

6. Rechnerarchitektur

- Aufbau von Digitalrechnern

- Von-Neumann-Rechnerarchitektur
 - Komponenten
 - Prinzipien

- Speicher
 - Begriffe
 - Speicherhierarchie
 - Entwicklungstrends

- Betriebssystem
 - Betriebssystem und Prozesse
 - Unix (Dateisystem, Shell-Benutzung)

- Alternative Rechnerarchitekturen

Aufbau eines Digitalrechners

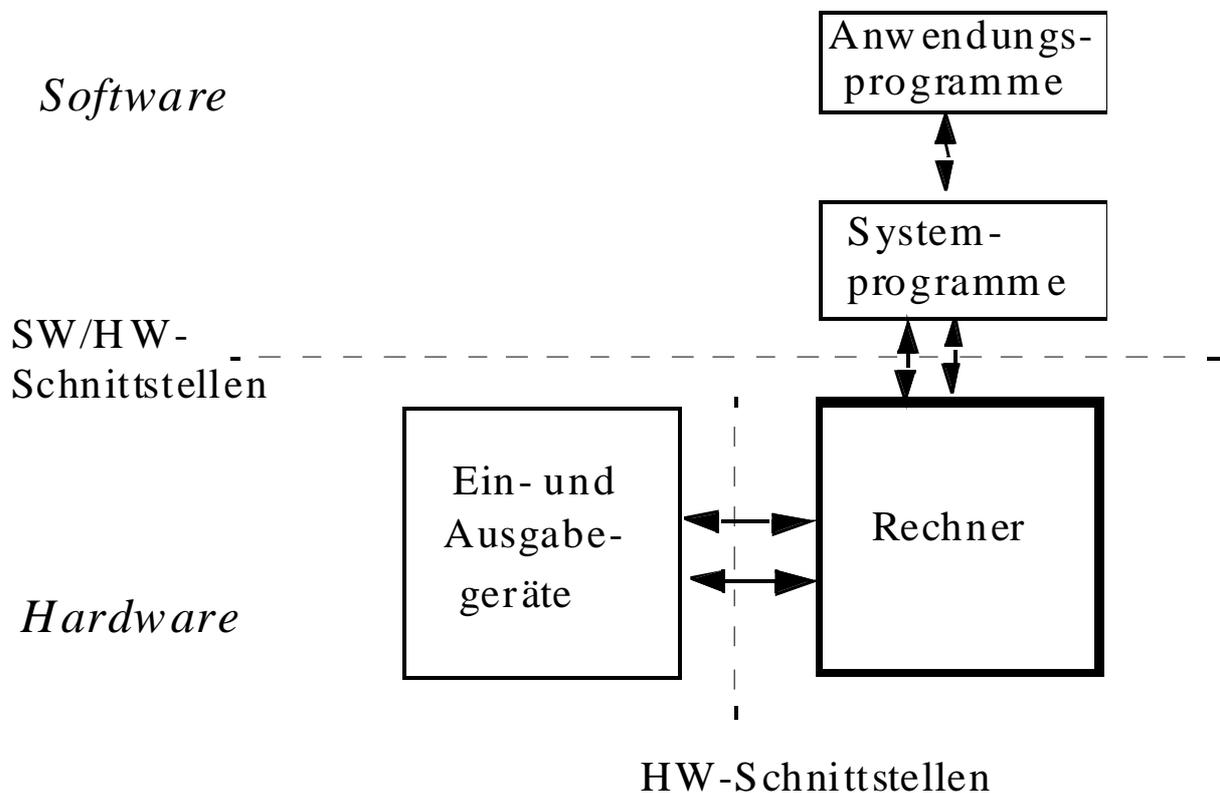
□ Digitalrechner (Computer)

universelles informationsverarbeitendes System, welches Folgen von Operationen (arithmetische, logische, u. a.) durch vorzugebende Programme gesteuert, selbsttätig (automatisch) ausführen kann.

Die Universalität besteht darin, daß im Rahmen der vom Rechner ausführbaren Operationen beliebige Algorithmen (Verfahren) in entsprechende Programme transformiert (übersetzt) werden können.

Es werden ausschließlich digitale Daten - in Dualdarstellung - verarbeitet.

