Rechnerstrukturen

Universität Trier Peter Sturm Wintersemester 2002/2003

Rechnerstrukturen

0. Organisatorisches

Organisatorisches

- Vorlesungszeiten
 - Dienstags, 14.00-15.30 Uhr, Hörsaal HS 4
 - Donnerstags, 12.30-14.00 Uhr, Hörsaal HS 4
- Übungsblätter
 - wird in der Vorlesung besprochen
- Übungen
 - Dienstags, 8.30-10.00 Uhr, Hörsaal HS 9
- http://www.syssoft.uni-trier.de/systemsoftware
 - eventuelle Termin- und Raumverschiebungen
 - Übungsblätter
 - Folienkopien
- Klausur
 - wird in der Vorlesung besprochen

Begleitliteratur

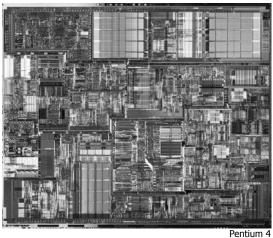
- Structured Computer Organization
 - Andrew S. Tanenbaum
 Prentice Hall, 4. Auflage, 1998
- Computer Architecture A Quantitative Approach
 - J.L. Hennessy, D.A. Patterson Morgan Kaufmann, 2. Auflage, 1996
- Computer Organization and Design The Hardware / Software Interface
 - J.L. Hennessy, D.A. Patterson Morgan Kaufmann, 2. Auflage, 1998

Rechnerstrukturen

1. Einführung

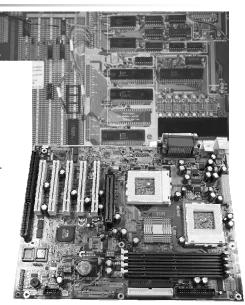
Gegenstand der Vorlesung

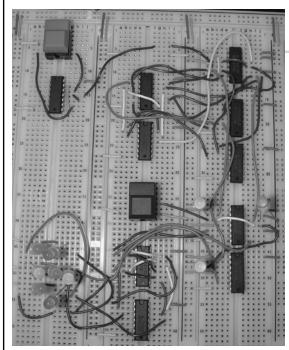
- Struktur moderner Computer
 - Welche Komponenten
 - Welche Architektur
- Funktionsweise
 - Funktion einzelner Komponenten
 - Zusammenspiel
- Bewertungskriterien
 - Effizienz
 - Parallelitätsgrad
 - Skalierbarkeit
 - Zuverlässigkeit
 - Kosten



Ziele

- Grundlagen über
 - Elektronische Schalter (Transistor)
 - Digitale Gatter und Bausteine
- Struktur moderner Computer
 - Aufbau gängiger PCs
 - Pentium, AMD, ...
 - L1- und L2-Cache
 - PCI / ISA
 - DDR-, FPM-, EDO- und SD-RAM, Rambus
 - aber auch SPARC, DEC Alpha, PowerPC
- Assemblerprogrammierung
- Weiterführende Vorlesungen
 - Compilerbau
 - Betriebssysteme

