

Von Mikro zu Nano: Mathematik für die Computer von morgen

Peter Benner

mit Unterstützung von André Schneider, Michael Striebel

Mathematik in Industrie und Technik
Fakultät für Mathematik
Technische Universität Chemnitz



Dies academicus
“MATHEMATIK — Alles, was zählt”
TU Chemnitz
12. November 2008

Verbundprojekt im BMBF Programm "Mathematik für Innovationen in Industrie und Dienstleistungen"



TU Berlin (*T. Stykel, M.-S. Hossain*)

TU Braunschweig (*H. Faßbender, J. Amoroch, M. Bollhöfer, A. Eppler*)

TU Chemnitz (*P. Benner, A. Schneider, T. Mach*)

U Hamburg (*M. Hinze, M. Vierling, M. Kunkel*)

FhG-ITWM Kaiserslautern (*P. Lang, O. Schmidt*)

Infinion Technologies AG (*P. Rotter*)

NEC Europe Ltd. (*A. Basermann, C. Neff*)

Qimonda AG (*G. Denk*)



TECHNISCHE UNIVERSITÄT
CAROLO-WILHELMINA
ZU BRAUNSCHWEIG



TECHNISCHE UNIVERSITÄT
CHEMNITZ



Universität Hamburg



Fraunhofer
Institut
Techno- und
Wirtschaftsmathematik



O-MOORE-NICE!

Operational model order reduction for nanoscale IC electronics

EU support via Marie Curie Host Fellowships for the Transfer of Knowledge (ToK) Industry-Academia Partnership Scheme.



TU Chemnitz (*P. Benner, M. Striebel*)

TU Eindhoven (*W. Schilders, D. Harutyunyan*)

U Antwerpen (*T. Dhaene, L. Di Tommasi*)

NXP Semiconductors (*J. ter Maten, J. Rommes*)

- 1 Einführung in die Nanoelektronik**
 - Mikro-/Nanoelektronik
 - Miniaturisierung
 - Schaltkreissimulation

- 2 Mathematische Modellierung von Schaltkreisen**

- 3 Modellreduktion**
 - Ziele
 - Grundideen für Algorithmen

- 4 Fazit**

Worum geht's?

- Entwicklung elektronischer Bauteile und integrierter Schaltkreise ("integrated circuits", ICs), deren Strukturen Größenordnungen im **Mikro-** ($1 \mu\text{m} = 10^{-6}\text{m}$) bzw. **Nano-** ($1 \text{nm} = 10^{-9}\text{m}$) **meterbereich** aufweisen.

Worum geht's?

- Entwicklung elektronischer Bauteile und integrierter Schaltkreise ("integrated circuits", ICs), deren Strukturen Größenordnungen im **Mikro-** ($1 \mu\text{m} = 10^{-6}\text{m}$) bzw. **Nano-** ($1 \text{nm} = 10^{-9}\text{m}$) **meterbereich** aufweisen.
- Übergang von der **Mikro-** zur **Nanoelektronik**, um noch höhere Leistung in noch kleineren Bauteilen bei noch geringeren Kosten zu ermöglichen.
- Der Verkleinerung herkömmlicher Schaltkreise aus Silizium sind physikalische Grenzen gesetzt, auch die Taktfrequenz lässt sich nicht unbegrenzt erhöhen.
- Weitere Miniaturisierung erfordert Einsatz von Nanotechnologie, bisher allerdings noch kein Einsatz von Quanteneffekten.

Anwendungen

- Unterhaltungselektronik (Spielekonsolen, MP3 Player, ...)
- Mobiltelefone, PDAs
- Computertechnik (CPUs, Speicherbausteine, ...)
- Navigation, Sensorik
- Fahrzeugelektronik (ASICs = **Anwendungs-Spezifische ICs**)
- Automatisierungstechnik
- Medizintechnik
- ...



Moore's Law (1965/75) besagt, dass sich die Anzahl der Transistoren auf einem Chip alle 24 Monate verdoppelt.

Mikro zu Nano

Peter Benner

Nanoelektronik

Mikro-
/Nanoelektronik

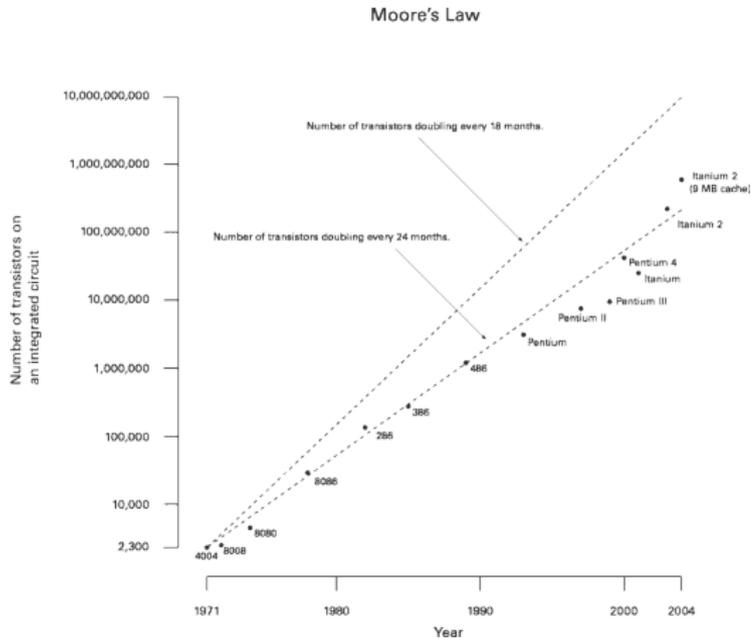
Miniaturisierung

Schaltkreis-
simulation

Modellierung

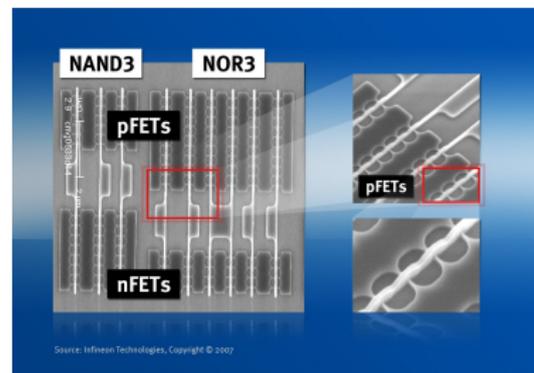
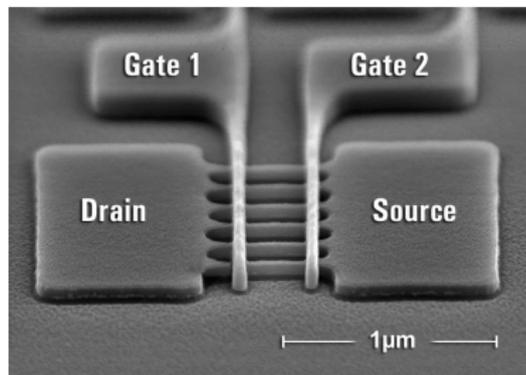
Modellreduktion

Fazit



Herstellungsprozess von ICs beruht heutzutage meist auf der CMOS Technologie ("Complementary Metal Oxide Semiconductor" , dt. *komplementärer Metall-Oxid-Halbleiter*)

Dabei werden immer 2 komplementäre (n- bzw. p-Kanal) Feldeffekttransistoren (MOSFETs) zusammengeschaltet, um Logikoperationen durchzuführen.



