



Kapitel 11: Rechnersysteme

- Aufbau und Funktionsweise von Rechnern
- Elektronische Bauelemente (Chips)
- Arten von Rechnern
- Aufbau und Funktionsweise von Software
- Bestandteile von Software
- Betriebssysteme
- Virtualisierung



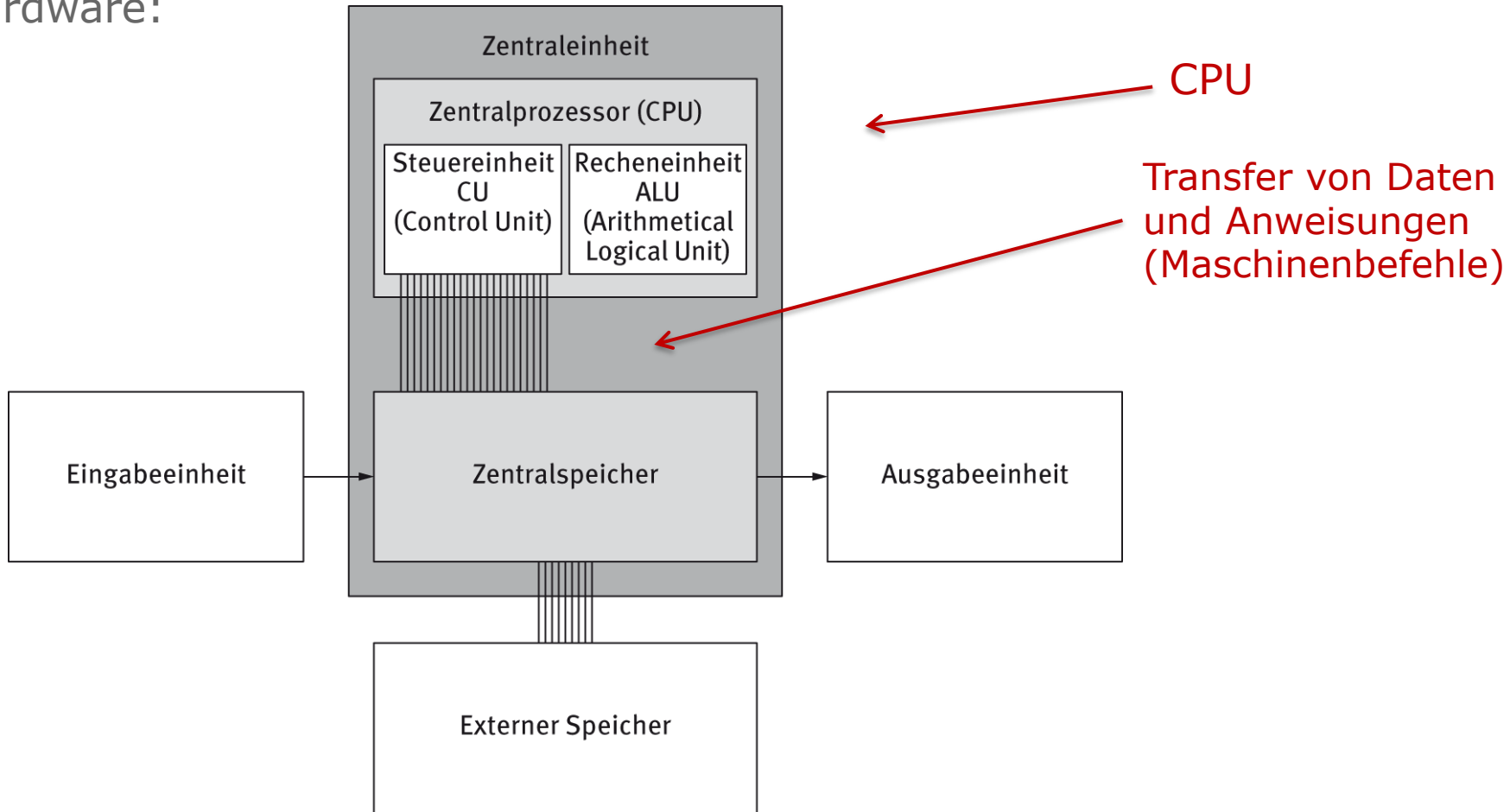
11.1 Aufbau und Funktionsweise von Rechnern

Rechner und Hardware

- Als Sammelbegriff für die Baueinheiten, die bei einem Rechnersystem genutzt werden, hat sich die englische Bezeichnung **Hardware** durchgesetzt.
- Die **Zentraleinheit** (engl.: central unit) ist eine Funktionseinheit innerhalb eines Rechners, die einen oder mehrere *Prozessoren* und *Zentralspeicher* umfasst.
 - Ein **Zentralprozessor** (engl.: central processing unit; Abkürzung: **CPU**) steuert entsprechend den jeweiligen Programmen den Gesamtablauf der Informationsverarbeitung, koordiniert die beteiligten Funktionseinheiten und führt Rechenoperationen aus.
 - Der **Zentralspeicher** (engl.: main memory) enthält die aktuell laufenden Programme und die aktuell zu verarbeitenden Daten.

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
```

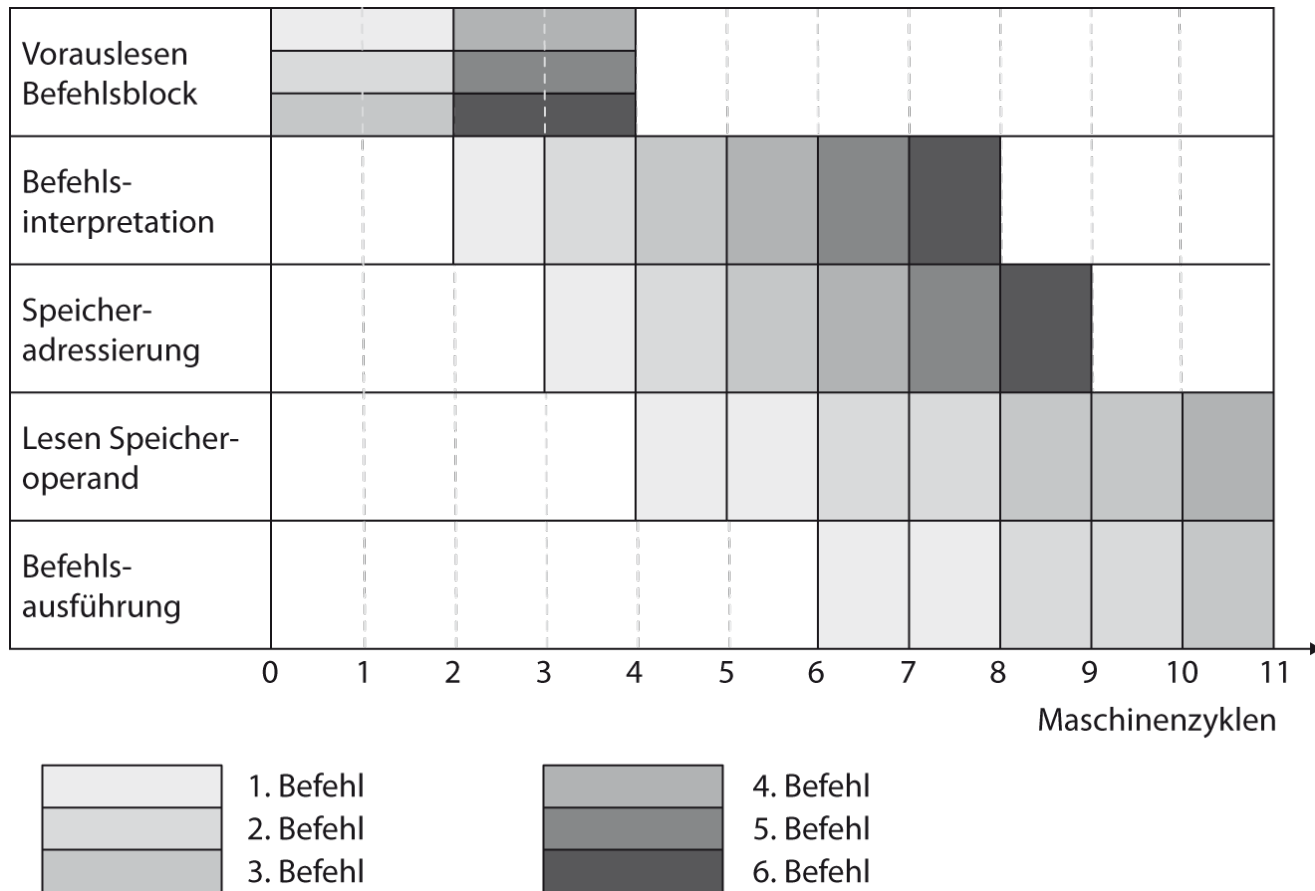
- Die gleichen **Maschinenbefehle in Binärcode:**

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

Ausführung von Maschinenbefehlen im Prozessor

- 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

Fließbandverarbeitung innerhalb eines Prozessors



Speicher

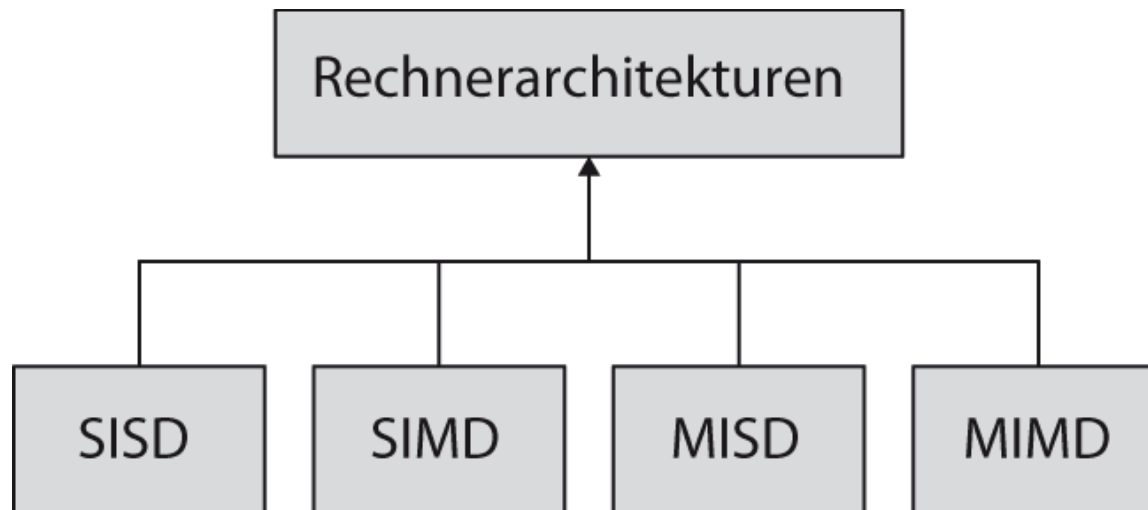
- Ein **Speicher** (engl.: storage; memory) ist eine Funktionseinheit eines Rechners, die Information aufnimmt, aufbewahrt und abgibt.