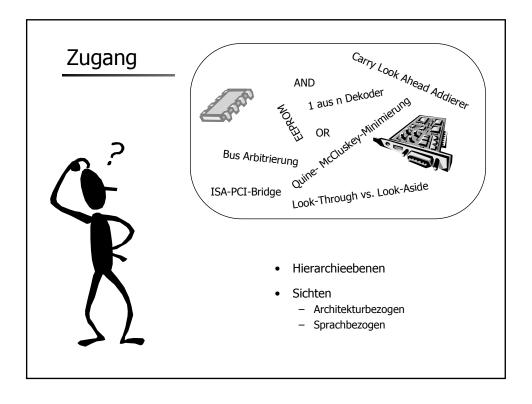


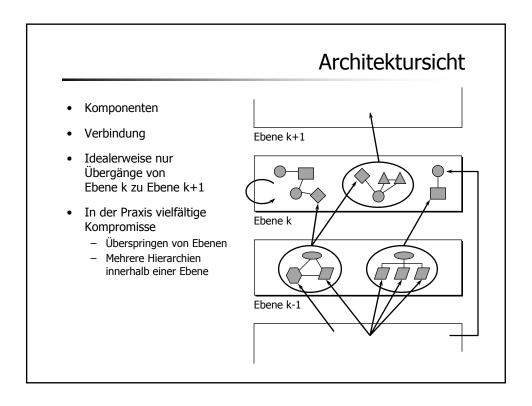


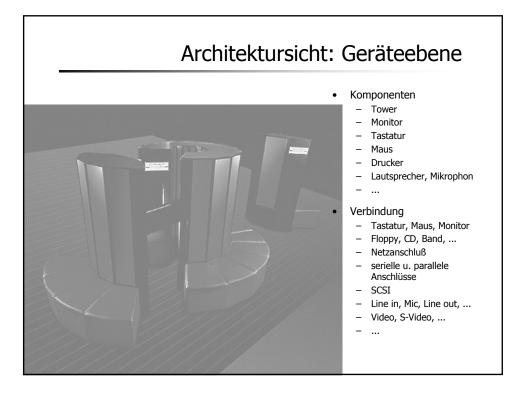
Praxis

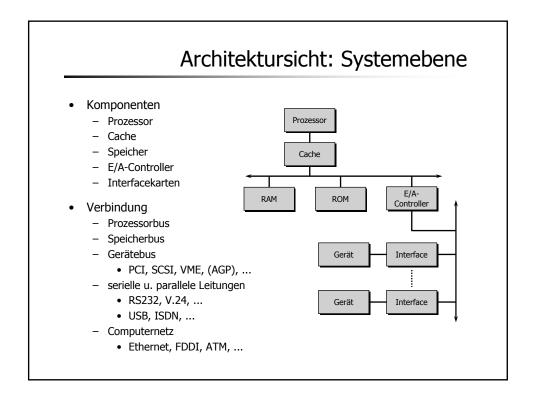
- Einfache Aufbauten auf sogenannten Proto-Boards
- Nur wenige Exemplare
- Freiwillige Übung
- ... auch privat erschwinglich

