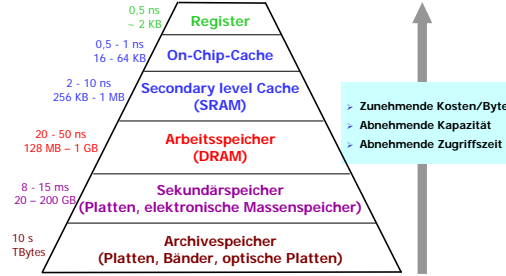


Cache-Speicher

- Speicherhierarchie
- Funktionsweise
- Aufbau
- Organisationsformen

Wdh: Speicherhierarchie



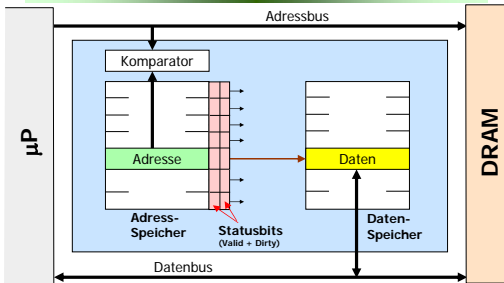
Cache- Speicher

- Pufferspeicher mit schnellem Zugriff
- Geringe Kapazität im Vergleich zum Hauptspeicher
- Cache-Speicher stellen die während einer Programmausführung aktuellen Hauptspeichereinhalte für Prozessorzugriffe als **Kopien** möglichst schnell zur Verfügung.
- Cache-Speicher-Verwaltung** sorgt dafür, dass der Cache-Speicher das Datum enthält, auf dem der Prozessor als nächstes zugreift.
- Cache-Controller (Hardware)** kopiert automatisch die Daten in den Cache, auf die der Prozessor zugreift.

Cache- Speicher

- Auf den Cachespeicher soll der Prozessor fast so schnell wie auf seine Register zugreifen können. Er ist deshalb bei Mikroprozessoren direkt auf dem Prozessorchip angelegt oder in der schnellen und teuren SRAM-Technologie realisiert.
- Cache-Treffer (Hit):** das angeforderte Datum ist im Cachespeicher vorhanden.
- Cache-Fehlzugriff (Miss):** das angeforderte Datum steht nur im Hauptspeicher.
- Hit-Ratio:** Trefferquote

Wdh. Aufbau eines Cache-Speichers



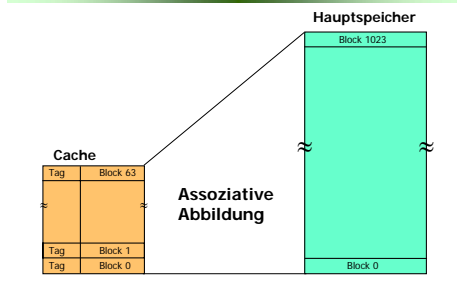
Cache-Strukturen

Wie wird festgestellt, ob die benötigten Daten im Cache sind und falls ja wie können diese gefunden werden?

3 Techniken für den Adressvergleich → 3 Cache-Typen:

- Voll-Assoziativer Cache
- Direct Mapped Cache
- n-Way Set Associative Cache

Voll-assoziativer Cache

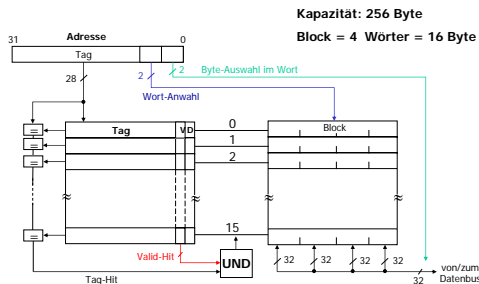


Voll-Assoziativer Cache

Vollparalleler Vergleich aller Adressen im Adressspeicher in einem einzigen Taktzyklus

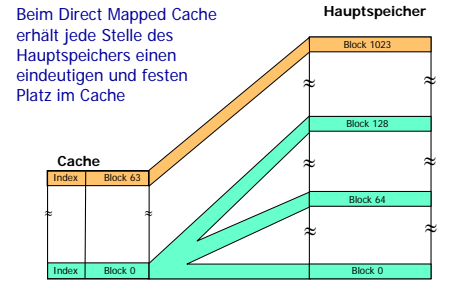
- Vorteil:**
- ein Datum kann an beliebiger Stelle im Cache abgelegt werden
 - Optimale Cache-Ausnutzung, völlig freie Wahl der Strategie bei Verdrängungen
- Nachteil:**
- Hoher Hardwareaufwand (für jede Cache-Zeile ein Vergleichler) → nur für sehr kleine Cachespeicher realisierbar
 - Die große Flexibilität der Abbildungsvorschrift erfordert eine weitere Hardware, welche die Ersetzungsstrategie (welcher Block soll überschrieben werden, wenn der Cache voll ist) realisiert.

