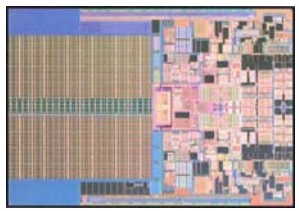
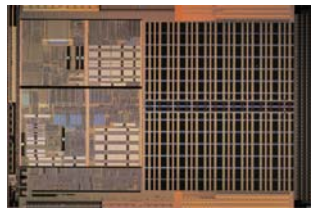
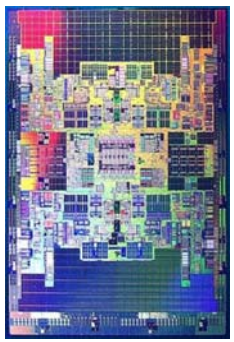
Moderne Mikroprozessoren

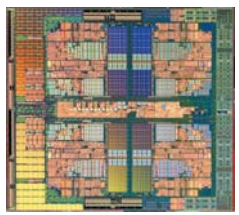
Hagen Gädke
TU Braunschweig
Prof. Dr. Andreas Koch
TU Darmstadt

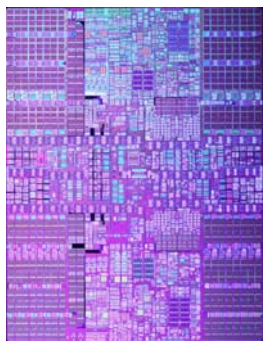
TECHNISCHE UNIVERSITÄT DARMSTADT

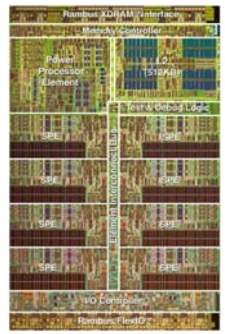












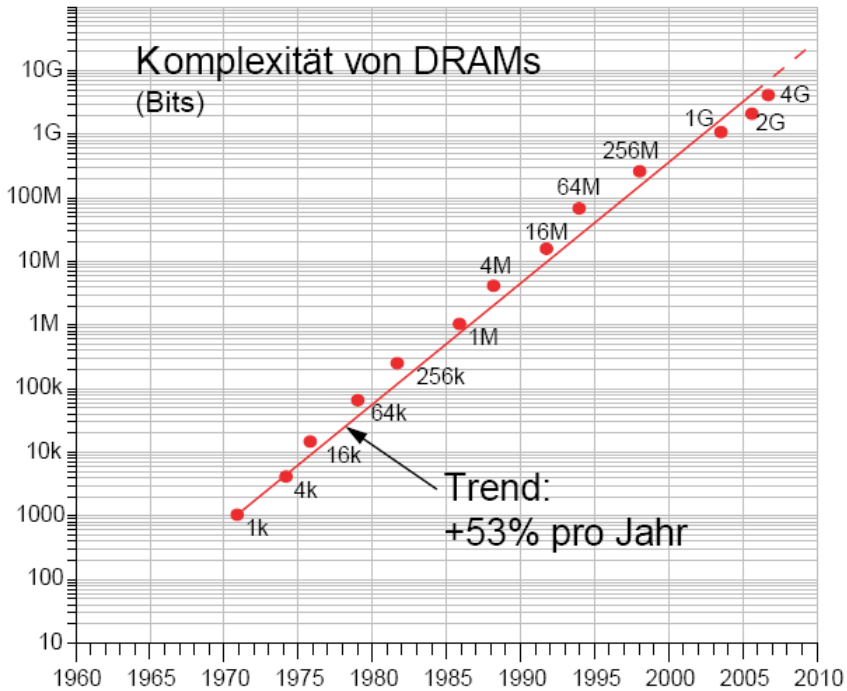
Inhalt

TECHNISCHE UNIVERSITÄT DARMSTADT

- CISC / RISC
  - Unterschiede, warum diese Trennung?
- RISC-Architekturen
  - MIPS, Sparc, POWER
- Techniken zum Beschleunigen einer CPU
- CISC-Architekturen
  - Intel, AMD
- VLIW
- Taktrate, Rechenleistung, Energie (sparen)
- Alternative Rechenarchitekturen
  - Cell, GPU, Adaptive Rechner
- Ausblick

Entstehung CISC

- Bis 1980: Speicher teuer
  - 1970-1980: typische DRAM-Größe: 1-64 kBit
- Programme verkleinern
- Mehrere Befehle ⇨ ein komplexer Befehl
- Complex Instruction Set Computing, CISC



Entstehung CISC

TECHNISCHE UNIVERSITÄT DARMSTADT

- CISC: Complex Instruction Set Computing
  - Programme belegen wenig Speicher
  - Hohe Anzahl möglicher Befehle
  - Starke Variation in Befehlslängen
    - Befehle sind z. B. 1 - 15 Bytes lang (x86)

⇨Komplexere Verarbeitungslogik in der CPU

⇨CPU-Aufbau ist komplexer

CISC vs. RISC

TECHNISCHE UNIVERSITÄT DARMSTADT

- Mitte 80er Jahre: höhere Integrationsdichten
  - Speicher billiger, CISC nicht mehr unbedingt nötig
- Aufspaltung in CISC...
  - Bestehendes Modell, z. B. Intel
- ...und RISC
  - Reduced Instruction Set Computing
  - Zurück zu ursprünglichen, einfachen Befehlen
- CISC:
  - Abwärtskompatibel (zu bestehendem CISC-Code)
- RISC:
  - – Größere Programme
  - – Nicht abwärtskompatibel (zu best. CISC-Code)
  - \+ Weniger Befehle
  - \+ Einfacherere und kürzere Befehle

CISC vs. RISC

TECHNISCHE UNIVERSITÄT DARMSTADT

- RISC: einfacheres CPU-Design
  - Bessere Energieeffizienz
  - Höhere Taktraten
    - (Siehe später)

- Beispiele:
  - ab 1981: MIPS
  - ab 1985: Sun SPARC
  - ab 1990: IBM POWER
  - ab 1992: DEC Alpha

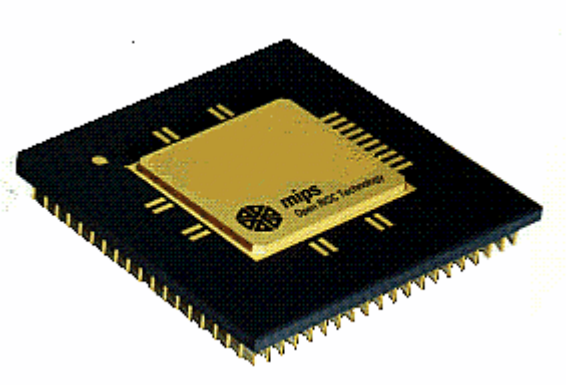
embedded systems + applications

MIPS

TECHNISCHE UNIVERSITÄT DARMSTADT

- Microprocessor without Interlocked Pipeline Stages

MIPS TECHNOLOGIES



- "Keine Pipeline-Sperren"
  - Traditionell:
    - Bei Abhängigkeiten zwischen aufeinanderfolgenden Befehlen:
    - Nachfolgender Befehl wartet, bis aktueller Befehl fertig
      - c = a + b; d = c + 5;
    - CPU erkennt Abhängigkeiten
  - Neu:
    - Compiler fügt NOP-Befehle ein
    - CPU muss weniger tun ⇨ einfacheres Pipelining
  - (Heute: Forwarding statt NOPs)

MIPS

TECHNISCHE UNIVERSITÄT DARMSTADT

- 1981 Stanford (John Hennessy)
- MIPS-Rechner bis 1998
  - SGI-Server und -Workstations
  - Siemens- und DEC-Server
- Letztes MIPS-Modell 2004
  - R16000A, 1000 MHz, 110 nm
- MIPS heute
  - Handel mit geistigem Eigentum („IP Blöcke“)
  - Viele eingebettete Systeme basieren auf MIPS
    - Cisco-Router
    - BMW-Navis
    - Nintendo 64, Sony PlayStation 2
    - Fritz!Box

SGI Onyx 2

embedded systems + applications

Sun SPARC

TECHNISCHE UNIVERSITÄT DARMSTADT

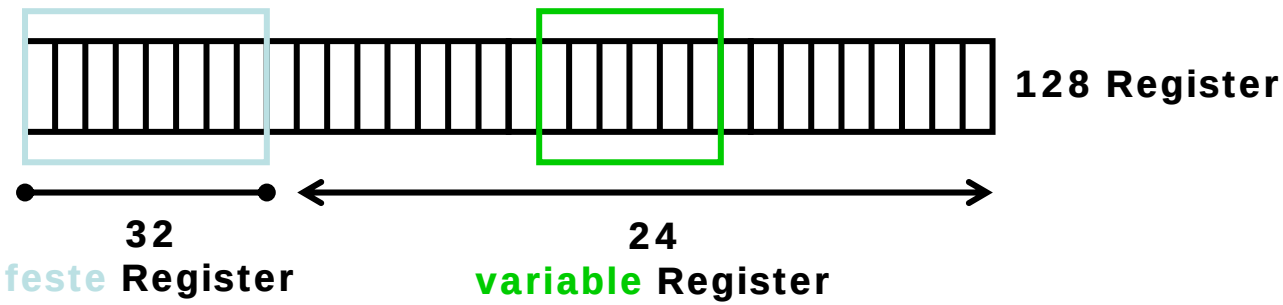
- SPARC = Scalable Processor ARChitecture
  - David Patterson, Berkeley
    - 1980: RISC I
- SPARC-Geschichte
  - 1985: Sun SPARC
  - 1989:
    - Offene Architektur
    - Ziel: Verbreitung
  - 2005: 8 Cores à 4 Threads
    - UltraSPARC T1: 1,2 GHz, 90 nm
  - 2008: 4 Kerne à 2 Threads
    - SPARC64 VII: 2,7 GHz, 65 nm
  - (2009: 16 Cores à 2 Threads)
    - UltraSPARC Rock: 2,3 GHz, 65 nm

UltraSPARC T2 (aktuelles Modell)

embedded systems + applications

Sun SPARC

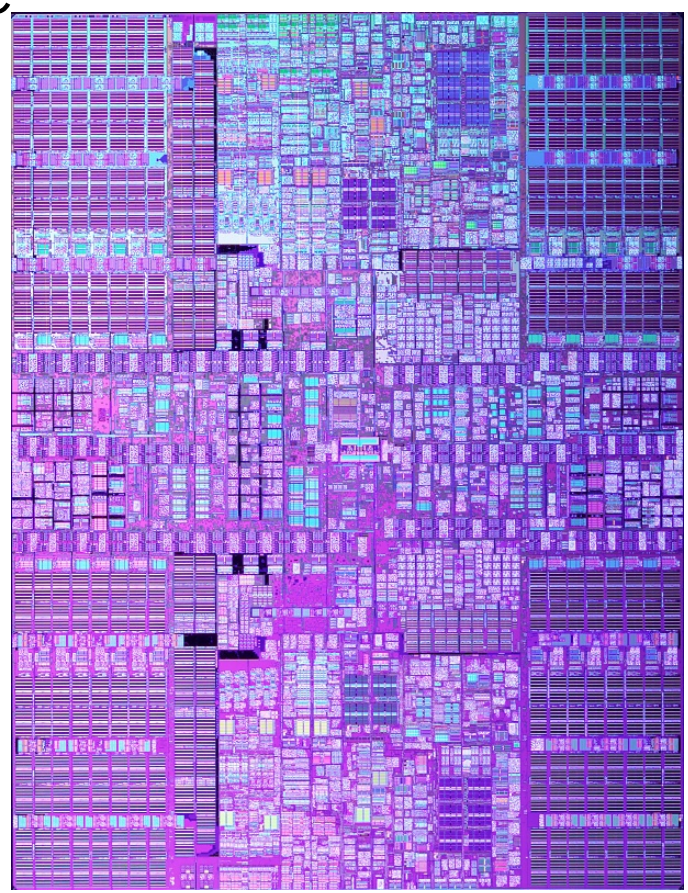
- CPU-Register: 128 x 32 Bit
  - CPU sieht davon stets 32,
  - Sowie 24 weitere
    - Per Software verschiebbares Fenster
    - Vorteil: Fenster-Verschieben effizienter als Nachladen aus Hauptspeicher



  - Register: Vergleich mit Intel 386er (1985)
    - Nur 16 Register,
    - Davon 4x 32-Bit, 4x 16-Bit, 8x 8-Bit

IBM POWER

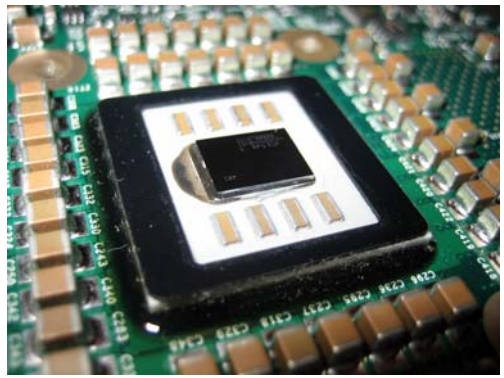
- Performance Optimized With Enhanced RISC
- Entwicklung
  - Seit 1990
  - 2001: Dual-Core mit gemeinsamem L2-Cache
    - Erster Dual-Core-Chip
  - 2004: Multi-Threading, integrierter Mem-Controller
  - 2005: Quad-Core
- Aktuell: POWER6
  - 4,7 GHz, 65 nm
    - Cache: 4 MB L2, 32 MB L3
  - 2 Cores, 2 Threads



IBM POWER

TECHNISCHE UNIVERSITÄT DARMSTADT

- 1993: PowerPC
  - AIM: Apple, IBM, Motorola
  - 64 Bit von Anfang an
  - POWER-Architektur mit geringen Änderungen
    - Einige Multiplikations- und Divisions-Befehle ersetzt
    - Support für symmetrisches Multi-Processing (SMP)
- Modellpflege
  - PowerPC xxx(x) ⇨ Einteilung von Apple in "Generationen"
    - G1: 0,6µ, 50 - 120 MHz, 1993
    - G2: 0,5-0,25µ, 66 - 375 MHz
    - G3: 0,25-0,13µ, 200 - 1000 MHz 1997 - 1999
    - G4: 0,2-0,13µ, 350 - 1700 MHz 1999 - 2004
    - G5: 90 nm 1,4 - 3 GHz, ab 2003

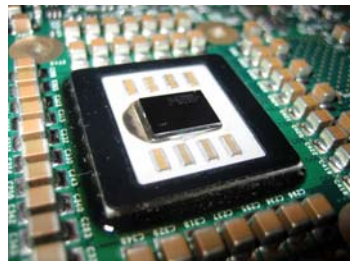


PowerPC 970FX

embedded systems + applications

IBM POWER - Einsatzgebiete

- IBM-Server
  - BlueGene: PowerPC 440
- Apple Macintosh
  - 1993: PowerPC (AIM)
  - 2005: Wechsel PowerPC ⇨ x86
- Spielkonsolen
  - Nintendo GameCube, Wii
  - Microsoft XBox 360
  - Sony Playstation 3 (Cell-CPU)
- Eingebettete Systeme
  - Z. B. Virtex-II Pro: PowerPC 405
  - PKW
  - Luft- und Raumfahrt



PowerPC 970FX

Inhalt

TECHNISCHE UNIVERSITÄT DARMSTADT

- CISC / RISC
  - Unterschiede, warum diese Trennung?
- RISC-Architekturen
  - MIPS, Sparc, POWER
- Techniken zum Beschleunigen einer CPU
- CISC-Architekturen
  - Intel, AMD
- VLIW
- Taktrate, Rechenleistung, Energie (sparen)
- Alternative Rechenarchitekturen
  - Cell, Adaptive Rechner
- Ausblick

CPU beschleunigen

TECHNISCHE UNIVERSITÄT DARMSTADT

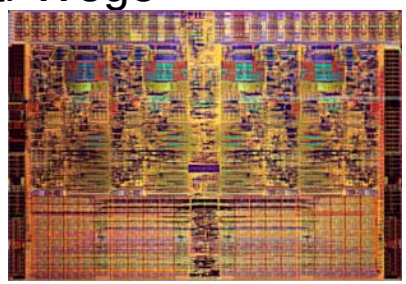
- Gilt für CISC als auch für RISC:
  - Taktrate erhöhen
  - Pipelining
  - Spekulative Ausführung
  - Cache
  - Parallele Recheneinheiten
  - Integration externer Komponenten
  - SIMD
  - Multi-Threading
  - Multi-Core

Taktrate erhöhen

TECHNISCHE UNIVERSITÄT DARMSTADT

- Höhere Taktrate ⇨ schnelleres Rechnen
- Kleinere Strukturen
  - Geringere Transistor-Kapazitäten
  - Niedrigere Versorgungsspannung
  - Kürzere Transistor-Schaltzeiten und Signal-Wege

1978: 3000 nm (Intel 8086)



2009: 45 nm (Intel Core i7 Nehalem)

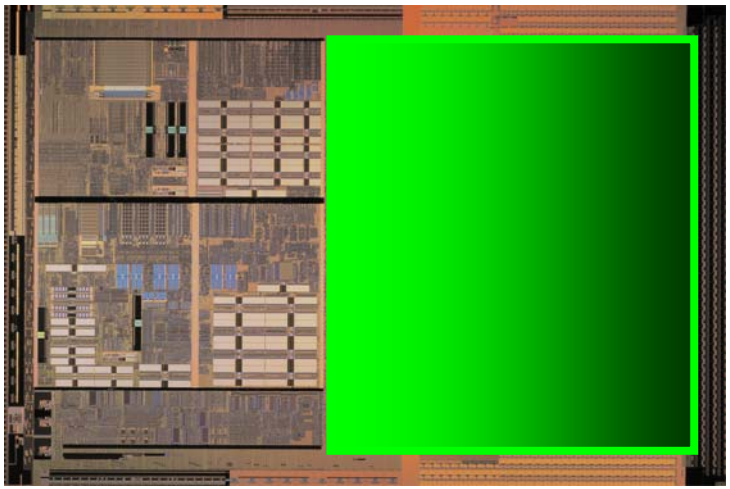
67 mal kleinere Strukturen

- Stand Taktraten:
  - 1978: 5 MHz (Intel 8086)
  - Heute: 4,7 GHz (IBM POWER6)
    - Intel Core i7 Nehalem: 3,2 GHz

940 mal mehr

embedded systems + applications

Cache

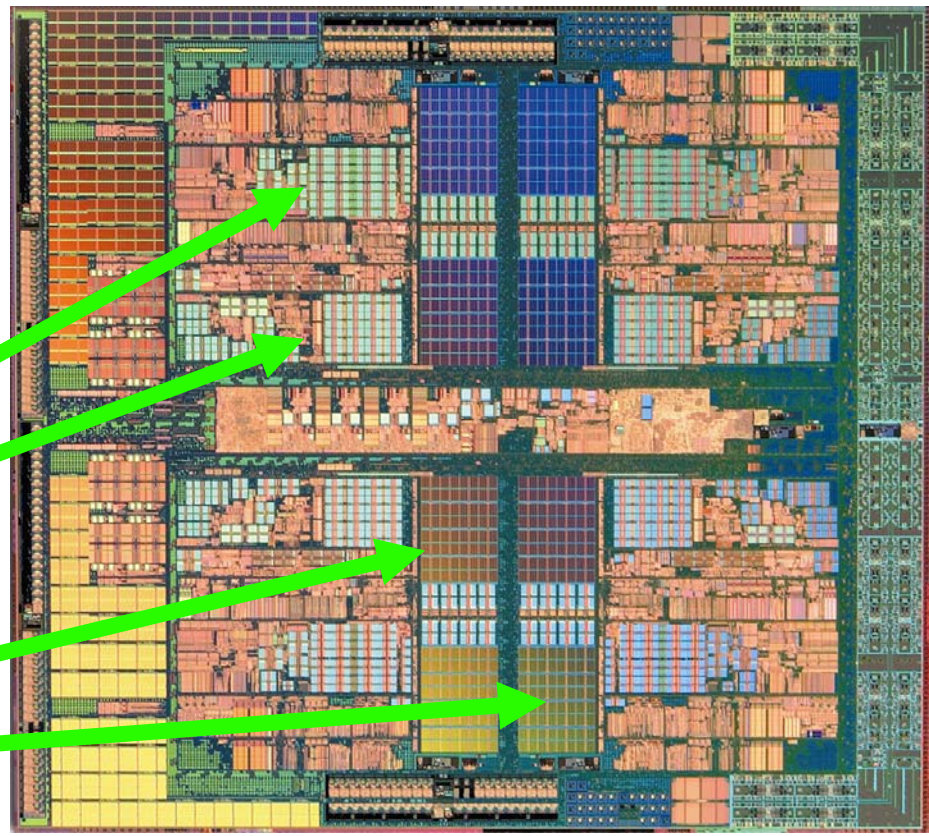


L2-Cache des AMD Athlon64

- Zwischenstufe zum Hauptspeicher MEM
- Schneller, weil:
  - Kleiner als MEM
    - Weniger Adressleitungen müssen ausgewertet werden
  - Dichter an CPU, heute sogar auf gleichem Die
    - Höhere Taktung und
    - Höhere Bitbreite der Anbindung
  - SRAM schneller ansprechbar als DRAM

Cache

- Nur kleiner, schneller Cache:
  - Zu schnell voll
- Nur großer Cache:
  - Zu langsam
- Cache-Hierarchie
  - L1-, L2-, L3-Cache
- Bsp.:
  - AMD Phenom, Quad-Core
  - L1: 64 KB Daten + 64 KB Instr.
    - Pro Kern
  - L2: 512 KB
    - Pro Kern
  - L3: 2 MB



Cache

TECHNISCHE UNIVERSITÄT DARMSTADT

- Unzählige Cache-Parameter, z. B.:
  - Größe, Latenz
  - Assoziativität :Mapping: Bereiche des RAM ⇨ Bereiche des Caches
  - Write-through: Nach Schreiben sofort an nächste Ebene weiterreichen
  - Write-back: Erst weiterreichen, wenn neue Cache-Line benötigt wird
  - Multi-Core: gemeinsamer Cache (z. B. Intel CoreDuo)
    - Schnellerer Datenaustausch zwischen Cores
    - Cores können sich Cache nach Bedarf aufteilen
  - ...oder getrennt (z. B. L2 bei AMD Athlon64 X2)
    - Einfacher skalierbar
    - Kohärenz
  - Smart-Cache
    - Lade schon vorab Daten aus dem Hauptspeicher

