

Kapitel 4.2 - Grundlagen zur Parallelverarbeitung

SWT I – Sommersemester 2010

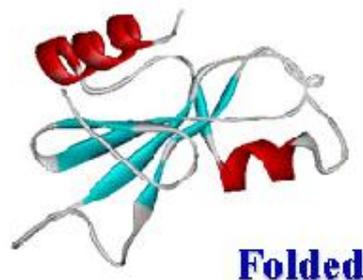
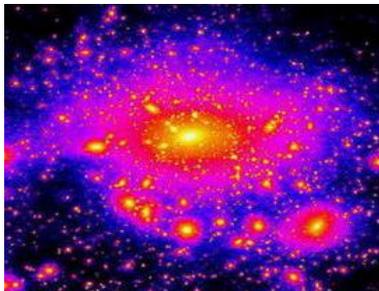
Walter F. Tichy, Andreas Höfer, Korbinian Molitorisz

IPD Tichy, Fakultät für Informatik



Warum Parallelverarbeitung?

- Anwendungen brauchen **mehr Leistung** oder **mehr Ressourcen** als Ein-Prozessorsysteme bieten können.
- Warten auf nächste Hardware-Generation hilft nicht mehr, denn Taktfrequenzen steigen kaum noch, Parallelität auf Instruktionsebene ist **ausgeschöpft**.
- "Grand-Challenge" Applikationen aus Chemie, Astronomie, Bioinformatik, CAD/CAM, ...



- Simulationen, wenn Experimente unmöglich sind

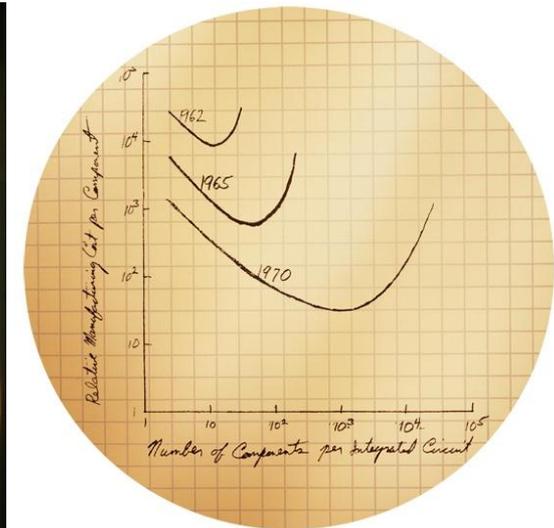
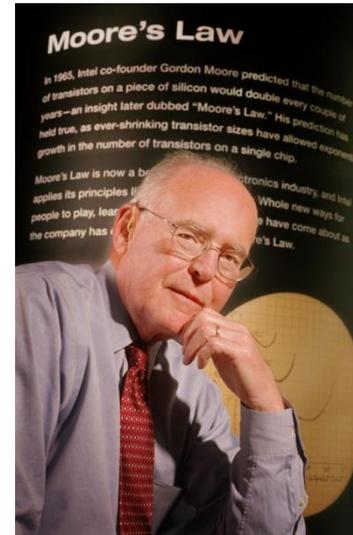
[physorg.com]

[IBM]

[Linux-Magazin.com]

Die Moore'sche Regel (Moore's Law)

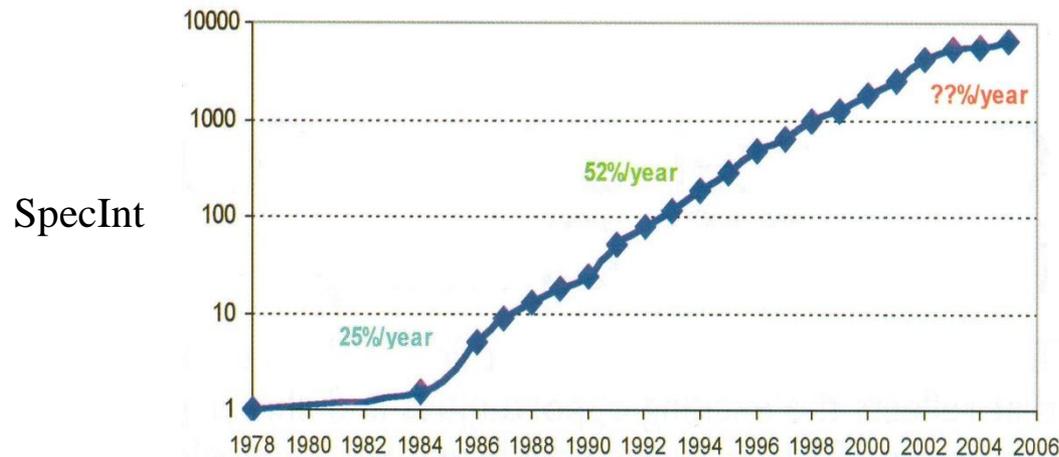
- Moore, G.E.:
 “The complexity for minimum component costs has increased at a rate of roughly a factor of two **per year.**” (1965)
- Moore, G.E.:
 “The new slope might approximate a doubling **every two years**, rather than every year, by the end of the decade.” (1975)



