



Foliensatz Wirtschaftsinformatik

Lehrveranstaltungsinhalte

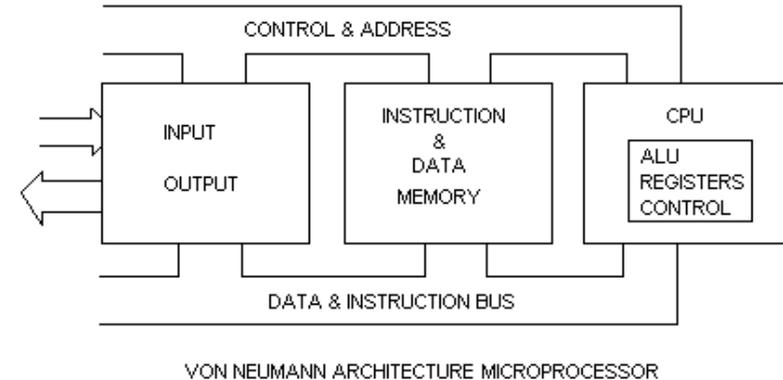
Termine	Inhalte	Kapitel	Textbuch-Seiten
1. LV-Termin	<ul style="list-style-type: none"> • Einführung • Informationssysteme in Wirtschaft und Gesellschaft • Geschäftsprozessmanagement 	1, 2	1 – 56 57 – 96
2. LV-Termin	<ul style="list-style-type: none"> • Modellierung betrieblicher Informationssysteme • Unterstützung betrieblicher Leistungsprozesse durch ERP-Systeme 	3, 4	97 – 134 135 – 188
3. LV-Termin	<ul style="list-style-type: none"> • Außenwirksame Informationssysteme und Electronic Commerce • Managementunterstützungs-systeme 	5, 6	189 – 266 267 – 316
4. LV-Termin	<ul style="list-style-type: none"> • Planung, Entwicklung und Betrieb von Informationssystemen • Informationssicherheit und Datenschutz 	7, 8	317 – 368 369 – 422
5. LV-Termin	<ul style="list-style-type: none"> • Datenspeicherung • Rechnersysteme 	9, 10	423 – 494 495 – 539

Kapitel 10

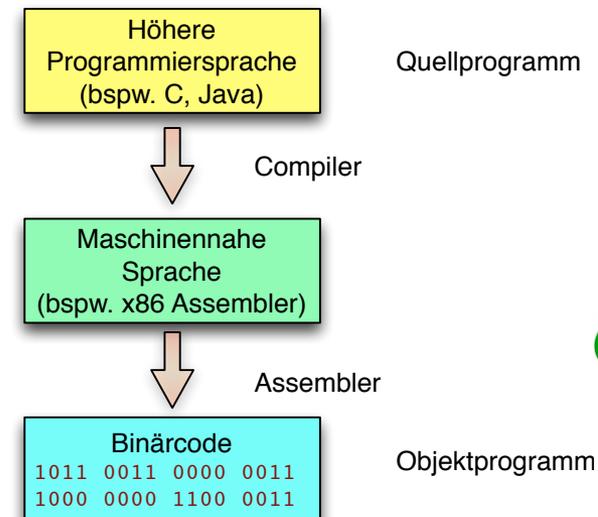
Rechnersysteme

Wiederholungsfragen Kapitel 10

- Was war bei der Einführung der von-Neumann-Architektur das Besondere?

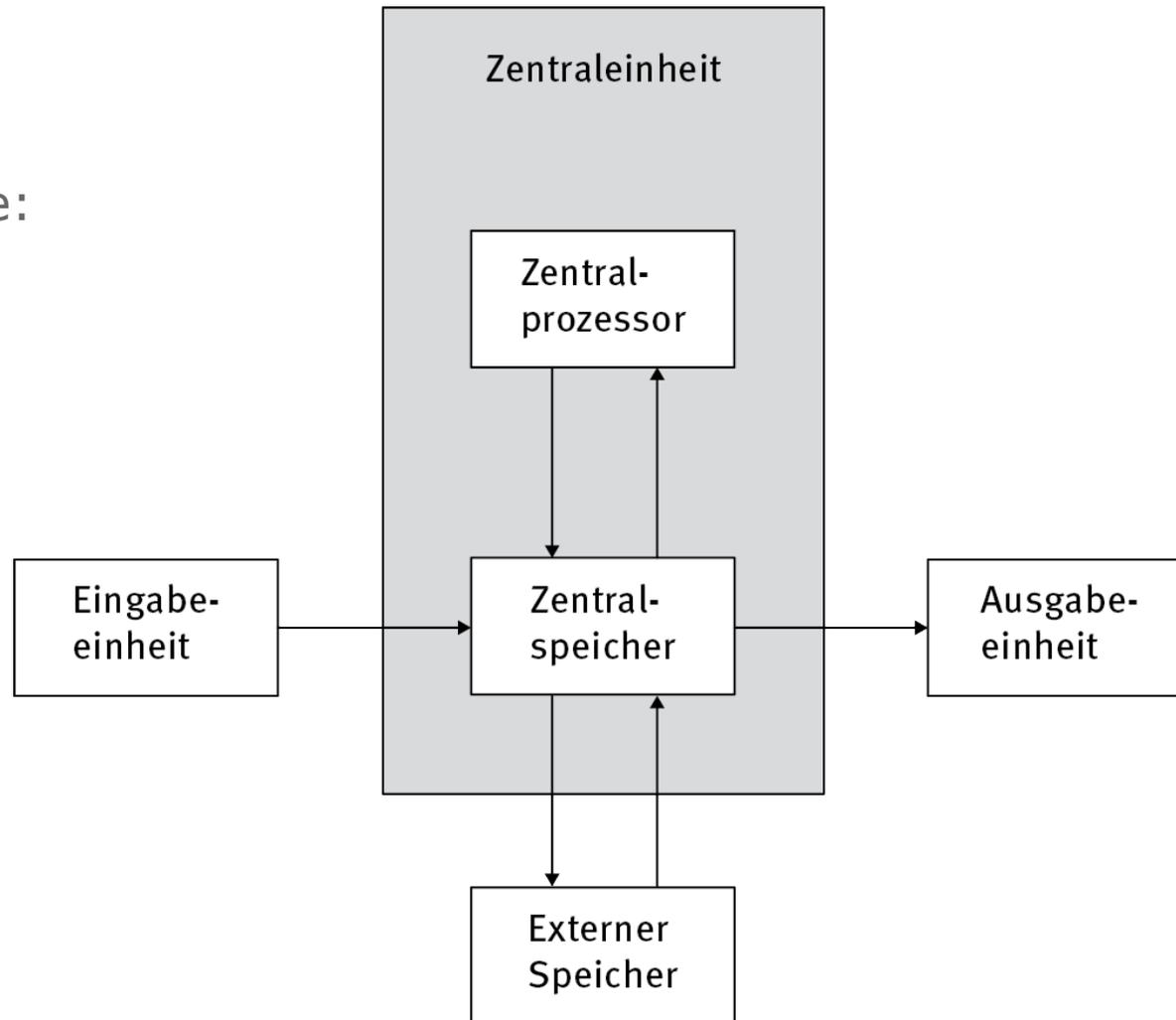


- Was ist der Unterschied zwischen einer maschinennahen Programmiersprache und einer höheren Programmiersprache?



