

Vorlesung: Rechnerstrukturen, Teil 2 (Modul IP7)

J. Zhang

zhang@informatik.uni-hamburg.de

Universität Hamburg

Fachbereich Informatik

AB Technische Aspekte Multimodaler Systeme

10. Januar 2003

Inhaltsverzeichnis

1. Allgemeine Informationen	3
2. Einführung in die Rechnerarchitektur	4
“Moore’s Law”	6
Motivation	9
Beschreibungsebenen	12
Rechnerstrukturen	22
Wie rechnet ein Rechner?	31

Allgemeine Informationen

Vorlesung (1):	Di. u. Do. 10:00 c.t - 11:00
Pause:	11:00 - 11:10
Vorlesung (2):	11:10 - 11:55
Raum:	Phil A
Web:	http://tams-www.informatik.uni-hamburg.de/lehre/
Name:	Prof. Dr. Jianwei Zhang
Büro:	F308
E-mail:	zhang@informatik.uni-hamburg.de
Sprechstunde:	Donnerstag 15:00 - 16:00
Sekretariat:	Tatjana Tetsis
Büro:	F311
E-mail:	tetsis@informatik.uni-hamburg.de

Einführung in die Rechnerarchitektur

Rapide sich veränderndes Gebiet:

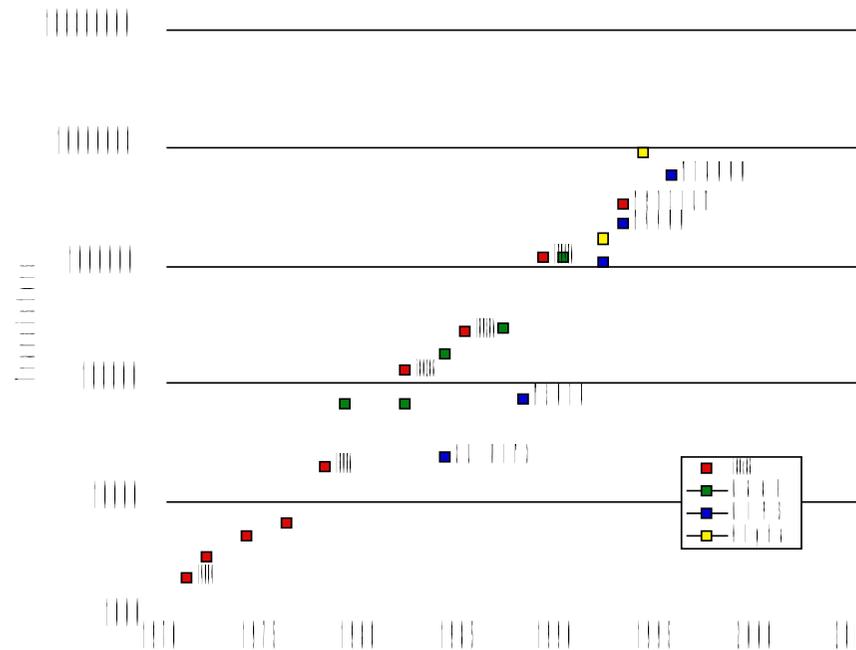
- Vakuum Röhre Transistor IC VLSI
- verdoppelt sich alle 1,5 Jahre ("Moore's Law"):
 - ◆ Speicherkapazität
 - ◆ Prozessorgeschwindigkeit (durch Fortschritt in der Technologie und der Organisation)

Überblick der Entwicklungsgeschichte

DRAM chip capacity

DRAM	
Year	Size
1980	64 Kb
1983	256 Kb
1986	1 Mb
1989	4 Mb
1992	16 Mb
1996	64 Mb
1999	256 Mb
2002	1 Gb

Microprocessor Logic Density



"Moore's Law"

Umgangssprachliche Formulierung:

Bei konstanten Kosten verdoppelt sich die Rechenleistung eines Mikroprozessors alle 18 Monate.

Bemerkungen:

- Moores Vermutung bestätigt sich seit 1975.
- Moores ursprüngliche Aussage betrifft die Steigerungsrate für die Anzahl der Schaltkreise auf einem Chip.
- Moores Vermutung wird allgemein als "Moore's Law" bezeichnet.

"Moore's Law" – Formel

$$L(t) = L(0) \times 2^{t/18}$$

Die Zeit t wird in Monaten gemessen.

$L(t)$ = Leistung zum Zeitpunkt t .

$L(0)$ = Leistung zum Zeitpunkt 0.

Einige Formelwerte:

Jahr 1:	1,5874
Jahr 2:	2,51984
Jahr 3:	4
Jahr 5:	10,0794
Jahr 6:	16
Jahr 7:	25,3984
Jahr 8:	40,3175

Leistungssteigerung der Spitzenrechner seit 1993

Jahr	Rechner	Linpackleistung in Gflop/s	Zahl der Prozessoren
1993	Fujitsu NWT	124	140
1994	Intel Paragon XP/S MP	281	6.768
1995	Intel Paragon XP/S MP	281	6.768
1996	Hitachi CP-PACS	368	2.048
1997	Intel ASCI Red (200 MHz Pentium Pro)	1.338	9.152
1998	ASCI Blue-Pacific (IBM SP 640E)	2.144	5.808
1999	ASCI Intel Red (Pentium II Xeon)	2.379	9.632
2000	ASCI White, IBM (SP Power 3)	4.903	7.424
2001	ASCI White, IBM (SP Power 3)	7.226	7.424
2002	Earth Simulator, NEC	35.610	5.104

Motivation

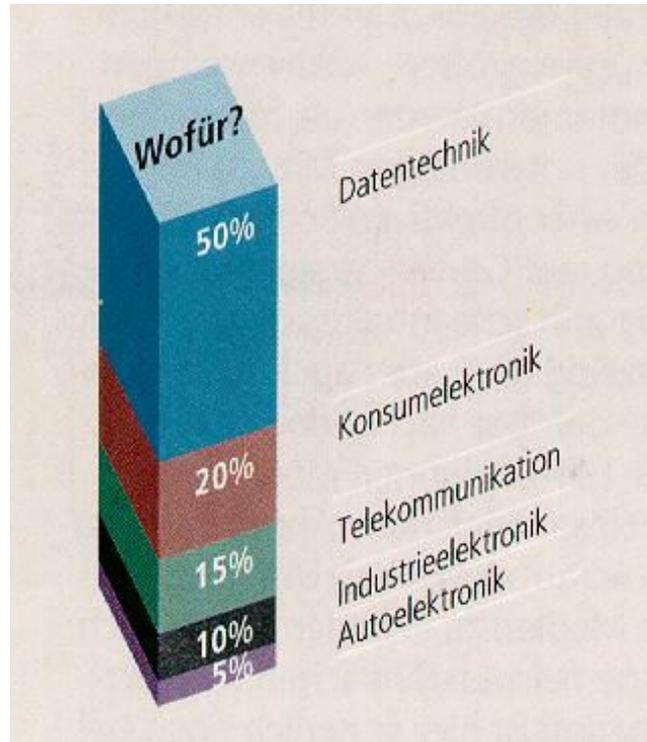
Was Sie lernen werden:

- wie Computer funktionieren, als grundlegende Basis
- wie man ihre Leistung analysiert (und wie nicht)
- Fragen moderner Prozessoren (Caches, Pipelines)

Warum sollte man diese Dinge lernen?