- Der **Arbeitsspeicher** (Synonym: Hauptspeicher; engl.: main memory; working storage) ist ein vom Prozessor direkt adressierbarer Zentralspeicher, der über eine Vielzahl von Speicherzellen verfügt, in denen die aktuell laufenden Programme und deren Daten gespeichert werden. Die Speicherzellen werden über *Speicheradressen* referenziert.
- Speicher, die nicht Bestandteil der Zentraleinheit sind, werden als **externer Speicher** (engl.: external storage) bezeichnet. Externe Speicher sind in der Regel nicht flüchtig.
- Ein **Puffer** (engl.: buffer) ist ein Speicher, der vorübergehend Daten aufnimmt, die von einer Funktionseinheit zu einer anderen übertragen werden.

Externe Funktionseinheiten

- Eine Funktionseinheit innerhalb eines Rechners, die nicht zur Zentraleinheit gehört, wird **periphere Einheit** (engl.: peripheral unit) genannt.
- Dementsprechend werden externe Speicher auch als *periphere Speicher* (engl.: peripheral storage) bezeichnet. Neben den externen Speichern zählen auch die *Ein- und Ausgabeeinheiten* zu peripheren Einheiten.
- Ein **Bus** (engl.: bus) ist ein Verbindungssystem, das von allen an den Bus angeschlossenen Einheiten gemeinsam genutzt wird.

Rechnerarchitekturen

- Eine **Rechnerarchitektur** (engl.: computer architecture) beschreibt auf hohem Abstraktionsniveau, aus welchen funktionalen Einheiten ein Prozessor/Rechner aufgebaut ist, in welcher Anzahl diese Komponenten vorhanden sind und wie diese verbunden sind.



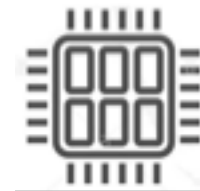
Rechnerarchitekturen

- Eine **Single-Instruction-Single-Data-Architektur** (Abkürzung: SISD-Architektur) beschreibt einen Rechner, bei dem in einem Bearbeitungsschritt jeweils nur eine Anweisung (engl.: single instruction) mit einem einfachen Operanden (engl.: single data) ausgeführt werden kann.
- Eine **Single-Instruction-Multiple-Data-Architektur** (Abkürzung: SIMD-Architektur) beschreibt einen Rechner, bei dem ein einzelner Befehl auf eine Vielzahl von Operanden angewendet werden kann. Man spricht hierbei auch von einem **Vektorrechner** (engl.: array processor).
- Eine **Multiple-Instruction-Multiple-Data-Architektur** (Abkürzung: MIMD-Architektur) beschreibt ein Rechnersystem, bei dem gleichzeitig mehrere Anweisungen (mehrere Befehlsströme) mit unterschiedlichen Daten (mehrere Datenströme) ausgeführt werden können.
- Eine **Multiple-Instruction-Single-Data-Architektur** (Abkürzung: MISD-Architektur) beschreibt ein Rechnersystem, bei dem mehrere Operationen gleichzeitig mit denselben Daten ausgeführt werden können.

Mehrkernprozessoren und Mehrprozessorsysteme

13

- **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.



Beispiel:
6-Kern Prozessor

- **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.

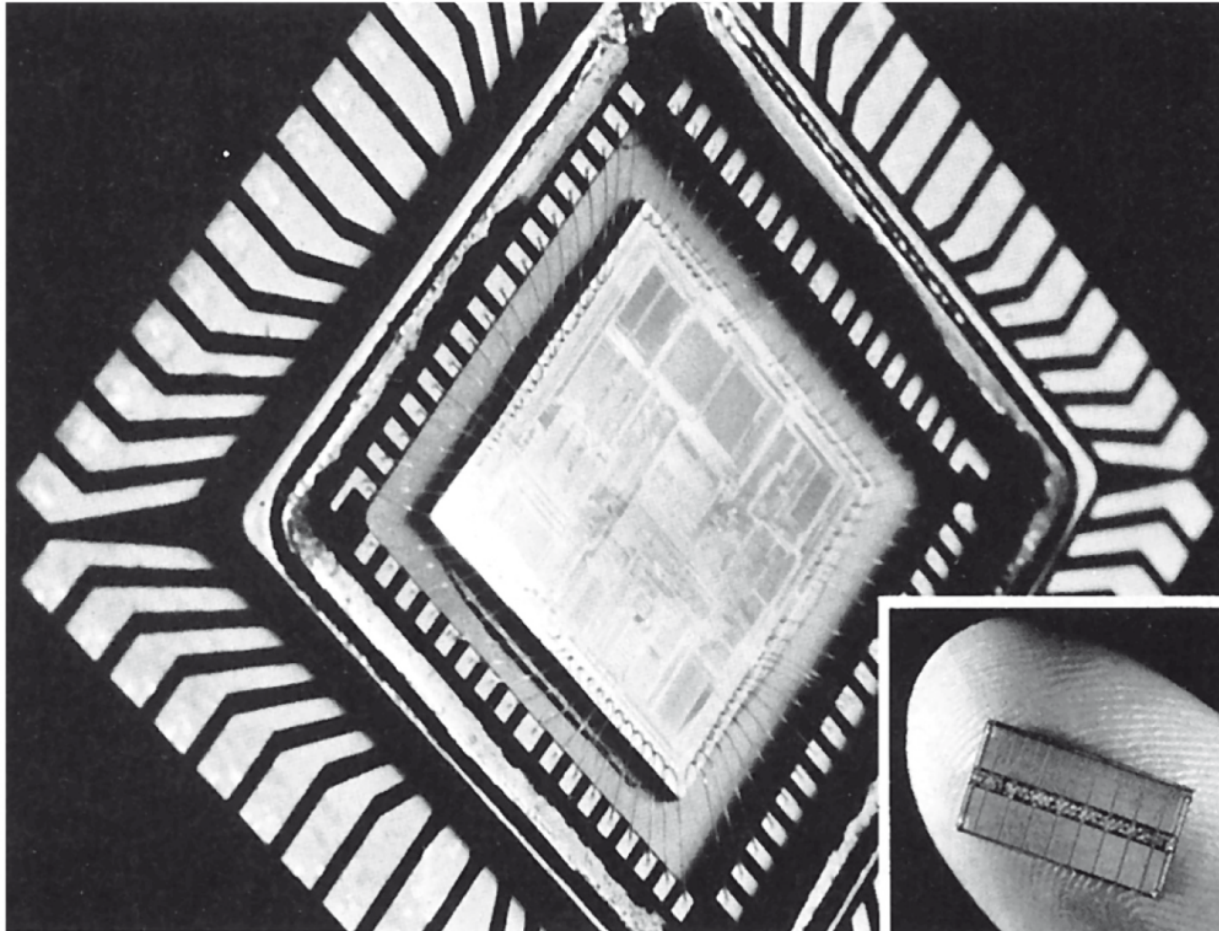


11.2 Elektronische Bauelemente (Chips)

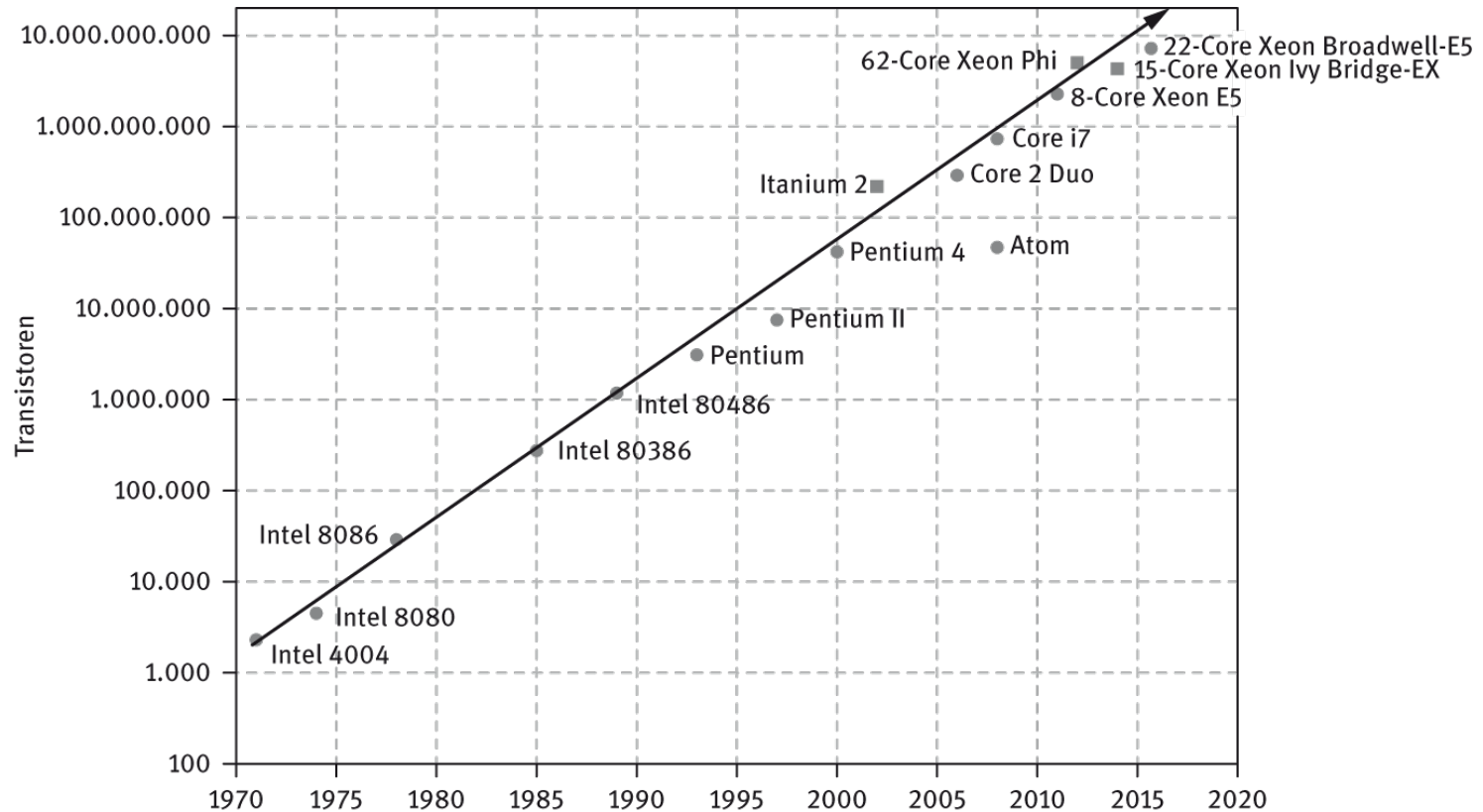
Chips

- Ein **Chip** (engl.: chip; Synonym: Mikrochip) ist ein elektronisches Bauelement aus halbleitendem Material (meist Silizium), das einen integrierten Schaltkreis (engl.: integrated circuit, Abkürzung: IC) für Berechnungs- und/oder Speicherfunktionen enthält. Die Bauelemente der Schaltungen sind zum größten Teil Transistoren, aber auch Widerstände, Kondensatoren und andere Halbleiterbauelemente.