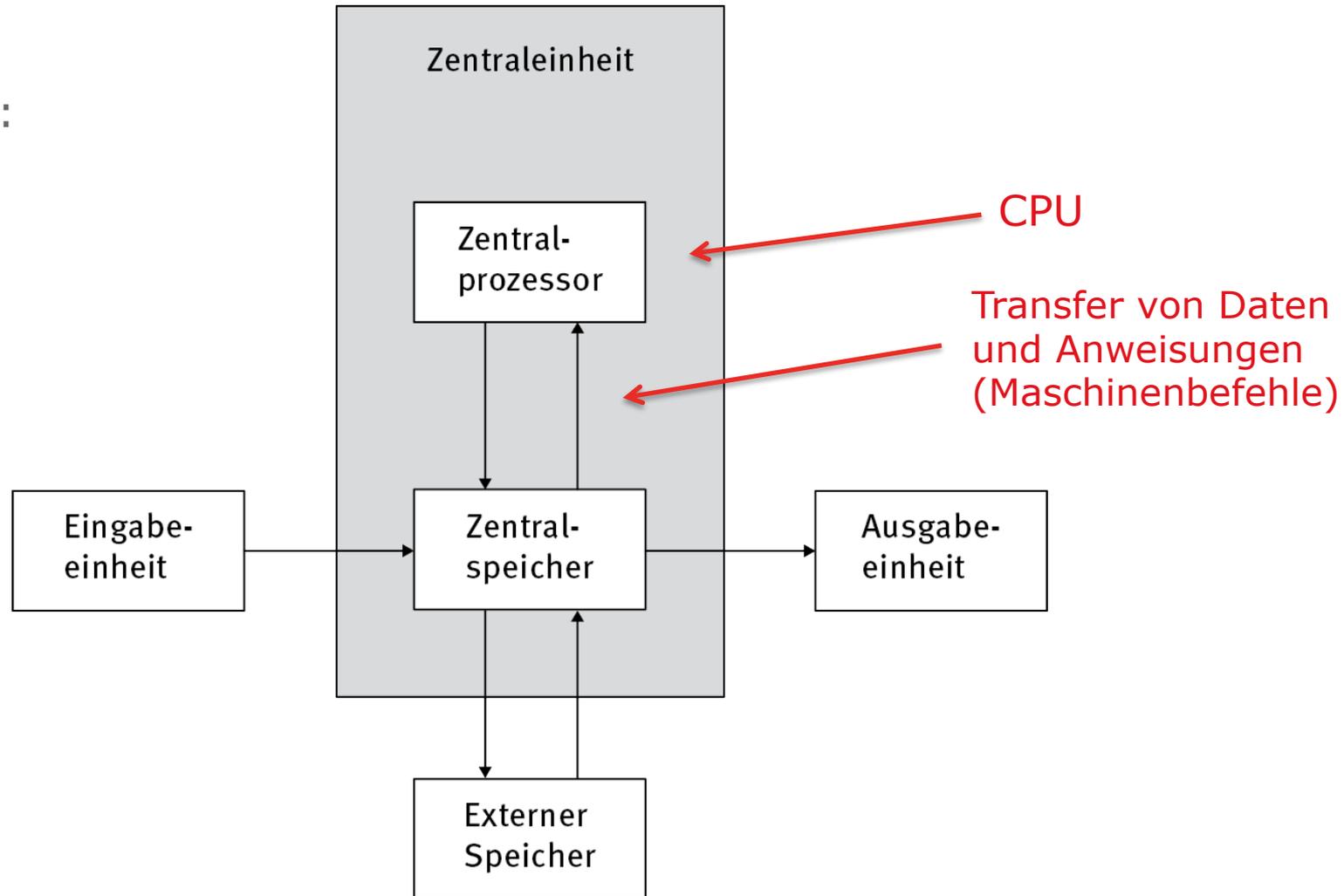
Funktionseinheiten eines Rechners

Hardware:



Funktionseinheiten eines Rechners

Hardware:



Prozessoren und deren Funktionen

- Ein **Prozessor** (CPU) ist eine Funktionseinheit, die für die Durchführung von **Maschinenbefehlen** sorgt.
- Durch den Prozessor werden *Daten und Befehle aus dem Zentralspeicher geladen*, es werden die *Befehle decodiert und ausgeführt*, wodurch *Ausgabedaten* erzeugt werden.
- **Maschinenbefehle:**
 - *arithmetische Befehle* (Addieren, Subtrahieren, Multiplizieren usw.)
 - *logische Befehle* (Vergleichen, Verknüpfen usw.)
 - *Datentransferbefehle* (Übertragen, Verschieben usw.)
 - *Ein- und Ausgabebefehle* (Lesen, Schreiben usw.)

Beispiel für Maschinenbefehle

- Annahme: Prozessor hat ein Register (Speicherbereich im Prozessor in der Größe eines Maschinenworts) namens „BL“
- Addition der Werte „3“ und „4“ im Register „BL“

```
MOV BL, 3           ; lade die Konstante „3“ nach BL (Datentransfer)
ADD BL, 4           ; addiere zu dem Wert die Konstante „4“ (arithm.B.)
...                ; weitere Befehle mit dem Ergebnis in BL
```

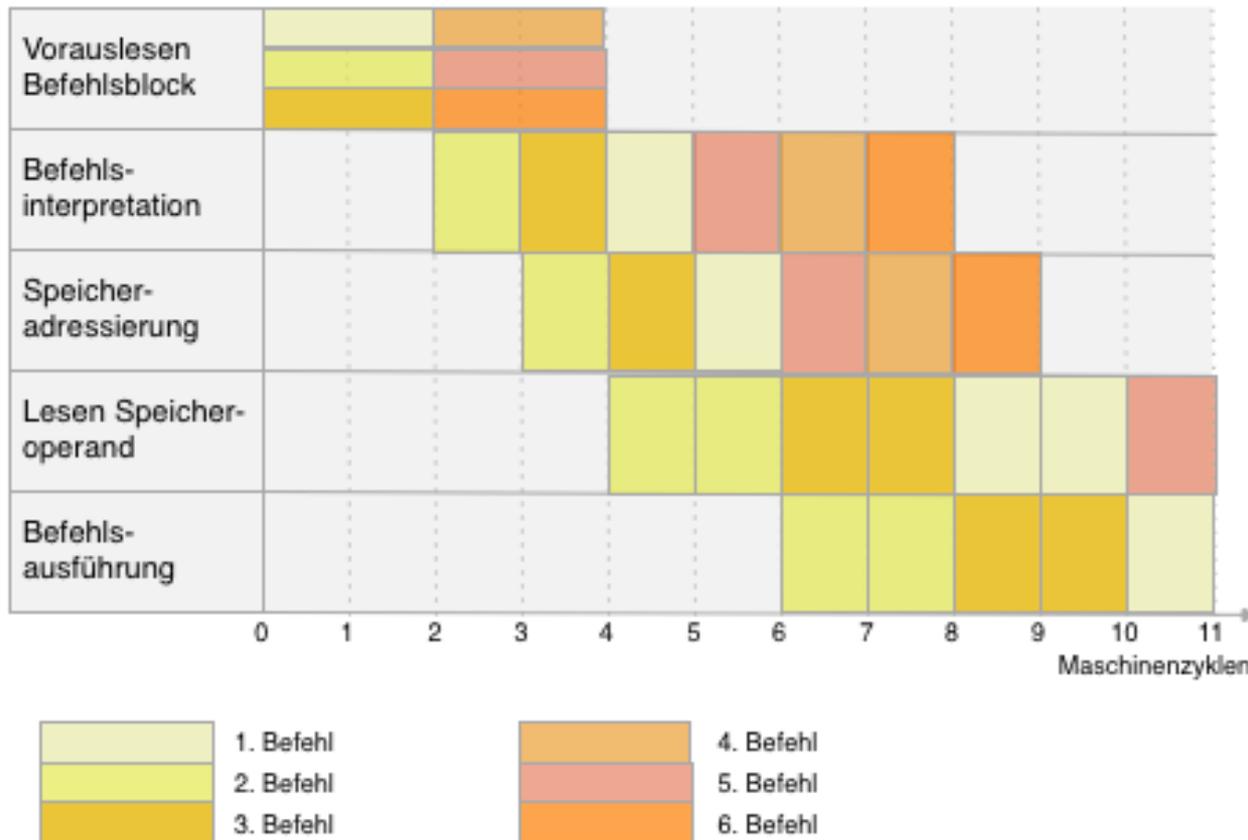
- Die gleichen Maschinenbefehle in Binärcode:

```
1011 0011 0000 0011
1000 0000 1100 0011 0000 0100
```

