

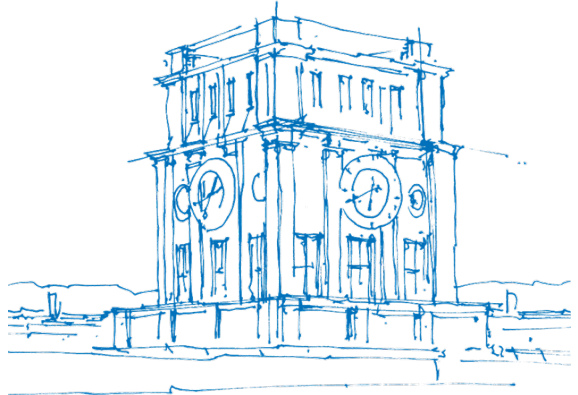
Übung 10: Caches

Einführung in die Rechnerarchitektur

Niklas Ladurner

School of Computation, Information and Technology
Technische Universität München

5. Januar 2024



TUM Uhrenturm

Durchzählen!

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

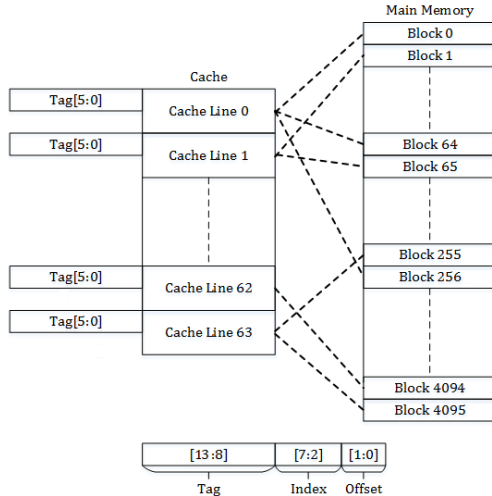
- Static RAM (SRAM): schnell, aber geringe Kapazität (Flipflops)
- Dynamic RAM (DRAM): langsam, dafür aber günstig in großer Kapazität herstellbar (Kondensatoren)
- Vorteile/Nachteile der beiden Arten sind einfache Klausuraufgabe!

- „Zwischenstation“ zwischen Registern (sehr schnell, sehr klein) und Hauptspeicher (sehr langsam, sehr groß)
- Idee: Häufig genutzte Daten im Cache zwischenspeichern
- 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)
- Formeln für CPU-Time und Memory Access Rate kennen \rightarrow war letztes Jahr Klausuraufgabe
- Ersetzungsstrategien auch wichtig (FIFO, LRU, LFU)

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
- Adresse eines Datums wird aufgeteilt!



Memory Size = 16Kbytes
Memory Block Size = 4 bytes
Cache Size = 256 bytes
Block Size = 4 bytes
Associativity = 1
Number of Sets = 64

Quelle: Wikipedia

Ein paar Formeln...

Für einen n-assoziativen Cache (jeweils n Cachezeilen in seinem 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?

- H10 - 4-Fach-Assoziativ bis 14.01.2024 23:59 Uhr
- Werte in einem Cache einfügen (+ Ersetzungen), Abgabe im Textformat
- gut machbar, Tests geben wieder vor der Deadline volles Feedback

- Zulip: „ERA Tutorium - Mi-1600-MI4“ bzw. „ERA Tutorium - Fr-1100-MW2“
- Wikipedia zu Caches
- Elektronik-Kompendium zu Caches
- Elektronik-Kompendium zu SRAM/DRAM

Übung 10: Caches

Einführung in die Rechnerarchitektur

Niklas Ladurner

School of Computation, Information and Technology
Technische Universität München

5. Januar 2024



TUM Uhrenturm