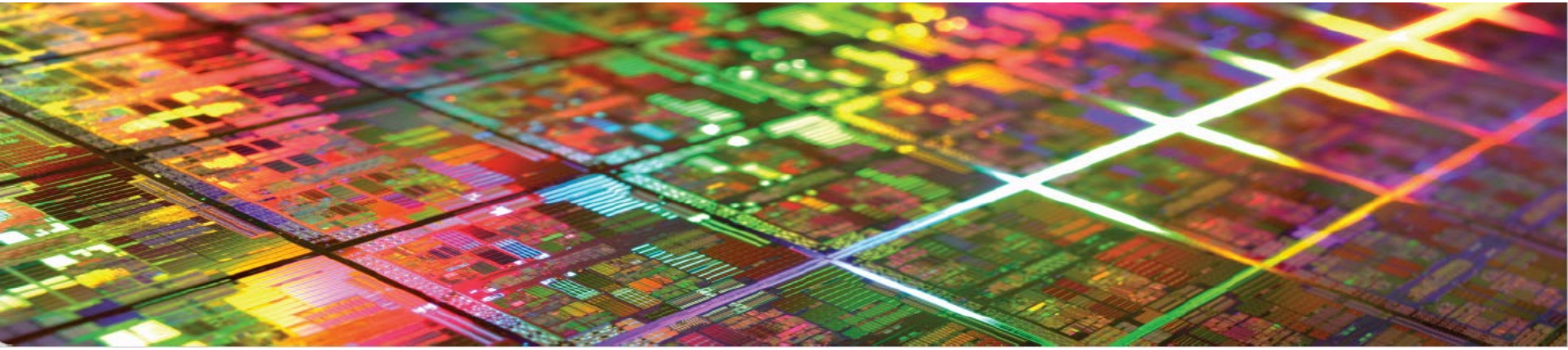
Karlsruher Institut für Technologie

Rechnerorganisation

Prof. Dr. Wolfgang Karl

Vorlesung im Wintersemester 2025/2026 – Foliensatz: RO25-FS10

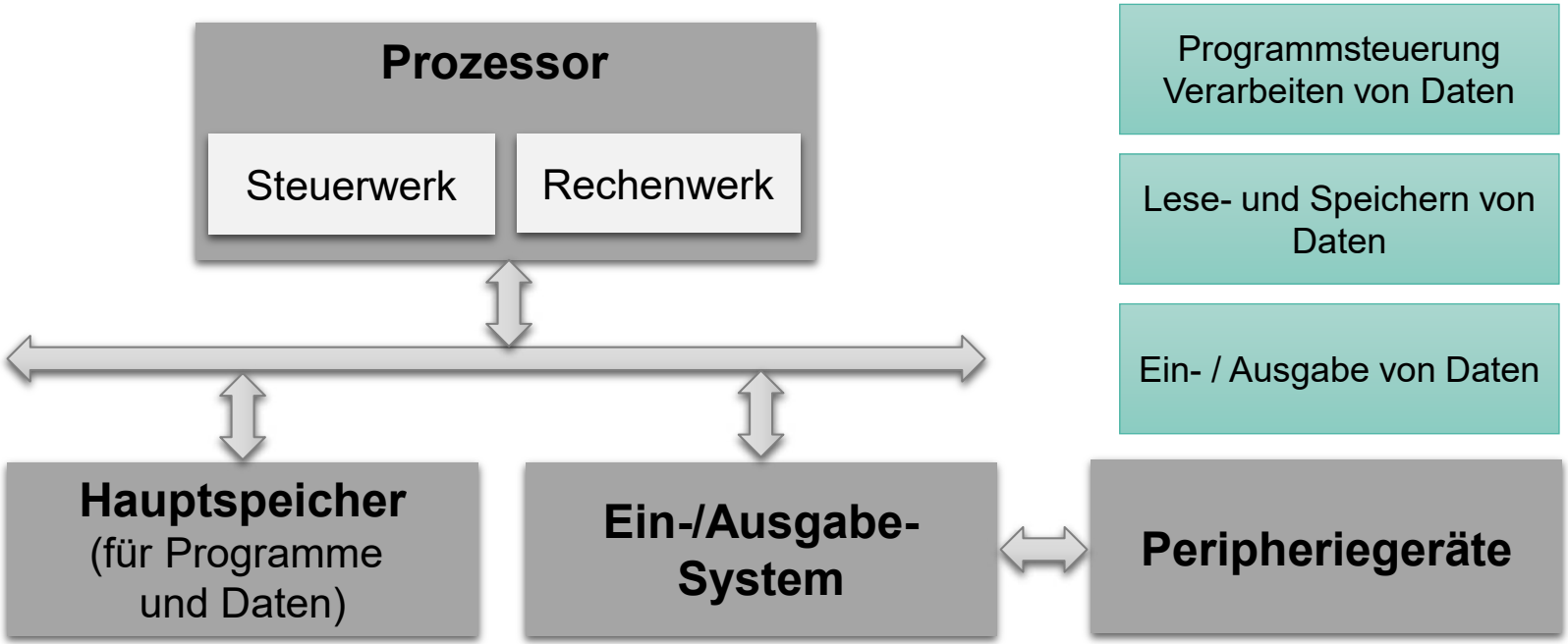


Kapitel 8
Cache-Speicher

- **Speicherhierarchie**
- Systemaufbau mit Cache-Speicher
- Grundlegende Arbeitsweise
- Cache-Organisationsformen
- Grundlegende Fragen beim Entwurf
- Gültigkeitsproblem