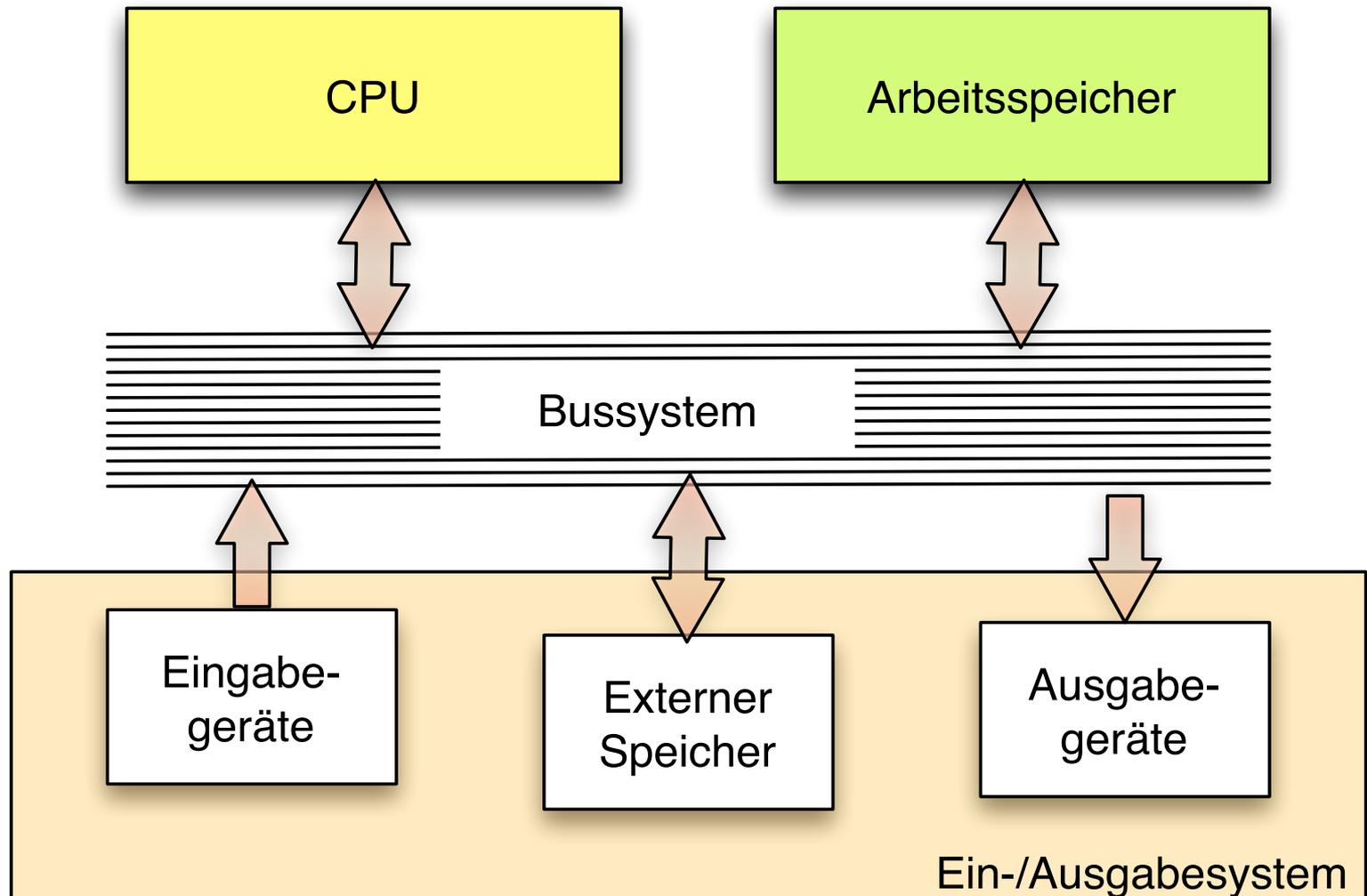
Ausführung von Maschinenbefehlen

- Ausführung von Maschinenbefehlen erfolgt in mehreren Stufen, bspw.:
 - Lesen von Instruktionen aus dem Arbeitsspeicher
 - Decodieren der Befehle
 - Adressierung von Speicherinhalten für Operanden der Befehle
 - Transfer des/der Operanden
 - Ausführung des Befehls
- Stufen können verzahnt „am Fließband“ ausgeführt werden
- Teile des Prozessors können so **gleichzeitig** an der Ausführung von mehreren Maschinenbefehlen arbeiten

Parallele Ausführung am Prozessor durch Fließbandverarbeitung

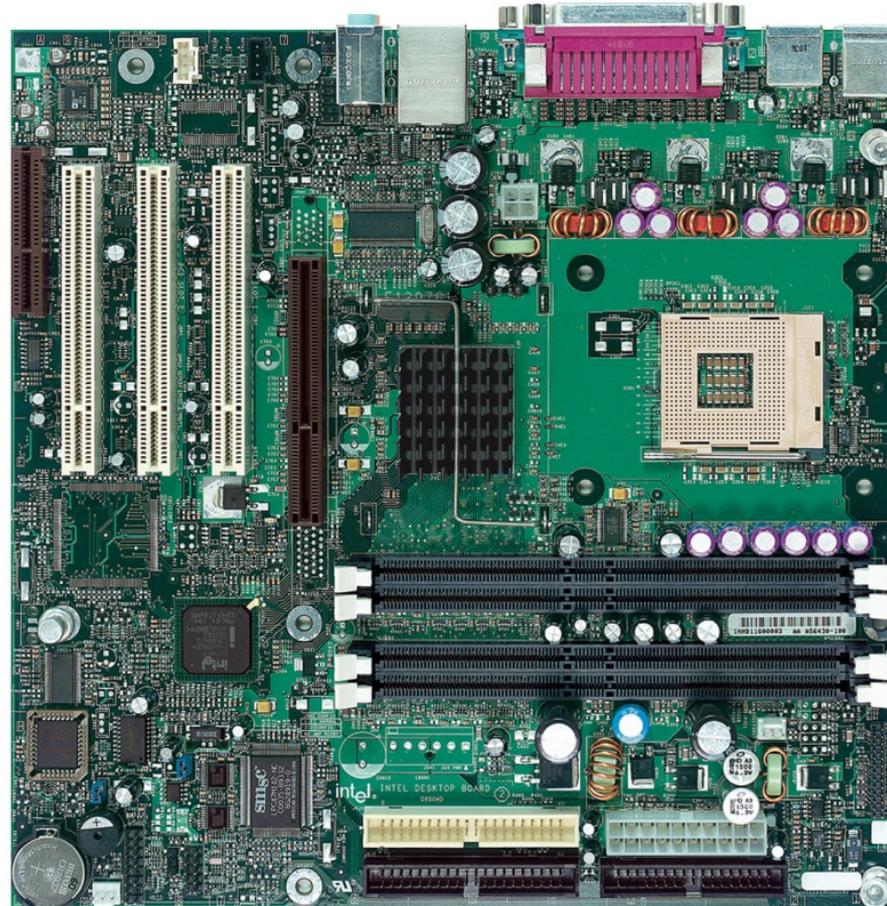


Verbindung von Funktionseinheiten durch ein Bussystem

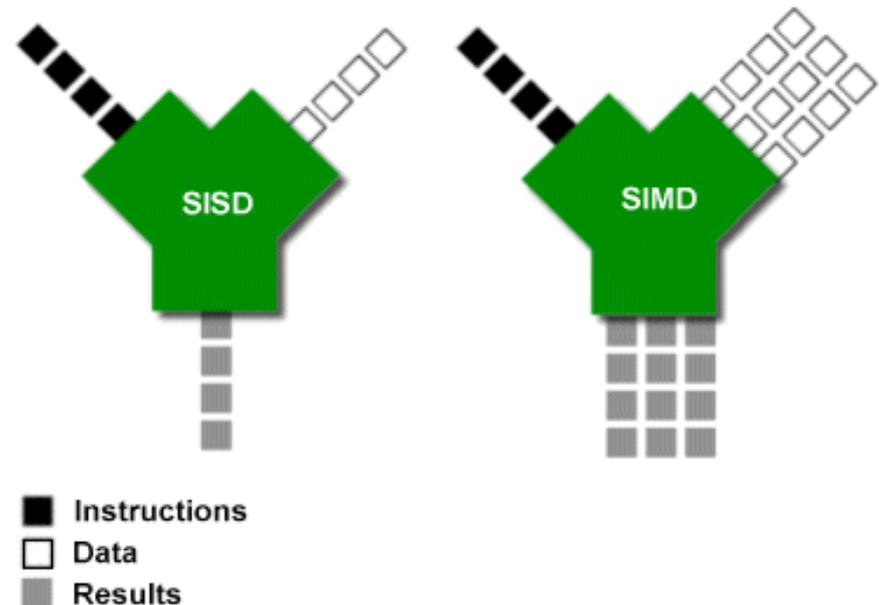
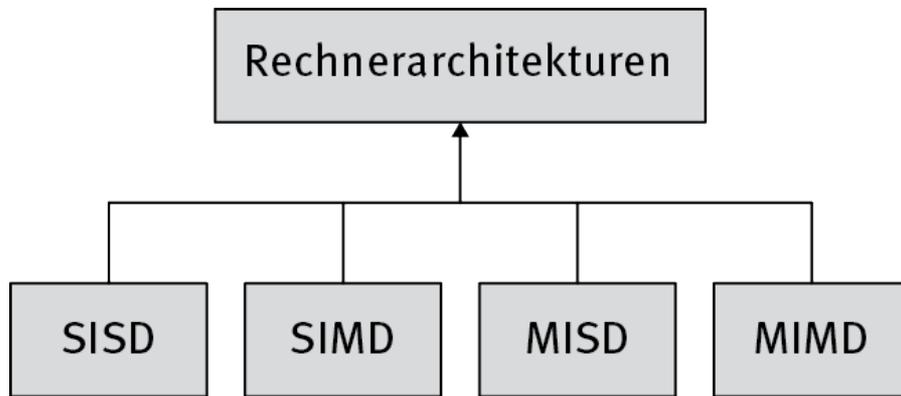


