
— Informatik I —
**Modul 5: Rechnerarchitekturen
und Rechnerorganisation**



**Universität
Zürich^{UZH}**



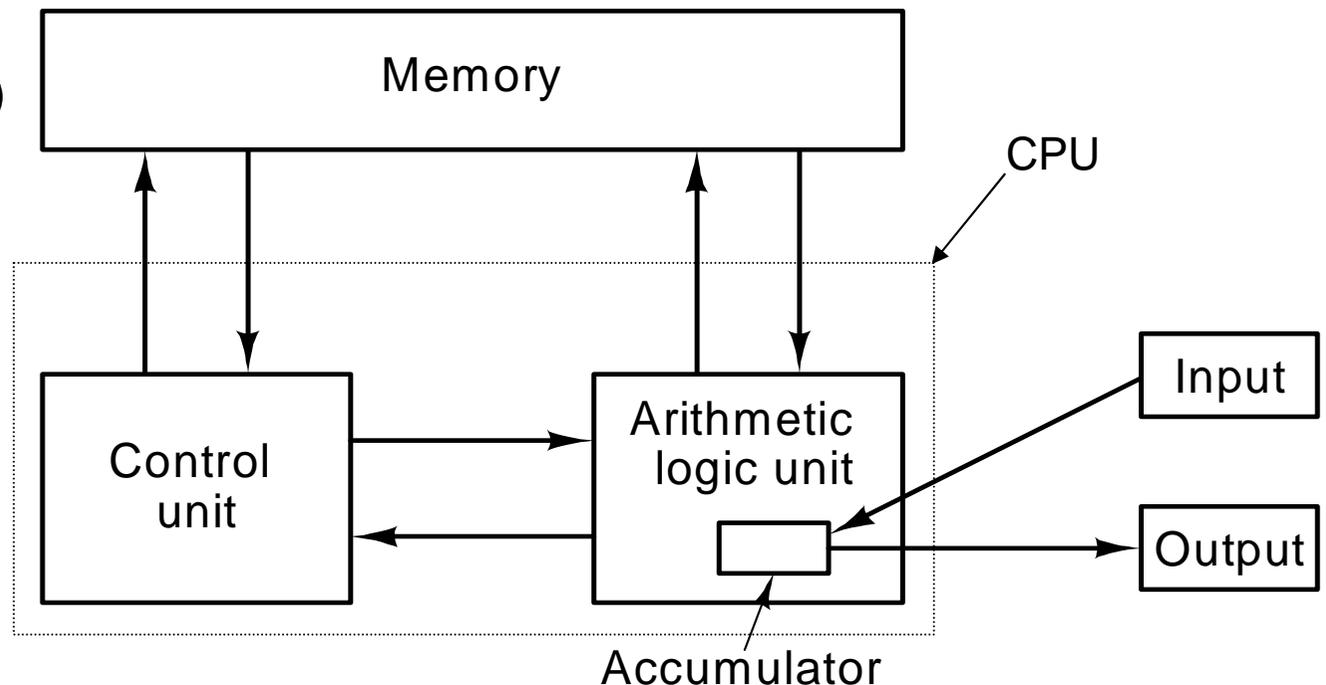
Modul 5: Rechnerarchitekturen & -organisation

- ❑ von-Neumann Architektur
- ❑ Aufbau und Funktionsweise
- ❑ Organisation
- ❑ Peripherie
- ❑ Technologieentwicklung

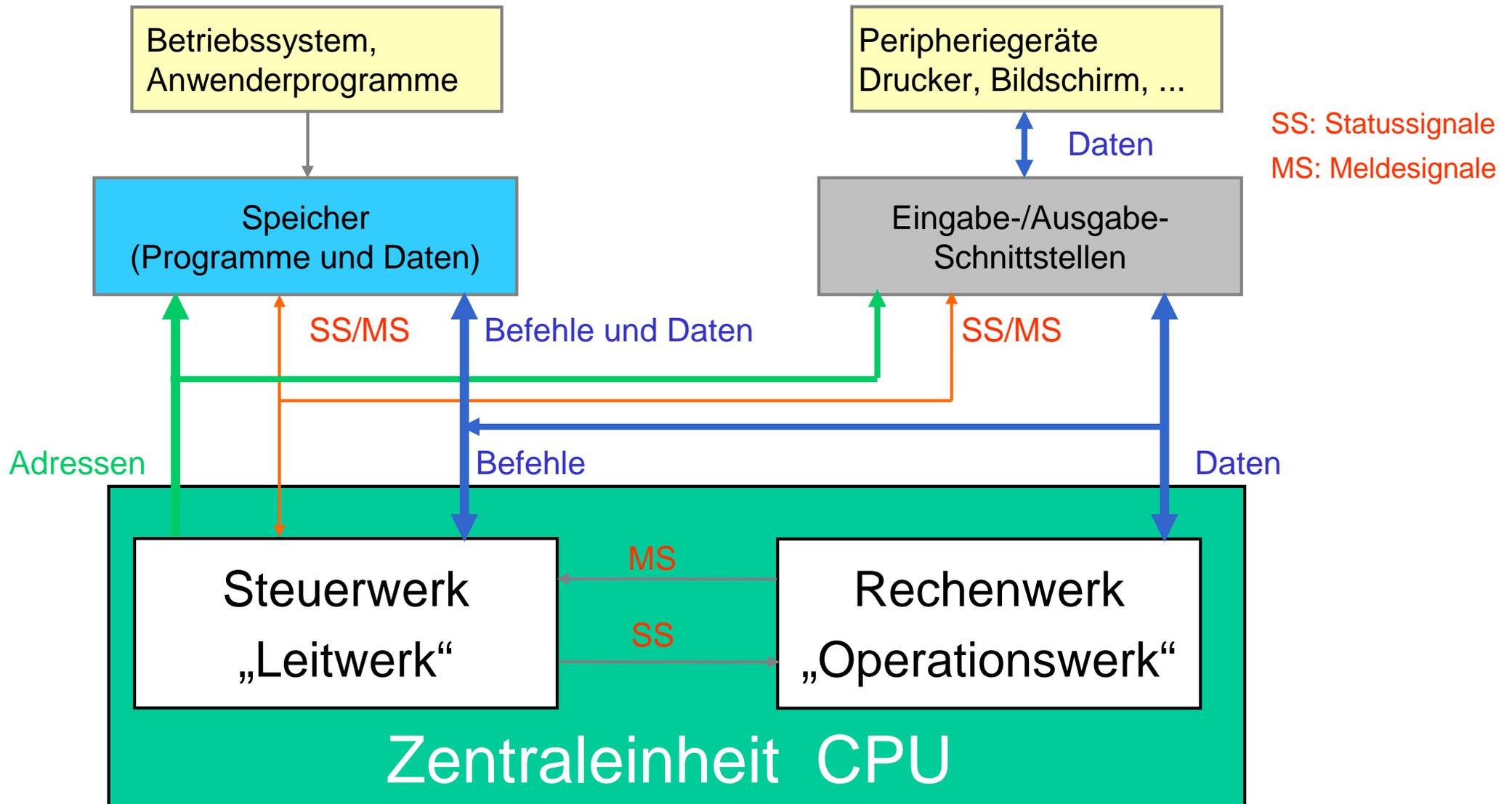
Die von-Neumann Architektur

- Die von-Neumann Architektur stellt die Basis von fast allen heutigen Hardware-Architekturen dar.
- Die Architektur umfaßt die folgenden sechs Hauptkomponenten:

- Rechner (CPU, Central Processing Unit)
 - Steuerungseinheit (Control Unit)
 - ALU
- Speicher (Memory)
- Ein/Ausgabeeinheiten (Input/Output Units)



Digitalrechner nach dem von-Neumann Konzept



EVA-Prinzip

- Computer arbeiten nach dem EVA-Prinzip (*Eingabe, Verarbeitung und Ausgabe*).
 - **Eingabe:** Über eine Eingabeeinheit wie z.B. eine Tastatur, eine Maus, einen Memorystick gelangen Daten in den Computer.
 - **Verarbeitung** dieser Daten findet dann in der Zentraleinheit statt.
 - **Ausgabe** erfolgt über ein Ausgabegerät wie Bildschirm, Drucker, Festplatte.

EVA in den verschiedenen Computergenerationen

Computergeneration	Eingabeeinheit	Verarbeitungseinheit	Ausgabeeinheit
Röhrenrechner	Schalttafeln	Trommelspeicher	Leuchtanzeigen, Drucker mit Endlospapier
Transistorrechner	Lochkarten	Magnetkernspeicher	Drucker mit Endlospapier
IC-Rechner	Terminal mit Tastatur, Magnetband	IC-Schaltungen	Terminal, Drucker mit Endlospapier, Magnetband
Mikrocomputer	Tastatur, Maus, Festplatte usw.	Mikroprozessor	Bildschirm, Festplatte, Farb-/Laserdrucker usw.

Das EVA-Verfahren lässt sich durch die gesamte Geschichte der Computer verfolgen

Modul 5: Rechnerarchitekturen & -organisation

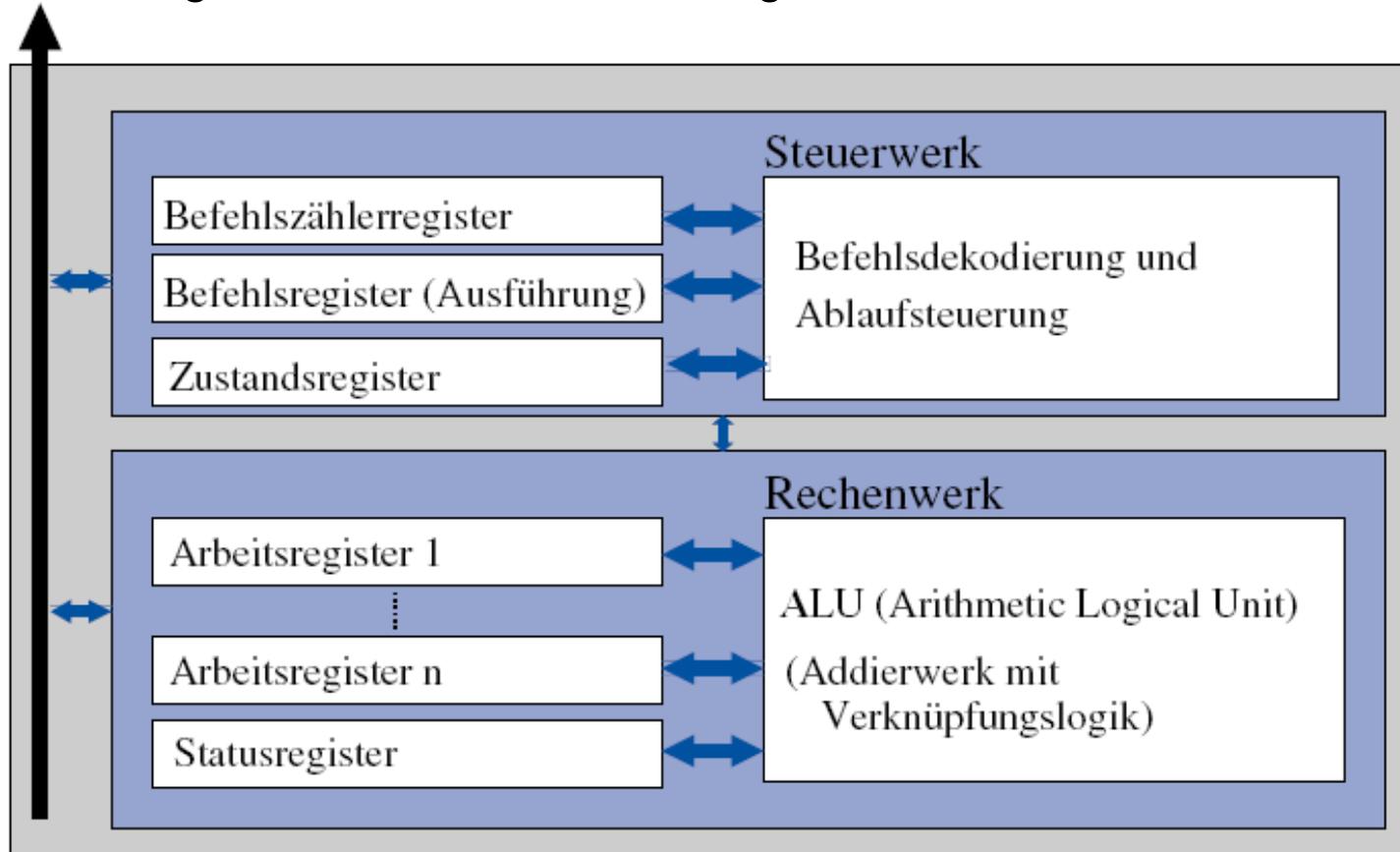
- ❑ von-Neumann Architektur
- ❑ Aufbau und Funktionsweise
- ❑ Organisation
- ❑ Peripherie
- ❑ Technologieentwicklung

Zentraleinheit

- Die Zentraleinheit besteht im wesentlichen aus den Komponenten der *Hauptplatine* (*Mainboard* oder *Motherboard*).
 - Der **Mikroprozessor** (*CPU = Central Processing Unit*): Ausführung der Programme, Steuerung und Verwaltung der Hardware verantwortlich.
 - Der **RAM-Arbeitsspeicher** (*RAM = Random Access Memory*): enthält Programme, die gerade ausgeführt werden, und verwendete Daten.
 - Der **ROM-Speicher** (*ROM = Read-only-Memory*): enthält meist ein Programm (*BIOS* bei IBM-PCs), das beim Einschalten die wichtigsten Hardwarekomponenten überprüft und dann das *Booten* des Betriebssystems von einem Speichermedium (Festplatte, CD) veranlaßt.
 - Die **Busse und Schnittstellen**: Kommunikation zwischen einzelnen Bestandteilen des Mainboards, zum Anschluss von Peripheriegeräten (z.B. Grafikkarten, Netzwerkkarten, Festplatten, Druckern).
 - Der **Chipsatz**: fest auf dem Mainboard untergebrachte Schaltkreise, z.B. Für die Steuerung sämtlicher Anschlüsse des Mainboards.