Parallele Recheneinheiten

- Mehr Decoder
- Mehr Execution Units
  - Standard-Recheneinheiten
    - ALUs, AGUs
  - Load/Store-Einheiten
- Grad an Superskalarität
  - Anzahl paralleler Rechenpfade

Ausschnitt aus Architektur des AMD Athlon64

3-fach skalar

Integration externer Komponenten

TECHNISCHE UNIVERSITÄT DARMSTADT

- Mehr externe Komponenten in den Chip hinein
  - Kürzere Latenzen
  - Höhere Taktraten / weniger Takte

- Geschichte (nicht vollständig):
  - 1971 Intel 4004 ← erste CPU in 1 Chip
  - 1989 Intel 80486 ← FPU, L1-Cache
  - 1995 Intel PentiumPro ← L2-Cache
  - 2000 Transmeta Crusoe ← Northbridge
  - 2001 IBM POWER5 ← Dual-Core
  - 2002 Intel Itanium 2 ← L3-Cache
  - 2003 AMD Athlon 64 ← DDR-Controller
  - ... Intel / AMD / ... ← mehr Kerne
  - 2009 Intel Westmere ← VGA-Controller

SIMD: Single Instruction Multiple Data

TECHNISCHE UNIVERSITÄT DARMSTADT

- Wende eine Operation auf viele Daten parallel an
  - Z. B. Berechnungen auf Arrays, Matrizen
- Zusätzliche Befehle / Register / Recheneinheiten
- Beispiele
  - MMX (Intel)
    - Matrix Math Extensions ⇨ Multi Media Extensions
  - 3DNow! (AMD)
  - ISSE (Intel)
    - Internet Streaming SIMD Extensions ⇨ SSE
  - AltiVec (PowerPC)
- Neuerungen:
  - Mehr Operationen parallel
  - Größere Bitbreiten
    - 3DNow!: 64 Bit; SSE: 128 Bit; AVX: 256 Bit

embedded systems + applications

Multi-Threading

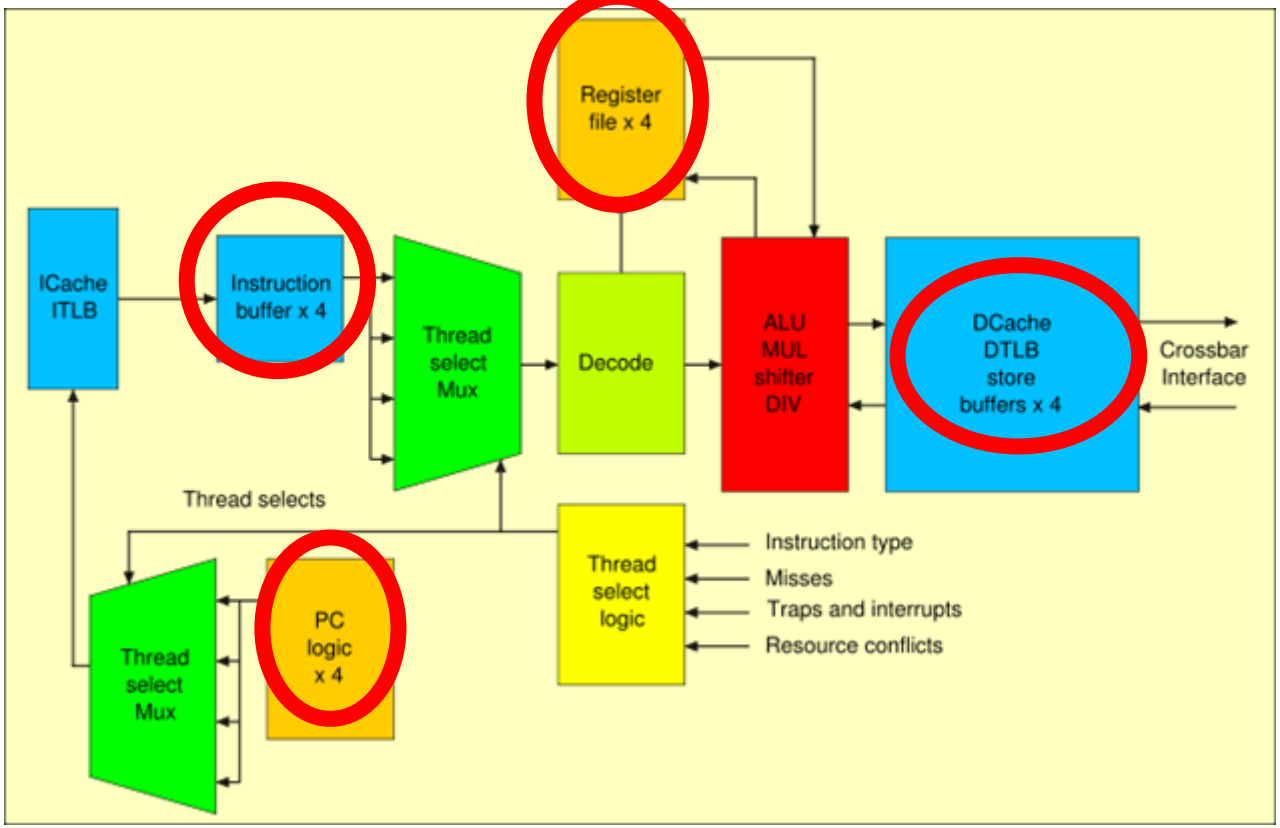
TECHNISCHE UNIVERSITÄT DARMSTADT

- Ohne Multi-Threading:
  Ungenutzte Ressourcen in der Pipeline
  - Z. B. Warten auf Speicherzugriff

- Mit Multi-Threading:
  Unabhängiger Thread nutzt Ressourcen
  - Einige CPU-Kompenten vervielfachen
    - Exklusiv für jeden Thread
    - Z. B. Datenpuffer, Register-File
  - Größere CPU-Komponenten gemeinsam nutzen
    - Z. B. Cache

Multi-Threading

- Beispiel UltraSPARC T1, 4 Threads

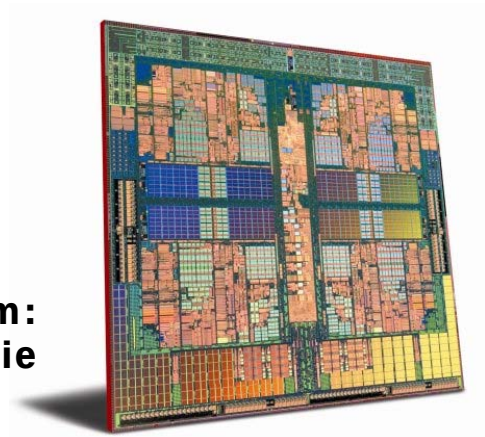


- Pentium 4, i7, Atom: Hyper-Threading (2 Threads)

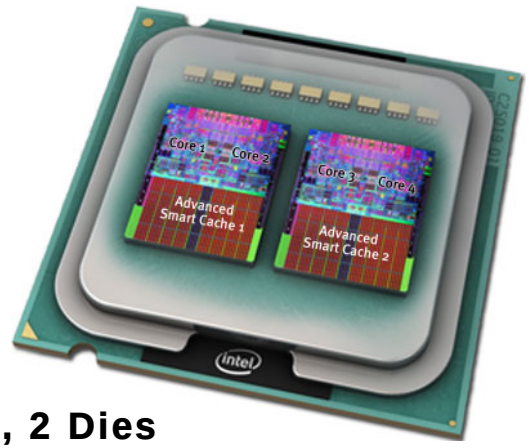
Multi-Core

TECHNISCHE UNIVERSITÄT DARMSTADT

- Mehrere Rechenkerne (Cores) in 1 Chip
  - Ggf. sogar auf 1 Die: monolithisch
- Dual-, Quad-Core
  - Core2 Quad
- Performance
  - Gut parallelisierbare Probleme
  - Multi-Tasking
- Beispiele
  - IBM POWER: Dual-Core (Cell: 9)
  - Sun SPARC: 2, 4, 8, (16) Cores
  - Intel: 2, 4, 6 Cores
    - 4: Penryn, Nehalem
    - 6: Penryn (Xeon Dunnington)
  - AMD: 2, 3, 4, 6 Cores



AMD Phenom: 4 Cores, 1 Die



Intel Penryn: 4 Cores, 2 Dies (Nehalem: 4 Cores, 1 Die)

embedded systems + applications

Inhalt

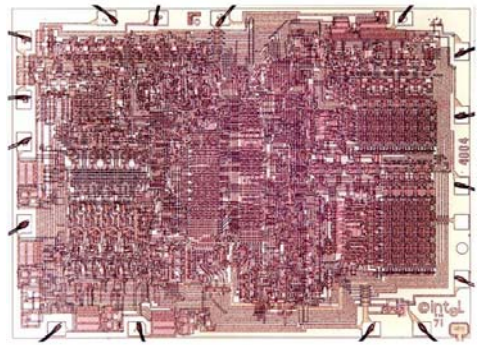
TECHNISCHE UNIVERSITÄT DARMSTADT

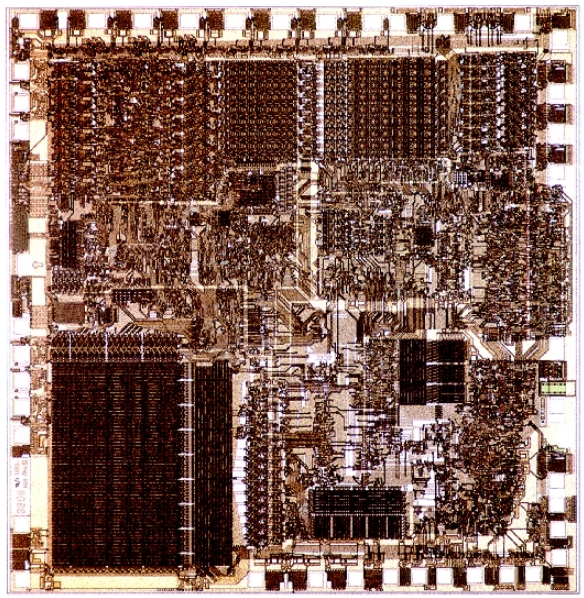
- CISC / RISC
  - Unterschiede, warum diese Trennung?
- RISC-Architekturen
  - MIPS, Sparc, POWER
- Techniken zum Beschleunigen einer CPU
- CISC-Architekturen
  - Intel, AMD
- VLIW
- Taktrate, Rechenleistung, Energie (sparen)
- Alternative Rechenarchitekturen
  - Cell, Adaptive Rechner
- Ausblick

CISC: Intel

- 1971: Intel 4004
  - Erste in Serie produzierte Ein-Chip-CPU
  - 4 Bit
- 1978: Intel 8086
  - 16 Bit
  - *x86-Architektur*
  - Binaries von 1978 auf aktuellen x86-CPUs ausführbar!
  - Heute 32 Bit: *IA-32* (Intel-Architektur, 32 Bit)

Intel 4004





Intel 8086

Nachteile IA-32-CISC vs. RISC

TECHNISCHE UNIVERSITÄT DARMSTADT

- Befehle: stark unterschiedliche Länge
  - 1-15 Bytes
  - Decodierung komplexer
- Meist Speicheradressen als Operanden
  - Mehr Speicherzugriffe
    - RISC: Speicherzugriffe in dedizierten Load/Store-Befehlen
- Weniger Register
  - IA-32: 8 General-Purpose-Integer-Register
    - RISC: typisch 32 - 128 Register
  - Register schnell aufgebraucht
  - Zwischenergebnisse: Speicher
  - Mehr Speicherzugriffe

Intel: Trotzdem CISC

TECHNISCHE UNIVERSITÄT DARMSTADT

- Intel steigt trotz vieler Nachteile nicht auf RISC um
  - Andrew Tanenbaum (Computerarchitektur), 3 Gründe:

Abwärtskompatibilität

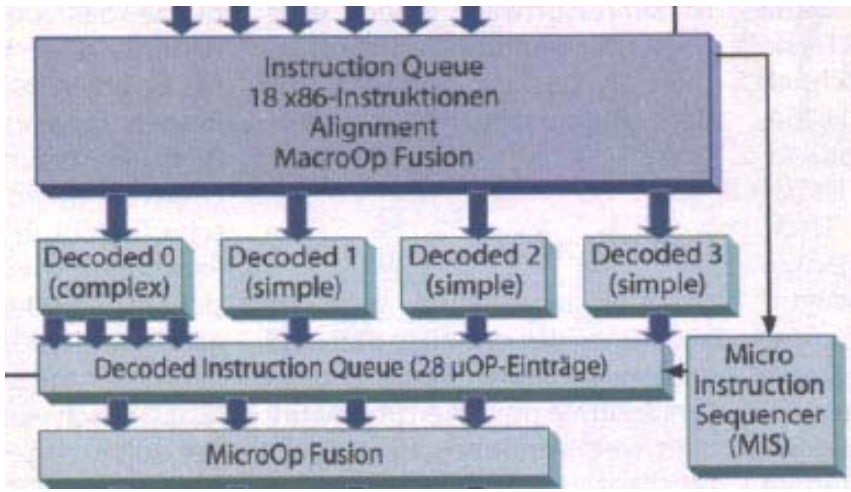
Abwärtskompatibilität

Abwärtskompatibilität

- Stattdessen immer wieder neue Tricks
  - Um doch konkurrenzfähig zu RISC zu bleiben

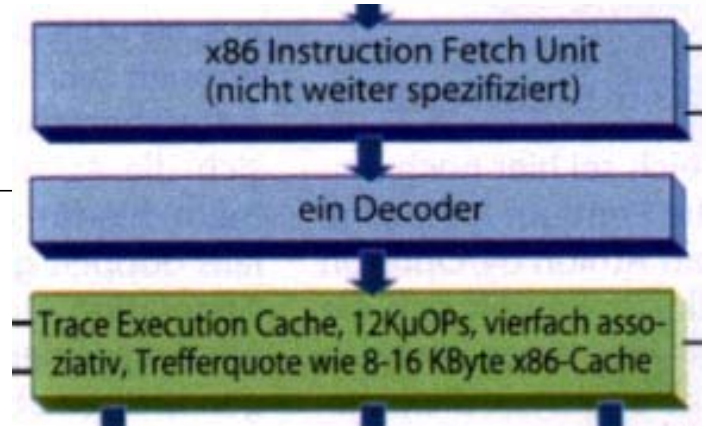
Intel: RISC-Kern

- Intel: seit 1995, Pentium Pro
- Zerlege komplexe x86-Befehle zur Laufzeit in µOPs
- µOP = einfacher Befehl / RISC-Operation
- 1 x86-OP = 1 - ca. 100 µOPs

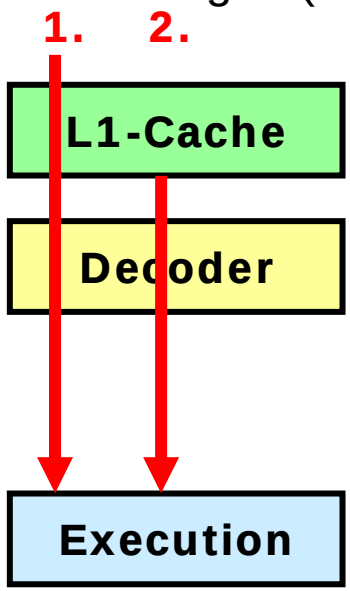


Ausschnitt aus Intels Nehalem-Architektur

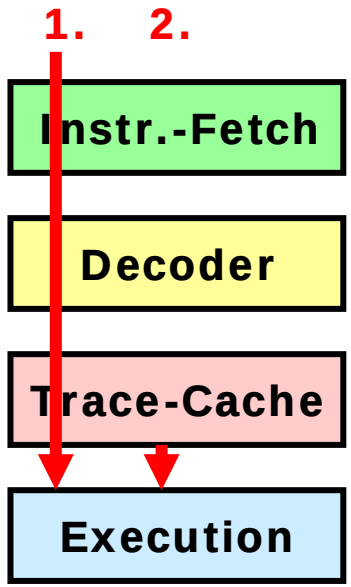
Intel: Trace-Cache



- Intel Pentium 4
  - 2000 - 2006
  - Netburst-Architektur
- Cacht µOPs (anstatt x86-OPs)
  - Spart erneute Umwandlungen (Schleifen!)



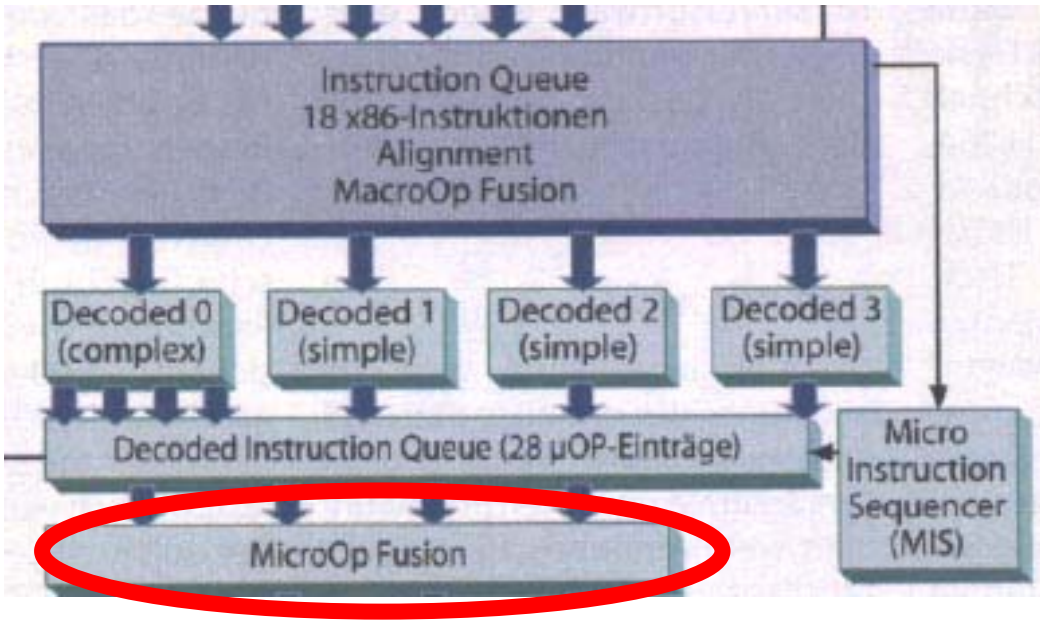
Ohne Trace-Cache



Mit Trace-Cache

Intel: µOP-Fusion

- Seit 2003, Pentium M



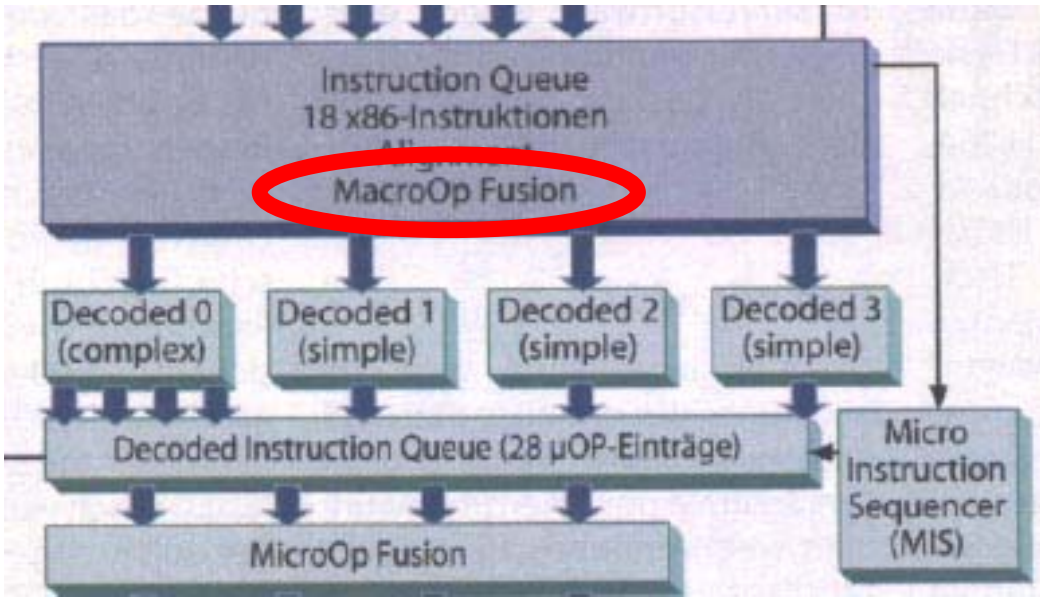
Ausschnitt aus Intels Nehalem-Architektur

- Fasse µOPs zu größeren Operationen zusammenfassen
  - Befehle werden gemeinsam durch Pipeline geschleust
  - Analogie Taxi

Intel: Macro-OP-Fusion

TECHNISCHE UNIVERSITÄT DARMSTADT

- Seit 2006, Core-Architektur



Ausschnitt aus Intels Nehalem-Architektur

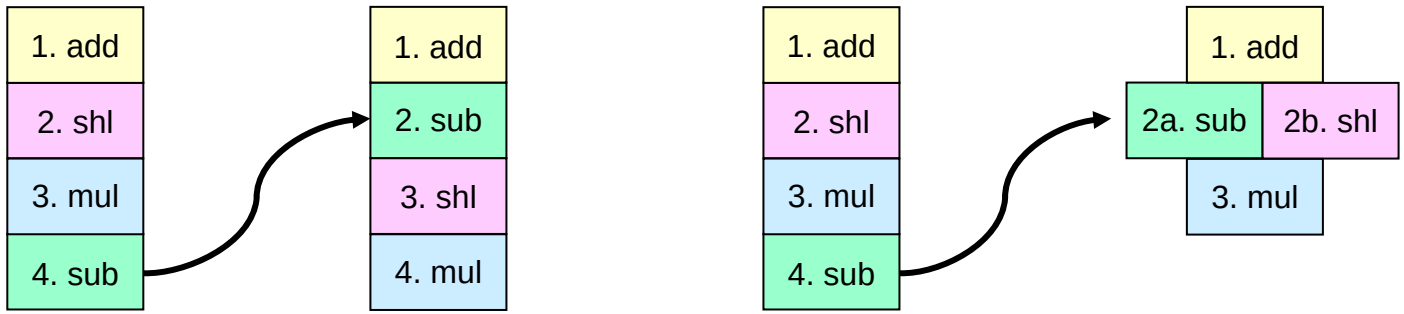
- Schon vor Zerlegung in µOPs:
  - Bestimmte Kombinationen von x86-OPs zu einer x86-OP zusammenfassen
    - Add + Mul ⇨ MAC (Multiply-ACcumulate-Befehl)

embedded systems + applications

Intel: Out-of-Order-Execution

TECHNISCHE UNIVERSITÄT DARMSTADT

- CPU ändert Befehlsreihenfolge zur Laufzeit
- Unabhängige Befehle parallelisierbar
  - Bessere Auslastung der parallelen Recheneinheiten
  - Wartezeiten sinnvoll nutzen