Quelle: <http://www.intel.com/technology/mooreslaw/>

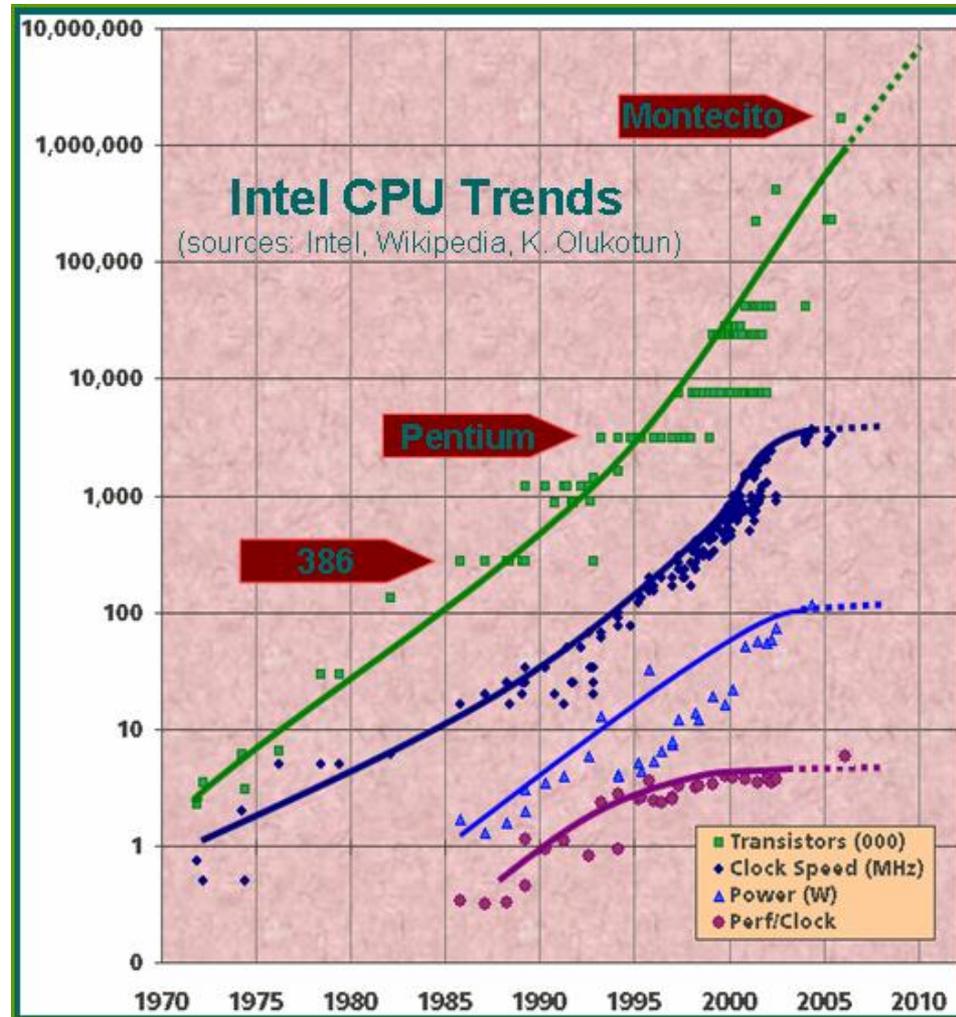
David Patterson, Präsident ACM:

„In case you weren't paying attention, the era of doubling performance every 18 months ended in 2002.“



President's Letter,
Comm of ACM,
März 2006

Was ist passiert?



Strategische Änderung: Multiprozessoren auf einem Chip

- Hersteller haben eine kritische Schwelle überschritten: Mehrere Prozessorkerne pro Chip sind **möglich** und **preiswert**:

Hersteller	#Kerne/Chip	etwa seit
IBM	2	2004
IBM	9	2005
AMD	2	2005
AMD	4	2007
Intel	2	2006
Intel	4	2007
Intel	6	2008
SUN	8	2005