Mikroprozessor (1)

- Mikroprozessoren sind integrierte elektronische Schaltkreise
 - Während sie ursprünglich nur aus wenigen Transistoren bestanden, enthalten sie heute mehrere Millionen Transistoren, wobei die Integrationsdichte auf Grund des technologischen Fortschritts ständig zunimmt.



Mikroprozessor (2)

- Ein Mikroprozessor setzt sich aus folgenden Komponenten zusammen:
 - **ALU** (*Arithmetic Logical Unit=Arithmetisch-logische Einheit*): **Rechenwerk**, das mathematische Operationen und logische Verknüpfungen durchführt.
 - **Register**: spezielle Speicherplätze innerhalb des Prozessorkerns. Die ALU rechnet mit den Werten, die sich in den Arbeitsregistern befinden. Typische Prozessoren verfügen nur über relativ wenige Register, wie z. B. 2, 4, 6, 8 usw.
 - **Steuerwerk**: übernimmt mittels zweier spezieller Register (Befehlszählerregister und Befehlsregister) die Kontrolle über die Ausführung des Programmcodes und initiiert andere Steuerungsfunktionen, verwaltet auch den Stack-Zeiger.
 - *Befehlstabelle (Instruction Table)* erlaubt Maschinenbefehle eines Programms zu dekodieren.
 - Über **Busse** (*Datenleitungen*) ist der Prozessor mit den Komponenten verbunden:
 - *Datenbus*: Austausch von Daten mit dem Arbeitsspeicher
 - *Adressbus*: Übertragen der zugehörigen Speicheradressen
 - *Steuerbus*: Ansteuerung der Peripherie-Anschlüsse

Register

□ Register

- sind **prozessorinterne Speicherplätze**, die jeweils ein (binäres) Datum bestimmter Länge (z. B. 32-Bit) aufnehmen können.
- besitzen zu anderen Prozessorkomponenten, wie Ablaufsteuerung und Verknüpfungslogik, sehr enge Verbindungen. Nur mit den Daten in den Arbeitsregistern können direkte logische Operationen durchgeführt werden.

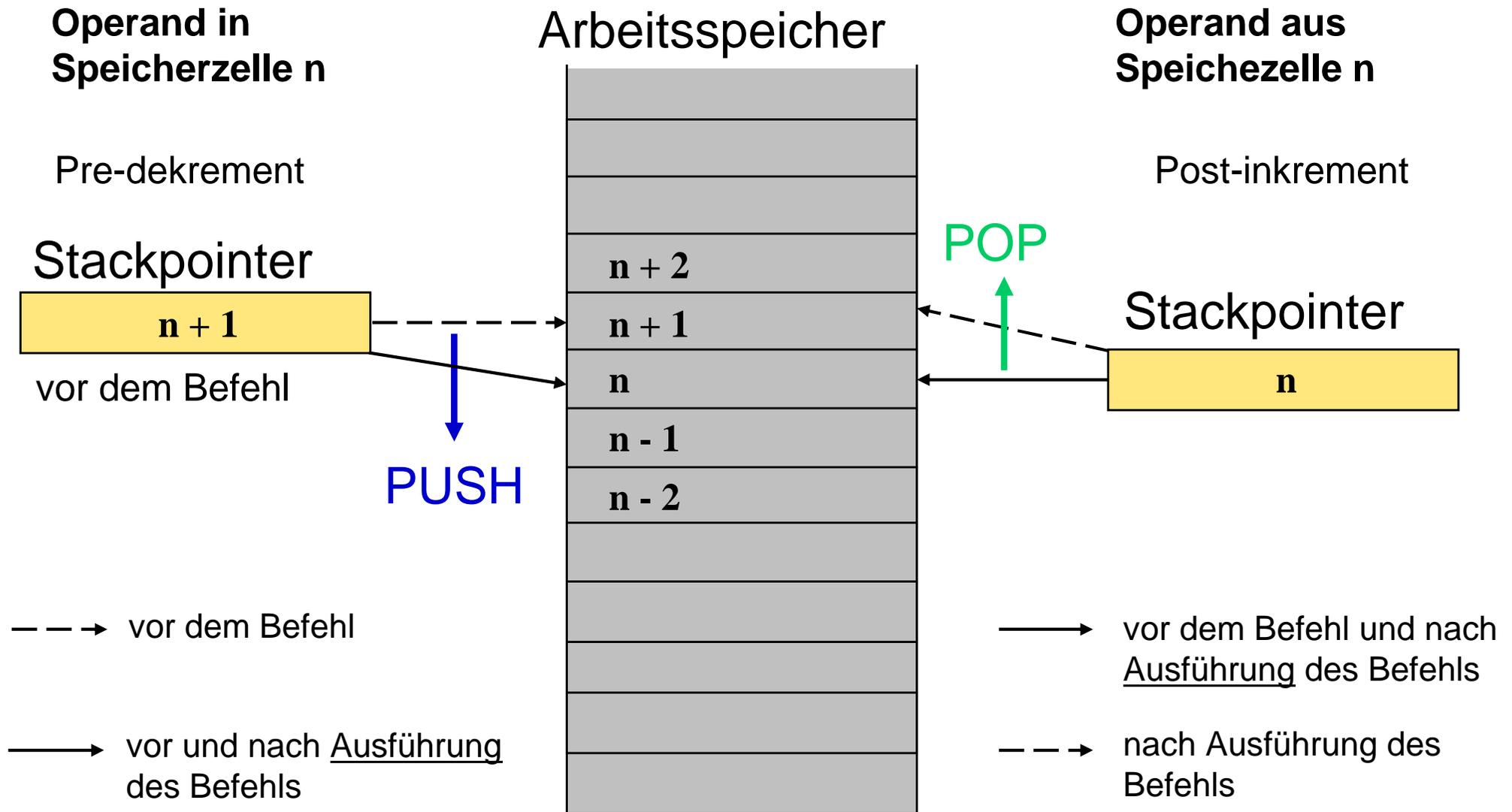
□ Registertypen

- **Arbeitsregister** werden in Befehlen durch Namen wie z.B. D1...D7 oder EAX...EDX angesprochen. Sie können Daten (Datenregister) und je nach Prozessortyp auch Adressen (Adressregister) aufnehmen.
- Das **Befehlszählerregister** beinhaltet immer die Adresse des nächsten auszuführenden Befehls (häufiger Name ist *Instruction Pointer = IP*).
- Das **Befehlsregister** kann einen (binären) Maschinenbefehl aufnehmen.
- Das **Stackregister** speichert den Prozessorstatus und den Programmzähler ab.

(Laufzeit-)Stack oder „Kellerspeicher“

- ❑ Besonderer Speicherbereich: normalerweise im Arbeitsspeicher angelegt (software stack) ist nach dem **Kellerprinzip** (*LIFO: Last-in-first-out*) organisiert und wird Kellerspeicher genannt.
- ❑ Funktion:
 - Abspeichern des Prozessorstatus und des Programmzählers beim Unterprogrammaufruf und Aufruf von Unterbrechungs-Routinen
 - Parameterübergabe
 - Kurzzeitige Lagerung von Daten bei der Ausführung
- ❑ Hardware-Unterstützung
 - Stackregister (Stapelzeiger, Stack Pointer SP):
 - Enthält die Adresse des zuletzt in den Stack eingetragenen Datums
 - Spezielle Befehle zur Datenübertragung in den bzw. aus dem Stack:
 - **PUSH**: Inhalt eines Registers wird in den Stack übertragen
 - **POP (PULL)**: Inhalt eines Registers wird vom Stack geladen

Verwaltung des Stacks



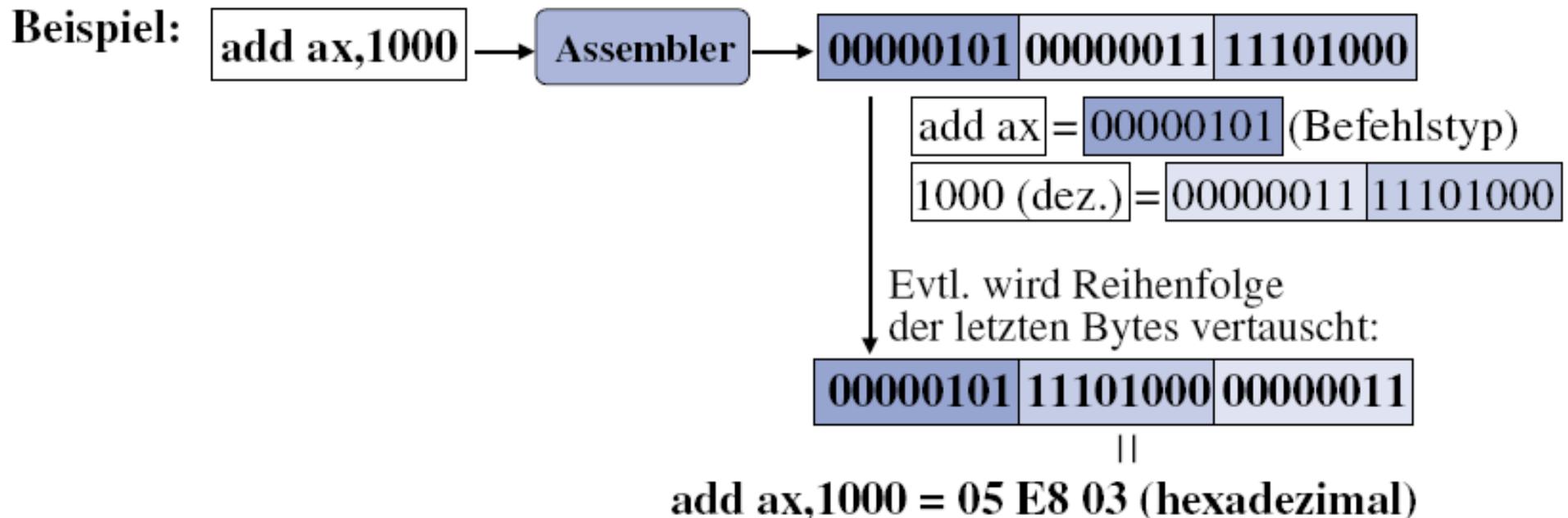
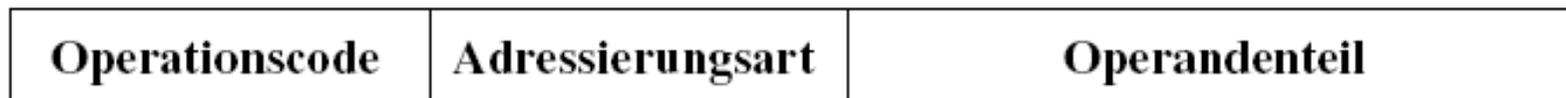
Funktionsweise eines Prozessors

1. Das Befehlszählerregister des Steuerwerks enthält die Adresse des nächsten Maschinenbefehls. Die Adresse des Befehls wird über den Adressbus an den Arbeitsspeicher übermittelt.
2. Der Befehl wird aus dem Arbeitsspeicher über den Datenbus in das Befehlsregister übertragen. Mittels Dekodierlogik wird der Befehl analysiert und die Ausführung angestoßen.
3. Der Befehl wird ausgeführt; abhängig vom jeweiligen Befehl wird dabei zusätzlich das Lesen von Daten aus dem Arbeitsspeicher, die Ansteuerung von Peripherieschnittstellen, das Rechnen in der ALU oder die Durchführung eines Sprungs im Programm erforderlich. Der Status der jeweiligen Operation wird im *Statusregister (Flagregister)* angezeigt
4. Falls ein Sprung stattfand, wird das Befehlszählerregister auf die entsprechende neue Adresse gesetzt, ansonsten wird das Befehlszählerregister um 1 erhöht.
5. Der Prozessor fährt wieder mit dem 1. Schritt fort.

Maschinenbefehle (1)

- Aufbau und Art von Maschinenbefehlen unterscheiden sich bei verschiedenen Prozessortypen und -herstellern sehr stark.
- **Gemeinsame Grundprinzipien:**

Aufbau eines binären Maschinenbefehls:



Maschinenbefehle (2)

- Maschinenbefehle bestehen aus mehreren Teilen. Diese umfassen im allgemeinen:
 - den eigentlichen Befehl (OP-CODE),
 - einen Operandenteil mit Angabe der Adressierungsart und
 - einen Operandenwert oder eine Adresse.

- Jedes Bit hat eine spezielle Bedeutung und durch die Interpretation im Steuerwerk werden die verschiedenen Reaktionen veranlasst.

- Je nach Art des Befehls können Maschinenbefehle auch verschiedene Längen und eine unterschiedliche Anzahl von Operanden haben.

Befehlstypen

- *Arithmetische und logische Befehle* erlauben Berechnungen und logische Entscheidungen in einem Programm.
- *Transportbefehle* dienen zum Transport der Daten zwischen Prozessor, Arbeitsspeicher und Ein-/Ausgabeeinheiten.
- *Prozessorkontrollbefehle* werden zur Priorisierung von wichtigen Aufgaben und allgemein zur internen Organisation und zur Verwaltung des Prozessors benötigt.
- *Sprungbefehle* ermöglichen unter Berücksichtigung entsprechender logischer oder arithmetischer Bedingungen Abweichungen vom linearen Fluß eines Programms (nichtlineare Programmausführung).
Mit diesen wird, zusammen mit arithmetischen und logischen Befehlen, die Vielfalt im Verhalten eines Programms und die sprichwörtliche „Intelligenz“ von heutigen Computern erreicht.

