

## Arbeitsweise eines Cache-Speichers

- ❑ Cache-Steuerung prüft, ob
  - Der zur Speicheradresse gehörende Hauptspeichereintrag als Kopie im Cache steht (**Bedingung 1**) und
  - Dieser Cache-Eintrag durch das Gültigkeits-Bit (Valid-Bit) als gültig gekennzeichnet ist (**Bedingung 2**)
- ❑ Prüfung führt zu einem Cache-Treffer oder zu einem Fehlzugriff.
- ❑ Cache-Fehlzugriff (Cache-miss): eine der beiden Bedingungen ist nicht erfüllt.



## Arbeitsweise eines Cache-Speichers

- ❑ Cache-Fehlzugriff (Cache-miss): eine der beiden Bedingungen ist nicht erfüllt.

### Lesezugriffe (read miss)

- Lesen des Datums aus dem Hauptspeicher und Laden des Cache-Speichers
- Kennzeichnen der Cache-Eintrag als gültig (V-Bit setzen)
- Speichern der Adressinformation im Adress-Speicher des Cache-Speichers



## Arbeitsweise eines Cache-Speichers

- ❑ Cache-Fehlzugriff (Cache-miss): eine der beiden Bedingungen ist nicht erfüllt.

### Schreibzugriffe (write miss)

Aktualisierungsstrategie bestimmt, ob

- der entsprechende Block in den Cache geladen und dann mit dem zu schreibenden Datum aktualisiert wird oder, ob
- nur der Cache aktualisiert wird und der Hauptspeicher unverändert bleibt



## Arbeitsweise eines Cache-Speichers

- ❑ Cache-Treffer (Cache-hit, read hit, write hit):
  - Beide Bedingungen 1 und 2 sind erfüllt
  - Zugriff erfolgt auf den Cache-Speicher



# Ersetzungsstrategie

- ❑ **Ersetzungsstrategie** gibt an, welcher Teil des Cachespeichers nach einem Cache-Miss durch eine neu geladene Speicherportion überschrieben wird.
- ❑ **Ersetzungsstrategie** nur bei voll- oder n-fach satzassoziativer Cachespeicherorganisation angewandt
- ❑ meist die sehr einfache Strategie gewählt, die am längsten nicht benutzte Speicherportion zu ersetzen (**LRU-Strategie, Least Recently Used**).



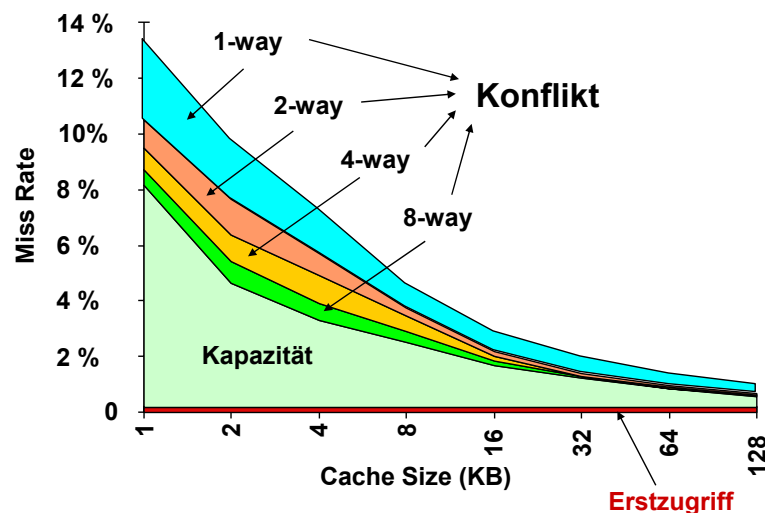
3C

# Ursachen für die Fehlzugriffe

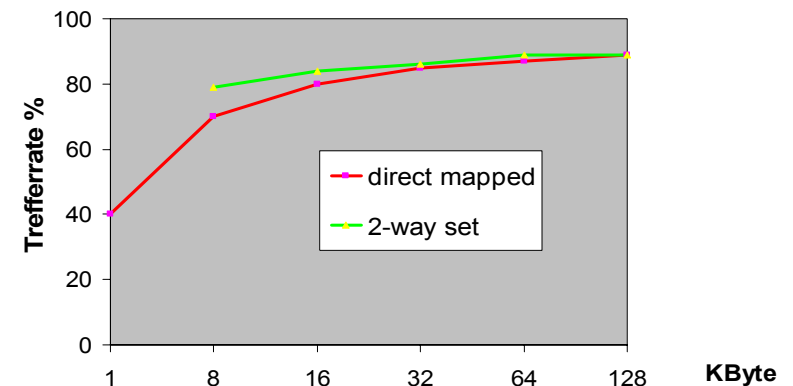
- **Erstzugriff** (*compulsory* - obligatorisch): Beim ersten Zugriff auf einen Cache-Block befindet sich dieser noch nicht im Cache-Speicher und muss erstmals geladen werden. Kaltstartfehlzugriffe (*cold start misses*) oder Erstbelegungsfehlzugriffe (*first reference misses*).
- **Kapazität** (*capacity*): Falls der Cache-Speicher nicht alle benötigten Cache-Blöcke aufnehmen kann, müssen Cache-Blöcke verdrängt und eventuell später wieder geladen werden.
- **Konflikt** (*conflict*): ein Cache-Block wird verdrängt und später wieder geladen, falls zu viele Cache-Blöcke auf denselben Satz abgebildet werden. Kollisionsfehlzugriffe (*collision misses*) oder Interferenzfehlzugriffe (*interference misses*). Kollisionsfehlzugriffe treten nur bei direkt abgebildeten oder satzassoziativen Cache-Speichern beschränkter Größe auf.