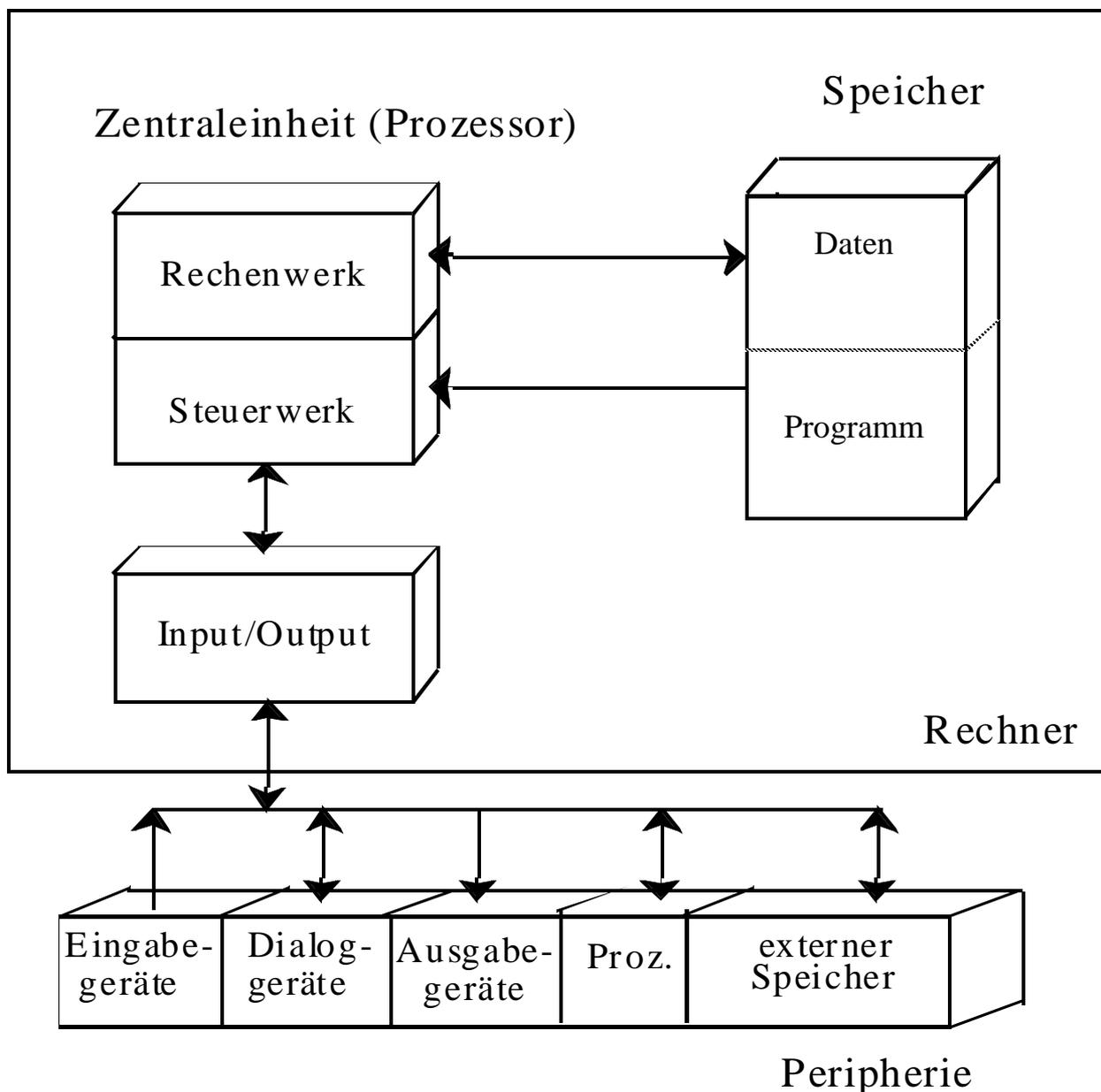
□ Ablaufumgebung



Von-Neumann-Prinzipien

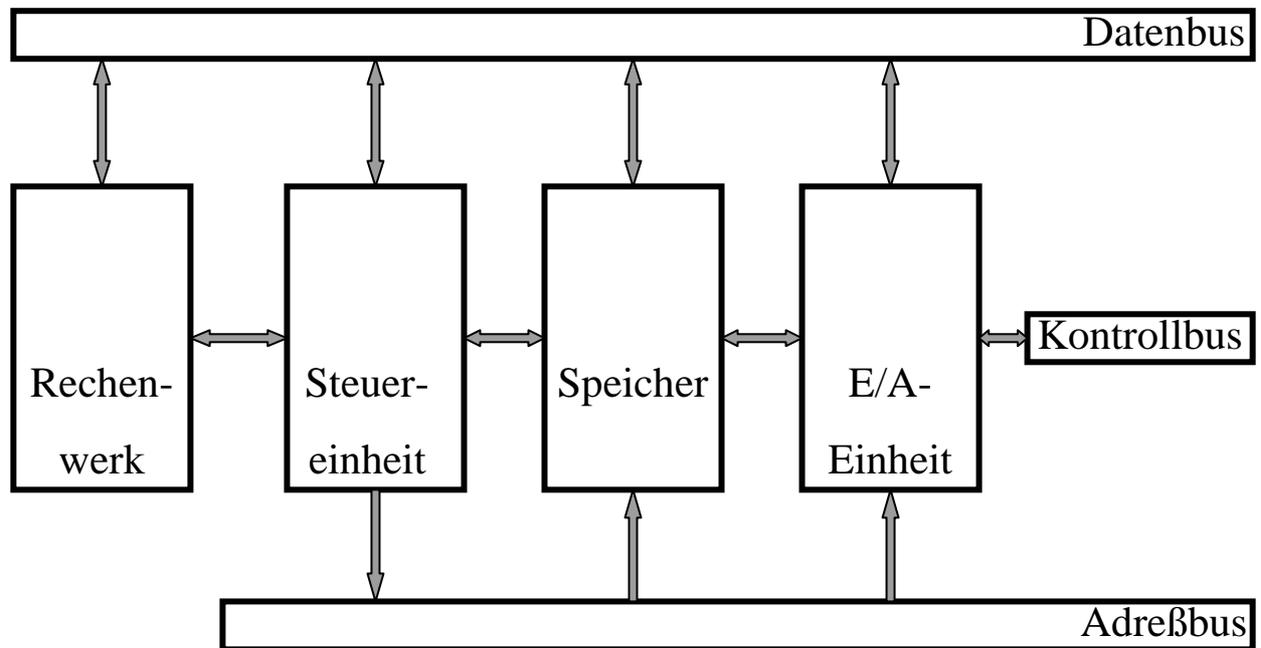
1. Der Rechner besteht aus fünf Funktionseinheiten (Steuerwerk, Rechenwerk, Speicher, Eingabewerk, Ausgabewerk).
2. Die Struktur des Von-Neumann-Rechners ist unabhängig von den zu bearbeitenden Problemen.
3. Programme, Daten, Zwischen- und Endergebnisse werden in demselben Speicher abgelegt.
4. Der Speicher ist in gleichgroße Zellen unterteilt, die fortlaufend durchnummeriert sind (Adressen).
5. Aufeinanderfolgende Befehle eines Programms werden in aufeinanderfolgenden Speicherzellen abgelegt.
6. Durch Sprungbefehle kann von der Bearbeitung der Befehle in der gespeicherten Reihenfolge abgewichen werden.
7. Es gibt zumindest
 - arithmetische Befehle
 - logische Befehle
 - Transportbefehle
 - sonstige Befehle wie Schieben, Unterbrechen, Warten usw.
8. Alle Daten (Befehle, Adressen usw) werden binär codiert.

Von-Neumann-Rechner

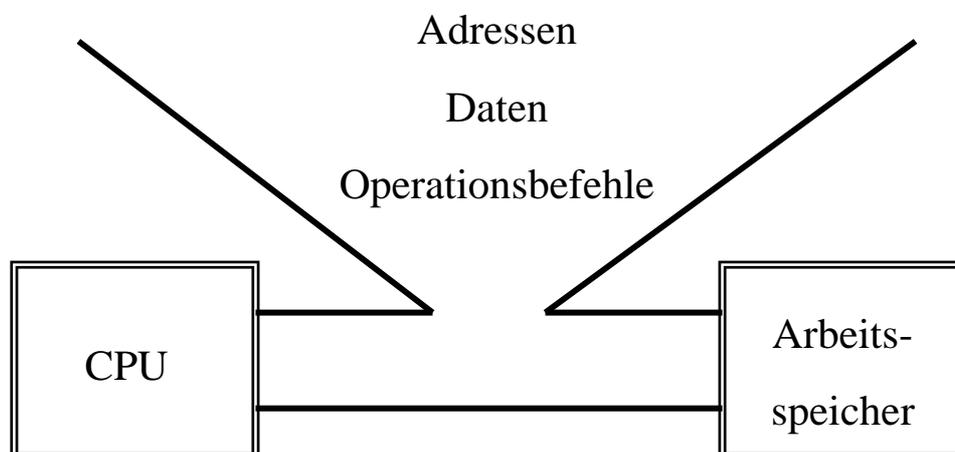


- ❑ Zentraleinheit = CPU (Central Processing Unit)
- ❑ Steuerwerk (Leitwerk): Laden, Dekodierung und Interpretation von Befehlen sowie Steuerung der an der Ausführung beteiligten Komponenten
- ❑ Rechenwerk (ALU, Arithmetic Logical Unit): Ausführung von arithmetischen und logischen Operationen (Verknüpfung von Operanden)

