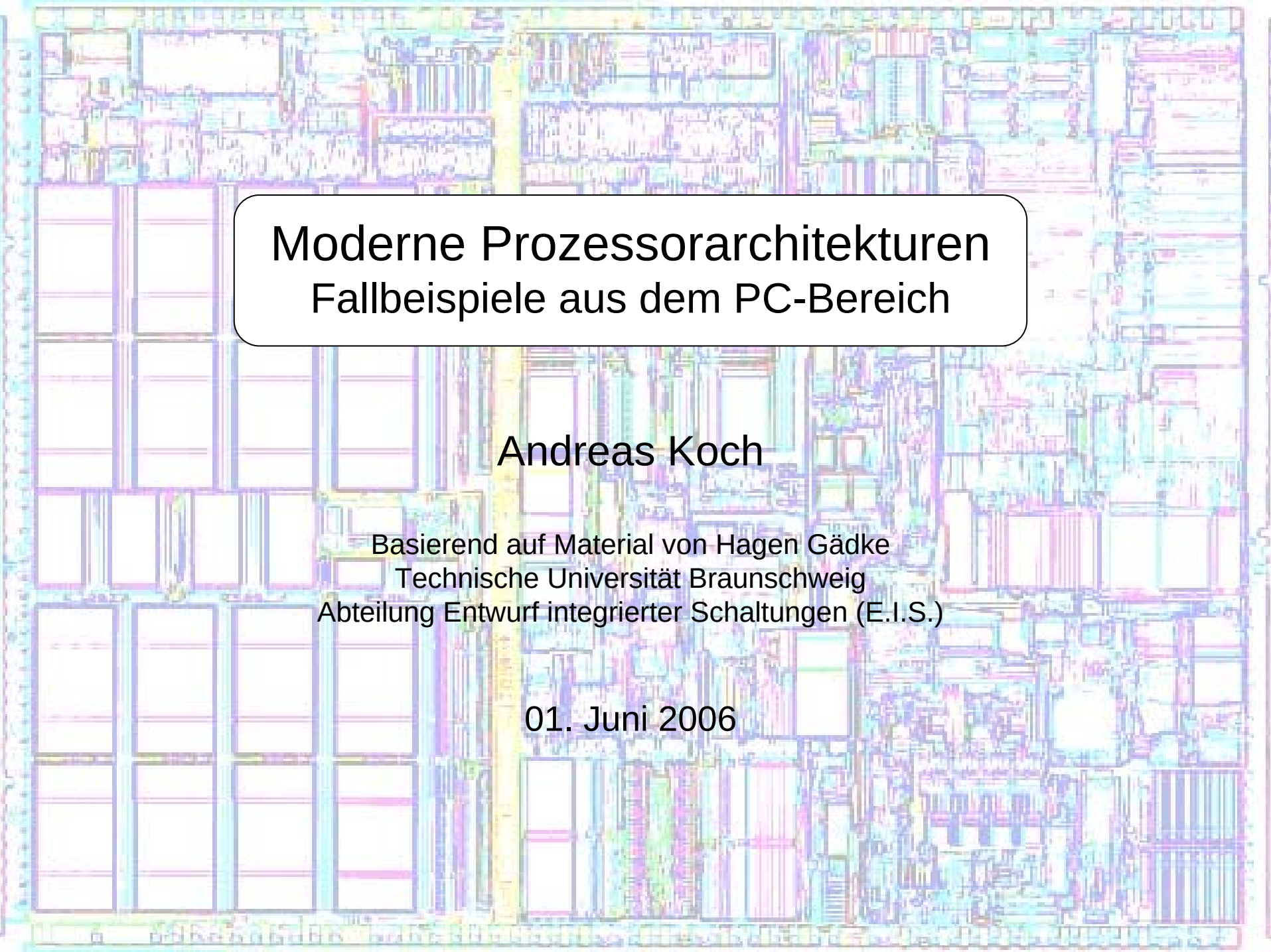


Moderne Prozessorarchitekturen
Fallbeispiele aus dem PC-Bereich

Andreas Koch

Basierend auf Material von Hagen Gädke
Technische Universität Braunschweig
Abteilung Entwurf integrierter Schaltungen (E.I.S.)

01. Juni 2006

Inhalt

+ Mikroarchitektur von PC-Prozessoren (x86)
+ Details zu ausgewählten Komponenten

- Board-Anbindung
- Schaltungstechnik auf Transistorebene
- Fertigungstechnik

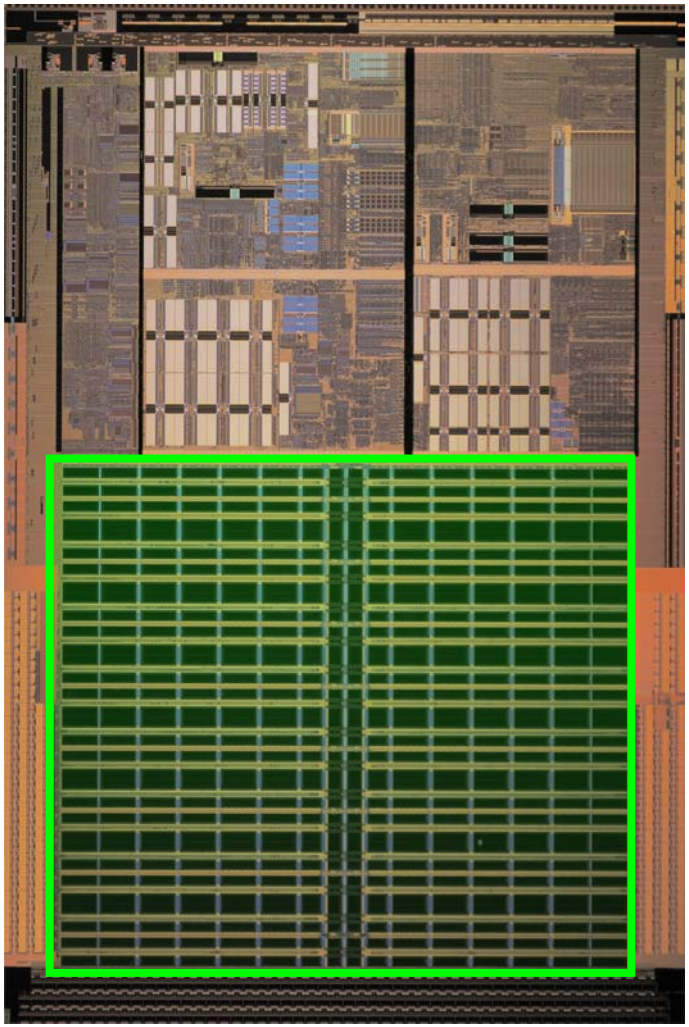
Inhalt

- Grundlagen
  - Cache, virtueller Adressraum, Pipelining, Superskalarität, Hyperthreading, Multi-Core
- Der Intel Pentium 4
  - NetBurst, ALUs
- Der AMD Athlon
  - AMD64, Pipeline, Cache
- Energiespartechniken
  - Techniken, CoreDuo
- Trends
  - Mooresches Gesetz, 4-GHz-Grenze?

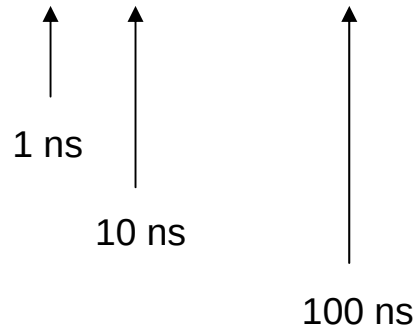
Inhalt

- Grundlagen
  - Cache, virtueller Adressraum, Pipelining, Superskalarität, Hyperthreading, Multi-Core
- Der Intel Pentium 4
  - NetBurst, ALUs
- Der AMD Athlon
  - AMD64, Pipeline, Cache
- Energiespartechniken
  - Techniken, CoreDuo
- Trends
  - Mooresches Gesetz, 4-GHz-Grenze?

Cache



- heute ca. ½ der CPU
- je größer, umso langsamer
  - längere Leitungen
  - # gleichzeitig zu treibender Speicherzellen
- Aufbau einer Speicher-Hierarchie
  - CPU-Reg., L1, L2, (L3,) RAM, Festplatte



  - schneller, billiger

Cache

- Cache-Prinzip: Lokalität
  - *Zeitlich:*
    „Was ich jetzt brauche,
    brauche ich bald wieder“
    - Schleifen: Daten und Instruktionen

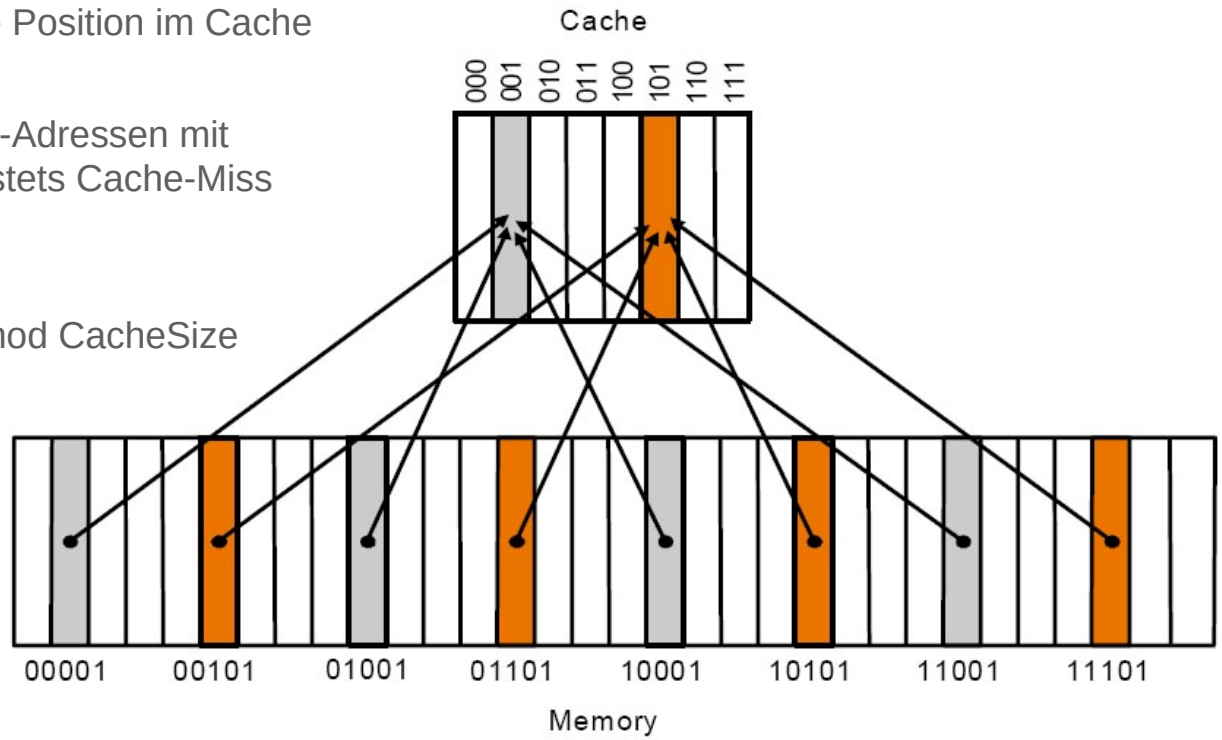
```
for (i = 0; i < n; i++) {
    value = a[i];
    if (value < min) min = value;
    if (value > max) max = value;
}
```

  - *Räumlich:*
    „Wenn ich jetzt MEM[**i**] brauche,
    brauche ich bald MEM[**i+1**]“
    - Instruktionsfolge ohne Verzweigung
    - Arrays: Daten
    - ↙ Cache-Line!

```
for (i = 0; i < n; i++) {
    value = a[i];
    if (value < min) min = value;
    if (value > max) max = value;
}
```

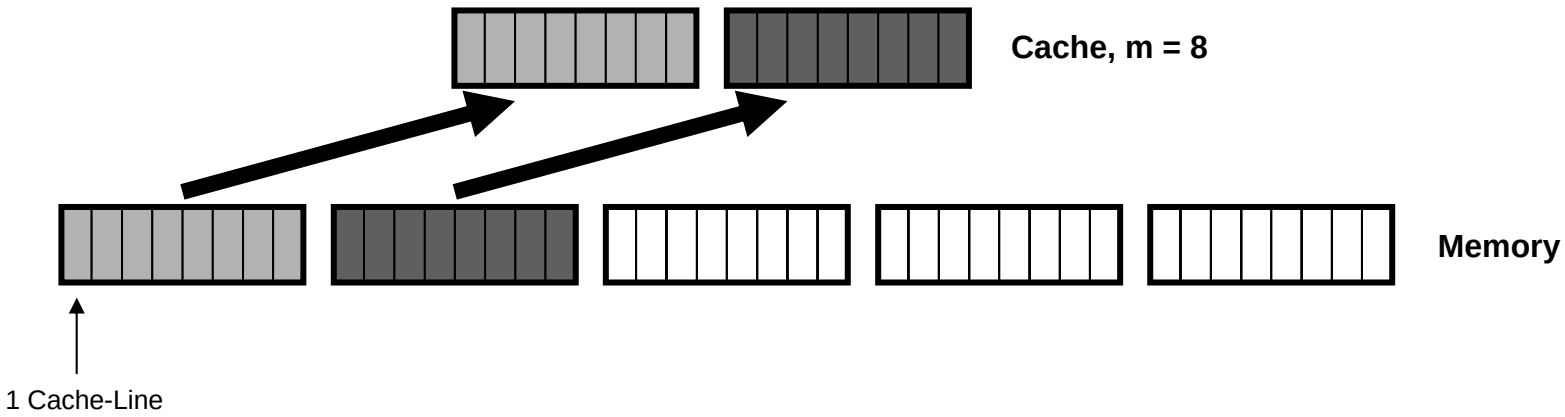
Cache

- Cache-Assoziativität
  - Abbildung von Speicher-Adressen auf Cache-Adressen
    - direct-mapped
      - jeder RAM-Block hat feste Position im Cache
      - Nachteil:
        Zugriffe auf zwei Speicher-Adressen mit gleicher Cache-Adresse: stets Cache-Miss
      - Vorteil:
        einfach zu realisieren:
        CacheAddr := RamAddr mod CacheSize