CMOS Technologie – Miniaturisierung

Mikrotechnologie

10 μm		z.B. Intel 4004 (1971)
3 μm		z.B. Intel 8086 (1978)
1.5 μm		z.B. Intel 80286 (1982)
1 μm		z.B. Intel 386 (1985)
800 nm	(0.80 μm)	z.B. Intel 486 (1989)
600 nm	(0.60 μm)	z.B. Intel 486DX2 (1990)
350 nm	(0.35 μm)	z.B. Intel Pentium Pro (1995), Nintendo 64
250 nm	(0.25 μm)	z.B. DEC Alpha (1999), Playstation 2
180 nm	(0.18 μm)	z.B. Intel Pentium P4 (2000)
130 nm	(0.13 μm)	

Nanotechnologie (< 100nm)

90 nm		
65 nm		z.B. Intel Core 2 Duo (2006)
45 nm		z.B. Intel Core 2 Extreme (2007)
32 nm		Produktion wird für 2009 erwartet
22 nm		Ende der herkömmlichen CMOS-Herstellung (2011/12)
16 nm		Übergang zur Nanoelektronik (2018)
11 nm		Nanoelektronik (Tunneleffekt) (2022)

Ein Größenvergleich

Transient-Voltage-Suppression (TVS)-Diode von Infineon.



Schutzeinrichtung gegen elektrostatische Entladung, für High-Speed-Schnittstellen, z.B. USB 2.0, Ethernet 10/1000, uvm.

Mikro zu Nano

Peter Benner

Nanoelektronik

Mikro-
/Nanoelektronik

Miniaturisierung
Schaltkreis-
simulation

Modellierung

Modellreduktion

Fazit

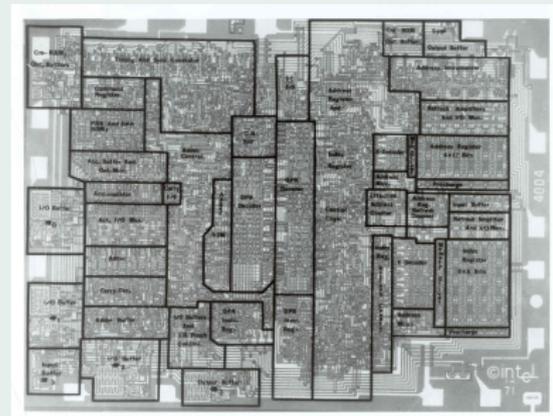
Intel 4004 — der erste Mikroprozessor (1971)

- 4 Bit Architektur,
- 1-lagig,
- 10 μ m Technologie,
- 2.300 Transistoren,
- 64/108 KHz.



Intel 4004 — der erste Mikroprozessor (1971)

- 4 Bit Architektur,
- 1-lagig,
- **10 μ m Technologie,**
- **2.300 Transistoren,**
- 64/108 KHz.



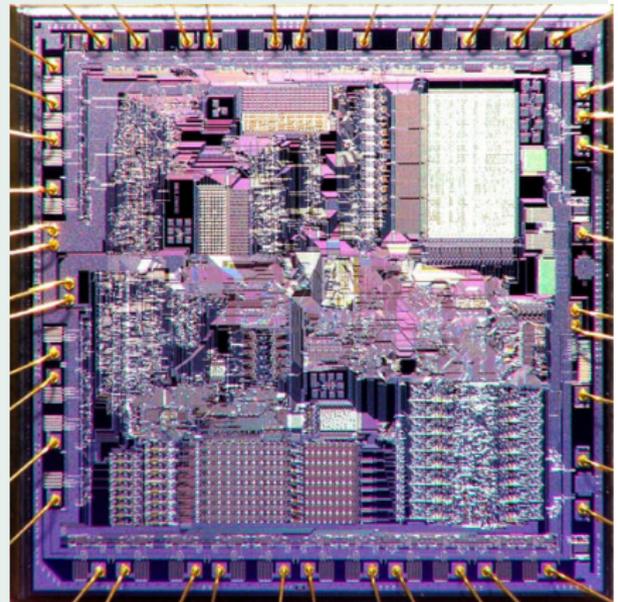
Intel 8086 — der erste 16 Bit-Prozessor (1978)

- 16 Bit Architektur,
- **3 μ m Technologie,**
- **29.000 Transistoren,**
- 5 MHz.



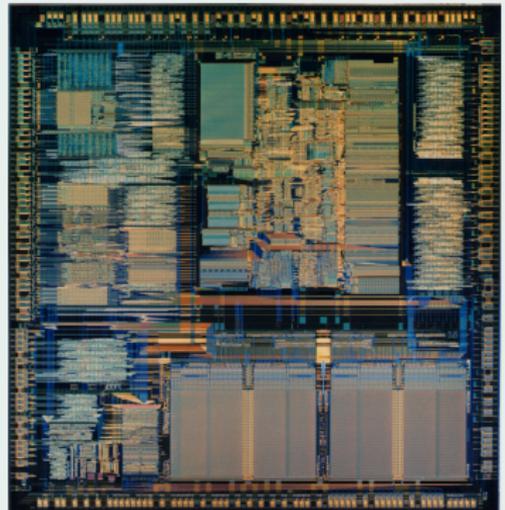
Intel 8086 — der erste 16 Bit–Prozessor (1978)

- 16 Bit Architektur,
- $3\mu\text{m}$ Technologie,
- 29.000 Transistoren,
- 5 MHz.



Intel 386 — der erste 32 Bit–Prozessor (1985)

- 32 Bit Architektur,
- $1\mu\text{m}$ Technologie,
- 275.000 Transistoren,
- 16 MHz.



Mikro zu Nano

Peter Benner

Nanoelektronik

Mikro-
/Nanoelektronik

Miniaturisierung
Schaltkreis-
simulation

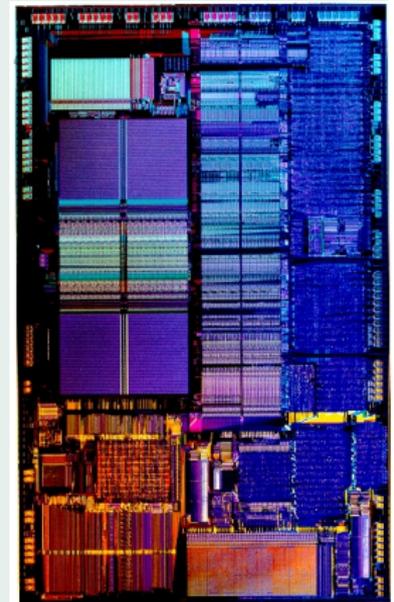
Modellierung

Modellreduktion

Fazit

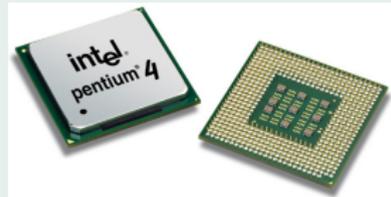
Intel 486 — mehr als 1 Million Transistoren (1989)

- 32 Bit Architektur,
- 600/800nm Technologie,
- 1.200.000 Transistoren,
- 25 MHz.



Intel Pentium 4 — erstmals “Nanotechnologie” (2000)

- 32 Bit Architektur,
- 7-lagig,
- **180nm Technologie,**
- **42.000.000 Transistoren,**
- 1.5 GHz,
- 2 km Leiterbahnen (Interconnect).



Intel Pentium 4 — erstmals “Nanotechnologie” (2000)

- 32 Bit Architektur,
- 7-lagig,
- 180nm Technologie,
- 42.000.000 Transistoren,
- 1.5 GHz,
- 2 km Leiterbahnen (Interconnect).



Mikro zu Nano

Peter Benner

Nanoelektronik

Mikro-
/Nanoelektronik

Miniaturisierung
Schaltkreis-
simulation

Modellierung

Modellreduktion

Fazit

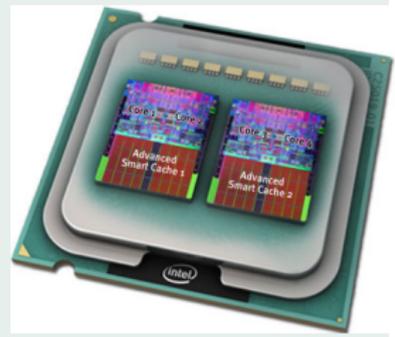
Intel Core 2 Extreme (quad-core) (2007)

- 64 Bit Architektur,
- 9-lagig,
- 45nm Technologie,
- 820.000.000 Transistoren,
- 3 GHz.