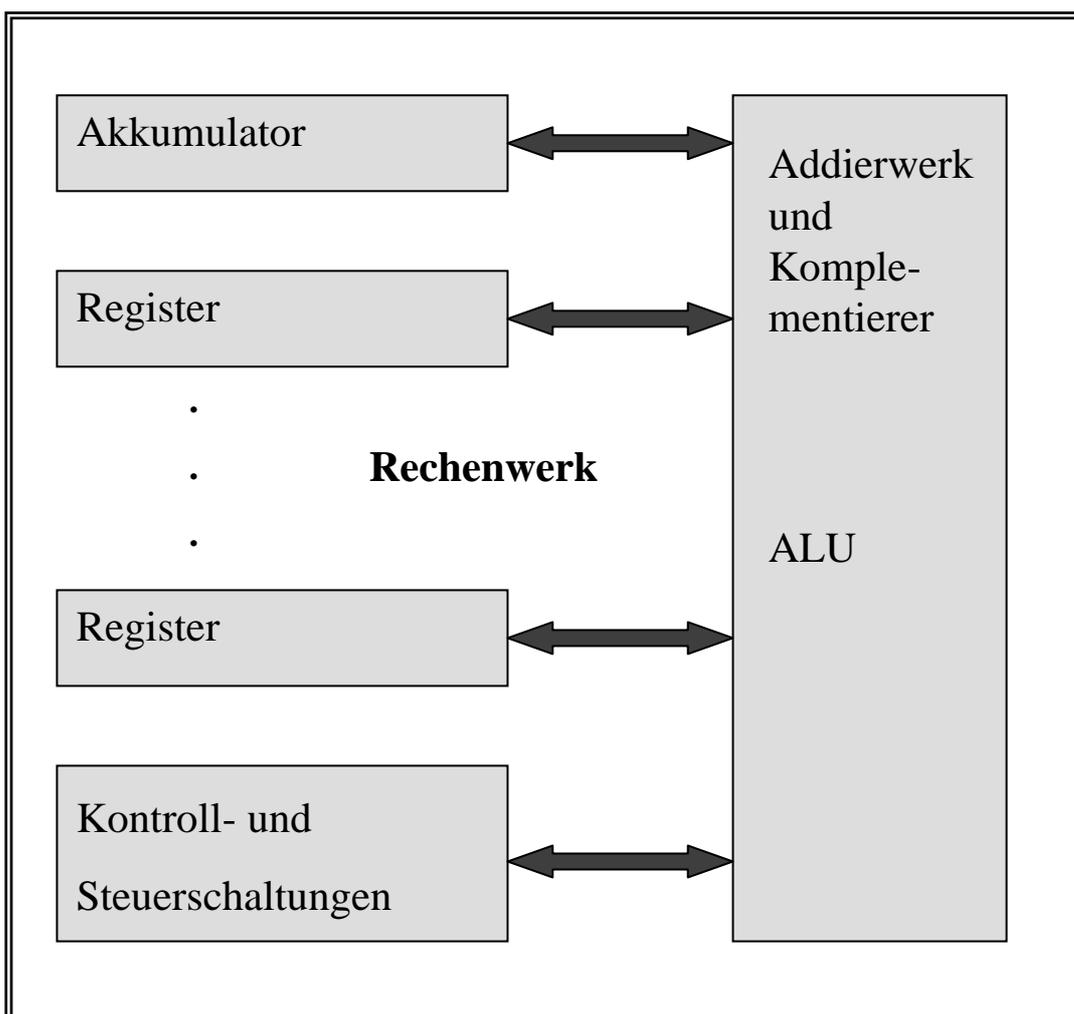
Busstruktur



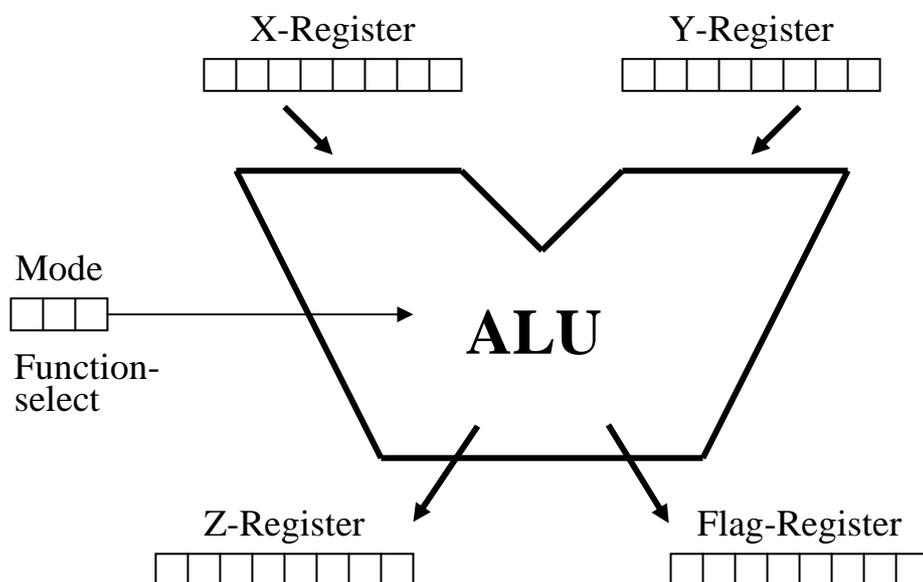
- Ein potentioneller Engpaß - Kommunikation zwischen CPU und Arbeitsspeicher:



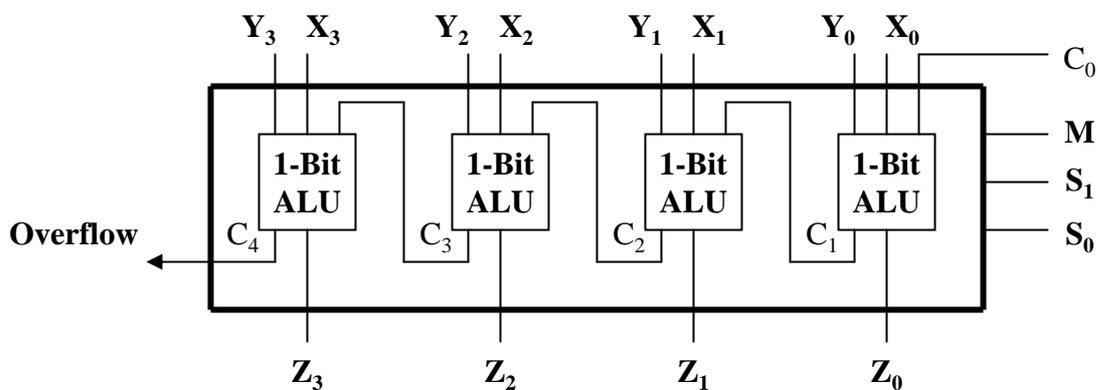
Aufbau des Rechenwerks



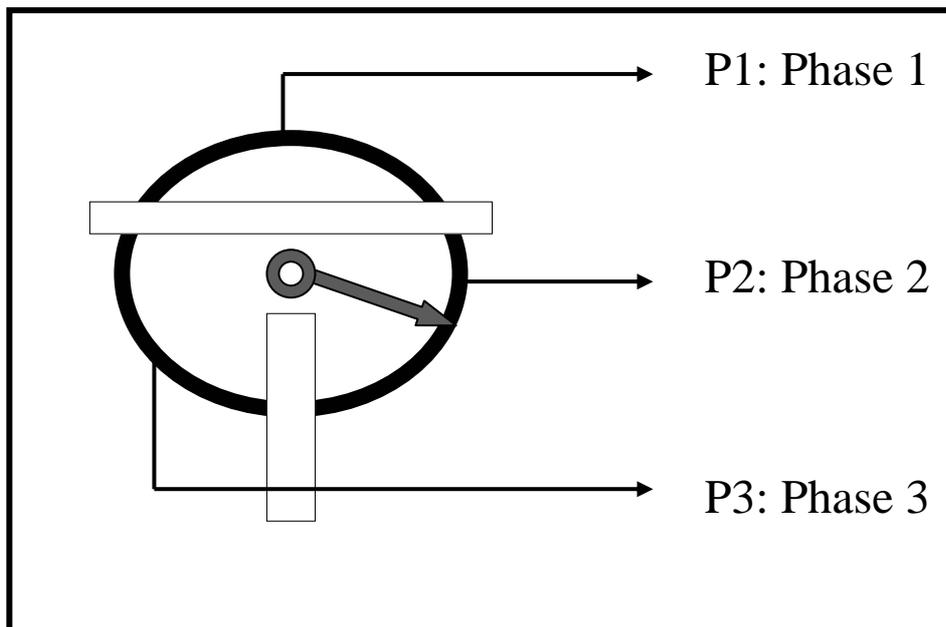
Arithmetisch-Logische Einheit



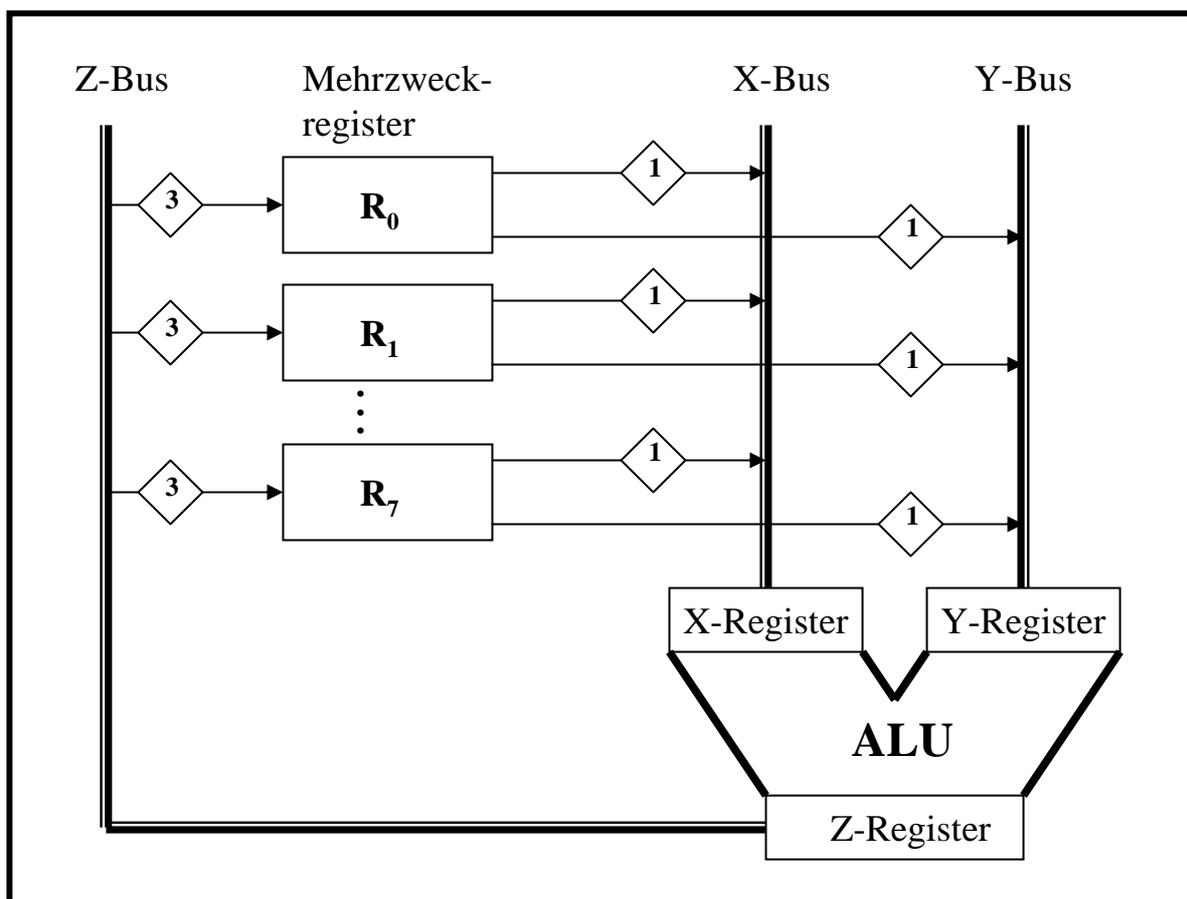
4-Bit-ALU



Taktgeber



Rechnerkern: ALU + Register + Busse



Vom Mikroprogramm zum Assemblercode

- Der **Mikrocode**: Beschreibt Ablauf der Operationen im Rechnerkern (Stellung der Schalter, Steuerleitungen). Dieser wird von der Mikroprogrammmeinheit des Steuerwerkes generiert.
- Das Mikroprogramm zur Ausführung der Befehle im ROM des Steuerwerkes gespeichert.
- **Maschinencode**: Nächsthöhere Ebene der Programmierung. Befehle und Adressen können direkt angesprochen werden. Jeder Befehl durch eine Zahl, seinen *OpCode*, repräsentiert.
- **Assemblercode**: Besser lesbare Form der Maschinsprache. Befehle durch Worte benannt.
- Befehle des Assemblercodes in aufeinanderfolgenden Zellen des RAM gespeichert. Änderung in der Reihenfolge der Abarbeitung durch Sprungbefehle.
- Einige Register der CPU erfüllen spezielle Funktionen (Akkumulator, Befehlszähler) und sind direkt ansprechbar.
- Demonstration am „Ein-Adreß-Rechner“.

Speicher I

- Daten und Programme im Speicher des Computers bereitgehalten.
- Programmbefehle im Hauptspeicher in aufeinanderfolgenden Speicherzellen abgelegt.
- Jedes Datum ist ein Speicherwort der Länge n Bits.
- Jedes Wort belegt eine durch eine Adresse ansprechbare Speicherstelle im Speicher.
- Aufbau eines Speicherwortes für einen Befehl:

Operationsteil	Adreßteil
----------------	-----------

Speicher II

- ❑ ausschließlich als Binärspeicher realisiert
 - kleinste Speichereinheit: 1 Bit
 - 1 Byte (B) = 8 Bit (erlaubt i.a. Speicherung von 1 Zeichen)
 - 1 Kilo-Byte (KB) = 2^{10} B = 1024 B, 1 Mega-Byte (MB) = 2^{20} B
 - 1 Giga-Byte (GB) = 2^{30} B; 1 Tera-Byte (TB) = 2^{40} B

- ❑ Speicherzelle (Adressierungseinheit) meist 1 B; mehrere Zellen können zu einem *Speicherwort* (Zugriffseinheit) zusammengefaßt werden

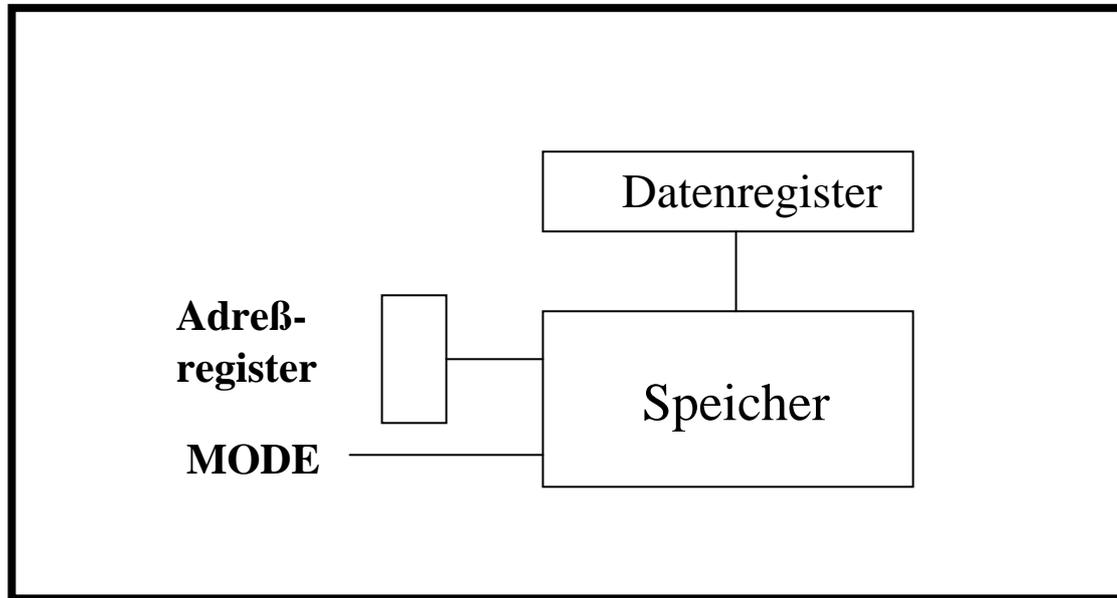
- ❑ Unterschiedliche Speichertypen
 - Hauptspeicher (Arbeits- bzw. Primärspeicher) vs. Hintergrundspeicher (Extern-bzw. Sekundärspeicher)
 - Halbleiter-, Magnet-, optische Speicher
 - flüchtige vs. nicht-flüchtige Speicher
 - direkte (wahlfreier) vs. sequentielle, lesende vs. schreibende Zugriffsmöglichkeit