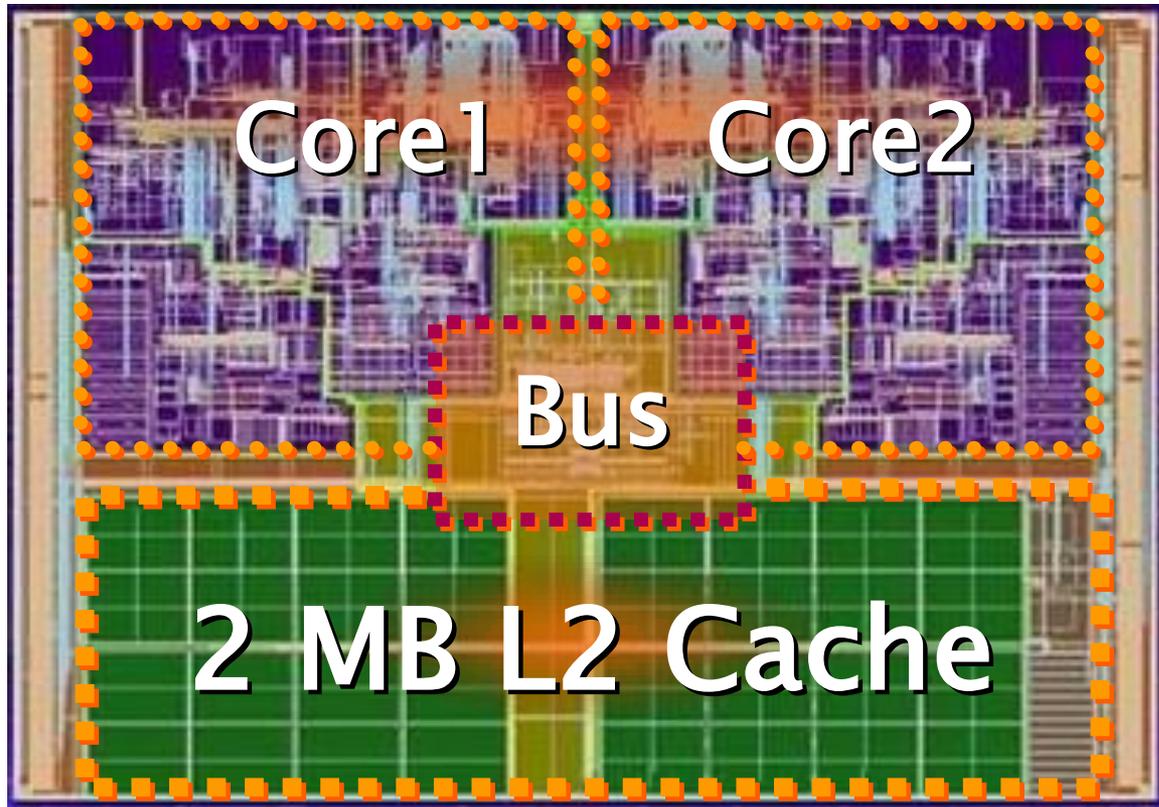
- Wachsende Anzahl von Prozessorkernen pro Chip sind prognostiziert (wegen weiterem Schrumpfen der Abmessungen der Transistoren, Leitungen, etc.)

Ihr Laptop – ein Parallelrechner?

 <p>Kundenbewertung ★★★★☆ 4,0 von 5</p> <p>Listenpreis 1.000 € Sparpotenzial* 471 €</p> <p>Preis ab 529 € inkl. MwSt. und 30 € Online-Rabatt, zzgl. 29 € Versand</p> <p>Leasing Sonderangebote</p> <p>auswählen Anpassen</p> <p>Prozessor Intel® Pentium® Dual Core Prozessor ULV SU4100 (1,30 GHz, 800 MHz, 2 MB cache)</p>	 <p>Kundenbewertung ★★★★☆ 4,0 von 5</p> <p>Preis ab 549 € inkl. MwSt. und 30 € Online-Rabatt, zzgl. 29 € Versand</p> <p>Leasing Sonderangebote</p> <p>auswählen Anpassen</p> <p>Prozessor Intel® Pentium® Dual-Core Prozessor ULV SU4100 (1,30 GHz, 800 MHz, 2MB cache) UMA GFX</p>	 <p>Kundenbewertung ★★★★☆ 4,3 von 5</p> <p>Listenpreis 725 € Sparpotenzial* 146 €</p> <p>Preis ab 579 € inkl. MwSt. und 30 € Online-Rabatt, zzgl. 29 € Versand</p> <p>Leasing Sonderangebote</p> <p>auswählen Anpassen</p> <p>Prozessor INTEL® CORE™ i3-330M (2,13 GHZ, 4 THREADS, 3 MB CACHE)</p>	 <p>Kundenbewertung ★★★★☆ 4,5 von 5</p> <p>Preis ab 649 € inkl. MwSt. und 30 € Online-Rabatt, zzgl. 29 € Versand</p> <p>Leasing Sonderangebote</p> <p>auswählen Anpassen</p> <p>Prozessor INTEL® CORE™ i5-430M (2,26 GHZ, 4 THREADS, TURBO BIS ZU 2,53 GHZ, 3 MB CACHE)</p>
--	---	---	---