Nichtlineare Programmausführung

- Gründe für eine nichtlineare Programmausführung:
 - Sprünge und Verzweigungen (jumps, branches)
 - Prozeduraufrufe, Unterprogramme, Methodenaufrufe
(Procedure calls, subroutines, method invocation)
 - Multiple Threads, parallele Prozesse, Coroutinen
(Multithreading, parallel processes, co-routines)
 - Hardware-Unterbrechungen aus prozessorexternen Gründen
(Hardware interrupts)
 - Software-Unterbrechungen oder Traps, aus prozessorexternen Gründen
(Software interrupts)

- Nichtlineare Programmausführung stellt den Normalfall dar!

Grundsätzliche Befehlsformate



(a)



(b)



(c)



(d)

□ Beispiel:

$C := A + B$

a) „Zero-address“-Befehl

- Kellerspeicherarchitekturen: `push A; push B; ADD; pop C`

b) „One-address“-Befehl

- Der Akkumulator ist Operand und Resultat: `load A; ADD B; st C`

c) „Two-address“-Befehl

- Ein Operand wird zum Resultat: `ADD B, A; move A, C`

d) „Three-address“-Befehl

- $C := A + B$: `ADD C, A, B`

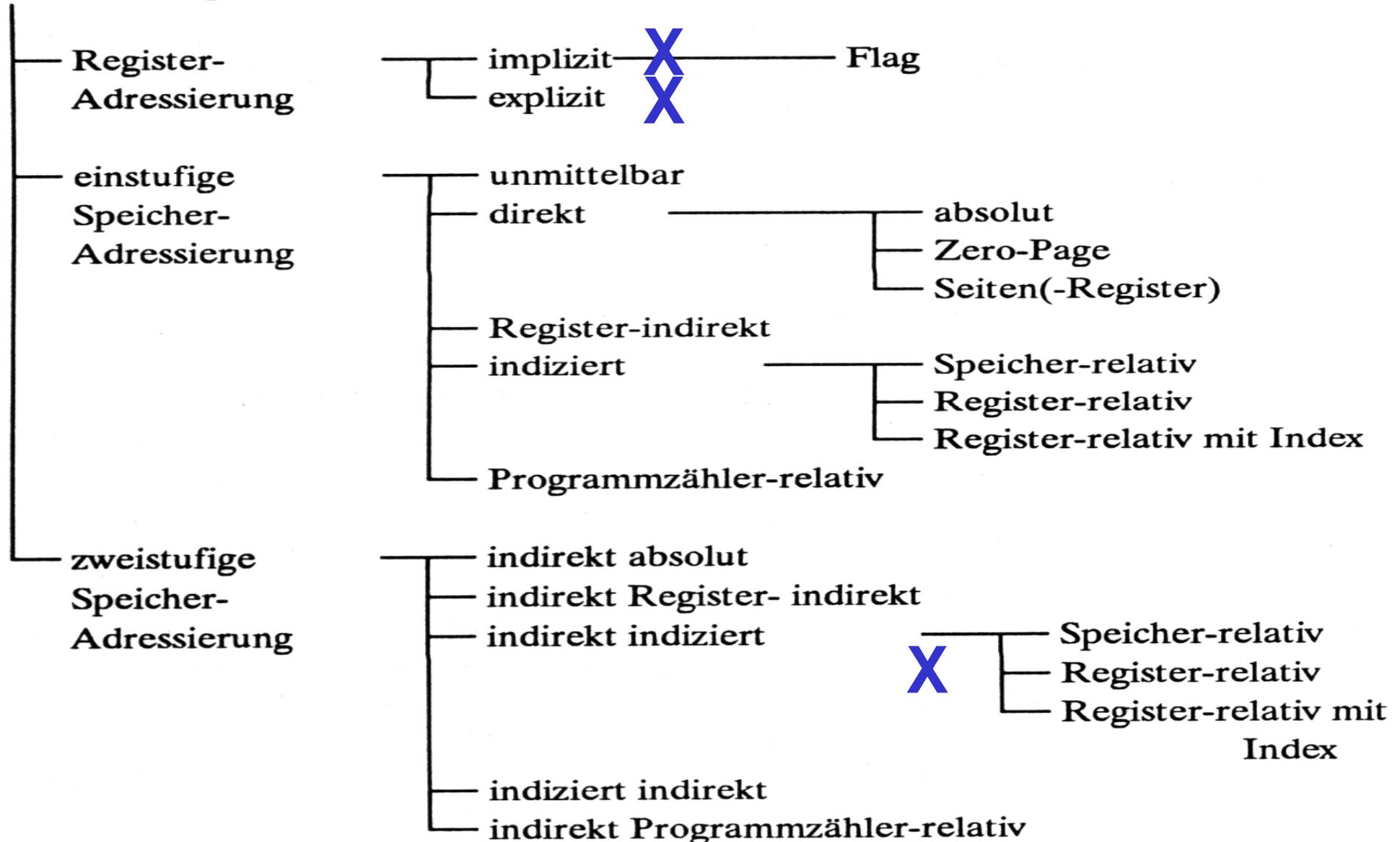
Adressierungsarten

- Die Adressierungsarten bieten verschiedene Möglichkeiten eines Prozessors, die Adresse eines Operanden oder eines Sprungziels im Speicher zu berechnen.
- Früher:
 - Adresse der Operanden und Sprungziele absolut im Befehl vorgegeben
- Nachteile:
 - Absolute Adressen müssen bereits zur Programmierzeit festgelegt werden
 - Programme sind lageabhängig im Speicher
 - Bei Tabellenzugriffen im Speicher muß die Adresse im Befehl geändert werden
 - Keine Festwertspeicher als Programmspeicher möglich
- Heute:
 - Adresse wird zur Laufzeit berechnet (dynamische Adreßberechnung)



Adressierungsarten — Überblick (1)

Adressierungsarten



Adressierungsarten — Überblick (2)

- Register-Adressierung
 - Operand steht bereits im Register
 - ➔ kein Speicherzugriff erforderlich

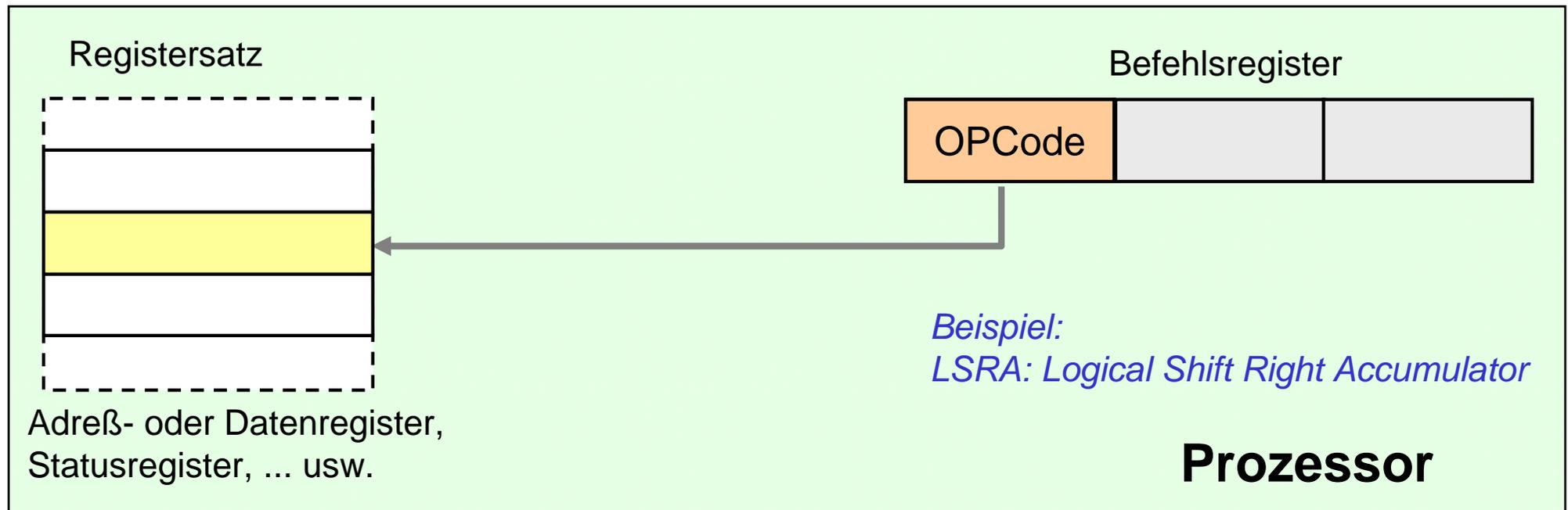
- Einstufige Speicher-Adressierung
 - Eine Adreßberechnung zur Ermittlung der effektiven Adresse notwendig, d.h. keine mehrfachen Speicherzugriffe zur Adreßermittlung

- Zweistufige Speicher-Adressierung
 - Mehrere sequentielle Adressberechnungen und Speicherzugriffe.
 - Ergebnis der ersten Berechnung liefert die Adresse einer Speicherzelle, deren Inhalt wieder eine Adresse oder ein Offset zur weiteren Berechnung ist

Implizite Adressierung

(inhärente Adressierung, implied-, inherent addressing)

- ❑ Die Nummer, d.h. die effektive Adresse des angesprochenen Registers ist codiert im Operations-Feld (OpCode) enthalten
- ❑ **Assemblerschreibweise: <Mnemo> A (A Akkumulator)**
- ❑ **Effektive Adresse: EA ist codiert im OpCode enthalten**



Explizite Register-Adressierung

(*register operand addressing*)

- ❑ Die Adresse (Nummer) des Registers wird im Operandenfeld des Befehls angegeben.
- ❑ **Assemblerschreibweise:** <Mnemo> Ri (Register i)
- ❑ **Effektive Adresse:** $EA = i$



Indizierte Adressierung

(indexed addressing, relative Adressierung)

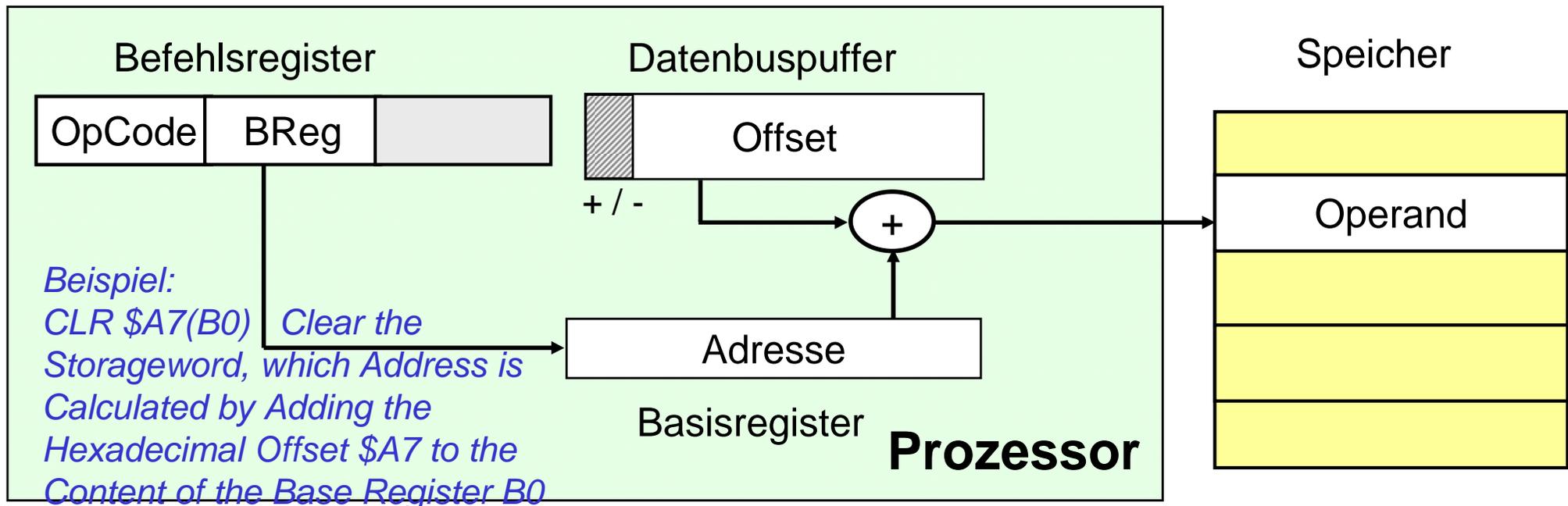
- Bei ihr wird die effektive Adresse durch die Addition des Inhalts eines Registers zu einem angegebenen Basiswert berechnet. (Adreßdistanz zu einem Basiswert, Tabellenverarbeitung)

- Je nachdem, in welcher Form der Basiswert vorgegeben wird, kann man unterscheiden zwischen:
 - Speicherrelative Adressierung
 - Registerrelative Adressierung

Register-relative Adressierung

(register relative addressing, based mode)

- Basiswert befindet sich in Basisregister, auf das durch das BReg-Feld im OpCode verwiesen wird. Im Befehl wird ein Offset angegeben, der zum Inhalt des Basisregisters addiert wird.
- **Assemblerschreibweise:** $\langle \text{Mnemo} \rangle \langle \text{Offset} \rangle (\text{Bi})$
- **Effektive Adresse:** $EA = (\text{Bi}) + ((\text{PC}) + 1)$



Peripheriespeicher (Massenspeicher)

- Eine Möglichkeit, Massenspeicher zu unterscheiden, ist deren physikalisches Schreib- und Leseverfahren.
 - *Magnetische Datenträger*: Hier werden die Bits durch magnetische Bereiche mit gegensätzlicher Polarität dargestellt. Hierzu gehören z. B. die Festplatte (Hard Disk, HD), das Diskettenlaufwerk, ZIP-/JAZ-Laufwerke, aber auch Bandlaufwerke (Streamer).
 - *Optische Datenträger*: Hier werden die Daten auf einer reflektierenden Metallfläche gespeichert, die von einem Laserstrahl abgetastet wird. Die Bits werden durch hineingebohrte Löcher (*Pits*) und unveränderte (nicht gebohrte) Stellen (*Land*) dargestellt.
 - *Magneto-optische (MO) Datenträger*: Hier wird ein Mischverfahren aus magnetischen und optischen Schreib- und Lesevorgängen verwendet.
- **Technologie:**
 - Relativ schnell, immer kleiner (3.5“, 2.5“, 1.8“), recht zuverlässig, sehr preisgünstig (1-10 Rp / GByte für 3.5“ HDs)
 - Solid State Disk (SSD) ist ein Festkörperlaufwerk ohne rotierende Elemente