- Sie wollen Computerwissenschaftler werden
- Sie wollen Software entwickeln, die praktisch genutzt wird (daher brauchen wir Leistung)
- Hardware wird immer enger mit Software als Embedded Systems gekoppelt
- Sie müssen Verkaufsentscheidungen treffen oder einen “Expertenrat” abgeben

Bereiche der Technischen Informatik



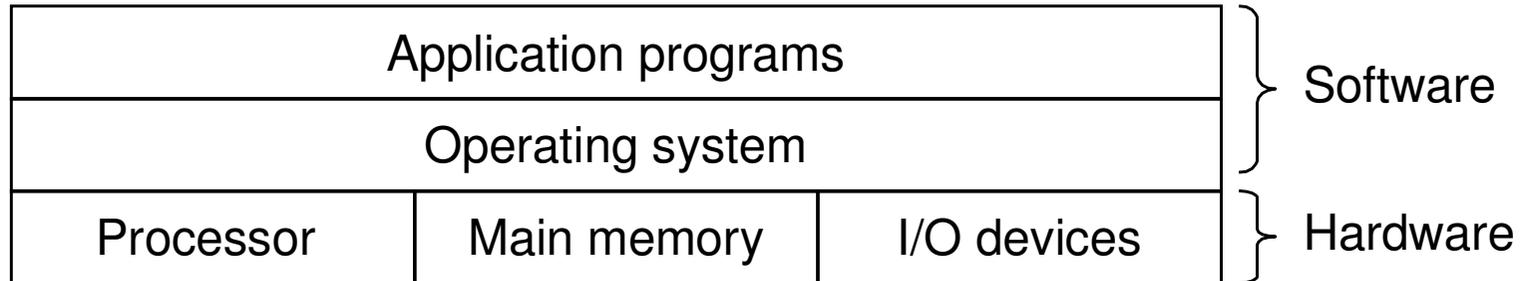
Mechatronik, Robotik, Echtzeit-Regelung, Prozeßrechentechnik, Embedded-Systems. Der Rest liegt in Anwendungsfeldern der Mikroelektronik.

Motivation - 2

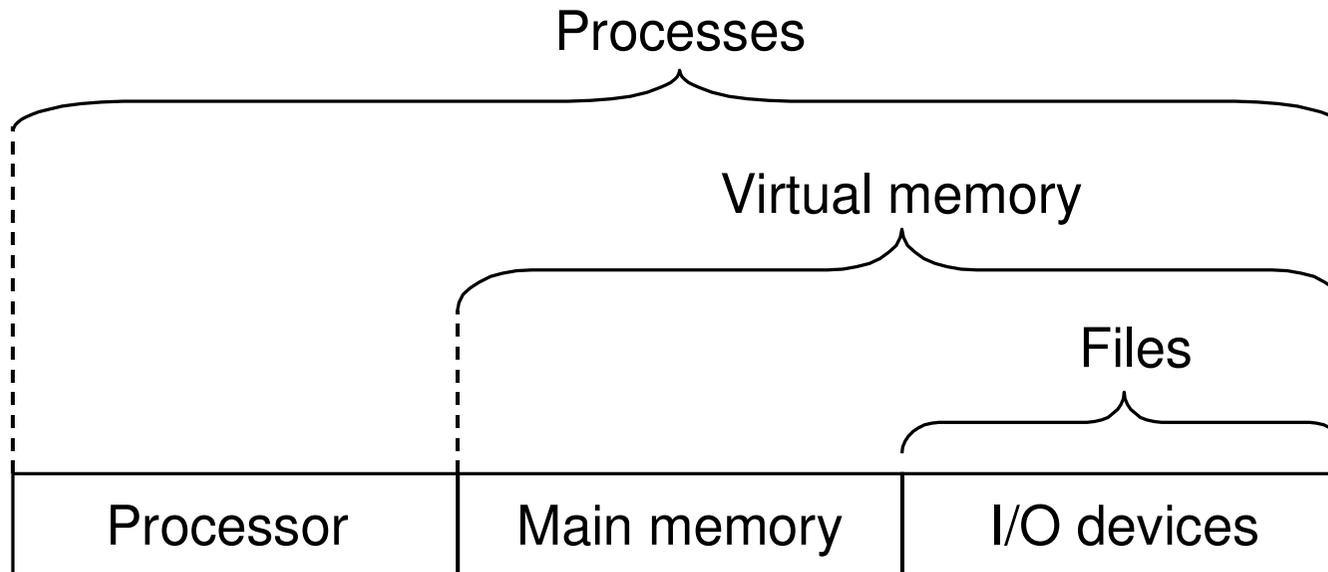
Was zieht uns an:

- Veränderung
- Es ist aufregend
- Es wird noch aufregender
- Große Bedeutung für alle anderen Aspekte der E-Technik und Informatik

Beschreibungsebenen – eine Schichten-Ansicht

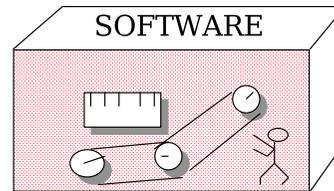


Von einem Betriebssystem gelieferte Abstraktionen



Befehlssatzarchitektur (ISA)