PC-Grundplatte (Motherboard)

12



Rechnerarchitekturen

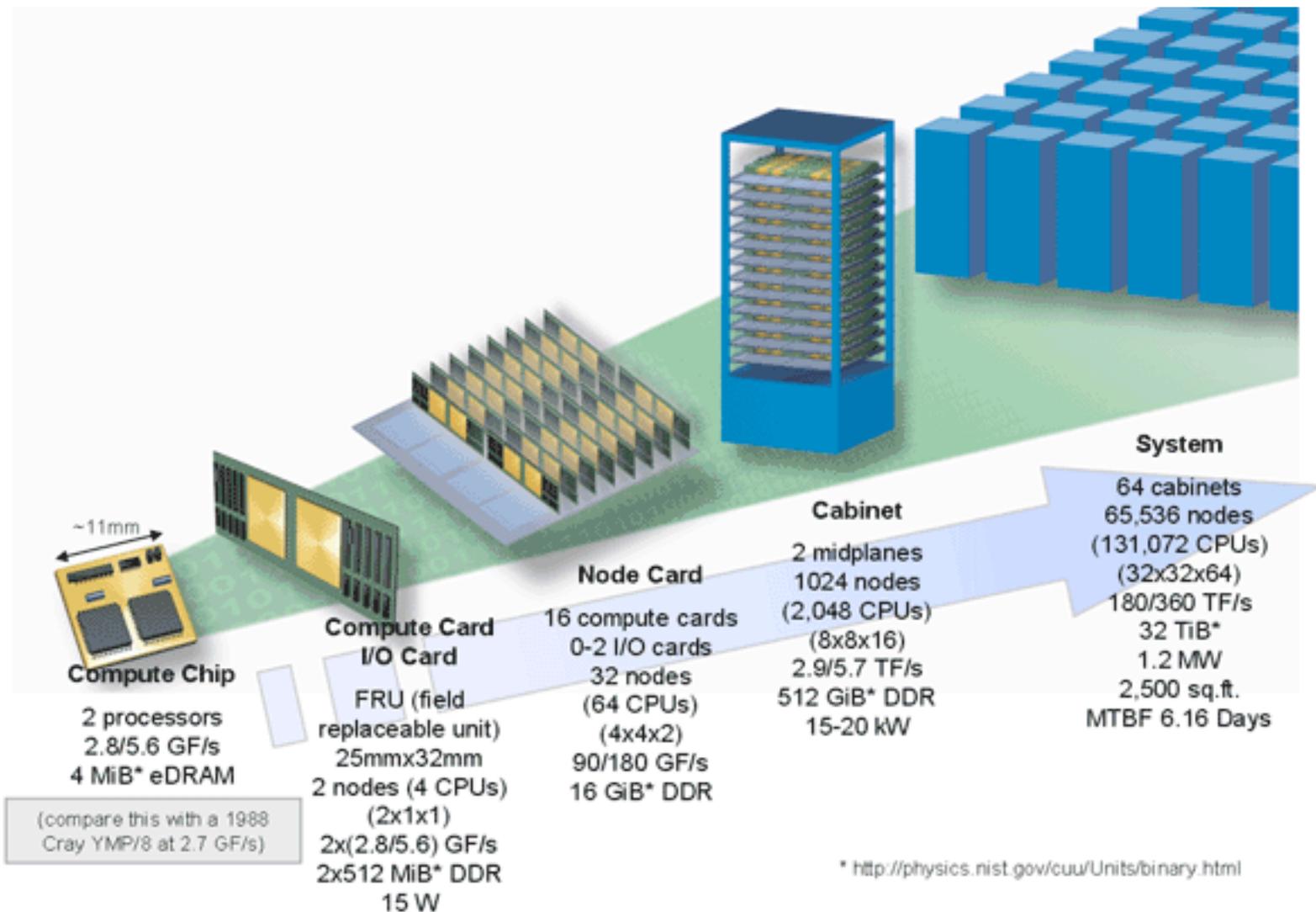


- Funktionseinheiten sind vielfach **mehrfach** ausgelegt
- Rechnerarchitektur bestimmt, ob und wie **Daten und/oder Programme** in einem System **parallel ausgeführt** werden können

Mehrkernprozessoren und Mehrprozessorsysteme

- **Mehrkernprozessor:** ein Prozessor, der zwei oder mehrere Prozessorkerne für die parallele Ausführung von Programmen enthält.
- **Prozessorkern:** eine Funktionseinheit innerhalb eines Prozessors, die unabhängig von anderen Prozessorkernen Programme ausführen kann.
- **Mehrprozessorsystem:** Rechnersystem, in dem mehrere Zentralprozessoren zusammenarbeiten. Es gibt Systeme, bei denen wenige (zwei bis 64) Prozessoren *eng gekoppelt einen gemeinsamen Arbeitsspeicher* benutzen, und Systeme, bei denen *einige oder viele Prozessoren lose gekoppelt* über jeweils *eigene Arbeitsspeicher* verfügen.