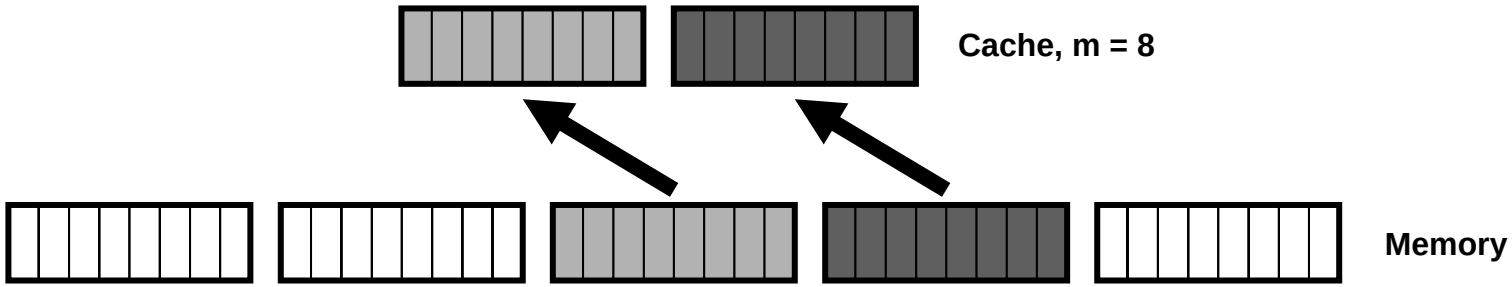
Cache

- Cache-Assoziativität
  - Abbildung von Speicher-Adressen auf Cache-Adressen
  - m-assoziativ
    - Aufteilung in Cache-Blöcke
    - Jeder Block hat m disjunkte Mengen
    - Zuordnung RAM-Block -> Cache-Block fest,



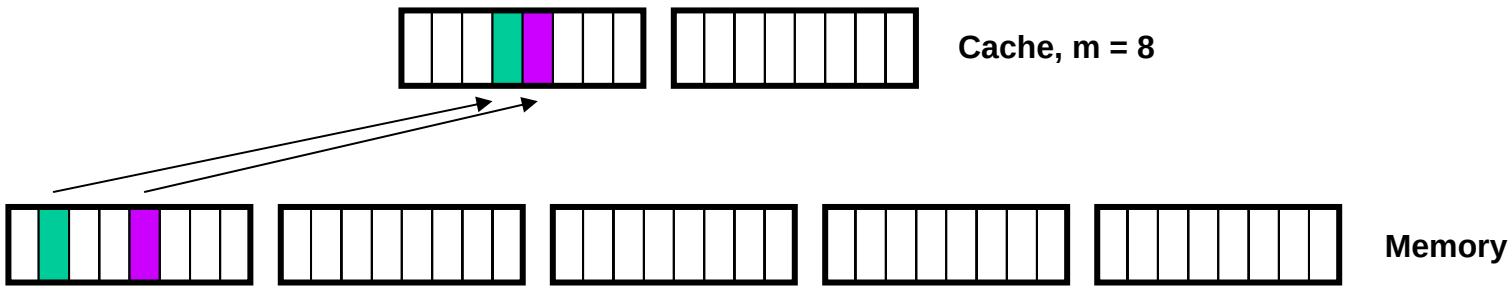
Cache

- Cache-Assoziativität
  - ↙ Abbildung von Speicher-Adressen auf Cache-Adressen
  - m-assoziativ
    - Aufteilung in Cache-Blöcke
    - Jeder Block hat m disjunkte Mengen
    - Zuordnung RAM-Block -> Cache-Block fest,



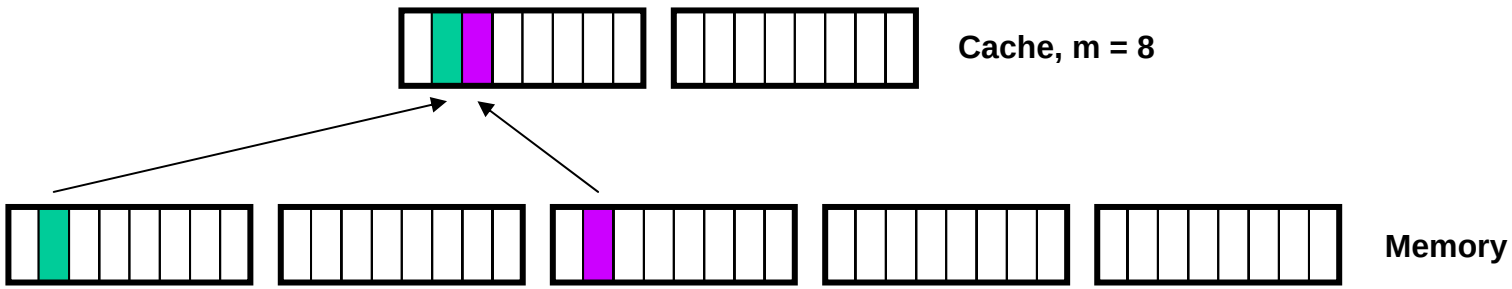
Cache

- Cache-Assoziativität
  - Abbildung von Speicher-Adressen auf Cache-Adressen
  - m-assoziativ
    - Aufteilung in Cache-Blöcke
    - Jeder Block hat m disjunkte Mengen
    - Zuordnung RAM-Block -> Cache-Block fest, aber innerhalb des Blockes beliebige Platzierung



Cache

- Cache-Assoziativität
  - Abbildung von Speicher-Adressen auf Cache-Adressen
  - m-assoziativ
    - Aufteilung in Cache-Blöcke
    - Jeder Block hat m disjunkte Mengen
    - Zuordnung RAM-Block -> Cache-Block fest, aber innerhalb des Blockes beliebige Platzierung



    - behebt Nachteil von direct-mapped Caches
    - erfordert m parallele (Hardware-)Vergleicher (Lesen vom Cache) sowie Einsortierungs-Mechanismus (Schreiben in den Cache)

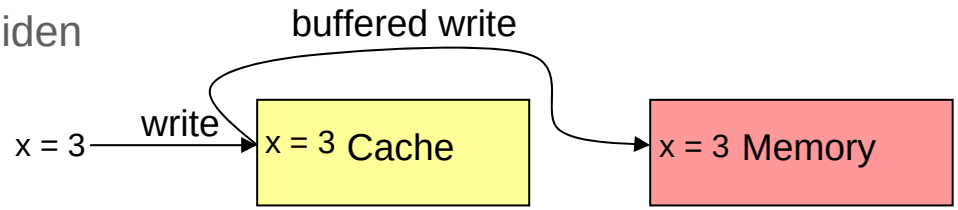
Cache

- Cache-Assoziativität
  - ↙ Abbildung von Speicher-Adressen auf Cache-Adressen
  - direct-mapped (m-assoziativ mit m = 1)
  - m-assoziativ
  - voll-assoziativ (m-assoziativ mit m = Cachegröße)

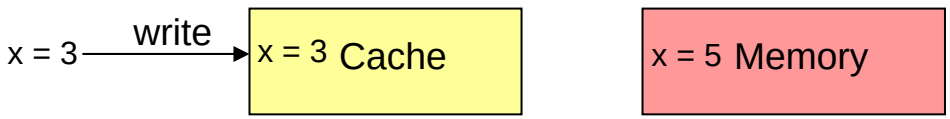
  ↙ heute häufig: 8-fach assoziativ (z. B. Pentium 4: L2-Cache)

Cache

- Cache-Zugriffsarten
  - Schreibzugriff, Block ist im Cache vorhanden:
    - **write-through**
      Nach dem Schreiben sofort an nächste Hierarchie-Ebene weiterreichen
      - Write-Buffers, um Stalls zu vermeiden

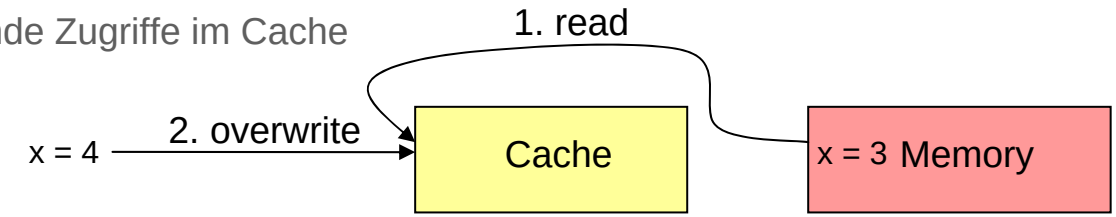


    - **write-back**
      Weiterreichen erst, wenn Verdrängung durch Lese-Miss
      - verbraucht weniger Speicher-Bandbreite
      - falls mehrere Teilnehmer auf Speicher zugreifen: Cache-Kohärenz-Protokoll nötig

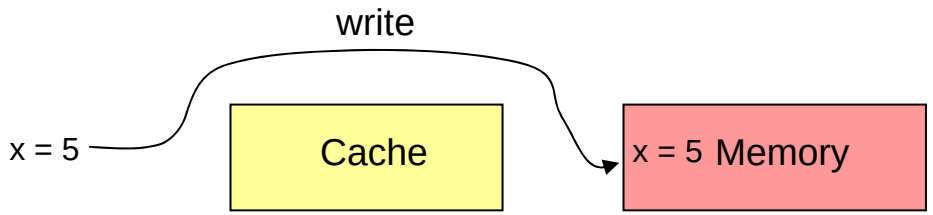


Cache

- Cache-Zugriffsarten
  - Schreibzugriff, Block ist *nicht* im Cache vorhanden:
    - **write-allocate**
      Lese Block aus Memory, dann im Cache überschreiben
      - Vorteil: Datum liegt für folgende Zugriffe im Cache



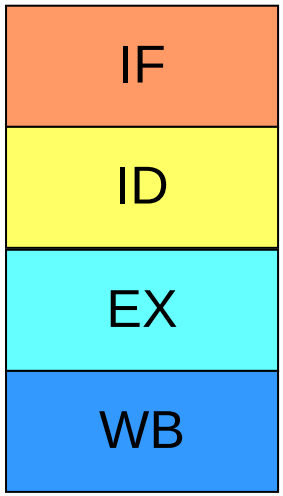
    - **non-write-allocate**
      Block wird am Cache vorbei in den Hauptspeicher geschrieben
      - Vorteil: andere (evtl. wichtigere) Blöcke werden nicht aus dem Cache verdrängt



Virtueller Adressraum

- virtueller Adressraum größer als physikalischer
- physikalisch getrennte Speicherseiten können im virtuellen Adressraum logisch zusammenhängen
  - Anlegen großer Arrays bei stark fragmentiertem Speicher
- Speicherschutz
  - Sperrung des Ausführens / Überschreibens von Speicherbereichen
  - z. B. darf ein Programm immer nur in seinen "eigenen Bereich" schreiben
- Swapping: transparente Auslagerung auf Festplatte (LRU)
  - Betriebssystem: 3 GB Arbeitsspeicher, obwohl nur 256 MB RAM
- Memory Management Unit
  - übersetzt virtuelle in physikalische Adressen
  - TLB (Translation Lookaside Buffer)
    Cache der Seitentabelle (speichert Adressen-Übersetzungen)

Pipelining



- Die Lehrbuch-Pipeline
  - Instruction Fetch: Laden der Instruktionen und
  - Instruction Decode: Decodieren,
  - Execute: Ausführen der Befehle,
  - Write-Back: Rückschreiben der Ergebnisse

- höhere Taktraten

  => kleinere (dadurch mehrere) Pipeline-Stufen

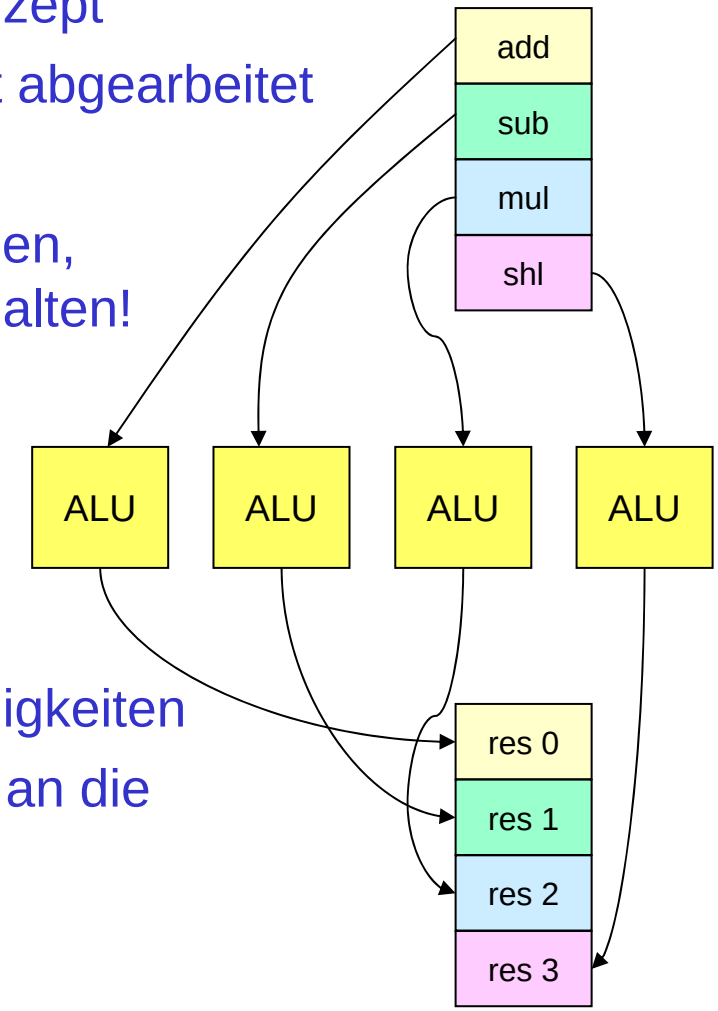
  => „Super-Pipeline“

  => Sprungziel-Vorhersage gewinnt an Bedeutung

  (Linearität => spekulativ arbeiten => potentiell Fehler)

Superskalarität

- genau entgegengesetzt zum Pipeline-Konzept
- mehrere Befehle werden zur gleichen Zeit abgearbeitet
- mehrere parallele Recheneinheiten
- intern parallele Ausführung der Instruktionen, aber nach außen keine Änderung im Verhalten!