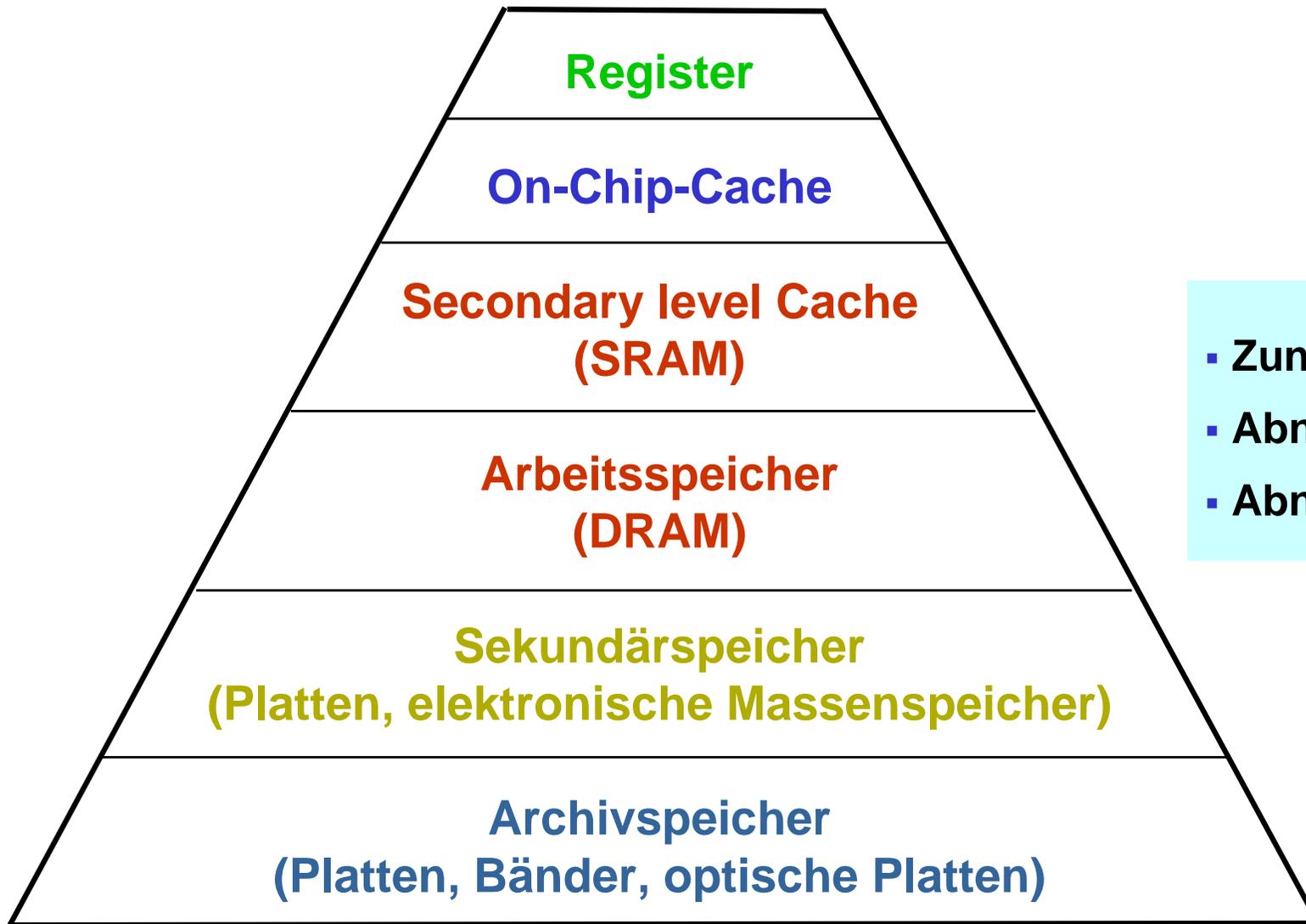
Speicherhierarchie (1)

- Ein *technologisch einheitlicher* Speicher mit *kurzer Zugriffszeit* und *großer Kapazität* ist aus *Kostengründen* i.a. nicht realisierbar.

- Lösung:
 - Schichtenweise Anordnung verschiedener Speicher und Verschiebung der Information zwischen den Schichten (Speicherhierarchie)
 - Speicherhierarchie zum Ausgleich der unterschiedlichen Zugriffszeiten der CPU und des Hauptspeichers.

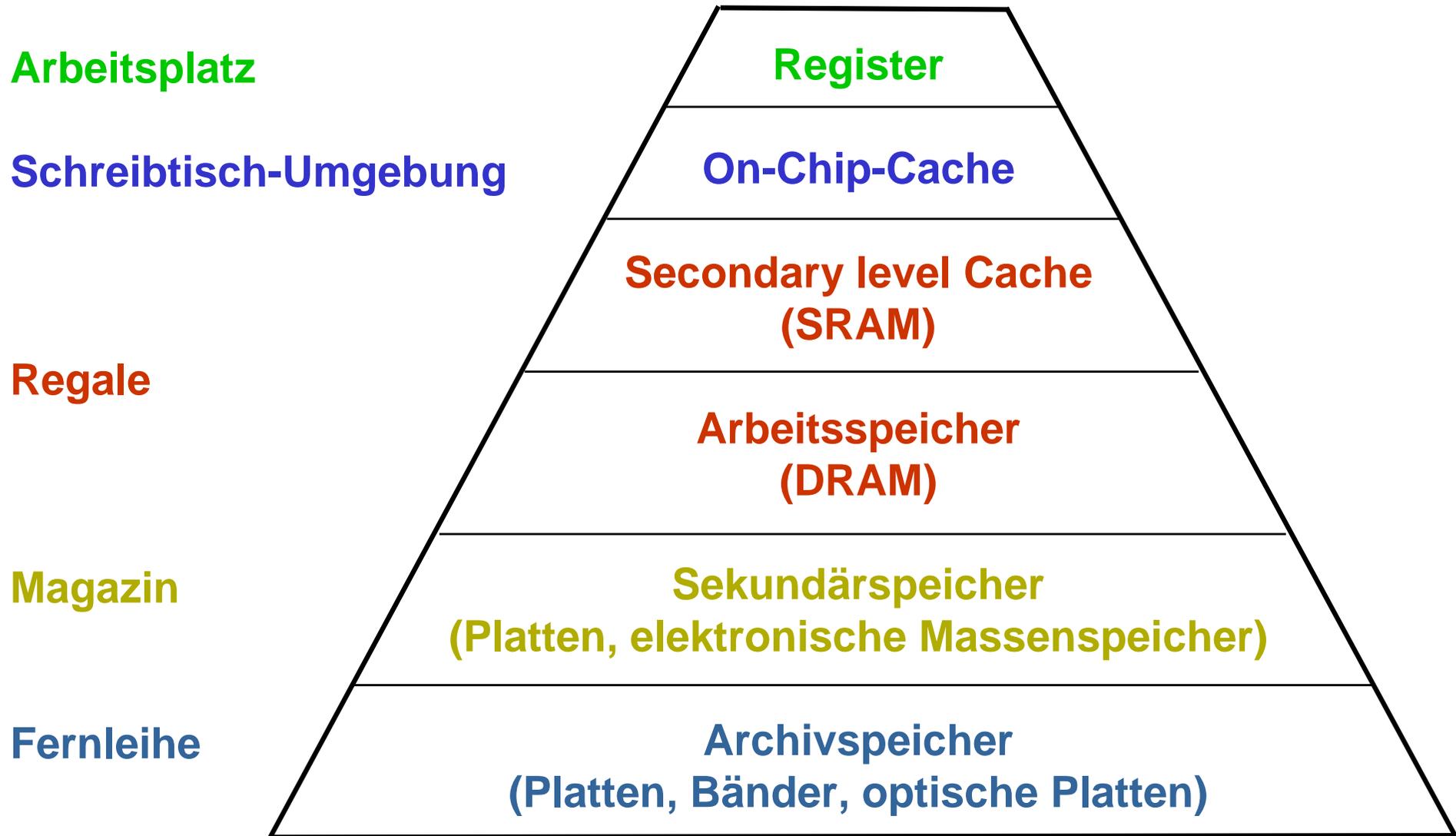
- Zwei Strategien sind möglich:
 - Cache-Speicher: Kurze Zugriffszeiten
 - ➔ Beschleunigung des Prozessorzugriffs
 - Virtueller Speicher:
 - Vergrößerung des tatsächlich vorhandenen Hauptspeichers
 - Z.B. bei gleichzeitiger Bearbeitung mehrerer Prozesse.

Speicherhierarchie (2)



- Zunehmende Kosten/Byte
- Abnehmende Kapazität
- Abnehmende Zugriffszeit

Speicherhierarchie (3) — Vergleich



Cache — Problem und Lösungsidee

□ Problem:

- Die Buszykluszeit moderner Prozessoren ist erheblich kürzer als die Zykluszeit preiswerter, großer DRAM-Bausteine
 - Dies zwingt zum Einfügen von Wartezyklen.
 - SRAM-Bausteine hingegen, die ohne Wartezyklen betrieben werden können, sind jedoch klein, teuer und besitzen eine höhere Verlustleistung.
- Nur relativ kleine Speicher können derartig aufgebaut werden.

□ Lösung:

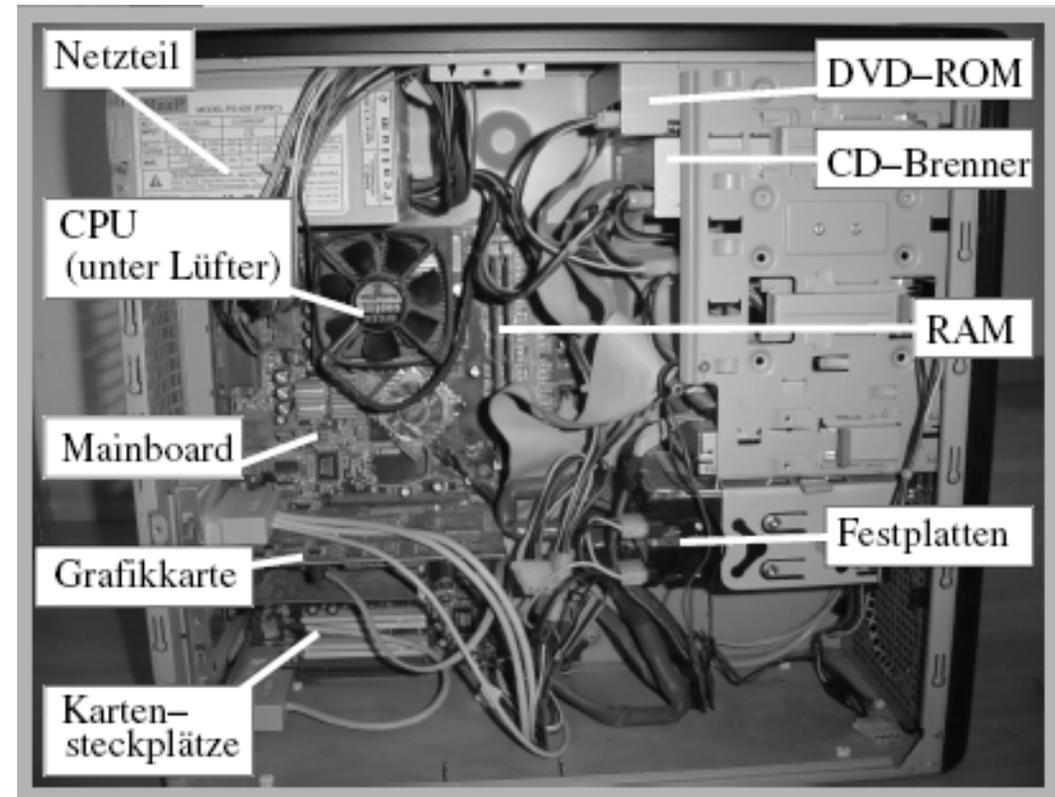
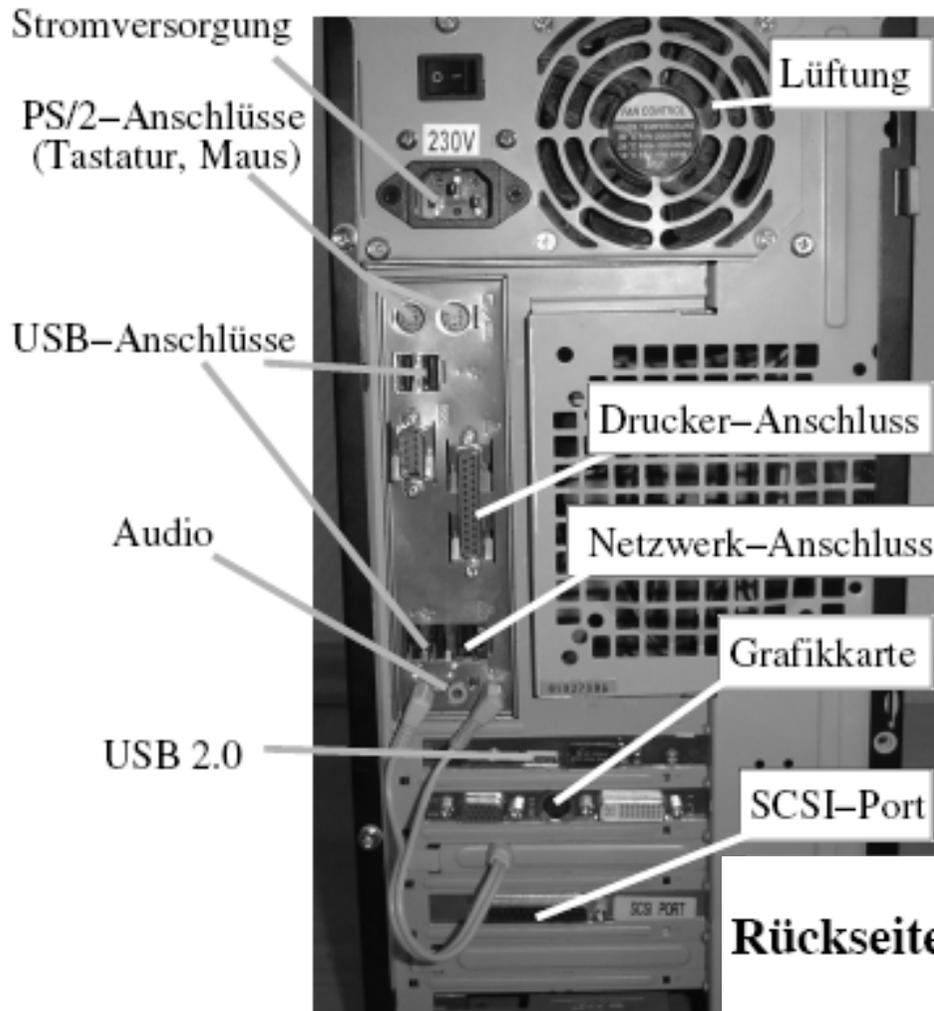
- Zwischen den Prozessor und den relativ langsamen, aber billigen Arbeitsspeicher aus DRAM-Bausteinen legt man einen kleinen, schnellen Speicher aus SRAM-Bausteinen, den sogenannten Cache-Speicher.