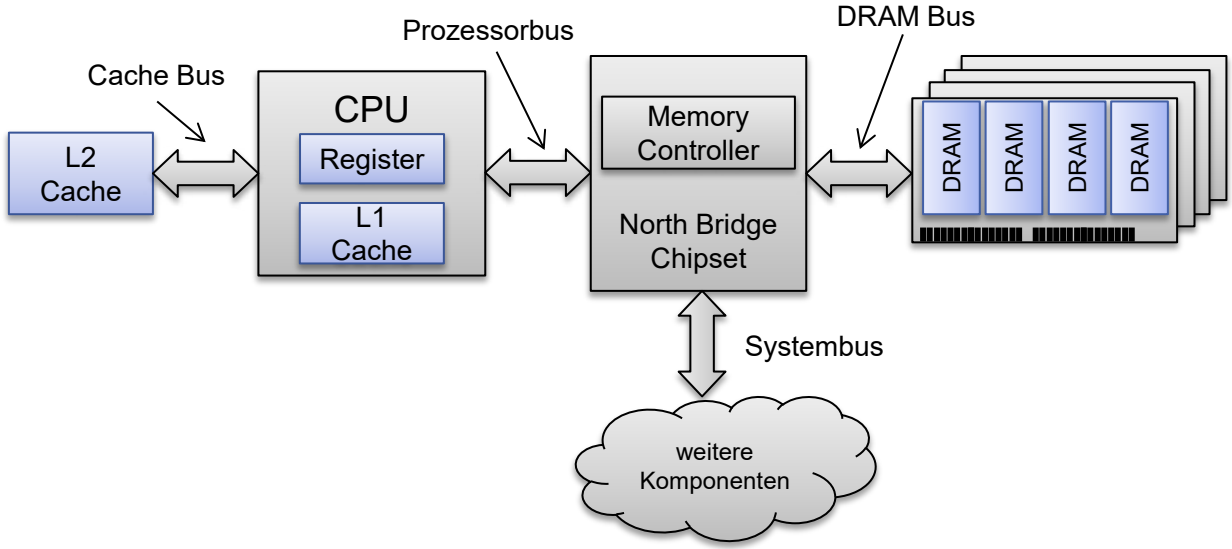
Programmierbarer Universalrechner

- **Einfaches Modell**



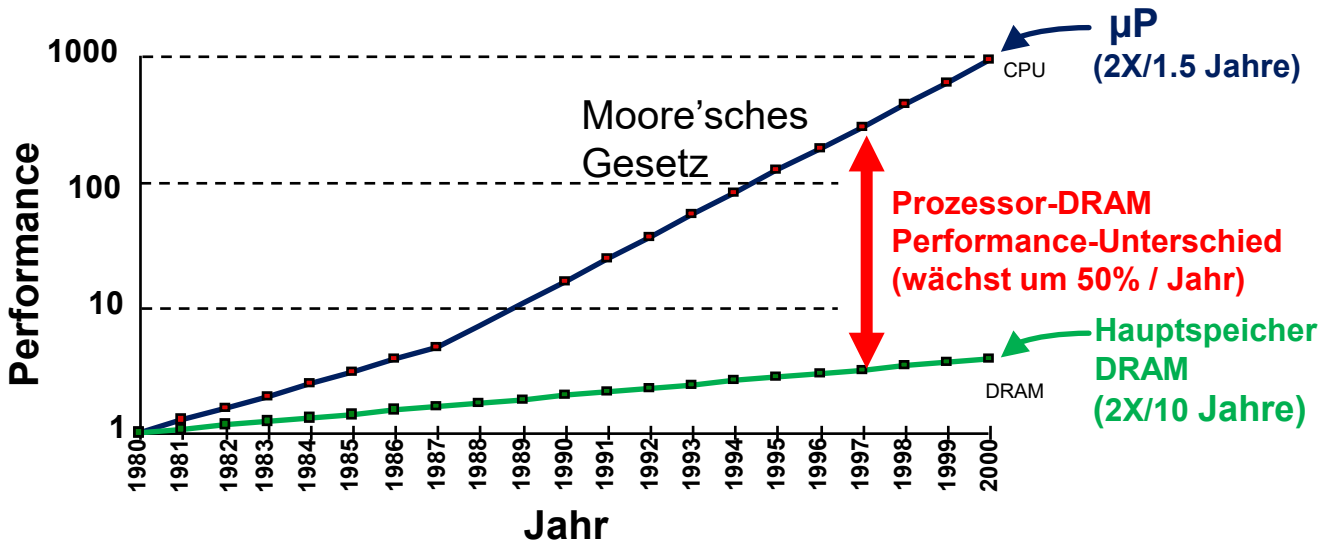
Programmierbarer Universalrechner

- **Beispiel: Aufbau eines klassischen PCs**



Programmierbarer Universalrechner

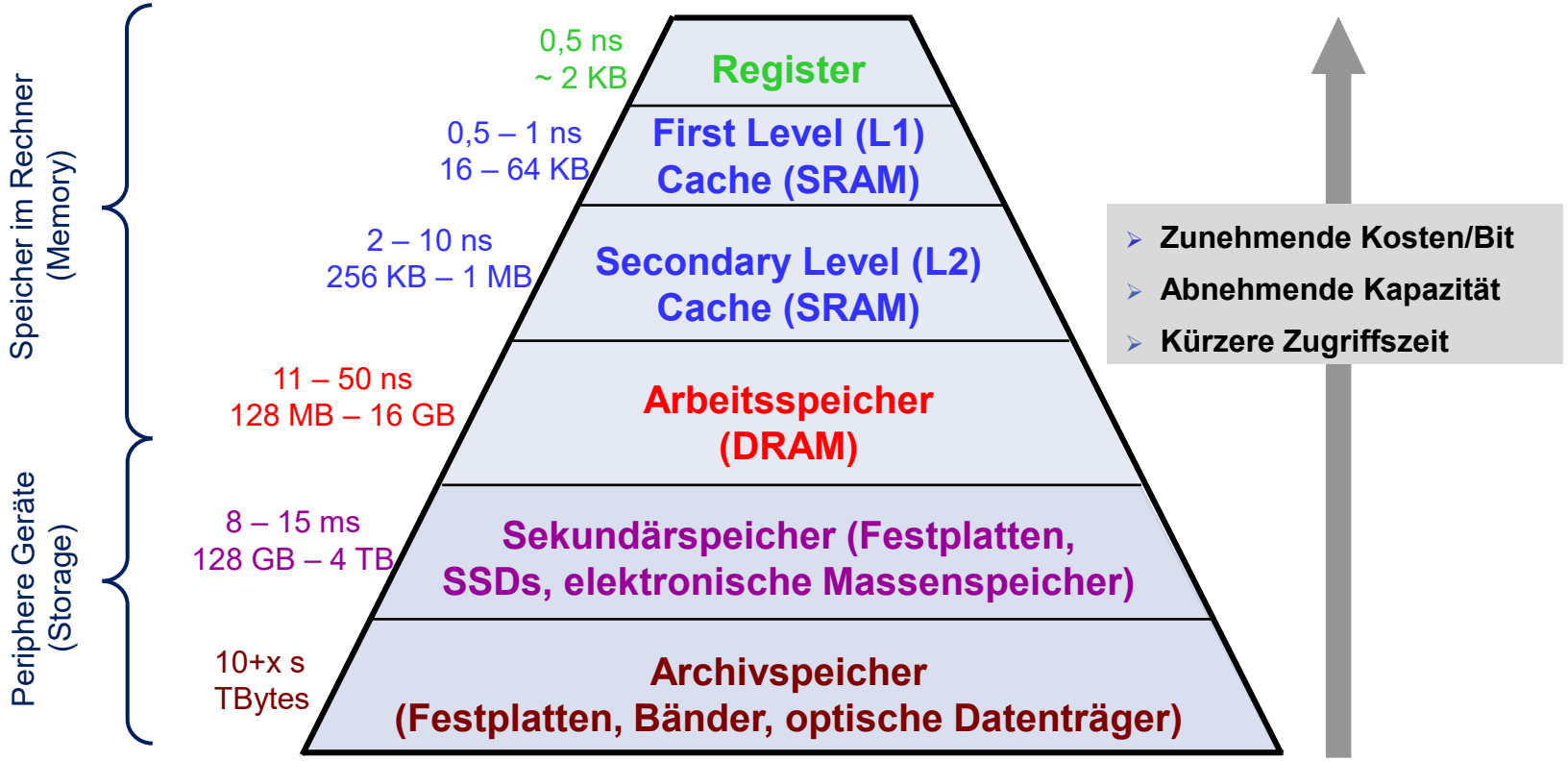
- **Leistungsbetrachtung:**
  - Vergleich Verarbeitungsgeschwindigkeit des Prozessors und Zugriffsgeschwindigkeit eines DRAM-Speichers



Programmierbarer Universalrechner

- **Leistungsbetrachtung:**
  - **Unterschiedliche Verarbeitungsgeschwindigkeit des Prozessors und Zugriffsgeschwindigkeit eines DRAM-Speichers**
    - Immer größer werdende Lücke zwischen Verarbeitungsgeschwindigkeit von Prozessoren und Zugriffsgeschwindigkeit der DRAM-Speicherchips des Hauptspeichers
    - Ein technologisch einheitlicher Speicher mit kurzer Zugriffszeit und großer Kapazität ist aus Kostengründen i.A. nicht realisierbar
    - Lösung: Hierarchische Anordnung verschiedener Speicher und Verschiebung der Information zwischen den Schichten (**Speicherhierarchie**)

