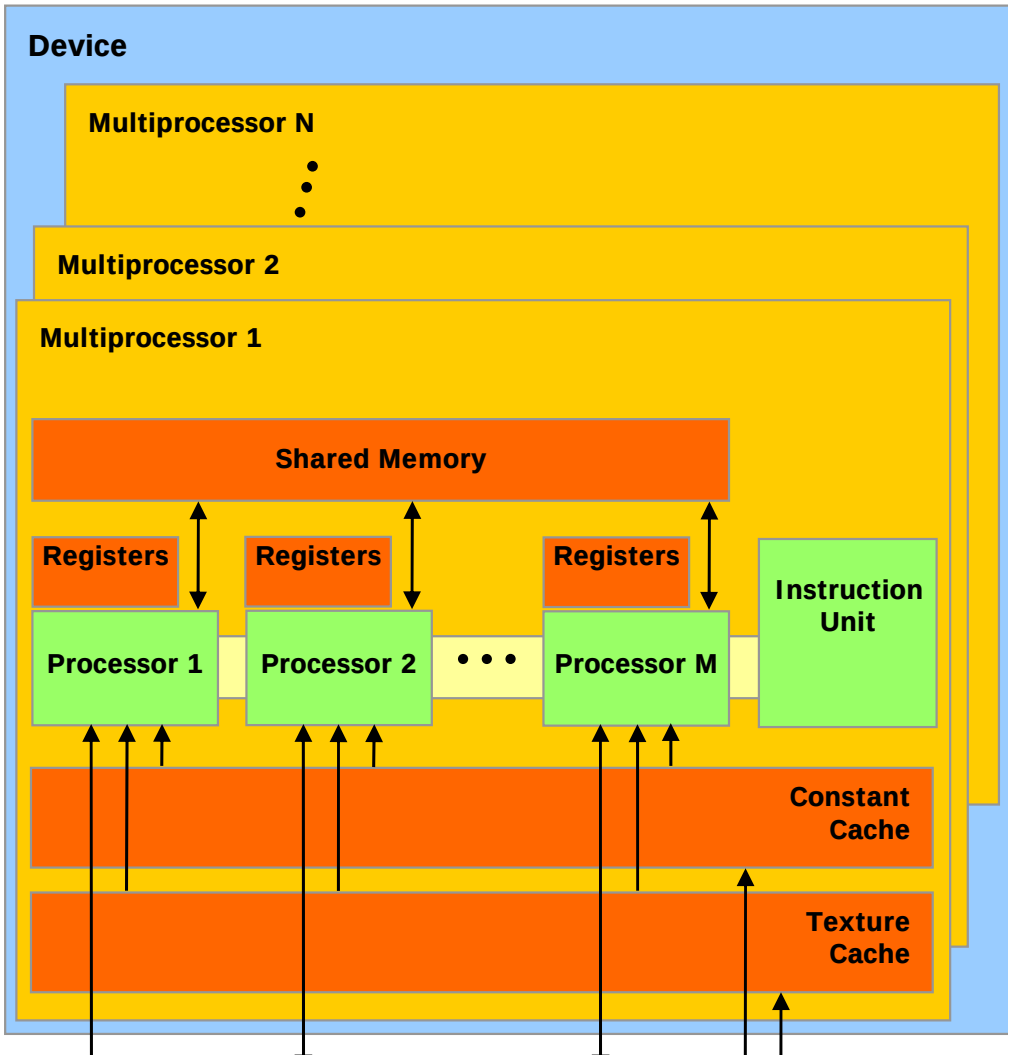
- schnelle Barrier-Synchronisation in SPMD-Einheiten, atomare Funktionen in Hardware
- schneller lokaler Speicher pro SPMD-Einheit

ATI Stream Processor



Grafik: ATI Stream Computing – Technical Overview [ati09]

Nvidia GPU



Grafik: NVIDIA CUDA Compute United Device Architecture 2.0, [nvi08]