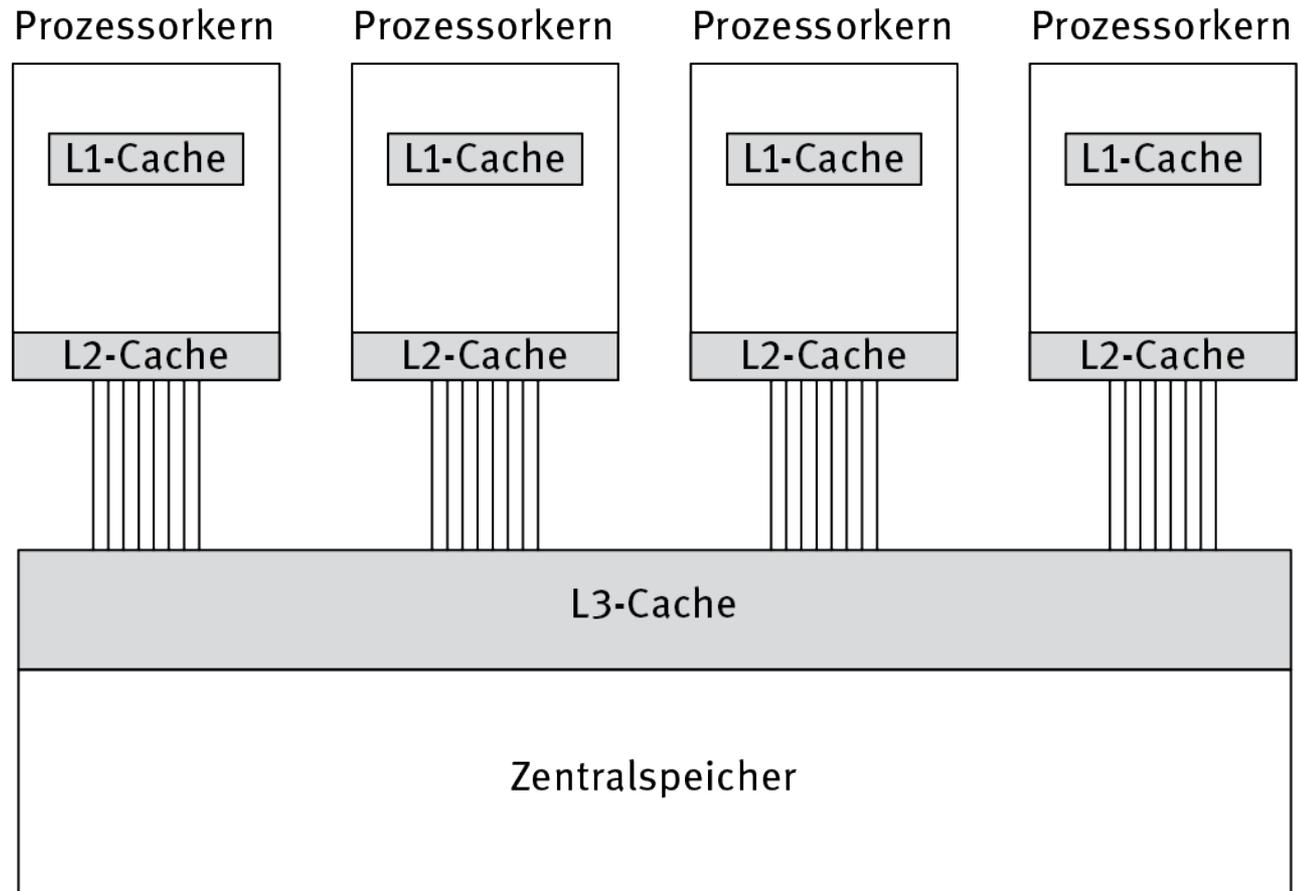
Beispiele für Mehrprozessorsysteme



Cache-Hierarchie

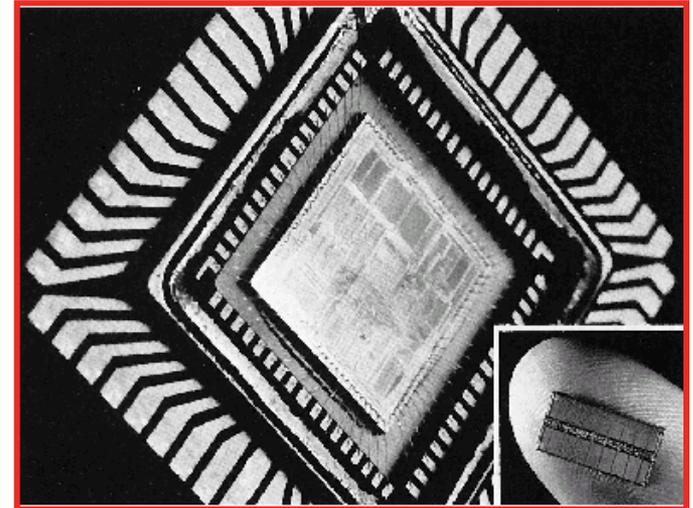
Cache:

- Pufferspeicher zwischen Zentralspeicher und Prozessorkern
- L1: direkt im Prozessorkern integriert

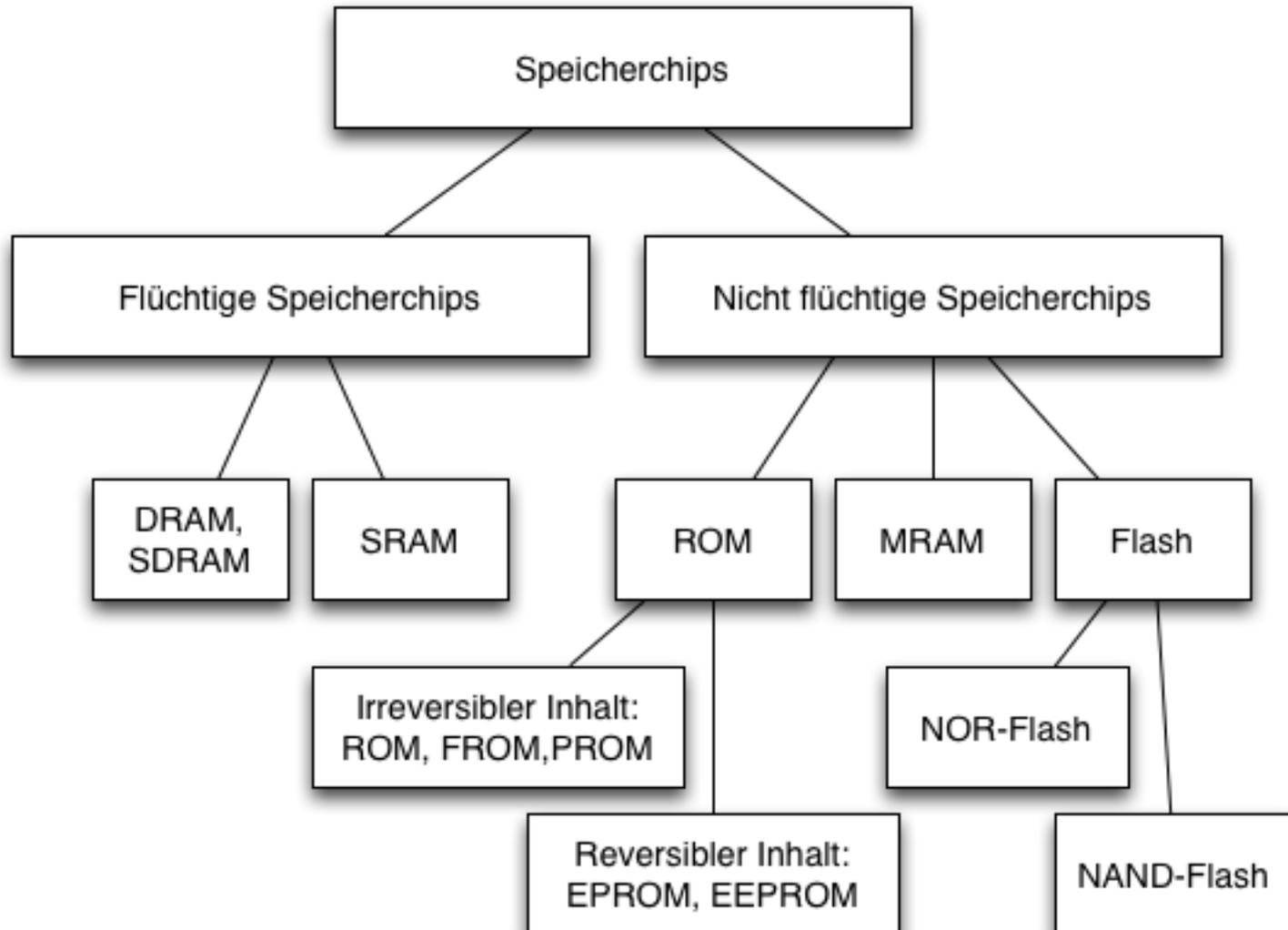


Bauelemente (Chips)

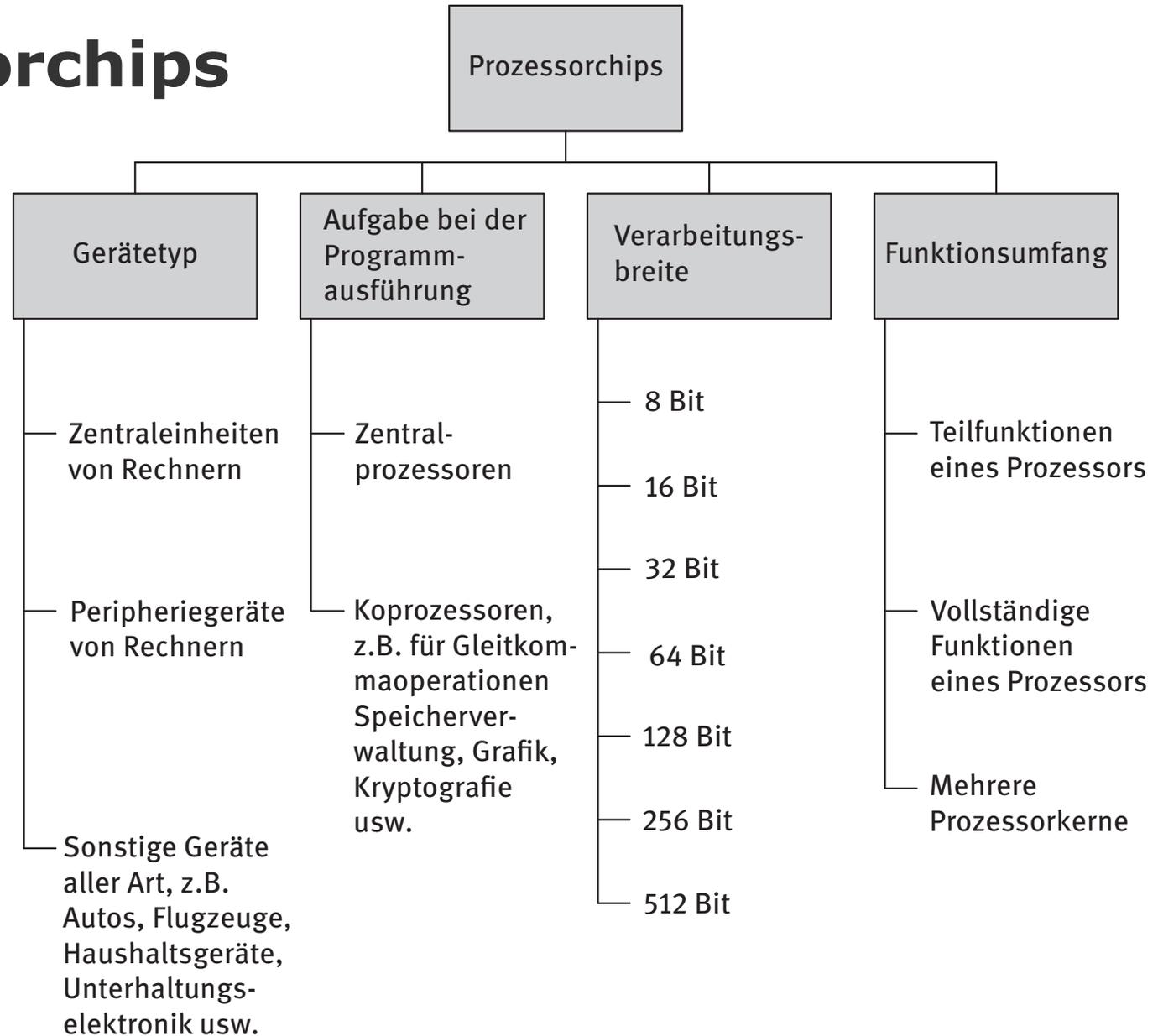
- Halbleiterplättchen (meist aus Silizium)
- Fläche (engl.: die size): meist 50-500 mm²
- Enthält Tausende bis Milliarden elektronische Bauelemente (Widerstände, Dioden und Transistoren) ...
- ... für Logik- und/oder Speicherfunktionen
- Wichtigsten Formen:
 - Speicherchips
 - Prozessorchips
 - Integrierte Speicher- und Prozessorchips
- Beispiel: Intel Broadwell-U (2015):
 - 82 – 133 mm²
 - 1,3 – 1,9 Mrd. Transistoren



