Cache-Einführung

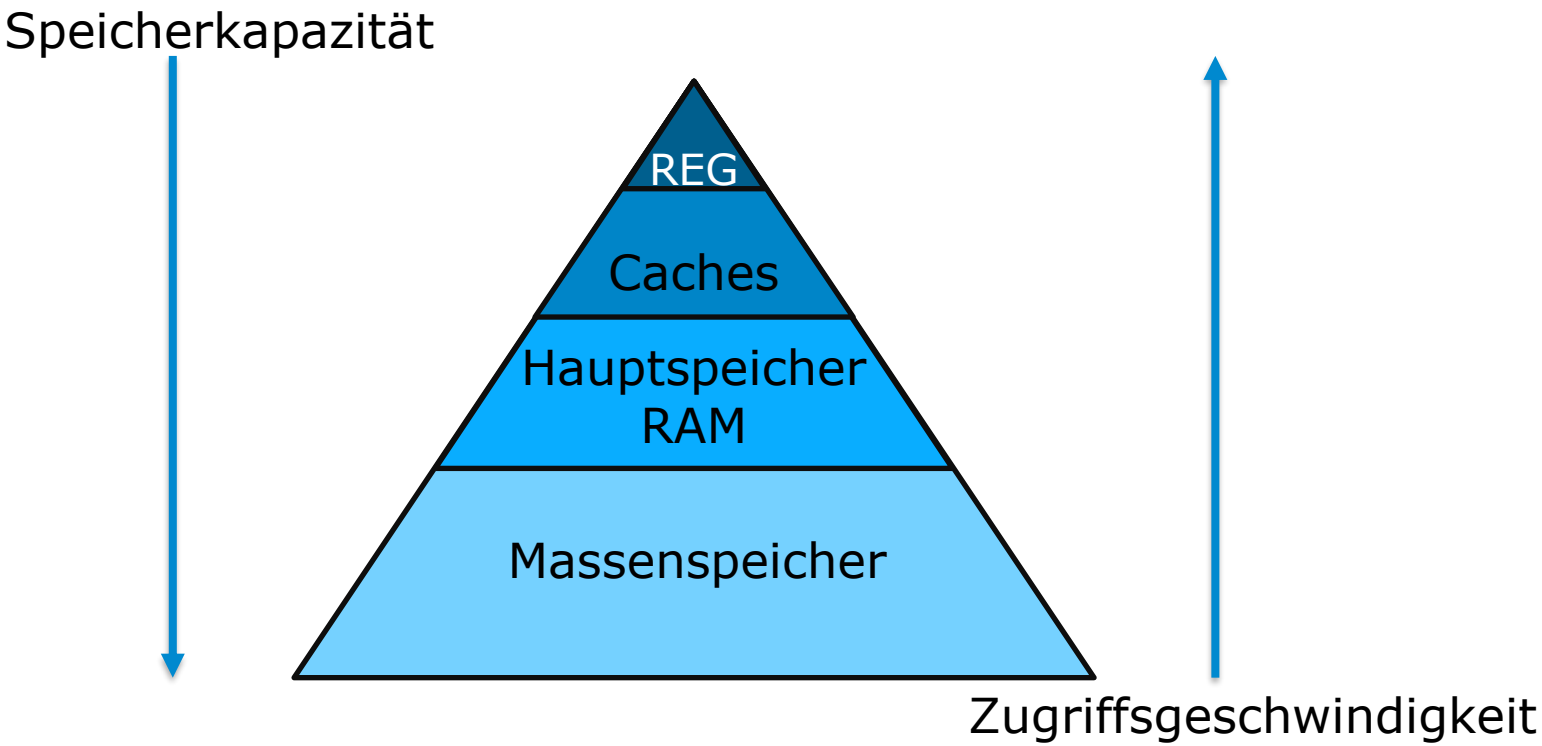
Architekturen und Entwurf von Rechnersystemen