Stand: Mai 2010

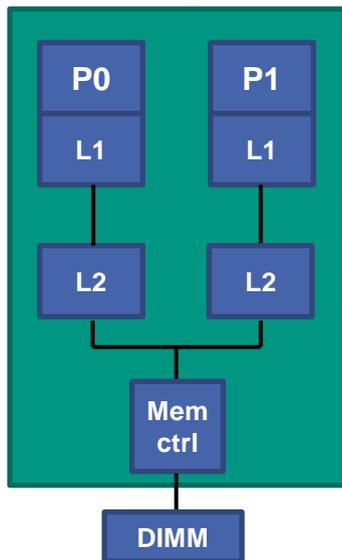
Doppelprozessor Intel Core Duo



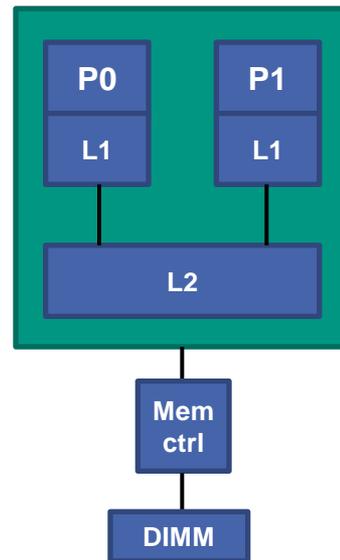
Mehrkern-Architekturen(2)