## Ursachen für die Fehlzugriffe



## Erzielbare Cache-Trefferquoten



**Cache-Größen < 64 kByte:** 2-Way Set Associative Cache besser als Direct Mapped Cache

**Cache-Größen ≥ 64 kByte:** kaum noch Unterschiede



# Erzielbare Cache-Trefferquoten

Nach Untersuchungen von Agarwal, Hennessy und Horowitz:

- Eine Cache-Trefferquote von circa 94% kann bei einem 64 kByte großen Cachespeicher erreicht werden (selbstverständlich gilt: je größer der Cachespeicher, desto größer die Trefferquote)
- Getrennte Daten- und Befehls-Cachespeicher sind bei sehr kleinen Cachespeichergrößen vorteilhaft, fallen jedoch bei Cachespeichergrößen ab ca. 8 KByte nicht mehr ins Gewicht
- Bei Cachespeichergrößen ab 64 KByte sind Direct Mapped Cachespeicher mit ihrer Trefferquote nur wenig schlechter als Cachespeicher mit Assoziativität 2 oder 4



# Erzielbare Cache-Trefferquoten

- ➔ Voll-assoziative Cachespeicher werden heute nur für sehr kleine auf dem Chip integrierte Caches mit 32 bis 128 Einträgen verwendet.

Bei größeren Cachespeichern findet sich zur Zeit ein Trend zur Direct Mapped Organisation oder 2 - 8 fach assoziativer Organisation.



# Verwendung mehrerer Caches

Oft findet sich eine mehrstufige Cache-Organisation:

- Auf dem Prozessor-Chip befindet sich der sogenannte **First-Level-Cache (On-Chip-Cache)**
- Häufig getrennte On-Chip-Caches: **Befehlscache** für die Befehle und **Datencache** für die Daten.
  - ➔ paralleler Zugriff auf Programm und Daten, wodurch die hohen Anforderungen bei heutigen Superskalar-Prozessoren an die Speicherbandbreiten erfüllt werden können

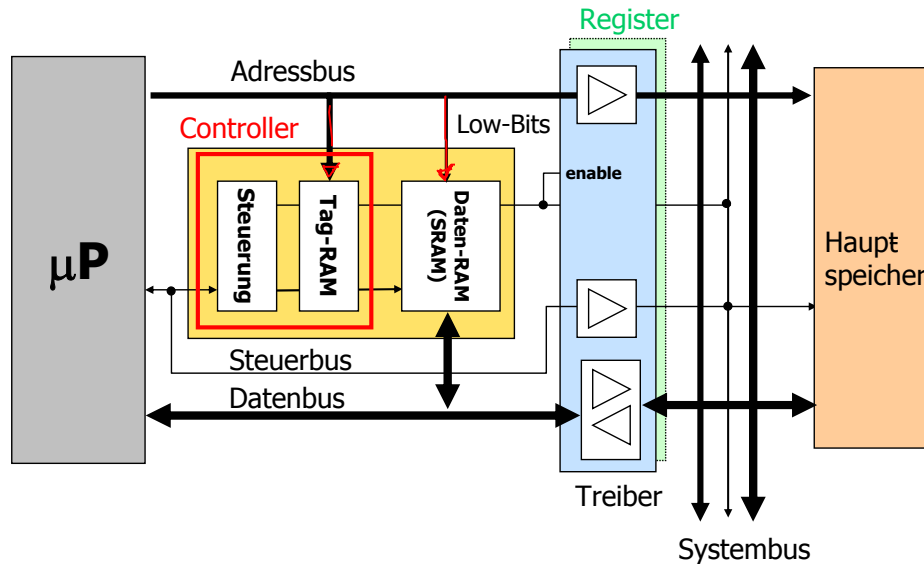


# Verwendung mehrerer Caches

- Auf dem Chip wird eine **Havard-Architektur** realisiert, bei der Programm und Daten in getrennten Speichern liegen
- Außerhalb des Prozessor-Chips befindet sich häufig ein weiterer, größerer Cache, der sogenannte **Secondary-Level-Cache (On-Board-Cache)**, 64 - 1024 KByte groß)
- Der Secondary-Level-Cache kann parallel zum Hauptspeicher an den Bus angeschlossen werden (**Look-Aside-Cache**). Er sorgt dafür, dass bei einem First-Level-Cache-Miss die Daten schnell nachgeladen werden können