Intel Core 2 Extreme (quad-core) (2007)

- 64 Bit Architektur,
- 9-lagig,
- 45nm Technologie,
- 820.000.000 Transistoren,
- 3 GHz.



(Elektronische) Schaltkreissimulation

- Verwendung mathematischer Modelle, um das Verhalten eines elektronischen Schaltkreises oder Bauteils (↔ **Bauelemente-Simulation**) zu replizieren.
- **VLSI Design**: Entwurf von ICs.
- Die Simulation des Verhaltens einer Schaltung **vor** der Herstellung eines ersten Prototyps erhöht die Effizienz des VLSI Designprozesses erheblich, Fehler können schnell korrigiert werden.
- Insbesondere für ICs ist
 - die Herstellung von Photomasken für Prototypen teuer,
 - das Testen des Verhaltens interner Signale extrem schwierig.

Daher ist VLSI Design heutzutage ohne numerische Simulation nicht mehr denkbar!

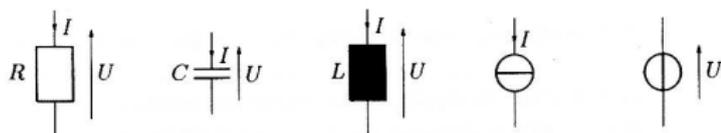
(Elektronische) Schaltkreissimulation

- Verwendung mathematischer Modelle, um das Verhalten eines elektronischen Schaltkreises oder Bauteils (↔ **Bauelemente-Simulation**) zu replizieren.
- **VLSI Design**: Entwurf von ICs.
- Die Simulation des Verhaltens einer Schaltung **vor** der Herstellung eines ersten Prototyps erhöht die Effizienz des VLSI Designprozesses erheblich, Fehler können schnell korrigiert werden.
- Insbesondere für ICs ist
 - die Herstellung von Photomasken für Prototypen teuer,
 - das Testen des Verhaltens interner Signale extrem schwierig.

Daher ist VLSI Design heutzutage ohne numerische Simulation nicht mehr denkbar!

Modellierung mit **modifizierter Knotenanalyse (MNA)**, basierend auf

Bauelementgleichungen (konzentrierte Elemente)



Linearer Widerstand: $I = U/R = G \cdot U$ $G = \text{Leitwert}$

Kapazität: $I = C \cdot \dot{U}$ $\dot{U} := \frac{d}{dt} U(t)$

Induktivität: $U = L \cdot \dot{I}$

unabh. Stromquelle: $I = s_1(t)$

unabh. Spannungsquelle: $U = s_1(t)$

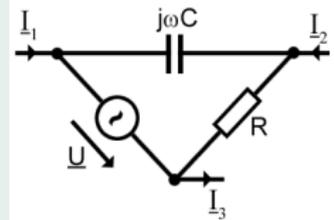
Modellierung mit **modifizierter Knotenanalyse (MNA)**, basierend auf

Kirchhoffschen Gesetzen

1. Knotenregel:

Die Summe aller Ströme in einem Knotenpunkt ist null,

$$\sum_{k=1}^N I_k = 0.$$



Quelle: http://de.wikipedia.org/wiki/Kirchhoffsche_Regeln

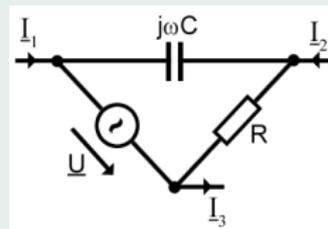
Modellierung mit **modifizierter Knotenanalyse (MNA)**, basierend auf

Kirchhoffschen Gesetzen

1. Knotenregel:

Die Summe aller Ströme in einem Knotenpunkt ist null,

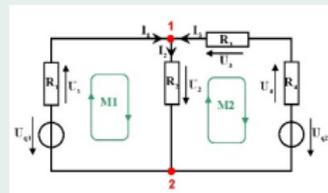
$$\sum_{k=1}^N I_k = 0.$$



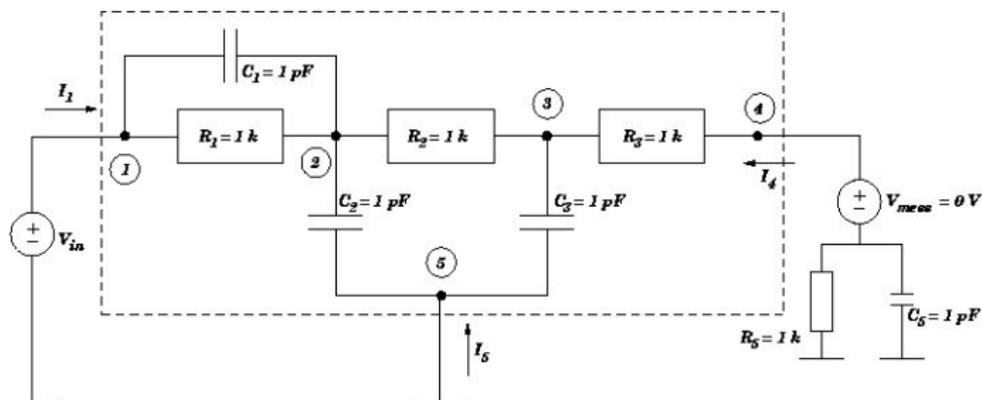
2. Maschenregel:

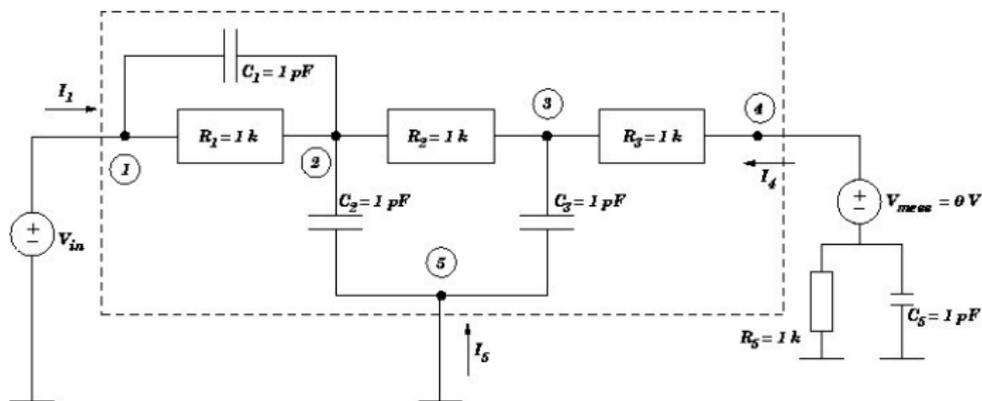
Alle Teilspannungen einer Masche in einem elektrischen Netzwerk addieren sich zu null,

$$\sum_{k=1}^N U_k = 0.$$



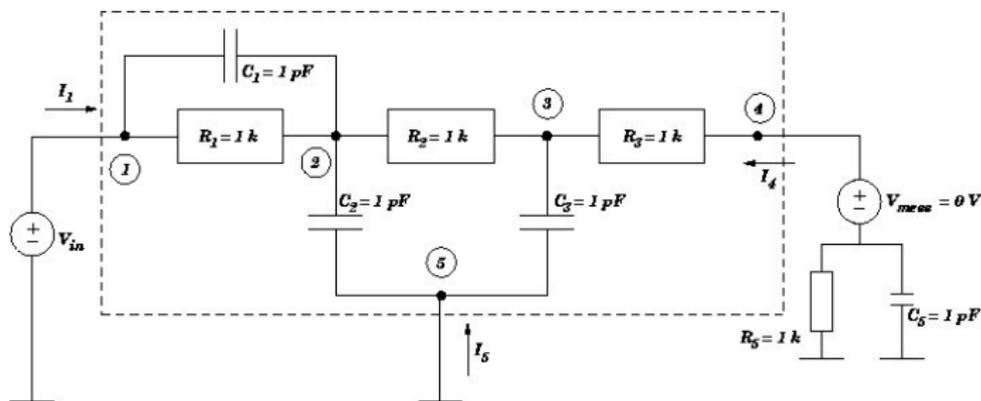
Quelle: http://de.wikipedia.org/wiki/Kirchhoffsche_Regeln