■ Dual-Core und Quad-Core

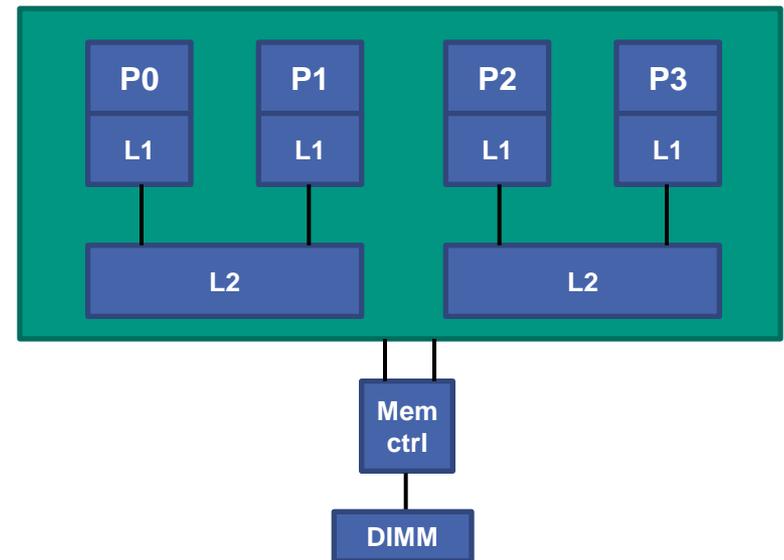
AMD Opteron



Intel Xeon



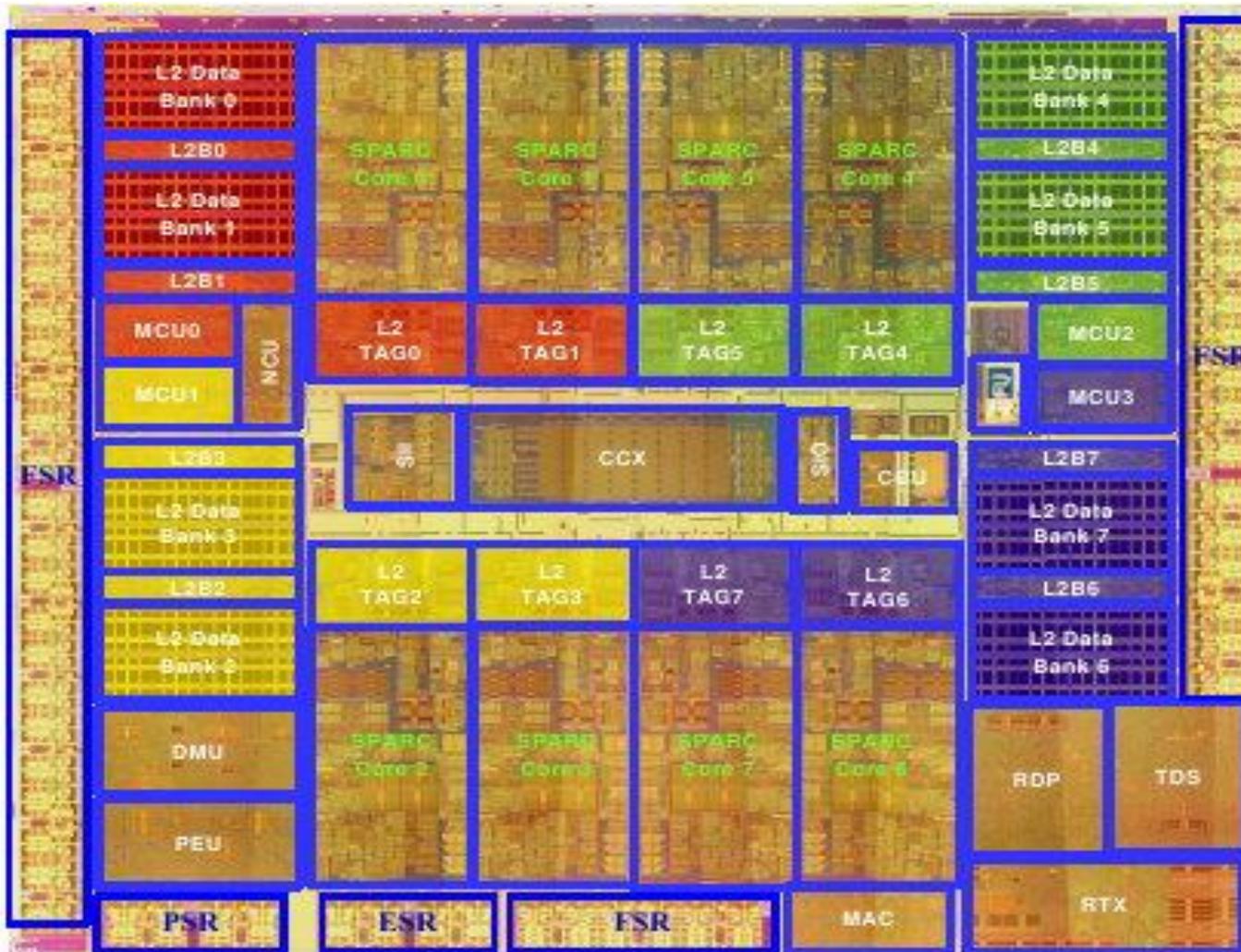
Intel Quad-Core



- Mehrere Prozessoren pro Chip
- Getrennte L1/L2-Caches, was zu divergenten Kopien der gleichen Variablen führt.
- Cache-Kohärenz nur bei bestimmten Ereignissen, z.B. beim Schreiben in den Speicher.

