Rechnerstrukturen

Universität Trier
Peter Sturm
Wintersemester 2001/2002

Rechnerstrukturen

0. Organisatorisches

Organisatorisches

- Vorlesungszeiten
  - Montags, 10-12 Uhr, Hörsaal HS 6
  - Freitags, 12-14 Uhr, Hörsaal HS 6
- Übungsblätter
  - Ausgabe in der Freitag-Vorlesung
  - Abgabe spätestens Donnerstag 12 Uhr
- Übungen: Steffen Rothkugel
  - Montags, 12-14 Uhr, Raum E 52
- http://www.informatik.uni-trier.de/~sturm/Professur
  - eventuelle Termin- und Raumverschiebungen
  - Übungsblätter
  - Folienkopien (meist aber knapp vorher)
- Klausur
  - Zulassungsvoraussetzung 50% der Übungspunkte

Begleitliteratur

- Structured Computer Organization
  - Andrew S. Tanenbaum
    Prentice Hall, 3. Auflage, 1990
- 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, 1997
- Computer Organization and Architecture
  - William Stallings
    Prentice Hall, 4. Auflage, 1995

Rechnerstrukturen

1. Einführung

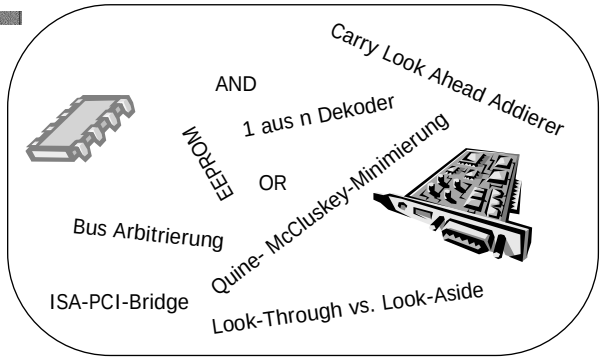
Gegenstand der Vorlesung

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

Ziele

- Grundlagen über
  - Elektronische Schalter (Transistor)
  - Digitale Gatter und Bausteine
- Struktur moderner Computer
  - Aufbau gängiger Pentium-PCs
    - Pentium, Pentium MMX, Pentium Pro und Pentium II
    - L1- und L2-Cache
    - FPM-, EDO- und SDRAM
    - PCI / ISA
  - aber auch SPARC, DEC Alpha, PowerPC
- Assemblerprogrammierung
- Weiterführende Vorlesungen
  - Compilerbau
  - Betriebssysteme

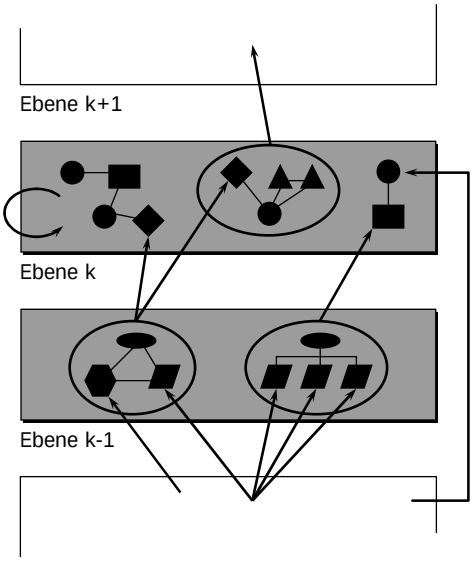
Zugang



- Hierarchieebenen
- Sichten
  - Architekturbezogen
  - Sprachbezogen

Architektursicht

- Komponenten
- Verbindung
- Idealerweise nur Übergänge von Ebene k zu Ebene k+1
- In der Praxis vielfältige Kompromisse
  - Überspringen von Ebenen
  - Mehrere Hierarchien innerhalb einer Ebene