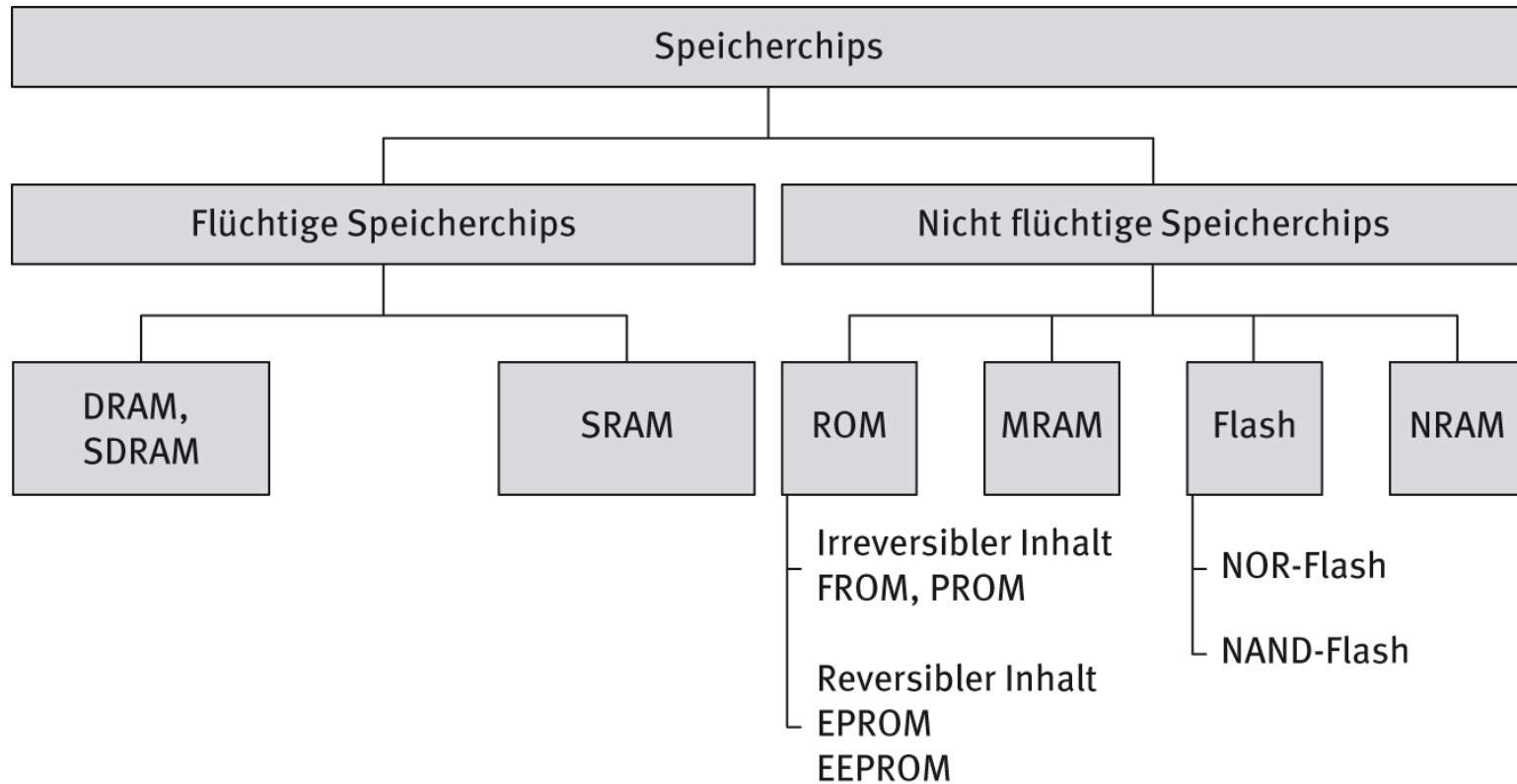
Makrofotografie eines Chips und Chip in Originalgröße (unten rechts)



Anzahl von Transistoren auf ausgewählten Intel-Prozessorchips



Speicherchips



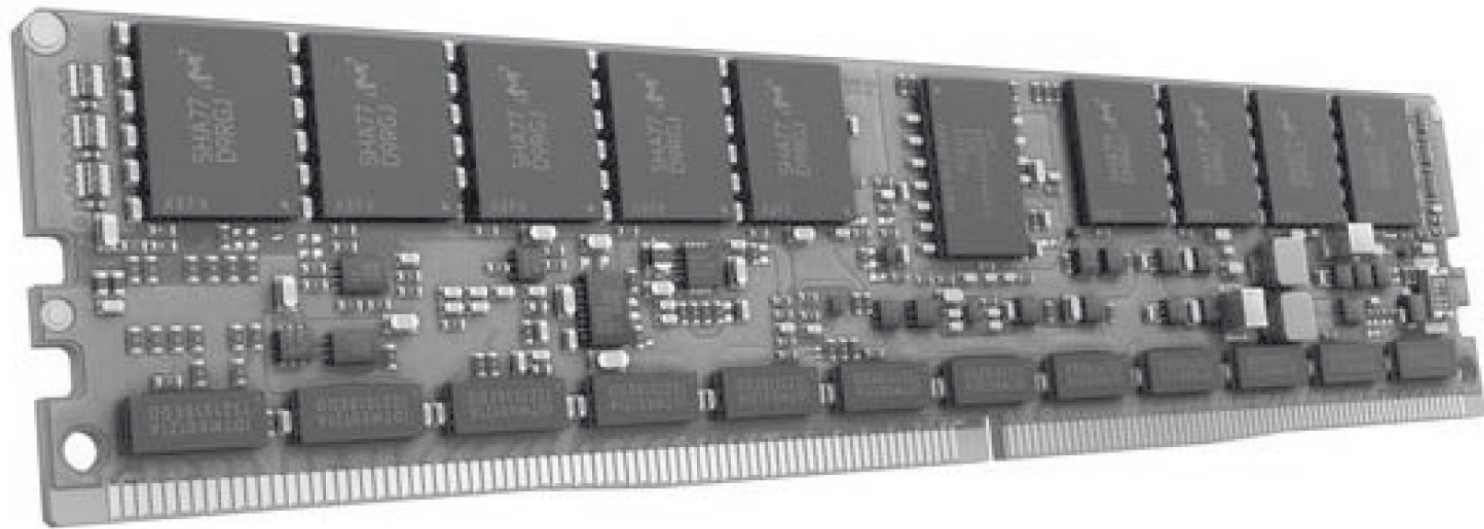
Speicherchips

- Ein **Schreib-Lese-Speicher** (engl.: read-write memory) ist ein Speicher, der gelesen und beschrieben werden kann. Im Gegensatz dazu ist ein **Festwertspeicher** (engl.: read only memory, Abkürzung: **ROM**) ein Speicher, der während des normalen Betriebs nur gelesen werden kann. Die gespeicherte Information ist entweder unveränderbar oder bedarf einer speziellen Operation zur Aktualisierung.
- Ein **Direktzugriffsspeicher** (engl.: random-access memory, Abkürzung: **RAM**) ist ein Speicher, bei dem jede einzelne Speicherstelle über ihre fest zugeordnete Adresse gelesen oder beschrieben (und damit auch gelöscht) werden kann. Die Zugriffszeit ist für alle Speicherstellen in etwa gleich lang.
- Ein **flüchtiger Speicher** (engl.: volatile memory) ist ein Schreib-Lese-Speicher, der bei Ausfall der Versorgungsspannung die gespeicherte Information verliert.
- Ein **nicht flüchtiger Speicher** (engl.: non-volatile memory) ist ein Speicher, dessen Inhalt auch ohne Versorgungsspannung erhalten bleibt.

Speicherchips

- Flüchtige Speicherchips
 - DRAM
 - SRAM
- Nicht flüchtige Speicherchips
 - ROM
 - Flash
 - MRAM
 - NRAM
- Ein **Flash-Speicher** ist ein spezieller EEPROM-Typ, dessen Inhalt durch Anlegen eines elektrischen Felds in bestimmten Speicherbereichen, sogenannten Sektoren oder Blöcken (zum Beispiel 512 Bytes), auf einmal gelöscht wird. Die Speicherzellen können nicht – wie bei gewöhnlichen EEPROM-Chips – einzeln gelöscht werden. Das Schreiben erfolgt je nach Chiparchitektur selektiv für jede einzelne Zelle oder auf einmal für einen gesamten Block. Flash-Speicher haben eine begrenzte Lebensdauer, die sich durch die maximale Anzahl von Löschzyklen ergibt.