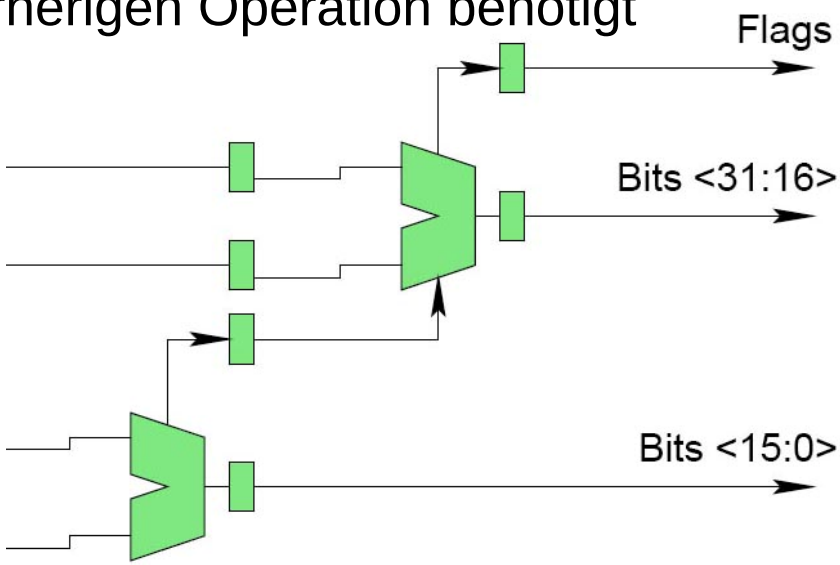
Änderung Reihenfolge

Änderung Reihenfolge + Parallelisierung (parallele Recheneinheiten)

embedded systems + applications

Intel: Double-Pumped ALUs

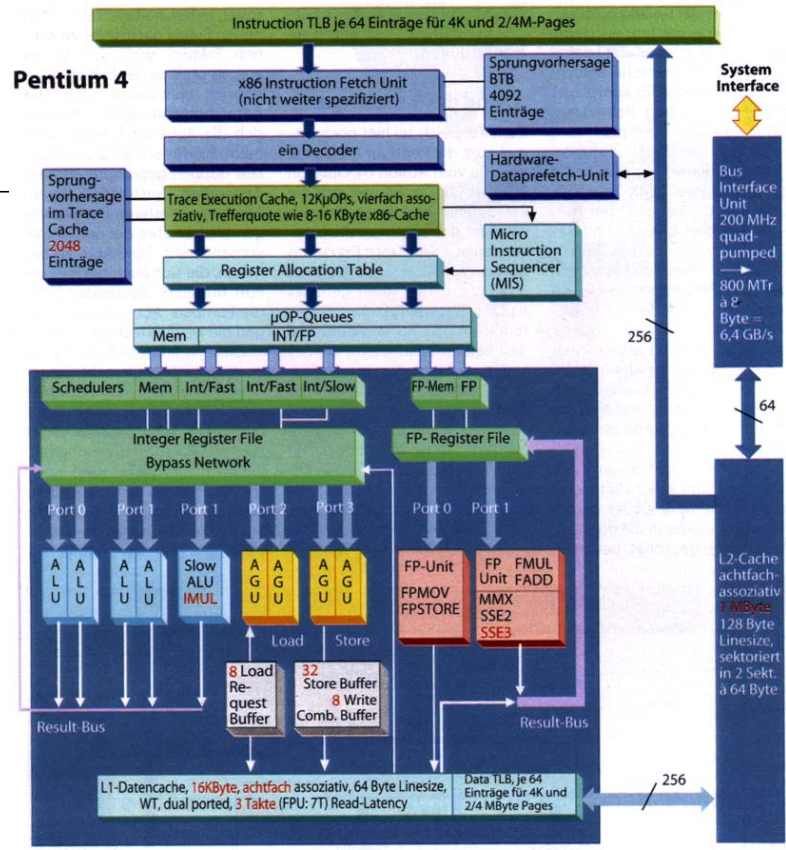
- Pentium 4 (Netburst, bis 2006)
- Trick: Sub-Pipeline in Add-/Sub-ALU
  - Verknüpfe im ersten Halb-Takt untere 16 Bits
  - Im zweiten Halbtakt obere 16 Bits
  - Doppelte ALU-Taktrate möglich (6,4 statt 3,2 GHz)
- Geht nur, wenn Folge-Operation nicht Ergebnis der vorherigen Operation benötigt



Halb-Takt	Operationen
1	ADD1[15:0]
2	ADD1[31:16] ADD2[15:0]
3	ADD2[31:16] ADD3[15:0]
4	ADD3[31:16]
...	

Intel: Längere Pipeline

- Problem:
  - Hohe Zeitverluste
    - Komplexe Decodierung
    - Umwandeln in µOPs
    - µOP-Fusion
    - Macro-OP-Fusion
    - Out-of-Order-Execution
    - Register Renaming
- Lösung: längere Pipeline
  - Pentium 4: bis zu 31 Stufen
- Problem:
  - Längere Pipeline ⇨ größerer Schaden bei Pipeline-Stall
  - Stalls dürfen nur sehr selten passieren
  - Sehr ausgefeilte Sprungvorhersage-Techniken nötig!
    - Verbraucht wieder Chipfläche und Energie



Netburst-Architktur, Pentium 4

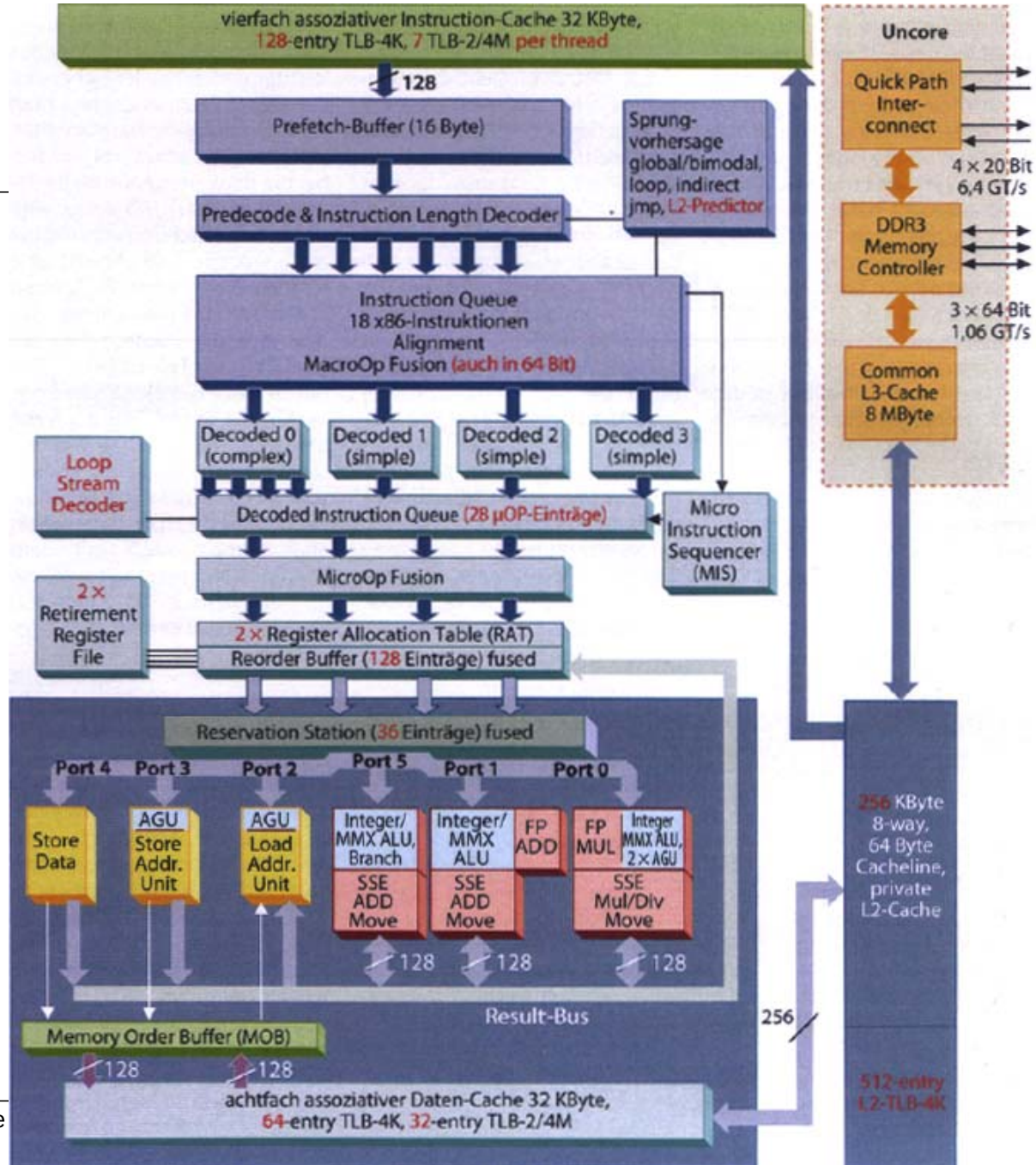
Zu komplex ist auch nicht gut

TECHNISCHE UNIVERSITÄT DARMSTADT

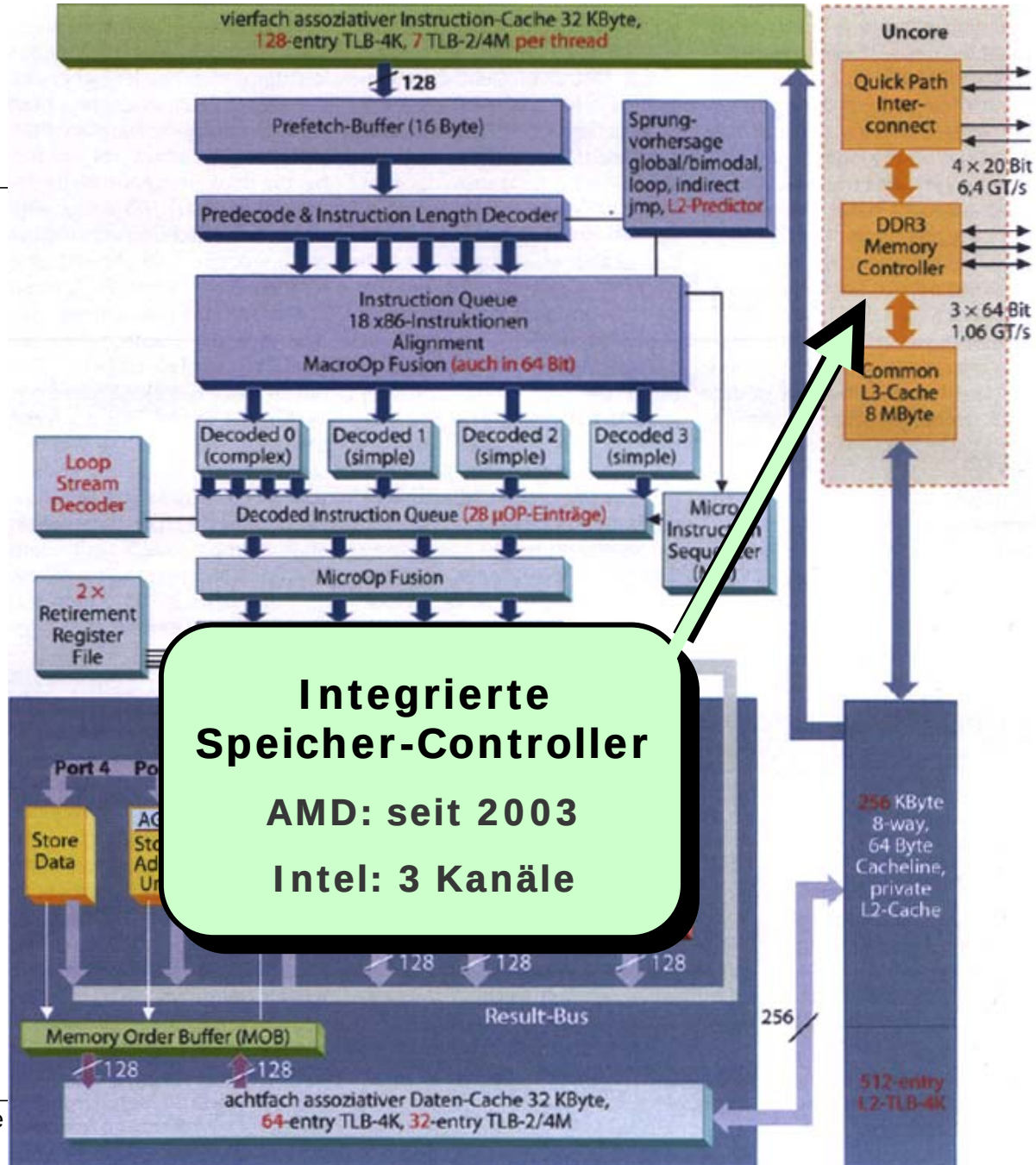
- Problem Netburst-Architektur:
  - Sehr hoher Energieverbrauch (130 Watt)
  - 2006: Intel verwirft Netburst-Architektur
- Nachfolger: *Core*-Architektur (2006)
  - Basiert auf alter P6 (Pentium III / M)-Architektur
  - Kürzere Pipeline
    - 14 statt 31 Stufen
  - Weniger komplexes Innenleben
    - Z. B. kein Trace-Cache mehr
  - Energiespar-Techniken (Pentium M)
  - Multi-Core