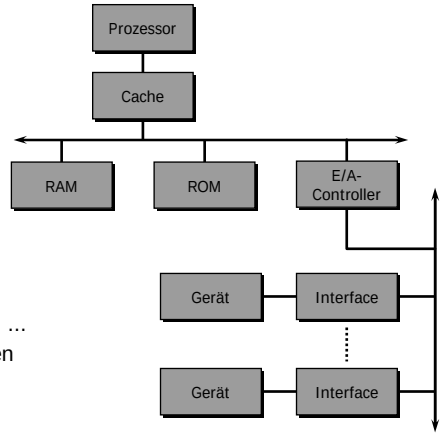


Architektursicht: Geräteebene

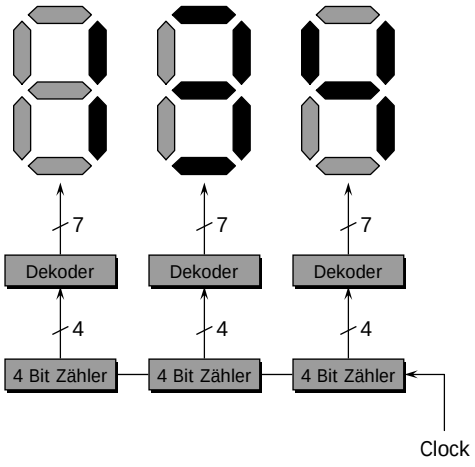
- Komponenten
  - Tower
  - Monitor
  - Tastatur
  - Maus
  - Drucker
  - Lautsprecher, Mikrophon
  - ...
- Verbindung
  - Tastatur, Maus, Monitor
  - Floppy, CD, Band, ...
  - Netzanschluß
  - serielle u. parallele Anschlüsse
  - SCSI
  - Line in, Mic, Line out, ...
  - Video, S-Video, ...
  - ...

Architektursicht: Systemebene

- Komponenten
  - Prozessor
  - Cache
  - Speicher
  - E/A-Controller
  - Interfacekarten
- Verbindung
  - Prozessorbus
  - Speicherbus
  - Gerätebus
    - PCI, SCSI, VME, (AGP), ...
  - serielle u. parallele Leitungen
    - RS232, V.24, ...
    - USB, ISDN, ...
  - Computernetz
    - Ethernet, FDDI, ATM, ...



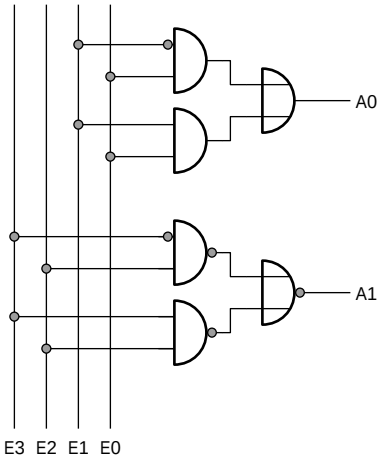
Architektursicht: Bausteinebene



- Komponenten
  - Register, Latches, ...
  - Decoder, Encoder, ...
  - Multiplexer, Demultiplexer, ...
  - arithmetische Bausteine
    - Addierer
    - Subtrahierer
    - Multiplizierer
    - Dividierer
    - ...
- Verbindung
  - Leitungen
  - Bus (mehrere Leitungen)
  - diskrete Logik

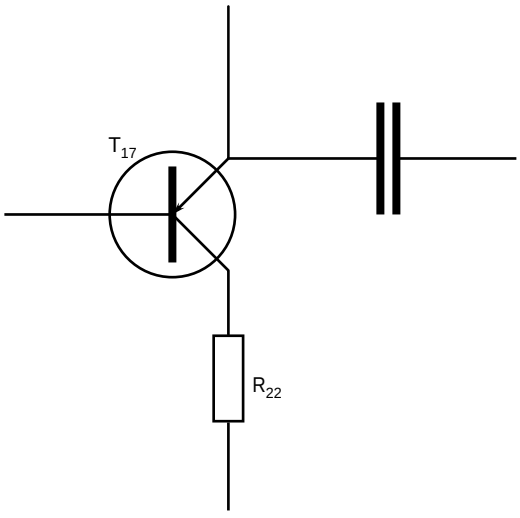
Architektursicht: Gatterebene

- Komponenten
  - AND-Gatter
  - OR-Gatter
  - NOT-Gatter
  - ...
- Verbindung
  - Leitungen
  - ev. elektronische Bausteine



Architektursicht: Schaltkreisebene

- Komponenten
  - Transistor
  - Kondensator
  - Widerstand
  - Spule
  - ...
- Verbindung
  - Leitungen