SDRAM-Speichermodul (Quelle: Samsung)



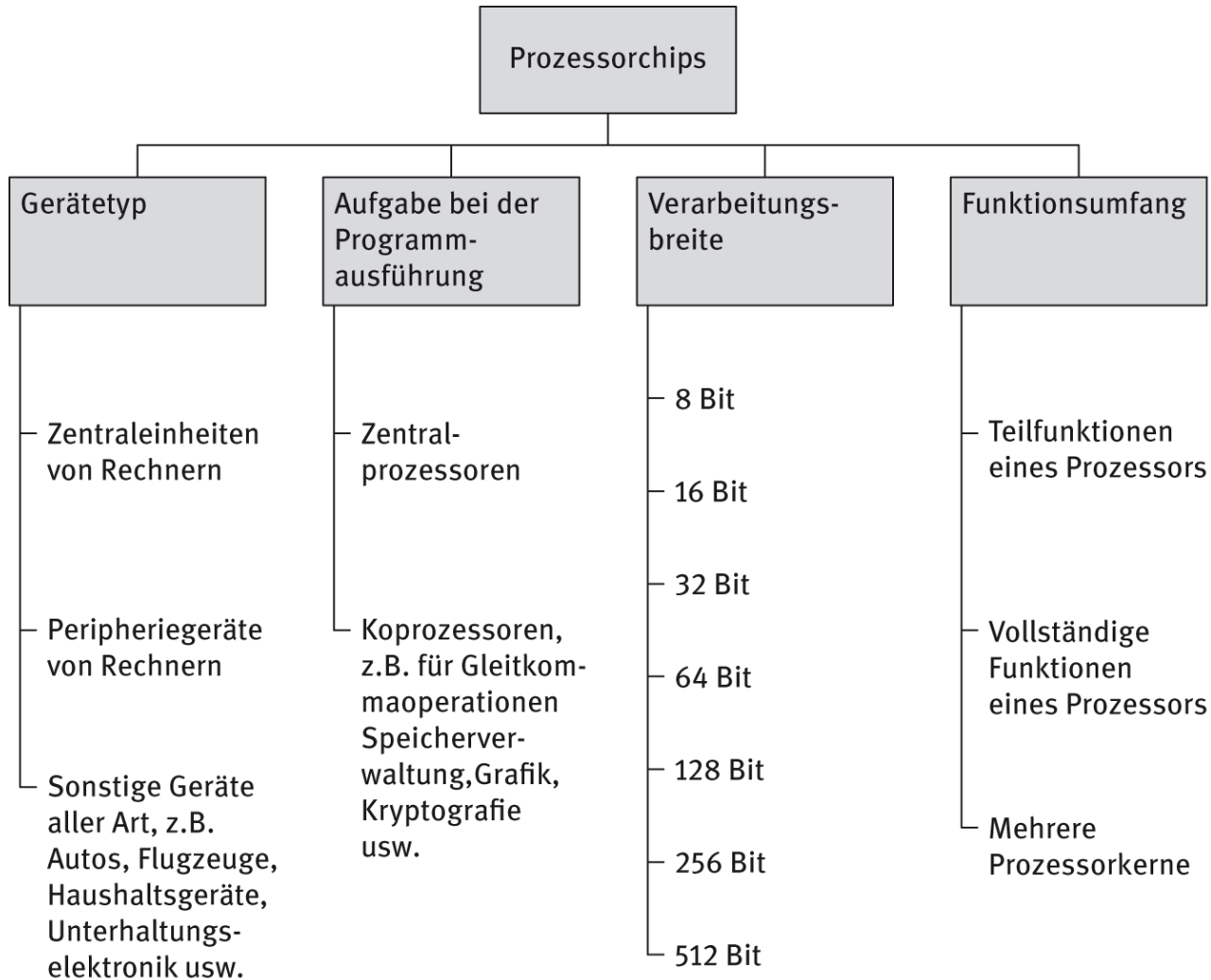
SSD-Platte mit 4 TB Kapazität (Quelle: Samsung)



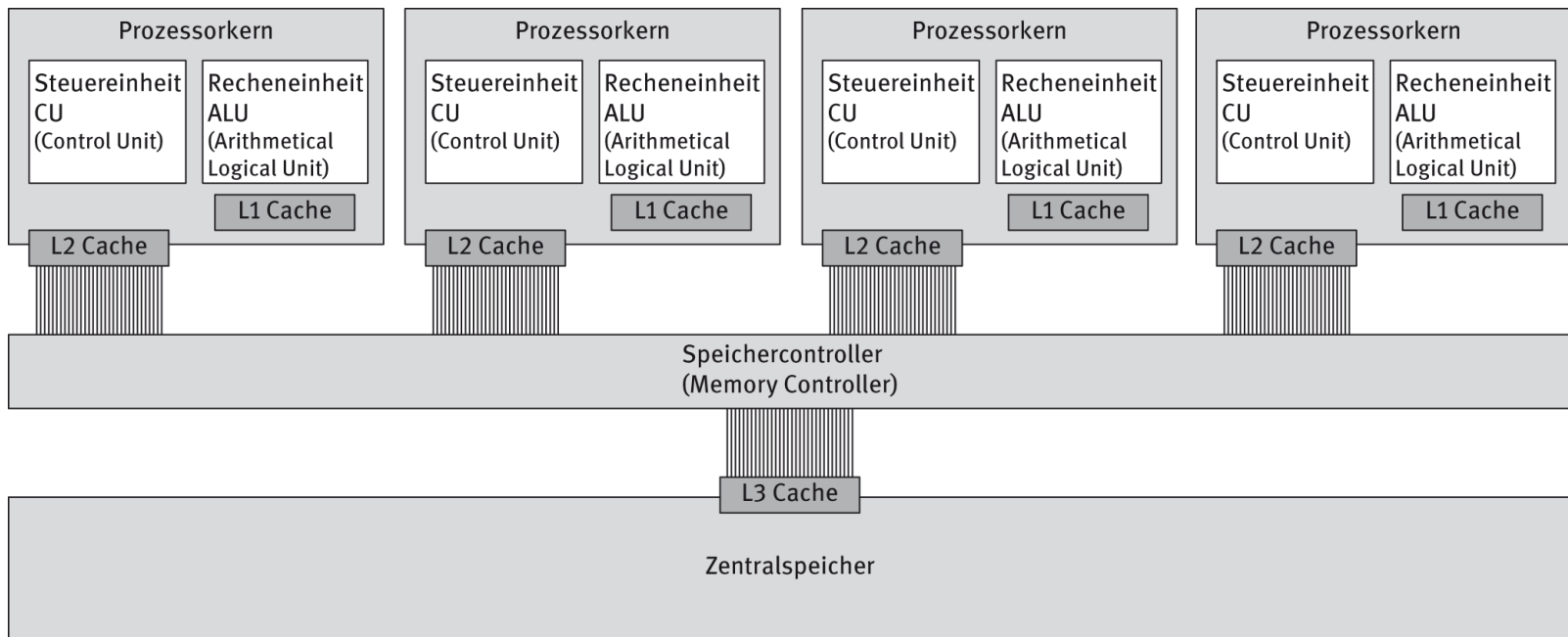
Prozessorchips

- Ein vollständiger Prozessor, der auf einem Chip untergebracht ist, heißt **Mikroprozessor** (engl.: microprocessor).
- **Mikroprozessoren** (engl.: microprocessor) werden gewöhnlich anhand der internen Verarbeitungsbreite in **8-Bit-, 16-Bit-, 32-Bit-, 64-Bit-, 128-Bit-, 256-Bit- und 512-Bit-Prozessoren** eingeteilt.
- Bei einem **Ein-Chip-Computer** (engl.: system on a chip; abgekürzt: SoC) sind alle Funktionseinheiten eines Rechners auf einem einzigen Chip integriert. Dieser Chip enthält neben der Zentraleinheit (Zentralprozessor und Zentralspeicher) auch Ein-/Ausgabekomponenten, Taktgeber, Schnittstellen zu externen Geräten (beispielsweise USB, Ethernet, FireWire), Komponenten für die Umwandlung von digitalen Signalen in analoge und umgekehrt, wie auch drahtlose Sende- und Empfangseinrichtungen.
- Ein-Chip-Computer dienen als sogenannte **Mikrocontroller** (engl.: microcontroller unit; Abkürzung: **MCU**) zur Überwachung und Steuerung von Prozessen oder Umweltbedingungen. Zusätzlich werden sie als **eingebettete Systeme** (engl.: embedded systems) in Haushaltsgeräten, Geräten der Unterhaltungselektronik, in Maschinen und Fahrzeugen eingesetzt.

Klassifikation von Prozessorchips



Cache-Hierarchie (Vierkernprozessor)



Raspberry Pi 3, Model B+

(Quelle: Raspberry Pi Foundation)





11.3 Arten von Rechnern

Unterscheidung nach Leistungsklassen

Früher ging man von den folgenden *Rechnerkategorien* aus:

- Großrechner (engl.: mainframe computer),
- Minicomputer (engl.: mini computer),
- Workstation (engl.: workstation),
- Personalcomputer, abgekürzt PC (engl.: personal computer)
- **Persönliche Informationshilfsmittel** (engl.: information appliance; Abkürzung: IA) sind mobile Endgeräte für Individuen, die als universelles Instrument verwendet werden, um sowohl geschäftliche als auch privaten Aufgaben zu erledigen.