TECHNISCHE UNIVERSITÄT DARMSTADT

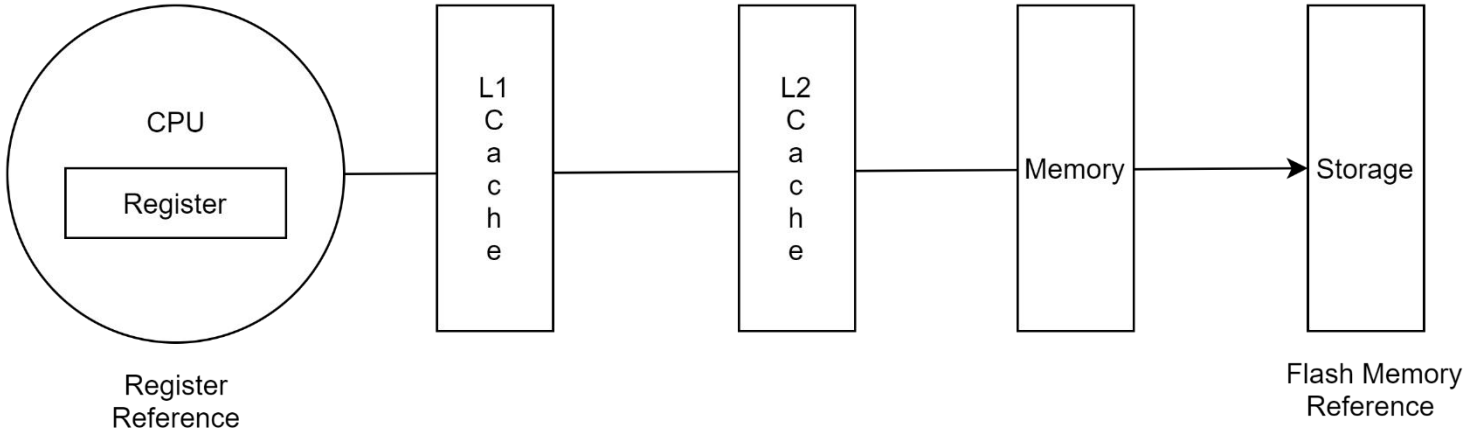
“Why did the developer go broke?”
…“Because he used up all his Cache”

[really_good_jokes@google]

Speicherhierarchie



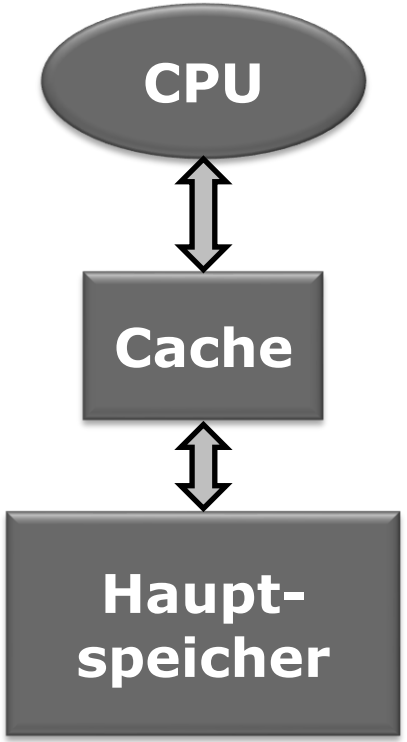
Speicherhierarchie



*John L. Hennessy und David A. Patterson. Computer Architecture: A Quantitative Approach. 5. Aufl. Amsterdam: Morgan Kaufmann, 2012. isbn: 978-0-12-383872-8.