- dynamische Beachtung von Datenabhängigkeiten
- dynamische Zuweisung von Instruktionen an die Recheneinheiten
- „Out-of-Order Execution“

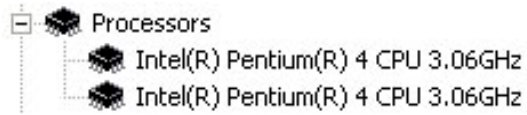
↙ hoher Schaltungsaufwand

HyperThreading

- Grundlage
  - Recheneinheiten der CPU werden wegen Pipeline-Stalls nicht optimal genutzt
- Idee
  - aktueller Thread: Pipeline-Stall => anderer Thread rechnet
- Realisierung
  - Duplizierung weniger Komponenten (z. B. Registersatz)
  - Aufteilung weniger Komponenten
  - gemeinsame Nutzung aller größeren Komponenten (z. B. Cache)

↙ intern: 1 aufgebohrter Prozessor

↙ extern: 2 „logische“ Prozessoren

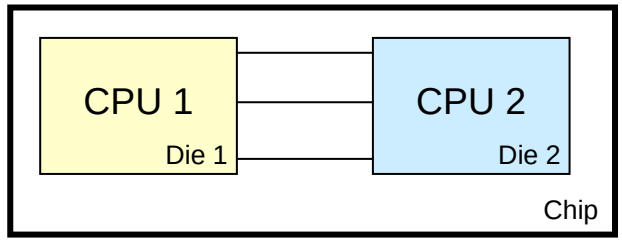


HyperThreading

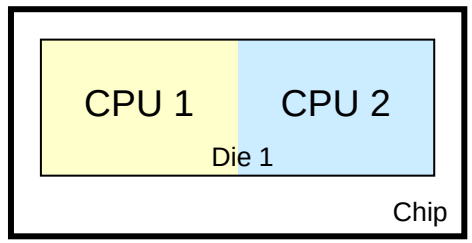
- Beschleunigung des Gesamtsystems mehrerer Threads
  - bei nur 1 Thread allein wird nichts beschleunigt!
- Einsatz besonders sinnvoll in Systemen,
  auf denen viele unabhängige Prozesse / Threads gleichzeitig laufen
  - Server
    - mehrere User
    - Server-Software ist auf mehrere Prozessoren ausgelegt
    - ↙ viele nebenläufige Prozesse
  - wurde zuerst eingeführt im Intel Xeon, danach auch im Pentium 4
    - im Desktop-Bereich Beschleunigung z. B. in Video-Bearbeitung,
    - nicht aber in Office-Anwendungen / Spielen

Multi-Core

- nicht nur mehrere logische Prozessoren, sondern tatsächlich mehrere CPUs in 1 Chip
- Server: schon länger (z. B. 2001: IBM Power 4)
- Desktop: seit 2005
- mögliche Ausführungen:
  - 2 getrennte Dies verdrahten und in 1 Chip verpacken
    - „Verwerten“ herkömmlicher Single-Core-CPUs
    - anfängliche Notlösung (z. B. Intel Smithfield)

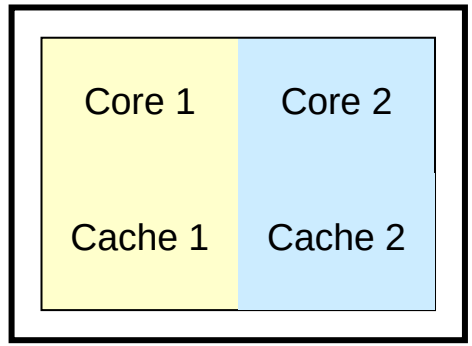


  - 2 CPUs auf 1 Die
    - kompakter
    - in großen Stückzahlen günstiger
    - State-of-the-Art

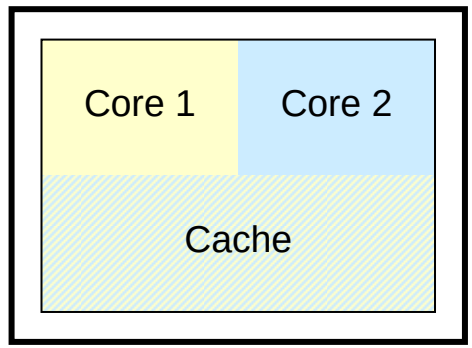


Multi-Core

- Cache-Struktur
  - getrennte Caches
    - einfaches Design:
      Kombination zweier herkömmlicher CPUs



  - gemeinsamer Cache
    - wenn Core 1 idle
      => Core 2 darf gesamten Cache nutzen
    - Schnellere Kommunikation zwischen den CPU-Cores
    - dynamische Speicher-Allozierung



Inhalt

- Grundlagen
  - Cache, virtueller Adressraum, Pipelining, Superskalarität, Hyperthreading, Multi-Core
- Der Intel Pentium 4
  - NetBurst, ALUs
- Der AMD Athlon
  - AMD64, Pipeline, Cache
- Energiespartechniken
  - Techniken, CoreDuo
- Trends
  - Mooresches Gesetz, 4-GHz-Grenze?

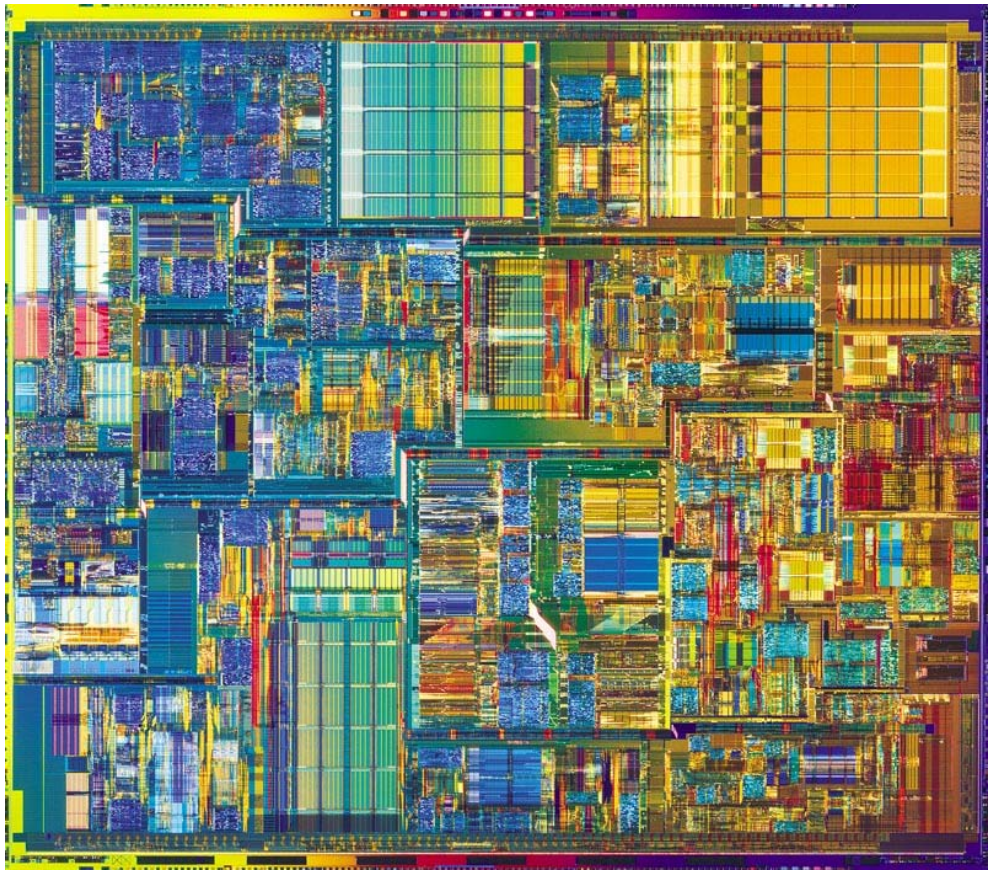
Der Intel Pentium 4

- Eingeführt: 2000
- NetBurst-Mikroarchitektur
- tiefe „Hyper-Pipeline“
  - optimiert für hohe Taktfrequenzen

- Modell-Geschichte
  - 2000: Willamette, 1,5 GHz, 180 nm
  - 2002: Northwood, 3,0 GHz, 130 nm (Hyper-Threading)
  - 2004: Prescott, 3,4 GHz, 90 nm (Pipeline-Verlängerung)
  - 2006: Cedar Mill, 3,8 GHz, 65 nm

- *Dual-Core:*
  - 2005: Smithfield (2x Prescott), 3,2 GHz, 90 nm
  - 2006: Presler (2x Cedar Mill), 3,7 GHz, 65 nm

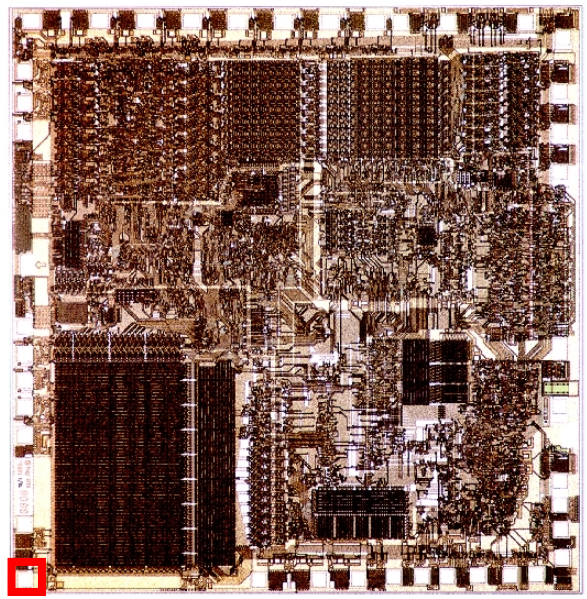
Der Chip



Pentium 4

2000, 180 nm, 88 mm², 1.5 GHz

42 Mio. Transistoren



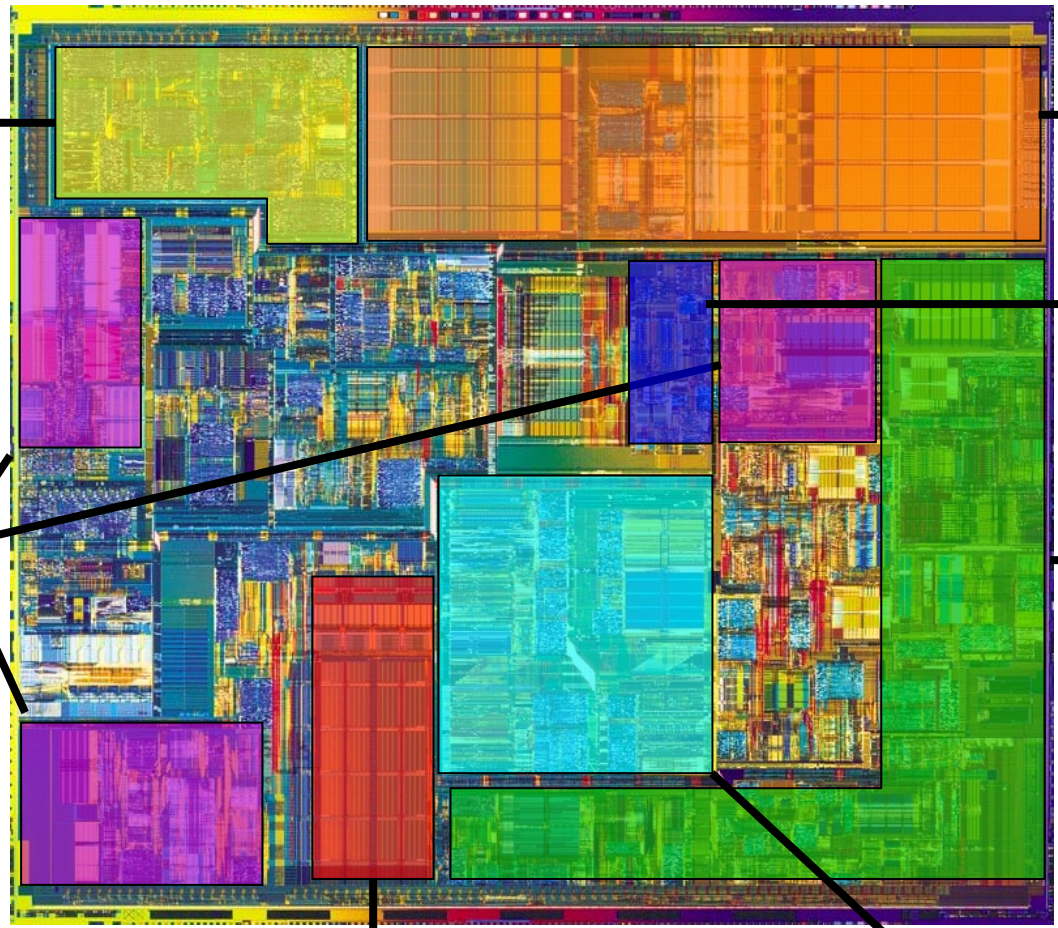
(180 nm)