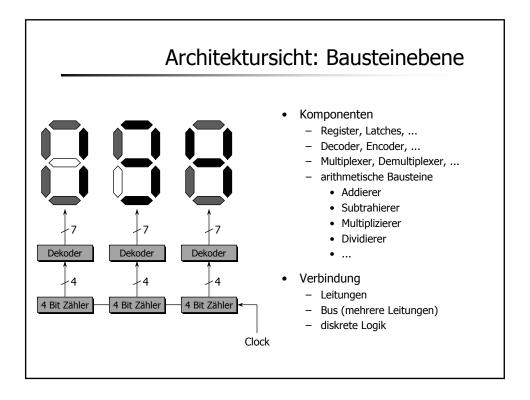
Elektronischer Würfel ©

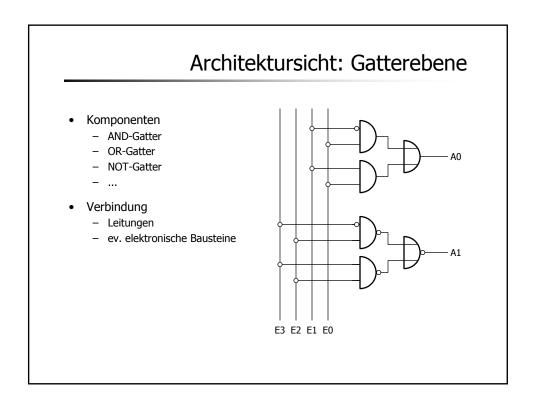


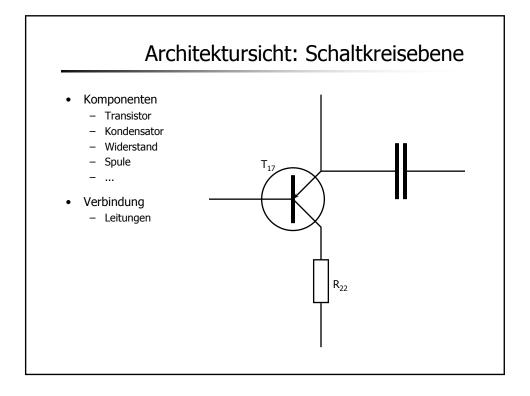










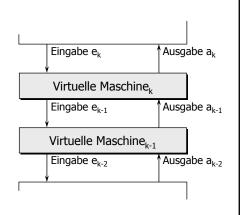


Sprachsicht

• Virtuelle Maschinen implementieren f_k mit

$$(a_k, \sigma', t+1) = f_k(e_k, \sigma, t)$$

- Syntax von a_k und e_k
- Semantik von f
- Interpretieren vs. Übersetzen



Sprachsicht: Differentialgleichungen

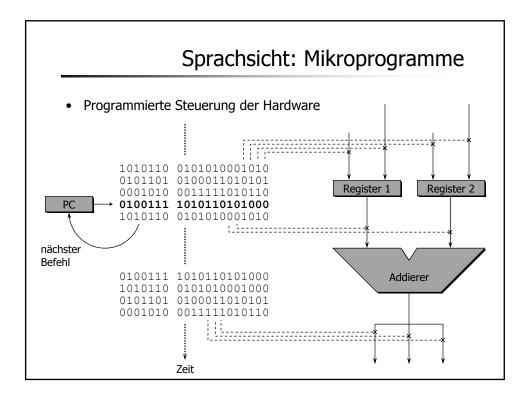
$$E(t) = L\frac{d^2Q(t)}{dt^2} + R\frac{dQ(t)}{dt} + \frac{1}{C}Q(t)$$

Sprachsicht: Boolesche Algebra / Gleichungssysteme

- Wahrheitswerte
 - -0 = False
 - -1 = True
- Grundoperationen
 - Konjunktion
 - Disjunktion
 - Negation
- Verknüpfungsregeln, Gesetze

$$\overline{(a+b)} \cdot (b+ac)$$

$$= \overline{a} \cdot \overline{b} \cdot (b+ac) = \overline{a} \cdot \overline{b} \cdot b + \overline{a} \cdot \overline{b} \cdot a \cdot c = 0$$



Sprachsicht: Register-Transfer-Sprachen

- Register, Datenpfade
- Sprachelemente
 - Ausdrücke (boolesche und arithmetische Funktionen)
 - Bedingte Anweisungen
 - Schleifen
 - ...

```
Operation

IF byte operation

THEN

AX ← AL * SRC

ELSE (* word or doubleword operation *)

IF OperandSize = 16

THEN

DX:AX ← AX * SRC

ELSE (* OperandSize = 32 *)

EDX:EAX ← EAX * SRC

FI;

FI;
```

Sprachsicht: Instruktionssatz

- · Funktionsumfang komplexer Bausteine
 - CPU
 - Coprozessoren
 - MMU
 - Signalprozessor
 - ...
- Ausführungsmodell
 - Kontrollfluß
 - Datenfluß
 - ...

intطا.

INSTRUCTION SET REFERENCE

MUL—Unsigned Multiply

Opcode F6 /4 F7 /4 F7 /4	Instruction	Description
F6 /4	MUL r/m8	Unsigned multiply (AX \leftarrow AL $+ r/m8$)
F7 /4	MUL r/m16	Unsigned multiply (DX:AX ← AX + r/m16)
F7 /4	MUL r/m32	Unsigned multiply (EDX:EAX ← EAX ≈ r/m32)

Description

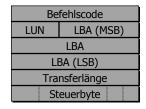
Performs an unsigned multiplication of the first operand (destination operand) and the second operand (source operand) and stores the result in the destination operand. The destination operand is an implied operand located in register AL, AV or EAV (depending on the size of the operand); the source operand is located in a general-purpose register or a memory location. The action of this instruction and the location of the result depends on the opcode and the operand size as shown in the following table.

Operand Size	Source 1	Source 2	Destination
Byte	AL	r/m8	AX
Word	AX	r/m16	DX:AX
Doubleword	EAX	r/m32	EDX:EAX

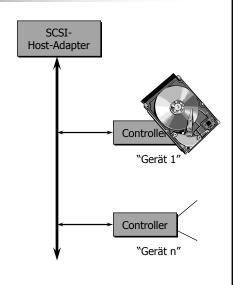
The result is stored in register AX, register pair DX:AX, or register pair EDX:EAX (depending on the operand size), with the high-order bits of the product contained in register AH, DX, or EDX, respectively. If the high-order bits of the product are 0, the CF and OF flags are cleared; otherwise, the flags are set.

Sprachsicht: Datenorientierte Protokolle

Datenpakete



- Protokoll zwischen Sender und Empfänger
 - Mitteilung / Auftrag
 - Synchron / Asynchron
 - Arbitrierung bei Bussen



Sprachsicht: Programmiersprachen

- Vielfältige Syntax und Semantik
- systemnahe Programmiersprachen, z.B. C oder C++
 - geeignete Datenstrukturen
 - Bitfelder
 - Zeiger
 - Repräsentation bekannt
 - einfache Datentypkonversionen (Cast)
 - geringer verdeckter Overhead (Performanz)
- Assembler
 - symbolische Befehle (Mnemonic)
 - 1:1-AbbildungAssemblerbefehl Instruktion
 - Makromechanismen

```
loop: dec r3
    jne out:
    move (r2++), (r1++)
    jmp loop:
out: move (r2++),#0
```