RAM = Random Access Memory

ROM = Read Only Memory

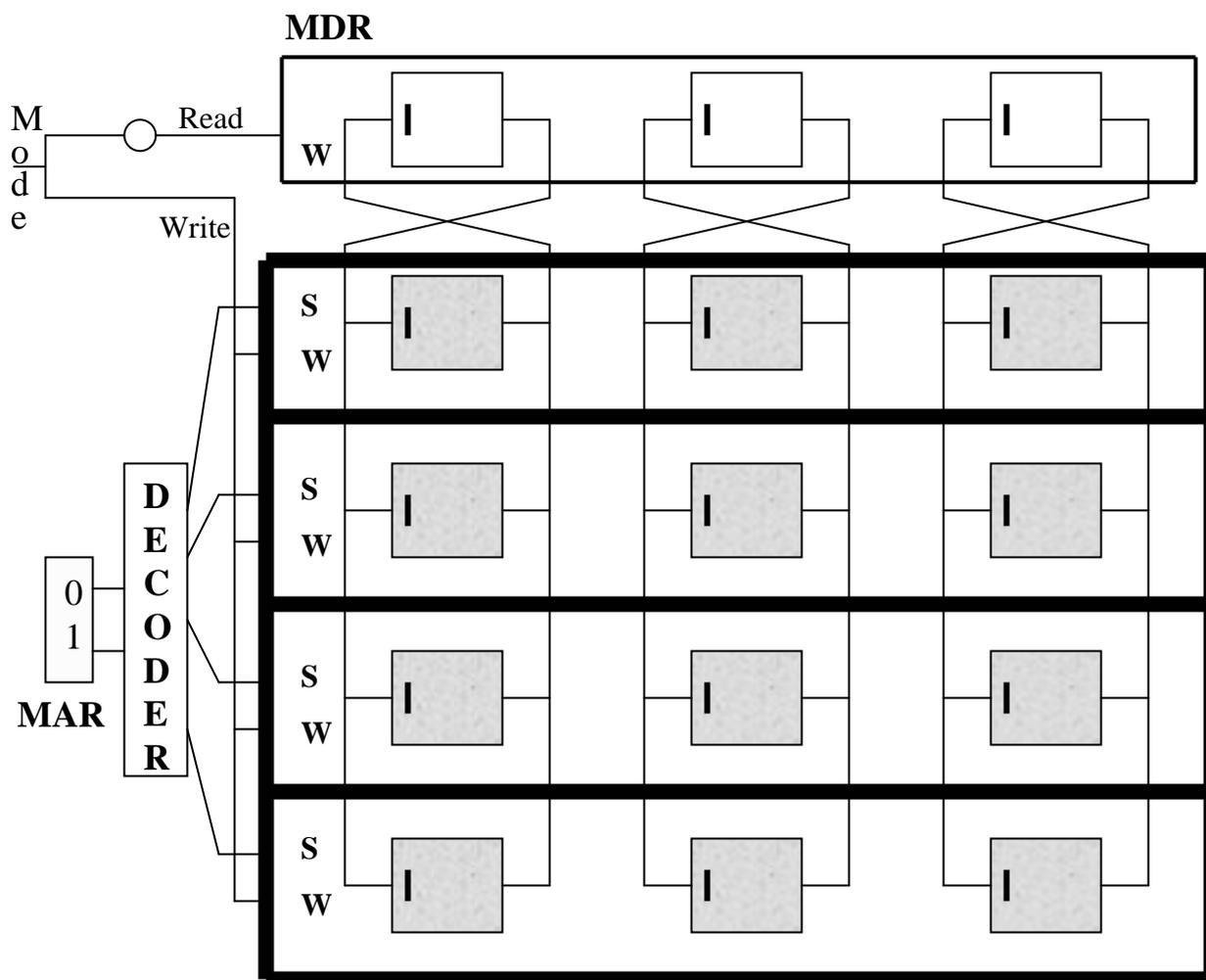
- ❑ Gegensätzliche Eigenschaften hinsichtlich Zugriffszeit, Kapazität und Kosten
=> Nutzung von Speicherhierarchien

Der Zugang zum Hauptspeicher



Blockschaltbild für den Speicher

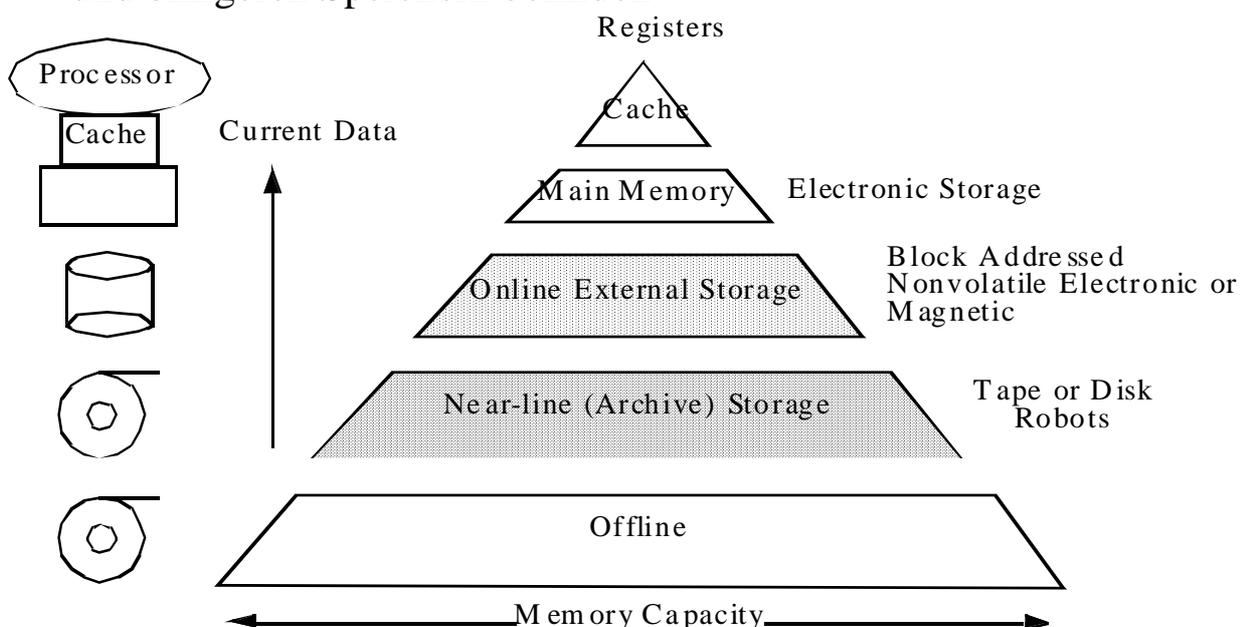
Speicher



Speicherhierarchie

□ Rekursives Prinzip

Kleinere, schnellere und teurere Cache-Speicher werden benutzt, um Daten zwischenzuspeichern, die sich in größeren, langsameren und billigeren Speichern befinden



□ Preis-Leistungs-Tradeoff (Stand 1992)

Schneller Speicher ist teuer und deshalb klein;
Speicher hoher Kapazität ist typischerweise langsamer

	Kapazität	Zugriffszeit	Zugriffseinheit	Flüchtigkeit	Instruktionsadressierbarkeit
Register	< 1 KB	5 ns	4 B	ja	ja
Cache	< 1 MB	20 ns	16 B	ja	ja
Hauptspeicher	< 1 GB	100 ns	32 B	ja	ja
Magnetplatte	> 1 GB	15.000.000 ns	4096 B	nein	nein