GPGPU Chips – Einschränkungen

- eingeschränkte Möglichkeiten
  (kein dynamischer Speicher, keine Rekursion)
- spezielle Programmiersprachen müssen benutzt werden
  (Brook+, CUDA, demnächst OpenCL)
- keinerlei Code-Kompatibilität
- Kommunikation zwischen SPMD-Einheiten nur über Hauptspeicher (langsam)
- kein Cache, lokaler Speicher muss explizit benutzt werden

IBM / Sony / Toshiba Cell Broadband Engine

- seit 2006 verfügbar
- 1 PowerPC-Kern mit 2 Hardware-Threads
- 8 Synergistic Processing Elements
  (Spezial-Kerne für Stream-Computing)
- verbunden über bi-direktionalen Ring
- SPEs können nur auf 256 KB lokalen Speicher zugreifen
- DMA für Kommunikation SPE $\leftrightarrow$ SPE
  und SPE $\leftrightarrow$ Hauptspeicher
- DMA-Transfers voll kohärent

IBM / Sony / Toshiba Cell Broadband Engine

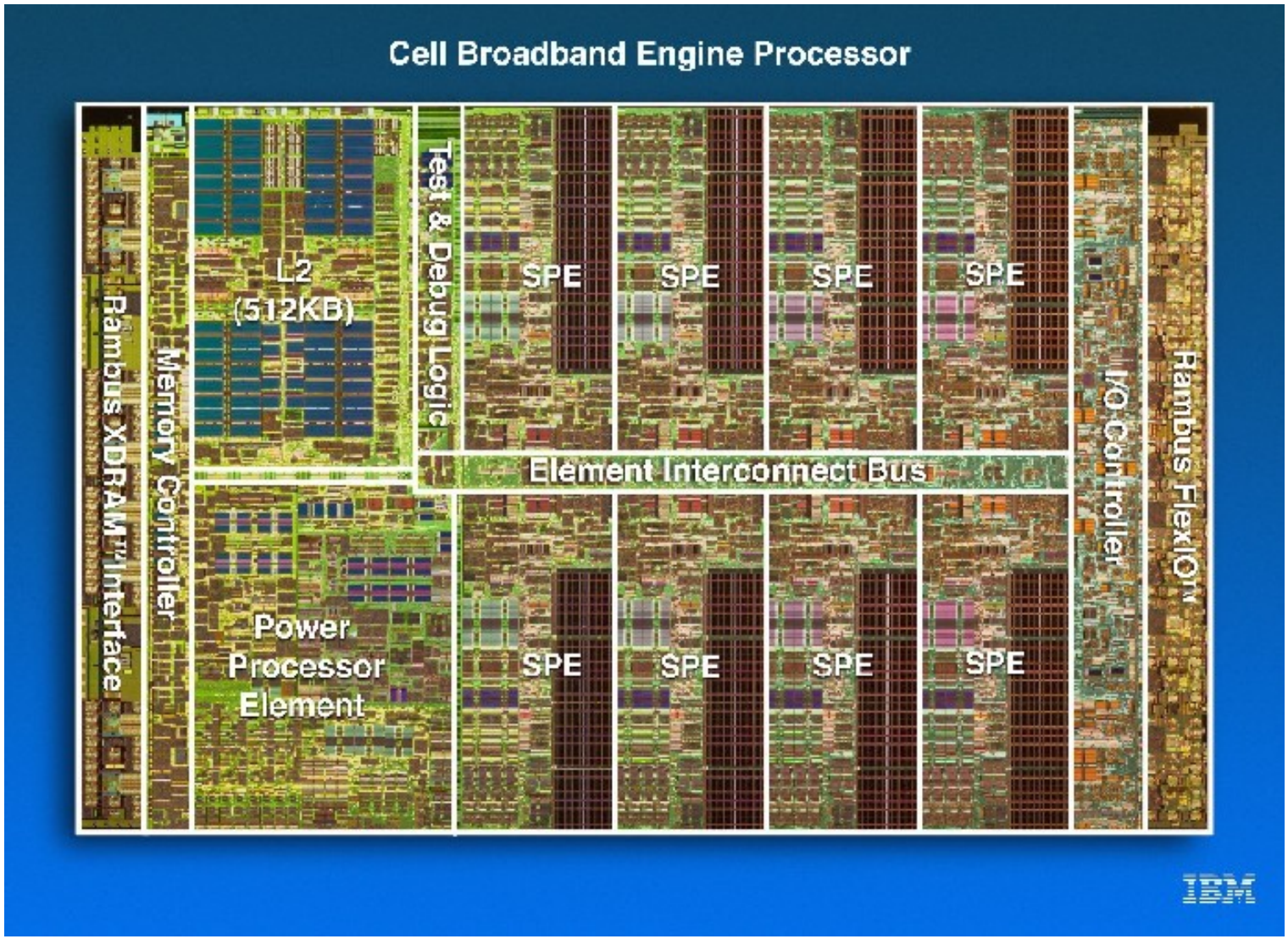


Foto: IBM, `http://www.research.ibm.com/cell/cell_chip.html`

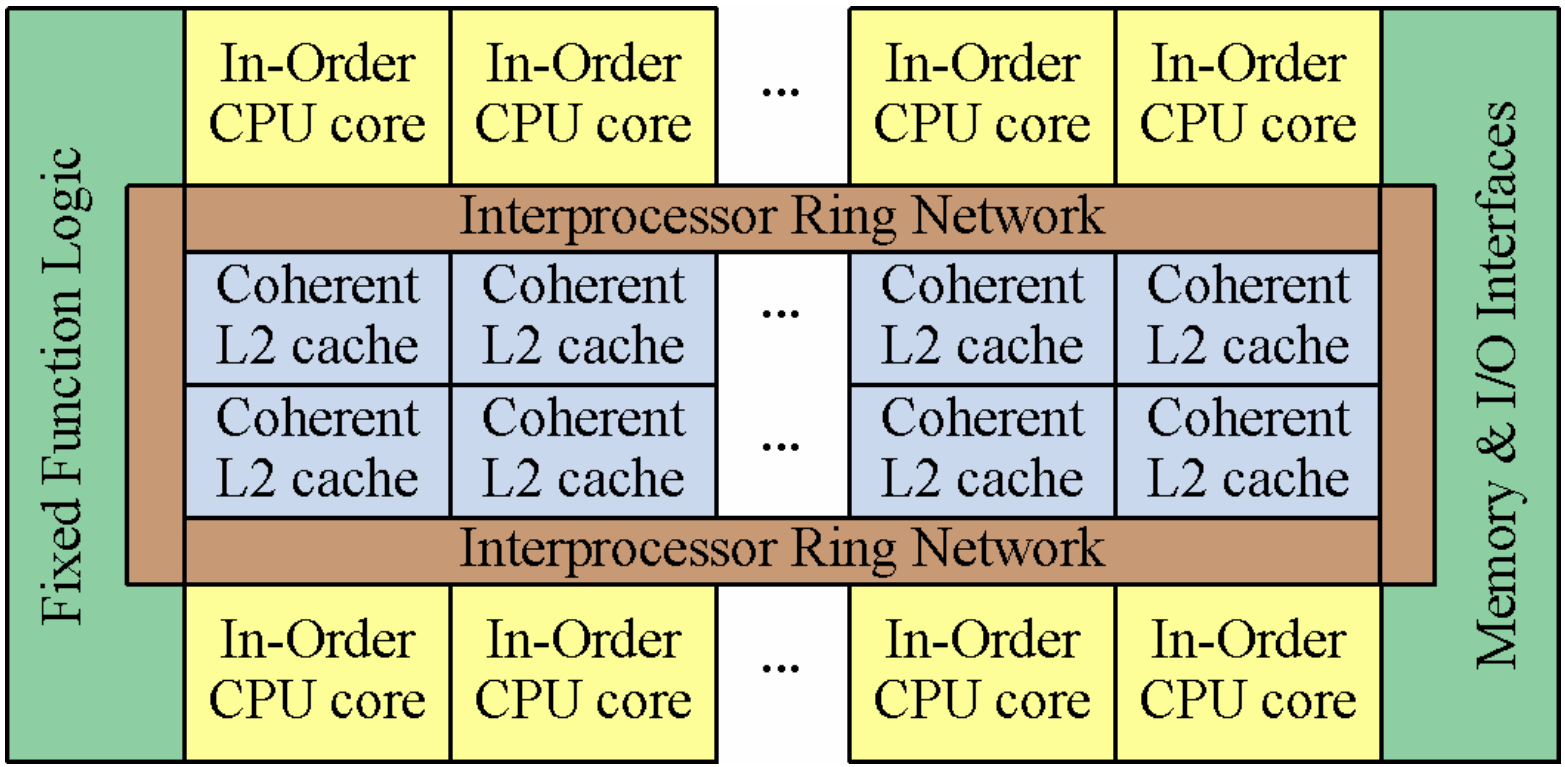
Intel Larrabee

- Intels Antwort auf GPGPU,
  sozusagen „GPCPU“ (Graphic Processing CPU)
- erwartet Ende 2009 / Anfang 2010
- ein x86-Multicore (kompatibel zu existierendem Code)
- unterschiedliche Konfigurationen denkbar,
  als Graphikprozessor mit Hardware-Textureinheiten
- initial wahrscheinlich 32 Kerne mit 4 Hardware-Threads
- jeder Kern relativ einfach
- 16-fach-parallele SIMD-Einheiten mit scather & gather

Intel Larrabee

- Level 2-Cache pro Kern, voll kohärent
- kein lokaler Speicher,
  aber: explizite Befehle zur Steuerung des L2-Cache
  (prefetch & evict)
- Kerne sind über einen oder mehrere bi-direktionale Ringe verbunden
- keine explizite Kommunikation,
  nur über Cache-Synchronisation
- Intel unterstützt explizit POSIX-Threads und OpenMP