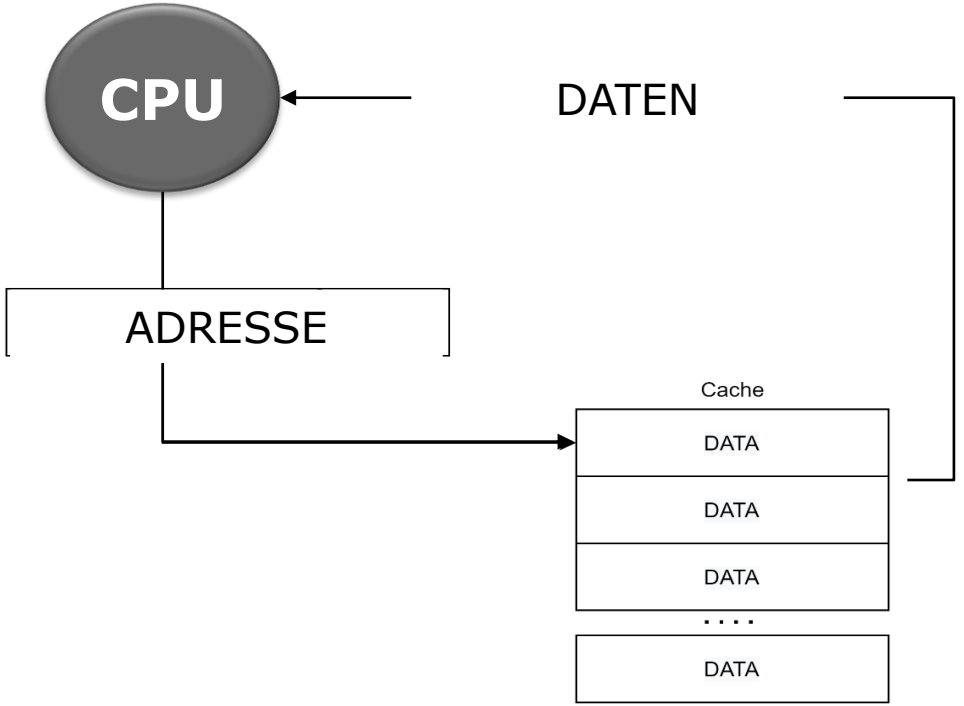
Caches

- Speichert Daten zwischen
  - Größer, aber langsamer als Registersatz
  - Kleiner, aber schneller als Hauptspeicher
- Sind Daten im Cache vorhanden?
  - Ja: **HIT**
  - Nein: **MISS**



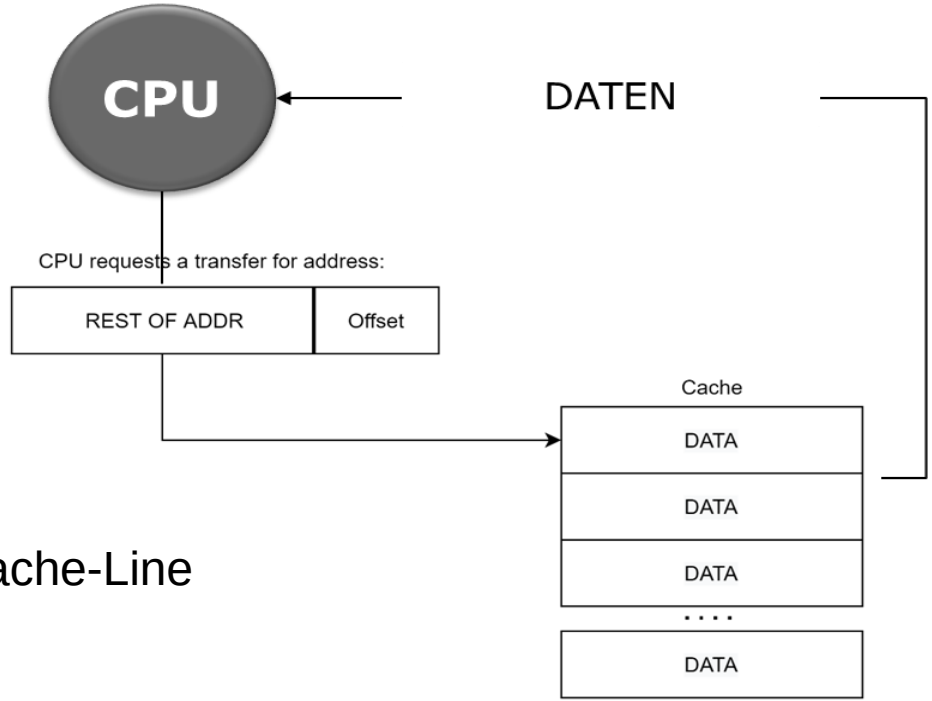
Cache - Funktion

- CPU ruft Daten an bestimmter Adresse aus Cache ab
- Annahmen
  - Speicher ist byte-adressierbar
  - Block-Größe des Caches: 32 Byte