Speicherhierarchie

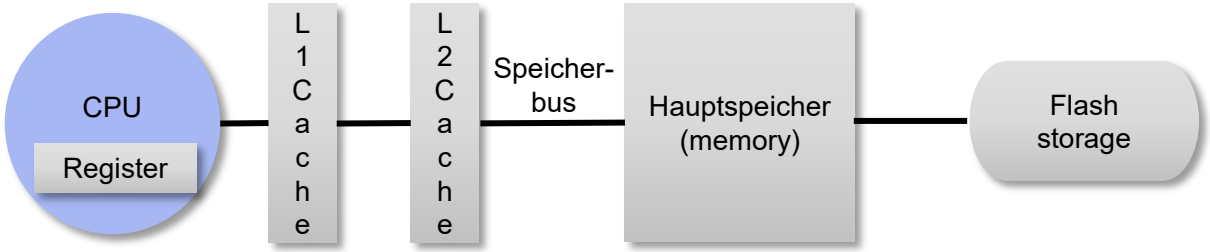


Speicherhierarchie

- **Ebenen der Speicherhierarchie**
  - Speicherkomponenten auf den verschiedenen Ebenen haben unterschiedliche Kapazitäten und Zugriffszeiten.
  - Ein schneller Speicher hat gegenüber einem langsamen Speicher höhere Kosten pro Bit und deshalb auch eine kleinere Kapazität.
  - Der schnelle Speicher auf Ebene i (höhere Ebene der Speicherhierarchie) ist näher am Prozessor als der langsame Speicher auf Ebene i+1
  - Ziel ist, dem Benutzer möglichst viel Speicher der billigeren Technologie bereitzustellen, während der Zugriff mit der vom schnellsten Speicher gebotenen Geschwindigkeit erfolgt soll.

Speicherhierarchie

- **Leistungsbetrachtung**
  - **Beispiel: Personal Mobil Devices (PMD)**

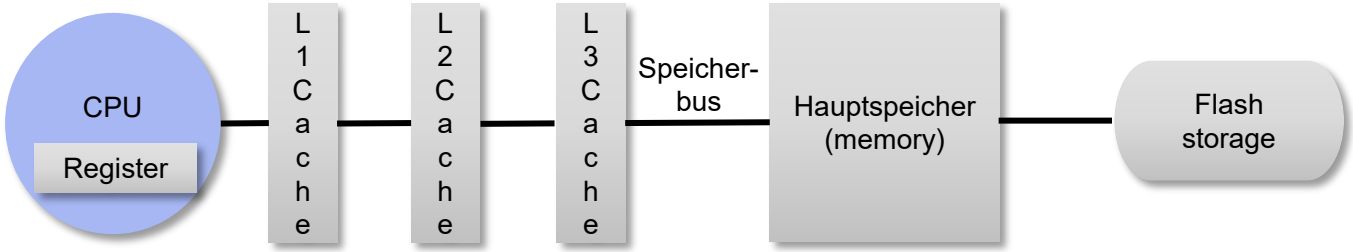


	Register	Level 1 Cache	Level 2 Cache	Hauptspeicher	Flash-Speicher
Kapazität:	1000 Bytes	64 KB	256 KB	1 -2 GB	4 – 64 GB
Geschwindigkeit:	300 ps	1 ns	5 -10 ns	50 – 100 ns	25 - 50µs

Quelle: Hennessy, J.; Patterson, D.: Computer Architecture A Quantitative Approach. Morgan Kaufman Publishers, 6th Edition, 2019