Speicherchips



Prozessorchips



Mikroprozessoren

- **Mikroprozessor:** vollständiger Prozessor, der auf einem Chip untergebracht ist (Beispiel: Intel Core i7)
- Marktführer: Intel
- Typischer Einsatzbereich: PCs (Notebook, Desktop), PCs, Server, Großrechner



Ein-Chip-Computer

- Chip, der *alle Funktionseinheiten* eines Rechners integriert (engl.: system on a chip; abgekürzt: SoC)
 - Zentraleinheit (Zentralprozessor und Zentralspeicher)
 - Ein-/Ausgabekomponenten, Taktgeber
 - Schnittstellen zu externen Geräten (beispielsweise USB, Ethernet, ...)
 - AD/DA-Umwandler
 - drahtlose Sende- und Empfangseinrichtungen



Rechnerkategorien

Computer-Ära	Großrechner	Minicomputer	Arbeitsplatzrechner	Persönliche Info-Hilfsmittel	Computer-gestützte Geräte
Blütezeit	1955 - 1985	1965 - 1985	1980 - 2000	2000+	2010 +
Installationsort	Klimatisiertes Rechenzentrum	Raum in einer Fachabteilung	Schreibtisch	In der Hand oder am Körper getragen	In Dingen des täglichen Lebens verbaut
Preis in Euro	250.000 – 10.000.000	10.000 – 100.000 +	400 – 60.000 +	150 - 700	20 - 300
Nutzungszeit in Jahren	5 +	4 +	2 +	1 +	1 +
Verkaufte Einheiten pro Jahr	10000	100000	300 Million	1 Milliarde	100 Milliarden
Treibende Kräfte	Unternehmensweite Massen- und Routine-DV, Time-Sharing	Einsatz in technischen und kaufmännischen Bereichen (getrennt), Individualprogrammierung	Persönliche Rechenleistung am Arbeitsplatz und in privaten Haushalten	Inanspruchnahme von Internet-Diensten jederzeit und überall	Internet der Dinge, intelligente Fertigungs- und Vertriebssysteme, Sensornetze

Serverrechner

- Dauerbetrieb, meist *unterbrechungsfreie Stromversorgung* (Abkürzung: USV)
- Oft: leistungsfähige PCs in *Racks* oder *Bladesystemen*
- Im *Hochleistungsbereich*: oft spezielle Hardware für kostengünstigen Betrieb für 100.000+ Rechner

Amazon Perdex
Container



Learn@WU-
Server



Software

- Sammelbegriff für Computerprogramme
- Detaillierte *Handlungsanweisungen* an einen Prozessor
- Prozessor führt letztendlich *Maschinenbefehle* aus
- Erstellung komplexer Handlungsanweisungen in Maschinenbefehlen für den Menschen *aufwendig und fehleranfällig, prozessorabhängig*
- Deshalb: (höhere) Programmiersprachen

Maschinennahe Programmiersprachen

- Behalten Aufbau der Befehle der Maschinensprache bei
- Erlaubten *symbolische Ausdrücke* (Namen) für Befehle und Speicheradressen
- Beispiel: Addition der Werte „3“ und „4“ im Register „BL“

```
MOV BL, 3
```

```
ADD BL, 4
```

- Nach Übersetzung mittels eines *Assemblers* in Binärcode:

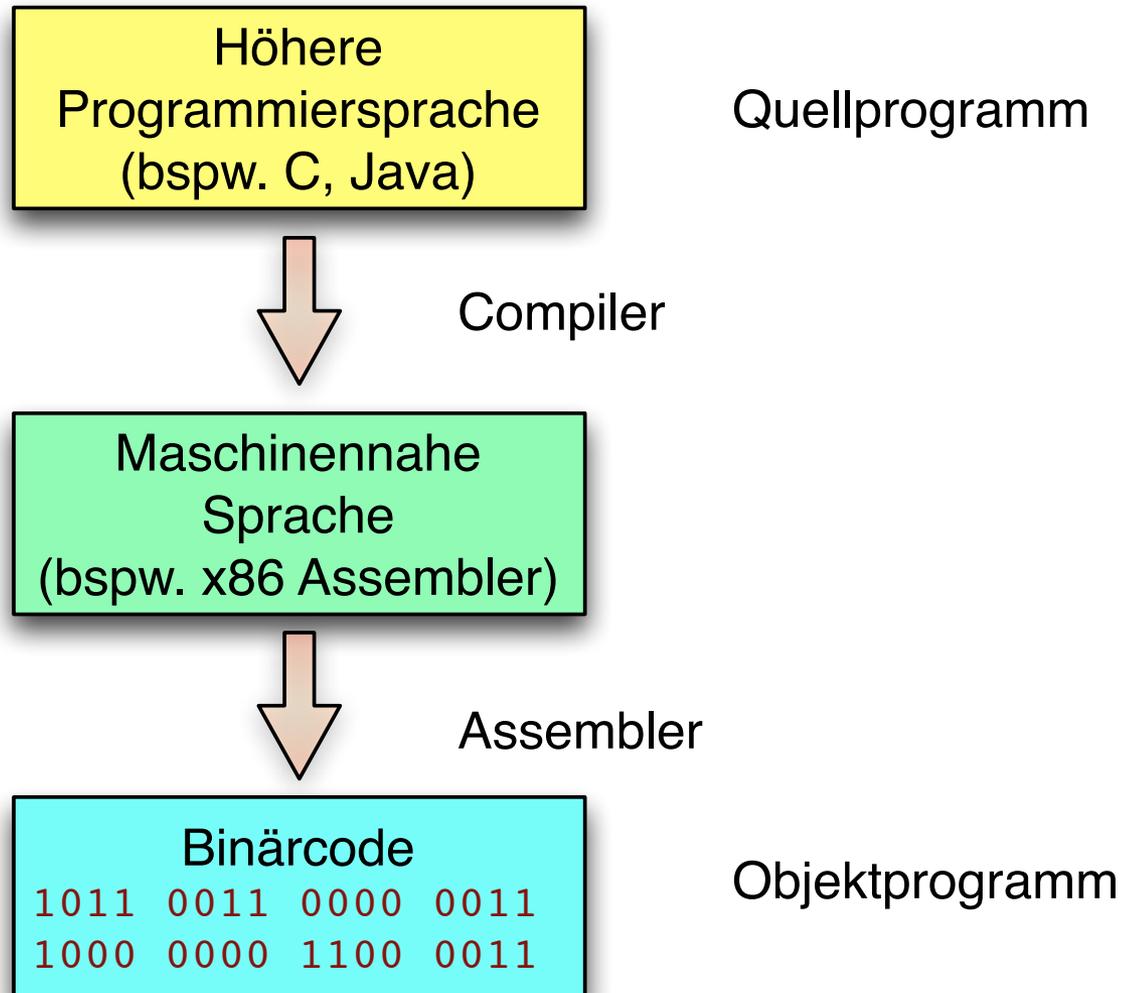
```
1011 0011 0000 0011
```

```
1000 0000 1100 0011 0000 0100
```