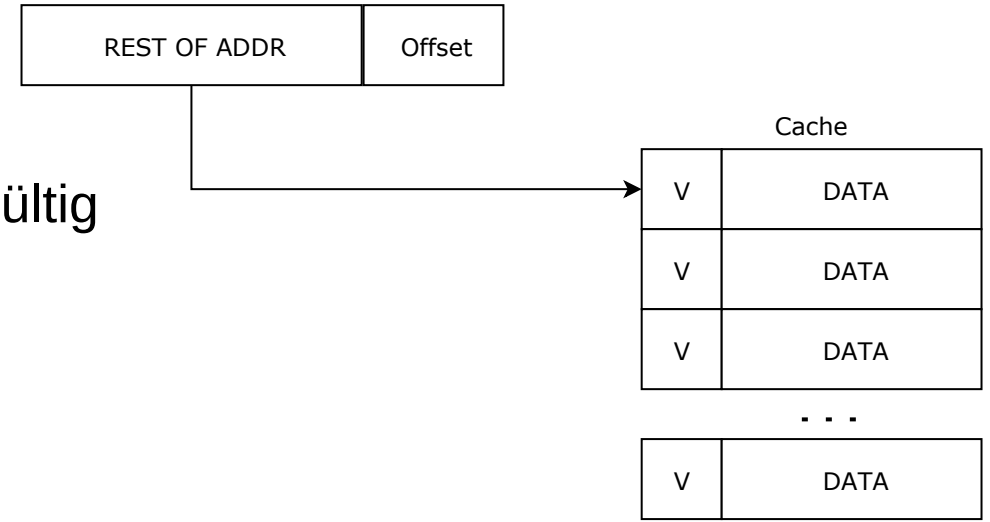
Cache - Funktion

- CPU ruft Daten an bestimmter Adresse aus Cache ab
- Annahmen
  - Speicher ist byte-adressierbar
  - Block-Größe des Caches: 32 Byte
- Adresse kann unterteilt werden in…
  - Block-Adresse des Cache-Blocks/Cache-Line
  - Offset innerhalb des Blocks



Status - Gültigkeit

- Im Cache gespeicherte Daten können (teilweise) ungültig sein
- Beispielsweise beim Systemstart: alle Daten sind ungültig

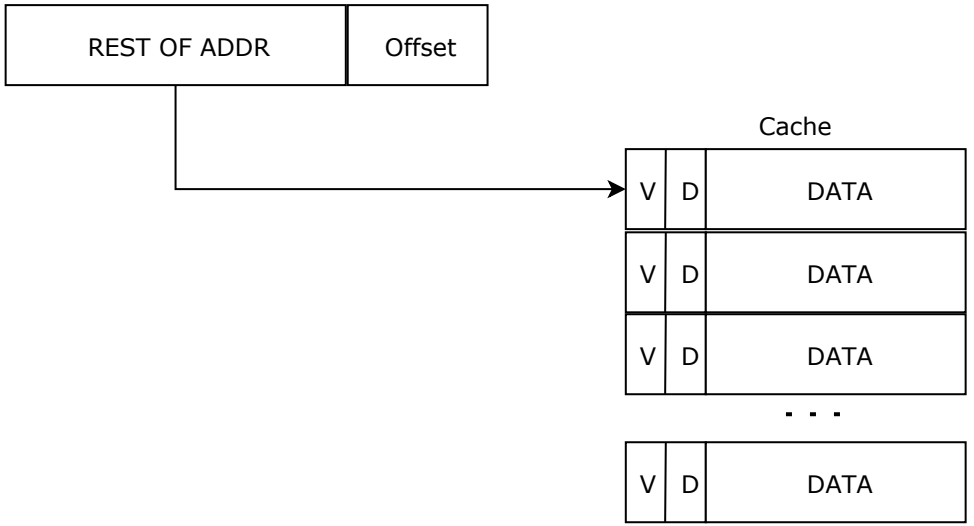


⇒ Valid Bit

Status - Dirty

- Cache enthält Kopie von Daten aus dem Hauptspeicher
- Änderungen im Cache müssen (irgendwann) an den Hauptspeicher übermittelt werden

