

I. CACHE-SPEICHER

Hintergrund

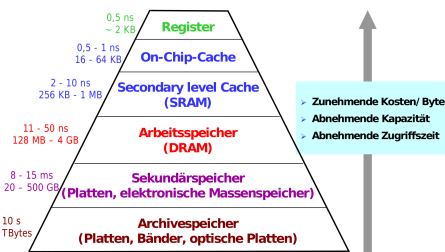
Lücke zwischen Verarbeitungs- (CPU) und Zugriffsgeschwindigkeit (DRAM) immer größer

Lösung: Hierarchische Anordnung verschiedener Speicher
 ↳ Ausgleich der unterschiedlichen Zugriffszeiten

Strategien:

1. **Cache-Speicher:** Kurze Zugriffszeiten
 ↳ Beschleunigung Prozessorzugriff
2. **Virtueller Speicher:** Vergrößerung des tatsächlich vorhandenen Hauptspeichers (z.B. bei gleichzeitiger Bearbeitung mehrerer Prozesse)

Leistung abhängig von Eigenschaften der Speichertechnologien, Adressierung und Organisation



CPU-Cache

kleiner, schneller Pufferspeicher

speichert *Kopien* von Hauptspeicherteilen, auf die mit hoher Wahrscheinlichkeit als nächstes zugegriffen wird

Effizienz durch *Lokalitätseigenschaft* von Programmen:

1. **zeitliche Lokalität:** zukünftig angesprochene Information mit hoher Wahrscheinlichkeit schon einmal angesprochen worden (z.B. Schleifen)
2. **örtliche Lokalität:** zukünftig angesprochene Information mit hoher Wahrscheinlichkeit in Nähe des bisherigen Zugriffs (z.B. Arrays)

Cache-Controller lädt alle Daten in Cache, auf die Prozessor zugreift

Daten werden aus Cache verdrängt, wenn sie nicht mehr benötigt werden

Cache – Funktionsweise

Lesezugriff: μP überprüft davor ob Datum in Cache steht

- **read hit:** Datum wird ohne Wartezyklen aus Cache geladen
- **read miss:** Datum wird mit Wartezyklen aus Arbeitsspeicher geladen und in Cache eingefügt

Schreibzugriff:

- **write miss:** Datum wird in DRAM und Cache geschrieben
- **write hit:** verschiedene Verfahren möglich

Schreibzugriff – Durchschreibverfahren

Datum wird von CPU immer gleichzeitig in Cache- und Arbeitsspeicher geschrieben

Vorteil: Konsistenzgarantie zwischen Cache und DRAM

Nachteil: Schreibzugriffe benötigen immer langsame Zykluszeit von Hauptspeicher, belasten Systembus

Schreibzugriff – gepuffertes Durchschreibverfahren

Verwendung von *Schreib-Puffer*, der zu schreibende Daten temporär aufnimmt

Daten werden dann automatisch von Cache-Controller in Hauptspeicher übertragen

Schreibzugriff – Rückschreibverfahren

Datum wird von CPU nur in Cachespeicher geschrieben und durch spezielles Bit (*dirty bit*) gekennzeichnet

Arbeitsspeicher wird nur geändert, wenn *dirty*-Datum aus Cache verdrängt wird

Vorteil: Alle Schreibzugriffe mit schneller Cache-Zykluszeit abwickelbar

Nachteil: Konsistenzprobleme zwischen Cache- und Hauptspeicher

Konsistenzprobleme

Andere Systemkomponenten (z.B. DMA-Controller) finden ggf. „veraltete Daten“ in DRAM vor, die von CPU längst geändert wurden

Andere Systemkomponenten können Daten in Hauptspeicher ändern, während CPU noch mit alten Daten aus Cache arbeitet

↳ aufwendige Verfahren bei Cache-Steuerung zur Inkonsistenzvermeidung erforderlich

Begriffe

Hit-Rate: = Anzahl Treffer pro Anzahl Zugriffe

Mittlere Zugriffszeit:

$$t_{\text{access}} = (\text{Hit-Rate})t_{\text{hit}} + (1 - \text{Hit-Rate})t_{\text{miss}}$$

Cache-Speicher – Aufbau

Besteht aus drei Teilen:

1. **Datenspeicher:** im Cache abgelegte Daten
2. **Adressspeicher:** Adresse dieses Datums im RAM
3. **Statusbits:** Geben an, ob Informationen gültig sind

Zusammen: *cache-line*