8086

1978, 3 µm, 33 mm², 5 MHz

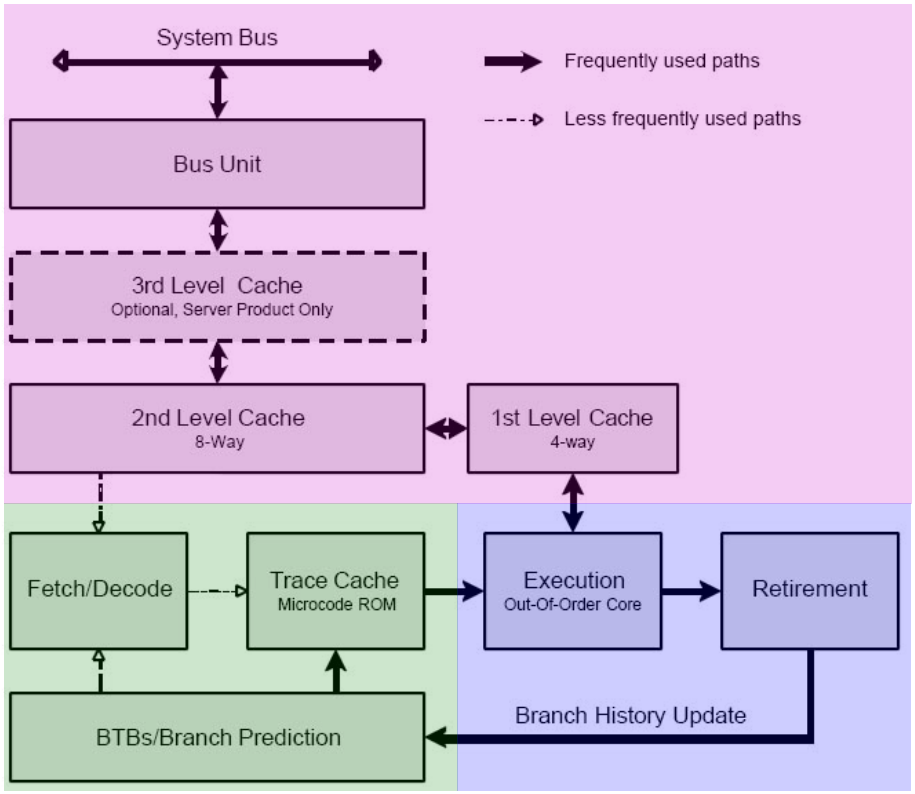
29.000 Transistoren

Strukturen auf dem Chip



Die NetBurst-Architektur

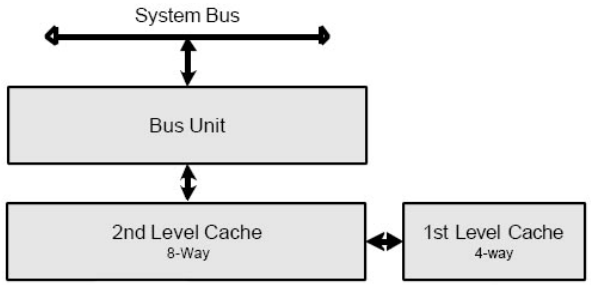
Memory-Subsystem



Front-End

Execution-Pipeline

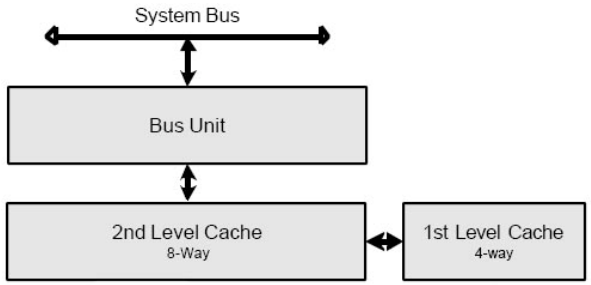
NetBurst – Memory Subsystem



konkrete Daten:
Pentium Extreme Edition

- FSB: Datendurchsatz bis 8500 MB/s
  - 266 MHz, 64 Bit, 4 Datenworte pro Takt, „quad-pumped“ (ähnlich AGP4x)
  - 8 Byte * 266 million/s * 4 = 8500 MB/s (für 1 MB = 1000 KB; eigentlich: 8118 MB/s)
    - Pentium 3: 1066 MB/s
- 2 MB L2-Cache
  - 8-fach assoziativ, 7 Takte Latenz
  - „Advanced Transfer Cache“: Hardware-Data-Prefetch-Mechanismus
    - von der Vergangenheit auf die Zukunft schließen
    - ↙ weniger Cache-Misses

NetBurst – Memory Subsystem

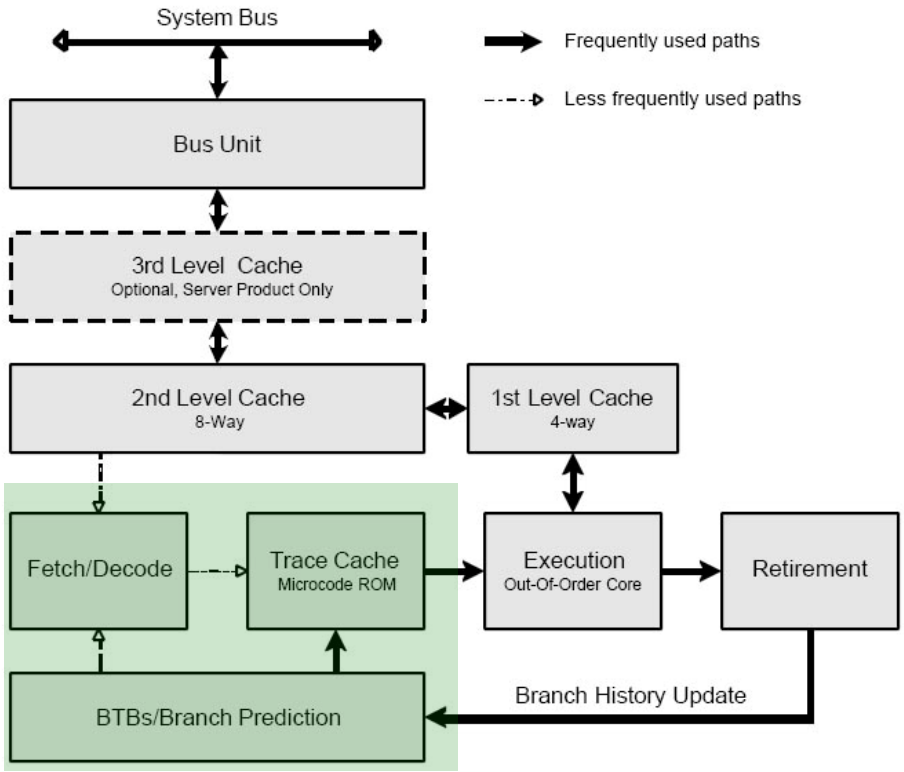


konkrete Daten:
Pentium Extreme Edition

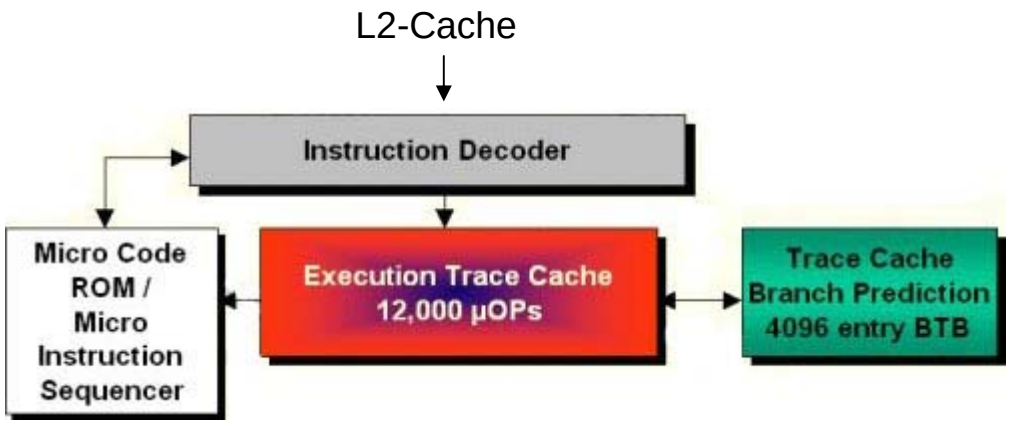
- 16 KB L1-*Data*-Cache
  - 8-fach assoziativ (seit Prescott, vorher 4-fach)
  - 2 Takte Latenz
  - Instruktionen: Trace-Cache, später

NetBurst – Front-End

Front-End

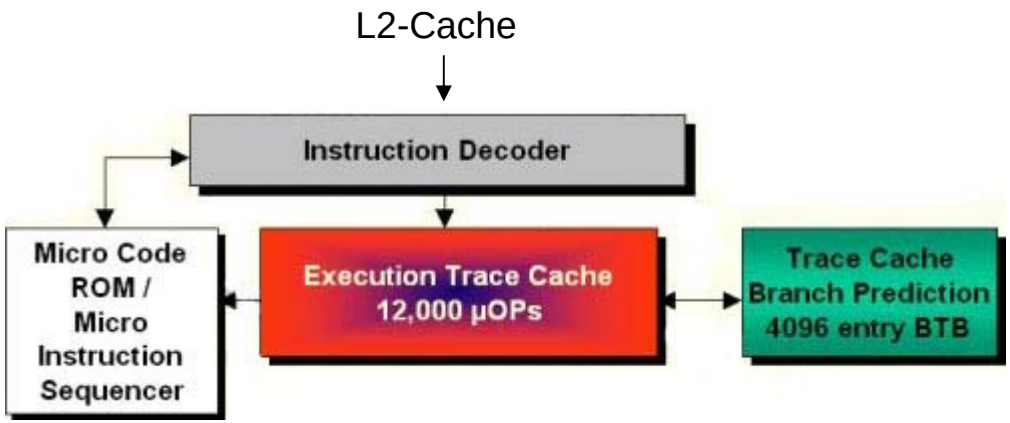


NetBurst – Front-End: Instruction Decoding



- umwandeln der (CISC) x86-Instruktionen in (RISC) „µOPs“
- Instruction Decoder
  - lädt x86-Instruktionen aus L2-Cache
  - erzeugt µOPs (feste Größe, schnellere Bearbeitung in der Pipeline)
    - einfache x86-OPs: 1 µOP, wird vom ID erzeugt (ADD, XOR)
    - komplexere x86-OPs: mehrere µOPs, werden vom ID erzeugt (DIV, MUL)
    - sehr komplexe x86-OPs: >100 µOPs, werden aus Micro Code ROM geladen (SIN, COS)
  - Umwandlungsrate: max. 1 x86-Befehl pro Takt

NetBurst – Front-End: Trace Cache

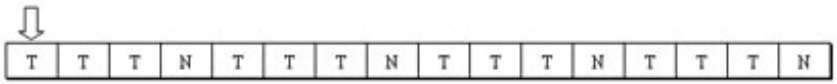


- Execution Trace Cache ersetzt L1-Instruction-Cache
  - speichert Folgen von µOPs (Traces) anstatt x86-Instruktionen
  - Prinzip: für‘s nächste Mal lieber das Ergebnis der Decodierung speichern anstatt es jedes Mal erneut auszurechnen!
  - Platz für 12.000 µOPs, sonst keine Größenangabe
  - 3 µOPs pro Takt werden vom Trace Cache an die Pipeline weitergerreicht

NetBurst – Front-End: Sprungvorhersage, BTB

Trace Cache
Branch Prediction
4096 entry BTB

- Branch Target Buffer, BTB
  - speichert für jeden Sprungbefehl die Zielsprungadresse,
  - zusammen mit 16 Bit History
    - „Wurde der Sprung die letzten 16 Male jeweils durchgeführt oder nicht?“



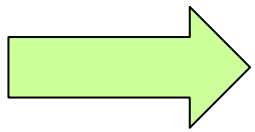
  - History erlaubt dynamische Sprungvorhersage:
    - „Wird der Sprung das nächste Mal durchgeführt?“
    - Intel verrät keine Details

  - falls noch keine History im BTB vorhanden: wende statische Vorhersage an
    - Rücksprünge ja (z. B. do-while-loop), Vorwärtssprünge nein (z. B. while-loop)
    - 2 Beispiele folgen

Sprungvorhersage (statisch): Beispiel 1

- Beispiel 1: Rückwärtssprung „jg“ wird vorhergesagt

```
//
// C code
//
do
{
    …
    data--;
}
while (data > 0);
```



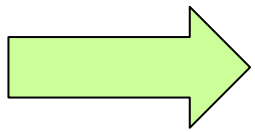
```
;
; Assembly code
;
    mov    eax, data
DoWhileLoop:
    …
    sub    eax, 1
    cmp    eax, 0
    jg     DoWhileLoop
EndDoWhile:
```

Rücksprung: ja!