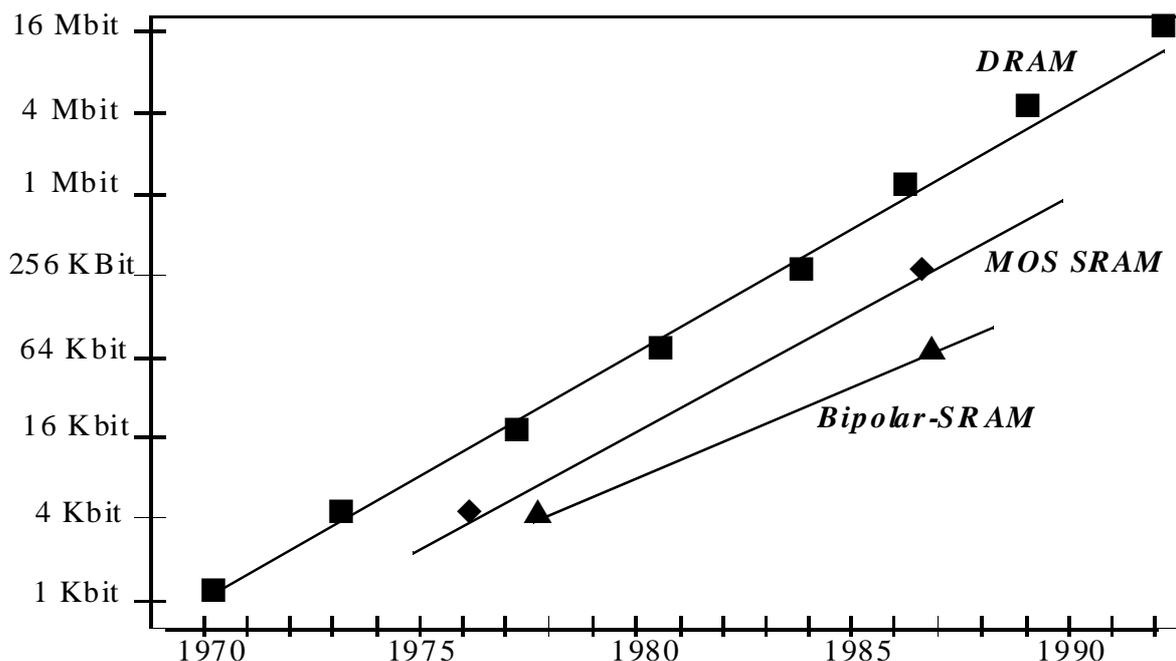
Entwicklungstrends: Elektronische Speicher

Die Speicherkapazität pro Chip erhöhte sich um den Faktor 4 alle 3 Jahre.

Moore's Law:

$$\text{MemoryChipCapacity (year)} = 4^{\frac{(\text{year} - 1970)}{3}} \text{ Kb/chip for year in [1970 ...2000]}$$

Chip-Kapazität

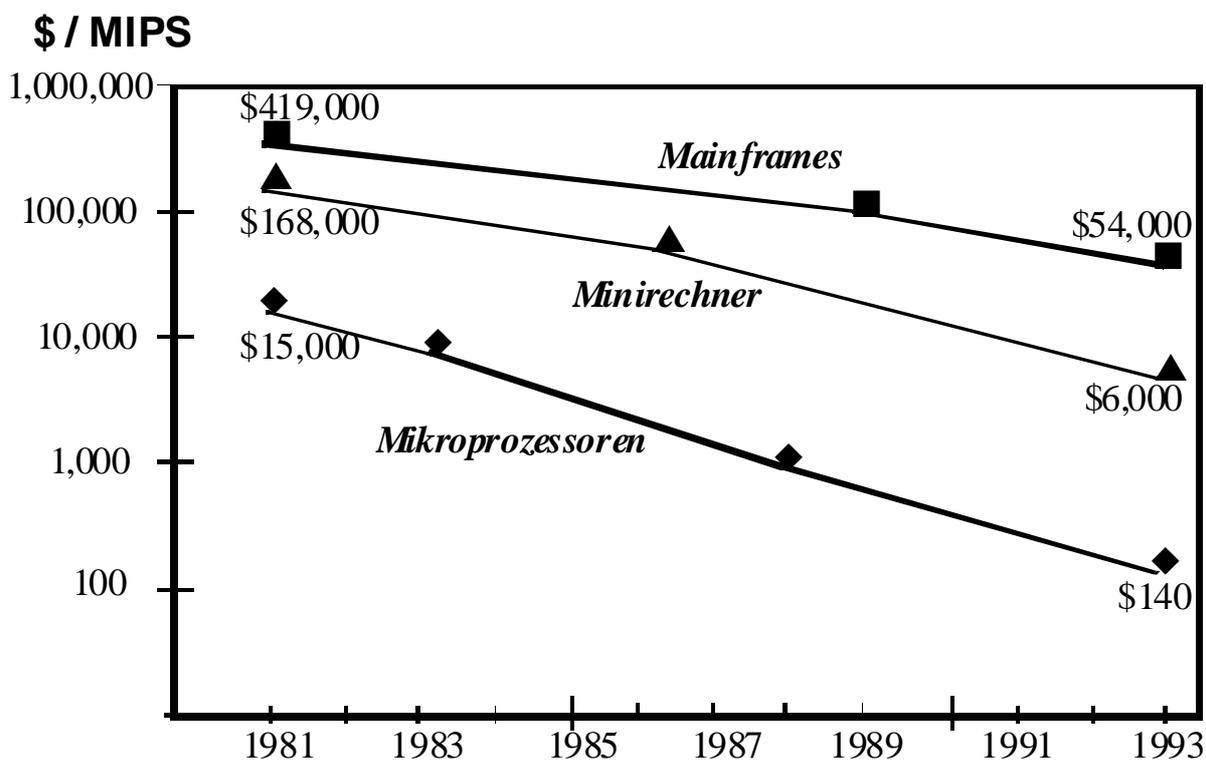
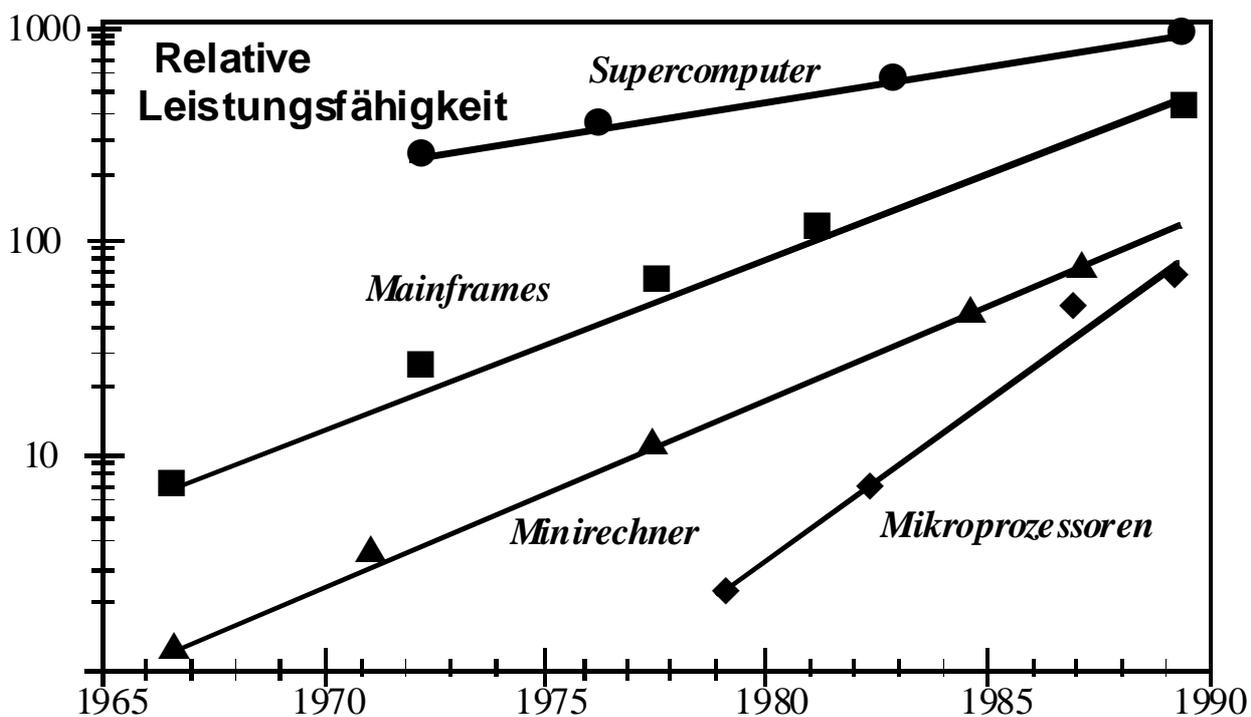