- Unter einem **Cache-Speicher** versteht man allgemein einen kleinen, schnellen Pufferspeicher, der vor einen langsamen, größeren Speicher geschaltet wird, um dessen Zugriffszeit zu verbessern.

Modul 5: Rechnerarchitekturen & -organisation

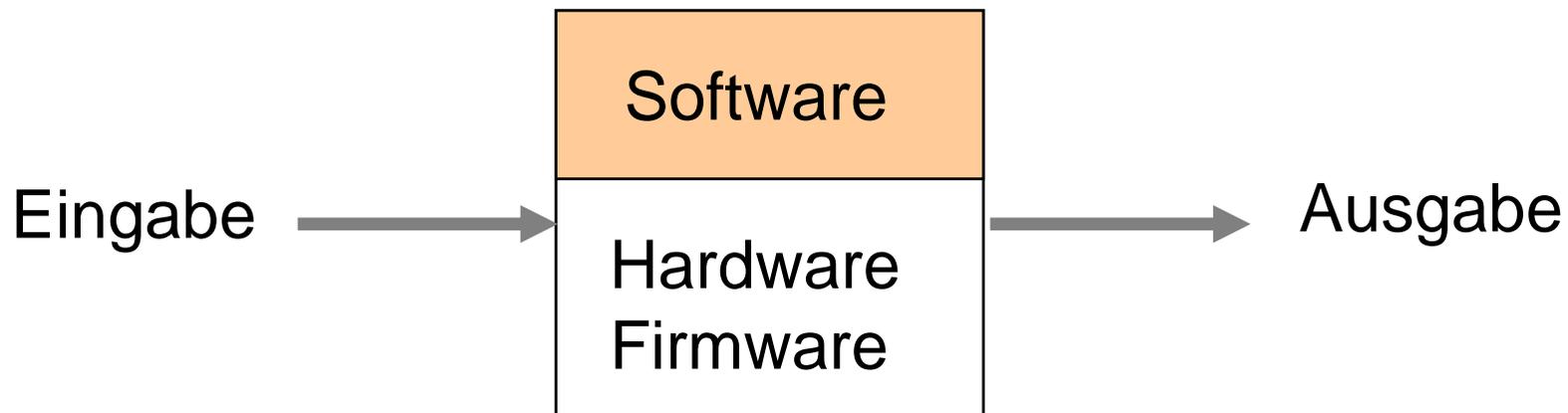
- ❑ von-Neumann Architektur
- ❑ Aufbau und Funktionsweise
- ❑ **Organisation**
- ❑ Peripherie
- ❑ Technologieentwicklung

Die heutigen Rechner (Personal Computer, PC)

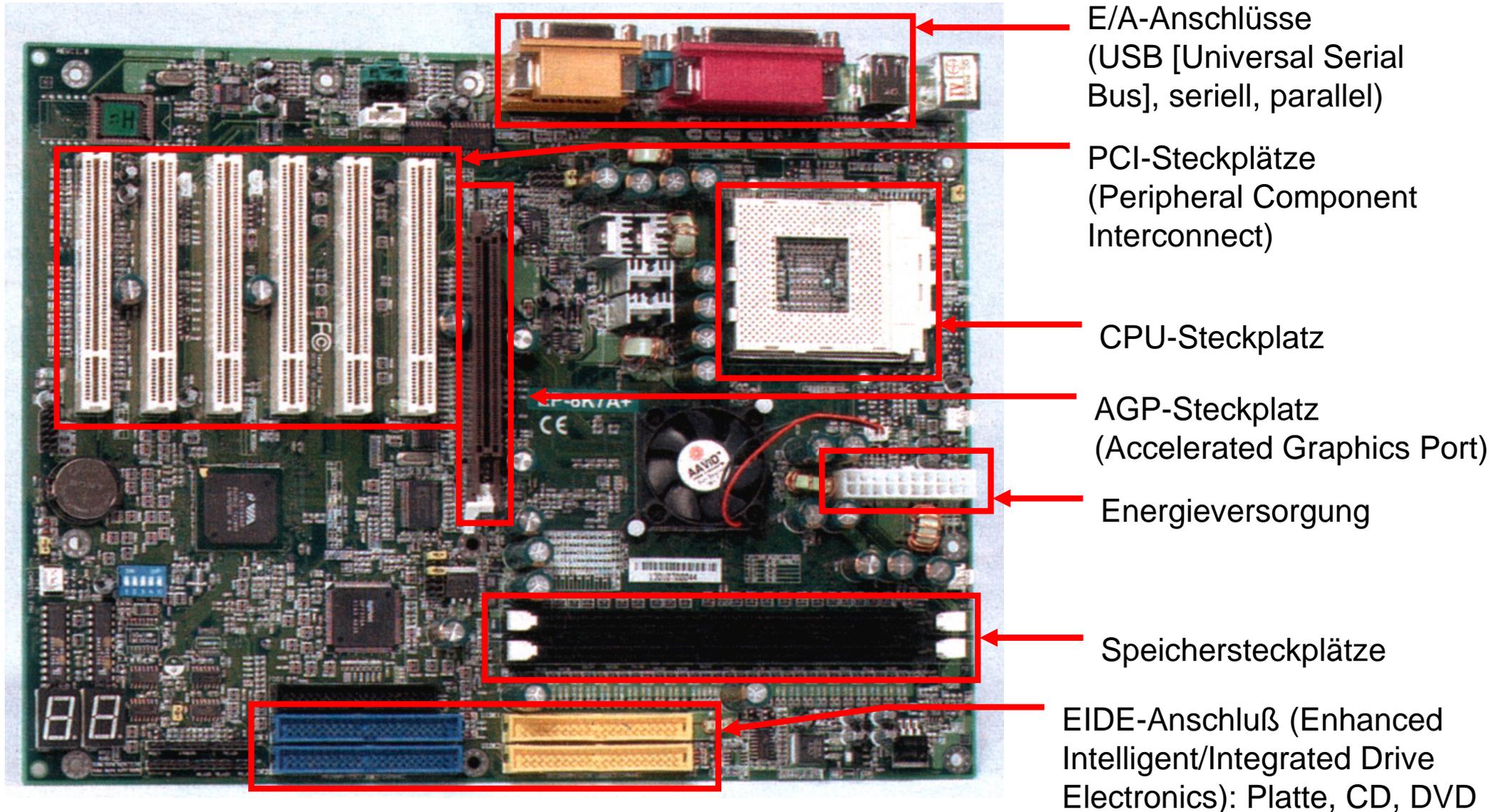


Komponenten eines Digitalrechners

- ❑ **Hardware (HW)**
Alle mechanischen und elektronischen Bauelemente
- ❑ **Software (SW)**
Alle Programme, die auf dem Rechner ablaufen
- ❑ **Firmware (FW)**
Mikroprogramme in ROMs, Mittelstellung zwischen Hardware und Software



Motherboard/Mainboard



Chip-Satz

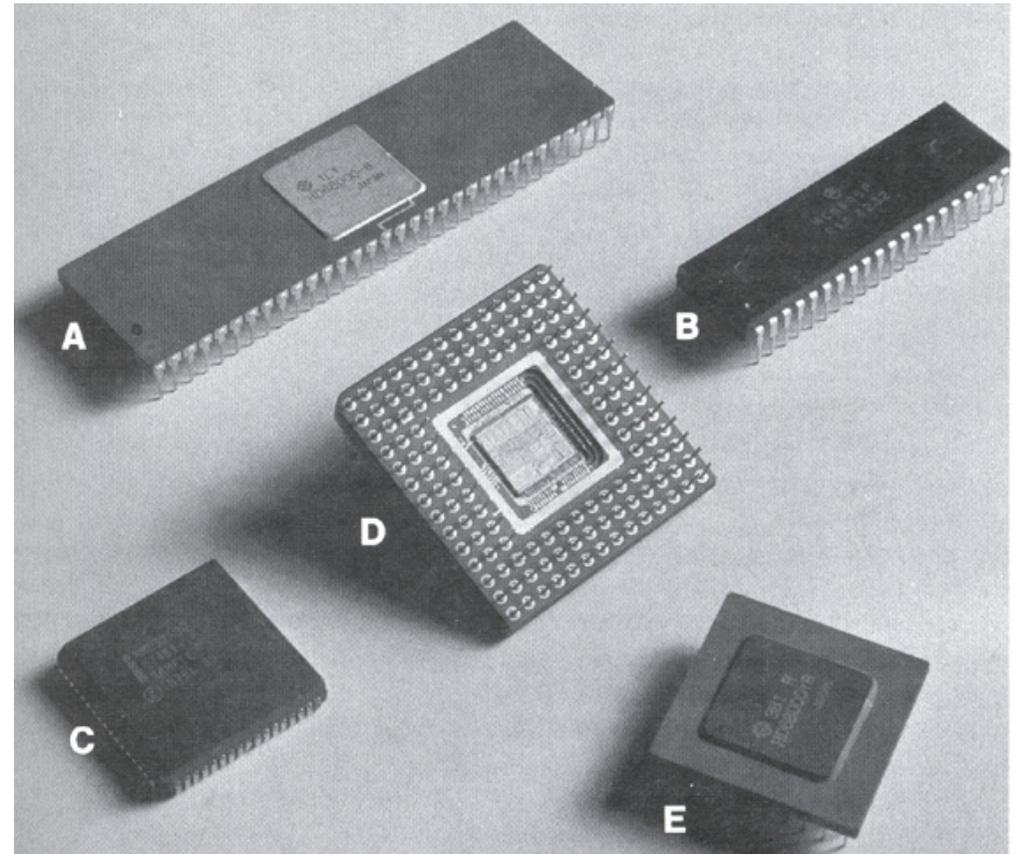
- Der Chip-Satz ist das Bindeglied zwischen den einzelnen Komponenten eines Computersystems.
- Die Chipsätze von verschiedenen Herstellern können teilweise Leistungsunterschiede von bis zu 10% haben.
- Der Chipsatz legt fest, welche Komponenten verwendet werden können:
 - Systembus
 - Speichertyp
 - Schnittstellen
 - Prozessortyp