Sprungvorhersage (statisch): Beispiel 2

- Beispiel 2: Vorwärtssprung „jle“ wird *nicht* vorhergesagt

```
//
// C code
//
while (data > 0)
{
    …
    data--;
}
```



```
;
; Assembly code
;
 mov    eax, data
WhileLoop:
    cmp    eax, 0
    jle    EndWhile
    …
    sub    eax, 1
    jmp    WhileLoop
EndWhile:
```

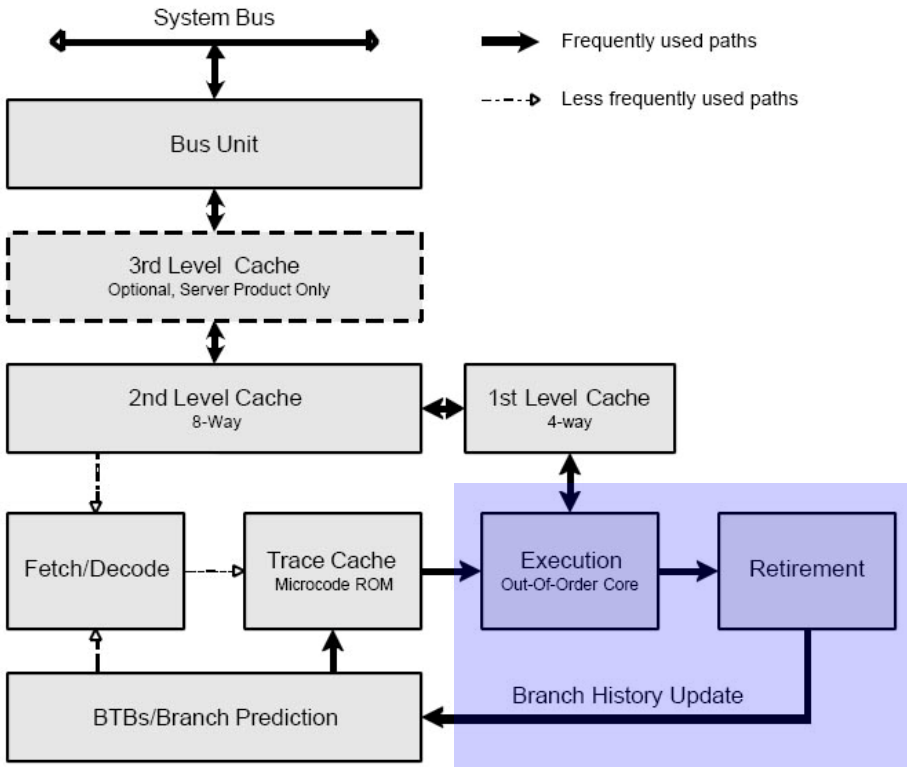
Vorwärts-sprung: nein!

Sprungvorhersage: Branch Hints, Abschluss

- Branch Hints:
  - Zusätze für x86-Sprung-Befehle
  - Programmierer kann die statische Vorhersage umkehren
  - Nachteil: Code wird länger

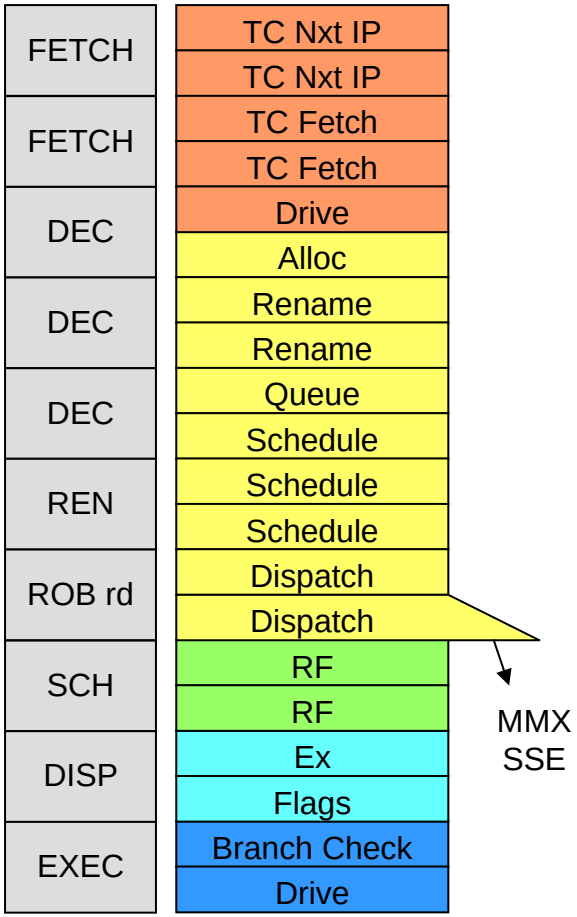
- Abschluss
  - Intel:
    „Branch-Prediction des P4 korrigiert 1/3 aller falschen Vorhersagen des P3“
  - Branch-Predictor P4: 8x so groß wie beim P3
  - Intel:
    „Branch-Prediction des P4 ist besser als alle anderen Sprung-Vorhersage-Algorithmen“

NetBurst – Pipeline



Execution-Pipeline

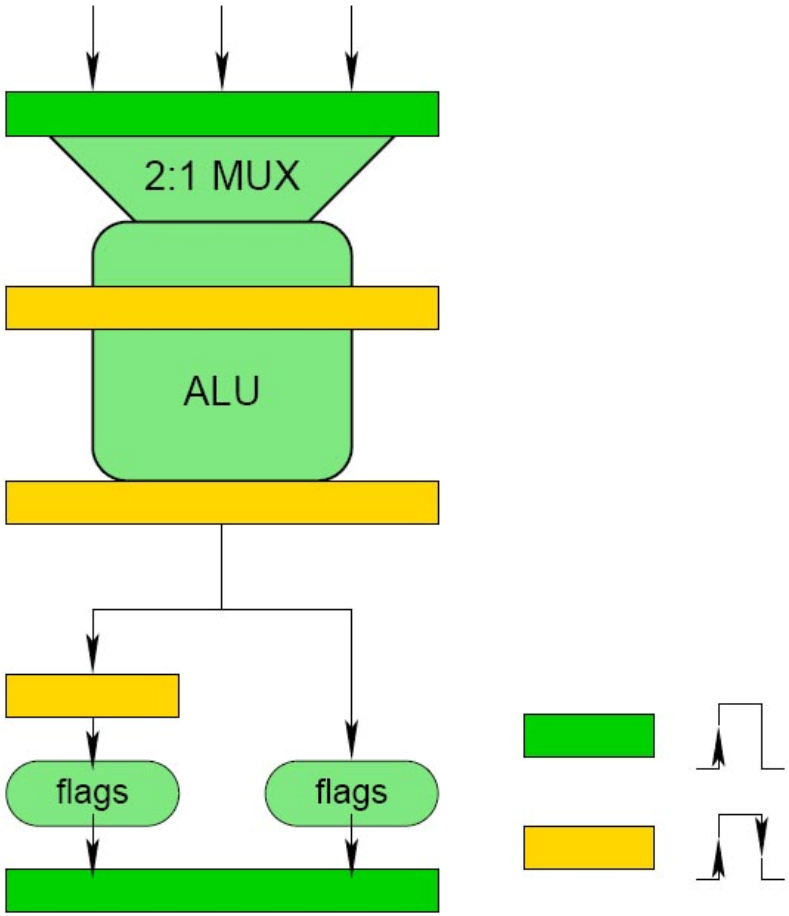
NetBurst – 20-stufige Hyper-Pipeline



Pentium III

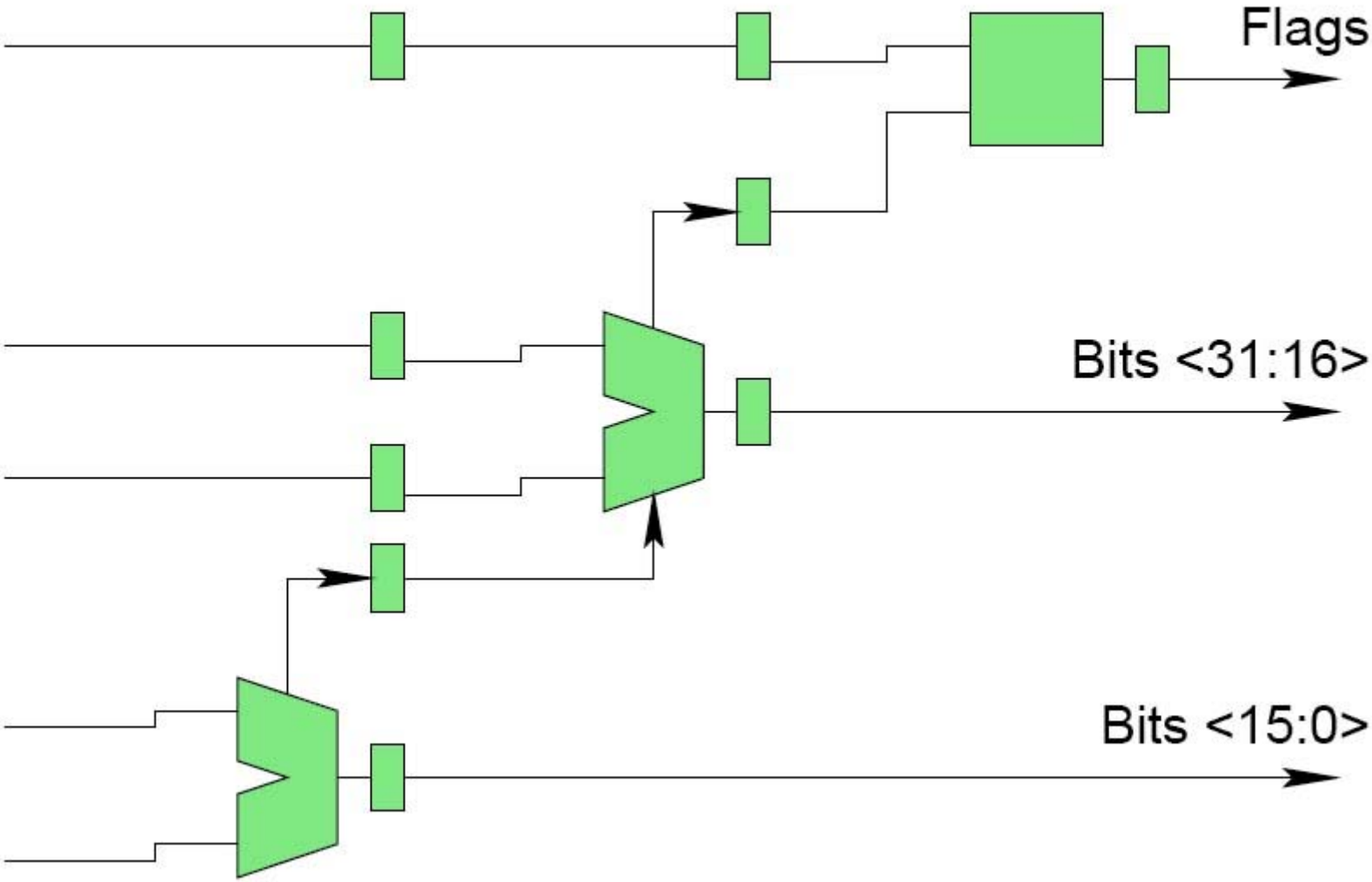
- Instruction Fetch
  - holt µOPs aus dem Trace-Cache
- Instruction Decode
  - Buffer und Register Allocation
  - x86-Register-Renaming
  - Queueing und Scheduling für Out-of-Order-Execution
  - Dispatch: µOPs auf die Execution Units verteilen
    - Abzweigung MMX / SSE
- Operand Fetch
  - Read Register File
- Execute
  - Ausführung in 7 parallelen Einheiten (*Execution-Units*),
  - 1 „normale“ ALU, 2 double pumped ALUs, Load / Store
  - Ergebnisse werden ins *Future Register File* geschrieben
  - Flags setzen (zero, negative, …)
- Write-Back
  - Branch-Check: Vorhersage OK?
  - Branch Check OK: Future Register File => Register File

Double Pumped ALUs



↙ akzeptiert in jedem Takt 2 µOPs!