N. Aggarwal et al., Isolation in Commodity Multicore Processors, IEEE Computer, 2007, 40, 49-59

Sun Niagara 2: 8 Prozessoren auf 3,42 cm²



8 Sparc
Prozessoren

8 HW-Fäden pro
Prozessor

8x9 Kreuzschiene

1,4 GHz

75 W

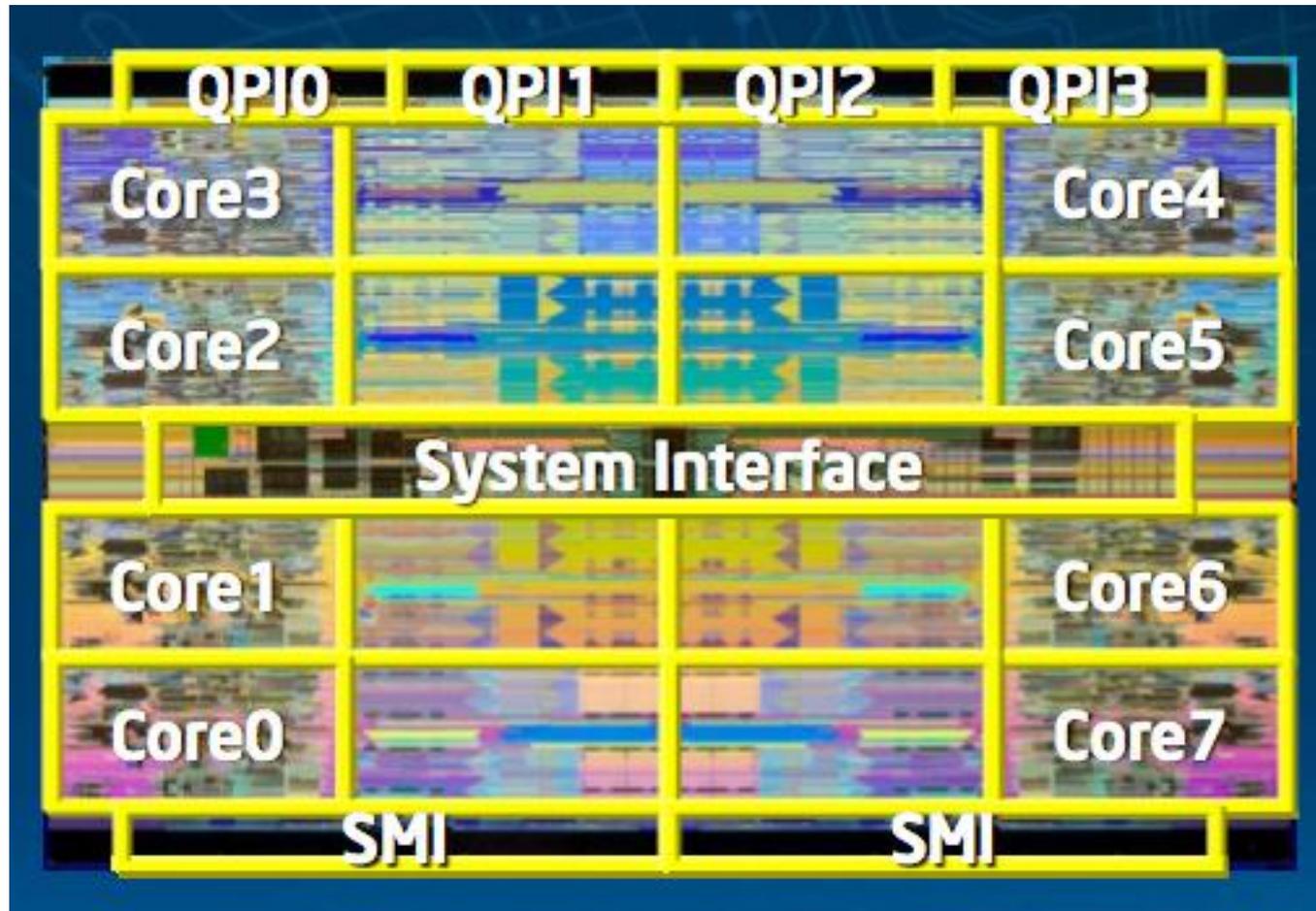
65nm Technik

Lieferbar 2007

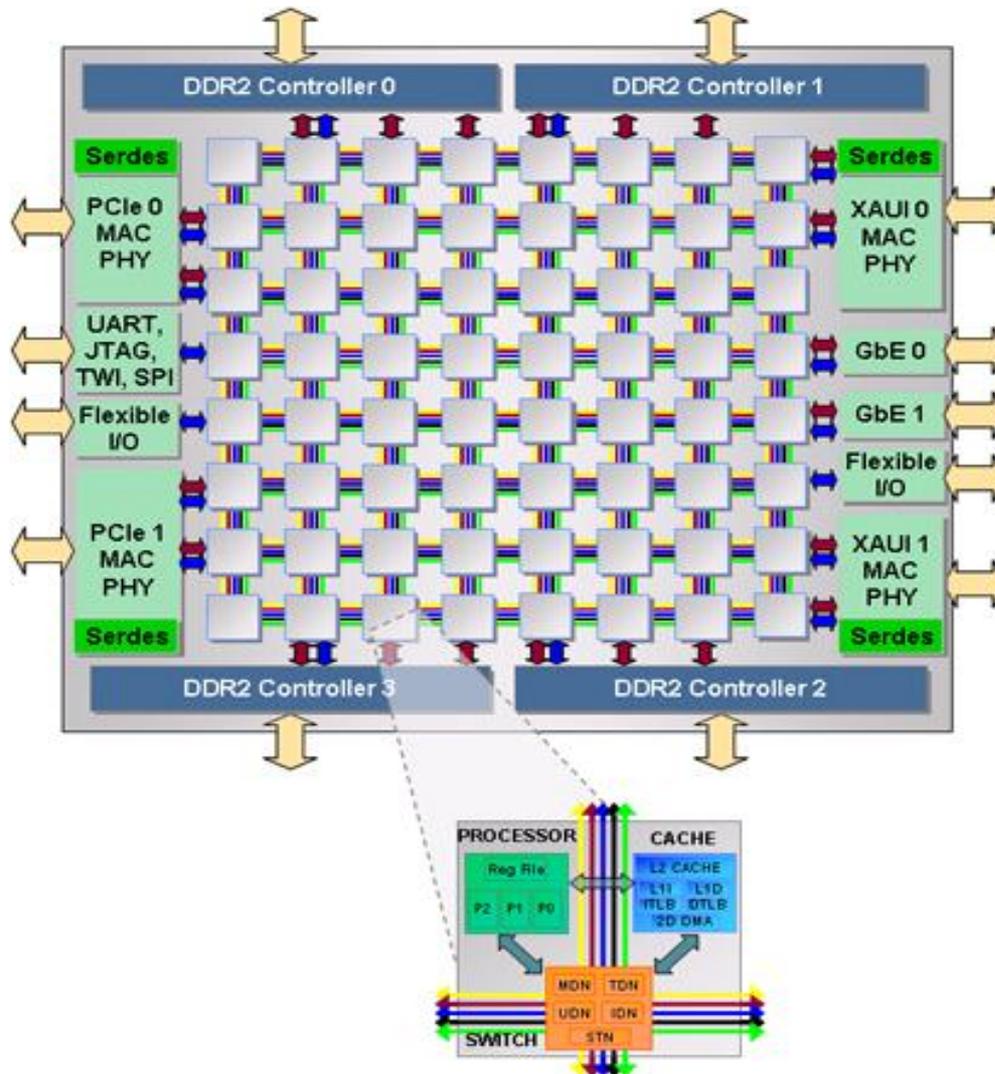
Erste Version 2005

Mehrkerner übernehmen die Prozessorlandschaft

- 2010: Intel Nehalem EX mit 8 Kernen, 16 Fäden



Tilera TILE64



64 VLIW Prozessoren plus Gitter auf einem Chip

Für eingebettete Anwendungen (Netz- und Videoanwendungen)