HW eines Prozessors/Mikroprozessors

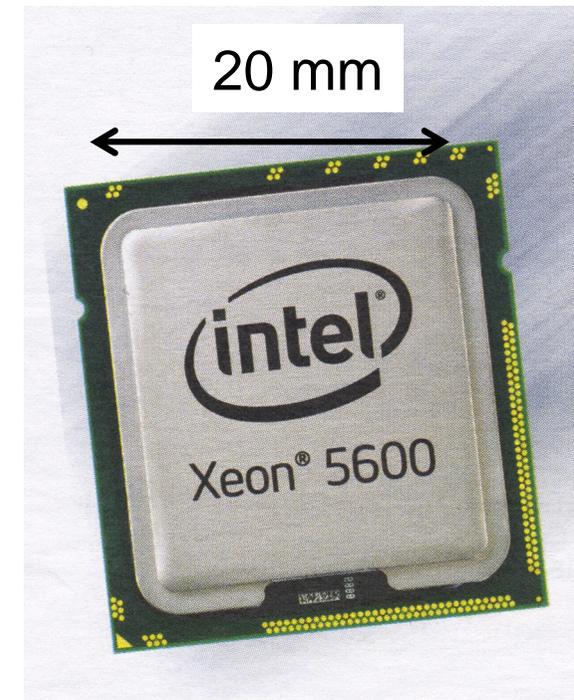
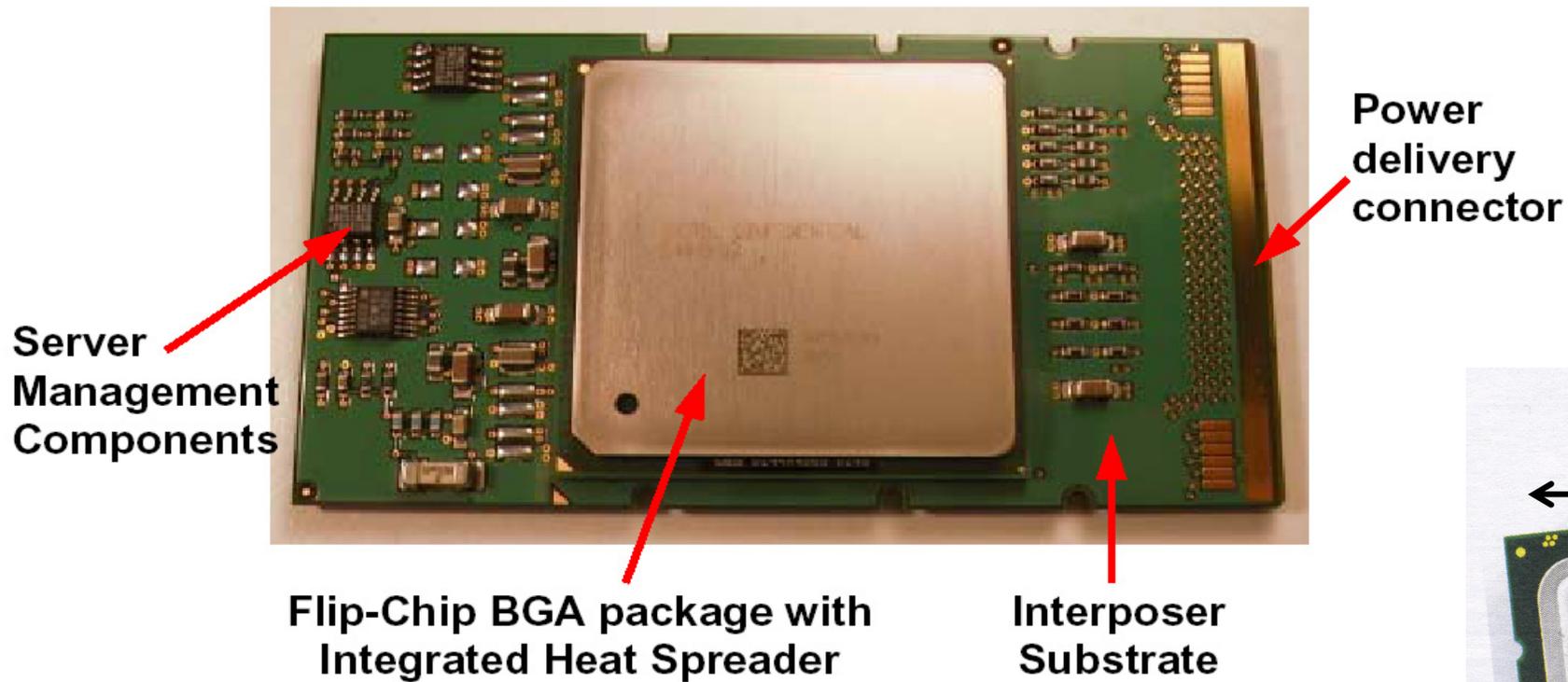
- Gehäuse und Anschlüsse
 - Ein Chip muß zur Erhöhung der mechanischen Stabilität, zur Herausführung der Anschlüsse sowie zur Ableitung der Wärme (z.B. bis 80 Watt) in ein Gehäuse untergebracht werden.

- Gehäusematerialien:
 - Plastik
 - Keramik

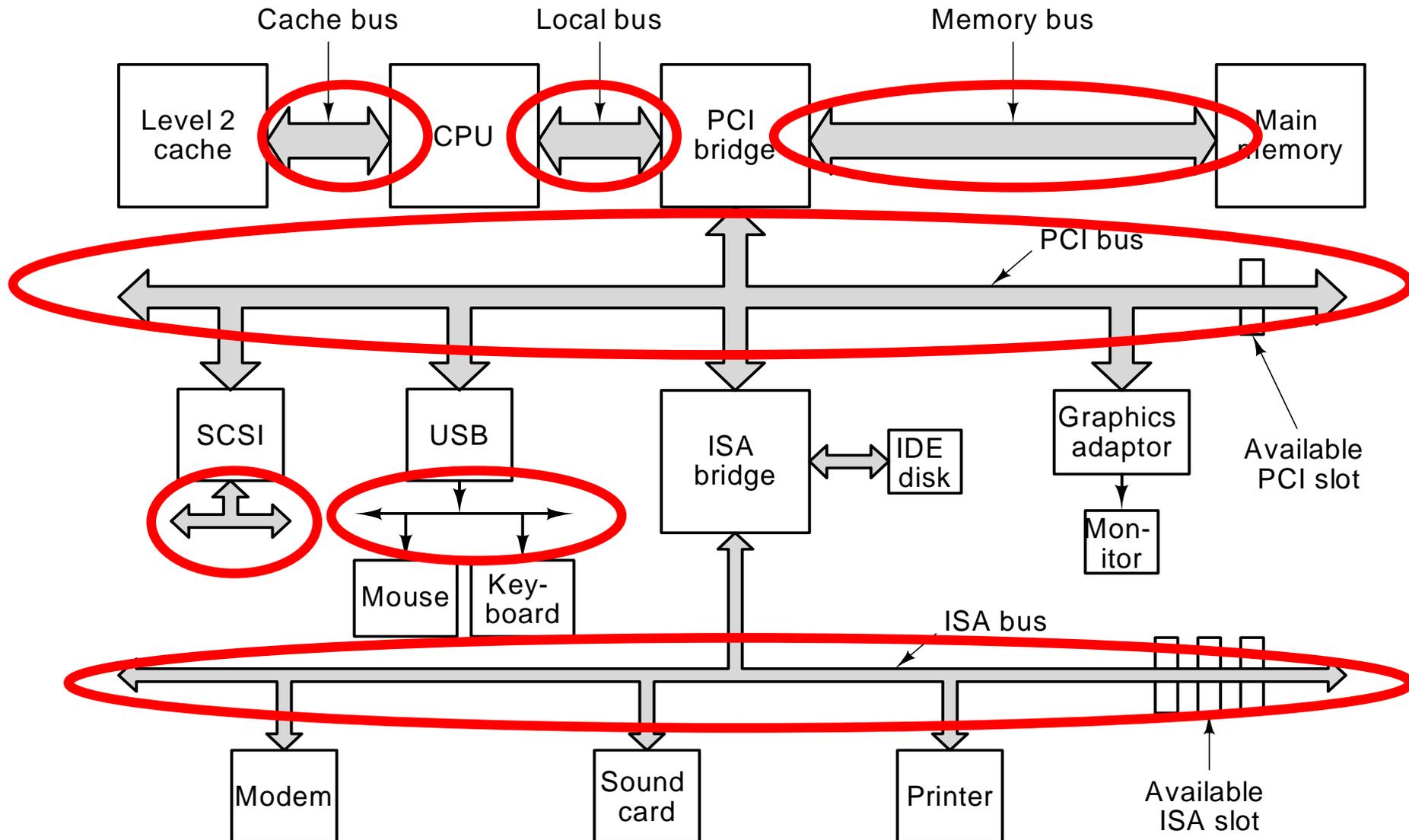
- Gebräuchlichste Gehäusetypen:
 - A bis E



Intel Itanium CPU Package und Xeon 5600 Serie



Die Großfamilie der Busse

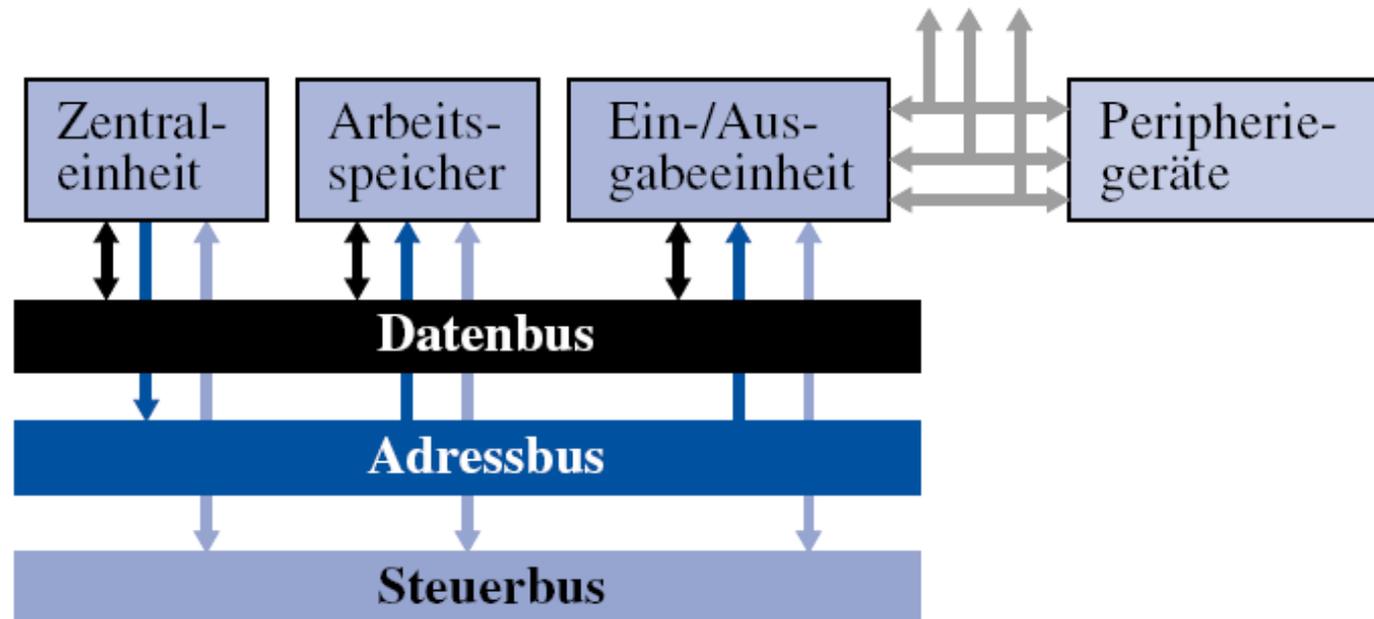


Busse und Schnittstellen

- ❑ Busse und Schnittstellen werden sowohl zur **Kommunikation zwischen den Bestandteilen des Mainboards** als auch zum Anschluß aller Arten von Peripheriegeräten benötigt, wie z.B. für Grafikkarten, Festplatten, Drucker.
- ❑ Dem **Transport von Daten** zwischen den Einheiten auf dem Mainboard, dem Prozessor, dem Arbeitsspeicher und der Ein-/Ausgabeeinheit dient ein so genanntes internes *Bussystem*.
- ❑ Aus Geschwindigkeitsgründen werden auf dem internen Bussystem **mehrere Bits parallel übertragen**. Die Anzahl der parallel zu übertragenden Bits hängt von der an den HW-Chips verfügbaren Busleitungen ab und korreliert sinnvollerweise mit der Bitlänge der Prozessorregister und der darin zu verarbeitenden maximalen Datengrößen.

Bustypen

- **Datenbus:** Er dient der *bidirektionalen Übertragung von Daten zwischen den Einheiten*.
- **Adreßbus:** Er dient der *unidirektionalen Übermittlung von Adressen zum Speicher (oder zu den Ein-/Ausgabeeinheiten)*.
- **Steuerbus:** Er dient zur Koordination exklusiver Zugriffe auf den Daten- und Adressbus (Bus reservieren, freigeben, ...).



Anschlüsse für Erweiterungskarten

- Es gibt vor allem im PC-Bereich verschiedene Arten von Schnittstellen zum Anschluss von Ergänzungs- bzw. Erweiterungskarten:
 - Der *PCI-Anschluss (Peripheral Component Interface)* ist der Standard-Kartenanschluss für PCs (32-Bit, 33 MHz).
 - Der *AGP-Anschluss (Accelerated Graphics Port)* ist ein spezieller Anschluss für Grafikkarten (64-Bit, \geq 66 MHz).
 - Der *PCMCIA-Anschluss (Personal Computer Memory Card International Association)* wird auch als *PC-Card-Anschluss* und häufig bei Notebooks als externer Anschluss für spezielle kleine Einsteckkarten eingesetzt.

Anschlüsse für Laufwerke

- Für Festplatten, CD-ROM-Laufwerke und andere Massenspeicher gibt es zwei wichtige Arten von Schnittstellen.
 - *EIDE (Enhanced Integrated Device Electronics)* ist auf den meisten PC-Mainboards integriert. Mit jedem der beiden Anschlüsse können je zwei Geräte verbunden werden: ein *Master* und ein *Slave*.
 - *SCSI (Small Computer System Interface)*: Eine SCSI-Schnittstelle erlaubt den Anschluss von sieben Geräten bzw. *Wide-SCSI* läßt sogar den Anschluß von 15 Geräten zu.
- Nachfolgetechnologie:
 - *S-ATA (Serial-Advanced Technology Attachment)* ist eine Nachfolgetechnologie zu IDE / EIDE (auch P-ATA) mit schneller, serieller Datenübertragung.

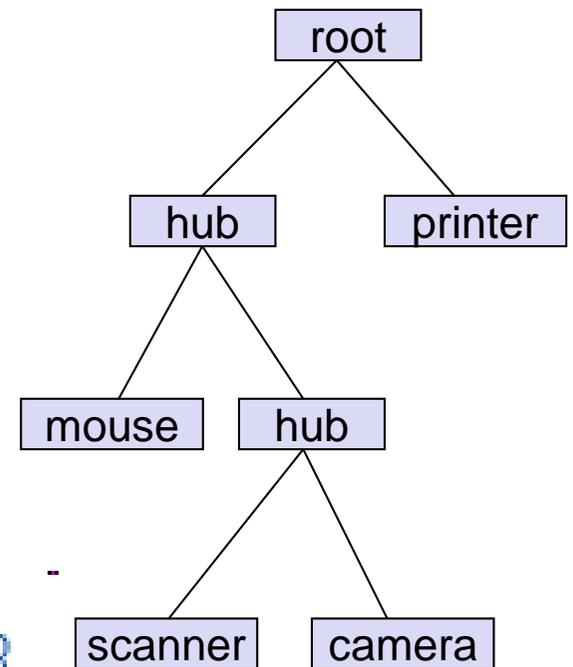
Anschlüsse für weitere Peripherie

- ❑ Heutige Rechner sind mit neuartigen, seriellen Schnittstellen für externe Geräte ausgestattet
- ❑ Diese unterstützen das so genannte *Hot-Plugging*-Verfahren:
 - Geräte können im laufenden Betrieb angeschlossen und wieder entfernt werden.
 - Der *USB-Anschluss (Universal Serial Bus)* arbeitet als *USB 1.0 und 1.1* mit einer Datenübertragungsrate von 12 Mbit/s und als *USB 2.0* mit einer Übertragungsrate von 480 Mbit/s. *USB 3.0* erreicht neuerdings eine Rate von bis zu 5 Gbit/s.
 - Die *IEEE-1394-Schnittstelle* (auch *FireWire* genannt) arbeitet mit einer Übertragungsrate von 400 Mbit/s bzw. das neuere *FireWire 800* mit einer Übertragungsrate von 800 Mbit/s.