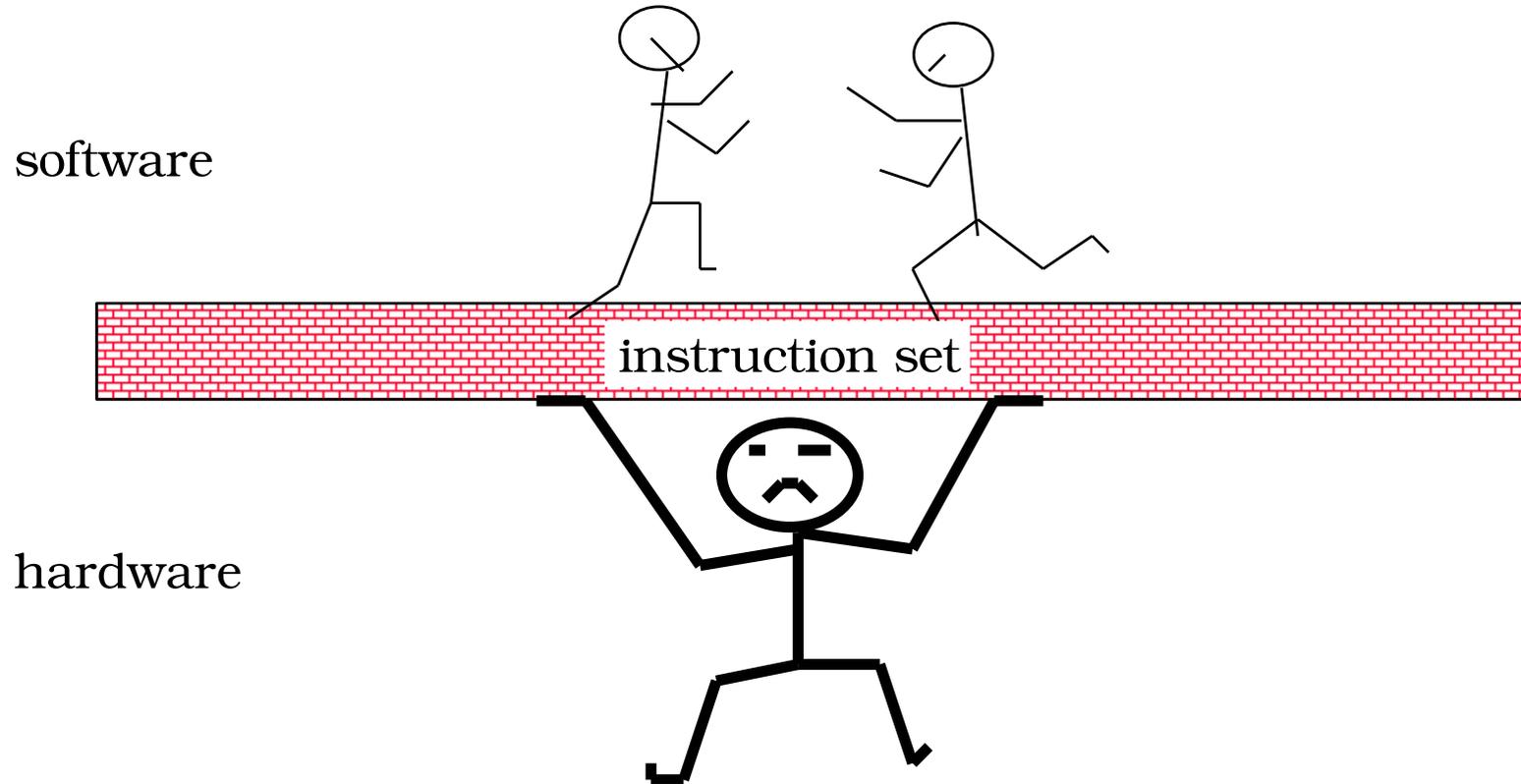
Befehlssatzarchitektur: Instruction Set Architecture (ISA):
die vom Programmierer betrachteten Attribute der konzeptionellen
Struktur



+ des funktionellen Verhaltens, die beinhalten:

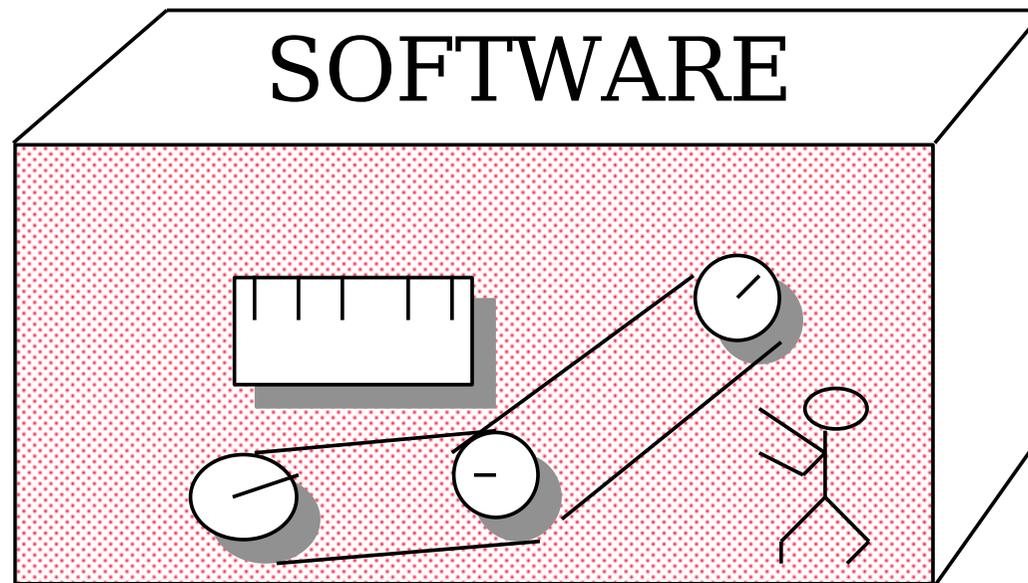
- Organisation des programmierbaren Speichers
- Datentypen und Datenstrukturen: Codierungen und Darstellungen
- Befehlssatz
- Befehlsformate
- Modelle für Befehls- und Datenzugriffe
- Ausnahmebedingungen

Der Befehlssatz: Ein kritisches Interface



Sichtweise des Logikdesigners

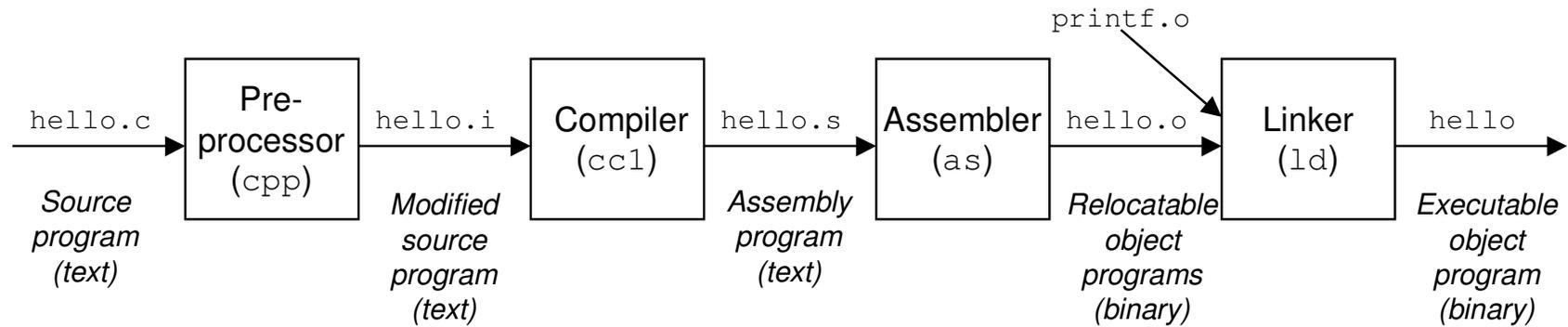
ISA EBENE
Funtionseinheiten u. Verbindungen