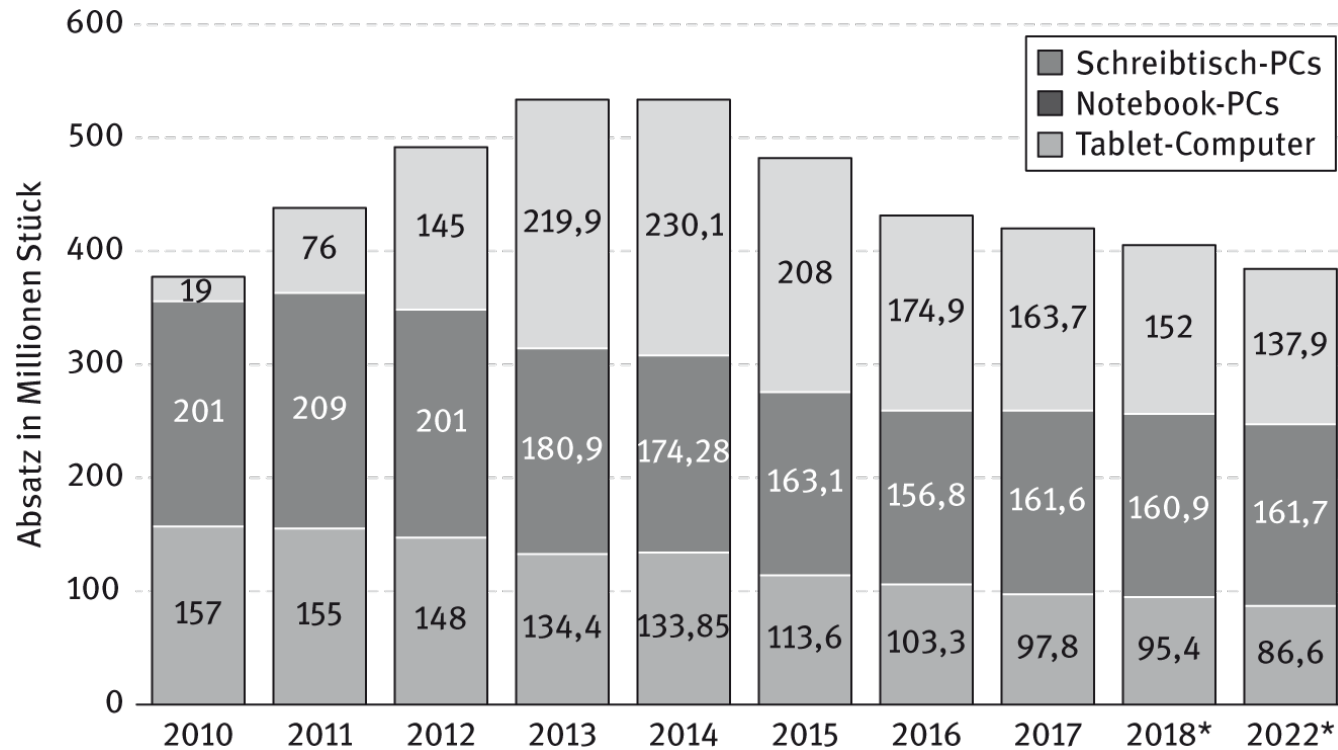
Rechnerkategorien nach Computerzeitalter

Computerära	Großrechner	Mini-computer	Arbeitsplatz-rechner	Persönliche Informations-hilfsmittel	Computer-gestützte Geräte
Blütezeit	1955–1985	1965–1985	1980–2010	2000 +	2010 +
Installationsort	Klimatisiertes Rechenzentrum	Büro	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	50–300
Zahl der Nutzer	100+	10 +	1 +	1	1 +
Nutzungszeit in Jahren	5–7	4 +	2 +	1 +	1 +
Verkaufte Einheiten pro Jahr in Millionen bzw. Marktpotenzial	0,010	0,100	300	1.000	100.000
Treibende Kräfte bzw. Schlüssel-nwendungen	Unternehmensweite Massen- und Routinedatenverarbeitung, Time-Sharing	Einsatz in technischen und kaufmännischen Bereichen	Rechenleistung am Arbeitsplatz und in privaten Haushalten	Inanspruchnahme von Internet-Diensten jederzeit und überall	Internet der Dinge, intelligente Fertigungs- und Vertriebssysteme, Sensornetze

Ein-Chip-Computer mit Bluetooth und aufladbarer Batterie (Quelle: Intel)



Weltweite Absatzzahlen von Schreibtisch-PCs, Notebook-PCs und Tablet-Computern, * für 2018 und 2022 geschätzte Werte (Quelle: www.statista.com)



Serverschrank des Learn@WU-Servers



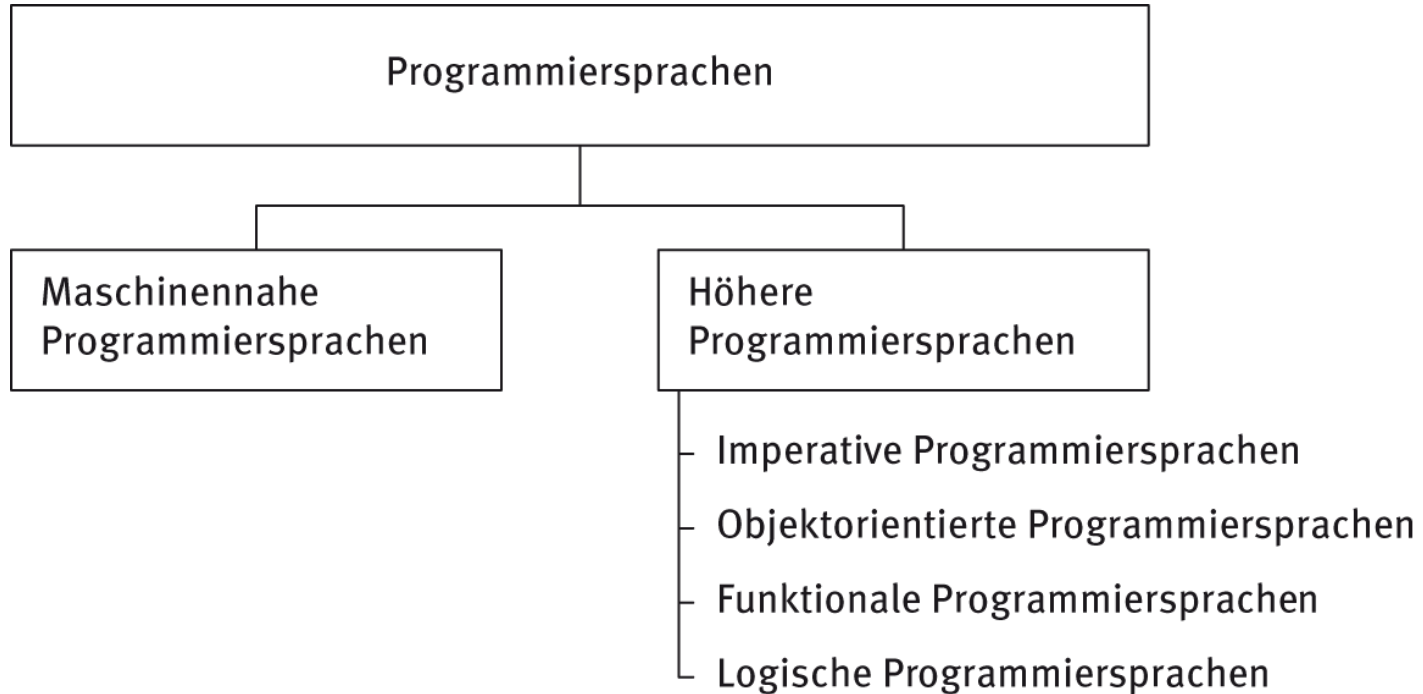


11.4 Aufbau und Funktionsweise von Software

Software

- **Software** (engl.: software) ist ein Sammelbegriff für Computerprogramme.
- Eine **Programmiersprache** (engl.: programming language) ist eine Sprache zur Formulierung von Rechenvorschriften (Algorithmen), die von einem Computer ausgeführt werden können. Mittels Programmiersprachen werden komplexe Aufgabenstellungen durch eine detaillierte Folge von Rechenanweisungen und Datenbeschreibungen in einer automatisierbaren Weise definiert.

Programmiersprachen



Maschinennahe Programmiersprachen

- Eine Sprache, die den Aufbau der Befehle der Maschinsprache beibehält und die Befehle und Speicheradressen durch symbolische Ausdrücke (Namen) beschreibt, wird **maschinennahe Programmiersprache** (Synonym: Assemblersprache; engl.: assembler language) genannt.
- Ein **Assembler** (auch: Assemblierer, engl.: assembler) ist ein Übersetzungsprogramm, das ein in einer maschinennahen Sprache abgefasstes Programm in ein auf dem Prozessor direkt ausführbares Programm umwandelt (assembliert).

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

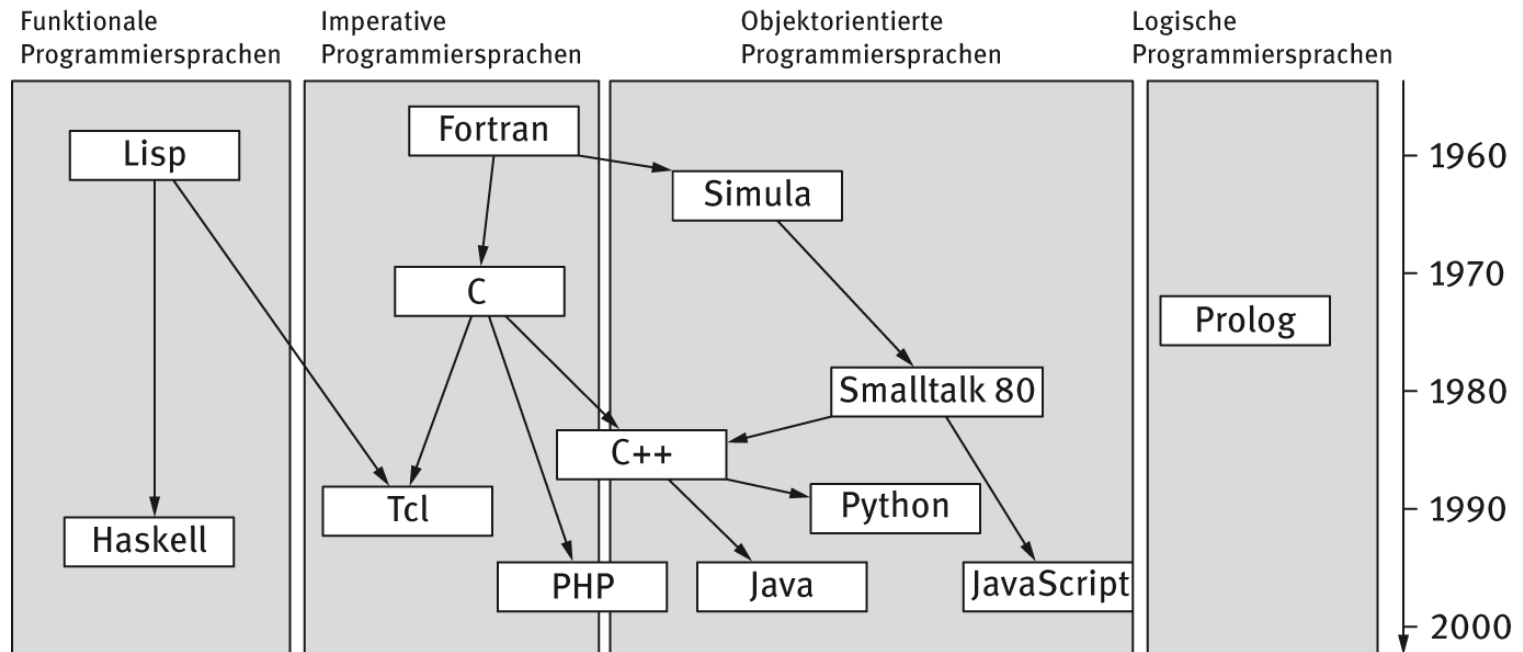
- Eine Sprache, deren Befehle sich am Anwendungsbereich orientieren, wird als **höhere Programmiersprache** (engl.: higher level programming language) bezeichnet. Höhere Programmiersprachen abstrahieren vom Befehlssatz des Prozessors und verwenden Elemente der natürlichen Sprache und/oder der Mathematik für das Formulieren von Programmen.