Intel Larrabee

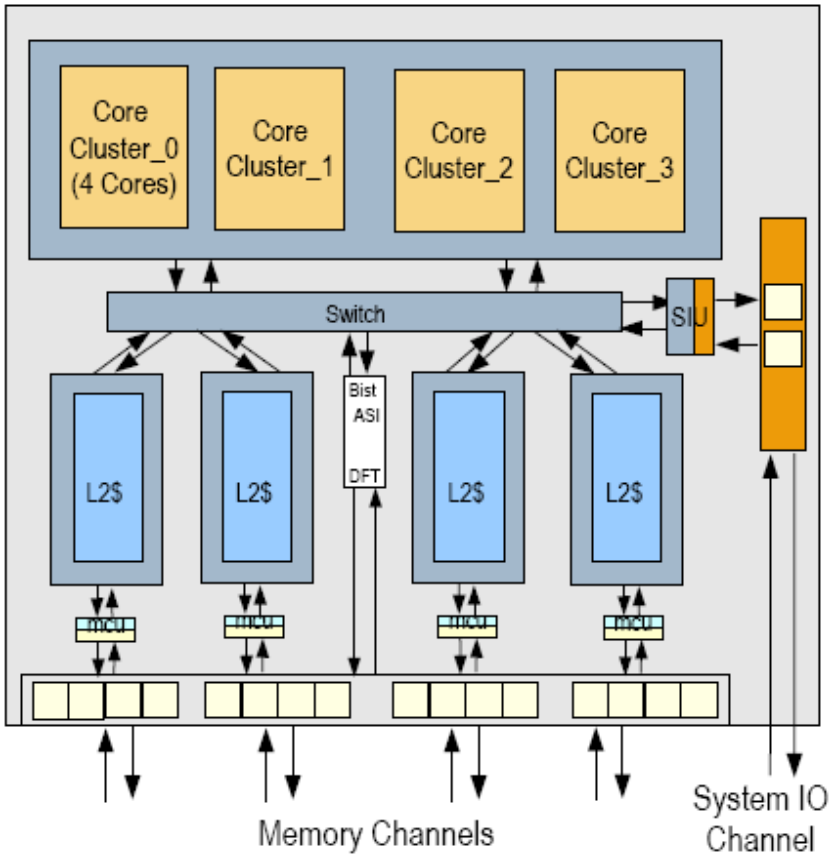


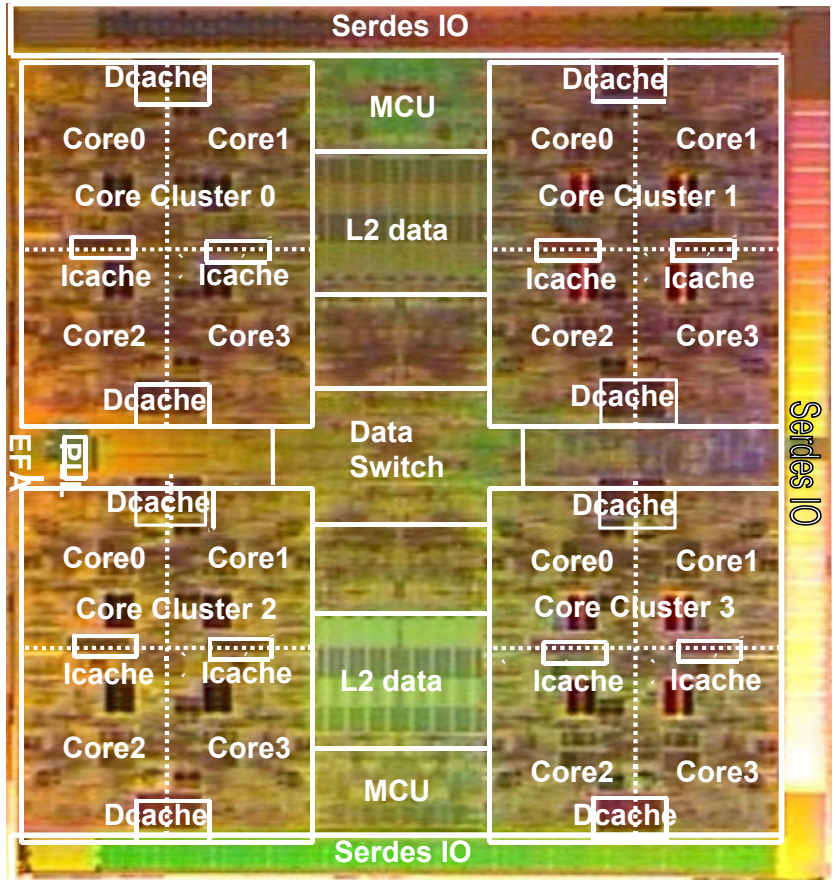
Grafik: Larrabee: a many-core x86 architecture [SCS+08]

Sun Rock

- erwartet noch 2009, funktionsfähige Prototypen seit 2007
- 16 Kerne
- 2 Threads pro Kern,
  plus 2 „Scout“-Threads (spekulative Ausführung)
- je 4 Kerne zu einer Gruppe zusammengefasst mit gemeinsamem L1-Cache
- Crossbar switch verbindet Gruppen aus Kernen mit L2-Cache-Bänken
- Transaktionsspeicher
  - zusätzliche Befehle für „begin“ und „commit“
  - realisiert über gesperrte Cache-Lines

Sun Rock





Grafik: A Third-Generation 65nm 16-Core 32-Thread Plus 32-Scout-Thread CMT SPARC Processor [TC08]

Zusammenfassung

Es gibt sehr viele Multicore-Prozessoren.

Es gibt sehr große Architektur-Unterschiede.

Es ist wichtig, für eine Aufgabe den passenden Chip auszuwählen!