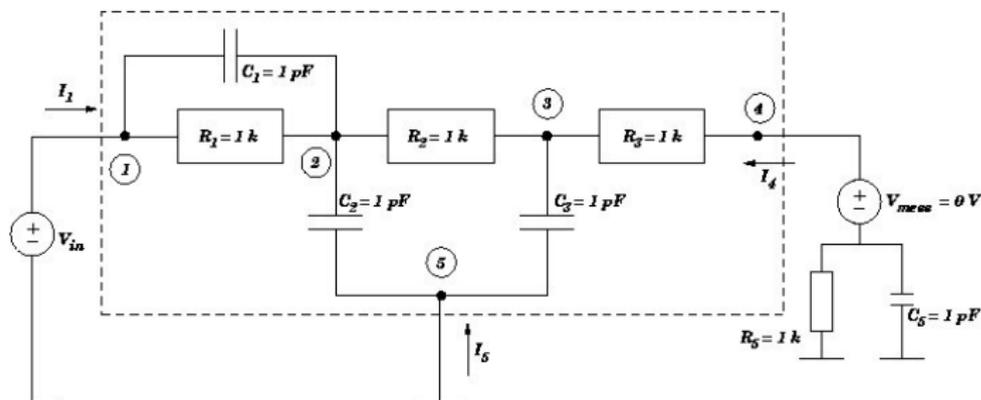
MNA \rightarrow

Knoten 1: $I_1 =$



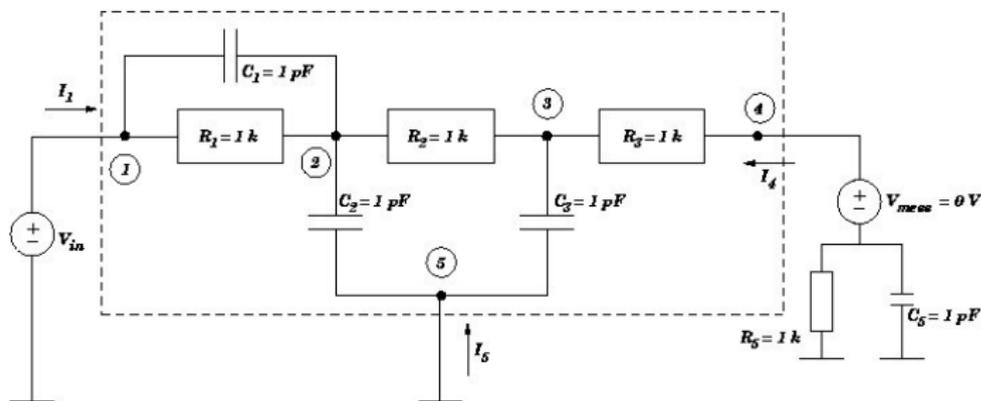
MNA \rightarrow

$$\text{Knoten 1: } I_1 = G_1(U_1 - U_2)$$



MNA →

$$\text{Knoten 1: } I_1 = G_1(U_1 - U_2) + C_1(\dot{U}_1 - \dot{U}_2)$$



MNA \rightarrow

$$\text{Knoten 1: } I_1 = G_1(U_1 - U_2) + C_1(\dot{U}_1 - \dot{U}_2)$$

$$\text{Knoten 2: } 0 = G_1(U_2 - U_1) + G_2(U_2 - U_3) + C_1(\dot{U}_2 - \dot{U}_1) + C_2(\dot{U}_2 - \dot{U}_5)$$

$$\text{Knoten 3: } 0 = G_2(U_3 - U_2) + G_3(U_3 - U_4) + C_3(\dot{U}_3 - \dot{U}_5)$$

$$\text{Knoten 4: } I_4 = G_3(U_4 - U_3)$$

$$\text{Knoten 5: } I_5 = C_2(\dot{U}_5 - \dot{U}_2) + C_3(\dot{U}_5 - \dot{U}_3)$$

MNA ergibt differentiell-algebraische Gleichung (DAE)

$$C\dot{x} = Gx + Bu, \quad y = L^T x,$$

mit Eingängen $u(t)$, Zuständen $x(t)$, Ausgängen $y(t)$:

$$u = \begin{bmatrix} I_1 \\ I_4 \\ I_5 \end{bmatrix}, \quad x = [U_1 \quad U_2 \quad U_3 \quad U_4 \quad U_5]^T, \quad y = \begin{bmatrix} U_1 \\ U_4 \\ U_5 \end{bmatrix}$$

und

$$C = \begin{bmatrix} C_1 & -C_1 & 0 & 0 & 0 \\ -C_1 & C_1 + C_2 & 0 & 0 & -C_2 \\ 0 & 0 & C_3 & 0 & -C_3 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & -C_2 & -C_3 & 0 & C_2 + C_3 \end{bmatrix}, \quad B = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix},$$

$$G = \begin{bmatrix} G_1 & -G_1 & 0 & 0 & 0 \\ -G_1 & G_1 + G_2 & -G_2 & 0 & 0 \\ 0 & -G_2 & G_2 + G_3 & -G_3 & 0 \\ 0 & 0 & -G_3 & G_3 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \quad L = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

- Gesamtmodelle moderner ICs lassen sich nicht mehr komplett aufstellen/simulieren, Teilschaltungen bestehen aus wenigen Hundert bis Millionen Gleichungen.
- Numerische Simulation wird ermöglicht durch Kopplung von **kompakten Modellen** der Teilschaltkreise.
- Modellierung nichtlinearer Bauelemente (Dioden, Transistoren) führt zu nichtlinearen DAE-Systemen.
- Kritische Bauelemente können nicht mehr ohne weiteres durch Ersatzschaltungen **konzentrierter Elemente** modelliert werden.
- Bauelemente-Simulation benötigt partielle Differentialgleichungen (PDEs, **Drift-Diffusions-Modell**).
- Kopplung von Schaltkreis mit nicht-konzentriertem Bauelement führt auf partiell differentiell-algebraische Gleichungen (PDAEs).

Idee:

Ersetze dynamisches System durch **Ersatzschaltbild**.
 Ersatzschaltbild = elektrisches Netzwerk.

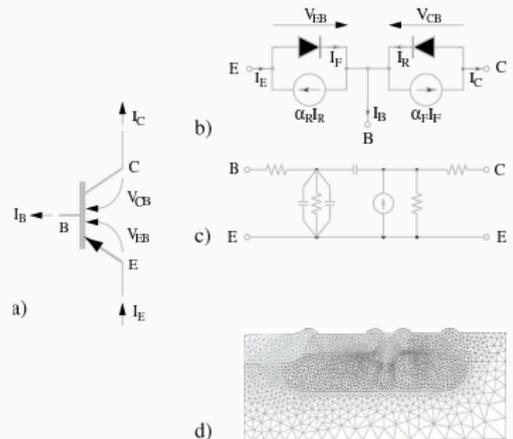
Beispiel: PNP Transistor

Vollständige Beschreibung erfordert Kombination von Transportgleichung, Maxwellgleichung, Poisson-Boltzmann-Gleichung.

Ersatzschaltung: Modellierung durch 2 Dioden.