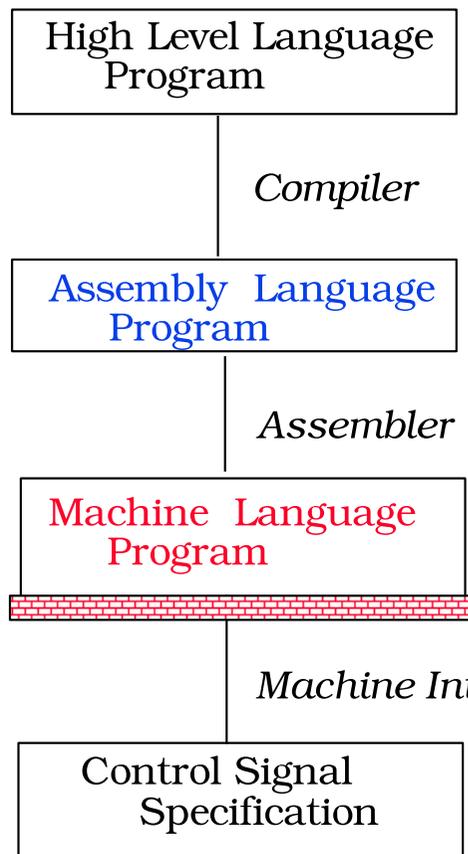
Beispiele für Befehlssatzarchitekturen

Digital Alpha	1992-97
HP PA-RISC	1986-96
Sun Sparc (v8, v9)	1987-95
SGI MIPS (MIPS I, II, III, IV, V)	1986 - 96
Intel (8086, 80286, 80386, 80486, Pentium, MMX, ...)	1978 -
Motorola/IBM/Apple (PowerPC 602/603/604/740/750/7400)	1993 -
IA 64	2000

Das Kompilierungssystem



Repräsentationsebenen



```
temp = v[k];  
v[k] = v[k+1];  
v[k+1] = temp;
```

```
lw $15, 0($2)  
lw $16, 4($2)  
sw $16, 0($2)  
sw $15, 4($2)
```

```
0000 1001 1100 0110 1010 1111 0101 1000  
1010 1111 0101 1000 0000 1001 1100 0110  
1100 0110 1010 1111 0101 1000 0000 1001  
0101 1000 0000 1001 1100 0110 1010 1111
```

```
ALUOP[0:3] <= InstReg[9:11] & MASK
```

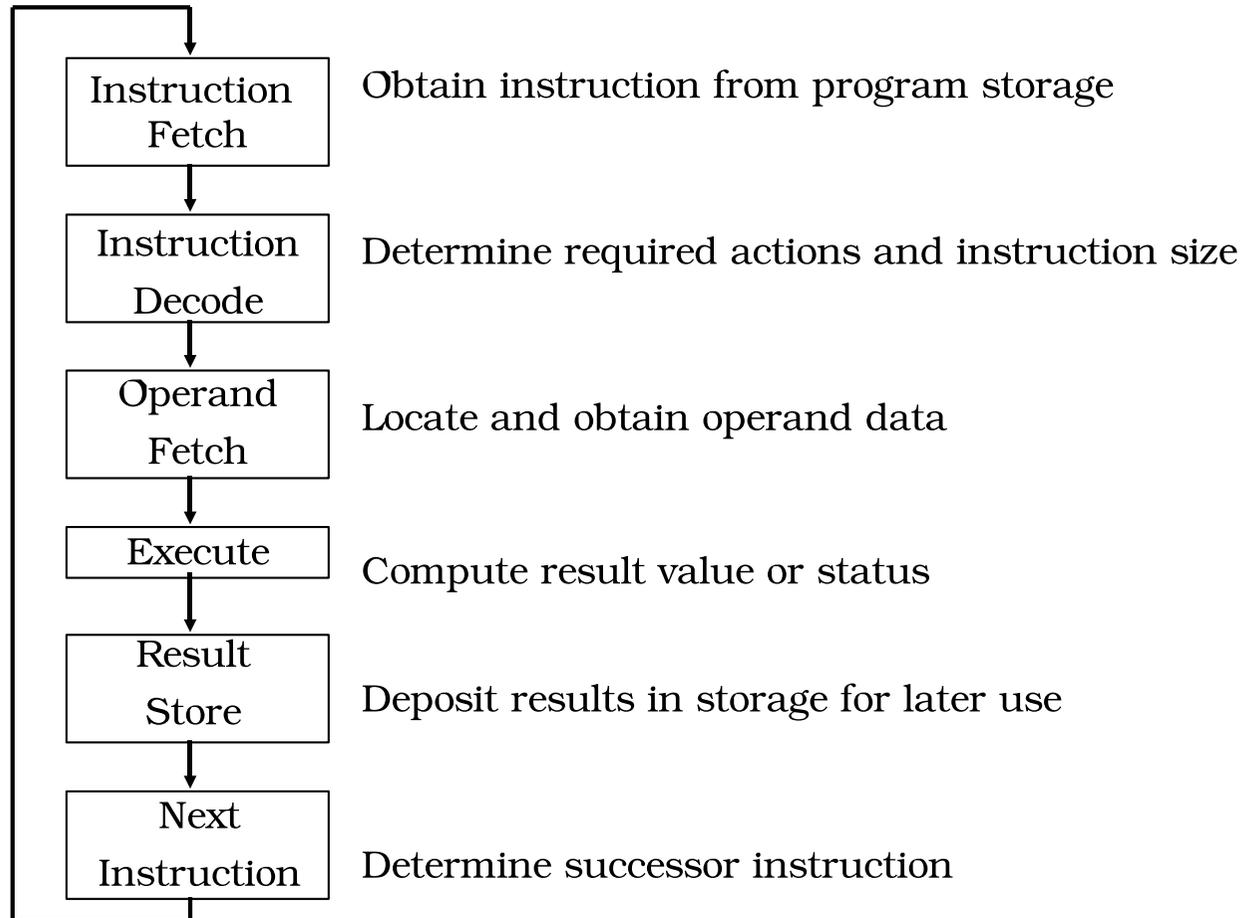
Assembler

In Assembler werden Sie selten ein Programm schreiben müssen.

Das Verständnis des Assemblers ist unerlässlich für das Verständnis des Ausführungsmodells auf der Maschinenebene:

- Programmverhalten bei Fehlern
 - ◆ Das High-Level Sprachmodell ist dort nicht anwendbar
- Die Programmleistung verstärken
 - ◆ Die Ursachen für Programm-Ineffizienz verstehen
- Systemsoftware implementieren
 - ◆ Der Compiler hat den Maschinencode als Ziel
 - ◆ Die Betriebssysteme müssen den Prozesszustand verwalten

Ausführungszyklus



Rechnerstrukturen

Bestandteile eines Rechners:

- Input (Maus, Keyboard)
- Output (Bildschirm, Drucker)
- Speicher (Laufwerke, DRAM, SRAM, CD)
- Netzwerk

Unser primärer Schwerpunkt: der Prozessor (Datenpfad und Kontrolle)

- wird mit Hilfe von Millionen Transistoren implementiert
- ist unmöglich durch die Untersuchung jedes einzelnen Transistors zu verstehen
- Wir brauchen...