Double Pumped ALUs



Pentium 4: Modellpflege

- Änderungen vom ersten Pentium 4 bis zum aktuellen Cedar Mill
  - Taktrate: 1,5 GHz -> 3,8 GHz
  - Strukturgröße: 180 nm -> 65 nm (Pentium Extreme Edition (dual) + CoreDuo)
  - Anzahl Transistoren: 42 Mio. -> 188 Mio.
    - Vergleich NVidia G80 GPU: 500 Mio.!
  - Die-Größe: 170 mm² -> 88 mm²
  - FSB: 400 (100 * 4) MHz -> 1066 (266 * 4) MHz
  - Core-Spannung: 1,75 V -> 1,4 V
  - Leistung: ca. 70 W -> 130 W
  - L2-Cache: 256 KB -> 2 MB
  - L1-Data-Cache: 8 KB -> 16 KB
  - Trace-Cache: 12 kµOPs -> 16 kµOPs
  - Gleitkomma-Beschl.: MMX, SSE -> MMX, SSE, SSE2, SSE3
  - Pipeline: 20 -> 31 Stufen
  - verbesserte Sprungvorhersage, Vergrößerung des BTB

Pentium 4: Weitere Features

- Execute Disable Bit
  - Markierung von Daten-Speicheradressen als „nicht ausführbar“
  - ↙ verringert Risiko durch Buffer-Overflow-Attacken
- Intel Virtualization Technology
  - Beschleunigung für VM-Software
- EM64T = Extended Memory 64
  - Erweiterung von NetBurst um die Ausführung der 64-bittigen AMD64-Befehle
  - erlaubt mehr als 4 GB RAM
- Hyperthreading
- Dual-Core
  - aber noch getrennte L2-Caches
  - gemeinsamer Cache erst ab Core Duo

Pentium 4: Zusammenfassung

- Pipeline hat sehr viele Stufen, *damit* hohe Taktfrequenzen möglich sind

↙ hohe Kosten bei einem Branch Target Miss

↙ sehr aufwendige Sprungzielvorhersage

- Trace Cache: L1-Instruction-Cache für µOPs (x86-Instr. vordecodiert)
- sehr kleiner L1-Data-Cache für möglichst schnellen Zugriff

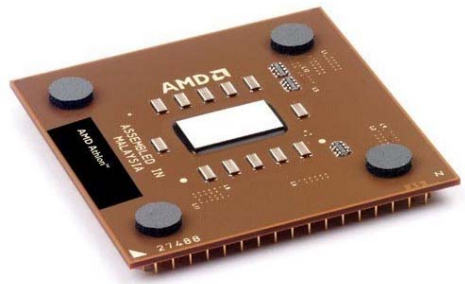
Inhalt

- Grundlagen
  - Cache, virtueller Adressraum, Pipelining, Superskalarität, Hyperthreading, Multi-Core
- Der Intel Pentium 4
  - NetBurst, ALUs
- Der AMD Athlon
  - AMD64, Pipeline, Cache
- Energiespartechniken
  - Techniken, CoreDuo
- Trends
  - Mooresches Gesetz, 4-GHz-Grenze?

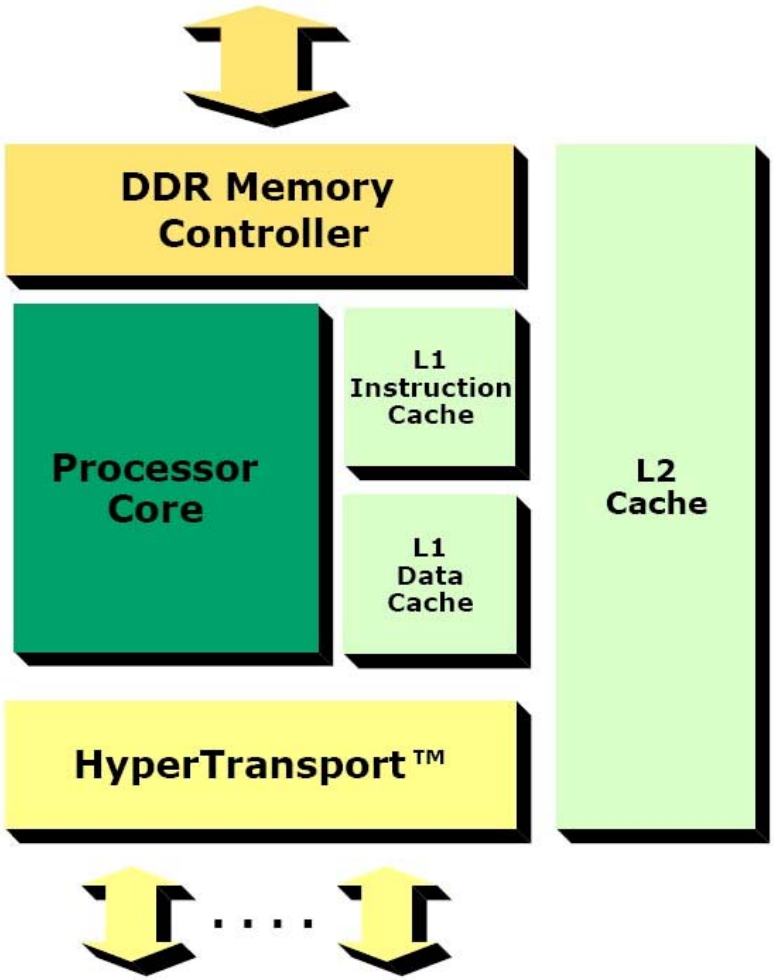
Der AMD Athlon XP / 64

- Eingeführt: 2001, Athlon-Nachfolger
- kürzere Pipeline als Pentium 4
  - 10 (int) bzw. 15 (float) Stufen
- Superskalar, Out-of-Order Execution
- Quantispeed-Rating
  - ApplicationPerformance = GHz x IPC
- Einige Modelle
  - 2001: Palomino, 1,6 GHz, 1900+, 180 nm
  - 2003: Clawhammer, 2,0 GHz, 3200+, 130 nm, 64 Bit, RAM-Controller, HT
  - 2005: San Diego, 2,8 GHz, - 90 nm

  - *Dual-Core:*
  - 2005: Manchester, 2,4 GHz, 4600+, 90 nm, (2x 512 KB L2)
  - 2005: Toledo, 2,4 GHz, 4800+, 90 nm, 2x 1 MB L2
  - 2006: Toledo, 2,6 GHz, - 90 nm

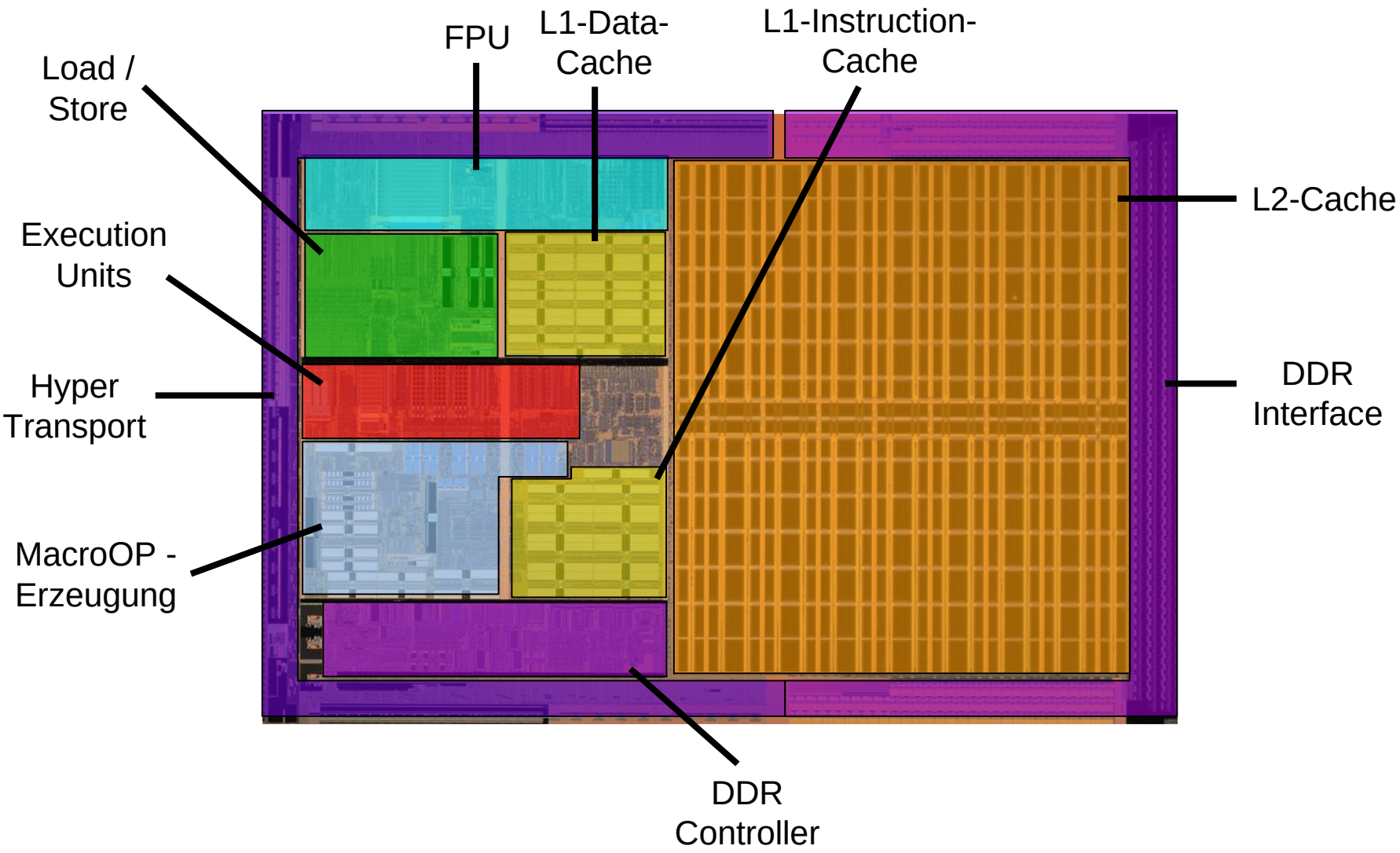


Blockbild Athlon 64

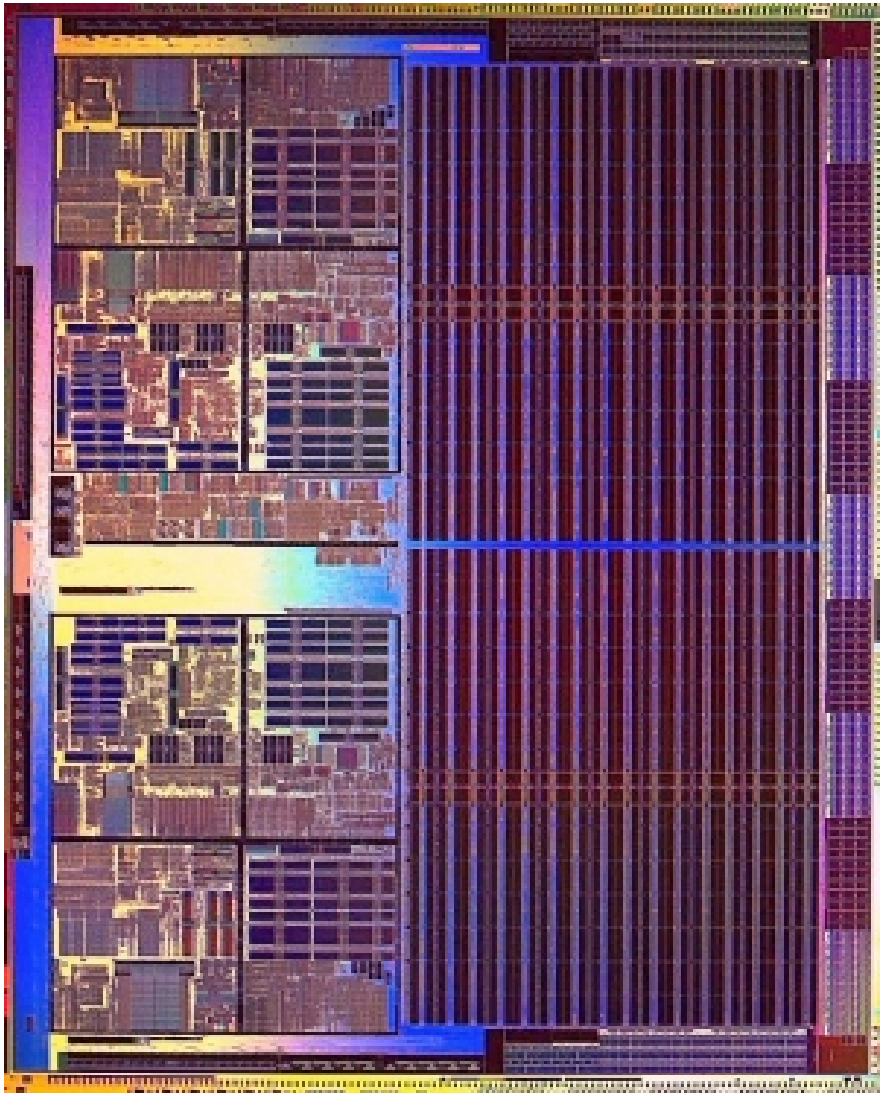


- L1-Cache
  - 64 KB Data + 64 KB Instruction
  - 2-fach assoziativ
- L2-Cache
  - 1 MB
  - 16-fach assoziativ
- DDR-Controller
  - 64: 3,2 GB/s (200 MHz, DDR, 64 Bit, 2 Takte, dualchannel)
  - 64 FX: 6,4 GB/s (128 Bit)
  - \+ weniger Latenz, mehr Bandbreite
  - \- weniger Modularität
- HyperTransport
  - 4 GB/s (1 GHz, 16 Bit, dualchannel)
  - für Peripherie, z. B. PCI-Express
  - Opteron: Verbindung mehrerer CPUs