⇒ Dirty Bit

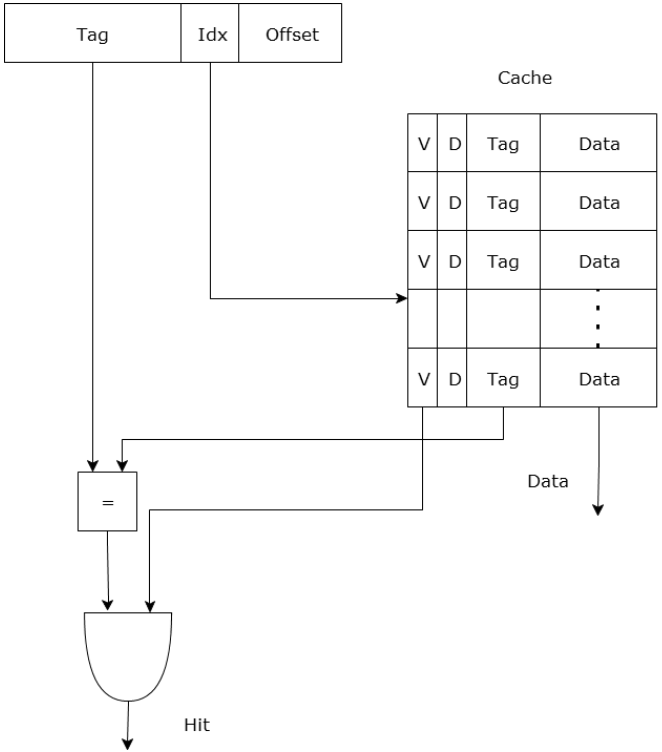


Cache - Adressen

- Hauptspeicher ist größer als Cache

  ⇒ Wie werden Adressen behandelt?

- Adresse wird unterteilt in…
  - Offset (innerhalb eines Cache-Blocks)
  - Index (Auswahl des Cache-Blocks)
  - Tag