Entwicklung für Speicher und Prozessoren

Jahr	dyn. Speicher		Mikroprozessoren			Verdr.eb.	I/O p.Chip
	Größe [mm ²]	Mrd.Bits per Chip	Größe [mm ²]	Mio.Tr. per cm ²	Taktfr. [MHz]		
1995	190	0.064	250	4	300	4-5	900
1998	280	0.256	300	7	450	5	1350
2001	420	1	360	13	600	5-6	2000
2004	640	4	430	25	800	6	2600
2007	960	16	520	50	1000	6-7	3600
2010	1400	64	620	90	1100	7-8	4800

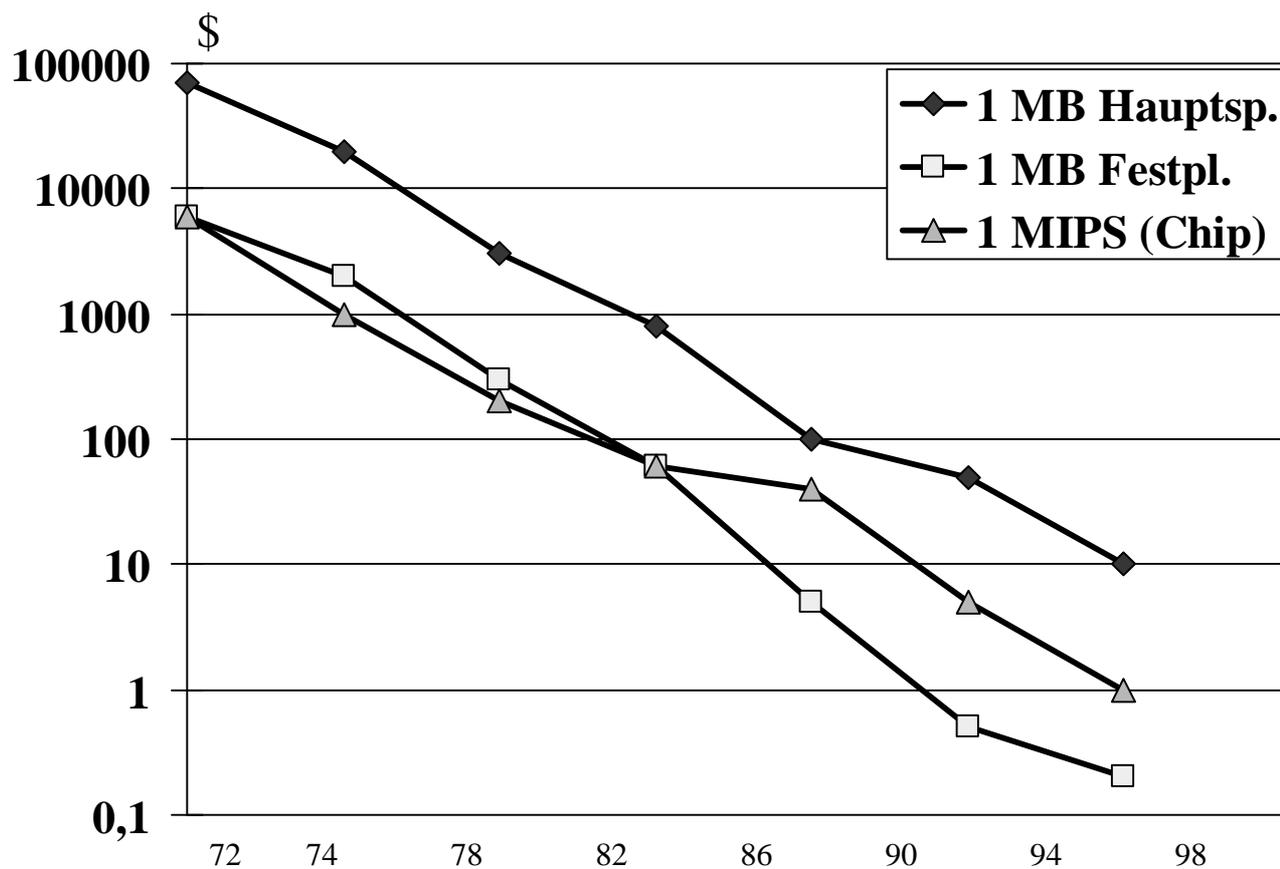
Entwicklungstrends: Prozessoren

Die Mips-Raten von Prozessoren stiegen zwischen 1965 und 1990 von etwa 0.6 Mips auf 40 Mips (Skalarleistung), was einem Faktor von 70 entspricht. Die Leistungsfähigkeit von Mikroprozessoren (Ein-Chip-Prozessoren) erhöhte sich seit 1980 wesentlich stärker; sie verdoppelte sich ungefähr alle 18 Monate.



Preisverfall durch Technologieentwicklung

Kosten reduzieren sich im Mittel um 43% pro Jahr



Aufgaben eines Betriebssystems

- ❑ Das Betriebssystem eines Rechners ist die zusammenfassende Bezeichnung für alle Programme, die
 - die **Ausführung der Benutzerprogramme** und
 - die **Verteilung der Betriebsmittel** auf die einzelnen Benutzerprogramme steuern und überwachen.

- ❑ Das Betriebssystem erlaubt dem Anwendungsprogrammierer, von der **Rechner-Hardware zu abstrahieren** (z.B. bei der Speicherabbildung oder beim Externspeicherzugriff). Es stellt jedem Benutzer eine **virtuelle Maschine** (Prozeß) für die Ausführung seines Programms zur Verfügung

- ❑ **Wichtige Funktionen** des Betriebssystems
 - Ein-/Ausgabe
 - Daten- und Programmverwaltung
 - Programmausführung
 - Bereitstellung von Dienstleistungsprogrammen wie Übersetzer, Texteditoren usw.
 - Koordination der zeitlich verzahnten (überlappten) Ausführung der Programme verschiedener Benutzer
 - Durchführung und Überwachung von Schutzmaßnahmen

Bekannte Betriebssysteme

<i>Betriebssystem</i>	<i>Computer</i>
MS-DOS	PC
MS-DOS mit MS Windows 3.x	PC
OS/2	PC
Windows 95, 98, 2000 und XP	PC
Windows NT	PC und Workstation
MVS, VM/SP	Mainframes
UNIX	Workstations
LINUX	PC

Betriebsmittel

Betriebsmittel (engl. *ressources*) sind alle Hard- und Softwarekomponenten, die zur Ausführung eines Programmes gebraucht werden. Dazu zählen also

- Hardwarekomponenten wie Prozessor, Speicher und E/A Geräte.
- Softwareobjekte wie Programmvariable, Befehle, Daten und Dateien.