Intel 2009: Nehalem-Architektur

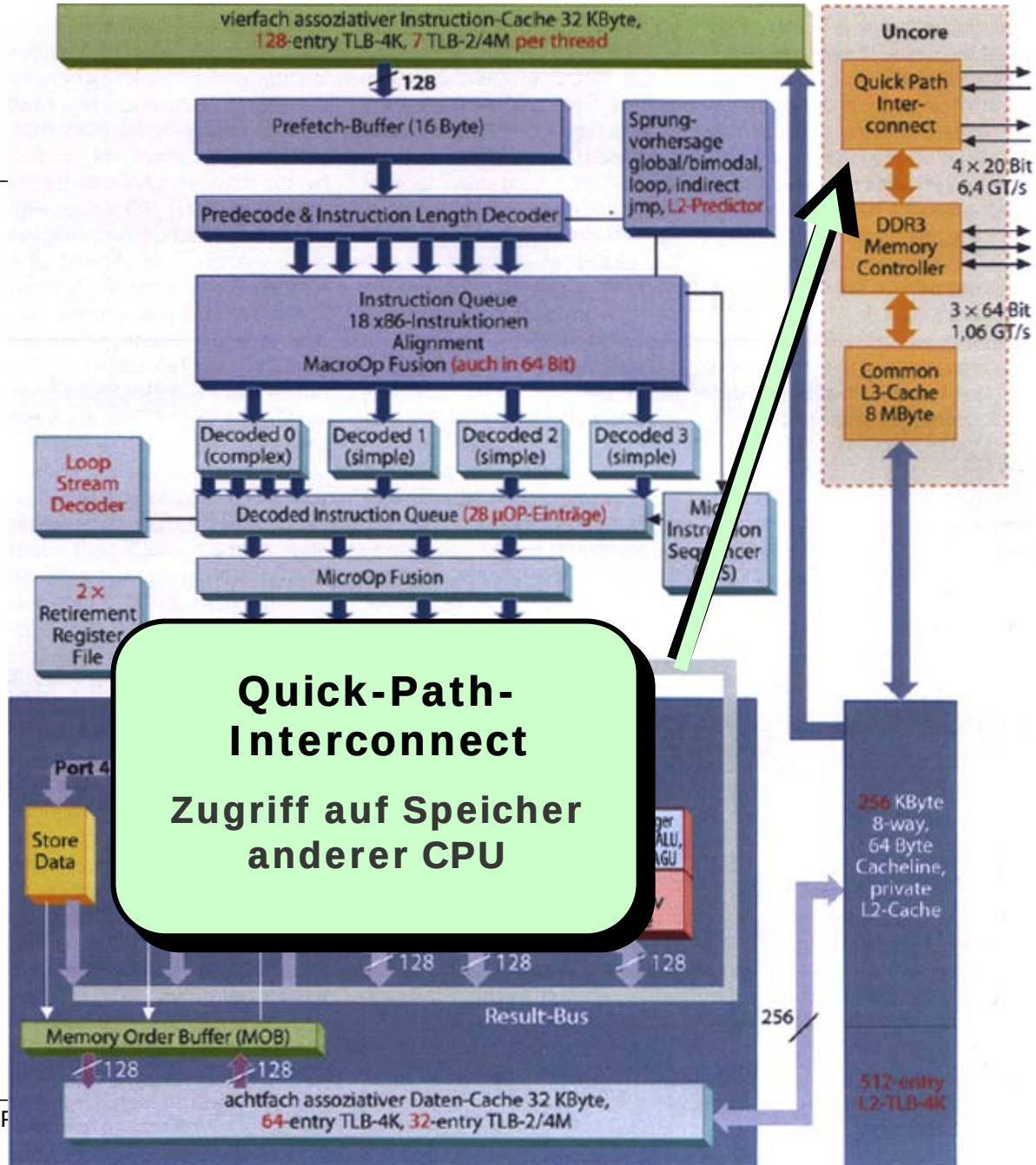
Basiert auf Core-Architektur



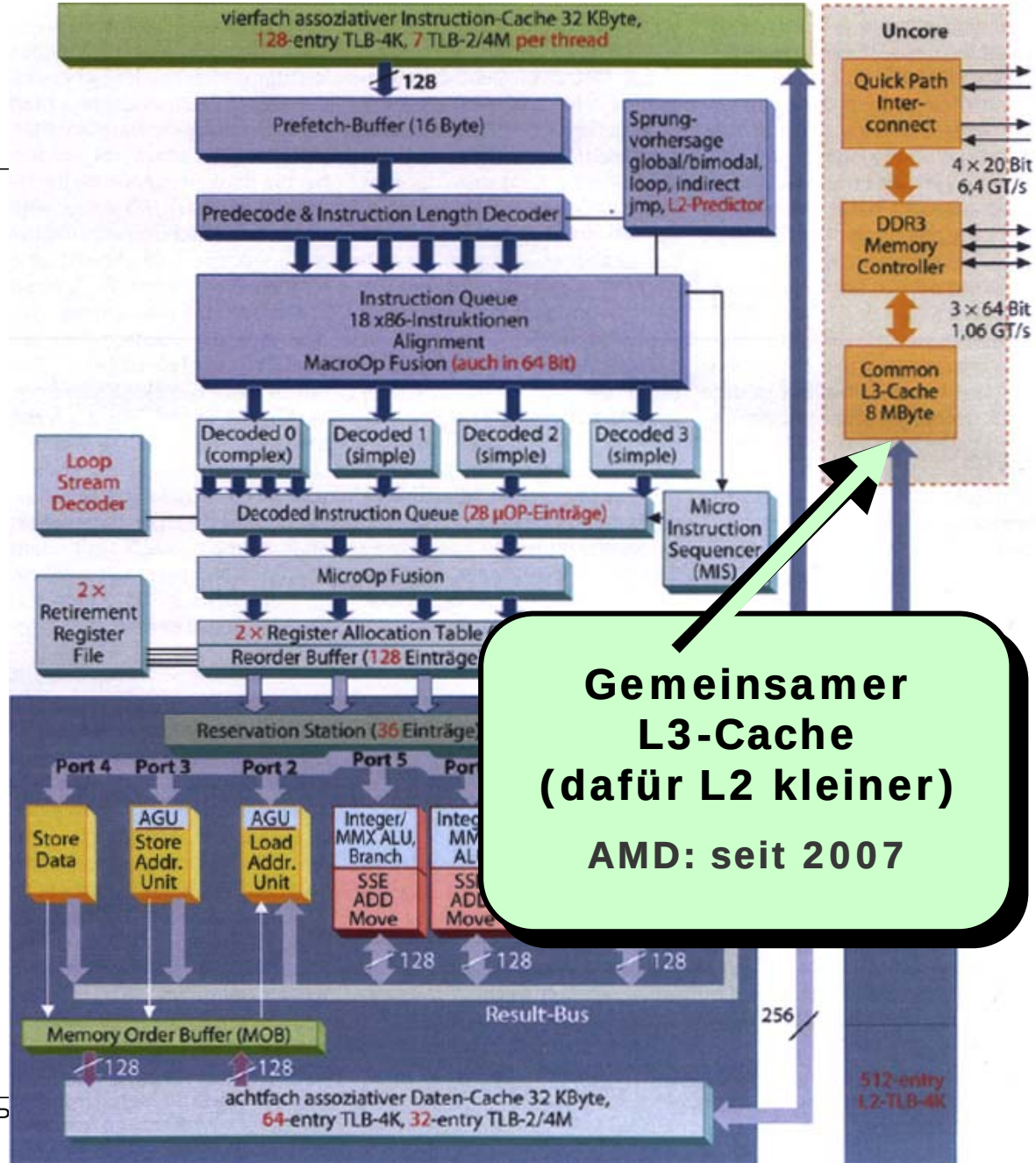
Intel 2009: Nehalem-Architektur



Intel 2009: Nehalem-Architektur



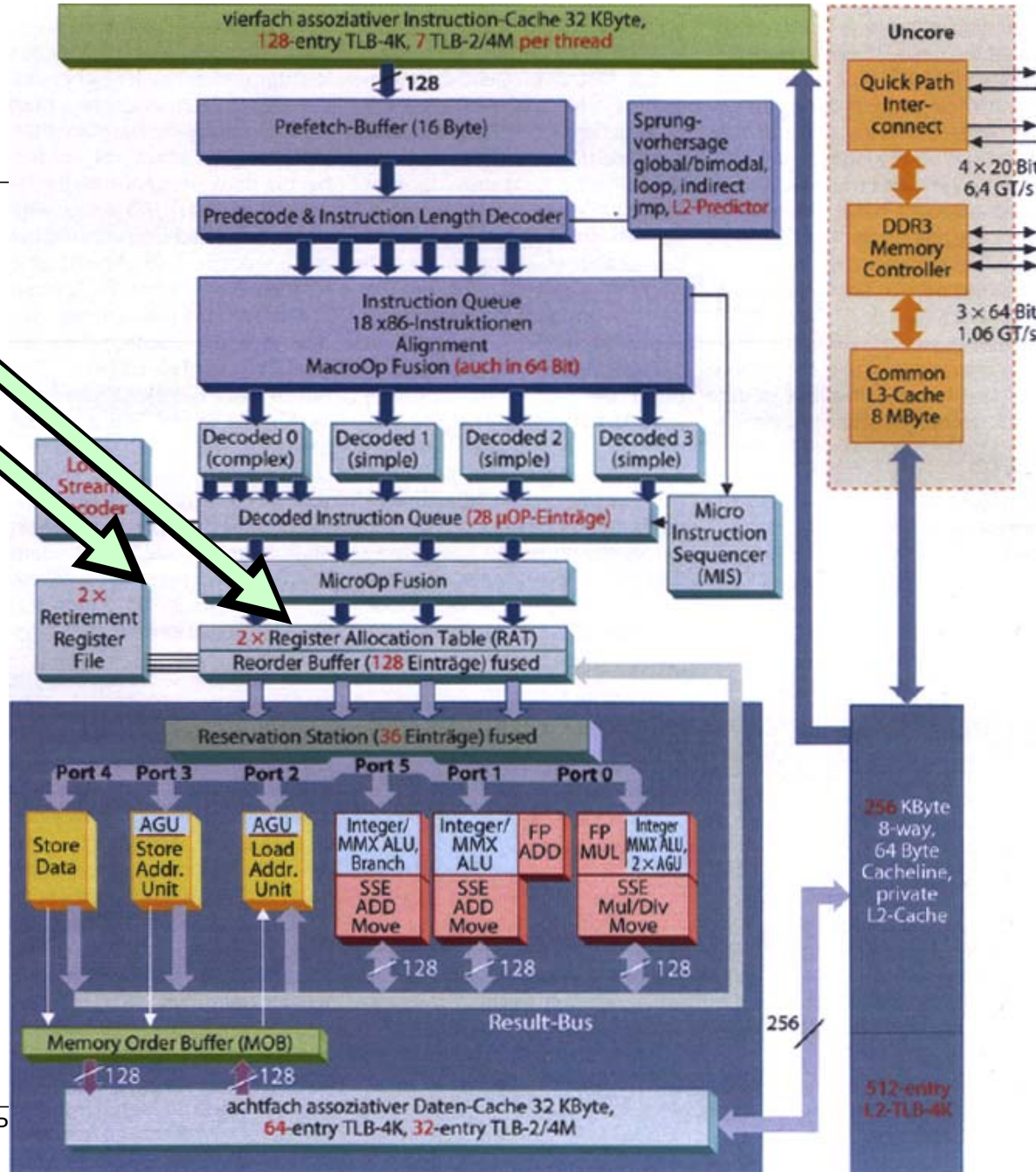
Intel 2009: Nehalem-Architektur



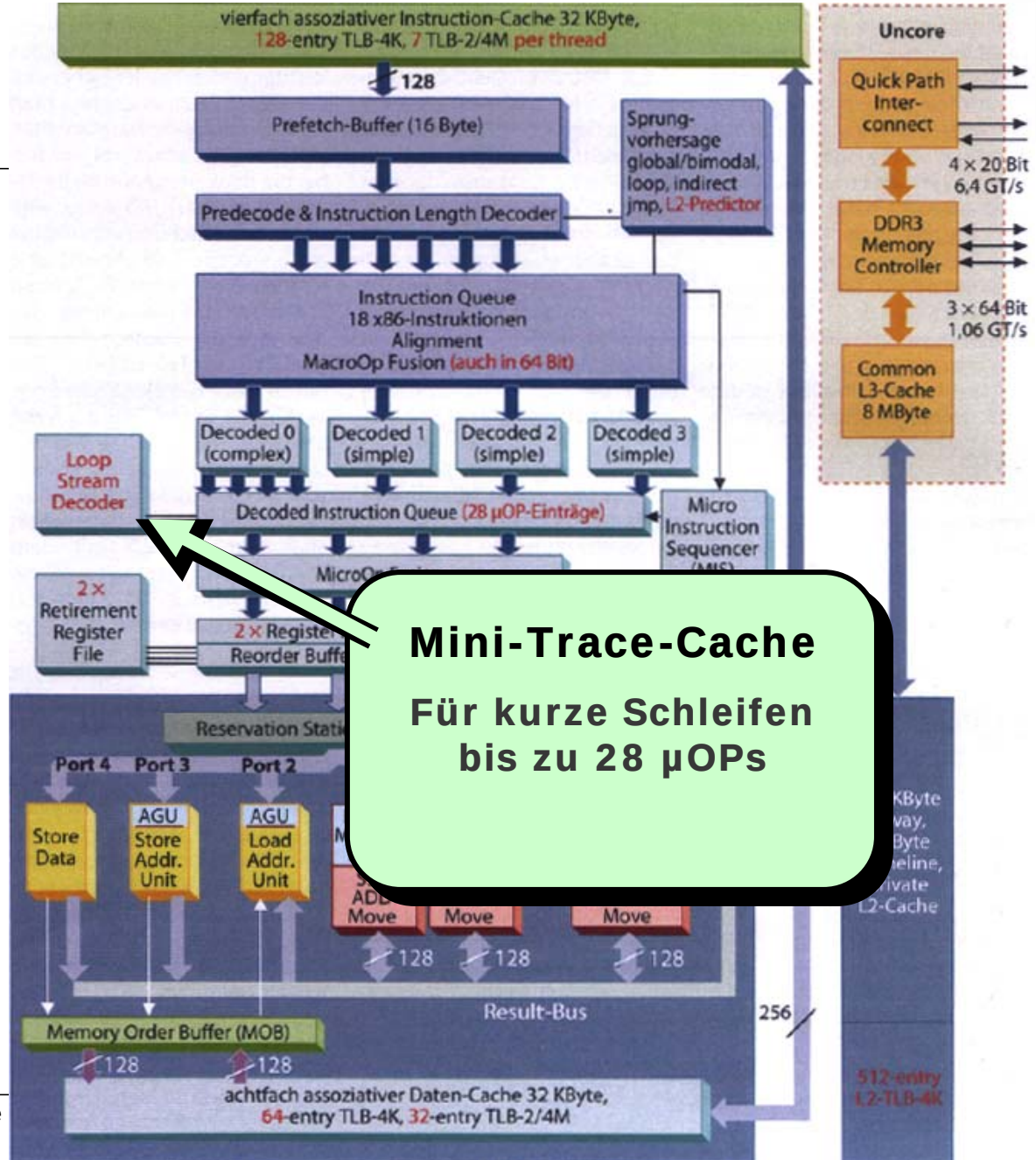
Intel 2009: Nehalem-Architektur

Hyper-Threading

Pipeline: +2 Stufen



Intel 2009: Nehalem-Architektur



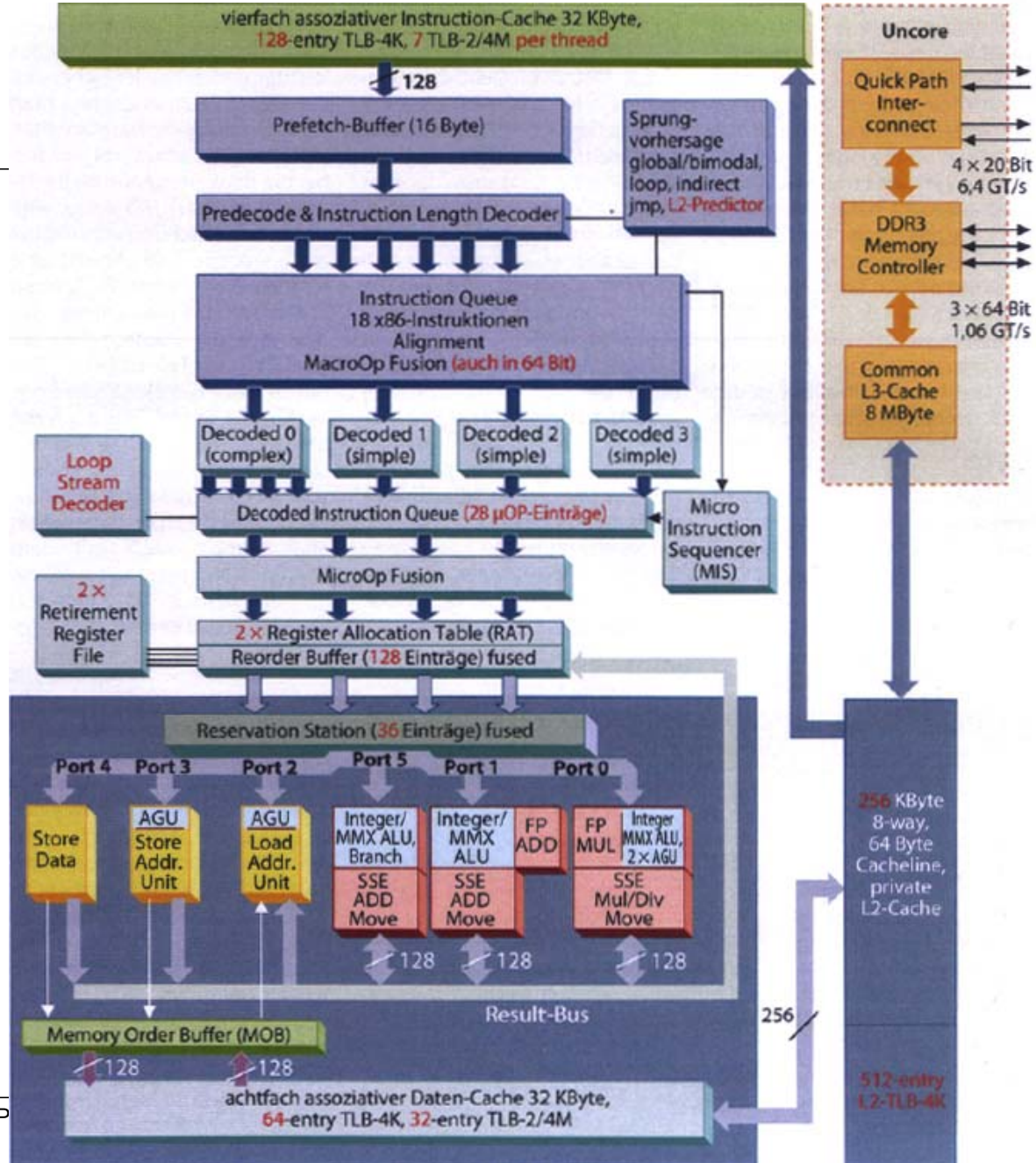
Intel 2009: Nehalem-Architektur

Store-Forwarding

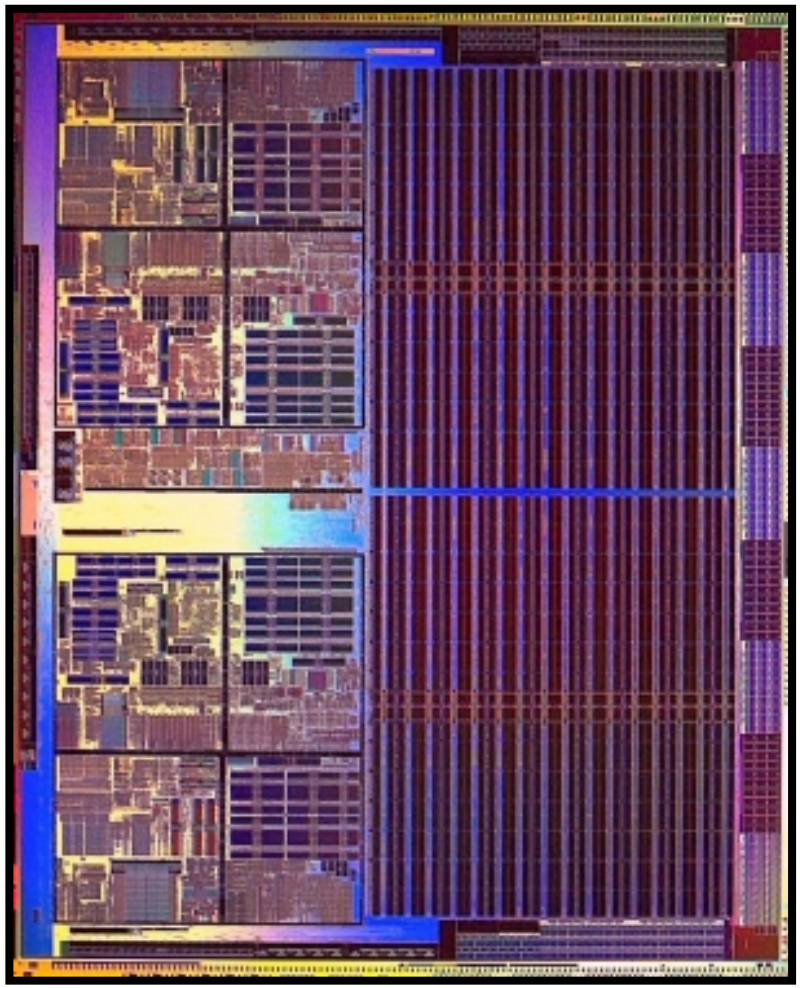
SSE4.2 (Strings)

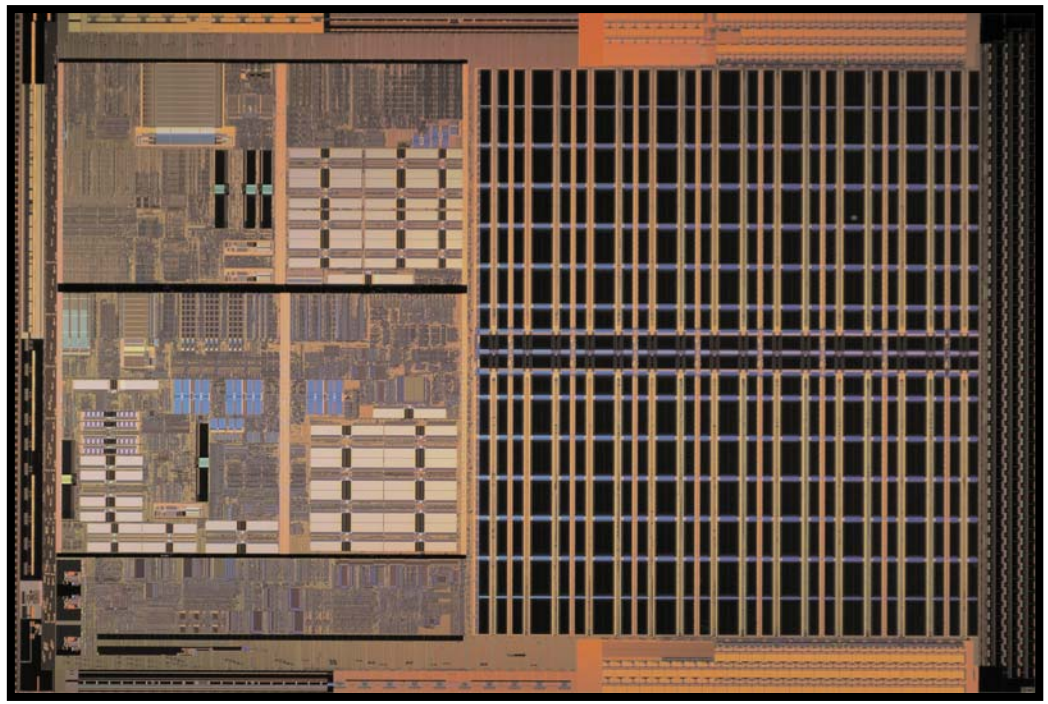
Turbo Boost

Taktrate dynamisch änderbar



CISC: Advanced Micro Devices





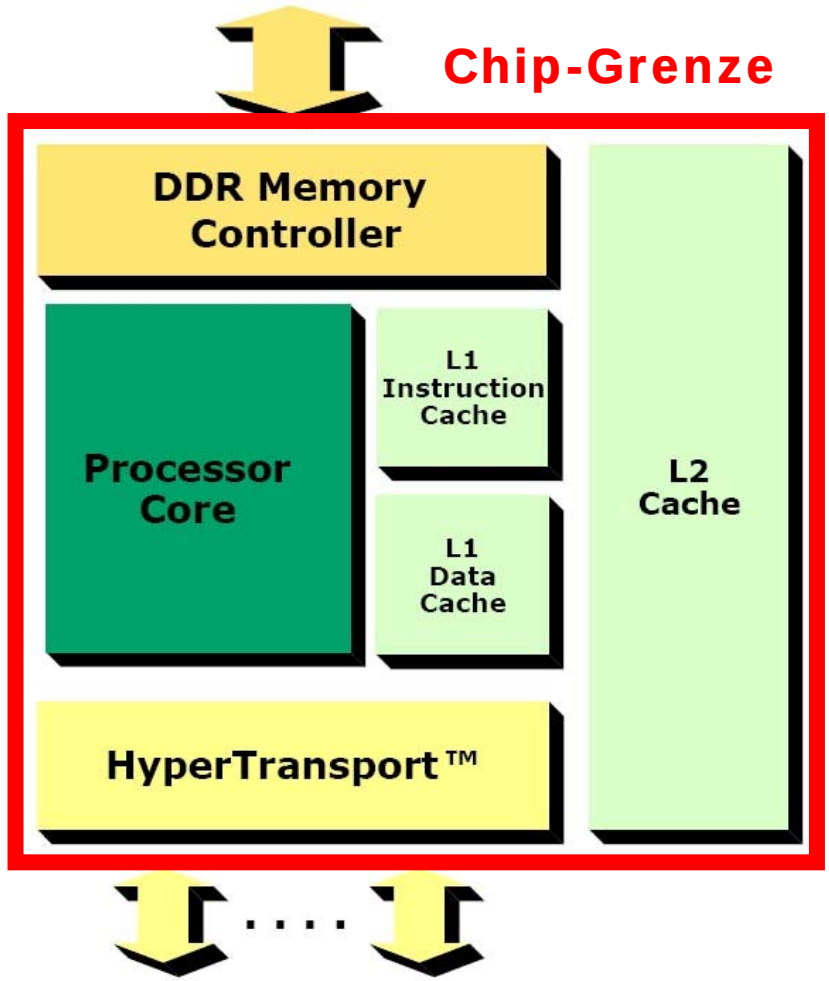
CISC: AMD

TECHNISCHE UNIVERSITÄT DARMSTADT

- Ab 1982: Zweithersteller von Intels 8086 und 8088
- Ab 1986: AMD bietet geklonte Intel-CPUs günstig an
  - Am286, Am386, Am486, Am5x86
- 1996: K5
  - Erste komplett von AMD entwickelte CPU
  - K: Krypton
    - Comic-Literatur: einzige Substanz, die Superman besiegen kann
- 1996: AMD kauft NexGen auf
  - NexGen: x86-CPU-Hersteller mit RISC-Kern
- 1997: K6, basierend auf NexGen-Design
- 1999: K7, auch Athlon (ab 2001 Athlon XP)
- 2003: K8, Athlon64
- 2005: "K9", Dual-Core-K8
- 2007: K10

AMD K8: Athlon64

- DDR-Controller im Chip
  - Intel: Northbridge (bis 2008)
- HyperTransport
  - Schnelle I/O-Anbindung
    - Erweiterungs-Karten
    - Andere CPUs
  - Intel: erst ab 2009 QuickPath



Blockdiagramm Athlon64 (2003)

AMD seit 2007: K10

TECHNISCHE UNIVERSITÄT DARMSTADT

- Bis zu 4 Kerne auf 1 Die
- Gemeinsamer L3-Cache mit Smart-Fetch
  - Kern-Zustand im L3-Cache zwischenspeichern
  - Kerne nahezu komplett abschalten
- Speicher-Adressierung: 48 Bit
  - Bis zu 128 TB adressierbar
  - K8: 40 Bit
- SSE4a
  - 128 Bit pro Takt
  - K8: 64 Bit
- Verbesserter Speicher-Controller
  - Mehr DRAM-Bänke
  - Größere Burst-Längen, Gleichzeitiges Lesen und Schreiben auf den Speicher
- Split Power Planes (CPU cores / memory controller)

Inhalt

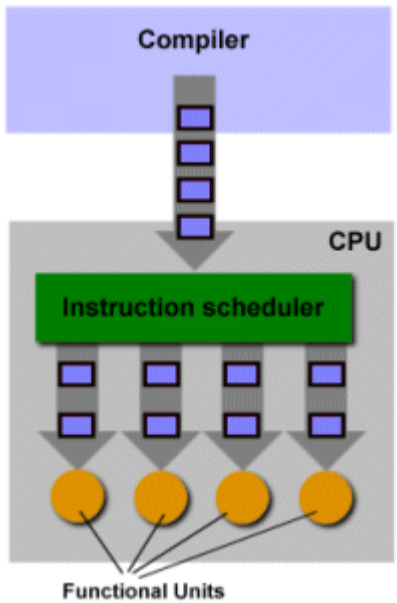
TECHNISCHE UNIVERSITÄT DARMSTADT

- CISC / RISC
  - Unterschiede, warum diese Trennung?
- RISC-Architekturen
  - MIPS, Sparc, POWER
- Techniken zum Beschleunigen einer CPU
- CISC-Architekturen
  - Intel, AMD
- VLIW
- Taktrate, Rechenleistung, Energie (sparen)
- Alternative Rechenarchitekturen
  - Cell, GPU, Adaptive Rechner
- Ausblick