- Ein **Compiler** (auch: Kompilierer, engl.: compiler) ist ein Übersetzungsprogramm, das ein in einer höheren Programmiersprache abgefasstes Quellprogramm in eine Maschinensprache übersetzt (kompiliert). Die Übersetzung erfolgt häufig über den Zwischenschritt der Übersetzung in eine maschinennahe Sprache, wobei dann ein Assembler zur Erzeugung des Maschinencodes herangezogen wird.
- Ein **Binder** (engl.: linker) ist ein Dienstprogramm, das mehrere kompilierte Teilprogramme zu einem gemeinsamen Binärprogramm zusammenfasst und dabei Verweise (wie beispielsweise Funktionsnamen) auf externe Softwarekomponenten durch Sprungadressen ersetzt. Gleichzeitig werden auch Speicheradressen von Variablen ermittelt und im Programm eingesetzt.
- Ein **Lader** (engl.: loader) ist ein Dienstprogramm, das ein Binärprogramm von einem externen Speichermedium liest und in den Arbeitsspeicher kopiert, Kommandozeilenargumente übergibt und Register initialisiert, und schließlich das Programm startet (an die Startadresse des Programms verzweigt).
- Ein **Interpreter** (engl.: interpreter) ist ein Programm, das ein in einer höheren Programmiersprache abgefasstes Quellprogramm bei der Ausführung laufend interpretiert, das heißt, die Anweisungen in dem Programm laufend überprüft, analysiert und entsprechende Operationen vornimmt. Meist wird dabei aus Effizienzgründen beim Laden das Quellprogramm in einen *Zwischencode* (engl.: intermediate code) übersetzt, der interpretiert wird, wodurch wiederholte Überprüfungen von Anweisungen reduziert werden können.

Arten von höherem Programmiersprachen

- **Imperative Programmiersprachen:**
Formulierung von komplexen Anweisungen
- **Objektorientierte Programmiersprachen:**
Beschreibung des Realweltprogramms durch Objekte und Klassen des Anwendungsbereichs

Liefen die Möglichkeit,
 - Funktionen ohne großen Rechenaufwand in Abhängigkeit des Typs des betreffenden Objektes auszuwählen
 - Datenkapselung in Objekten
- **Funktionale Programmiersprachen:** Definition von Funktionen (Ermittlung von Ergebnissen aus Eingabewerten)
- **Logische Programmiersprachen:** Beschreibung des Realweltprogramms durch (Prädikaten-)Logik

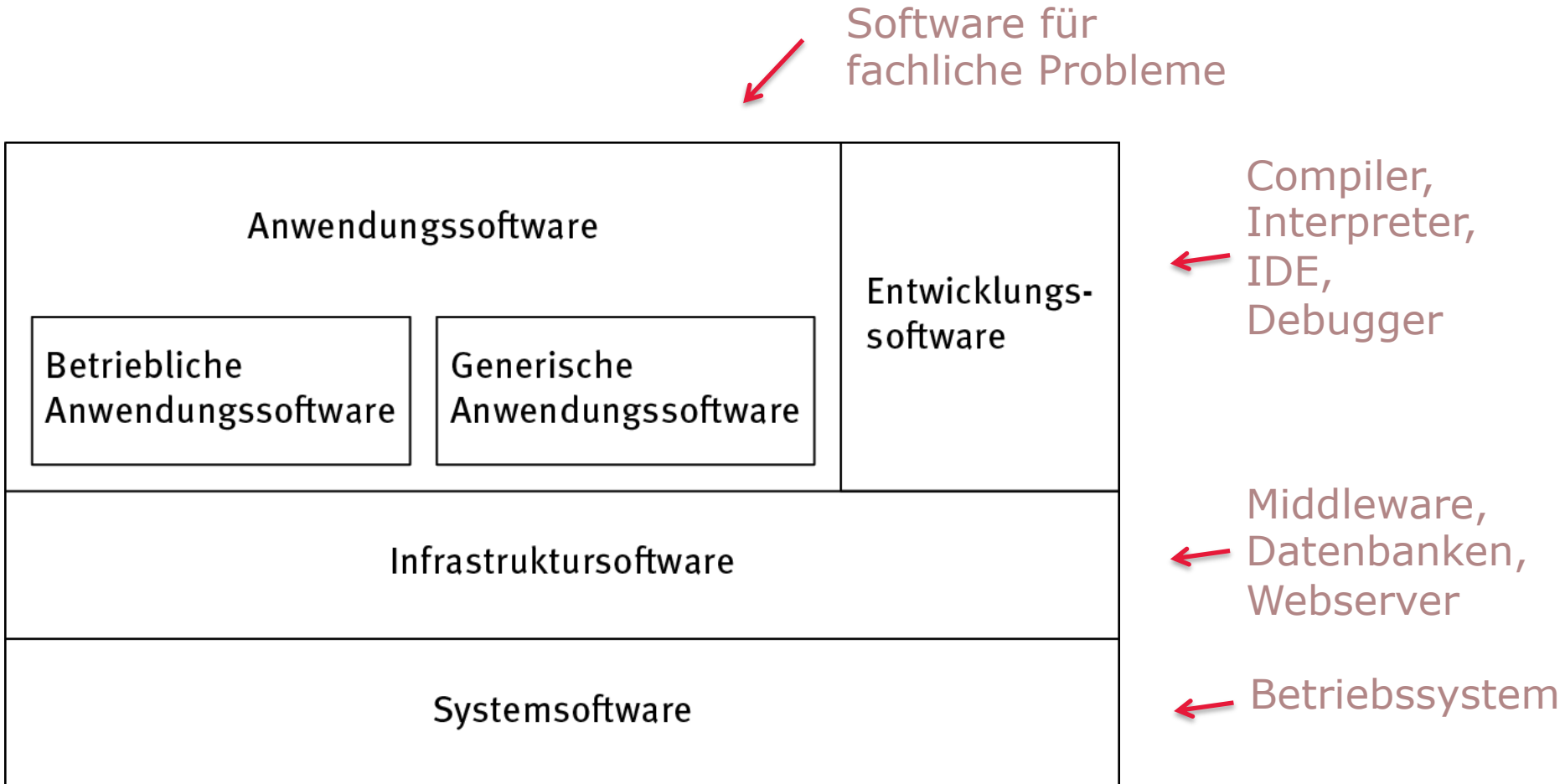
Arten von höheren Programmiersprachen





11.5 Bestandteile von Software

Arten von Software

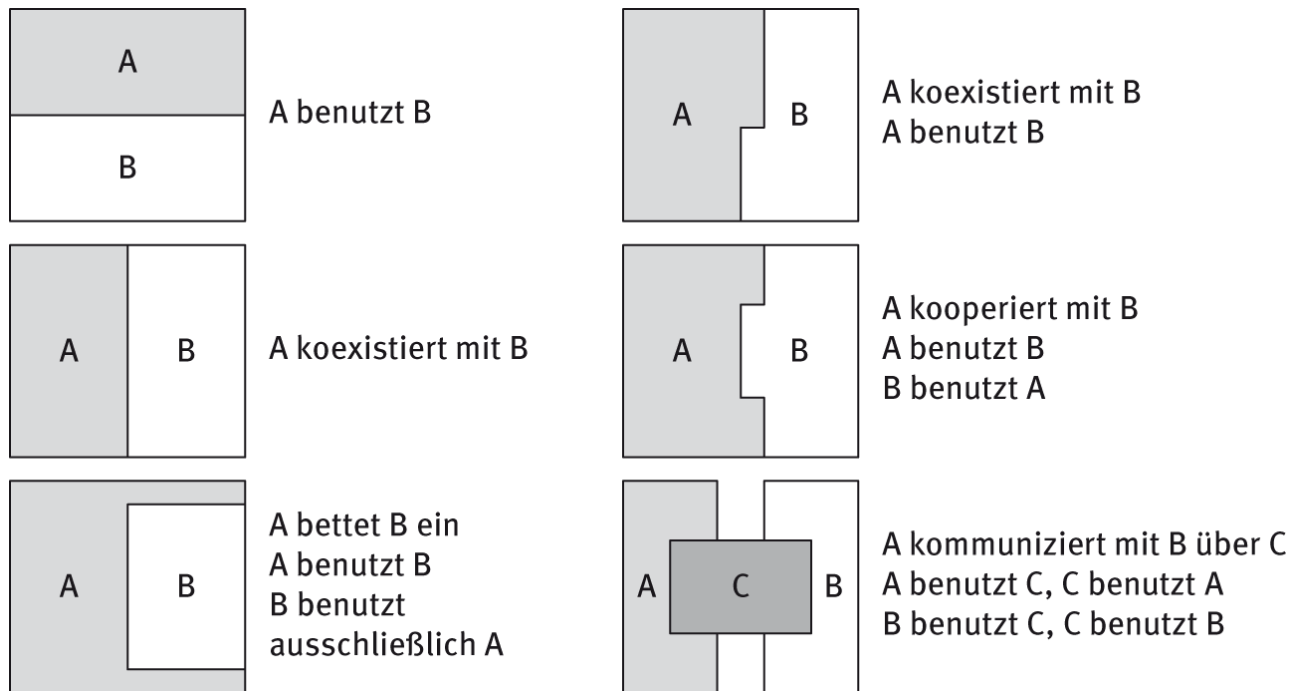


Kategorien von Softwarekomponenten

- Systemsoftware
- Infrastruktursoftware
- Anwendungssoftware
- Entwicklungssoftware

Abhängigkeiten von Softwarekomponenten

- Das Maß, in dem die verschiedenen Elemente innerhalb einer Softwarekomponente miteinander interagieren beziehungsweise miteinander verbunden sind, wird hierbei als **Kohäsion** (engl.: cohesion) der Komponente bezeichnet.
