Mehrstufige Caches

Xuanqi Meng, Jeremias Rieser, Artem Bilovol

GRA SystemC Gruppe 192

2024

Inhalt

Problemstellung

- Moderne Prozessoren sind deutlich schneller als der Hauptspeicher.
- Von Neumann-Flaschenhals: Die beschränkte Bandbreite und langsamen Datenübertragungsraten zwischen Prozessor und Hauptspeicher führen zu einem Leistungsengpass..
- Ziel: Untersuchung der Auswirkungen von mehrstufigen Caches auf Laufzeit und Latenz.

Von-Neumann-Flaschenhals

- Prozessoren können Daten schneller verarbeiten als diese aus dem Hauptspeicher gelesen werden können.
- Der Hauptspeicher stellt einen Flaschenhals dar, der die Gesamtleistung des Systems limitiert.
- Lösung: Einsatz von Caches als Puffer zwischen Prozessor und Hauptspeicher.

Lösungsansätze und Optimierungen

- Implementierung einer Cache-Simulation in SystemC und C++.
- Simulation eines direkt-assoziativen Caches.
- Untersuchung von Parametern wie Cachegröße, Assoziativität, Cachezeilenanzahl und Zeilengröße und deren Auswirkung auf die Zugriffszeiten.

Cache-Architektur

- Verwendung von L1- und L2-Caches zur Reduktion der Zugriffszeit
- L1-Cache: kleine Größe aber geringe Latenz (wenige Zyklen)
- L2-Cache: größer als L1-Cache (hier: inklusive) aber höhere Latenz
- Background-memory: Haupspeicher der Systems, höchste Latenz
- Inklusivität: alle Adressen, die in L1 speichern auch in L2 aber nicht umgekehrt.

Assoziativität

- Direkt abgebildeter Cache: Jeder Speicherblock verwaltet eine Speicheradresse (auch einfach assoziativ genannt)
- Alternativ: höhere Assoziativität erlaubt effizientere Cache-Nutzung aber höhere Kosten

Direkt abgebildeter Cache

- Jeder Speicherblock hat genau eine mögliche Position im Cache.
- Einfacher Aufbau und schnelle Zugriffszeiten.
- Nachteile: Höhere Konfliktwahrscheinlichkeit und niedrigere Trefferquote im Vergleich zu assoziativen Caches.

Write-Through Cache

- Write-Through: Daten werden gleichzeitig in den Cache und den Hauptspeicher geschrieben.
- Vorteile: Daten im Hauptspeicher sind immer aktuell.
- Nachteile: kumulative Latenz bei Schreiboperationen.

Daten- und Adressbus

- Datenbus und Adressbus sind 4 Byte breit.
- Der Hauptspeicher ist jedoch byte-adressiert.

Literaturrecherche

- Übliche Cachegrößen: L1 (32KB), L2 (256KB)
- Latenzen: L1 (3 - 5 Zyklen), L2 (10 - 20 Zyklen)
- Vergleich der eigenen Implementierung mit den Ergebnissen