Weitere Vereinfachung für Analyse kleiner Signale im Wesentlichen durch einfaches RLC Netzwerk.

Problem: Weder automatische Modellgenerierung noch Fehlerabschätzungen.



Idee:

Ersetze dynamisches System durch **Ersatzschaltbild**.
 Ersatzschaltbild = elektrisches Netzwerk.

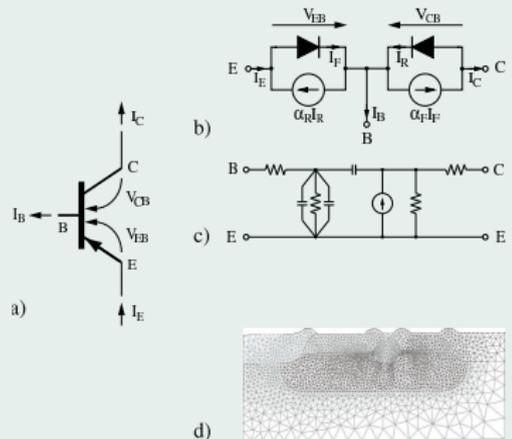
Beispiel: PNP Transistor

Vollständige Beschreibung erfordert Kombination von Transportgleichung, Maxwellgleichung, Poisson-Boltzmann-Gleichung.

Ersatzschaltung: Modellierung durch 2 Dioden.

Weitere Vereinfachung für Analyse kleiner Signale im Wesentlichen durch einfaches RLC Netzwerk.

Problem: Weder automatische Modellgenerierung noch Fehlerabschätzungen.



Idee:

Ersetze dynamisches System durch **Ersatzschaltbild**.
 Ersatzschaltbild = elektrisches Netzwerk.

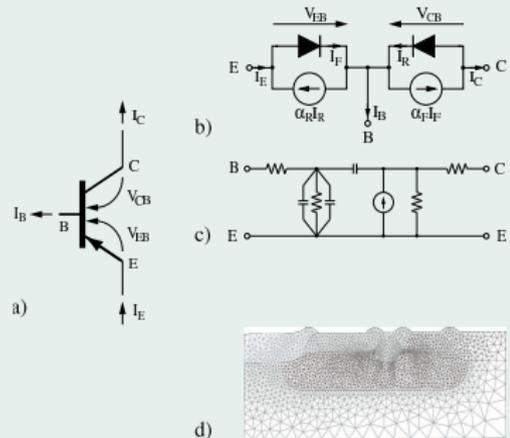
Beispiel: PNP Transistor

Vollständige Beschreibung erfordert Kombination von Transportgleichung, Maxwellgleichung, Poisson-Boltzmann-Gleichung.

Ersatzschaltung: Modellierung durch 2 Dioden.

Weitere Vereinfachung für Analyse kleiner Signale im Wesentlichen durch einfaches RLC Netzwerk.

Problem: Weder automatische Modellgenerierung noch Fehlerabschätzungen.



Fortschreitende Miniaturisierung — Konsequenzen

- **Moore's Law** \Rightarrow zur Beschreibung der Schaltkreise werden immer mehr Gleichungen nötig, zur Modellierung der Netzwerk-Topologie (Kirchhoffsche Gesetze) einerseits und charakteristische Gleichungen der Bauelemente andererseits.

Fortschreitende Miniaturisierung — Konsequenzen

- **Moore's Law** \Rightarrow zur Beschreibung der Schaltkreise werden immer mehr Gleichungen nötig, zur Modellierung der Netzwerk-Topologie (Kirchhoffsche Gesetze) einerseits und charakteristische Gleichungen der Bauelemente andererseits.
- Verifizierung eines VLSI Designs benötigt eine Vielzahl von Simulationsläufen für verschiedene Eingangssignale.

Fortschreitende Miniaturisierung — Konsequenzen

- **Moore's Law** \Rightarrow zur Beschreibung der Schaltkreise werden immer mehr Gleichungen nötig, zur Modellierung der Netzwerk-Topologie (Kirchhoffsche Gesetze) einerseits und charakteristische Gleichungen der Bauelemente andererseits.
- Verifizierung eines VLSI Designs benötigt eine Vielzahl von Simulationsläufen für verschiedene Eingangssignale.
- Mit steigender Integrationsdichte und durch **mehrlagiges Layout** nehmen **parasitäre Effekte** (Signalstörungen durch Überlagerung elektromagnetischer Felder und thermische Effekte) zu und müssen modelliert werden.

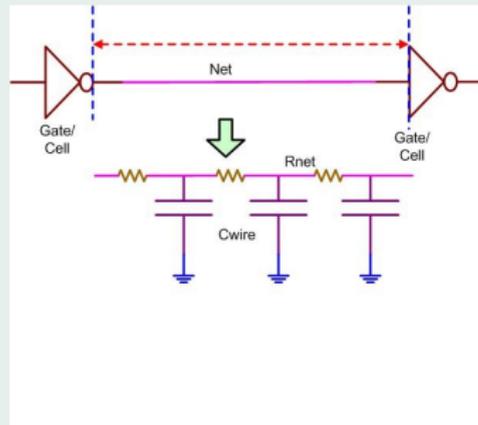
Fortschreitende Miniaturisierung — Konsequenzen

- **Moore's Law** \Rightarrow zur Beschreibung der Schaltkreise werden immer mehr Gleichungen nötig, zur Modellierung der Netzwerk-Topologie (Kirchhoffsche Gesetze) einerseits und charakteristische Gleichungen der Bauelemente andererseits.
- Verifizierung eines VLSI Designs benötigt eine Vielzahl von Simulationsläufen für verschiedene Eingangssignale.
- Mit steigender Integrationsdichte und durch mehrlagiges Layout nehmen parasitäre Effekte (Signalstörungen durch Überlagerung elektromagnetischer Felder und thermische Effekte) zu und müssen modelliert werden.

Modellierung der Leiterbahnen (**Interconnect**) durch RLC Netzwerke, thermische Effekte oder kritische Bauteile durch örtlich diskretisierte partielle Differentialgleichungen!

Interconnect Modellierung

Modellierung der Leiterbahnen (**Interconnect**) durch RLC Netzwerke, thermische Effekte oder kritische Bauteile durch örtlich diskretisierte partielle Differentialgleichungen!



Fortschreitende Miniaturisierung — Konsequenzen

- **Moore's Law** \Rightarrow zur Beschreibung der Schaltkreise werden immer mehr Gleichungen nötig, zur Modellierung der Netzwerk-Topologie (Kirchhoffsche Gesetze) einerseits und charakteristische Gleichungen der Bauelemente andererseits.
- Verifizierung eines VLSI Designs benötigt eine Vielzahl von Simulationsläufen für verschiedene Eingangssignale.
- Mit steigender Integrationsdichte und durch mehrlagiges Layout nehmen parasitäre Effekte (Signalstörungen durch Überlagerung elektromagnetischer Felder und thermische Effekte) zu und müssen modelliert werden.
- **Aktuelle Computergeneration liefert nicht die Rechenleistung, die nötig wäre, das Verhalten ihrer Nachfolger numerisch zu simulieren!**

Fazit

Methoden zur Reduktion der Komplexität der mathematischen Schaltkreismodelle für CAD von ICs bzw. VLSI Design unumgänglich!



System- bzw. Modellreduktion

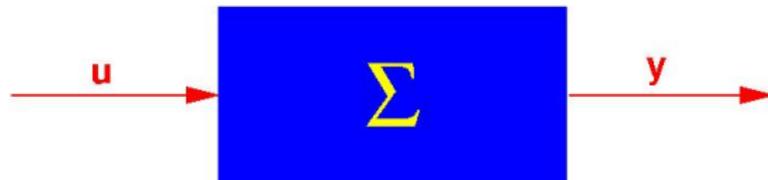
Dynamische Systeme/DAEs

$$\Sigma : \begin{cases} \frac{d}{dt} q(x(t)) = f(x(t), u(t)), & x(t_0) = x_0, \\ y(t) = g(x(t), u(t)), \end{cases}$$

mit

- **Zuständen** $x(t) \in \mathbb{R}^n$,
- **Eingängen** $u(t) \in \mathbb{R}^m$,
- **Ausgängen** $y(t) \in \mathbb{R}^p$.