Rechnerstrukturen - Weitere Betrachtung

Rechnerstrukturen =

- Rechnerarchitektur
- Implementierung

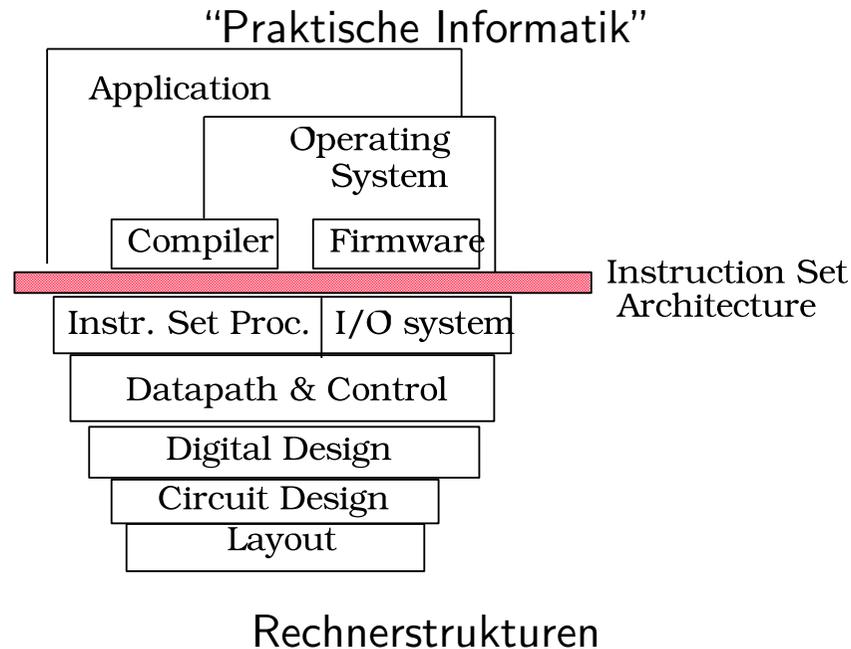
Rechnerarchitektur =

- Schnittstelle zwischen Rechner und Benutzer =
 - ◆ Befehlssatzarchitektur
 - ◆ Maschinenorganisation

Implementierung

- Hardware-Aufbau von Komponenten, die die Rechnerarchitektur realisieren
- Speichereinheiten, Recheneinheiten, Verbindungssysteme,...

Was umfasst der Begriff Rechnerstrukturen?



- Koordination vieler Abstraktionsebenen der Beschreibung
- Berücksichtigung ständig wechselnder äußerer Einflüsse
- Entwurf, Leistungsmessung, Leistungsbewertung

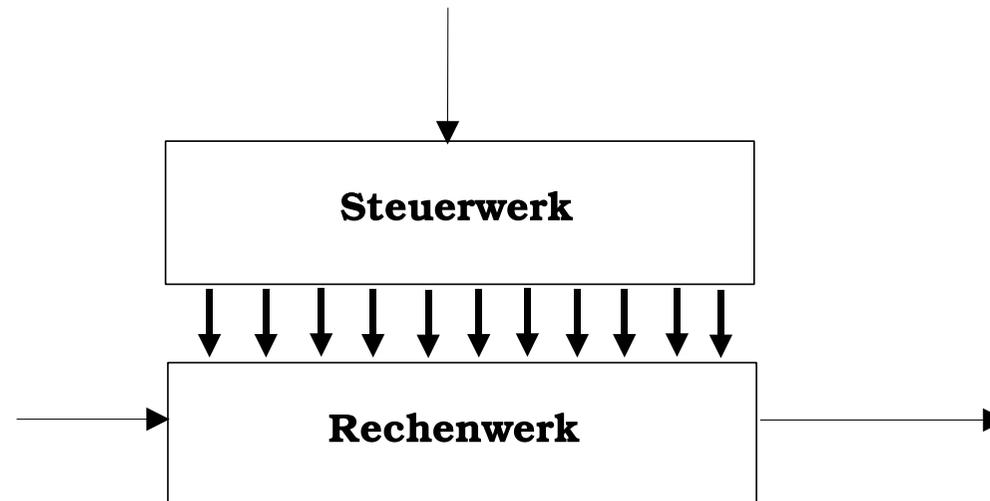
Maschinenorganisation

- Fähigkeiten und Leistung der prinzipiellen Funktionseinheiten
 - ◆ (z.B., Registers, ALU, Shifters,...)
- Verbindungen zwischen diesen Einheiten
- Informationsfluss zwischen den Komponenten
- Logik und Methoden zur Realisierung des Informationsflusses
- “Choreographie” der Funktionseinheiten
- Beschreibung auf Register-Transfer-Ebene (Register Transfer Level (RTL))

von Neumann Architektur

Charakteristika:

1. Daten und Befehle liegen im gleichen Schreib-Lese-Speicher.
2. Der Zugriff auf Speicherinhalte erfolgt über die Adresse der Speicherzelle, Speicherinhalte sind nicht typisiert.
3. Die Programmausführung erfolgt sequentiell, Befehl für Befehl.



Treibende Kräfte auf Rechnerstrukturen

- Technologie
- Anwendungen
- Programmiersprachen
- Betriebssysteme
- Geschichte

Technologie

- Gegen 1985 entstehen Ein-Chip 32-Bit Prozessoren und Single-Board-Computer
 - ◆ Workstations, PCs, Multiprozessoren basieren seither auf dieser Technik
- Mit den neu auf dem Markt erscheinenden Prozessoren ist Mainframe-Leistung auf einem Chip verfügbar

Technologiebedingtes Wachstum in Zahlen

- Prozessor
 - ◆ Transistorzahl: ca 30% pro Jahr (heute ca. 100 Mio.)
 - ◆ Taktrate: ca 20 % pro Jahr
- Speicher
 - ◆ DRAM Kapazität: ca 60% pro Jahr (4x alle 3 Jahre)
 - ◆ Zugriffsgeschwindigkeit: ca. 10 % pro Jahr
 - ◆ Kosten pro Bit: sinken um ca. 25% pro Jahr
- Plattenspeicher
 - ◆ Kapazitätssteigerung: ca. 60% pro Jahr