700 MHz

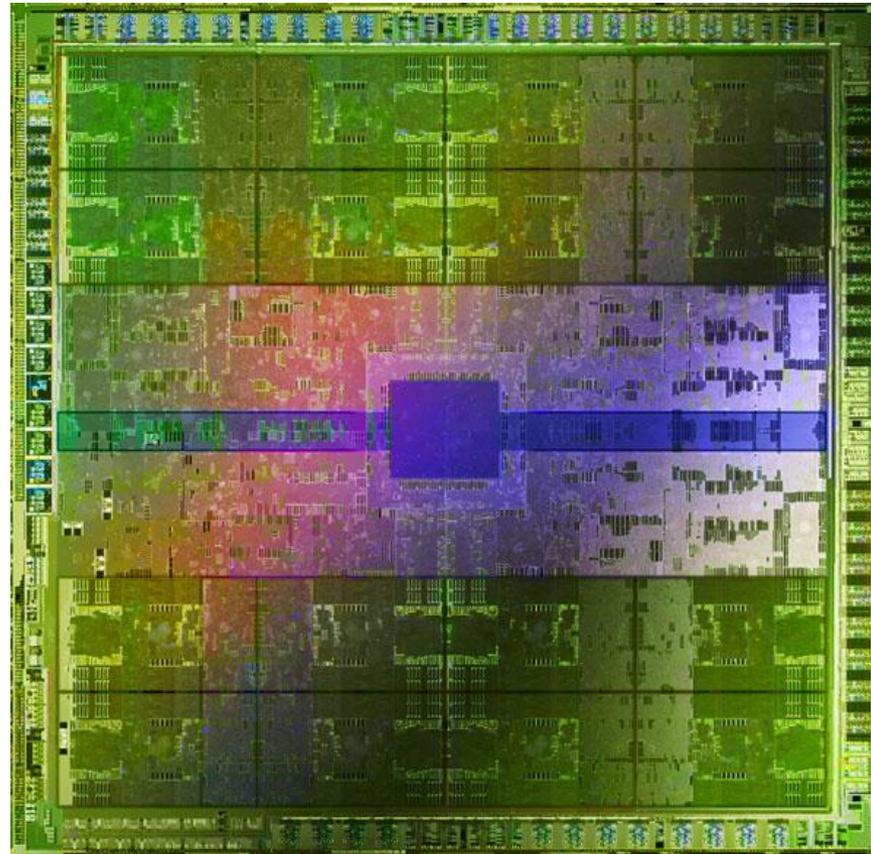
22 W

Lieferbar 2007

2011: 100 Prozessoren

Vielkerner in Graphik-Prozessoren

- 2010: Nvidia GeForce GTX 480M (Fermi) mit 352 Kernen



Moore'sche Regel, neue Version

Verdopplung der Anzahl
Prozessoren pro Chip
mit jeder Chip-Generation,
bei etwa gleicher Taktfrequenz

- **Parallelrechner** werden in naher Zukunft flächendeckend zur Verfügung stehen.

Was sind die Folgen? (1)

- Hauptproduktlinie der Prozessor-Hersteller sind **Multikern-Chips**.
- Server werden bereits seit 2005 mit Multikern-Chips ausgeliefert.
- Laptops werden seit 2006 mit Doppelprozessor-Chips ausgestattet (Dell, Apple, u.s.w.)
- Eingebettete Anwendungen (Steuerungsanwendungen, z.B. in Autos) werden auf Mehrkerner konsolidiert werden.

Was sind die Folgen? (2)

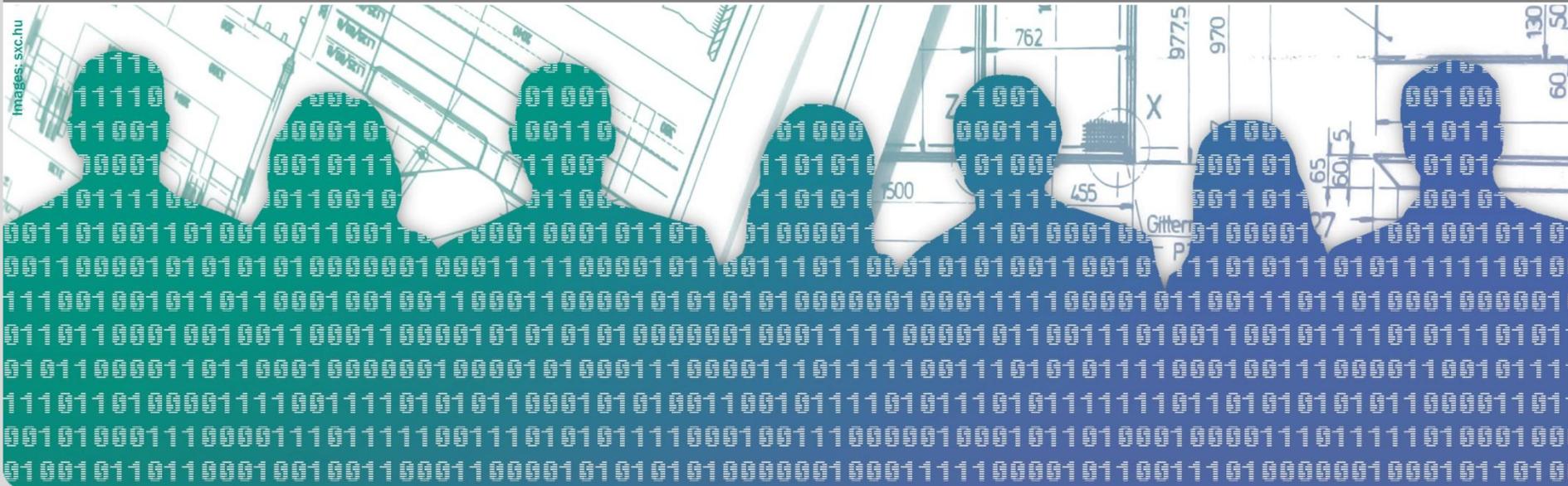
- Leistungssteigerungen von alltäglichen Anwendungen wird durch Parallelisierung erreicht.
- Parallelismus wird zum Normalfall.
- Informatiker müssen Parallelismus beherrschen lernen.
- **Fundamentaler Übergang** vom **sequentiellen** zum **parallelen** Rechnen auf breiter Front mit dramatischen Auswirkungen auf Anwendungen, Forschung, und Lehre.