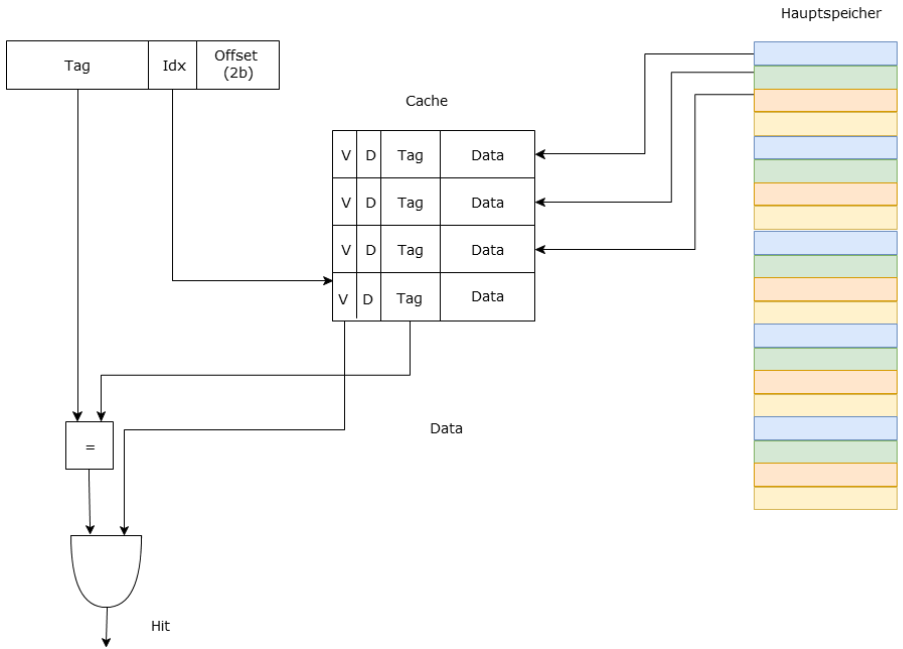
Cache - Adressen

- Hauptspeicher ist größer als Cache

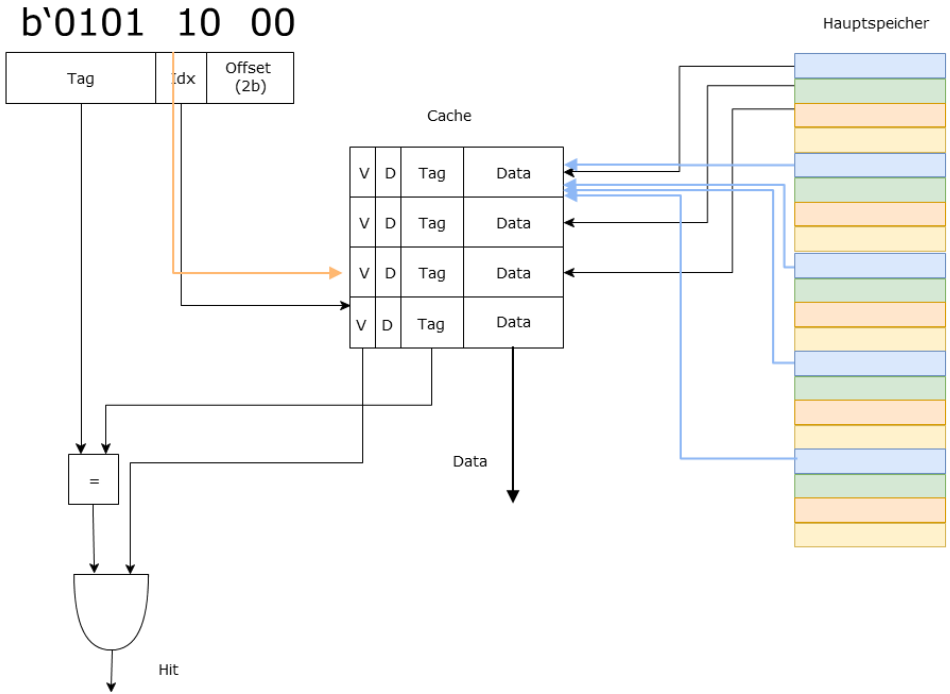
  ⇒ Wie werden Adressen behandelt?

- Adresse wird unterteilt in…
  - Offset (innerhalb einer Cache-Line)
  - Index (Auswahl der Cache-Line)
  - Tag