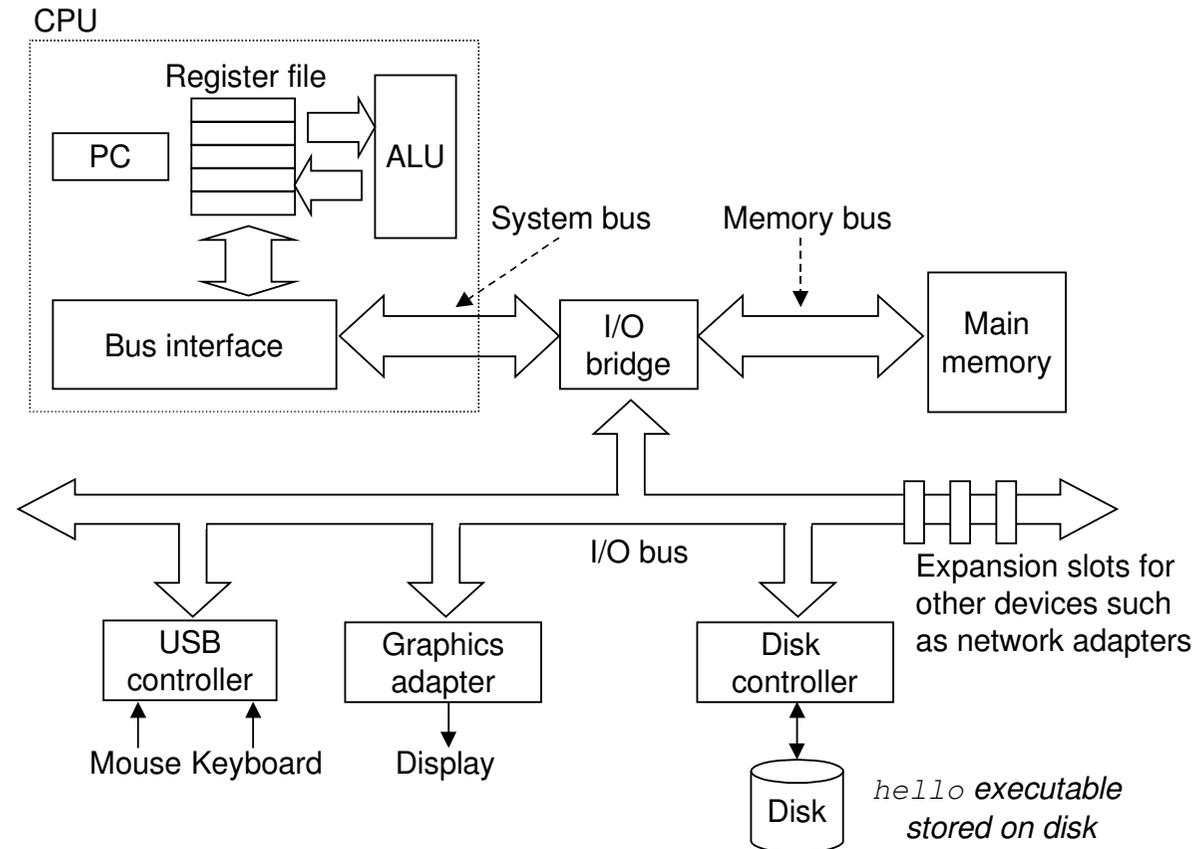
Beispiel: Der PowerPC 7540 (G4)

Geschwindigkeit	bis 1,2 GHz
Busfrequenz	133 MHz
Bus Interface	64 Bits
L1 Cache	32/32 kByte
L2 Cache	256 kByte
Prozess	0,18 μm
U_{int}	1,3 - 1,6V
$U_{I/O}$	1,8V / 2,5 V
Leistung	21,3 W / 1 GHz
Performance	2280 Drystone MPS @ 1GHz

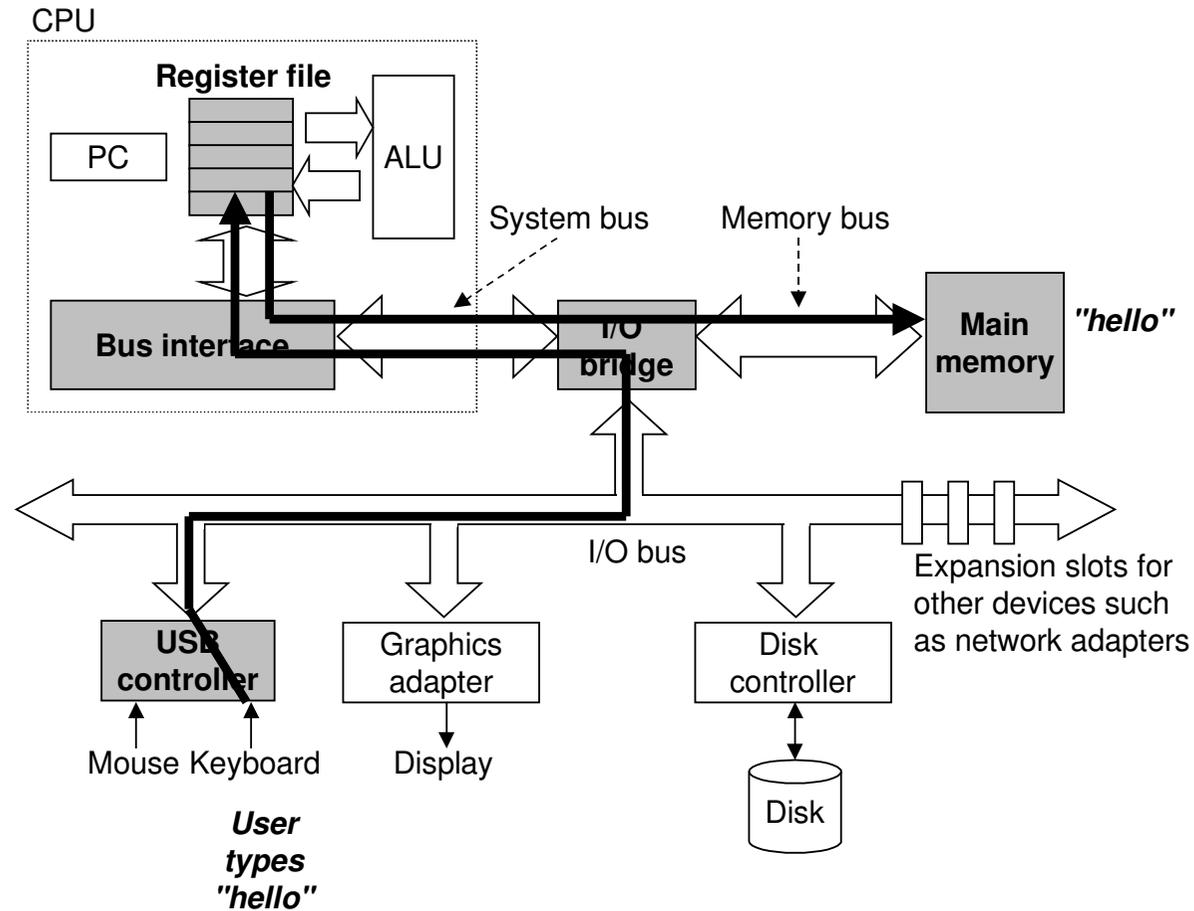
Wie rechnet ein Rechner?