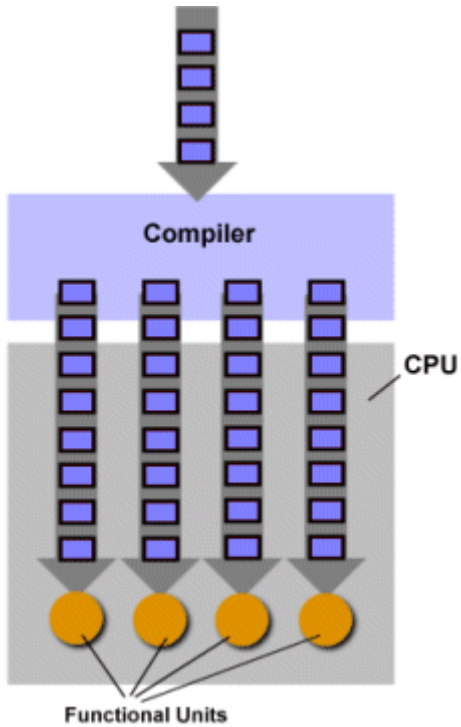
Intel: VLIW

- 2001: Intel + HP entwickeln VLIW-CPU "Itanium"
- VLIW = Very Long Instruction Word
  - Compiler gruppiert parallel ausführbare Befehle
  - Ein "langes" Befehlswort
    - Bestehend aus mehreren Befehlen
    - Werden parallel bearbeitet

superskalar



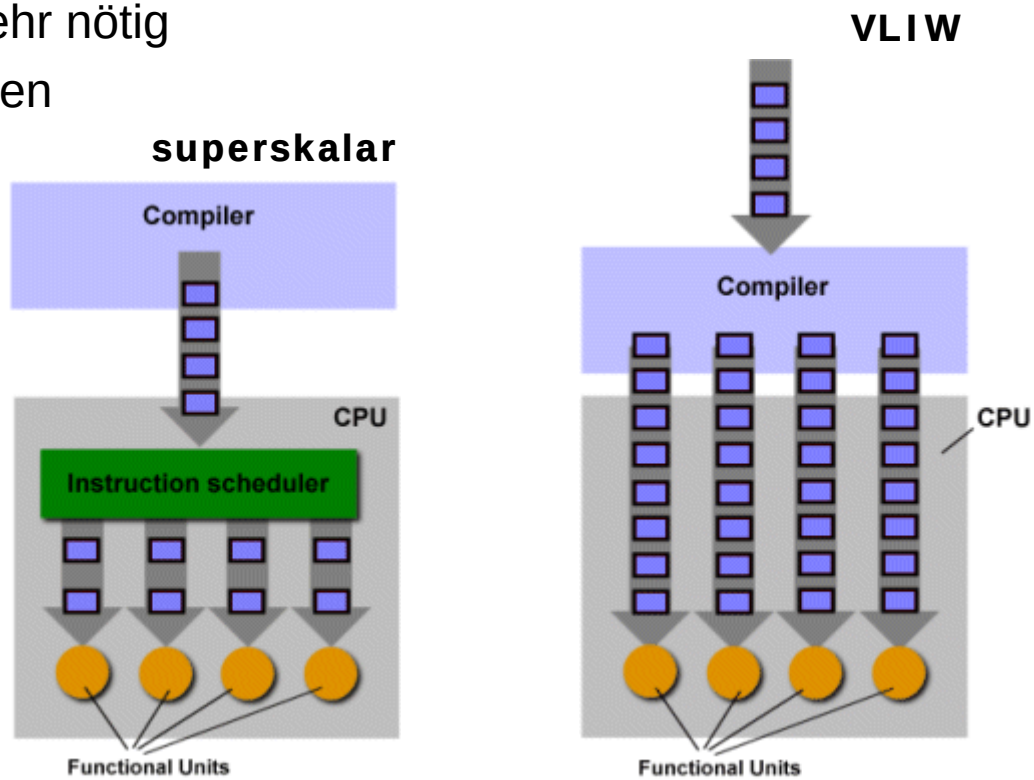
VLIW



Intel: VLIW

TECHNISCHE UNIVERSITÄT DARMSTADT

- Vorteile VLIW gegenüber herkömmlich superskalar:
  - CPU-Fläche kann eingespart werden
    - Instruction-Scheduler in CPU nicht mehr nötig
    - Out-of-Order-Execution nicht mehr nötig
  - Compiler kann besser parallelisieren
    - Hat mehr Zeit für Optimierungen
    - Weiß mehr über den Programmfluss



Intel: VLIW

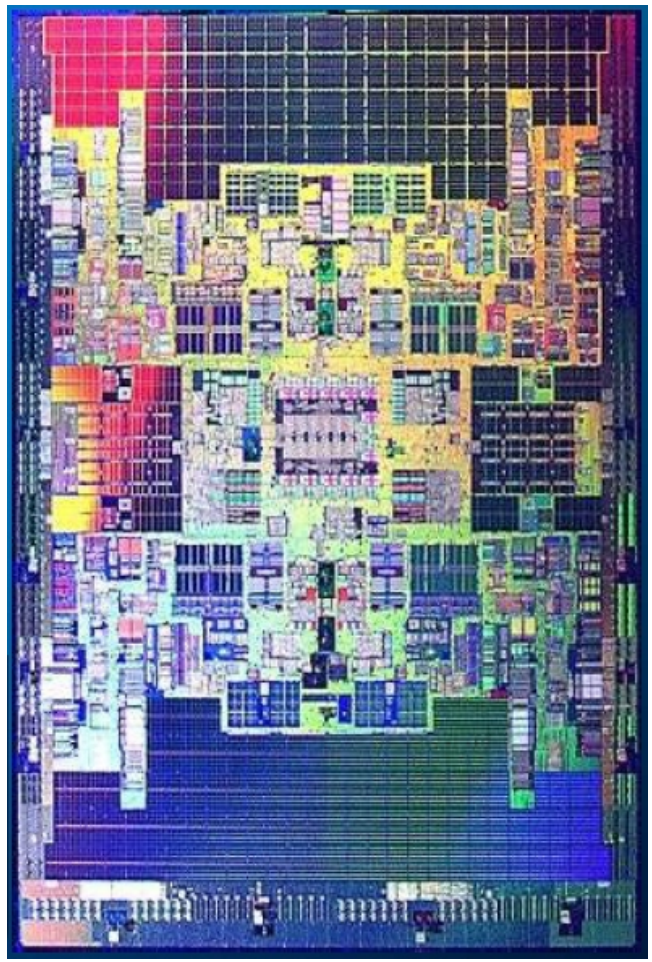
TECHNISCHE UNIVERSITÄT DARMSTADT

- Nachteile VLIW:
  - Compiler müssen intelligenter sein
  - Keine Kompatibilität zu bestehendem (superskalarem) Maschinencode
    - Bzw. nur langsamer Emulations-Modus
- Itanium
  - 2001 Itanium:
    - Nur knapp schneller als vergleichbare superskalare CPUs
    - Grund: sehr hohe Cache-Latenzen
  - 2002 Itanium 2:
    - Geringere Cache-Latenzen
    - L3-Cache im Chip
    - Schnellerer und breiterer externer Bus (*front-side bus, FSB*)
      - Itanium 2: 400 MHz bei 128 Bit
      - Itanium 1: 266 MHz bei 64 Bit

Probleme Itanium

- Keine Kompatibilität zu bestehenden Programmen
  - Emulations-Modus zu langsam
- Theoretische VLIW-Vorteile nicht in Praxis beobachtet
- Compiler-Entwicklung komplex

- Modelle
  - 2001: Itanium
  - 2002: Itanium 2
  - 1Q2010: Itanium Tukwila (Mehrfach verschoben)
    - 4 Kerne
    - QuickPath
    - 30 MB L3-Cache
    - Scalable Buffer Memory



Intel Itanium Tukwila

Inhalt

TECHNISCHE UNIVERSITÄT DARMSTADT

- CISC / RISC
  - Unterschiede, warum diese Trennung?
- RISC-Architekturen
  - MIPS, Sparc, POWER
- Techniken zum Beschleunigen einer CPU
- CISC-Architekturen
  - Intel, AMD
- VLIW
- Taktrate, Rechenleistung, Energie (sparen)
- Alternative Rechenarchitekturen
  - Cell, GPU, Adaptive Rechner
- Ausblick

Taktrate, Rechenleistung, Energie

TECHNISCHE UNIVERSITÄT DARMSTADT

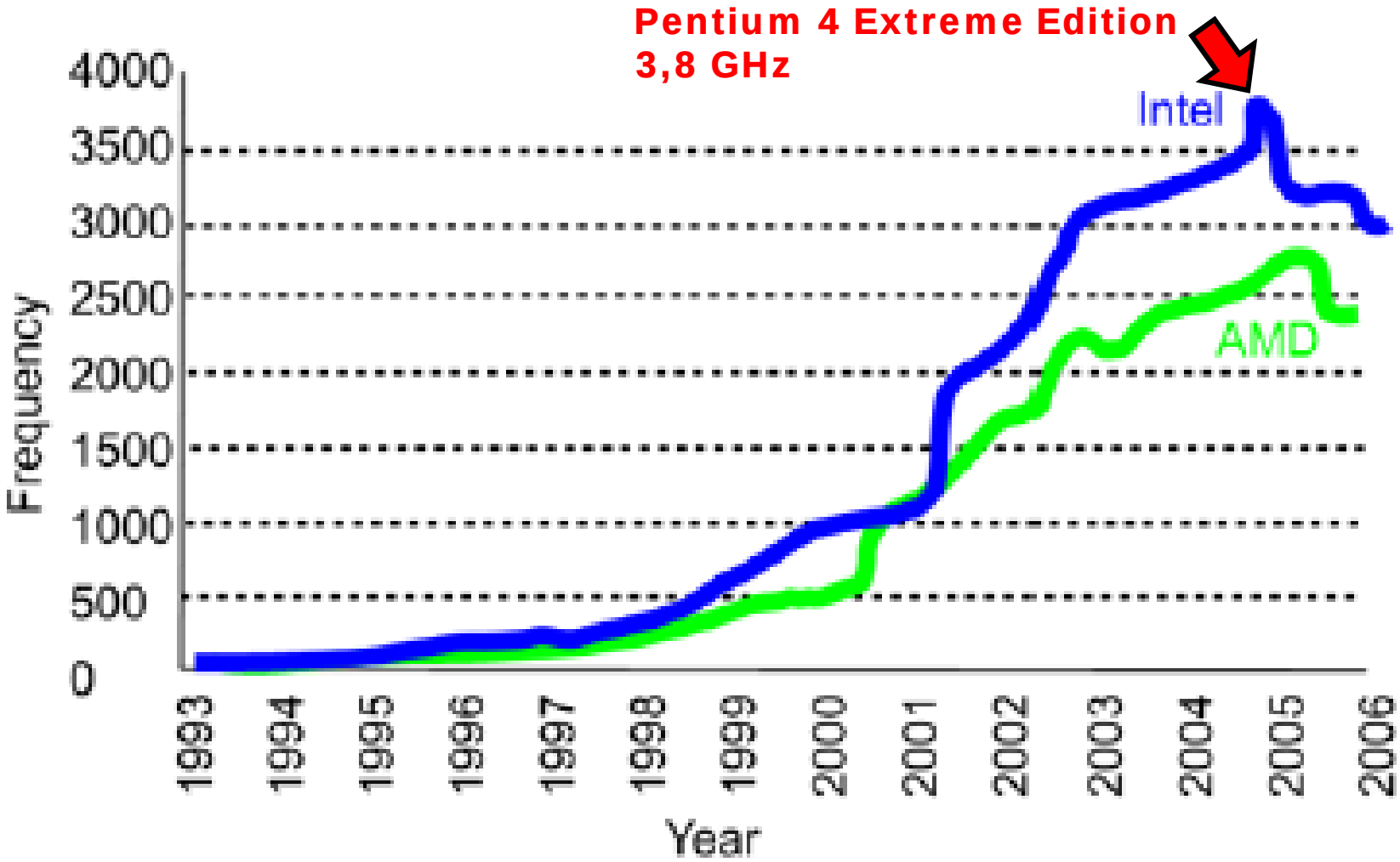
- Traditionell:
  Rechenleistung erhöhen = Taktrate erhöhen
- Intel-Roadmap 2003
  - Für heute waren > 10 GHz geplant

	Quartal 4/2003	Quartal 2/2004	Quartal 4/2004	Quartal 2/2005	2006	2006/2007
CPU-Kern	Prescott		Tejas		Nehalem	
Sockel	Sockel 478	Sockel 775	Sockel 775	?	?	?
Takt	3,2-3,6 GHz	3,6-4,4 GHz	4,4-5,6 GHz	5,6-9,2 GHz	ab 9,6 GHz	ab 10,2 GHz
FSB	800	800	800	1066	1066	1200
Fertigung	90 nm	90 nm	90 nm	65 nm	65 nm	?
Sonstiges	16kB L1-Cache 1MB L2-Cache 12.000 MicroOps Trace-Cache PNI (Prescott New Instructions)		24kB L1-Cache 1MB L2-Cache 16.000 MicroOps Trace-Cache TNI (Tejas New Instrucions) **16kB L1-Instructions** **Enhanced HyperThreading**		unbekannt	

CPU-Roadmap Intel, 2003

embedded systems + applications

Taktrate, Rechenleistung, Energie



Problem Energieverbrauch

TECHNISCHE UNIVERSITÄT DARMSTADT

- Kühlprobleme
  - Intel Core 2 Extreme QX6700:
    - 45 W / cm²
  - Bügeleisen:
    - 5 W / cm²
  - Übertaktungs-Rekord Pentium 4:
    - 8180,4 MHz mit Stickstoffkühlung
- Notebooks: Batterien schneller leer
  - Gewinnt immer mehr an Bedeutung:
  - Immer mehr Notebooks, immer weniger Desktops
  - Aber: Notebook-CPUs sparsamer, siehe später
- Server: Energiekosten
  - Google's jährliche Stromrechnung: 50.000.000 US-$
  - Amsterdam: Server-Farmen schlucken 25 % des Stroms

Pentium 4 stickstoff-gekühlt

embedded systems + applications

Energieverbrauch in CMOS

- Energie = Kapazität * Spannung² * Taktrate

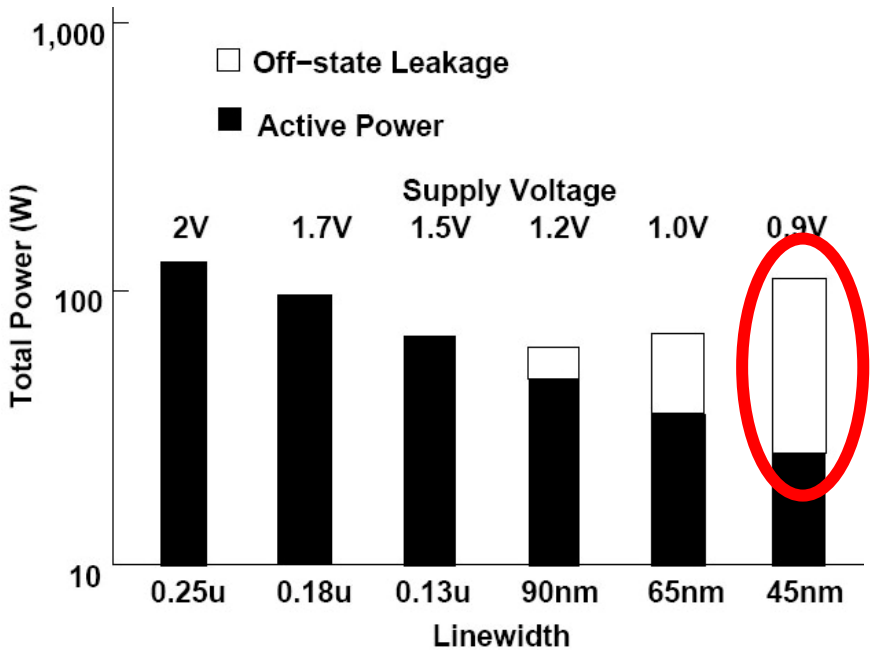
Strukturgröße

1978:		**3000 nm, 5 V**	**5 MHz**
	/ 1667		**x 600**
heute:		**45 nm, 1 V**	**3000 MHz**

Energieverbrauch in CMOS

TECHNISCHE UNIVERSITÄT DARMSTADT

- Problem: Leckströme



embedded systems + applications

Energie sparen

TECHNISCHE UNIVERSITÄT DARMSTADT

- Fertigungsebene
  - Kleinere Strukturen
  - Neue Materialien (Transistoren, Leitungen)
  - ...
- *Architekturebene*

Energie sparen generell

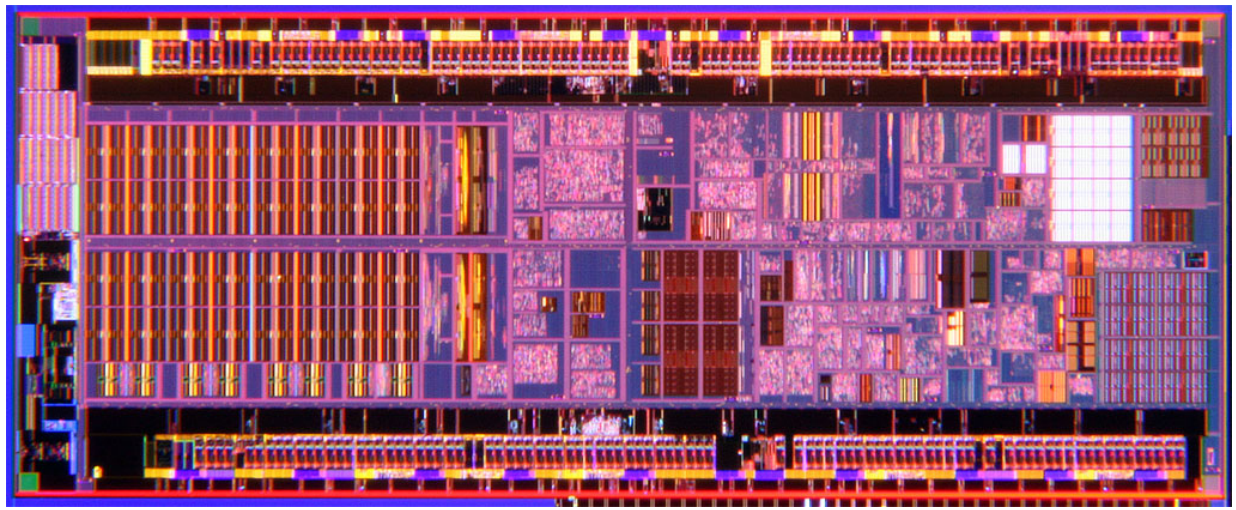
TECHNISCHE UNIVERSITÄT DARMSTADT

- Energie = Kapazität * Spannung² * Taktrate
  - Geringere Kapazität: weniger Transistoren
    - Einfachere Pipeline
      - Weniger parallele Recheneinheiten
      - In-Order- statt Out-of-Order-Execution
      - Keine Spekulation
    - Kleinere Caches
      - Oder dynamisch verkleinern
    - Speicheranbindung: geringere Bitbreite
  - Geringere Taktrate / Spannung
    - Generell
    - Taktrate der Rechenlast anpassen (Intel Nehalem)
    - Komplettabschaltung im Idle-Betrieb

Beispiel: Intel Atom

TECHNISCHE UNIVERSITÄT DARMSTADT

- Architektur
  - 2-fach skalar
  - In-Order-Pipeline
  - Keine Spekulation
  - Cache dynamisch verkleinerbar



Intel Atom CPU

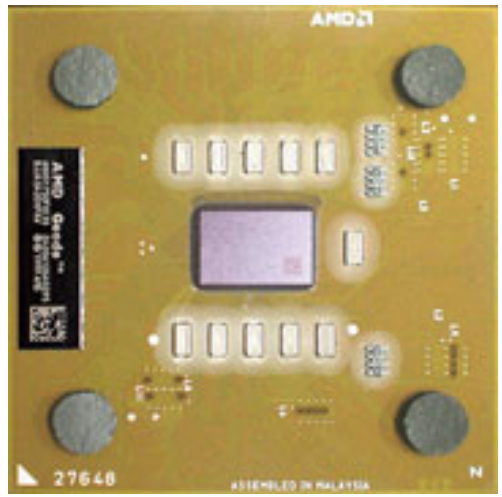
- Beispiel-Chips
  - Silverthorne: 45 nm, 800 MHz, 2 W
  - Diamondville: 45 nm, 1,6 GHz, 4 W
  - Dual Diamondville: 45 nm, 1,6 GHz, 8 W

embedded systems + applications

Beispiel: AMD Geode

TECHNISCHE UNIVERSITÄT DARMSTADT

- Ursprung: Cyrix
- AMD: Geode GX, LX
  - Sockel 7
  - Integrierter Grafikcontroller
- AMD Geode NX (2004)
  - Basis Athlon K7, Sockel A
  - 3-fach skalar
  - Out-of-Order-Pipeline



AMD Geode NX

- Beispiel-Chips (alle 130 nm)
  - Geode LX, 600 MHz, 5,1 W
  - Geode NX, 667 MHz, 6 W
  - Geode NX, 1 GHz, 9 W
  - Geode NX, 1.4 GHz, 25 W