Speicherhierarchie

- **Leistungsbetrachtung**
  - **Beispiel: Laptop, Desktop**

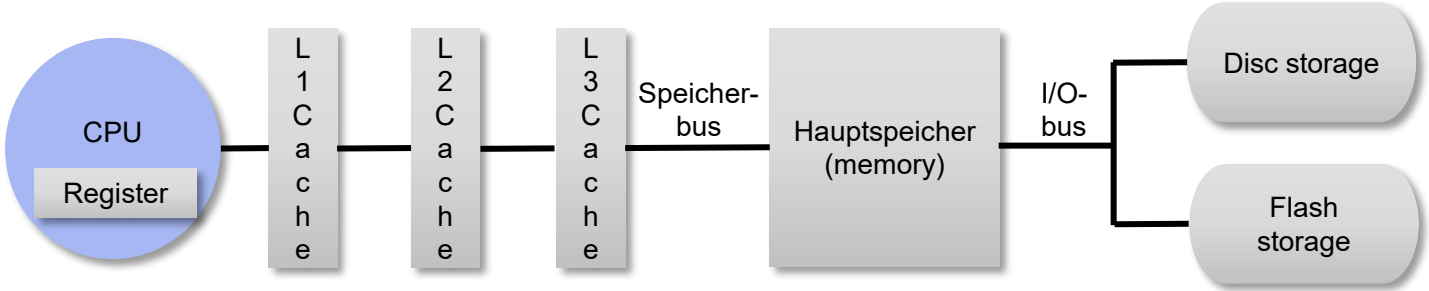


	Register	Level 1 Cache	Level 2 Cache	Level 3 Cache	Hauptspeicher	Flash-Speicher
Laptop						
Kapazität:	1000 Bytes	64 KB	256 KB	4 - 8 MB	4 – 16 GB	256 GB - 1 TB
Geschwindigkeit:	300 ps	1 ns	3 -10 ns	10 -20 ns	50 – 100 ns	50 – 100 µs
Desktop						
Kapazität:	2000 bytes	64 KB	256 KB	8 – 32 MB	8 – 64 GB	256 GB – 1 TB
Geschwindigkeit:	300 ps	1 ns	3 -10 ns	10 -20 ns	50 – 100 ns	50 – 100 µs

Quelle: Hennessy, J.; Patterson, D.: Computer Architecture A Quantitative Approach. Morgan Kaufman Publishers, 6th Edition, 2019

Speicherhierarchie

- **Leistungsbetrachtung**
  - **Beispiel: Server**



Kapazität:	4000 bytes	64 KB	256 KB	16 -64 MB	32 -256 GB	16 -64 TB	1 -16 TB
Geschwindigkeit:	200 ps	1 ns	3 -10 ns	10 -20 ns	50 – 100 ns	5 – 10 ms	100 – 200 µs

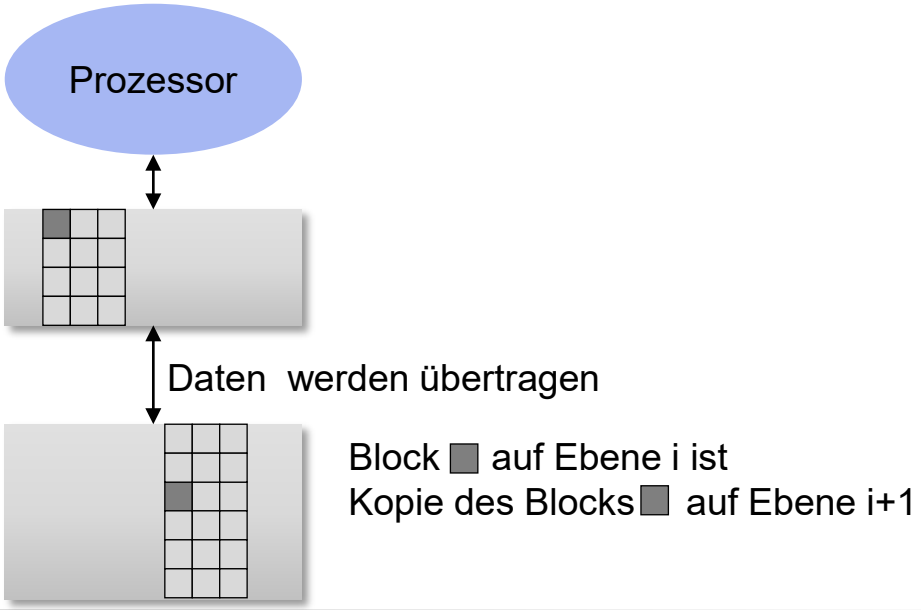
Quelle: Hennessy, J.; Patterson, D.: Computer Architecture A Quantitative Approach. Morgan Kaufman Publishers, 6th Edition, 2019

Speicherhierarchie

- **Zusammenspiel verschiedener Speicherkomponenten mit unterschiedlichen Technologien**
  - Die Speicherhierarchie wirkt für den Benutzer wie ein großer und schneller Speicher.
  - Daten werden nur jeweils gleichzeitig zwischen zwei benachbarten Ebenen übertragen.
  - Ein **Block** ist die kleinste Informationseinheit, die in der zweistufigen Hierarchie vorhanden oder nicht vorhanden sein kann.