- Datenpfad und Steuerwerk
- Choreografie der Funktionseinheiten
- Caches und Speicherhierarchien
- Wie man Daten sicher verwahrt
- Ein-/Ausgabe und Netzwerke
- “Hello World” - Wie man mit Rechnern kommuniziert

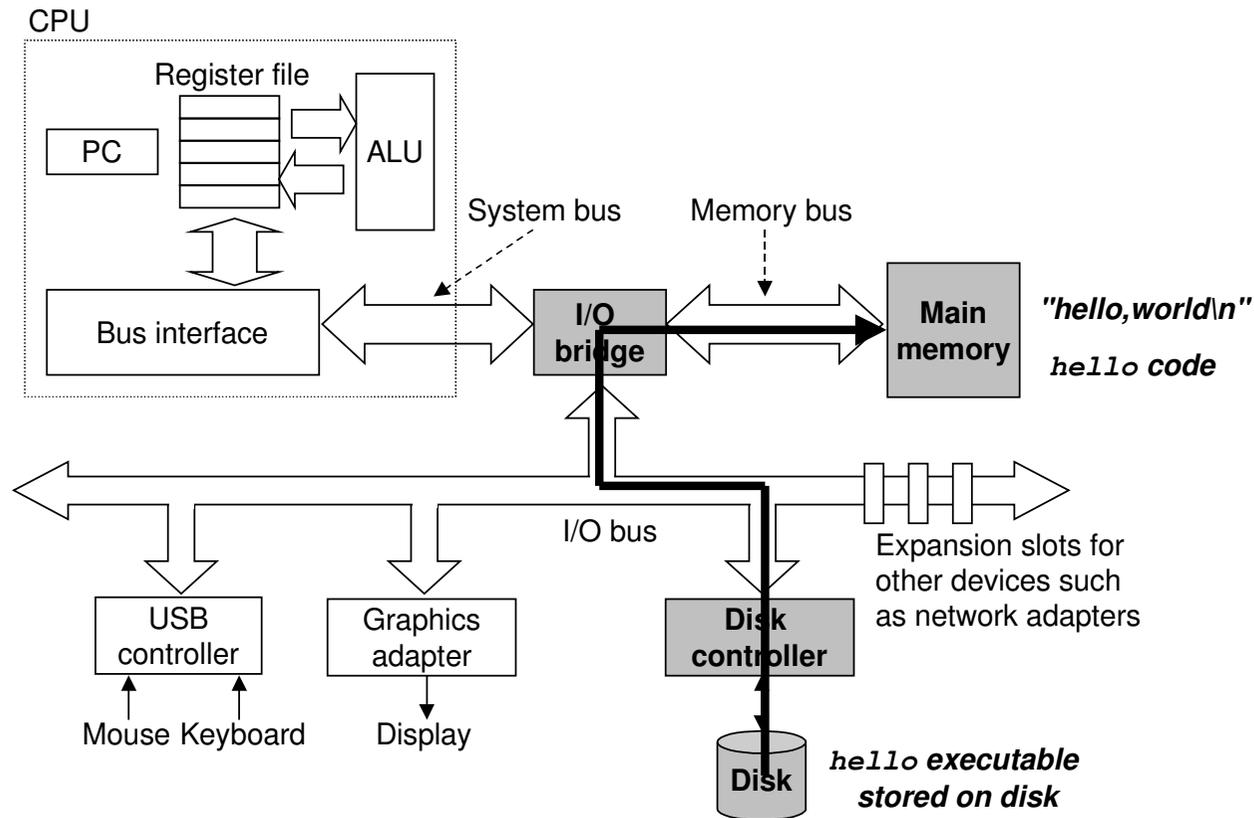
Hardwareorganisation eines typischen Systems