Strukturen auf dem Chip

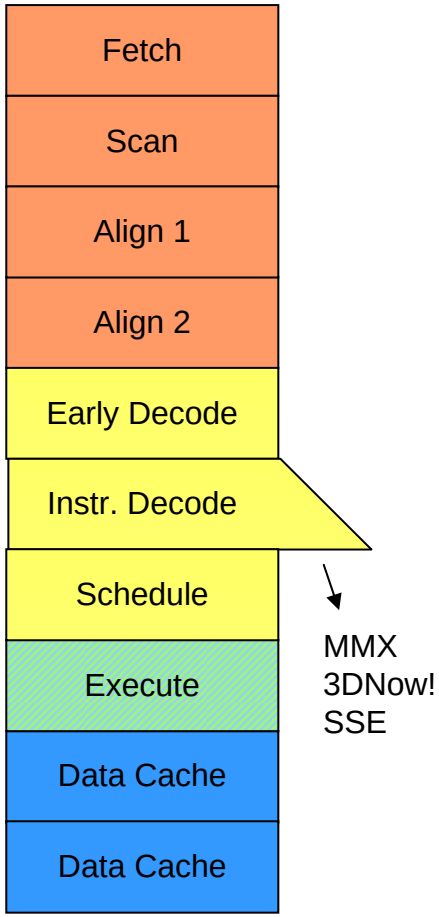


ZUm Vergleich: Doppelkern-Athlon



- Auf den ersten Blick:
  - Copy & Paste des Einzelkerns

Pipeline



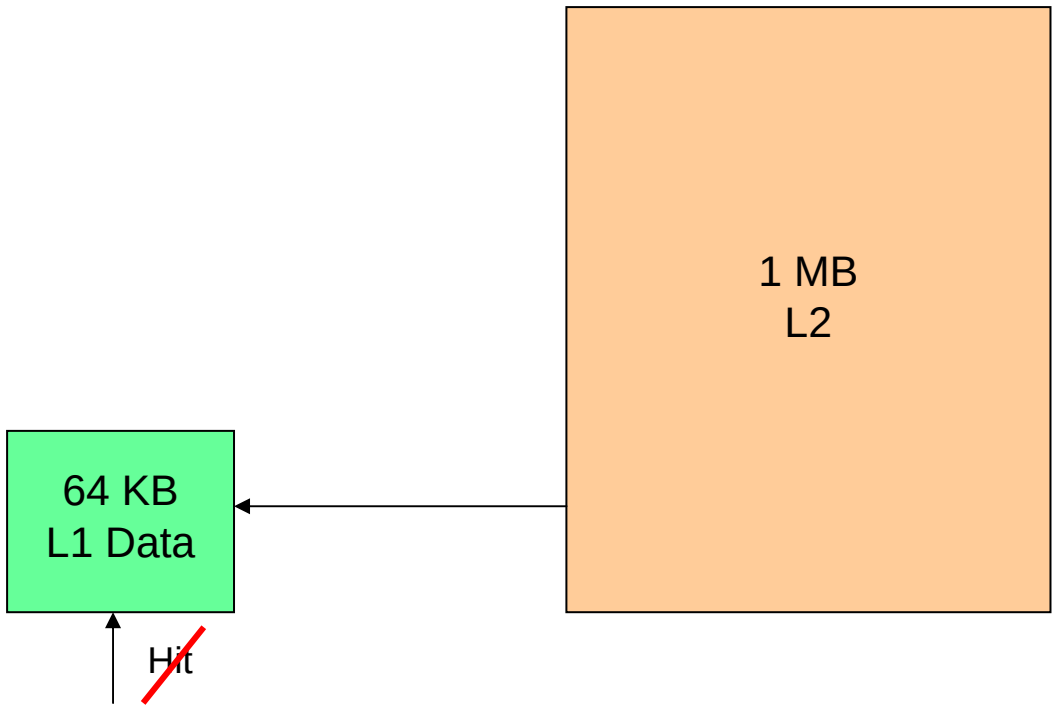
- Instruction Fetch
  - x86-Instruktionen aus dem L1-Cache holen, die verschieden langen Befehle ausrichten
- Decode
  - Early Decode in MacroOPs (3 pro Takt)
  - Instruction Decode:
    - Aufteilung nach **Int** bzw. **Float / MMX / …**
    - bis zu 72 Integer-MacroOPs werden in Instruction Control Unit gepuffert
  - Schedule: Aufteilung auf die parallelen Ausführungseinheiten
- Execute
  - Ausführung in 9 parallelen Einheiten (Execution-Units)
  - darunter 3 Integer-ALUs
  - Aufteilung der MacroOPs in ALU-OPs und Load / Store
- Write-Back
  - Ordnen der Results (Re-Order-Buffer)
  - Rückschreiben in den L1-Cache
  - L1-Cache:
    Lesen + Schreiben durch ausgegliederte Load/Store-Unit
    - Puffert bei Write 64 Byte, Übertragung an Cache dann als 1 Burst

MacroOPs

- Erzeugung von MacroOPs aus x86-OPs in Early-Decode-Phase
- feste Länge, ähnlich Intel µOPs, aber
- etwas komplexer als µOP
  - 1 MacroOP = ca. 2 µOPs
  - 1 MacroOP = 1 ALU-OP + 1 Load/Store
- komplexe x86-Instruktionen werden auch hier mittels ROM in viele MacroOPs zerlegt

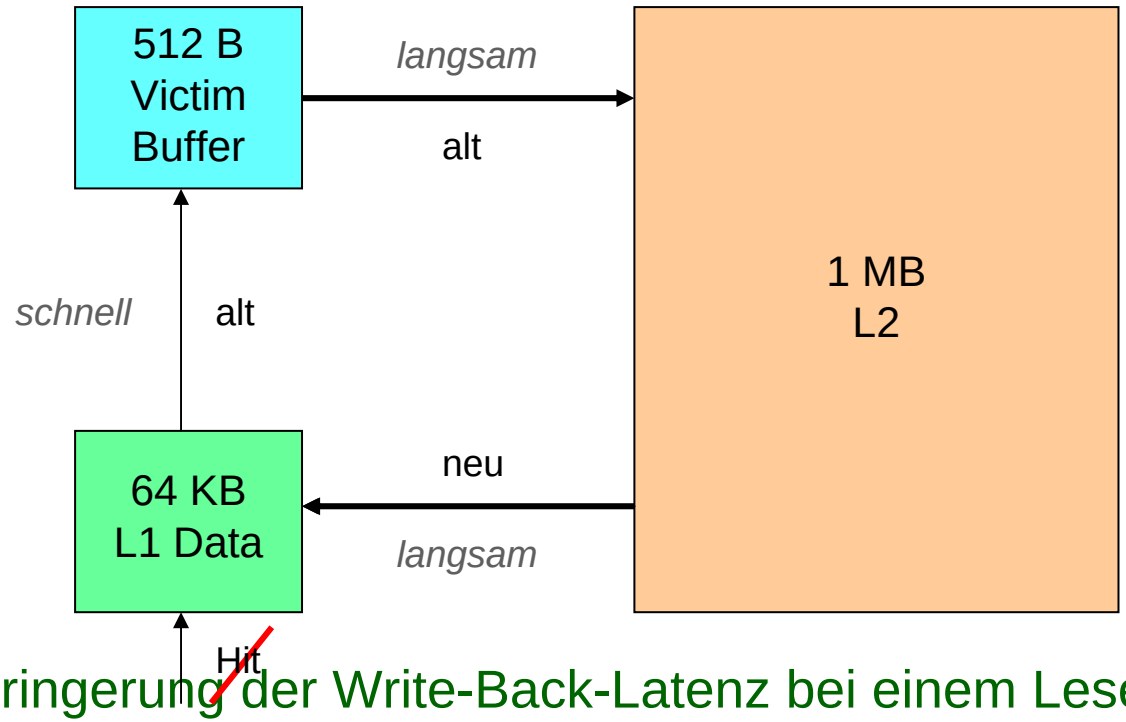
Der Cache des Athlon

- L1-Data-Cache: Write-Back, nicht Write-Through (P4)
  - L1- und L2-Cache ergänzen sich (keine Redundanz)



Der Cache des Athlon

- L1-Data-Cache: Copy-Back, nicht Write-Through (P4)
  - L1- und L2-Cache ergänzen sich (keine Redundanz)
- Victim-Buffer zum Rückschreiben in den L2-Cache



↙ Verringerung der Write-Back-Latenz bei einem Lese-Miss

64 Bit ?

- Warum 64 Bit?
  - ALU verarbeitet 64 Bit-Werte
    - bringt für Programme im Desktop-Umfeld wenig, da hier meist nur 32 Bit-Operationen
    - *wenn* höhere Bitbreite nötig: häufig per SSE (wird auch von 32 Bit CPUs unterstützt)
  - > 4 GB RAM möglich
    - momentan Grenze bei 2 oder 3 GB, wegen Memory-Mapping sonstiger I/O
  - höherer virtueller Adressraum
    - einfachere Adressierung für Datenmengen im Bereich > 4 GB
    - Datenbankanfragen schneller
- Nachteile von 64 Bit gegenüber 32 Bit
  - höherer Speicherverbrauch, da Adressen und Daten länger sind
  - komplexere Hardware

64 Bit

- 1991: MIPS-Architektur (RISC) wurde auf 64 Bit erweitert
- 2003: AMD64 (bis kurz vor Markteinführung: „x86-64“)
  - 48 Bit virtueller Adressraum
  - 40 Bit physikalischer Adressraum => max. 1 TB RAM

  Maßnahmen:
  - IA-32 Befehlssatz um 64-Bit-Befehle erweitert
  - Verbreiterung der bestehenden 32-Bit-Register auf 64 Bit
  - Verdoppelung der General-Purpose-Register
  - Verdoppelung der 128-bittigen SSE-Register

  ↙ daher abwärtskompatibel zu 32 Bit
    - Legacy Mode: nur 32 Bit werden genutzt
    - Compatibility Mode: OS ist 64-bittig, Software 32 bittig, Ausführung in 32 Bit
    - 64-Bit-Mode: OS und SW sind 64 bittig

Athlon: Modellpflege

- Änderungen vom Athlon XP Palomino (2001) bis heute (Toledo)
  - Taktrate: 1,3 GHz -> 2,6 GHz
  - Strukturgröße: 180 nm -> 90 nm
  - Core: single -> dual (seit 2005)
  - Bitbreite: 32 Bit -> 64 Bit (seit 2003)
  - Anzahl Transistoren: 37 Mio. -> 233 Mio.
  - Die-Größe: 129 mm² -> 199 mm²
  - FSB: 133 MHz -> 200 MHz + HyperTransport
  - Core-Spannung: 1,75 V -> 1,3 V
  - Leistung: ca. 66 W -> 110 W
  - L2-Cache: 256 KB -> 1 MB
  - L1-Cache: 64 KB + 64 KB -> 64 KB + 64 KB
  - Gleitkomma-Beschl.: MMX, 3DNow!, SSE -> MMX, 3DNow!, SSE, SSE2, SSE3
  - Pipeline: 10 / 15 -> 12 / 17 Stufen
  - Weitere Features heute:
    - Execute Disable Bit („NX“)
    - HW-Virtualisierung („Pacifica“)

Athlon: Zusammenfassung

- Pipeline hat weniger Stufen als P4, Taktfrequenzen etwas niedriger (Quantispeed-Idee etwas früher als Intel)
- Decodierung der x86-Befehle in Macro-OPs statt µOPs
- Superskalar und Out-of-Order wie auch beim P4
- L1-Data und Instruction Cache (kein Trace-Cache)
- DDR-Speichercontroller auf dem Chip (statt in der Northbridge)
- FSB langsamer, dafür HyperTransport-Links
- 64-Bit-Erweiterungen der IA-32-Befehle

Inhalt

- Grundlagen
  - Cache, virtueller Adressraum, Pipelining, Superskalarität, Hyperthreading, Multi-Core
- Der Intel Pentium 4
  - NetBurst, ALUs
- Der AMD Athlon
  - AMD64, Pipeline, Cache
- Energiespartechniken
  - Techniken, CoreDuo
- Trends
  - Mooresches Gesetz, 4-GHz-Grenze?

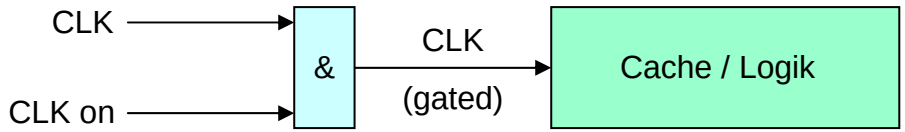
Cool bleiben

- Aktuelle CPUs verheizen > 100 W, Lötkolbenklasse!