Speicherhierarchie

- **Zusammenspiel verschiedener Speicherkomponenten mit unterschiedlichen Technologien**
  - Datentransfer zwischen zwei Ebenen der Speicherhierarchie



Speicherhierarchie

- **Datentransfer zwischen zwei Ebenen der Speicherhierarchie**
  - **Terminologie**
    - **Treffer (hit)**
      - Die vom Prozessor geforderten Daten sind in einem Block, der im Speicher auf der oberen Ebene vorhanden ist.
    - **Fehlzugriff (miss)**
      - Die vom Prozessor gefordeten Daten werden nicht im Speicher auf der oberen Ebene gefunden
      - Es erfolgt ein Zugriff auf den Speicher der untere Ebene in der Hierarchie, um den Block mit den angeforderten Objekten zu finden

Speicherhierarchie

- **Datentransfer zwischen zwei Ebenen der Speicherhierarchie**
  - **Terminologie**
    - **Trefferrate (hit rate)**
      - Anteil der Speicherzugriffe, bei denen der gesuchte Block im Speicher auf einer Ebene gefunden wird
    - **Fehlzugriffsrate (miss rate)**
      - Anteil der Speicherzugriffe, bei denen der gesuchte Block nicht im Speicher auf einer Ebene gefunden wird

Speicherhierarchie

- **Datentransfer zwischen zwei Ebenen der Speicherhierarchie**
  - **Terminologie**
    - **Zugriffszeit bei Treffer (hit time)**
      - Die Zeit für den Zugriff auf den Speicher der oberen Ebene und
      - die Zeit, die benötigt wird, um festzustellen, ob der Zugriff ein Treffer ist oder nicht.
    - **Fehlzugriffsaufwand (miss penalty)**
      - Die Zeit, die benötigt wird, um einen Block von der unteren Ebene in die obere Ebene der Speicherhierarchie zu laden;
        - Umfasst die Zeit für die Übertragung und das Einfügen des Blocks in den Speicher der höheren Ebene, auf der der Fehlzugriff stattgefunden hat.

Speicherhierarchie

- **Ausnützen des Lokalitätsprinzips bei der Programmabarbeitung**
  - Programme greifen zu jedem Zeitpunkt nur auf einen kleinen Bereich ihres Adressraums zu;
  - **Zeitliche Lokalität (temporal locality)**
    - Tendenz des Prozessors, auf Objekte zuzugreifen, auf die er erst kürzlich zugegriffen hat
    - Beispiel: Schleifeniterationen: Prozessor greift wiederholt auf dieselbe Befehlsfolge zu
    - Beispiel: häufig benutze Variablen
  - **Räumliche Lokalität (spatial locality)**
    - Tendenz des Prozessors, dass, wenn er auf ein Objekt zugegriffen hat, er auch auf Objekte zugreift, deren Adressen nahe der des zugegriffenen Objekts sind
    - Beispiel: sequentielle Programmausführung
    - Beispiel: Zugriff auf Datenstrukturen wie Felder

Speicherhierarchie

- **Ausnützen des Lokalitätsprinzips bei der Programmabarbeitung**
  - **Virtueller Speicher**
    - Vorspiegelung eines Hauptspeichers großer Kapazität
      - Kapazität des physikalischen Speichers ist begrenzt
      - Kapazitätserweiterung durch Hintergrundspeicher
    - Bereitstellen der zu einem Zeitpunkt benötigten Objekte im Hauptspeicher
      - Der Block mit den benötigten Daten wird vom Hintergrundspeicher als Kopie in den Hauptspeicher geladen.
    - Stellt jeweils einen großen und einheitlichen Adressraum für die einzelnen Prozesse bereit;
    - Die Verwaltung ist Aufgabe des Betriebssystems und wird vom Prozessor unterstützt.

Speicherhierarchie

- **Ausnützen des Lokalitätsprinzips bei der Programmabarbeitung**
  - **Cache-Speicher**
    - Kleiner schneller Speicher, der als Puffer für einen langsameren, aber größeren Speicher dient;
      - Enthält Kopien der Daten im Hauptspeicher;
    - Ist mit SRAM-Bausteinen aufgebaut