⇒ Cache-Lines werden mehrfach zugewiesen

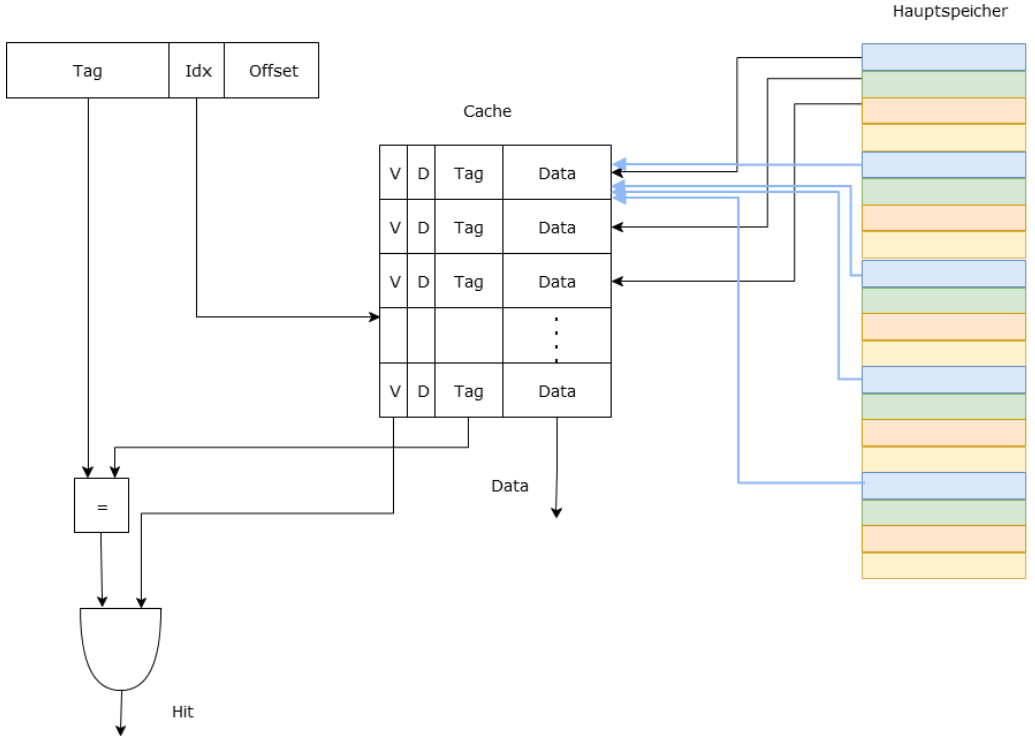


Cache - Funktion



Cache - Konflikte

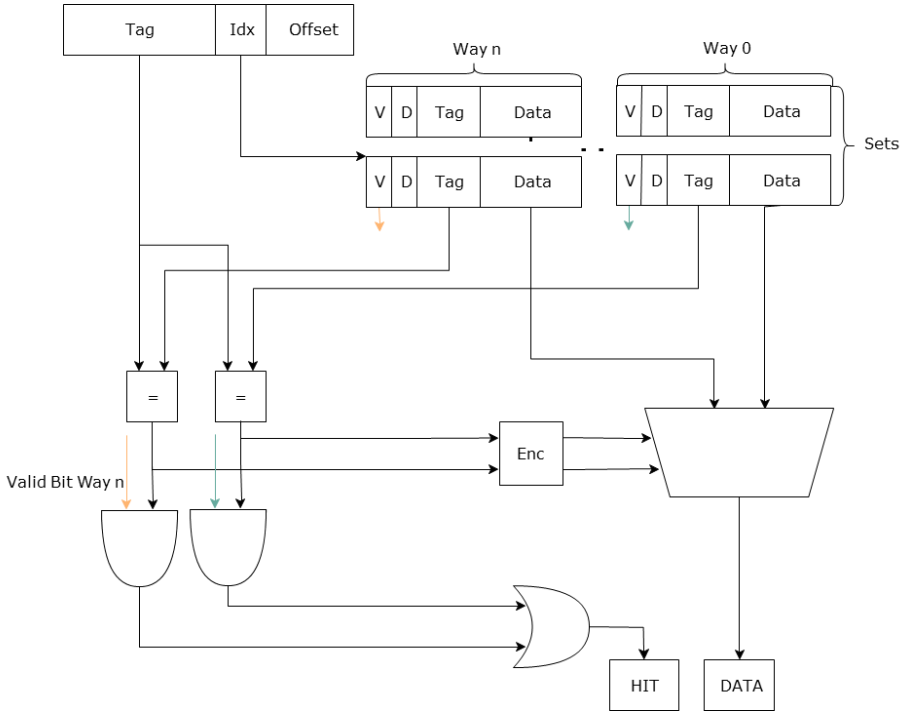
- Cache-Lines werden mehrfach zugewiesen
- Zugriff verdrängt alte Daten