↪ System differentiell-algebraischer Gleichungen (DAEs).



Originalsystem

$$\Sigma : \begin{cases} \frac{d}{dt} q(x(t)) = f(x(t), u(t)), \\ y(t) = g(x(t), u(t)). \end{cases}$$

- Zustände $x(t) \in \mathbb{R}^n$,
- Eingänge $u(t) \in \mathbb{R}^m$,
- Ausgänge $y(t) \in \mathbb{R}^p$.



Reduziertes System

$$\hat{\Sigma} : \begin{cases} \frac{d}{dt} \hat{q}(x(t)) = \hat{f}(t, \hat{x}(t), u(t)), \\ \hat{y}(t) = \hat{g}(t, \hat{x}(t), u(t)). \end{cases}$$

- Zustände $\hat{x}(t) \in \mathbb{R}^r$, $r \ll n$
- Eingänge $u(t) \in \mathbb{R}^m$,
- Ausgänge $\hat{y}(t) \in \mathbb{R}^p$.



Ziel:

$\|y - \hat{y}\| < \text{tol} \cdot \|u\|$ für alle zulässigen Eingangssignale.

Originalsystem

$$\Sigma : \begin{cases} \frac{d}{dt} q(x(t)) = f(x(t), u(t)), \\ y(t) = g(x(t), u(t)). \end{cases}$$

- Zustände $x(t) \in \mathbb{R}^n$,
- Eingänge $u(t) \in \mathbb{R}^m$,
- Ausgänge $y(t) \in \mathbb{R}^p$.



Reduziertes System

$$\hat{\Sigma} : \begin{cases} \frac{d}{dt} \hat{q}(x(t)) = \hat{f}(t, \hat{x}(t), u(t)), \\ \hat{y}(t) = \hat{g}(t, \hat{x}(t), u(t)). \end{cases}$$

- Zustände $\hat{x}(t) \in \mathbb{R}^r$, $r \ll n$
- Eingänge $u(t) \in \mathbb{R}^m$,
- Ausgänge $\hat{y}(t) \in \mathbb{R}^p$.



Ziel:

$$\|y - \hat{y}\| < \text{tol} \cdot \|u\| \text{ für alle zulässigen Eingangssignale.}$$

- **Automatische Berechnung kompakter Modelle.**
- Fehler in den Ausgangssignalen des reduzierten Modells dürfen eine vorgegebene Fehlertoleranz nicht überschreiten.
Benötige berechenbare Fehlerabschätzungen!
- Erhalte physikalische Eigenschaften:
 - Stabilität
 - Passivität, d.h. es wird von Modellen passiver Bauteile keine Energie erzeugt.
- Reduzierte Modelle sollten erheblich schnellere Simulation erlauben.
- Reduzierte Modelle sollten wieder als Schaltkreis realisierbar sein.

- Automatische Berechnung kompakter Modelle.
- Fehler in den Ausgangssignalen des reduzierten Modells dürfen eine vorgegebene Fehlertoleranz nicht überschreiten.
Benötige berechenbare Fehlerabschätzungen!
- Erhalte physikalische Eigenschaften:
 - Stabilität
 - Passivität, d.h. es wird von Modellen passiver Bauteile keine Energie erzeugt.
- Reduzierte Modelle sollten erheblich schnellere Simulation erlauben.
- Reduzierte Modelle sollten wieder als Schaltkreis realisierbar sein.

- Automatische Berechnung kompakter Modelle.
- Fehler in den Ausgangssignalen des reduzierten Modells dürfen eine vorgegebene Fehlertoleranz nicht überschreiten.
Benötige berechenbare Fehlerabschätzungen!
- **Erhalte physikalische Eigenschaften:**
 - Stabilität
 - Passivität, d.h. es wird von Modellen passiver Bauteile keine Energie erzeugt.
- Reduzierte Modelle sollten erheblich schnellere Simulation erlauben.
- Reduzierte Modelle sollten wieder als Schaltkreis realisierbar sein.

- Automatische Berechnung kompakter Modelle.
- Fehler in den Ausgangssignalen des reduzierten Modells dürfen eine vorgegebene Fehlertoleranz nicht überschreiten.
Benötige berechenbare Fehlerabschätzungen!
- Erhalte physikalische Eigenschaften:
 - **Stabilität**
 - Passivität, d.h. es wird von Modellen passiver Bauteile keine Energie erzeugt.
- Reduzierte Modelle sollten erheblich schnellere Simulation erlauben.
- Reduzierte Modelle sollten wieder als Schaltkreis realisierbar sein.

- Automatische Berechnung kompakter Modelle.
- Fehler in den Ausgangssignalen des reduzierten Modells dürfen eine vorgegebene Fehlertoleranz nicht überschreiten.
Benötige berechenbare Fehlerabschätzungen!
- Erhalte physikalische Eigenschaften:
 - Stabilität
 - Passivität, d.h. es wird von Modellen passiver Bauteile keine Energie erzeugt.
- Reduzierte Modelle sollten erheblich schnellere Simulation erlauben.
- Reduzierte Modelle sollten wieder als Schaltkreis realisierbar sein.

- Automatische Berechnung kompakter Modelle.
- Fehler in den Ausgangssignalen des reduzierten Modells dürfen eine vorgegebene Fehlertoleranz nicht überschreiten.
Benötige berechenbare Fehlerabschätzungen!
- Erhalte physikalische Eigenschaften:
 - Stabilität
 - Passivität, d.h. es wird von Modellen passiver Bauteile keine Energie erzeugt.
- **Reduzierte Modelle sollten erheblich schnellere Simulation erlauben.**
- Reduzierte Modelle sollten wieder als Schaltkreis realisierbar sein.

- Automatische Berechnung kompakter Modelle.
- Fehler in den Ausgangssignalen des reduzierten Modells dürfen eine vorgegebene Fehlertoleranz nicht überschreiten.
Benötige berechenbare Fehlerabschätzungen!
- Erhalte physikalische Eigenschaften:
 - Stabilität
 - Passivität, d.h. es wird von Modellen passiver Bauteile keine Energie erzeugt.
- Reduzierte Modelle sollten erheblich schnellere Simulation erlauben.
- **Reduzierte Modelle sollten wieder als Schaltkreis realisierbar sein.**