USB (Universal Serial Bus – Basic Idea)

- ❑ Connection of low-bandwidth peripherals
 - Mouse, keyboard, scanner, printer, camera, PDA, ...
- ❑ Plug-and-Play
 - Simple connection, every time possible, no configuration
- ❑ Services
 - 12 Mbit/s shared bandwidth (USB2.0: 480 Mbit/s
USB 3.0: 4,8 Gbit/s)
 - Isochronous and bulk data transfer
- ❑ System
 - Tree topology
 - Root hub at root polls devices (leaves)
 - 4 wires/cable: 2 power supply, 2 data

- ❑ See <http://www.usb.org> for more info.



Modul 5: Rechnerarchitekturen & -organisation

- ❑ von-Neumann Architektur
- ❑ Aufbau und Funktionsweise
- ❑ Organisation
- ❑ **Peripherie**
- ❑ Technologieentwicklung

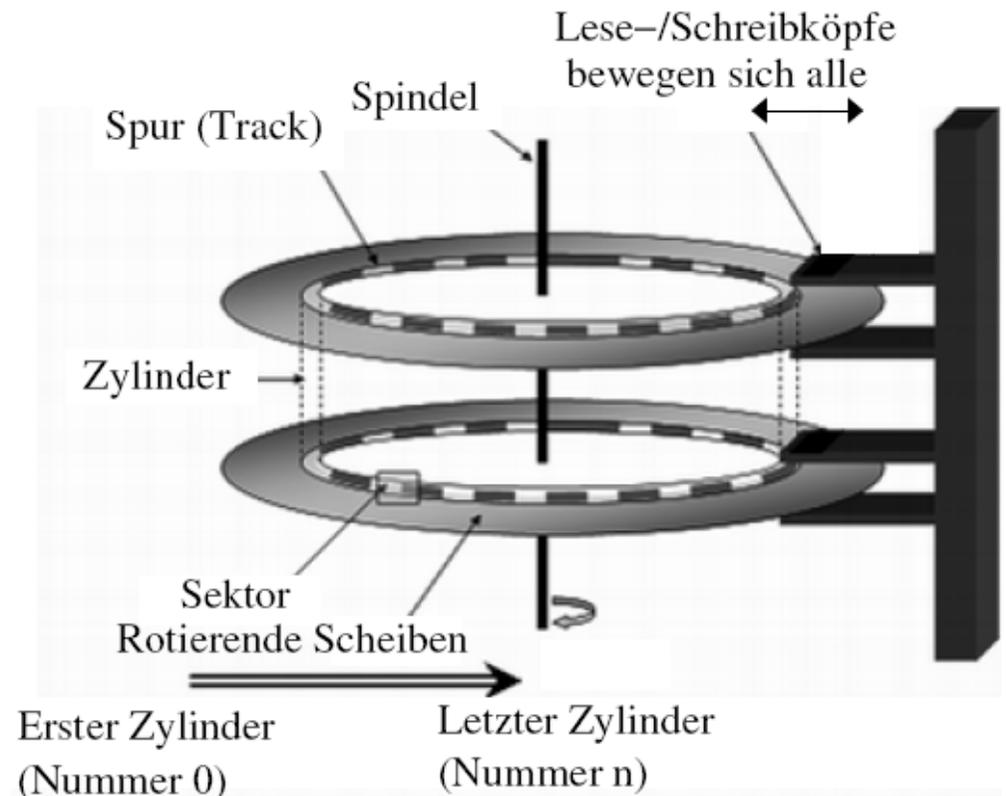
Peripherie

- E/A-/Speichergeräte, die an einen Rechner angeschlossen sind.
 - Es existieren viel zu viele Peripherieelemente und –systeme, als daß auch nur in einem annähernd sinnvollen Ansatz ein Überblick gegeben werden könnte.
 - **HD**, SSD, **CD**, **DVD**, scanner, laser printer, CRT, LCD, audio, mouse, trackball, touchpad, keyboard, hard disk, camera, 3-D-glasses, 3-D-printer, force feedback joystick, ink printer, robot, card reader, PC-card, **modem**, LAN adapter, WLAN adapter, GSM adapter, fax, sensors, actors, ...

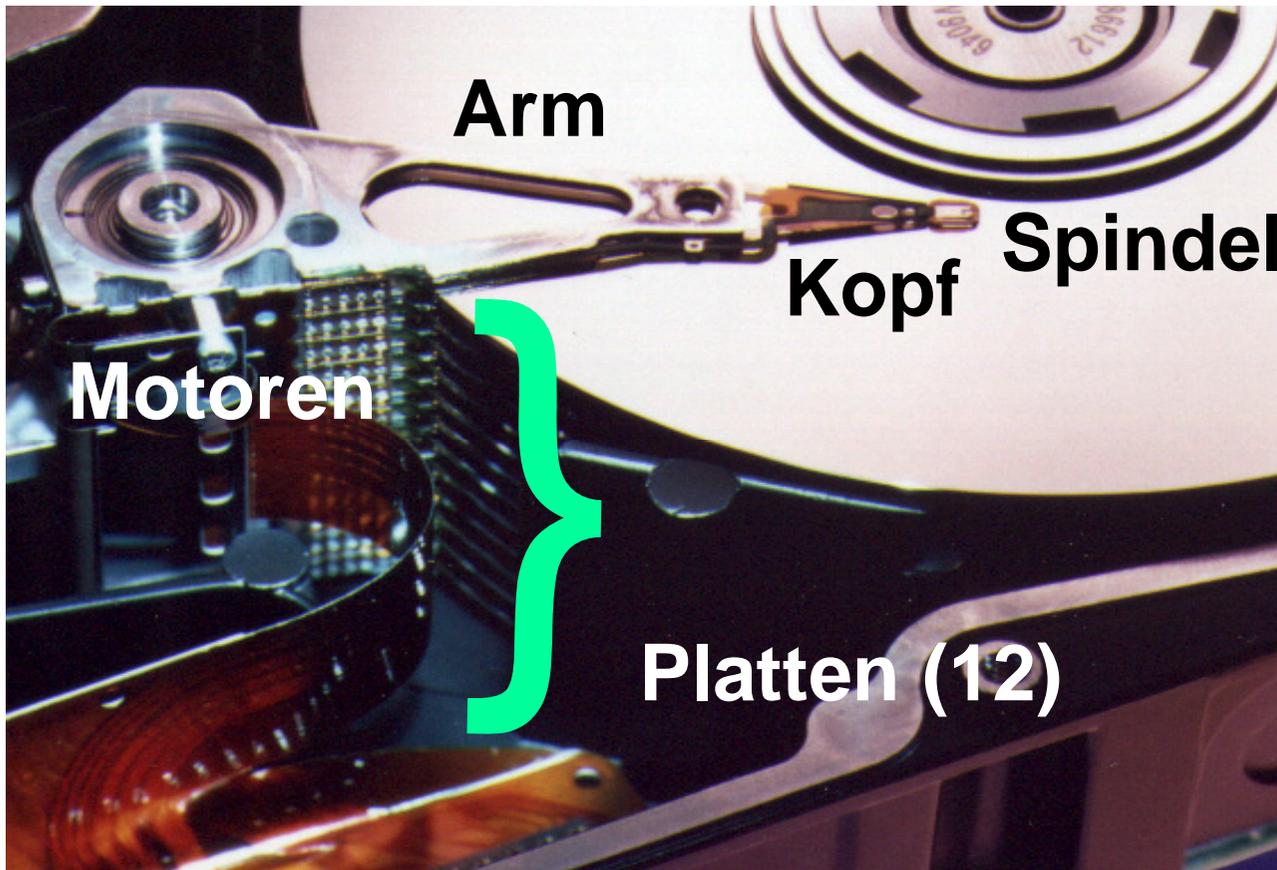
Geräte	Eingabe	Ausgabe	Massenspeicher
Tastatur, Maus, Scanner, Digitalkamera	X		
CD-ROM-Laufwerk	X		X
Grafikkarte, Bildschirm, Drucker, Lautsprecher		X	
Netzwerkkarte, Modem, Soundkarte	X	X	
Festplatte, USB-Stick, CD-Brenner	X	X	X

Festplatte (Harddisk, HD) (1)

- Verschiedene Speicherplatten:
 - Information ist magnetisch auf beiden Seiten der Platten gespeichert.
 - Bits sind in Spuren, die wiederum in Sektoren unterteilt sind (z.B. 512 Byte), gespeichert.
 - Motoren bewegen die Köpfe (≥ 1 /Plattenseite) über die Spuren einer selektierten Oberfläche. Warten bis der zu lesende bzw. zu schreibende Sektor unter dem Kopf vorbeikommt.
 - Zylinder: Alle Spuren unter den Köpfen



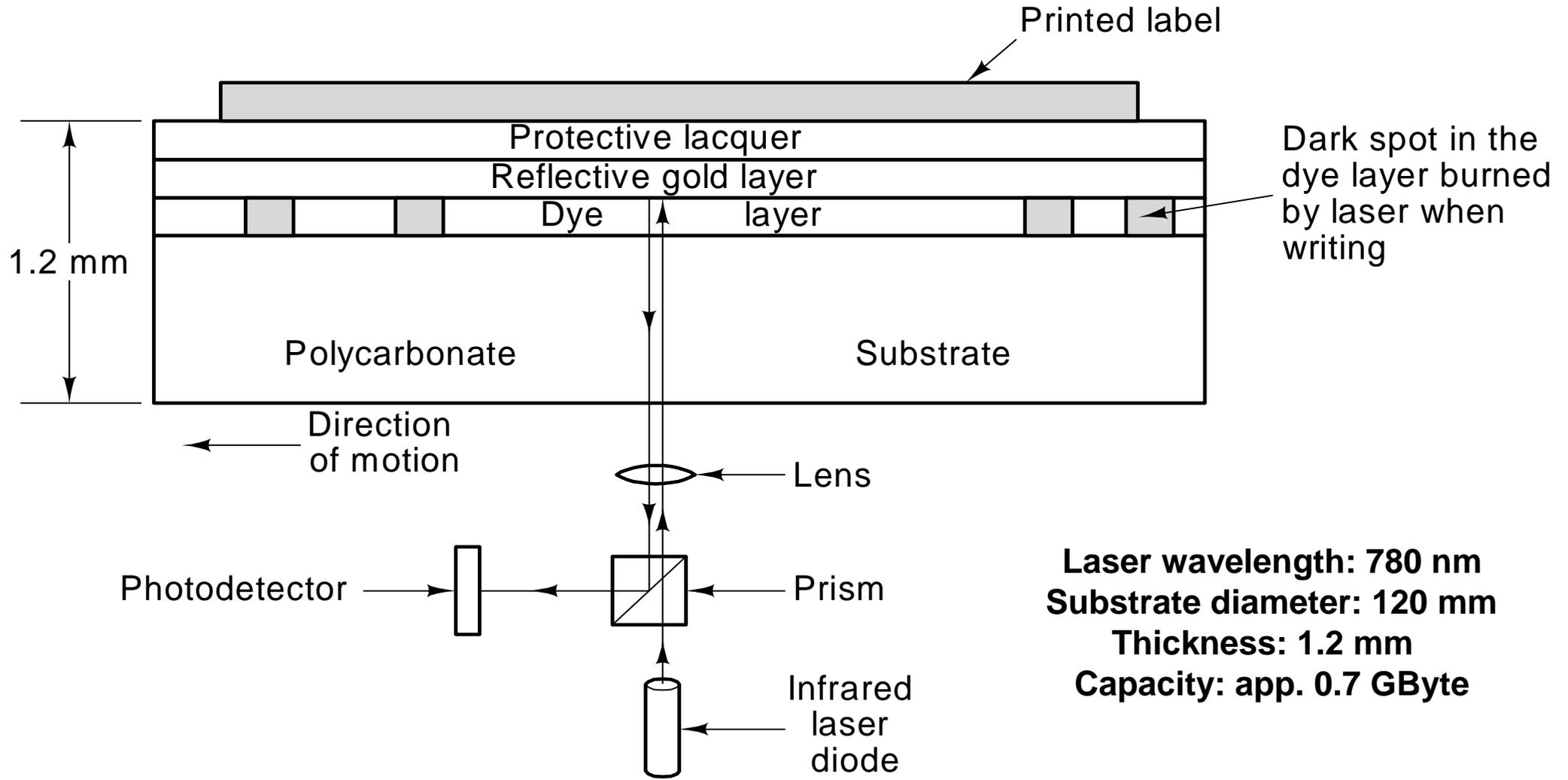
Festplatte (Harddisk, HD) (2)



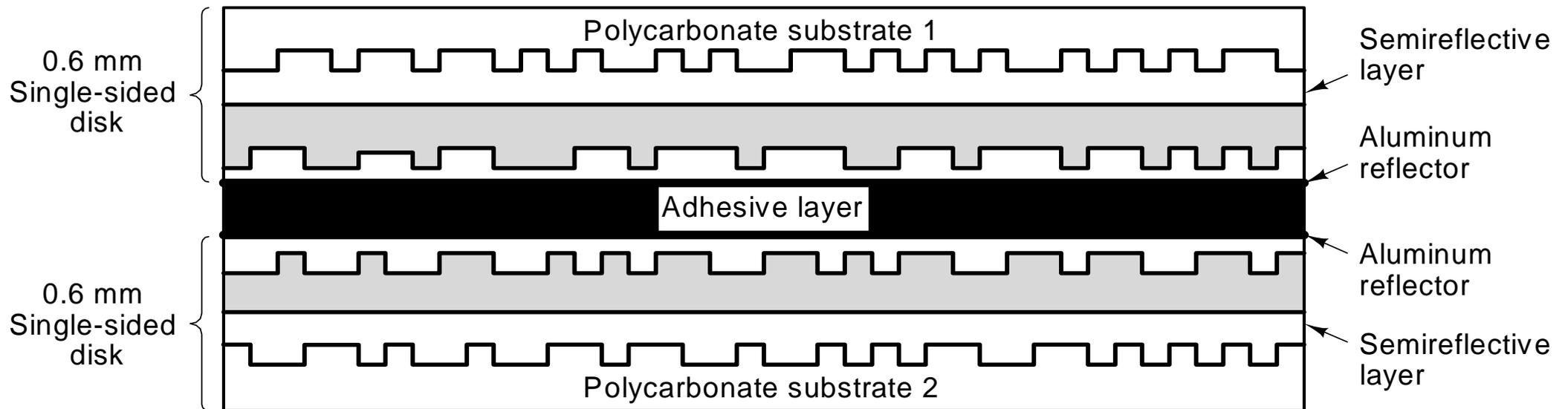
Optische Disks

- ❑ Die *CD* ist ein optischer Datenträger und wurde 1982 von Sony und Philips zur Ablösung der Schallplatte vorgestellt.
- ❑ Die Oberfläche einer CD besteht aus einer sehr dünnen Metallschicht, die von einem Laserstrahl abgetastet wird. Vertiefungen in dieser Oberfläche (*Pits*) wechseln sich mit der normalen, unversehrten Fläche (*Land*) ab und bilden so das Bitmuster, das von CD-Spielern als Daten, Musik oder Video interpretiert wird.
- ❑ Musik, Speicherung von Daten im Computerbereich; damit existieren mehrere Datenformate für die CD-ROM.
- ❑ Die DVD (Digital Versatile Disc) besitzt mit 4,7 GByte eine nochmals erheblich höhere Speicherdichte. Es gibt drei zueinander inkompatible Formate.