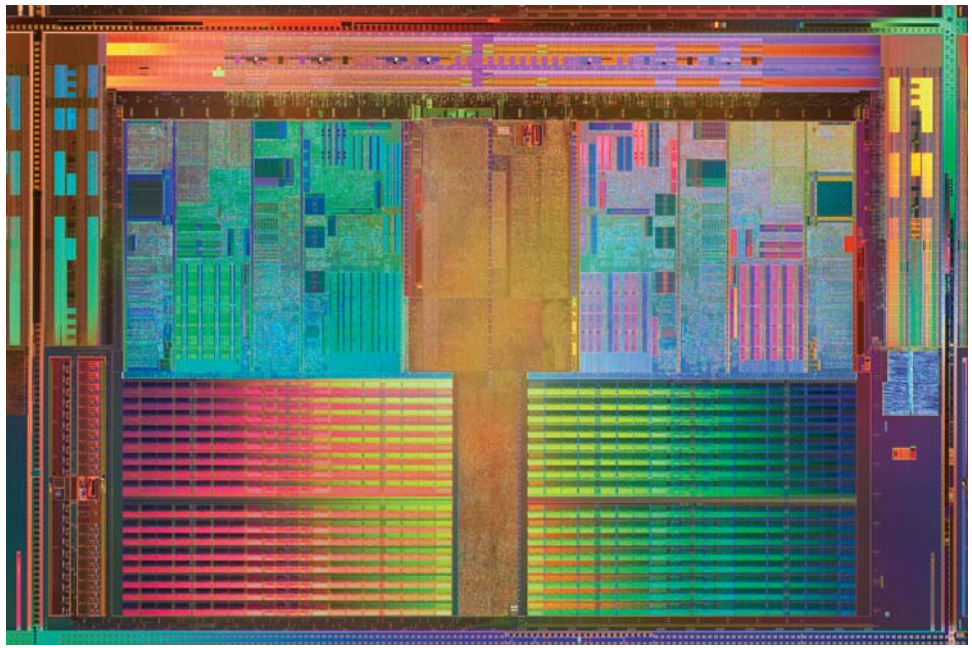
ab 2009 keine Weiterentwicklung mehr

embedded systems + applications

Beispiel: AMD Turion

TECHNISCHE UNIVERSITÄT DARMSTADT

- Architektur
  - AMD K8L
    - 2008: + Phenom
  - Split Power Planes
  - Kein L3-Cache



AMD Turion X2 CPU

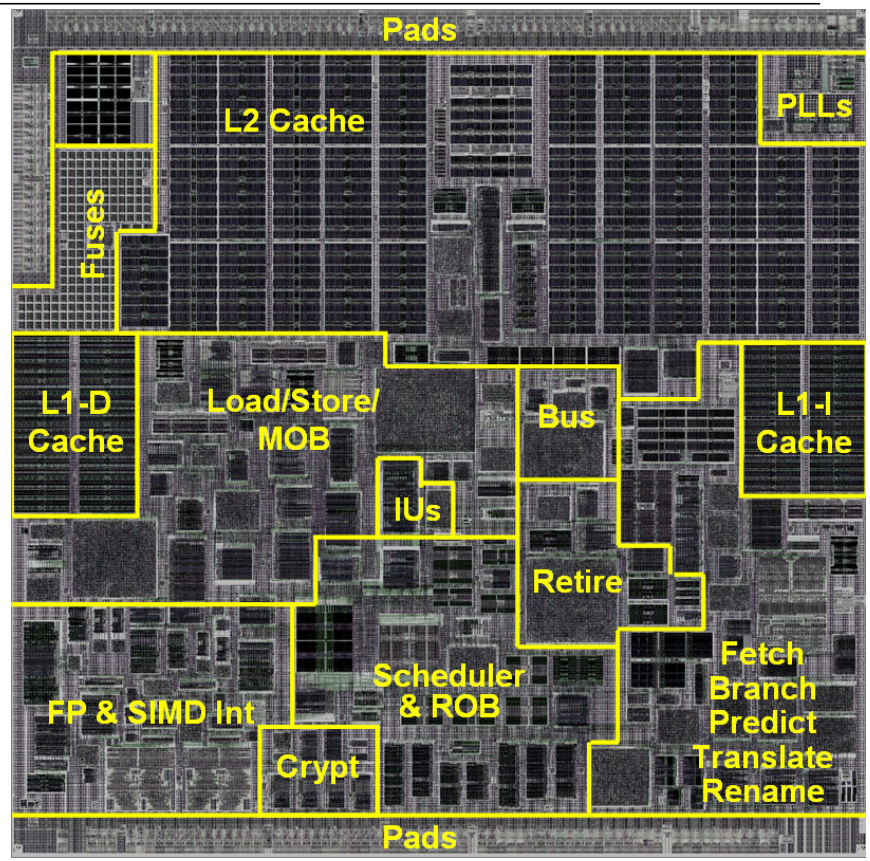
- Beispiel-Chips
  - 2005: 90 nm, 2,2 GHz, 25 W
  - 2007: 65 nm, 2,4 GHz, 35 W, Dual-Core
  - 2008: 65 nm, 2,0 GHz, 31 W, Dual-Core
- 2009: Athlon Neo, 65 nm, 1,6 GHz, 15 W
  - Basis: Athlon 64 2600

embedded systems + applications

Beispiel: VIA Nano

- Architektur
  - 3-fach skalar
  - Out-of-Order-Execution
  - µOP-/macroOP-Fusion

- Beispiel-Chips
  - 65 nm, 1 GHz, 5 W
  - 65 nm, 1,3 GHz, 8 W
  - 65 nm, 1,8 GHz, 25 W



VIA Nano CPU

- 2007: Eden ULV, 90 nm, 500 MHz, 1 W
- Mitte 2009: Nano-E, 1333 MHz FSB (bisher 800)
- 2010: Dual-Core

Inhalt

TECHNISCHE UNIVERSITÄT DARMSTADT

- CISC / RISC
  - Unterschiede, warum diese Trennung?
- RISC-Architekturen
  - MIPS, Sparc, POWER
- Techniken zum Beschleunigen einer CPU
- CISC-Architekturen
  - Intel, AMD
- VLIW
- Taktrate, Rechenleistung, Energie (sparen)
- Alternative Rechenarchitekturen
  - Cell, GPU, Adaptive Rechner
- Ausblick

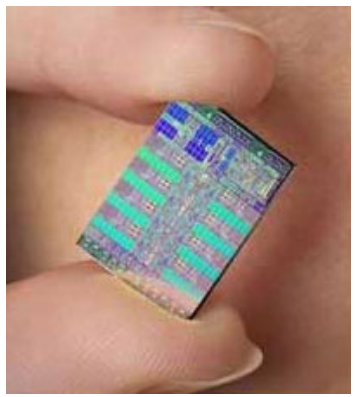
Cell-Prozessor: Cell Broadband Engine Architecture

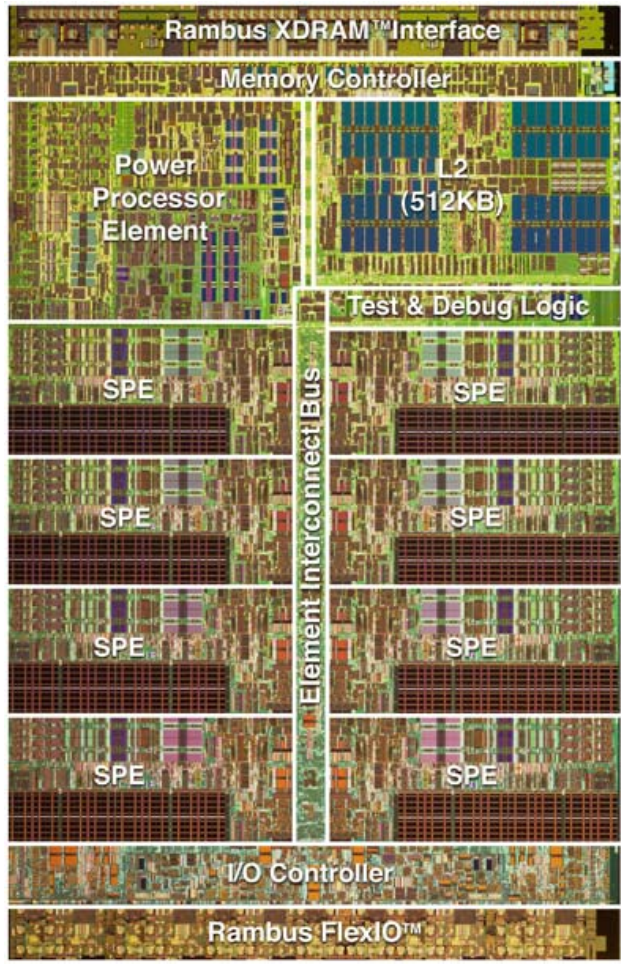
TECHNISCHE UNIVERSITÄT DARMSTADT

Entwickelt von STI (Sony, Toshiba, IBM)

- Basis: IBM POWER-Architektur (In-Order-Execution)

- Heterogen:
  - 1 Haupt-CPU
    - PPE = POWER Processor Element
    - Betriebssystem
  - 8 Co-Prozessoren
    - SPE = Synergistic Processing Elem.
    - Lokaler SRAM
    - 128 Bit SIMD
  - Zirkulärer Datenbus
    - EIB = Element Interconnect Bus
    - 4 Ringe à 128 Bit
    - Bis zu 3 Transfers pro Ring parallel

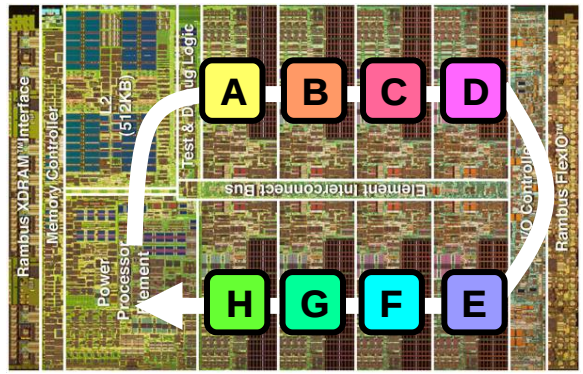




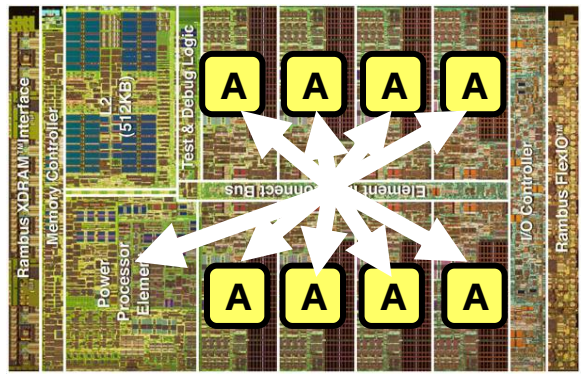
embedded systems + applications

Typische Anwendungen für Cell

- SPEs bilden Berechnungs-Pipeline
  - Z. B. Bilddatenverarbeitung in mehreren Schritten



- SPEs machen alle das gleiche
  - Parallele CPUs
  - Z. B. rechnen auf unterschiedlichen Bildbereichen



- Programmierer hat viele Freiheiten
  - Hardwarenahe Programmierung
  - Programmieren aber auch schwieriger!
    - Wie teile ich meinen Algorithmus auf PPE/SSEs auf?
    - Kommunikation

Cell-Prozessor: Einsatz

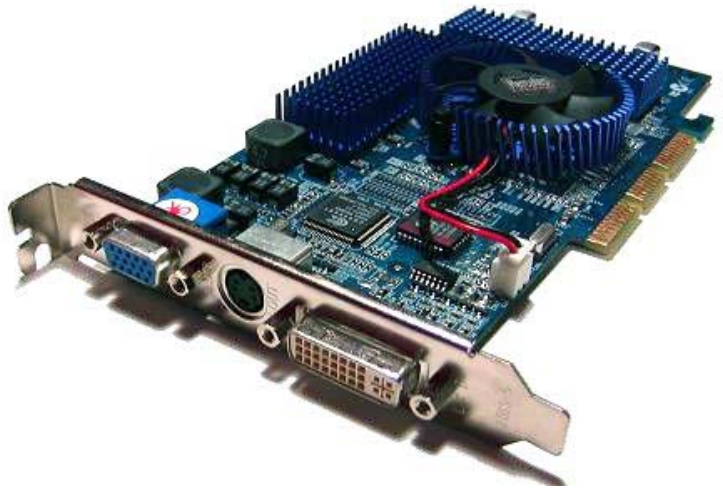
TECHNISCHE UNIVERSITÄT DARMSTADT

- 2005: 90 nm, Cell BE
  - Sony Playstation 3
  - Eingebettete Systeme
    - Medizin, Luftfahrt, Telekommunikation
  - IBM Blade-Server
    - Z. B. Filmindustrie
- 2007: 65 nm, PowerXCell 8i
  - IBM Roadrunner
    - Schnellester Supercomputer (seit Mai 2008), 1 Peta-FLOPS
      - 12.960 PowerXCell + 6.480 Opteron Dual-Core
  - Weitere Supercomputer
    - Green500: Plätze 1 - 7
      - Platz 1: 536 MFLOPS / Watt
- 2009: 45 nm

Grafiprozessor (*graphics processing unit, GPU)*

TECHNISCHE UNIVERSITÄT DARMSTADT

- 2001: Programmierbare Vertex-/Pixel-Shader (NVIDIA, GeForce 3)
  - Vertex-Shader: 3D-Punkt ⇨ 2D-Pixel
    - Berechnet Position, Farbe, Textur-Koordinate
  - Pixel-Shader: ändert Farbe von 2D-Pixeln
  - Shader werden programmiert mit
    - GLSL (OpenGL ARB)
    - Cg (NVIDIA)
    - HLSL (Microsoft)



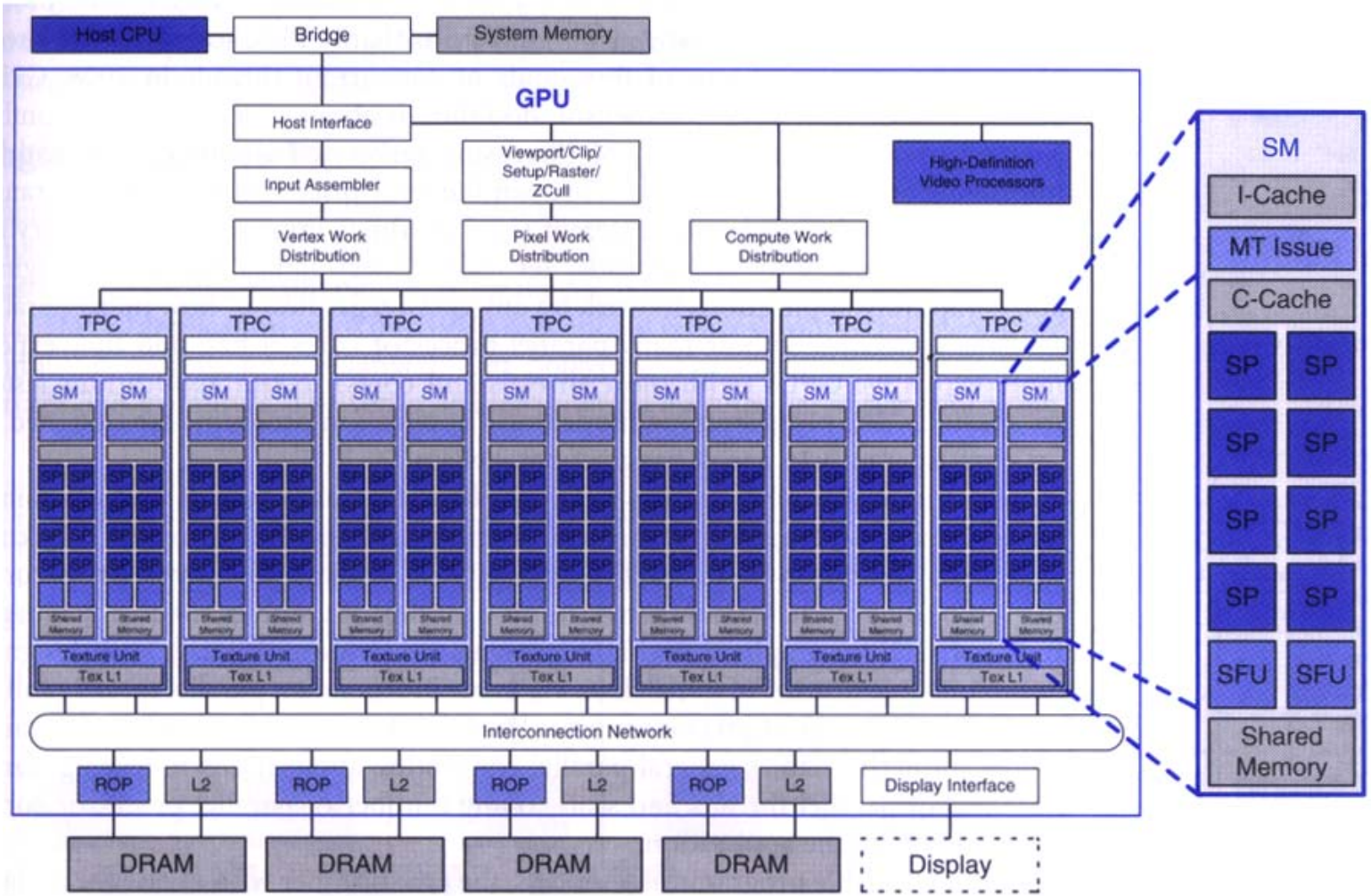
- Nutzung von GPUs als Co-Prozessoren ("GP-GPU")
  - Bilde Berechnungsprobleme auf Grafik-Semantik ab
  - Sehr umständlich

GPU: NIVIDIA GeForce 8

TECHNISCHE UNIVERSITÄT DARMSTADT

- Einheitliches Shader-Modell
  - Pixel- und Vertex-Shader nicht mehr getrennte GPU-Komponenten
- Einheitlicher programmierbar

GPU: NIVIDIA GeForce 8



GPU: NIVIDIA GeForce 8

TECHNISCHE UNIVERSITÄT DARMSTADT

- 1 Thread pro SP, 8 Threads gleichzeitig
  - Logisch: 4x8=32 Threads gleichzeitig
- Bis zu 1024 Threads verwaltbar pro SM
  - Sehr schnelle Task-Switches
    - Memory-Latenzen kaschieren
- Alle SPs führen denselben Code aus
  - Aber Verzweigungen möglich
    - Z. B. Thread-Nummer abfragbar
- SMs laufen voneinander unabhängig, Threads innerhalb SMs nicht
- Thread-Synchronisation nur innerhalb SM
- Speicherzugriffe nur schnell, wenn regulär und aligned
- Irreguläre Zugriffe: Shared Memory

embedded systems + applications

NVIDIA: CUDA

TECHNISCHE UNIVERSITÄT DARMSTADT

- CUDA: Compute Unified Device Architecture
- Eingeschränktes C mit Erweiterungen
- GPU-Funktionen:
  - Keine Rekursion
  - Keine variable Anzahl Argumente (z. B. printf)
  - Keine statischen Variablendefinitionen
  - Keine Funktions-Pointer
  - Funktions-Zuweisung an CPU / GPU:
    - __global__ / __device__: Ausführung auf GPU
      - __global__: CPU ruft auf
      - __device__: GPU ruft auf
    - __host__: Ausführung auf CPU, Aufruf durch CPU
- myFunction<<< gridDim, blockDim >>>(args);

 # SM-Zuweisungen

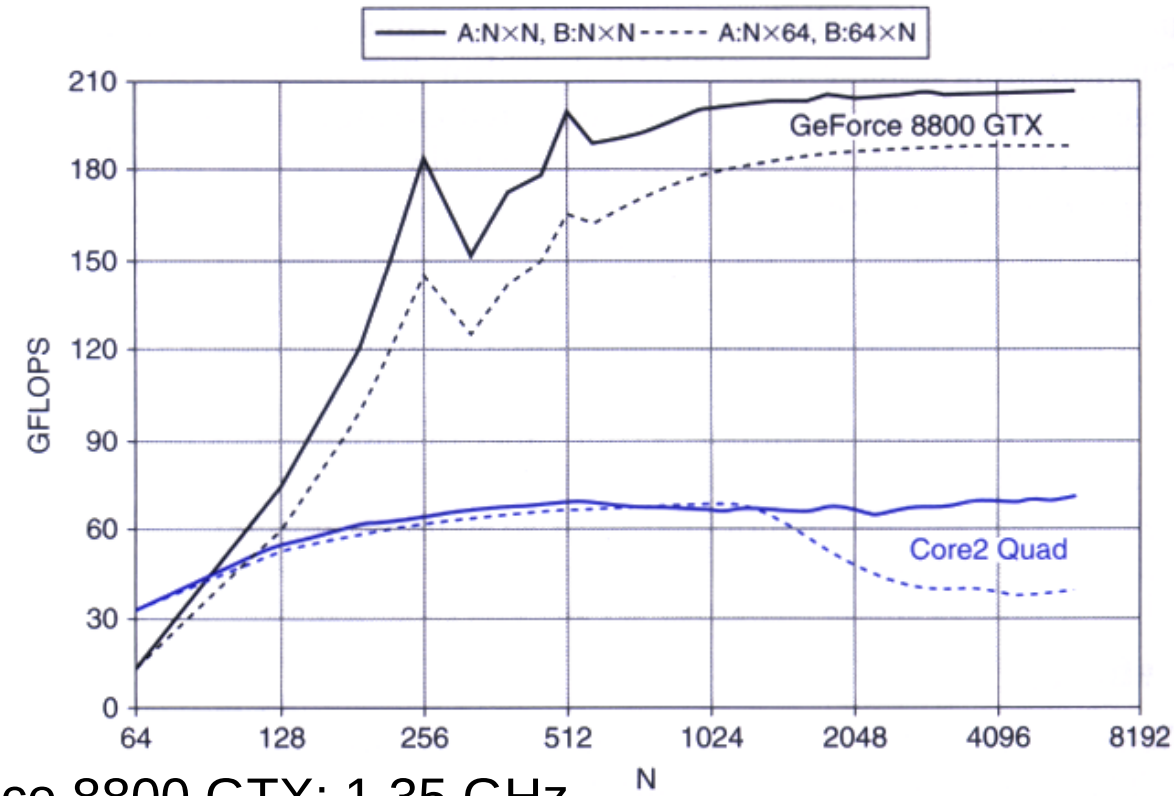
 # Threads pro SM

embedded systems + applications

GPU: Rechenleistung

TECHNISCHE UNIVERSITÄT DARMSTADT

- SGEMM dense matrix-matrix multiplication

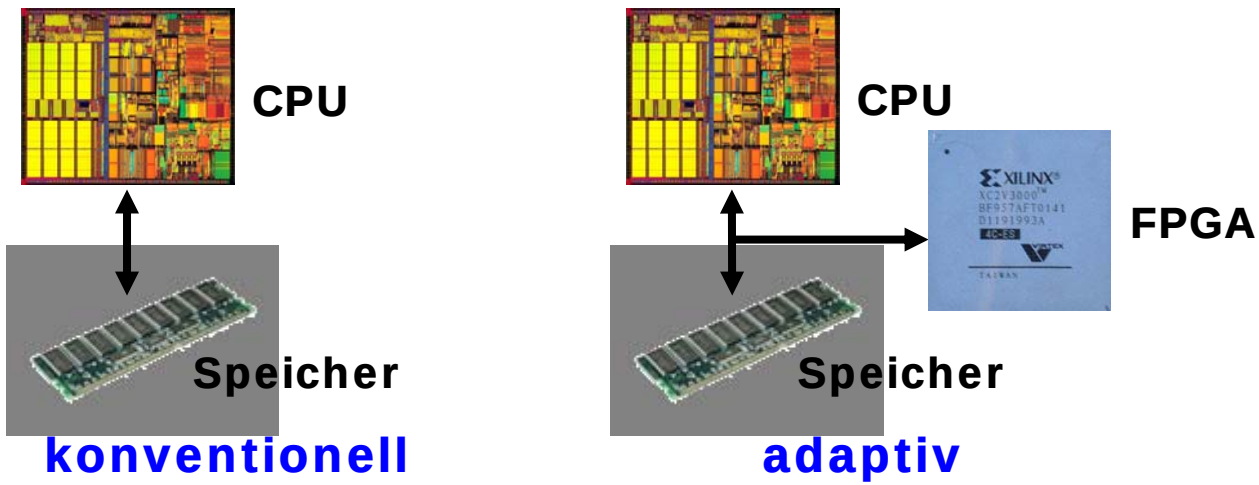


- GeForce 8800 GTX: 1,35 GHz
- Core2 Quad: 2,4 GHz Q6600, 64-bit Linux

embedded systems + applications

Adaptive Rechner

- Idee: Passe Hardware an Anwendung an
- Adaptiver Rechner = CPU + FPGA
  - FPGA beschleunigt laufzeitintensive Teile
    - Parallelisieren
    - Speicherzugriffe wenn möglich streamen
  - Rest: CPU