Mikro zu Nano

Peter Benner

Nanoelektronik

Modellierung

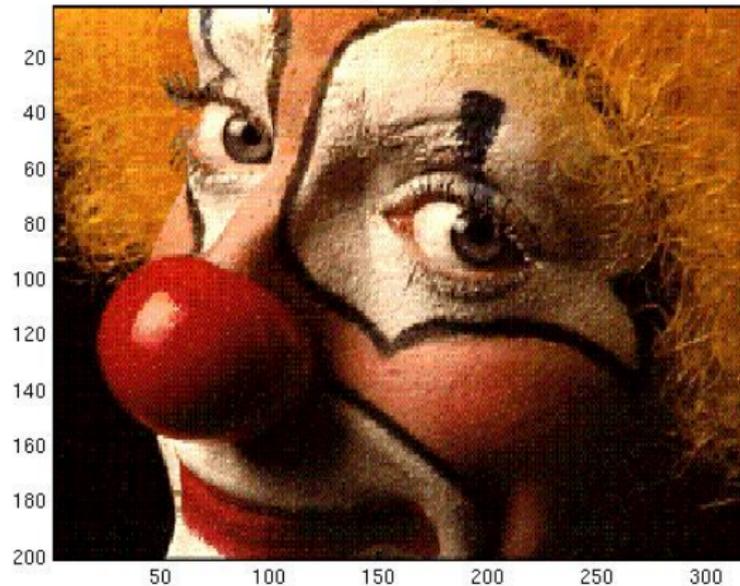
Modellreduktion

Ziele

Grundideen für
Algorithmen

Fazit

Original image



Mikro zu Nano

Peter Benner

Nanoelektronik

Modellierung

Modellreduktion

Ziele

Grundideen für
Algorithmen

Fazit

Original image

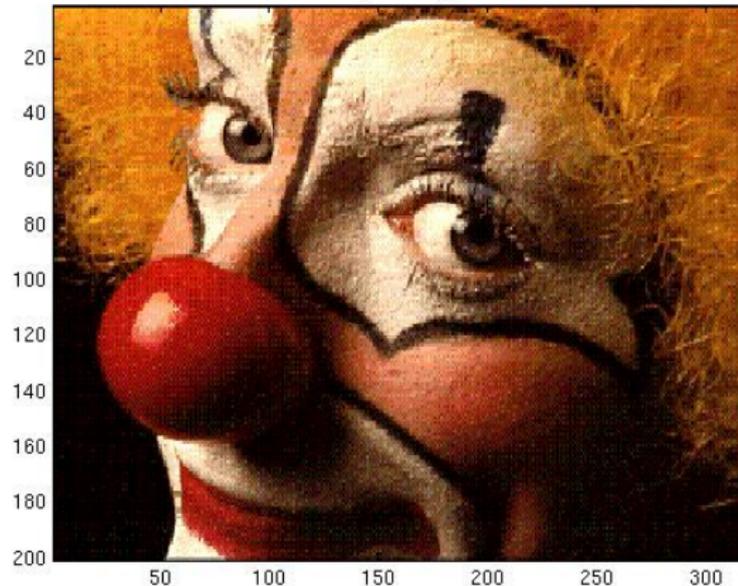


Bild = $x \cdot y$ Pixel = x Spalten (**Vektoren**) mit y Einträgen (Farbwerte)

Mikro zu Nano

Peter Benner

Nanoelektronik

Modellierung

Modellreduktion

Ziele

Grundideen für
Algorithmen

Fazit

Original image

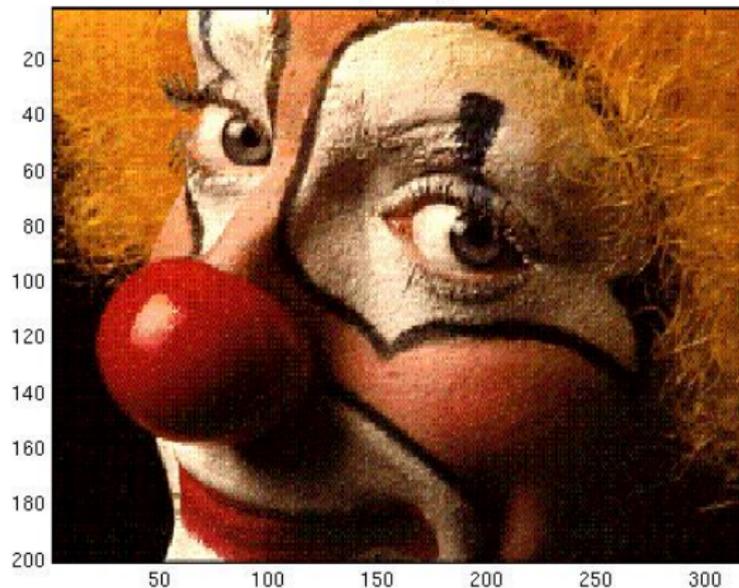


Bild = $x \cdot y$ Pixel = x Spalten (Vektoren) mit y Einträgen (Farbwerte)
 = $y \times x$ Matrix.

Satz: (Schmidt-Mirsky/Eckart-Young)

Beste Rang- r Approximation an $X \in \mathbb{R}^{y \times x}$:

$$\hat{X} = \sum_{j=1}^r \sigma_j u_j v_j^T,$$

wobei $X = U\Sigma V^T$ die **Singulärwertzerlegung (SVD)** von X ist und $U = [u_1, \dots]$, $V = [v_1, \dots]$, $\Sigma = \text{diag}(\sigma_1, \dots)$.

Der Approximationsfehler ist $\|X - \hat{X}\|_2 = \sigma_{r+1}$.

Satz: (Schmidt-Mirsky/Eckart-Young)

Beste Rang- r Approximation an $X \in \mathbb{R}^{y \times x}$:

$$\hat{X} = \sum_{j=1}^r \sigma_j u_j v_j^T,$$

wobei $X = U \Sigma V^T$ die **Singulärwertzerlegung (SVD)** von X ist und $U = [u_1, \dots]$, $V = [v_1, \dots]$, $\Sigma = \text{diag}(\sigma_1, \dots)$.

Der Approximationsfehler ist $\|X - \hat{X}\|_2 = \sigma_{r+1}$.

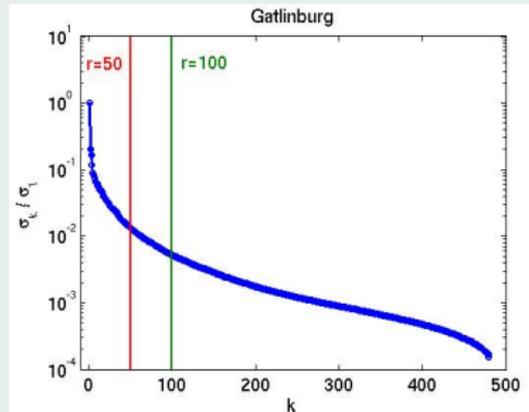
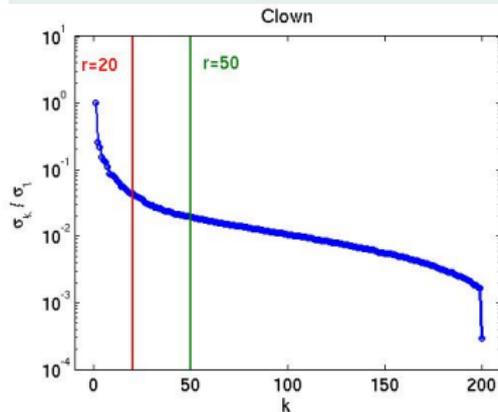
Idea für Dimensionsreduktion

Statt X speichere $u_1, \dots, u_r, \sigma_1 v_1, \dots, \sigma_r v_r$.

\Rightarrow benötigter Speicherplatz = $r \cdot (x + y)$ statt $x \cdot y$.

Datenkomprimierung mit SVD funktioniert, wenn die meisten Singulärwerte sehr klein sind.