Was sind die Folgen? (3)

- Die Softwarehersteller und die Informatikausbildung müssen rasch auf Parallelverarbeitung umstellen, um Wettbewerbsfähigkeit zu erhalten.
 - Umstellung von existierenden Anwendungen
 - Erstellung neuer, paralleler Anwendungen
- Wir bilden seit 2009 ab dem 2. Semester (SWT 1) in Parallelprogrammierung aus.
- Beherrschung der Parallelprogrammierung wird zur Job-Garantie werden.
- Noch viel mehr darüber in späteren Vorlesungen (Programmierparadigmen, Softwareentwicklung für moderne parallele Plattformen, Multikernrechner und Rechnerbündel, Praxis der Parallelprogrammierung, u.a.)

Grundlagen der Parallelverarbeitung

IPD Tichy, Fakultät für Informatik



Programmieransätze & Terminologie

Def.: **parallel** [griechisch]

Allgemein: nebeneinander verlaufend, in gleichem Abstand [Brockhaus]

Informatik: gleichzeitig ablaufend

- Anmerkung: In der Informatik-Literatur wird manchmal zwischen
 - „nebenläufig“ (im Sinne von „nicht kausal von einander abhängig“) und
 - „parallel“ (im Sinne von „simultan“) unterschieden.
- Diese Unterscheidung werden wir an keiner Stelle brauchen und verwenden daher beide Begriffe synonym.

Programmieransätze & Terminologie: Überblick

- Grundsätzlich zwei wichtige Programmieransätze:
- Für Parallelrechner mit gemeinsamem Speicher
 - Prozessoren haben einen gemeinsamen Speicherbereich, den sie gemeinsam benutzen können. Jeder Prozessor kann jede Speicherzelle ansprechen.
 - z.B. Anwendung bei Multikernrechnern
- Für Parallelrechner mit verteiletem Speicher
 - Jeder Prozessor hat seinen eigenen Speicher, der nur ihm zugänglich ist.
 - Zur Kommunikation schicken sich die Prozessoren Nachrichten (engl. *message passing*)
 - z.B. Anwendung bei Rechnerbündel
 - Wird in dieser Vorlesung nicht weiter behandelt.

Programmieransätze & Terminologie: gemeinsamer Speicher (1)

Prozess (engl. Process)

- Wird durch **Betriebssystem** erzeugt
- Enthält Informationen über **Programmressourcen** und **Ausführungszustand**, z.B.
 - Code-Segment (Programminstruktionen)
 - Daten-Segment (für globale Variablen, Halde)
 - Mind. 1 Kontrollfaden
- CPU-Kontextwechsel zwischen Prozessen langsam

Kontrollfaden (engl. Thread)

- Unabhängiger Instruktionsstrom, der ausgeführt werden kann
- Existiert in einem Prozess
- Ein Faden hat eigenen
 - Befehlszeiger
 - **Keller**
 - **Register-Kopien**
- Teilt sich mit anderen Fäden
 - **Adressraum**
 - Code/Daten-Segment
 - Andere Ressourcen (z.B. geöffnete Dateien, Sperren, etc.)
- CPU-Kontextwechsel zw. Fäden des gleichen Prozesses schneller als zw. unterschiedl. Prozessen.

Programmieransätze & Terminologie: gemeinsamer Speicher (2)

- Prinzipielles Vorgehen bei gemeinsamen Speicher
 - Fäden enthalten parallel auszuführende **Aufgaben** (Instruktionen)
 - Informationsaustausch über gemeinsam genutzte **Variablen** im Speicher
 - Synchronisationskonstrukte koordinieren Ausführung im Falle von Daten- oder Kontrollabhängigkeiten
 - Fäden werden grundsätzlich vom **Betriebssystem** erzeugt und auf Prozessoren bzw. Kerne verteilt
 - Schnittstelle dazu meist in Programmiersprachen eingebaut (z.B. Java Threads) oder über Bibliotheken.