

Übung 05: Caches

Einführung in die Rechnerarchitektur

Niklas Ladurner

School of Computation, Information and Technology
Technische Universität München

15. November 2024



TUM Uhrenturm

Keine Garantie für die Richtigkeit der Tutorfolien.
Bei Unklarheiten/Unstimmigkeiten haben VL/ZÜ-Folien recht!

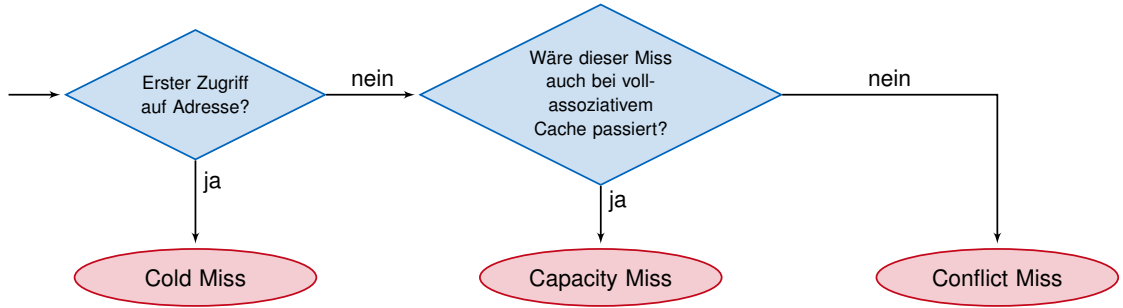
- Zugriffe auf Hauptspeicher (\equiv RAM) sind **extrem** langsam. Lösung: Caches
- „Zwischenstation“ zwischen Registern (sehr schnell, sehr klein) und Hauptspeicher (sehr langsam, sehr groß)
- Idee: Häufig genutzte Daten im Cache zwischenspeichern, der Rest wird bei Bedarf aus dem Hauptspeicher geholt
- heutzutage meist L1/L2/L3-Caches: Caches aufsteigender Größe, aber absteigender Zugriffszeit

- **Hit**: Datum liegt im Cache, **Miss**: Datum nicht im Cache, muss erst aus Hauptspeicher geholt werden
- Ziel: möglichst hohe **Hitrate** (Hits/Anfragen), d.h. häufig genutzte Daten liegen im Cache
- **zeitliche Lokalität**: Zugriff auf $x \rightarrow$ wschl. Zugriff auf x in Zukunft
- **räumliche Lokalität**: Zugriff auf $x \rightarrow$ Zugriff auf Daten in der Nähe (oft durch Cacheline abgedeckt)
- Verdrängungsstrategien: Random, LRU, LFU, FIFO

Cachestrukturen

- Direct Mapped Cache: direkte Abbildung Hauptspeicheradresse \rightarrow Cache-Adresse (jede Cachezeile kann nur an einer bestimmten Stelle im Cache stehen)
- Fully Associative Cache: eine Cacheline kann an einer beliebigen Stelle im Cache stehen
- Set Associative Cache (Mengenassoziativer Cache): Aufteilung in sog. Sets, Set wird durch Adresse bestimmt, aber innerhalb des Sets kann die Cacheline an einer bel. Stelle stehen
- Tag: Identifikation der Cacheline im Set, Index: bestimmt Set im Cache, Offset: bestimmt Datum innerhalb einer Cacheline
- Tag, Index, Offset werden aus Adresse eines Zugriffs berechnet
- In jeder Cachezeile liegen mehrere Speicherzellen (deswegen Offset benötigt)

Klassifikation von Misses



Nach Gschoßmann et al., 2023

Ein paar Formeln...

Für einen n-assoziativen Cache (jeweils n Cachezeilen in einem Set):

- $\text{Anzahl Cache-Lines} = \frac{\text{Cachegröße}}{\text{Cachezeilengröße}}$
- $\text{Anzahl Cache-Sets} = \frac{\text{Anzahl Cache-Lines}}{n}$
- $\text{Anzahl Index-Bits} = \lceil \log_2(\text{Anzahl Cache-Sets}) \rceil$
- $\text{Anzahl Offset-Bits} = \lceil \log_2(\text{Cachezeilengröße}) \rceil$
- $\text{Anzahl Tag-Bits} = \text{Anzahl Adressbits} - \text{Anzahl Index-Bits} - \text{Anzahl Offset-Bits}$

Fragen?

- „H05 — Cache-Simulation“ bis 24.11.2024 23:59 Uhr
- Caching von Adressen auf verschiedenen Cachetypen und Verdrängungsstrategien

- [Zulip: „ERA Tutorium - Do-1600-1“ bzw. „ERA Tutorium - Fr-1500-2“](#)
- [RISC-V-Spezifikation](#)
- [ERA-Moodle-Kurs](#)
- [ERA-Artemis-Kurs](#)
- [Wikipedia zu Caches](#)
- [Elektronik-Kompendium zu Caches](#)

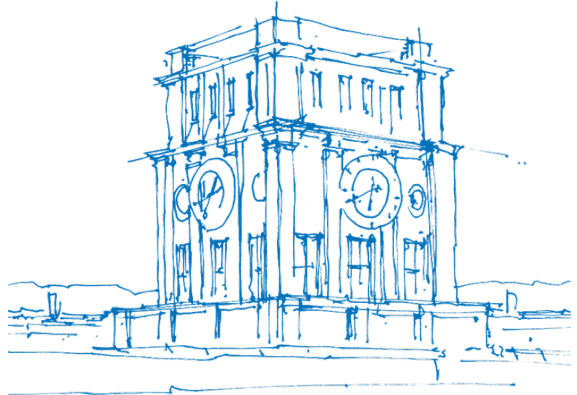
Übung 05: Caches

Einführung in die Rechnerarchitektur

Niklas Ladurner

School of Computation, Information and Technology
Technische Universität München

15. November 2024



TUM Uhrenturm