Verhalten der Singulärwerte



Mikro zu Nano

Peter Benner

Nanoelektronik

Modellierung

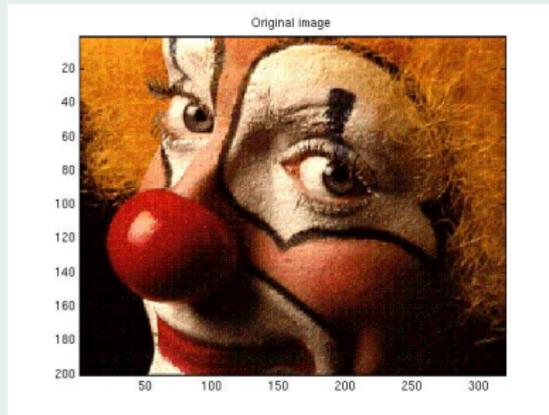
Modellreduktion

Ziele

Grundideen für
Algorithmen

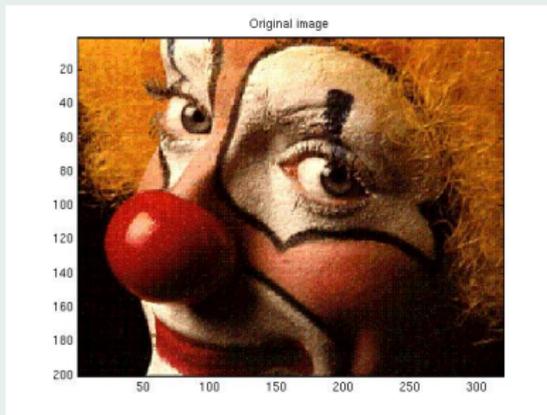
Fazit

Beispiel: Clown



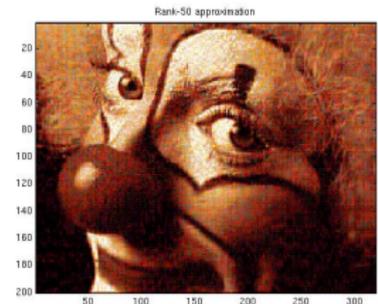
320×200 Pixel
 $\rightsquigarrow \approx 256$ Kb

Beispiel: Clown

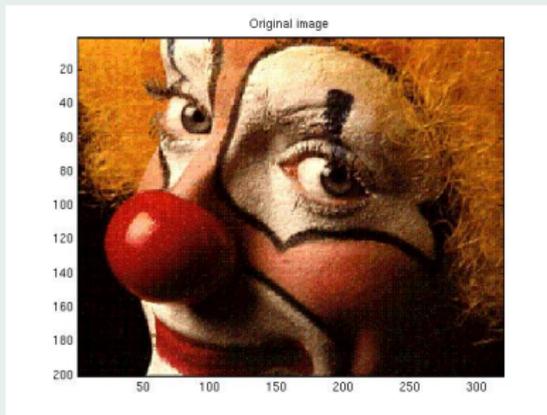


320×200 Pixel
 $\rightsquigarrow \approx 256$ Kb

■ $r = 50, \approx 104$ Kb

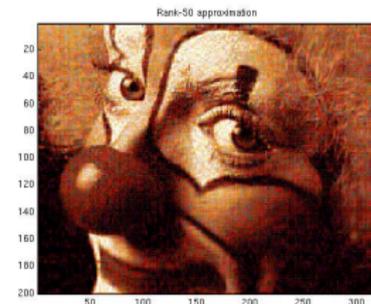


Beispiel: Clown

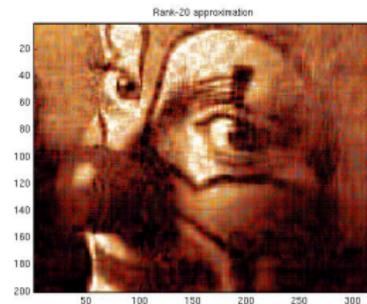


320×200 Pixel
 $\rightsquigarrow \approx 256$ Kb

■ $r = 50, \approx 104$ Kb



■ $r = 20, \approx 42$ Kb





Mikro zu Nano

Peter Benner

Nanoelektronik

Modellierung

Modellreduktion

Ziele

Grundideen für
Algorithmen

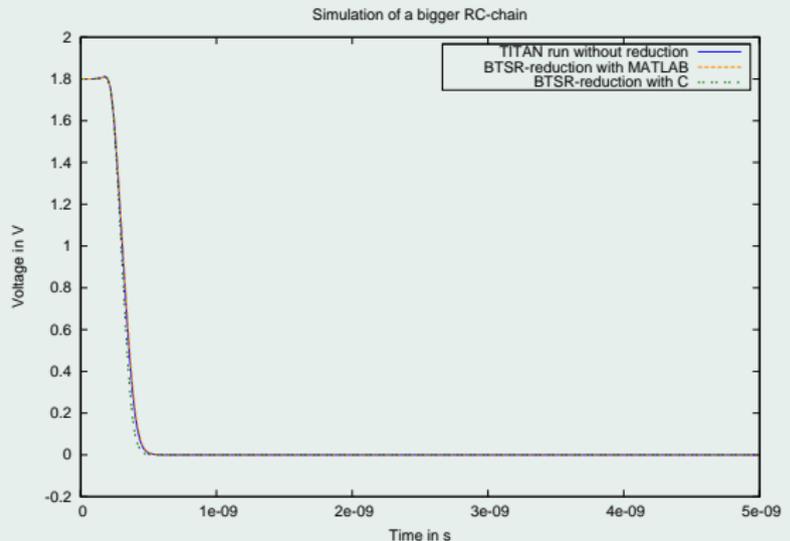
Fazit

Ähnliche Ideen lassen sich für dynamische Systeme verwirklichen:
Bestimme charakteristische Größen und erhalte nur die wesentliche Information.

Beispiel 1

297 Widerstände, 268 Kapazitäten, 4 Spannungsquellen, 8 MOSFETs.
Lineare Teilschaltung der Ordnung $n = 297$ extrahiert, reduziert auf
Ordnung $r = 31$.

Simulationsergebnisse (TITAN):



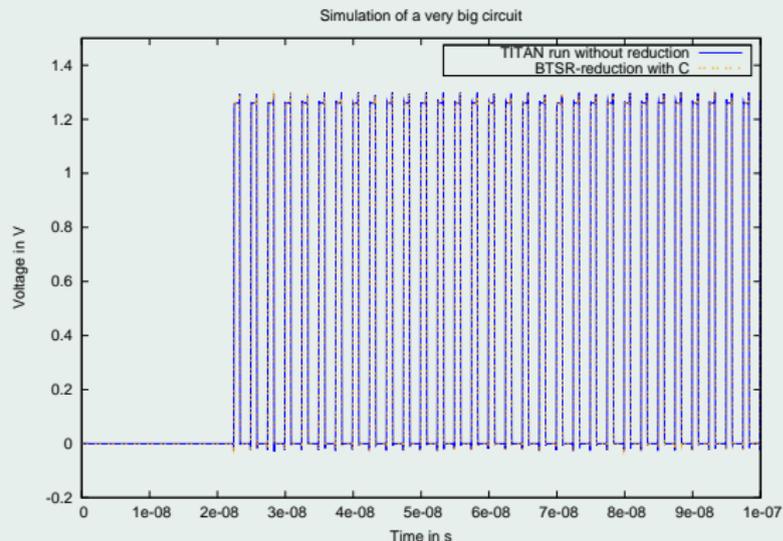
Beispiel 2: industrieller Schaltkreis von Qimonda

14.677 Widerstände, 15.404 Kapazitäten,

14 Spannungsquellen, 4.800 MOSFETs.

14 lineare Teilschaltkreise variierender Ordnung extrahiert und reduziert.

Simulationsergebnisse (mit TITAN):



Fazit des Vortrags

Der Übergang von der Mikro- zur Nanoelektronik stellt immer größere Herausforderungen an die **Angewandte Mathematik**, insbesondere die **Numerik**.

Oder . . .

Bevor Computer das Zählen lernen, benötigen sie jede Menge Mathematik!

⇒ **MATH inside**, to be continued . . .

Fazit des Vortrags

Der Übergang von der Mikro- zur Nanoelektronik stellt immer größere Herausforderungen an die **Angewandte Mathematik**, insbesondere die **Numerik**.

Oder . . .

Bevor Computer das Zählen lernen, benötigen sie jede Menge Mathematik!

⇒ **MATH inside**, to be continued . . .

Fazit des Vortrags

Der Übergang von der Mikro- zur Nanoelektronik stellt immer größere Herausforderungen an die **Angewandte Mathematik**, insbesondere die **Numerik**.

Oder . . .

Bevor Computer das Zählen lernen, benötigen sie jede Menge Mathematik!

⇒ **MATH inside**, to be continued . . .