- Das Ausmaß, in dem eine Komponente mit anderen Komponenten interagiert, wird durch die **Kopplung** (engl.: coupling) gemessen.



Wiederverwendung von Softwarekomponenten

- Unter einem **komponentenorientierten System** (engl.: component based system) versteht man ein Softwaresystem, dessen Funktionalität auf klar abgrenzbare Komponenten verteilt wird, die jeweils eine bestimmte Teilfunktionalität zur Verfügung stellen. Unter einem **monolithischen System** (engl.: monolithic system) versteht man im Gegensatz dazu ein System, das als untrennbares Ganzes betrachtet wird.

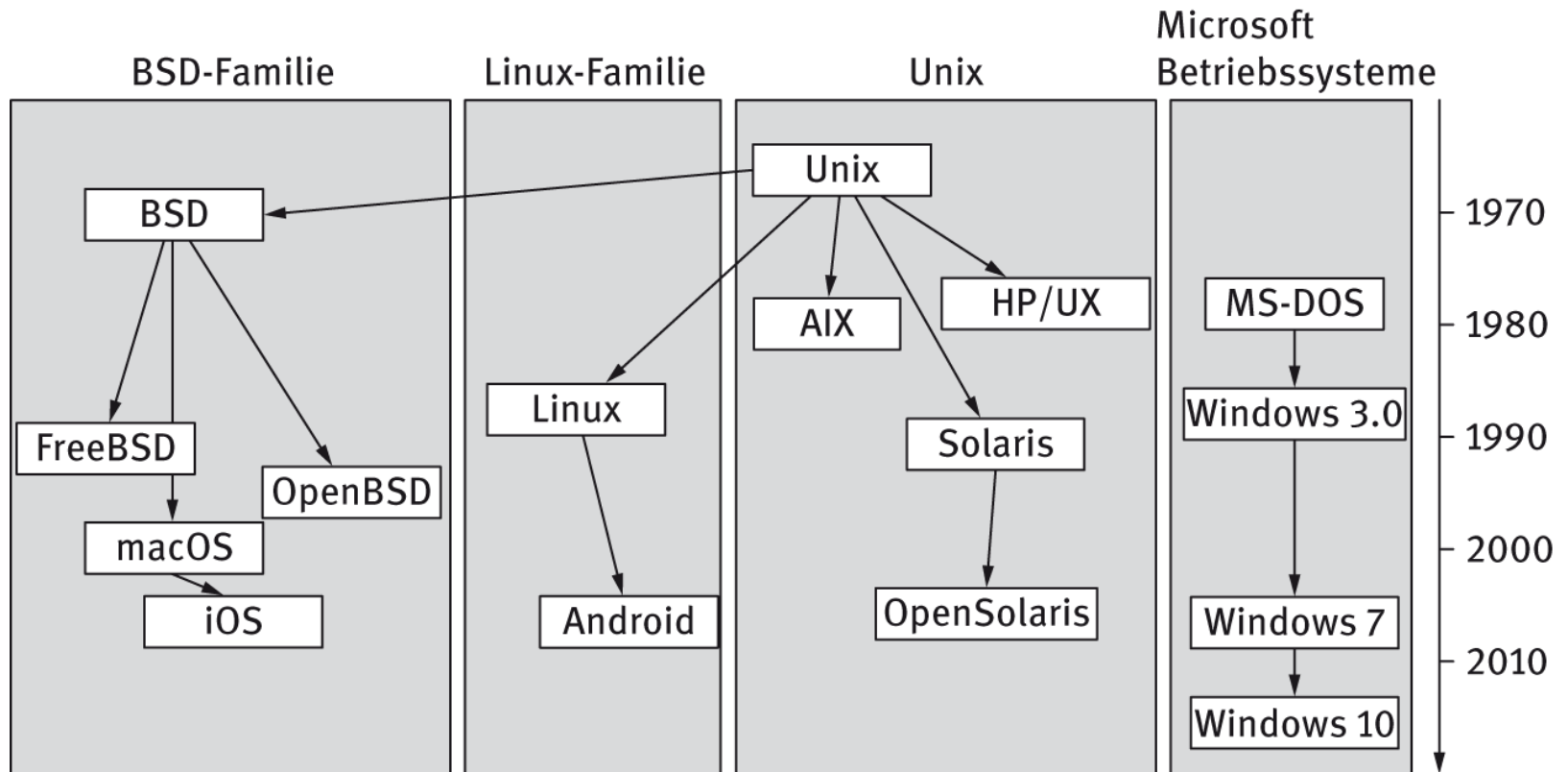


11.6 Betriebssystem

Betriebssystem

- Das **Betriebssystem** (engl.: operating system) umfasst die Gesamtheit der Systemprogramme eines Rechners, die (zusammen mit den Eigenschaften der Rechnerhardware) die grundlegende Infrastruktur für die Ausführung von Anwendungssoftware bilden. Das Betriebssystem bildet eine Abstraktion von Hardwareeigenschaften und ist für die Steuerung und Überwachung von Anwendungsprogrammen zuständig. Es besteht aus einem *Betriebssystemkern* (engl.: operating system kernel) und aus *Dienstprogrammen* (engl.: utility program).

Betriebssysteme



Auftragsverwaltung

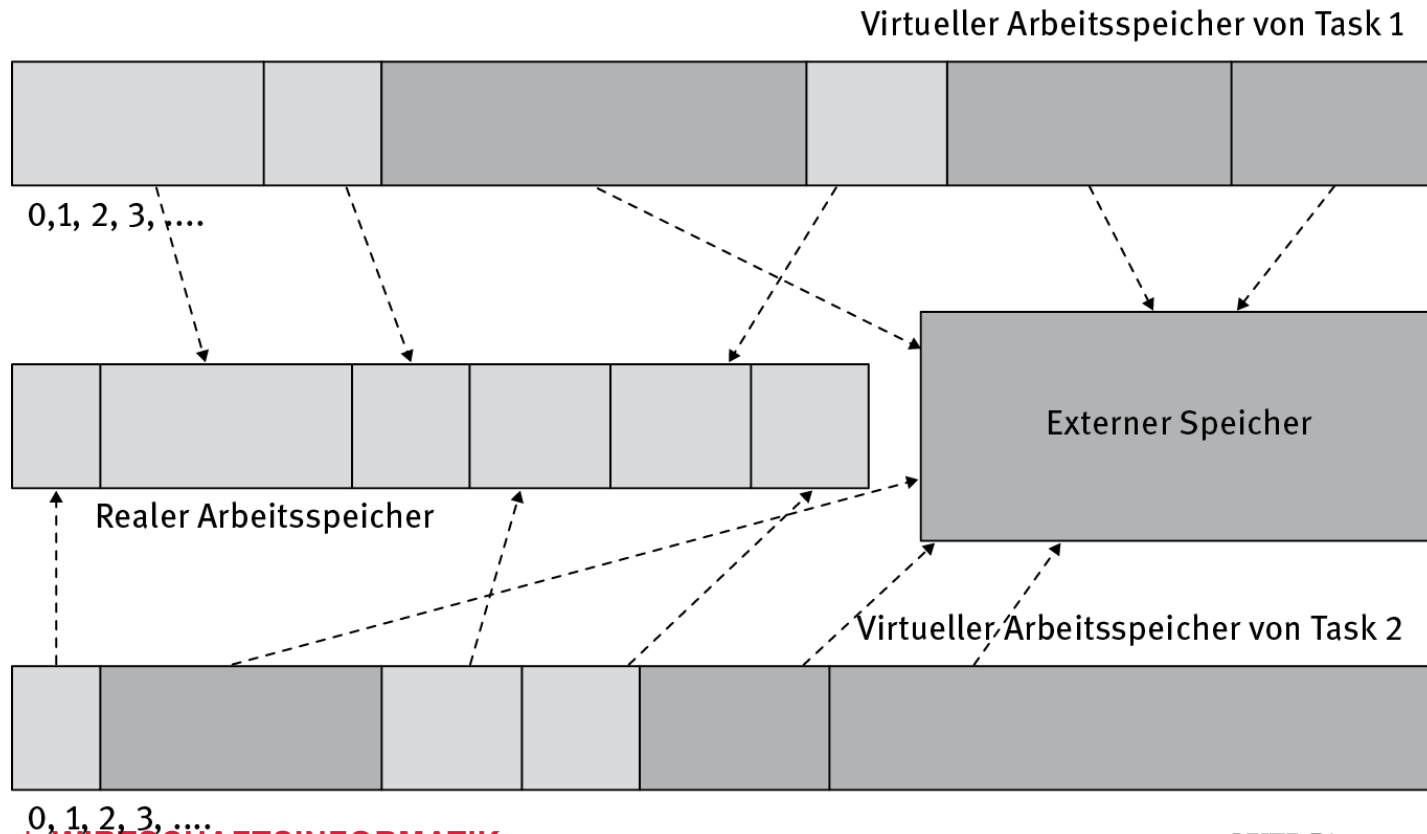
- Unter einem **Task** (deutsch: Aufgabe, engl.: task) versteht man ein ablaufendes Programm im Arbeitsspeicher eines Rechners.
- Ein **Prozess** (engl.: process) ist ein Task, dessen Ablauflogik und Daten im Arbeitsspeicher als eine Einheit aufgefasst und gemeinsam geschützt werden. Möchte ein Prozess auf die Daten eines anderen Prozesses zugreifen, ist *Intraprozesskommunikation* notwendig.
- Ein **Thread** (engl.: thread) ist ein Task, bei dem die Ablauflogik und die Datenbereiche im Arbeitsspeicher als getrennte Einheit betrachtet werden; mehrere parallel ablaufende Instanzen desselben Programms besitzen einen gemeinsamen Datenbereich im Arbeitsspeicher.

Programmbetrieb

- Beim **Einprogrammbetrieb** (engl.: single-tasking mode) werden einzelne Tasks von der Zentraleinheit nacheinander bearbeitet. Es befindet sich jeweils nur ein Task im Arbeitsspeicher, der für seinen gesamten Ablauf alle vorhandenen Betriebsmittel (Prozessoren, Speicher, Kanäle, periphere Geräte) zugeteilt erhält.