- Direkte Gegenmaßnahmen (z. B. wichtig für den mobilen Bereich)
  - Verringerung der Kernspannung
    - Geht quadratisch in Leistungsaufnahme ein
    - Chips funktionieren nicht mehr
  - Verringerung der Taktfrequenz
    - Geht linear in Leistungsaufnahme ein
    - Leistungseinbußen

Cool bleiben 2

- Weitere Gegenmaßnahmen
  - Optimierung der Mikroarchitektur (Verringerung von Schaltflanken) (dynamisch)
    - nicht-verwendete Schaltungsteile abschalten (SpeedStep / Sleep-Modi, komplett abschalten)
    - Gated Clock



  - vertikale Integration
    - Chip-Komponenten auch übereinander statt nur nebeneinander (kürzere Wege => kürzere Latenzzeiten)
  - Optimierung der Fertigungstechnologie (statisch)
    - neue Materialien, z. B. Kupfer statt Aluminium (schon seit 2000)
    - Low-k-Isolatoren (Leiterbahnen: weniger parasitäre Kapazitäten)
    - High-k-Isolatoren (Transistor-Gate: geringere Leckströme)
    - Silicon-on-Insulator-Technologie, SOI (weniger Leck-Ströme im Off-State)
    - gestreckte Silizium-Struktur (höhere Elektronenbeweglichkeit zwischen Source und Drain)
    - Unterschiedliche Oxiddicken (→ unterschiedlich schnelle Transistoren)

Einige Stromspar-CPUs

Transmeta Crusoe TM5900	1 GHz	9,5 W (inkl. integrierter Northbridge)
AMD Mobile Sempron 800	800 MHz	13 W
AMD Turion MT-30	1,6 GHz	25 W
AMD Turion 64	2,4 GHz	35 W
AMD Turion 64 X2 ML-48	2,2 GHz	35 W
Intel Celeron M 373 ULV (1 V)	1 GHz	5 W
Intel Pentium M 770	2,13 GHz	27 W
Intel Core Duo T2600	2,16 GHz	31 W
Intel Pentium 4 M 2600	2,6 GHz	46 W
zum Vergleich: Pentium Extreme Edition 965	3,73 GHz	130 W

Der Intel CoreDuo

- Erster Dual-Core Mobil-CPU
- Nachfolger des Pentium M
  - Basiert auf P6 Architektur (PentiumPro von 1995)
- 65 nm
- gemeinsamer L2-Cache (2 MB)
- 31 W @ 2,16 GHz; 15 W @ 1,66 GHz (ULV)
- Enhanced SpeedStep
  - Variation in Kernspannung und Taktrate
    - immer gleiche Spannung, aber unterschiedliche Taktraten möglich
  - automatisches Senken der Taktrate, wenn der Die zu heiß wird
  - Abschaltung des L2-Cache möglich, auch partiell
- MicroOPs Fusion
  - Zusammenfassung von 2 µOPs zu einer größeren OP, die auch in 1 Takt verarbeitet werden kann
  - Intel: „ca. jede 10. µOP kann mit der darauffolgenden zusammengefasst werden“
  - war schon im Pentium M vorhanden

Der Intel Core Duo (Yonah)

- klassischer L1 Data und Instruction Cache (je 32 KB, write-back) ersetzen Trace-Cache
- 666 MHz FSB
- MMX, SSE, SSE2, SSE3
- Virtualisierung, Execute Disable Bit
- Integer Operationen etwas langsamer als im Vorgänger Dothan
  - Kern-zu-L2 Latenz
  - Dothan 10 Takte
  - Yonah 14 Takte
- Kein EM64T!
- → gilt nur als Zwischenschritt zu Core 2

Inhalt

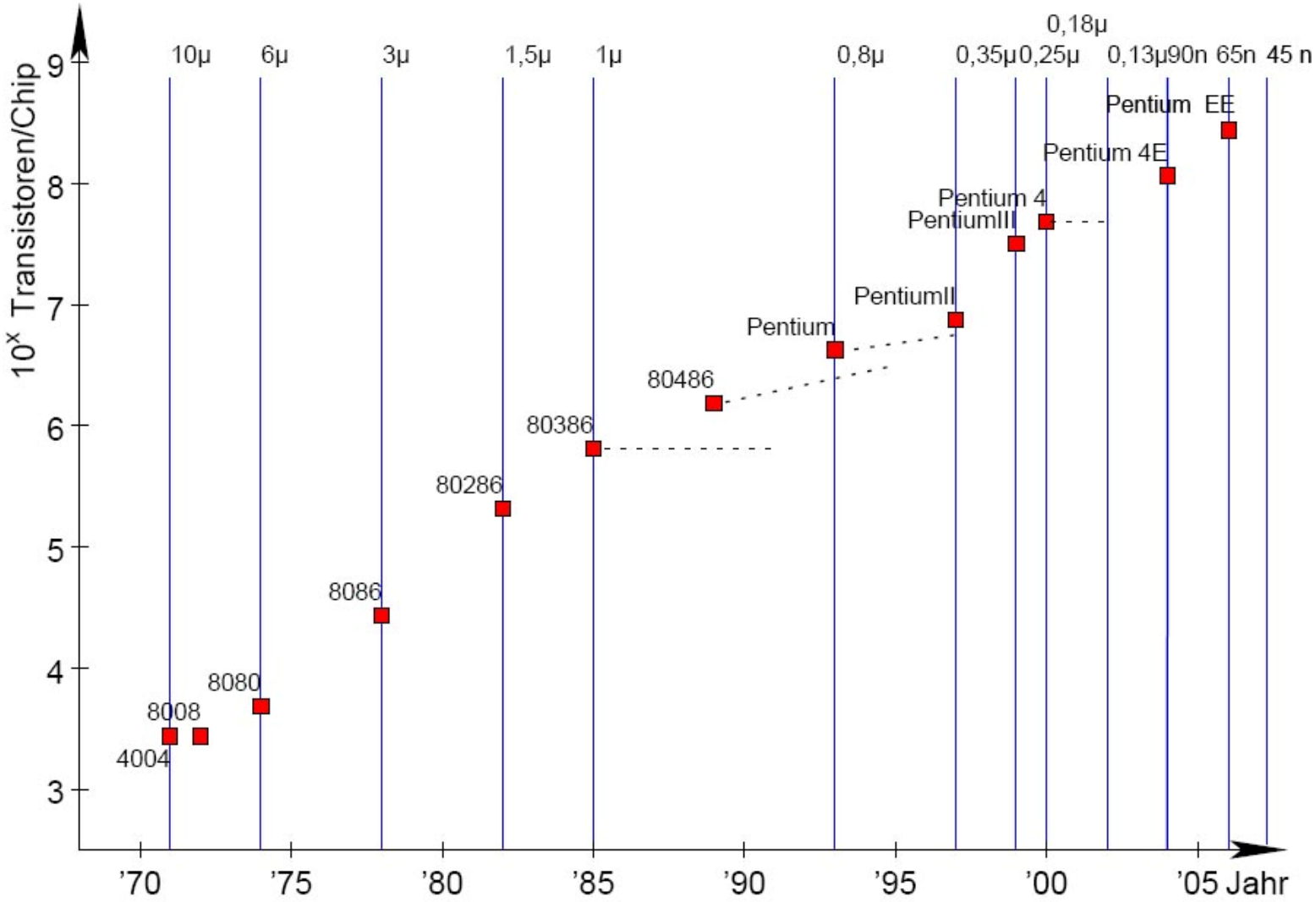
- Grundlagen
  - Cache, virtueller Adressraum, Pipelining, Superskalarität, Hyperthreading, Multi-Core
- Der Intel Pentium 4
  - NetBurst, ALUs
- Der AMD Athlon
  - AMD64, Pipeline, Cache
- Energiespartechniken
  - Techniken, CoreDuo
- Trends
  - Mooresches Gesetz, 4-GHz-Grenze?

Trend: Immer mehr

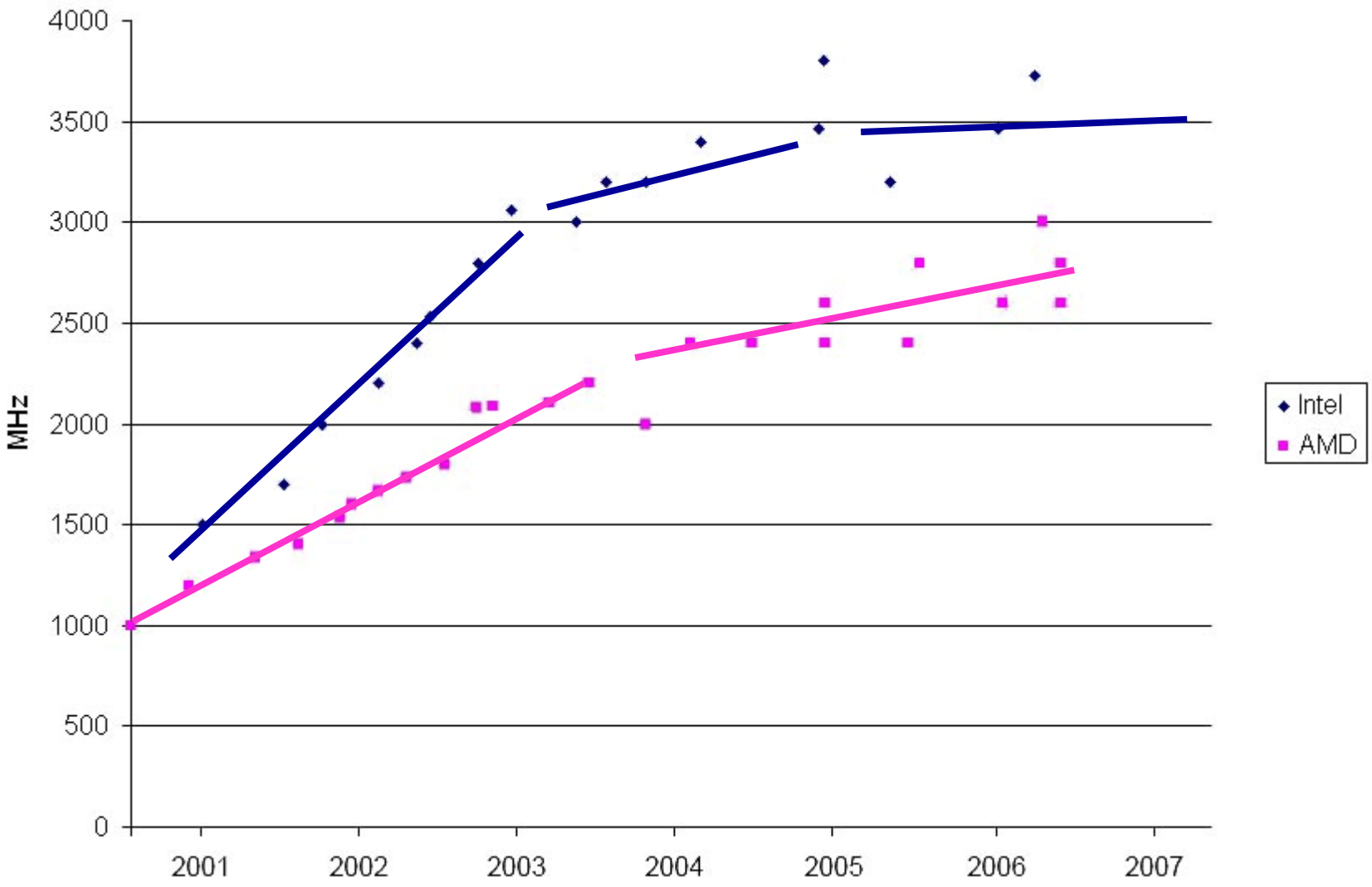
- Immer mehr Komponenten auf einem Chip

1978	8086	
1985	80386	<= 32 Bit
1989	80486	<= FPU, L1-Cache
1995	PentiumPro	<= L2-Cache
1997	Pentium	<= MMX
1999	Pentium III	<= SSE
2000	Crusoe	<= Northbridge
2002	Pentium 4	<= HyperThreading
2003	Athlon 64	<= DDR-Controller, HyperTransport, 64 Bit
2004	Pentium 4	<= Execute Disable Bit
2005	Athlon 64 / Pentium 4	<= Dual-Core

Mooresches Gesetz am Beispiel der Intel-CPUs



Trend: immer schneller?



Trend: immer schneller?

- 4 GHz – Grenze?

Intel kills plans for 4GHz Pentium

By Michael Kanellos
Staff Writer, CNET News.com
Published: October 14, 2004, 10:49 AM PDT

TalkBack E-mail Print

update Intel is dumping plans to release a Pentium 4 processor that runs at 4GHz, saying it will boost performance on next year's chips using other means than clock speed.

Trendwende: Taktrate => Parallelität

- > 4 GHz möglich, aber
  - Fabriken immer teurer
  - Energieverbrauch immer höher
  - CPU-Kühlung immer schwieriger

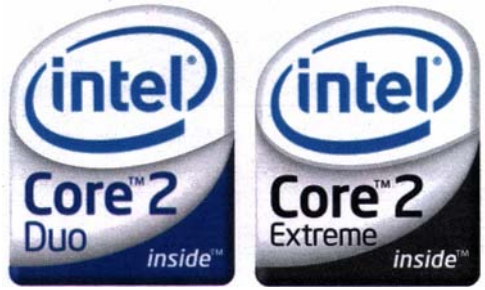
  ↙andere Beschleunigungsmaßnahmen billiger!

- weg von höherer Frequenz (eher niedriger), hin zu mehr Parallelität und höherer Energieeffizienz!
- Multi-Core!

Wie geht‘s weiter bei Intel?

- Intel Core Microarchitecture löst NetBurst ab
  - Ziel nicht mehr nur hohe Taktraten, sondern auch höhere Energie-Effizienz
  - CMA stammt von Mobilplattform ab,
    bezieht sich nun aber nicht nur auf Notebooks, sondern auch Desktops und Server
  - neue energieeffizientere + leistungsstärkere CPUs
  - 4-fach skalar (statt bisher 3-fach), d. h. 4 µOPs/Takt
  - 3 ALUs (statt 2); 128 Bit pro ALU
  - 14-stufige Pipeline (statt 31)
  - alle SSE-Befehle (auch die 128 Bit großen) in 1 Takt
  - shared L2-Cache (wie schon beim CoreDuo), verbesserter Prefetch-Mechanismus
  - DDR-Controller bleibt in der Northbridge
  - Micro-OPs-Fusion auch in Desktop-CPUs
  - Macro-OPs-Fusion: Micro OPs-Fusion auf x86-Befehls-Ebene
  - Energie sparen:
    - Einheiten, die gerade nicht benötigt werden, können komplett abgeschaltet werden
    - gated Clock

Wie geht‘s weiter bei Intel?



- Mitte 2006
  - „Core 2 Duo“ mit neuer Mikroarchitektur
  - „Core 2 Extreme“: höhere CPU- und Bus-Takte
- 2007
  - Quad-Core, längerfristig Multicore mit >= 64 Kernen
  - 45 nm
    - „Penryn“-Kern: „20% faster transistors, while requiring 30% less power than 65 nm“
- 2009
  - 32 nm

Wie geht‘s weiter bei AMD?

- 23.05.2006
  - „AM2“-CPUs: DDR2-Support
- Ende 2006
  - Produktionsstart 65 nm Athlon 64
- 2007
  - Erweiterung des Instruktionssatzes: schnellere Integer- und Float-Befehle
- Anfang 2008
  - Vierkern-CPUs mit gemeinsamem L2-Cache
  - DDR 3
  - HyperTransport 3.0

Ende