Compact Disk (CD)



DVD (Double-sided, Dual-layer)

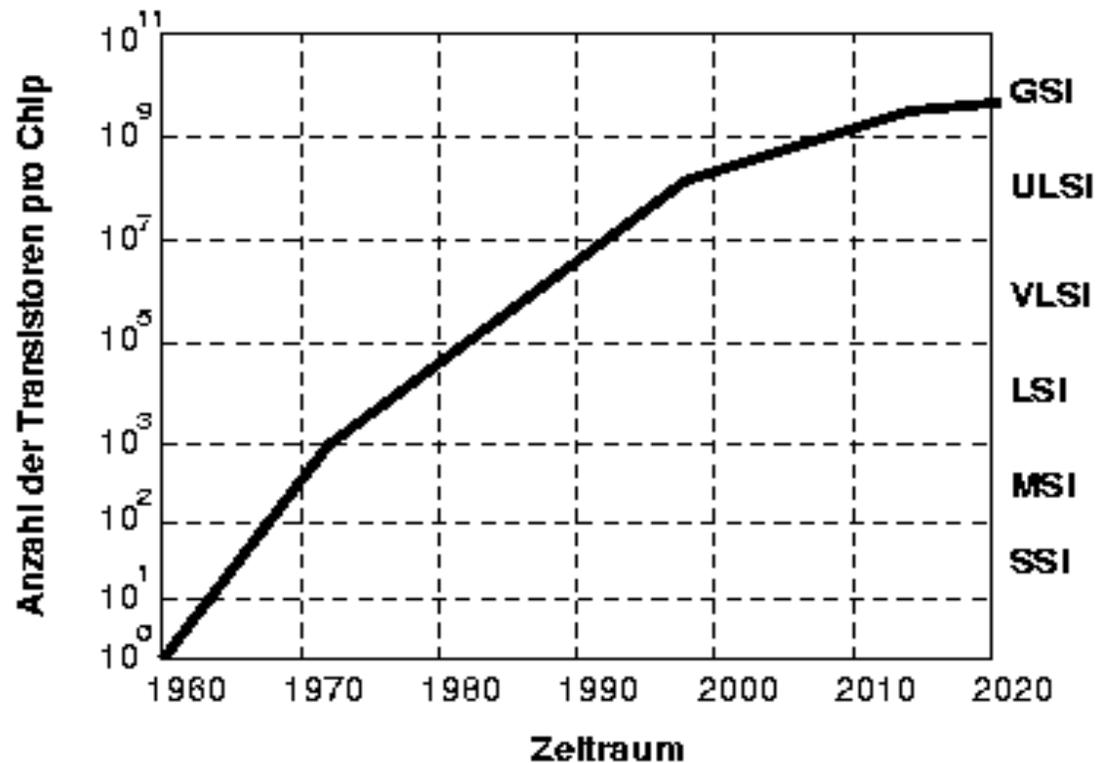


Laser wavelength: 635 or 650 nm
Substrate diameter: 120 mm
Thickness: 1.2 mm
Capacity: 4.7, 8.54, 9.4, or 17 GByte

Modul 5: Rechnerarchitekturen & -organisation

- ❑ von-Neumann Architektur
- ❑ Aufbau und Funktionsweise
- ❑ Organisation
- ❑ Peripherie
- ❑ Technologieentwicklung

Technologieentwicklung



SSI: Small Scale Integration

MSI: Medium Scale Integration

LSI: Large Scale Integration

VLSI: Very Large Scale Integration

ULSI: Ultra Large Scale Integration

GSI: Giga Scale Integration

Leistungssteigerung in Rechnersystemen

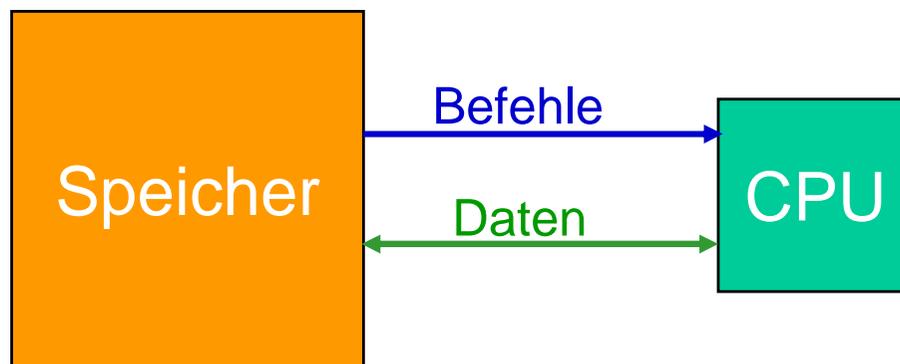
- Welche Möglichkeiten gibt es prinzipiell zur Leistungssteigerung in Rechnersystemen?

- Strukturelle Maßnahmen:
 - z.B. Zahl der Transistoren erhöhen
 - ➔ Parallelarbeit
 - Abfolge der Arbeitsschritte „verweben“
 - ➔ Pipelining

- Technologische Maßnahmen:
 - Anwendung schnellerer Technologien
 - ➔ Re-design ist nötig.
 - Alternative Ansätze
 - ➔ Vollständig neue Modelle nötig.

Strukturelle Maßnahmen (1)

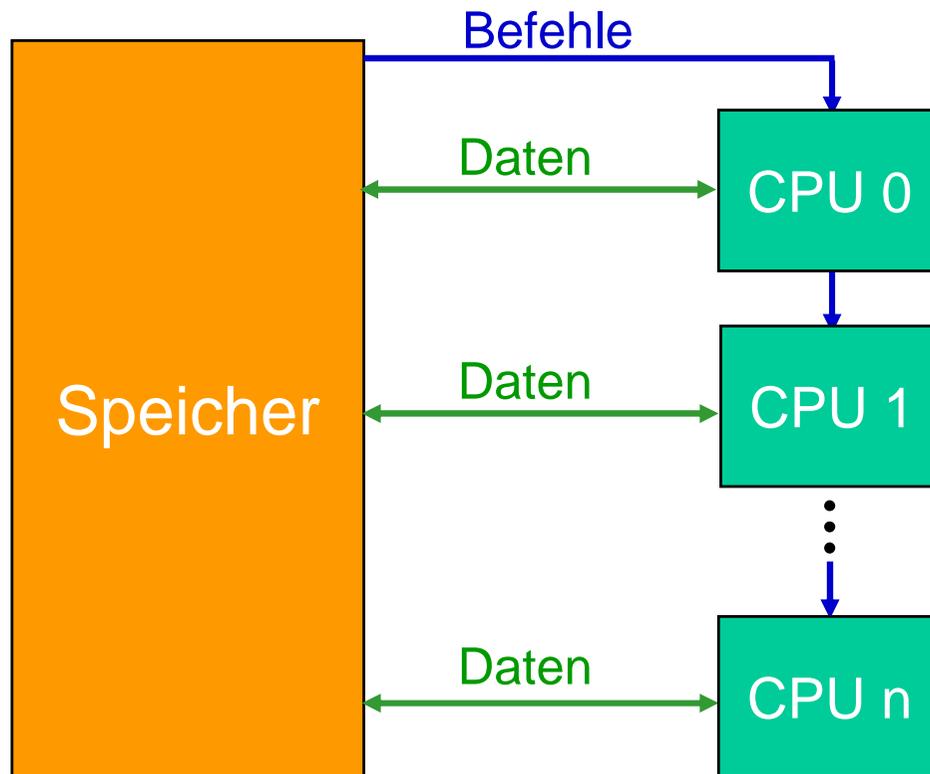
- Unterscheidung bezüglich der gleichzeitig bearbeiteten Befehls- und Datenströme:
 - SISD (Single Instruction Single Data)
Ein Datenstrom wird entsprechend einer seriellen Befehlsfolge verarbeitet (von-Neumann-Rechner)



Beispiele:
IBM-PC, IBM 370,
Micro-VAX von DEC

Strukturelle Maßnahmen (2)

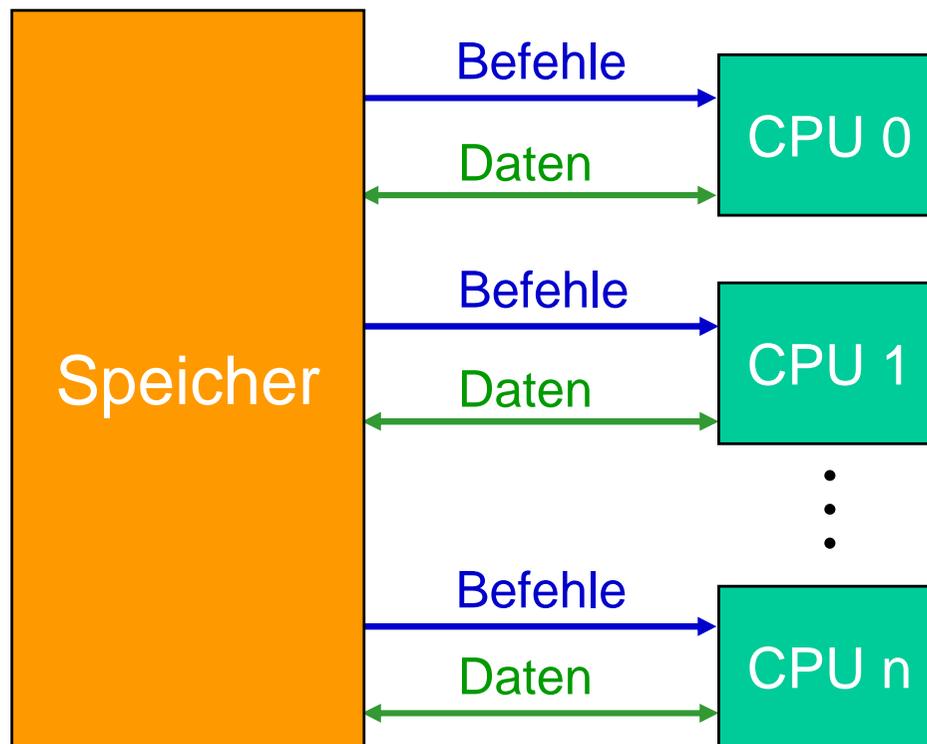
- SIMD (Single Instruction Multiple Data)
Alle Prozessoren führen gleichzeitig dieselben Befehle auf verschiedenen Daten aus (Array-Prozessoren)



Beispiel Bildverarbeitung:
Jedem Prozessor wird ein
Bildausschnitt zugeordnet.

Strukturelle Maßnahmen (3)

- MIMD (Multiple Instruction Multiple Data)
Alle Prozessoren führen gleichzeitig verschiedene Befehle auf verschiedenen Daten aus



Beispiele
Multiprozessor-
Systeme:

IBM 3084, Cray-2,
Multiprozessor PCs

Strukturelle Maßnahmen (4)

- MISD (Multiple Instruction Single Data)

Es wird nur ein Datenstrom bearbeitet. Bestimmte Ausführungseinheiten übernehmen die Ausführung bestimmter Teile einer Operation (Pipeline-Verarbeitung), was zu einer Parallelität auf Befehlsebene führt.

- Moderne“ Prozessoren: ab Intel 80286
- Bei vielen Autoren bleibt diese Klasse leer – darüber lässt sich diskutieren!

- Weitere Einteilung:

- **Mehrprozessorsysteme:**

- Mehrere Prozessoren mit unabhängigen Programmen arbeiten mit einem gemeinsamen Hauptspeicher

- **Feldrechner:**

- Mehrere Prozessoren arbeiten am gleichen Programm, aber mit verschiedenen Daten (Bsp: Bildverarbeitung)

- System mit **funktionsspezialisierten** Prozessoren:

- Mehrere Spezialprozessoren arbeiten unter einer CPU und mit einem Hauptspeicher

- **Fließbandverarbeitung (Pipeline-Struktur):**

- In einer Kette von Prozessoren übernimmt jeder die Ausführung bestimmter Teile einer Operation.