Beispiel: Vollassoziativer Cache



Direct-mapped-Cache

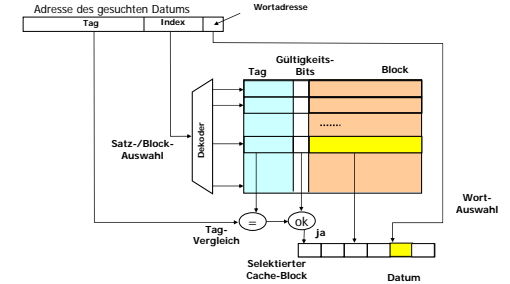
Beim Direct Mapped Cache erhält jede Stelle des Hauptspeichers einen eindeutigen und festen Platz im Cache



Direct-mapped-Cache

- Nachteile:**
- Ständige Konkurrenz der Blöcke (z. B. 0, 64, 128, ...), obwohl andere Blöcke im Cache frei sein können.
- Vorteile:**
- Geringer Hardwareaufwand für die Adressierung, da nur ein Vergleichler für alle Tags benötigt wird.

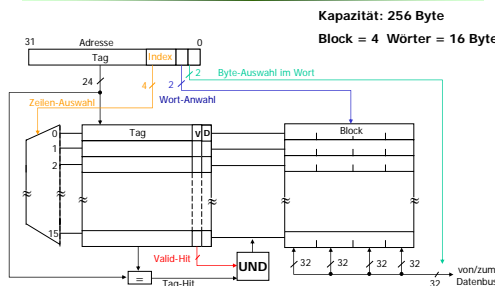
Adressierung im Direct-Mapped Cache



Merkmale des Direct Mapped Cache

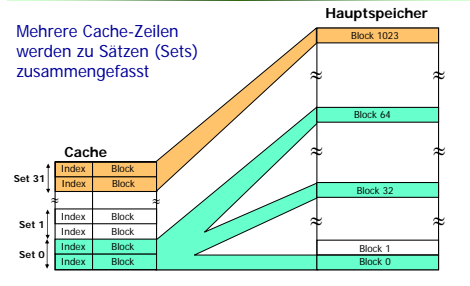
- Einfache Hardware-Realisierung (nur ein Vergleichler und ein Tag-Speicher)
- Der Zugriff erfolgt schnell, weil das Tag-Feld parallel mit dem zugehörigen Block gelesen werden kann
- Es ist keine Ersetzungsstrategie erforderlich, weil die direkte Zuordnung keine Alternativen zulässt
- Auch wenn an anderer Stelle im Cache noch Platz ist, erfolgt wegen der direkten Zuordnung eine Ersetzung
- Bei einem abwechselnden Zugriff auf Speicherblöcke, deren Adressen den gleichen Index-Teil haben, erfolgt laufendes Überschreiben des gerade geladenen Blocks. Es kommt zum "Flattern" (trashing)

Beispiel: Direct-mapped-Cache



n-way-set-assoziativer Cache

Mehrere Cache-Zeilen werden zu Sätzen (Sets) zusammengefasst



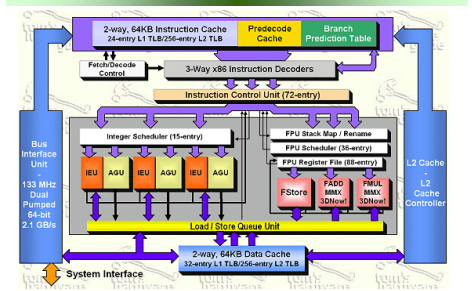
n-way-set-assoziativer Cache

- Kompromiss zwischen direct-mapped-Cache und vollassoziativen Cache.
- Verbesserte Trefferrate, da hier eine Auswahl möglich ist (der zu verdrängende Eintrag kann unter n ausgewählt werden).
- Ersetzungsstrategie notwendig:**
 - Zyklisch (der zuerst eingelagerte Eintrag wird auch wieder verdrängt, FIFO Strategie)
 - LRU Strategie (*least recently used*) der am längsten nicht mehr benutzte Eintrag wird entfernt.
 - Zufällig (durch Zufallsgenerator)

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 μ OPs „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)
 - Vollasoziativ, Satz Asoziativ, Direct- Mapped
- Wie kann ein Block gefunden werden? (Block-Identifikation)
 - Tag/Block
- Welcher Block soll bei einem Miss ersetzt werden? (Block-Ersetzungsstrategie)
 - Random, FIFO, LRU
- Was passiert bei einem Schreibzugriff? (Schreibe-Strategie)
 - Write back oder Write Through (mit Write Buffer)