Cache - Konflikte

- Cache-Lines werden mehrfach zugewiesen
- Zugriff verdrängt alte Daten

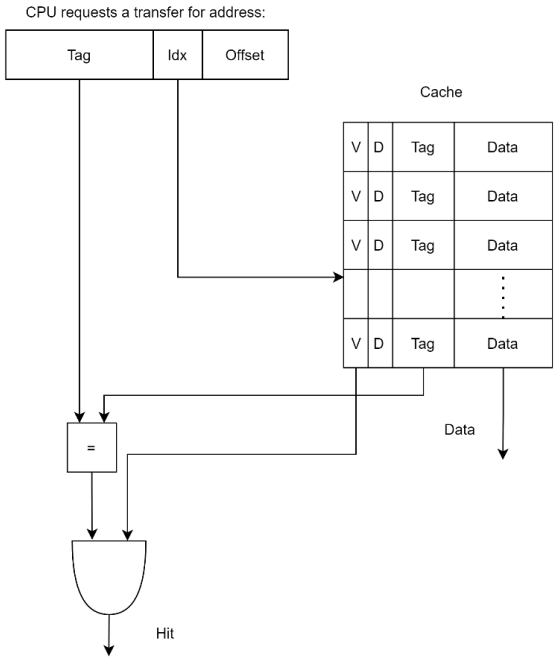
⇒ Cache mit mehreren „Ways“



Direct-Mapped Cache

- Mehrere Sets
- Ein *Way*

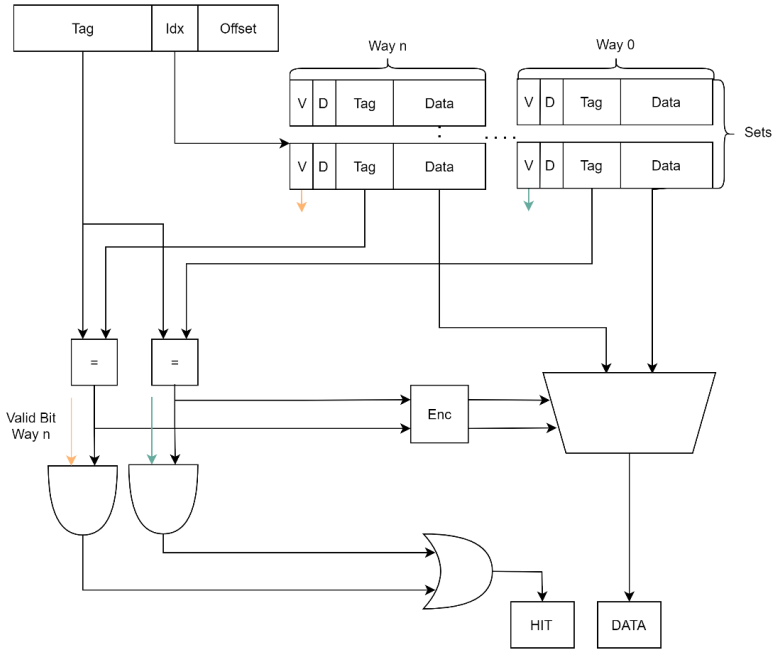
⇒ Ein Datum ist genau einer Cache-Line zugeordnet



Set-Associative Cache

- Mehrere Sets
- Mehrere *Ways*
  - „n-way set-associate“

⇒ Ein Datum ist mehreren Cache-Lines zugeordnet



Fully-Associate Cache

- Ein Set
- Mehrere *Ways*

⇒ Jedes Datum ist allen Cache-Lines zugeordnet

Cache Replacement Policies

- Keine dem Datum zugeordnete Cache-Line ist frei

  ⇒ Anderes Datum wird verdrängt

- Bei mehreren Optionen: Wie auswählen?
  - Random replacement (RR)
  - Least-Recently-Used (LRU)
  - Least-Frequently-Used (LFU)

Write Policies

TECHNISCHE UNIVERSITÄT DARMSTADT

- Daten werden im Cache aktualisiert

  ⇒ Hauptspeicher muss ebenfalls aktualisiert werden

- Wann findet Hauptspeicher-Aktualisierung statt?
  - Write-through
  - Write-back

Cache Misses

TECHNISCHE UNIVERSITÄT DARMSTADT

- Erster Zugriff auf Daten

  ⇒ Compulsory Miss
- Daten wurden verdrängt, da Set voll

  ⇒ Conflict Miss
- Daten wurden verdrängt, da Cache voll

  ⇒ Capacity Miss