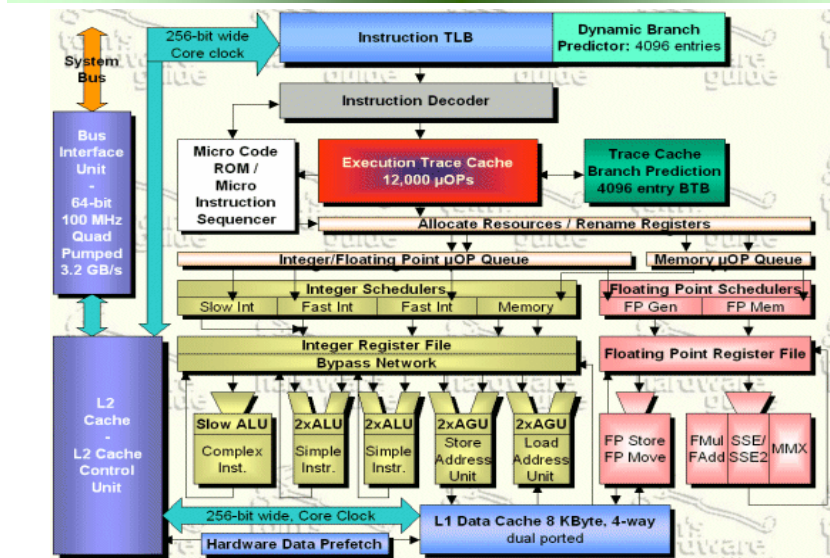
## Anbindung eines Off-Chip-Caches an den Systembus



## Anbindung des Caches an den Systembus

- **Cache-Controller:**  
Tag-RAM + Steuerung + Tag-Komparator  
Da dieser sehr schnell sein muss → auf einem Chip integriert
- Cachespeicher selbst ist separat mit SRAM-Bausteinen aufgebaut
- Cache-Controller übernimmt die Steuerung der Treiber zum Systembus (Systembuszugriff nur bei Cache-Miss, sonst ist der Systembus für andere Komponenten frei), sowie der Systembusseignale zur Einfügung von Wartezyklen bei Cache-Miss (READY, HOLD, HOLDA, ...)

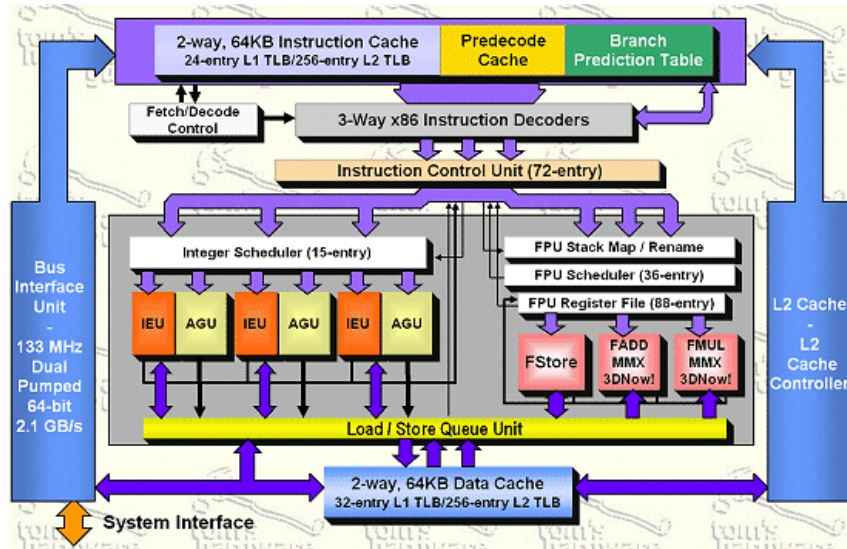
## Cache-Speicher in Pentium 4



## Cache-Speicher in Pentium 4

- ❑ **L1-Cache:**
  - 8 Kbyte Daten-Cache (Bei Pentium III: 16 Kbyte; bei Athlon: 64 Kbyte)
  - 4-way set associative Daten-Cache
  - Cache-lines mit 64 Byte (dual-pot-Architektur)
  - Write-Through
  - 12K µOPs „Trace“-Cache
- ❑ **L2-Cache:**
  - 256 Kbyte
  - 8-way set associative
  - Cache-lines mit 128 Byte
  - Write back
  - Bandbreite 44,8 Gbyte/s (16 Gbyte/s bei Pentium III)

# Cache-Speicher bei Athlon



## Fragen, die sich ein Speicherhierarchie-Designer stellen muss:

- ❑ Wohin kann ein Block abgebildet werden?  
(Block-Abbildungsstrategie)
  - Vollassoziativ, Satz-Assoziativ, Direct-Mapped
- ❑ Wie kann ein Block gefunden werden?  
(Block-Identifikation)
  - Tag/Block
- ❑ Welcher Block soll beim einem Miss ersetzt werden?  
(Block-Ersetzungsstrategie)
  - Random, FIFO, LRU
- ❑ Was passiert bei einem Schreibzugriff?  
(Schreibe-Strategie)
  - Write back oder Write Through (mit Write Buffer)