Höhere Programmiersprachen

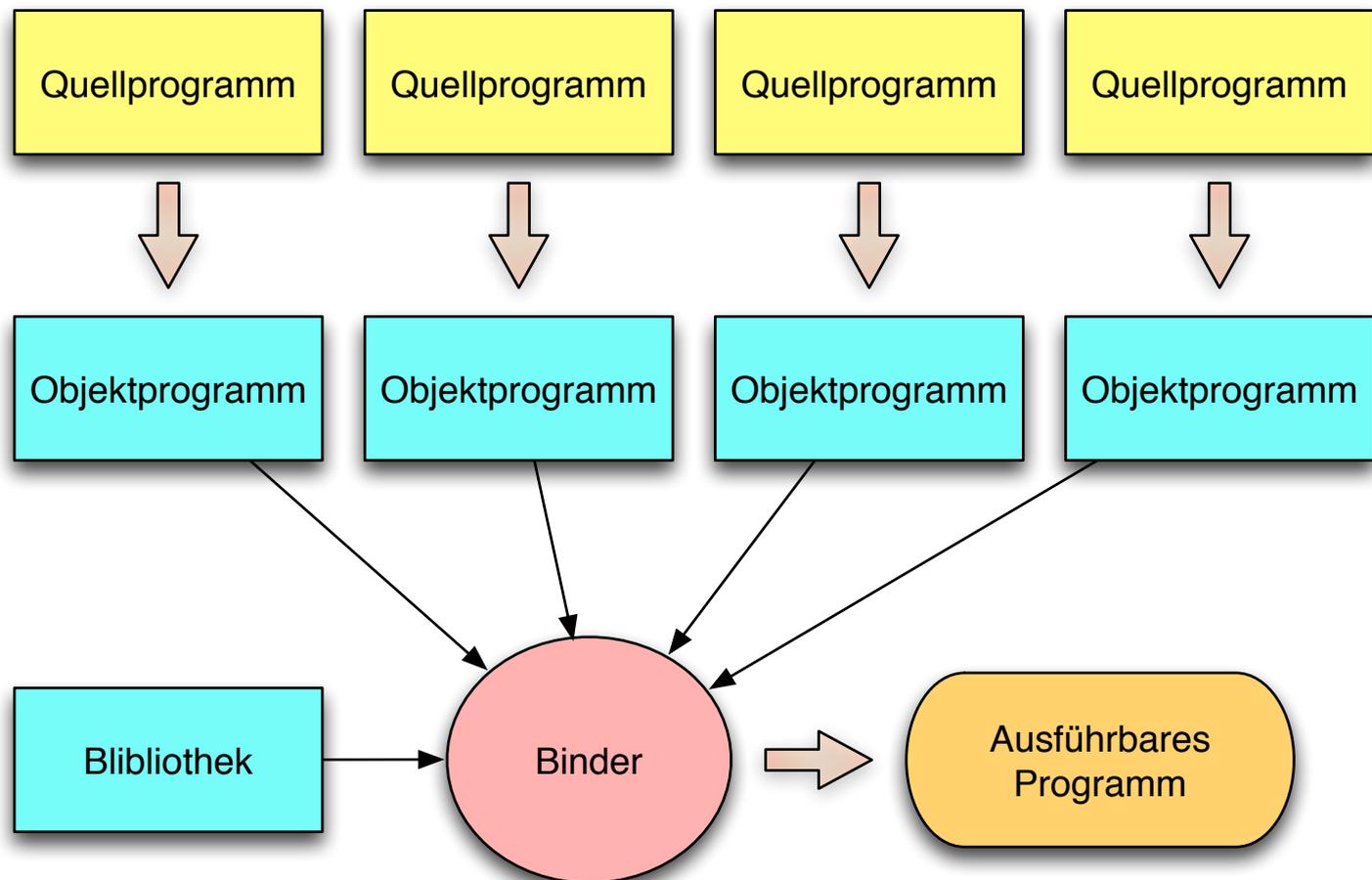
- Programmiersprachen, deren Befehle sich am Anwendungsbereich orientieren
- Abstrahieren vom Befehlssatz der Prozessoren
- Verwenden Elemente der natürlichen Sprache und Mathematik
- Für Ausführung: Quellprogramme werden **kompiliert** oder **interpretiert**

Übersetzung von Quellprogramm in höherer Programmiersprache

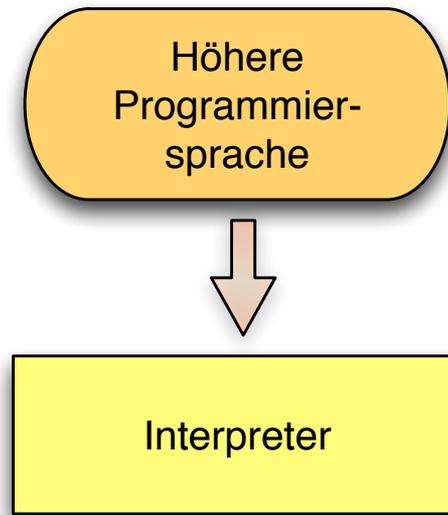


Ein **Compiler** ist ebenso ein Programm, das meist in einer höheren Programmiersprache implementiert ist.

Erzeugung von ausführbaren Programmen



Ausführung mittels Interpreter

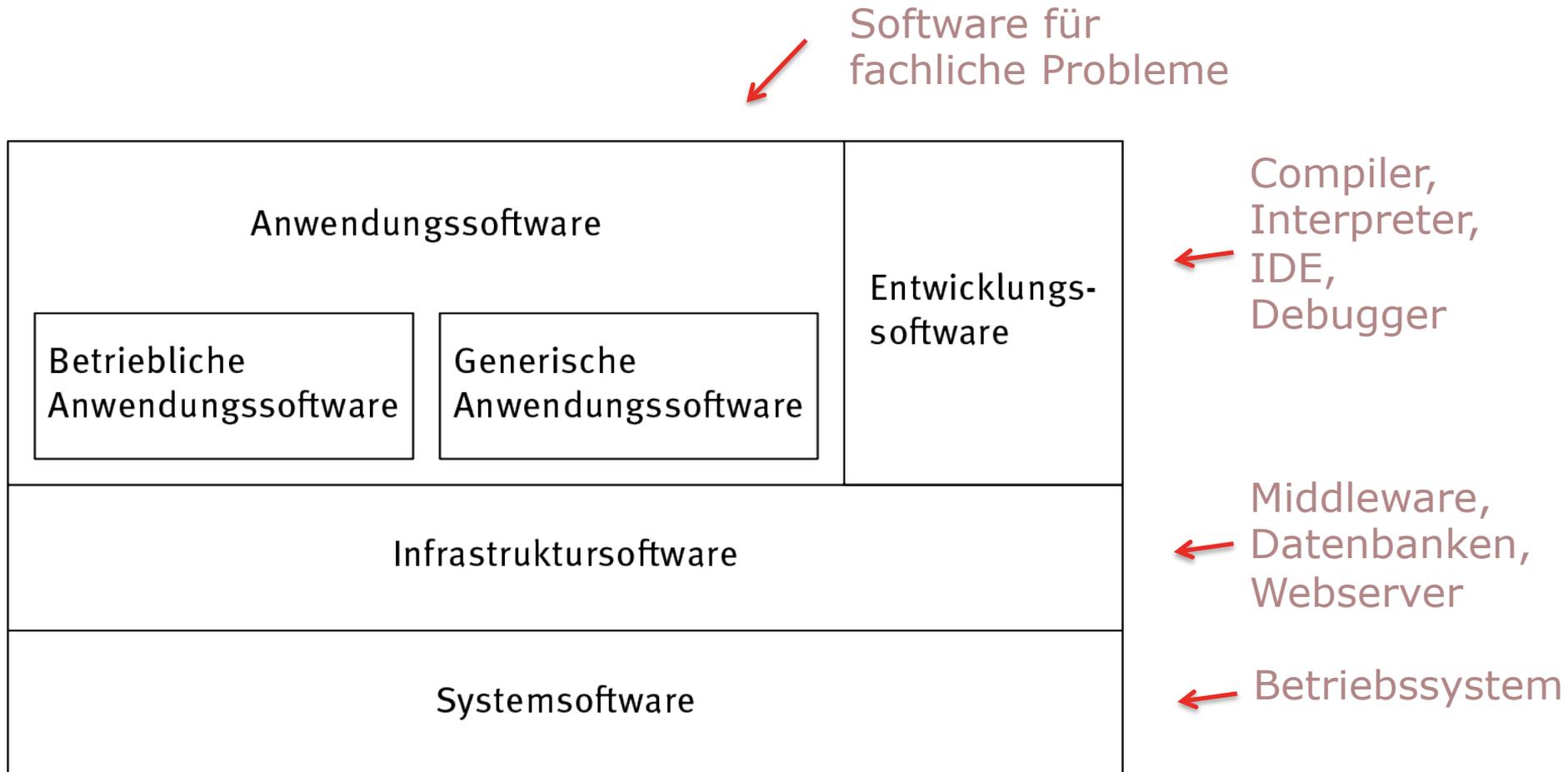


Ein **Interpreter** ist ebenso ein Programm, das meist in einer höheren Programmiersprache implementiert ist