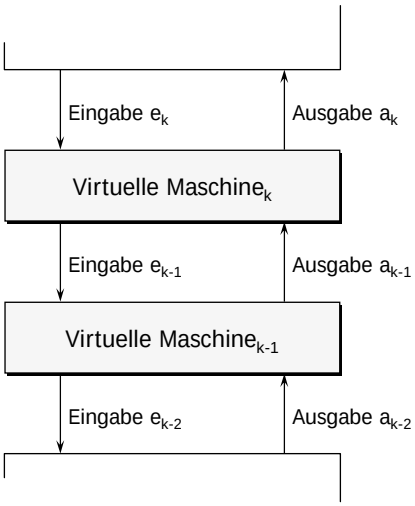


Sprachsicht

- 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



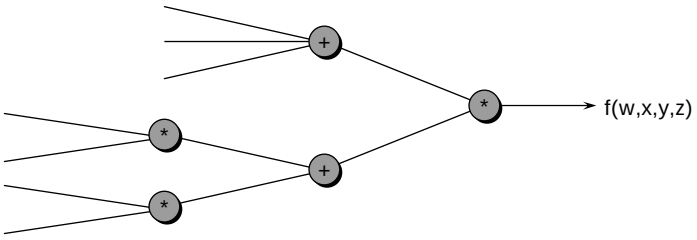
Sprachsicht: Differentialgleichungen

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

Sprachsicht: Boolesche Algebra / Gleichungssysteme

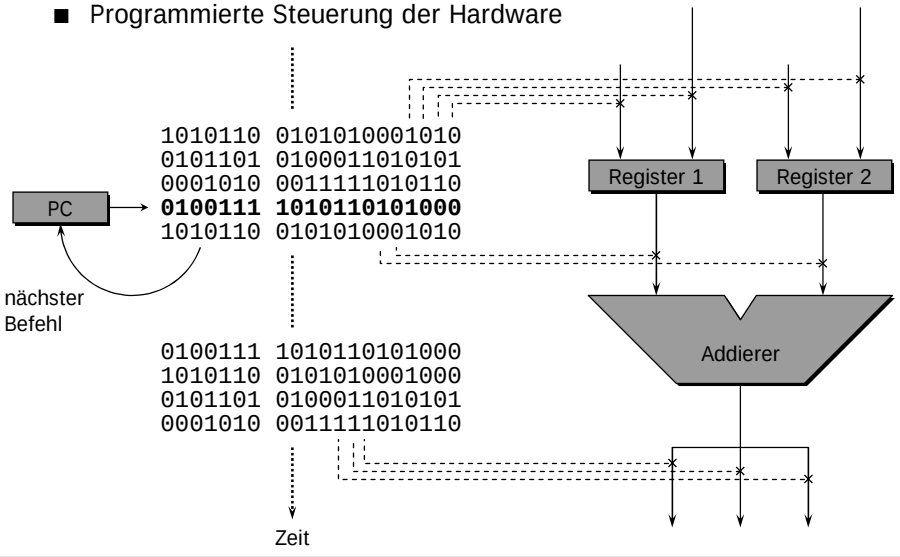
- Wahrheitswerte
  - 0 = False
  - 1 = True
- Verknüpfungsregeln, Gesetze

$$f(w,x,y,z) = (w + \overline{y} + \overline{x}) \cdot (xy + z\overline{y})$$



Sprachsicht: Mikroprogramme

- Programmierte Steuerung der Hardware



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;
```

Flags Affected

The OF and CF flags are cleared to 0 if the upper half of the result is 0; otherwise, they are set to 1. The SF, ZF, AF, and PF flags are undefined.

Sprachsicht: Instruktionssatz

- Funktionsumfang komplexer Bausteine
  - CPU
  - Coprozessoren
  - MMU
  - Signalprozessor
  - …
- Ausführungsmodell
  - Kontrollfluß
  - Datenfluß
  - …

INSTRUCTION SET REFERENCE

intel

MUL—Unsigned Multiply

Opcode	Instruction	Description
F6 /4	MUL *r/m8*	Unsigned multiply (AX ← 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, AX or EAX (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

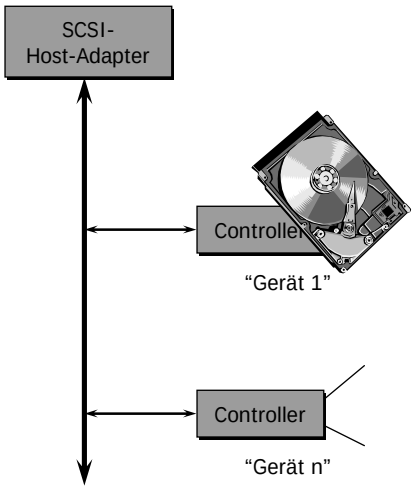
The result is stored in register AX, register pair DX:AX, or register pair EDX:EAX (depending on the operand size), with the high-order bits of the product contained in register AH, DX, or EDX, respectively. If the high-order bits of the product are 0, the CF and OF flags are cleared; otherwise, the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

Sprachsicht: Datenorientierte Protokolle

- Datenpakete

Befehlscode	
LUN	LBA (MSB)
LBA	
LBA (LSB)	
Transferlänge	
Steuerbyte	

- 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-Abbildung
    Assemblerbefehl - Instruktion
  - Makromechanismen

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

Beziehung: Architektur - Sprache