Betriebsmittel unterteilt in

- Reale BM: Die wirklich vorhandenen Hard- und Softwareobjekte.
- Virtuelle BM: Vom Betriebssystem simulierte BM mit scheinbar den gleichen Eigenschaften wie die realen BM. Anzahl unbegrenzt. Beispiel virtueller Bildschirm.
- Logische BM: Abstraktion von den technischen oder physikalischen Eigenschaften der realen BM. Beispiel Datei, von der nur der logische Aufbau wie Zugriffsart, Satzformat und Blockgröße interessiert, nicht aber ihre physikalische Speicherung z.B. auf der Festplatte.

UNIX-Dateisystem (1)

□ Haupteigenschaften

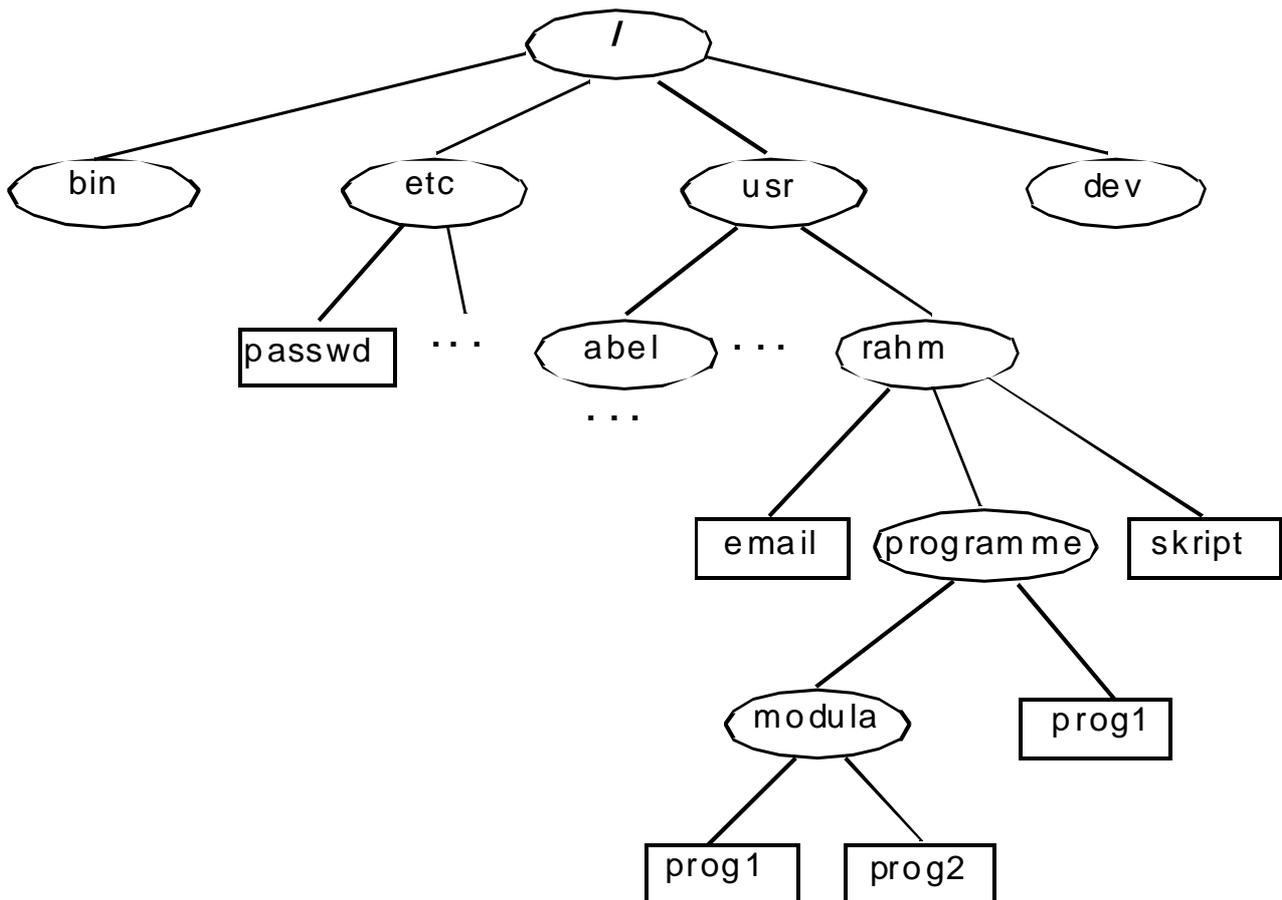
- Hierarchische Struktur des Dateisystems: Kataloge und Dateien
- Gleichförmige Behandlung aller Daten in verschiedenen Dateien
- Dynamische Erzeugung und Löschung von Dateien
- Schrittweises Wachstum von Dateien (nach Bedarf)
- Dateibezogener Schutz der Daten
- Behandlung von Peripheriegeräten (wie Terminals oder Bandgeräte) als Dateien

□ Aufbau

- Baumartige Struktur mit einem Wurzelknoten (/) (root)
- Jeder Nicht-Blattknoten (non-leaf) ist ein Katalog von Dateien (directory)
- Blattknoten sind entweder Kataloge, normale Dateien oder spezielle Gerätedateien
- Der eindeutige Pfad innerhalb des Dateibaumes zu einer bestimmten Datei bildet ihren eindeutigen Namen (Pfadnamen).

UNIX-Dateisystem (2)

- Beispiel eines hierarchisch strukturierten Katalogs (Dateibaum)



- Auswahl/Adressierung durch Pfadnamen

- absolute Adressierung

- /etc/passwd

- /usr/rahm/programme/modula/prog2

- relative Adressierung beginnt beim aktuellen Katalog

- (Working Directory /usr/rahm)

- programme/modula/prog2

- email

- programme/prog1

UNIX-Dateisystem (3)

□ Umgang mit Katalogen

mkdir directory1	Erzeugen des angegebenen Katalogs directory1
cd [directory1]	Ändern des aktuellen Katalogs; directory1 wird der neueaktuelle Katalog. Das Kommando cd (ohne Namensangabe) stellt den Katalog ein, der bei Beginn der Sitzung aktuell war (Home Directory)
pwd	Ausgabe des Namens des aktuellen Katalogs (Working Directory)
rmdir directory1	Löschen von Katalog directory1, wenn er keine Dateien mehr enthält.
Option: -r	
(Rekursiv-Modus)	löscht zuerst alle Dateien/Kataloge des Katalogs und dann den Katalog selbst

□ Umgang mit Dateien

ls dateinamen	Auflisten der angegebenen Dateinamen
Optionen: -t, -l, -u, ...	nach Uhrzeit, langes Format, nach letztem Zugriff, ...
cp datei1 datei2	Kopieren der datei1 nach datei2; falls vorhanden, wird datei2 überschrieben
mv datei1 datei2	Verlagern (Umbenennen) von datei1 nach datei2; falls vorhanden, wird datei2 ersetzt
rm dateinamen	Löschen der angegebenen dateinamen
cat dateinamen	Ausgeben der angegebenen dateinamen
vi datei1	Editieren der Datei datei1
sort datei1	Datei1 zeilenweise alphabetisch sortieren
diff datei1 datei2	Ausgabe aller Unterschiede
wc datei1	Zählen der Zeilen, Worte und Zeichen von datei1
man kommandoname	Ausgabe einer Beschreibung dieses Kommandos
<i>Bemerkung:</i>	
dateinamen	Name einer Datei oder eine Liste von Dateinamen; "*" dient als "Wild Card"
more dateinamen	Vergleiche cat, mit Funktionen zum Blättern
pr dateinamen	Formatierte Ausgabe der angegebenen Dateinamen