- Beim **Mehrprogrammbetrieb** (engl.: multi-tasking mode) können mehrere Tasks von der Zentraleinheit zeitgleich bearbeitet werden. Es befinden sich gleichzeitig mehrere Tasks ganz oder teilweise im Arbeitsspeicher, denen das Betriebssystem bei der Ausführung die benötigten Betriebsmittel abwechselnd zuteilt.

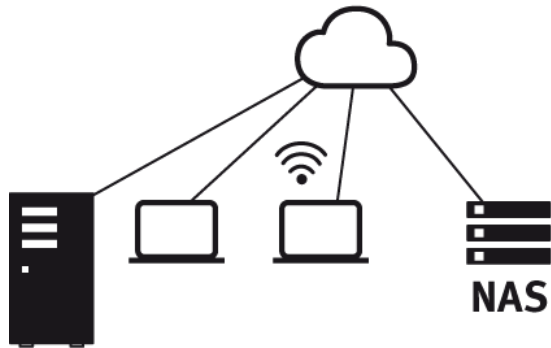
Arbeitsspeicherverwaltung

Virtueller Arbeitsspeicher: Teile des Arbeitsspeichers eines Tasks stammen aus dem realen Arbeitsspeicher oder von externen Speichern

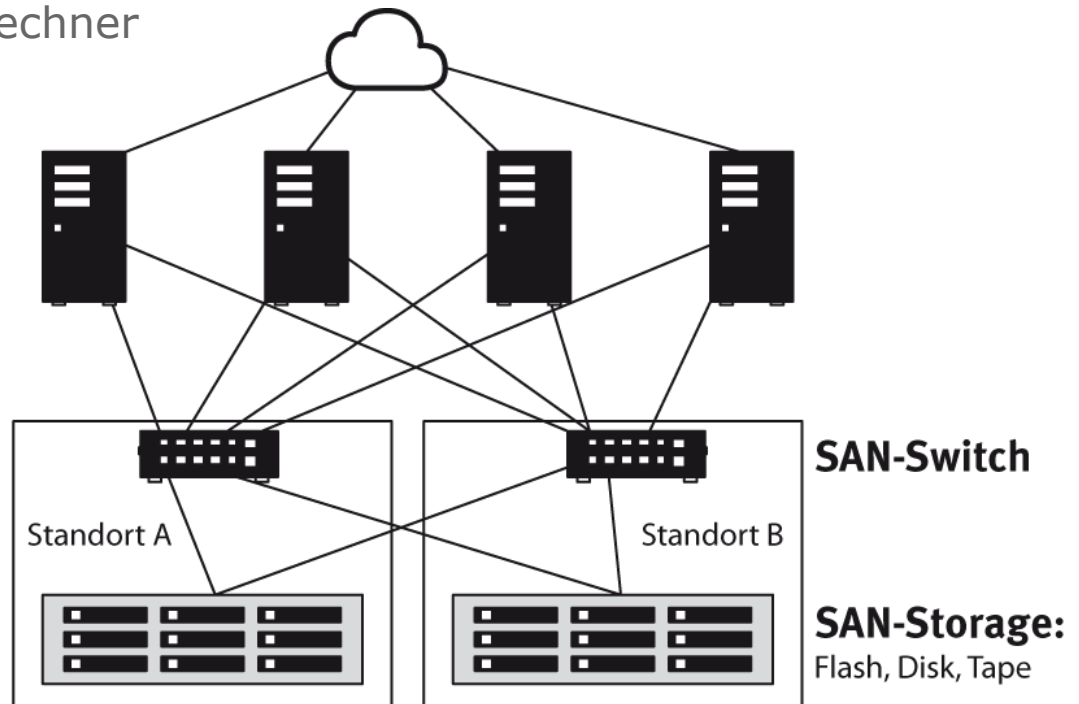


SAN- vs. NAS-Speichersysteme (Hardware)

Über Datennetze angebundene, externe Speicher für mehrere Rechner



Kleinere Systeme, gemeinsames Dateisystem, Bereitstellung von Dateien



Unternehmensweite Speichersysteme, redundant, ausfallssicher, Bereitstellung von Datenbereichen (Blöcken), auf denen bspw. Dateisysteme installiert werden können

Benutzerverwaltung

- Beim **Einbenutzerbetrieb** (engl.: single-user mode) kann zu einem Zeitpunkt nur ein Benutzer den Rechner nutzen. Beim **Mehrbenutzerbetrieb** (engl.: multi-user mode) können zu einem Zeitpunkt mehrere Benutzer den Rechner nutzen. Dementsprechend ist für den Mehrbenutzerbetrieb der Mehrprogrammbetrieb eine Voraussetzung.



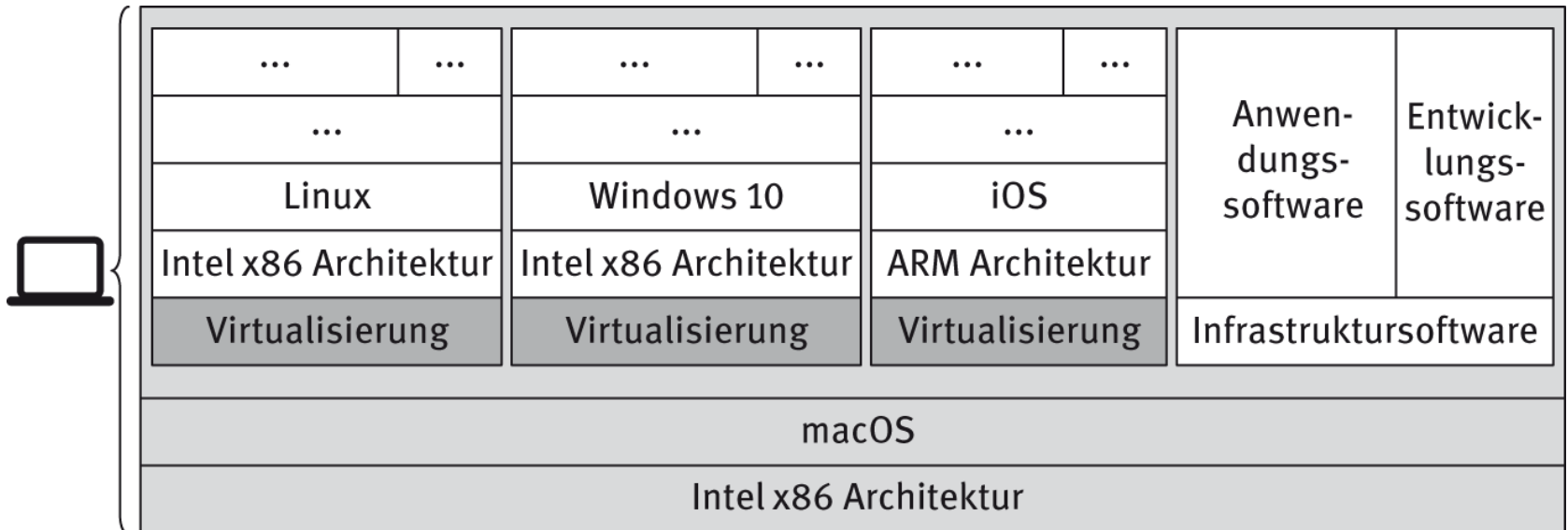
11.7 Virtualisierung

Virtualisierung

Virtualisierung : Nachbildung von physischen Hardwarekomponenten durch Software.

Beispiel: Rechnervirtualisierung

Auf einem physischen Rechner laufen u.a. mehrere logische Rechner



Die wichtigsten Punkte

1. Das Herz jedes Rechners ist die Zentraleinheit, die aus Zentralprozessor und aus Zentralspeicher besteht. Um ein Programm auf dem Zentralprozessor ausführen zu können, müssen sowohl das auszuführende Programm als auch die aktuell zu verarbeitenden Daten in den Zentralspeicher geladen werden. Dieses Modell basiert auf der von-Neumann-Architektur.
2. Programme werden durch Fließbandverarbeitung durchgeführt, wodurch gleichzeitig mehrere Befehle durch einen Prozessor verarbeitet werden können. Interne Funktionseinheiten übernehmen bei der Verarbeitung eines Befehls Teilaufgaben. Durch eine mehrfache Auslegung dieser Funktionseinheiten erhöht sich die Parallelverarbeitung, vor allem wenn Prozessoren mit mehreren Kernen ausgestattet sind. Während über viele Jahre Leistungssteigerungen durch Erhöhung der Taktrate erzielt wurden, werden nun diese durch Erhöhung der Parallelverarbeitung angestrebt.
3. Die meisten heutigen Rechnersysteme nutzen Mikroprozessoren, bei denen ein oder mehrere Zentralprozessoren auf einem Mikrochip untergebracht sind. Für eingebettete Systeme und das Internet der Dinge werden zunehmend Ein-Chip-Computer entwickelt, bei denen auch der Zentralspeicher und Ein-/Ausgabeeinheiten auf einem einzigen Chip integriert werden.
4. Komplexe Programmsysteme werden in höheren Programmiersprachen entwickelt, die von dem Befehlssatz des Prozessors abstrahieren. Programme, die in höheren Programmiersprachen geschrieben sind, werden entweder durch einen Compiler in Maschinensprache übersetzt oder durch einen Interpreter ausgeführt.
5. Komplexe Systeme bestehen aus einer Vielzahl von Programmkomponenten, die untereinander interagieren. Diese Programmkomponenten können in unterschiedlichen Programmiersprachen verfasst sein. Bei der Entwicklung von komplexen Programmsystemen ist es das Ziel, einerseits möglichst wiederverwendbare Komponenten zu verwenden und andererseits Modularität durch lose gekoppelte, ersetzbare Komponenten zu erzielen.
6. Das Betriebssystem hat die Aufgabe von der realen Hardware zu abstrahieren und für Anwendungsprogramme eine möglichst standardisierte Ausführungsumgebung bereitzustellen. Je nach Funktionsumfang des Betriebssystems kann man zwischen Ein- und Mehrprogrammbetrieb, beziehungsweise zwischen Ein- und Mehrbenutzerbetrieb unterscheiden.

Online-Materialien