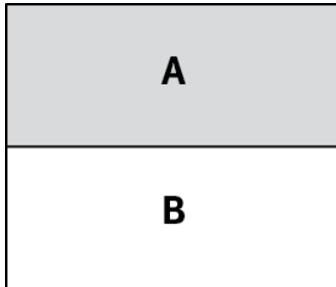
Arten von höherem Programmiersprachen

- **Imperative Programmiersprachen:** Formulierung von komplexen Anweisungen
- **Objektorientierte Programmiersprachen:** Beschreibung des Realweltprogramms durch Objekte und Klassen
- **Funktionale Programmiersprachen:** Definition von Funktionen (Ermittlung von Ergebnissen aus Eingabewerten)
- **Logische Programmiersprachen:** Beschreibung des Realweltprogramms durch (Prädikaten-)Logik

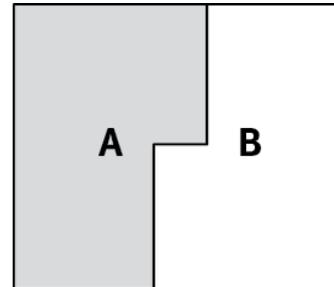
Arten von Software



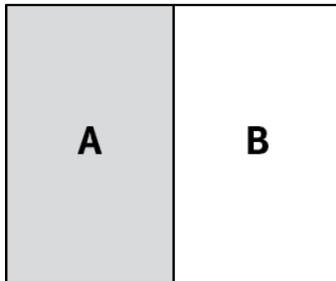
Abhängigkeiten von Softwarekomponenten



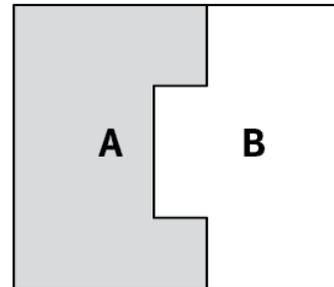
A benutzt B



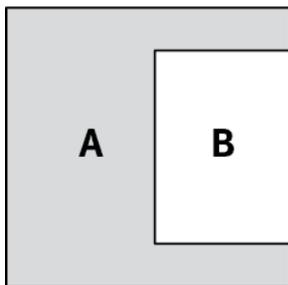
A koexistiert mit B
A benutzt B



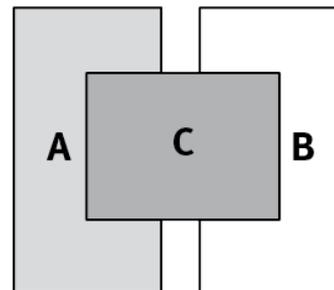
A koexistiert mit B



A kooperiert mit B
A benutzt B
B benutzt A



A bettet B ein
A benutzt B
B benutzt ausschließlich A

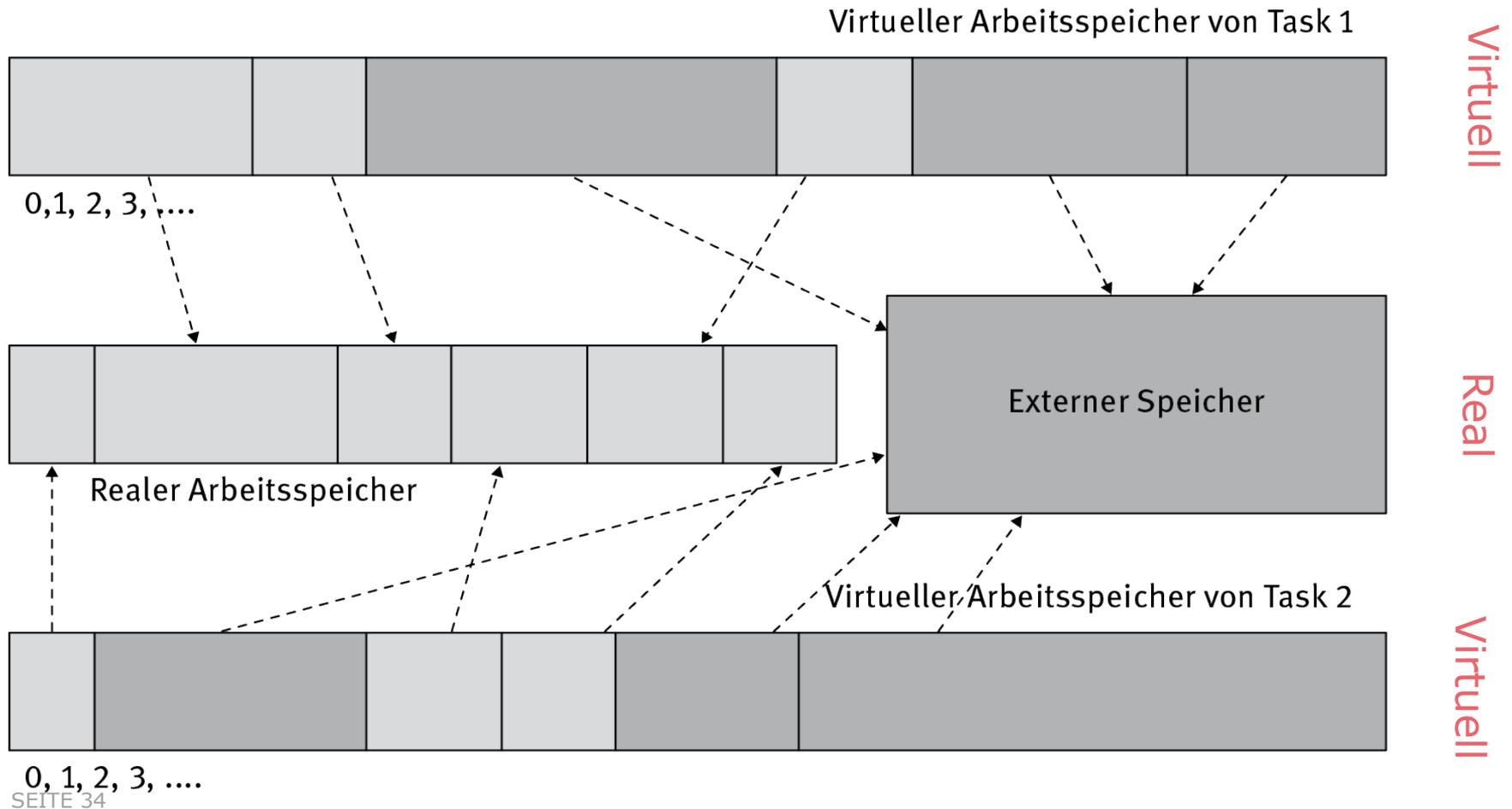


A kommuniziert mit B über C
A benutzt C, C benutzt A
B benutzt C, C benutzt B

Aufgaben des Betriebssystems

- **Auftragsverwaltung:**
Prozess vs. Thread, Ein- und Mehrprogrammbetrieb
- **Arbeitsspeicherverwaltung:**
Virtualisierung des Arbeitsspeichers, Speicherschutz
- **Dateiverwaltung:** Dateisysteme, SAN vs. NAS
- **Benutzerverwaltung:** Ein- und Mehrbenutzerbetrieb; Identitäts- und Rechteverwaltung

Arbeitsspeicherverwaltung



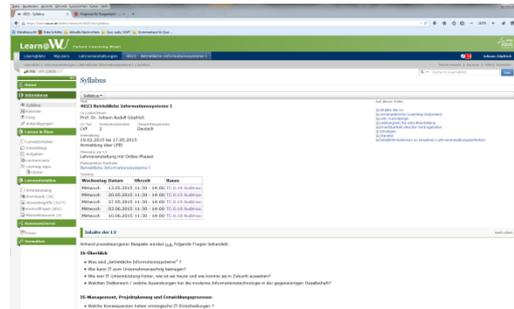
Diskussionsfragen Kapitel 10

- Was sind die Vorteile der Fließbandverarbeitung? Was wäre bei einem Prozessor ohne Fließbandverarbeitung schlechter?
- Warum gibt es mehrere Arten von höheren Programmiersprachen? Würde nicht eine einzige höhere Programmiersprache ausreichen und die Softwareentwicklung wesentlich vereinfachen?
- Welche Gründe sprechen für die Wiederverwendung von Softwarekomponenten?



Fallbezogene Aufgabe zu Kapitel 10

Learn@WU:



- Eignet sich das System für die Nutzung von Prozessoren mit mehreren Prozessorkernen?
- Welche Programmiersprache(n) würden sich für so ein System eignen?
- Warum sind einige Systemkomponenten in kompilierten, andere in interpretierten Programmiersprachen realisiert?
- Welche Rolle spielt Modularität und Wiederverwendung von Softwarekomponenten bei dem skizzierten System?