Adaptive Rechner

TECHNISCHE UNIVERSITÄT DARMSTADT

- Performance
  - SRC-6E
    - Brechen DES: 900x Speedup gegenüber Pentium 4 1,8 GHz
      - Extrem gut parallelisierbar, wenig speicherintensiv
  - ACE-V
    - Wavelet-Bildkompression
    - 5x Speedup gegenüber AMD Athlon 1,7 GHz
    - 1/50 des Stromverbrauchs

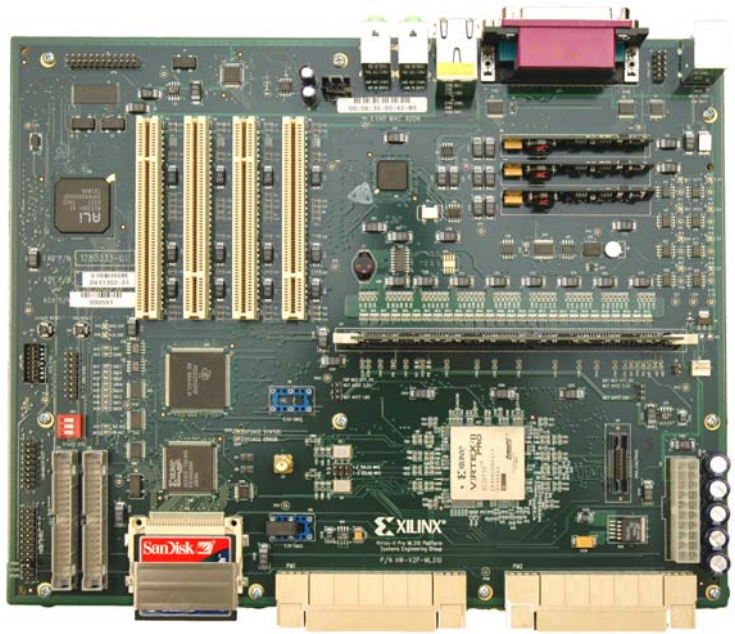
SRC-6E

ACE-V

embedded systems + applications

Adaptive Rechner

- Performance
  - ACE-M3
    - 32b-Wortspiegelung:
    - 14x Speedup gegenüber Pentium 4 @ 3,2 GHz,
    - ca. 1/30 - 1/100 des Stromverbrauchs



ACE-M3

- ACE-V: E.I.S. (TU BS)
- ACE-M3: ESA (TU DA)

Roadmap Intel

TECHNISCHE UNIVERSITÄT DARMSTADT

- 2009: 32 nm (Westmere)
  - Clarkdale: Westmere + Grafikcontroller
    - Separater Die, 45 nm
    - Verfügbar 1Q2010 als Core i3, i5 und mobile i7
- 2010: neue Architektur (Sandy Bridge)
  - 8 Kerne
  - 16 MB L3-Cache
- 2011: 22 nm (Ivy Bridge)
- 2012: neue Architektur (Haswell)
  - 8 Kerne
  - Neues Cache-Design
  - Fused-Multiply-Add

Roadmap AMD

TECHNISCHE UNIVERSITÄT DARMSTADT

- Bis 2010:
  - 4 Kerne max.
  - 8 MB Cache max.
  - DDR2 + DDR3
- 2010-2011: 32 nm
  - GPU-Integration „Fusion“ (Desktop + Notebook)
  - > 4 Kerne (Istanbul: 6, Magny Cours: 12)
  - > 8 MB Cache (Magny Cours: 12 MB)
  - DDR3

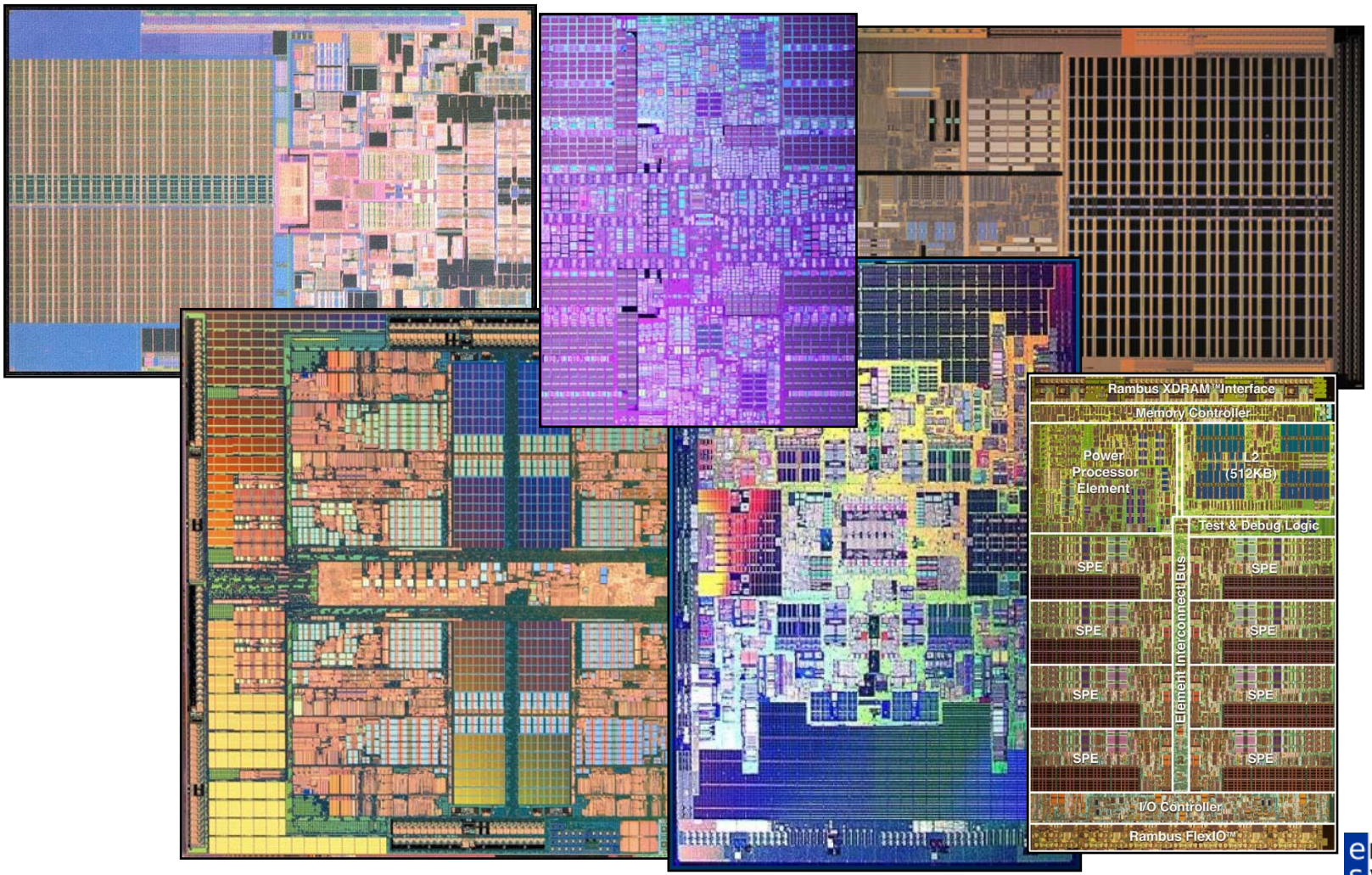
Ausbildung im Bereich Technische Informatik

TECHNISCHE UNIVERSITÄT DARMSTADT

- Kanonik „Computer Microsystems“
  - SS 2010: Wolfgang Heenes
  - SS 2011: Andreas Koch
- Grundlagen der Rechnertechnologie
  - SS 2010: Wolfgang Heenes
- Zahlreiche Spezialvorlesungen
  - Eingebettete Systeme: Sorin Huss
  - Rekonfigurierbare Prozessoren: Sorin Huss
  - Prozessorarchitekturen für rechenstarke eingebettete Systeme: Andreas Koch
  - Optimierende Compiler: Andreas Koch
  - …

Ende

TECHNISCHE UNIVERSITÄT DARMSTADT

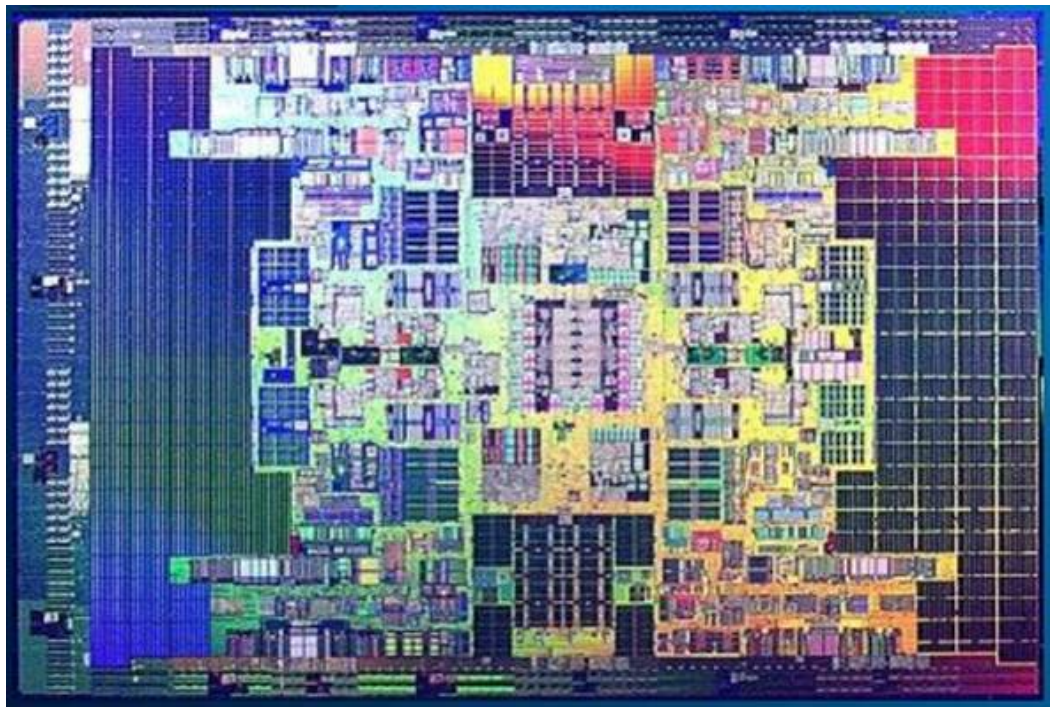


embedded systems + applications

Quiz 1 / 7

TECHNISCHE UNIVERSITÄT DARMSTADT

Wie heißt diese CPU?



Intel Itanium Tukwila

4 Cores

30 MB L3-Cache

embedded systems + applications

Quiz 2 / 7

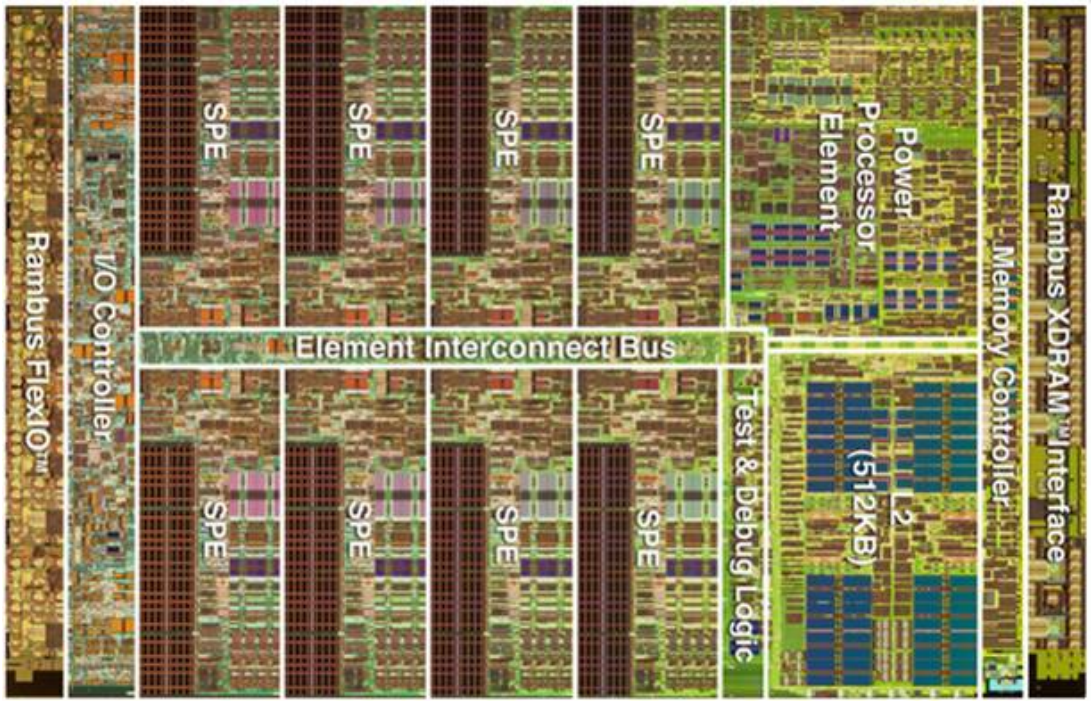
- Auf welcher CPU basieren
  - Cisco-Router
  - BMW-Navis
  - Nintendo 64, Sony PlayStation 2
  - Fritz!Box

- Antwort: MIPS

Quiz 3 / 7

TECHNISCHE UNIVERSITÄT DARMSTADT

Wie heißt diese CPU?



Cell-Prozessor

Heterogen: 1 PPE + 8 SPEs

3,2 GHz

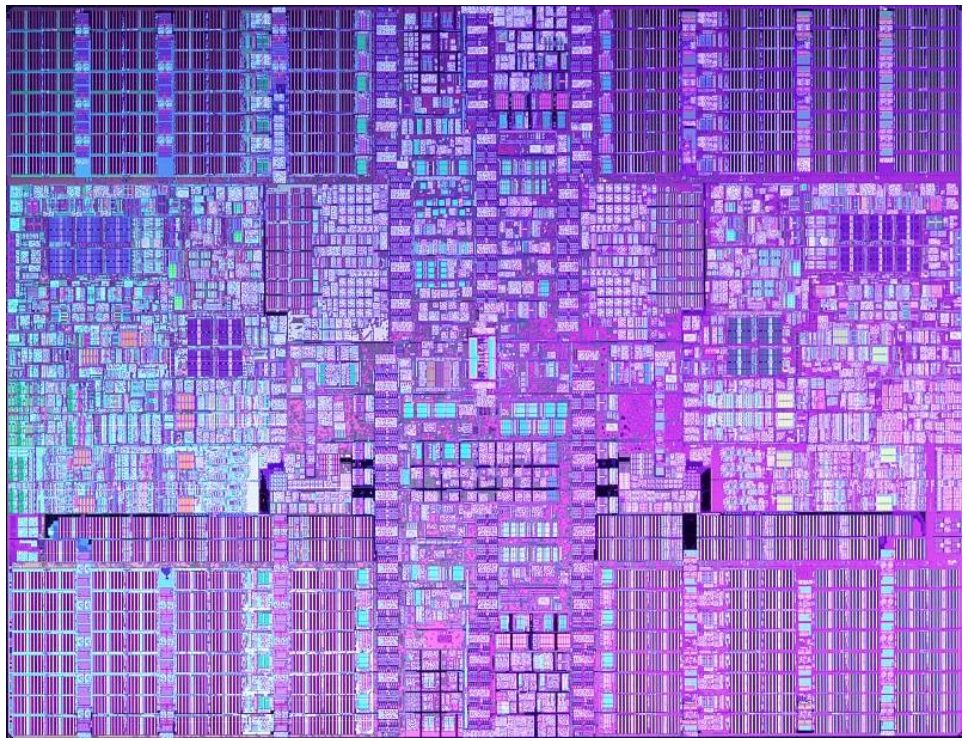
Playstation 3

embedded systems + applications

Quiz 4 / 7

- Welche maximale Taktrate erreicht der POWER6 von IBM?

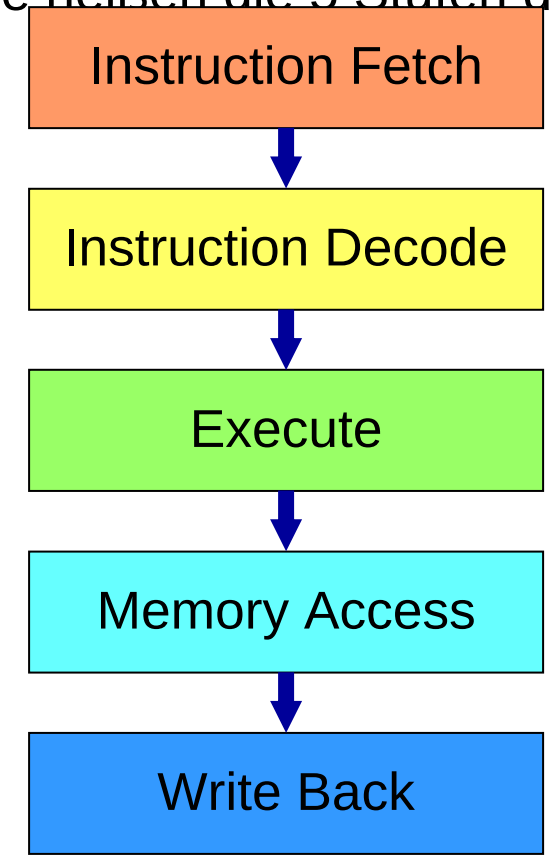
- Antwort:



Quiz 5 / 7

TECHNISCHE UNIVERSITÄT DARMSTADT

- Wie heißen die 5 Stufen der Befehlsbearbeitung?



embedded systems + applications

Quiz 6 / 7

TECHNISCHE UNIVERSITÄT DARMSTADT

- Welchen Vorteil hat CISC gegenüber RISC?

- Antwort: Abwärtskompatibilität

Quiz 7 / 7

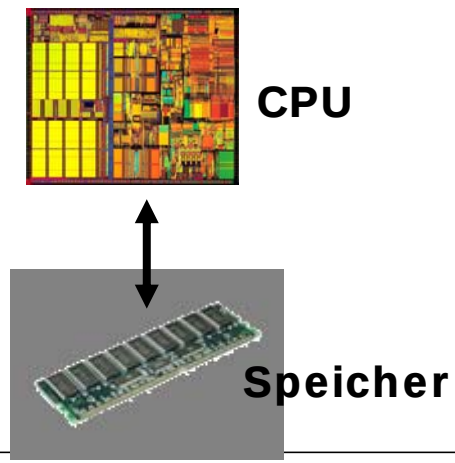
TECHNISCHE UNIVERSITÄT DARMSTADT

- Was ist COMRADE?

- Antwort: Ein Compiler für Adaptive Rechner

Wie rechnet eine CPU?