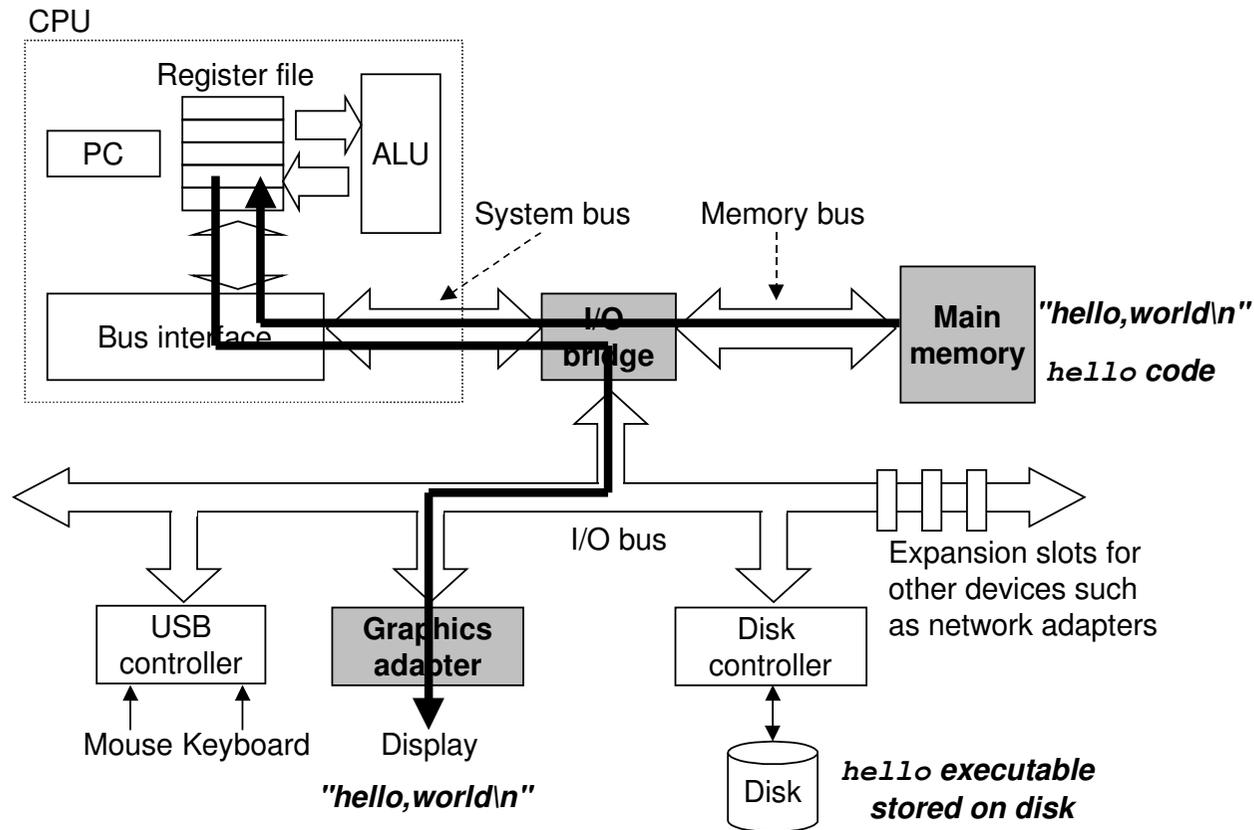
Lesen



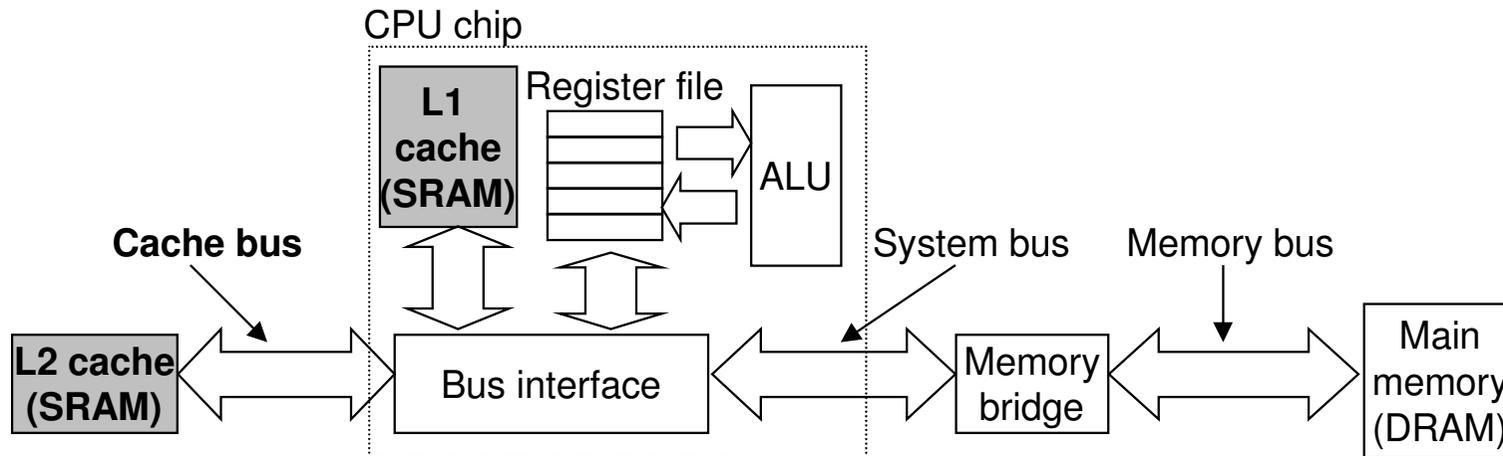
Laden



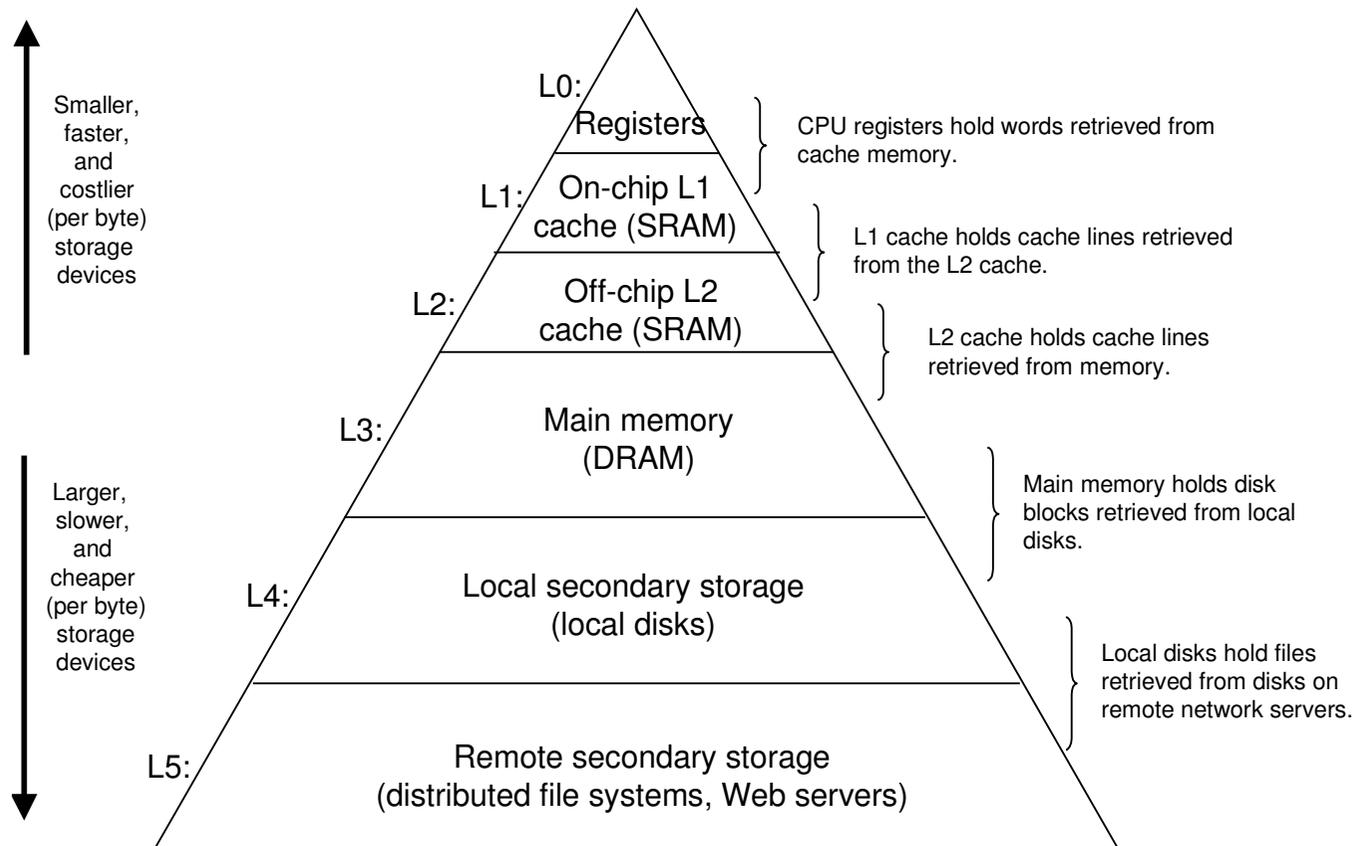
Schreiben



Cache-Speicher



Beispiel einer Speicher-Hierarchie



Der Speicher ist wichtig

- **Der Speicher ist nicht unbegrenzt**
 - ◆ Er muss zugeteilt und verwaltet werden.
 - ◆ Viele Anwendungen werden vom Speicher dominiert.
- **Fehler, die auf den Speicher verweisen, sind besonders gefährlich**
 - ◆ Auswirkungen sind sowohl zeitlich als auch räumlich entfernt
- **Speicherleistung ist nicht gleichbleibend**
 - ◆ Cache und Virtual Memory Auswirkungen können die Programmleistung stark beeinflussen
 - ◆ Die Anpassung des Programms an das Speichersystem kann die Geschwindigkeit bedeutend verbessern

Ergänzende Literatur

Zur Rechnerarchitektur (1. Teil)

Literatur

- [1] Randal E. Bryant and David O'Hallaron. *Computer Systems*. Pearson Education, Inc., New Jersey, 2003.
- [2] David A. Patterson and John L. Hennessy. *Computer Organization and Design. The Hardware / Software Interface*. Morgan Kaufmann Publishers, Inc., San Francisco, 1998.
- [3] Andrew S. Tanenbaum. *Computerarchitektur*. Pearson Studium München, 2006.