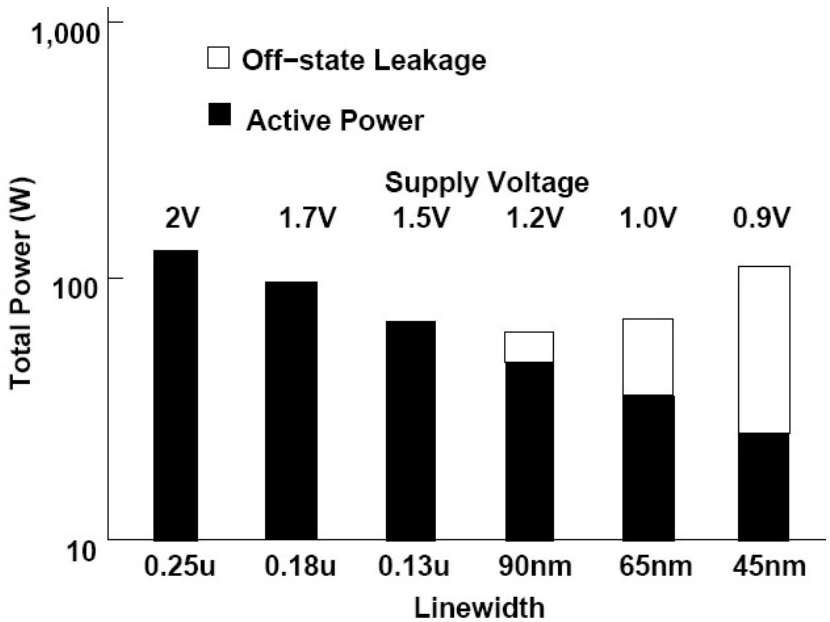
- Recheneinheit
  - ALU: Arithmetic Logic Unit
- Steuereinheit
  - Heute zusammen mit Recheneinheit in 1 Chip
    ⇨ CPU: Central Processing Unit
- Speicher
  - Daten
  - Programm (Folge von Instruktionen)



Rechnen
=
CPU transformiert Daten im Speicher

Warum nicht immer mehr GHz?

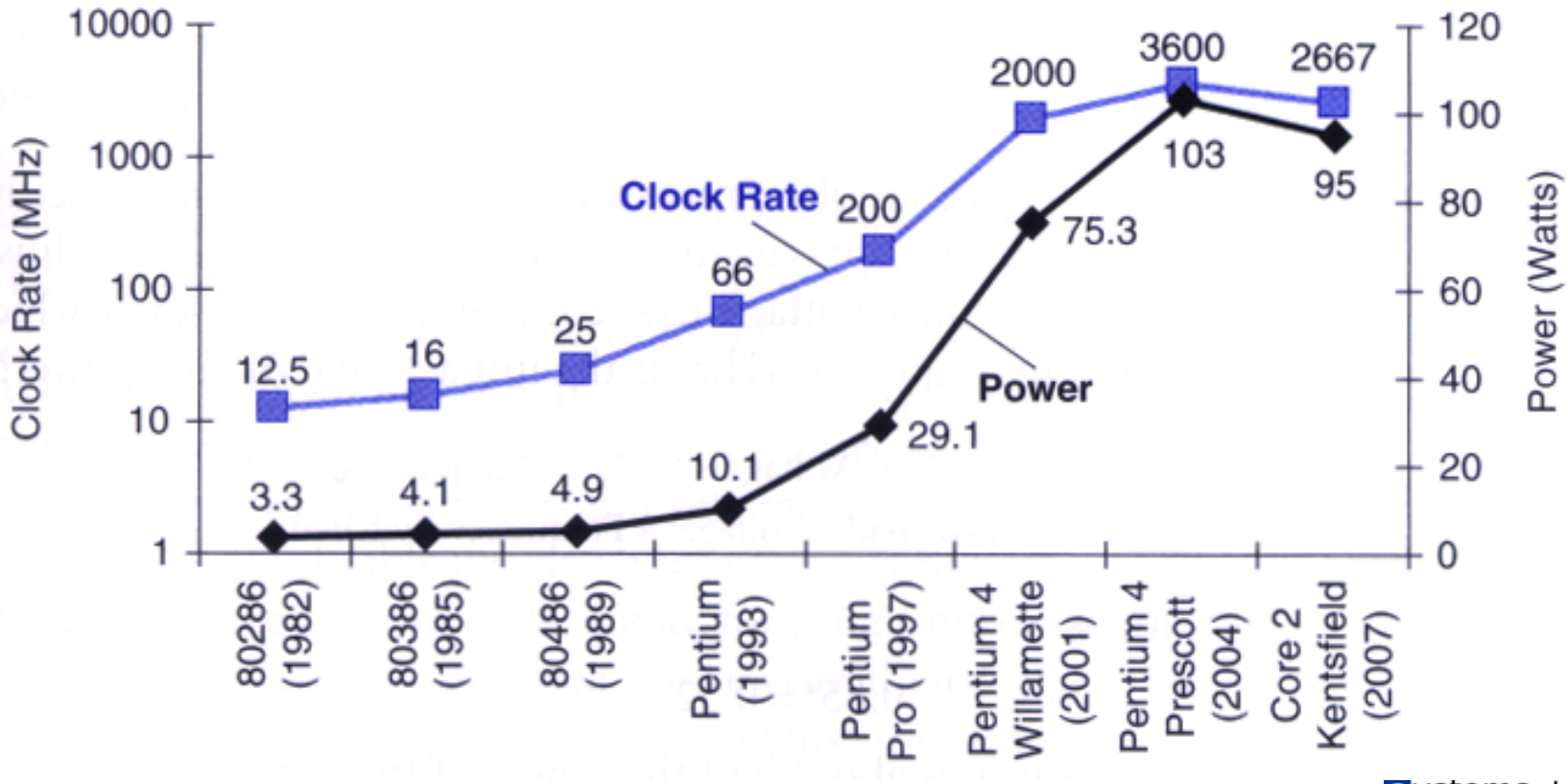
- Höhere Taktraten ⇨ höhere Anzahl Transistoren
  - Größere Caches
    - Weil Speicher-Takt langsamer wächst als CPU-Takt
  - Komplexere Einheiten und Steuerungslogiken
    - Z. B. wegen längeren Pipelines
- Mehr Transistoren ⇨ Mehr Energieverbrauch
  - Trotz immer kleineren Strukturen, weil:
  - Immer größere Leckströme



Taktrate, Rechenleistung, Energie

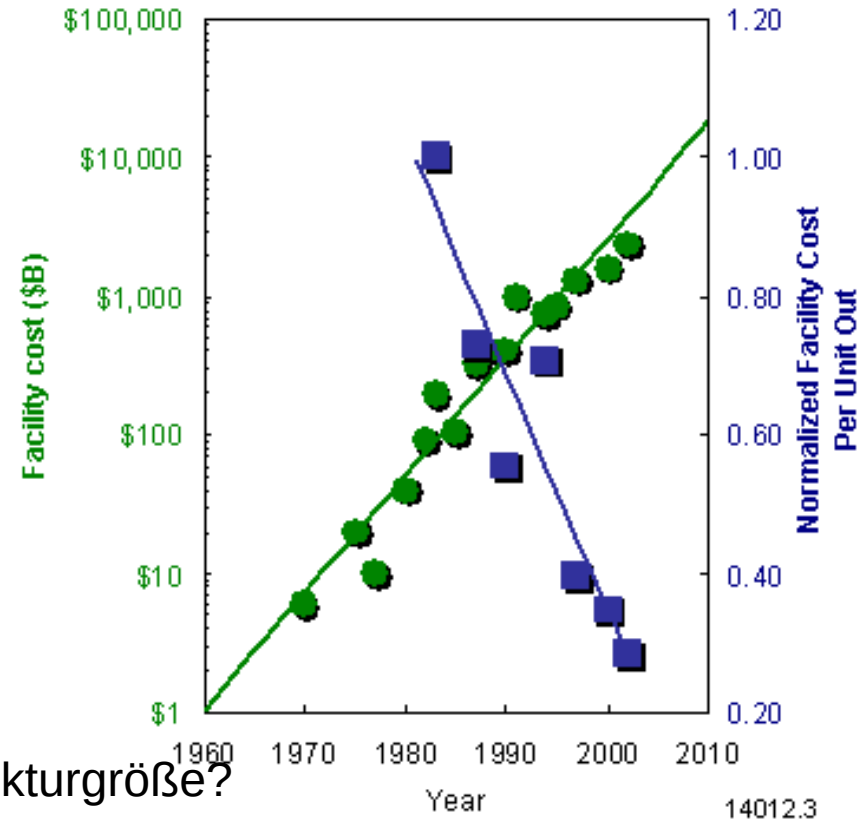
- Energie = Kapazität * Spannung² * Taktrate

1978:	**3000 nm**	**5 V**	**5 MHz**
heute:	**45 nm**	**1 V**	**3000 MHz**



Fabrikkosten

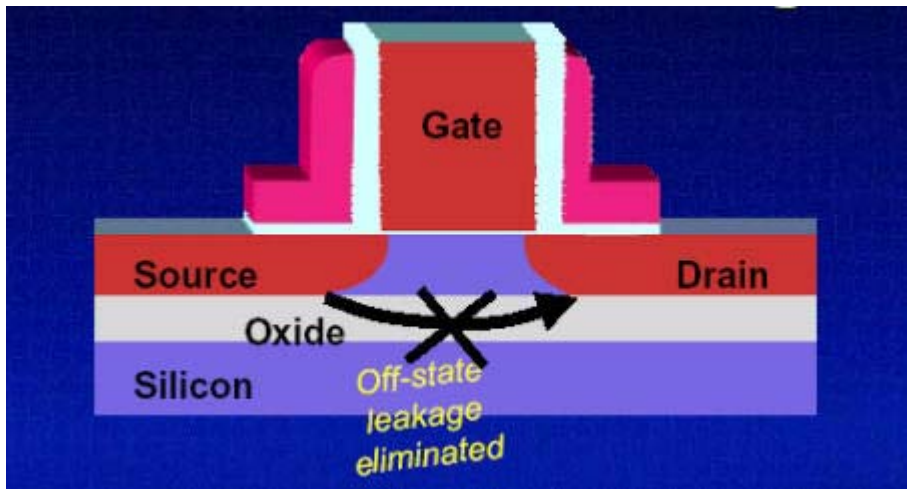
- Kleinere Strukturen ⇨
- Größere Kosten für Chipfabriken
  - 1970: 6 Mio. $
  - 2002: 2.000 Mio. $
  - 2010: 10.000 Mio. $
- Real 2009:
  - Intel plant 7 Mrd. $ für 3 Fabs à 32 nm



- Amortisierung:
  - Längere Lebenszeiten einer Strukturgröße?

Energie sparen

- Fertigungsebene:
  - Kleinere Strukturen
    - Geringere Versorgungsspannung
    - Weniger Kapazitäten, die umgeladen werden müssen
    - Allerdings größere Leckströme
      - Neue Technologien gefragt, z. B. SOI
  - Geringere Widerstände in Leitungen
    - Bessere Leitungsmaterialien, z. B. Cu statt Alu



Energie sparen

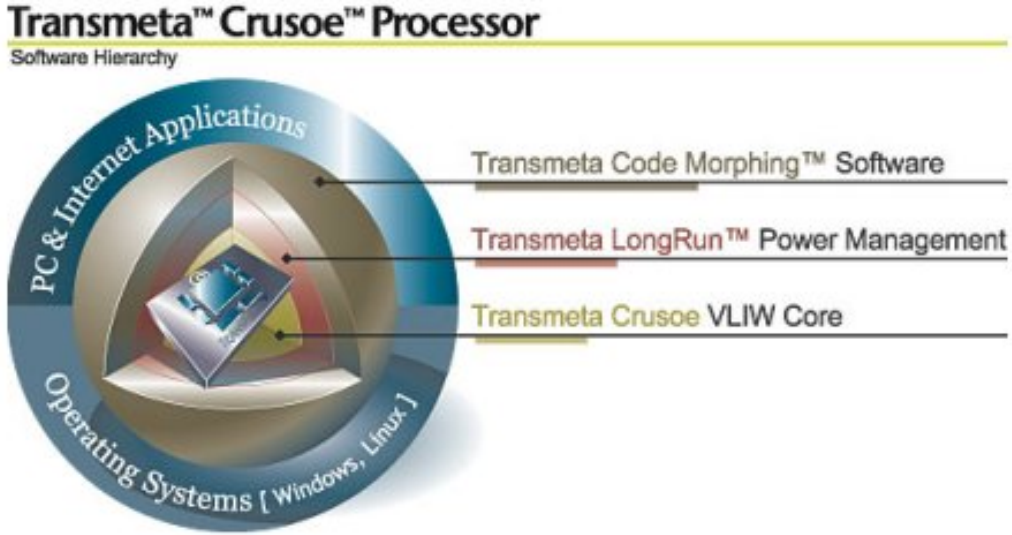
- Multi-Core
  - Verringere den Takt ein bißchen
    - Intel, Reinders: 87 %ige Performance ⇨ nur halber Stromverbrauch
  - Nun 2 Cores statt einem
  - Gleicher Energieverbrauch wie zuvor,
  - Aber knapp doppelte Rechenleistung
    - 74 % mehr

- Allheilmittel Multi-Core?
  - Server:
    - OK, ohnehin viele unabhängige Threads
  - Einzelne Applikation:
    - Nur parallelisierte Programme werden beschleunigt!
    - Programme meistens jedoch sequentiell

Transmeta

- Transmeta:
  - 1995 gegründet, Ziel: energie-sparsame CPUs
  - 2000: Crusoe
- Konzept:
  - Einfachere CPU
  - VLIW statt superskalar
  - Aber trotz VLIW: x86-kompatibel
    - Code-Morphing

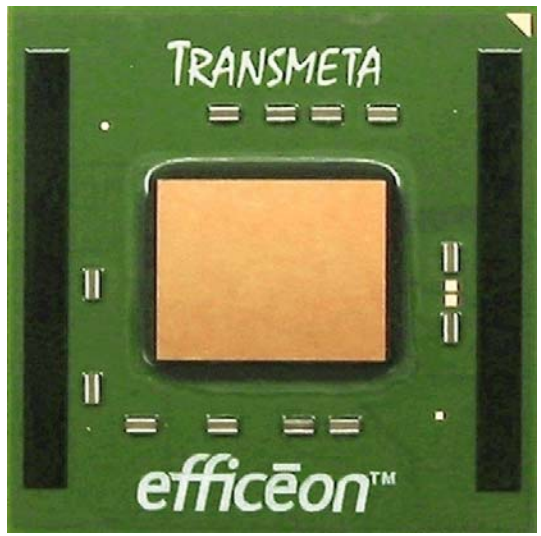
Transmeta

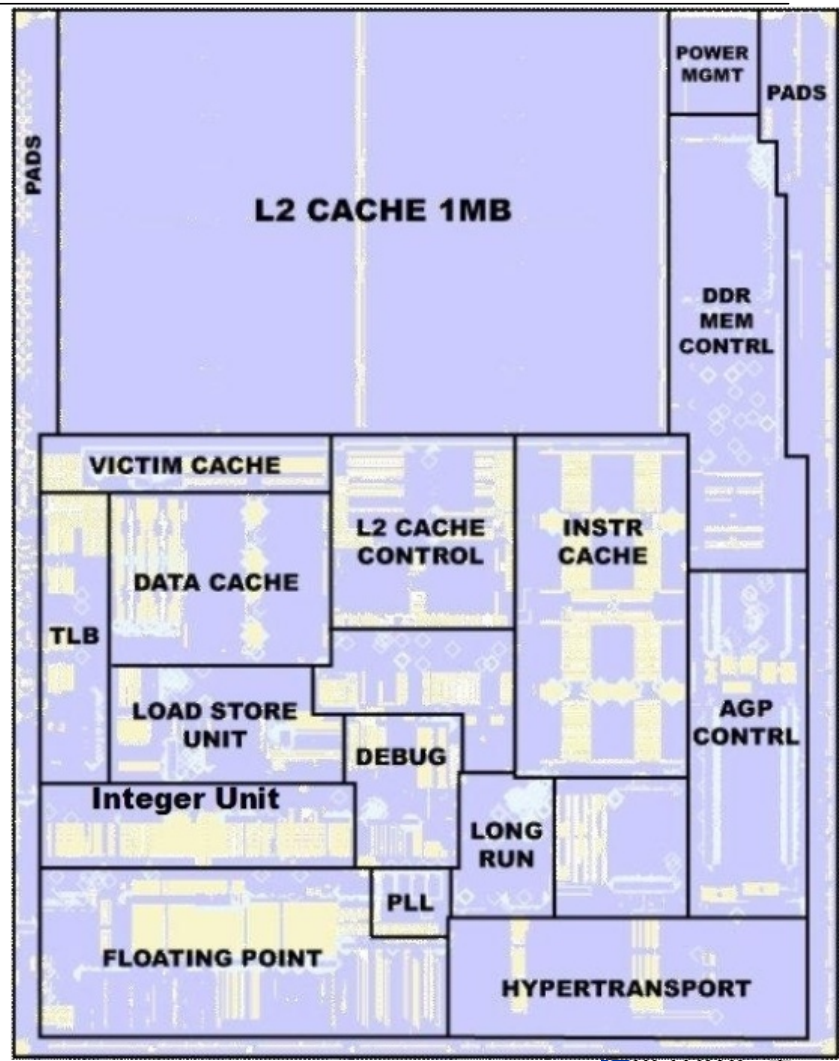


- Code-Morphing
  - VLIW-Parallelisierung nicht von Compiler
  - Sondern von CPU zur Laufzeit
  - Aber in Software (superskalar: in Hardware):
    - Rechner-Start
    - CPU lädt spezielle Codemorphing-Software (CMS) in RAM
    - CMS setzt x86-Befehle zur Laufzeit in VLIW-Befehle um
      - Cache VLIW-Befehle
    - Wiederholung einer Befehlssequenz (Schleife):
      - Verwende gecachten VLIW-Befehl
  - Je länger Programm läuft, umso schneller wird es

Transmeta

- 2003: Efficeon, verbesserter Crusoe:
  - VLIW: 256 statt 128 Bit
  - LongRun:
    - Wenn nicht volle Rechenpower nötig:
    - Taktrate verringern
    - Kernspannung absenken



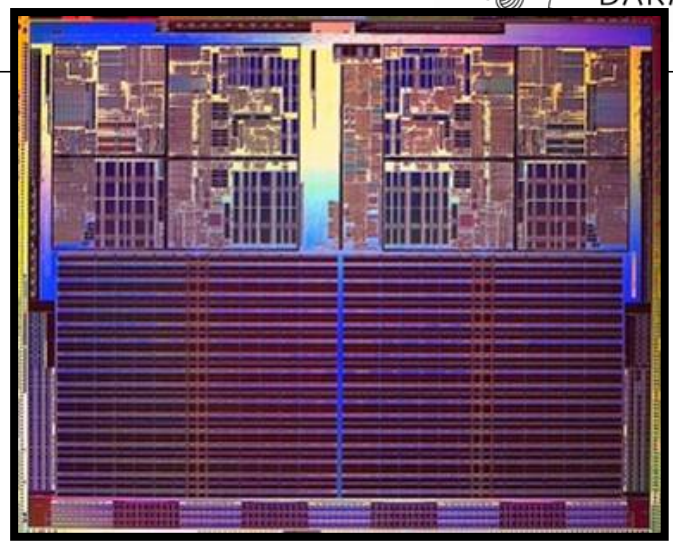


Transmeta

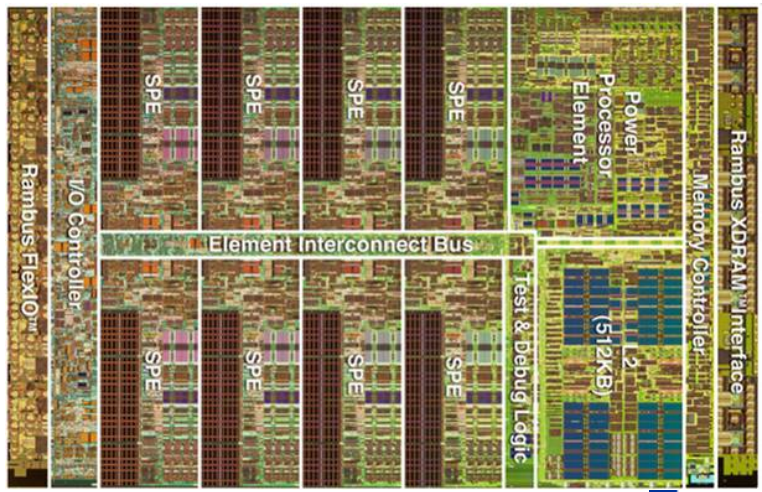
- Crusoe / Efficeon nicht erfolgreich genug
  - 1300 MHz Efficeon: 14 W
  - 1300 MHz Pentium M: 22 W
- 2005: Verkaufsversuch nach China, missglückt
- 2006: AMD vertreibt Efficeon
  - AMD's Cool'n'Quiet = Transmeta's LongRun?

Heterogene Strukturen

- Multicore bislang:
  - 2, 4, 8 identische Rechenkerne



- Heterogen:
  - 1 oder mehrere Haupt-Prozessoren
  - + mehrere Co-Prozessoren



Variable Hardware

- Bisher: Hardware ist fest vorgegeben
  - Fester Befehlssatz
  - Feste CPU-Komponenten
    - Execution Units
    - SIMD-Spezial-Einheiten
    - Cache
    - ...
  - Anwendungsentwicklung wird der HW angepasst
  - Warum nicht Hardware an Anwendung anpassen?

- Adaptive Rechner
  - Hardware ist flexibel
  - Passt sich den vorgegebenen Anwendungen an

Adaptive Rechner

- Problem:
  - Anwendungs-Entwicklung komplex
    - Partitionierung: welche Teile in SW, welche in HW?
    - *Synthetisierbare* HW-Beschreibungen erzeugen
      - Verilog, VHDL
      - Sequentiellen Code parallelisieren
  - Kein kommerzieller Compiler kann dies voll-automatisch und universell
    - HW-Entwickler für HW-Teil nötig ⇨ teuer, zeitaufwändig
    - Daher Adaptive Rechner bislang kaum verbreitet
- Aktuelles Forschungsthema:
  - Automatische Erzeugung von HW-SW-Systemen aus Hochsprachen wie C
  - Compiler *COMRADE*
    - Aus C automatisch HW-SW-Anwendung erstellen

Zukunfts-Perspektiven

- Multi-Core
  - Heute: SPARC: 8 Kerne
  - 2008: Intel: 6 Kerne
  - 2009: AMD: 8 Kerne; SPARC: 16 Kerne
- Multi-Core:
  - Viele gleichartige Kerne?
  - Einige komplexe Kerne +
    viele verschiedene Hilfs-Kerne?
- Programmierung wird komplexer
  - Parallelität
  - Heterogenität (Cell, GPU)
  - Hardware (FPGA)