

# Einführung in die Technische Informatik

## VLSI-Systementwurf

Modellierung  
Simulation

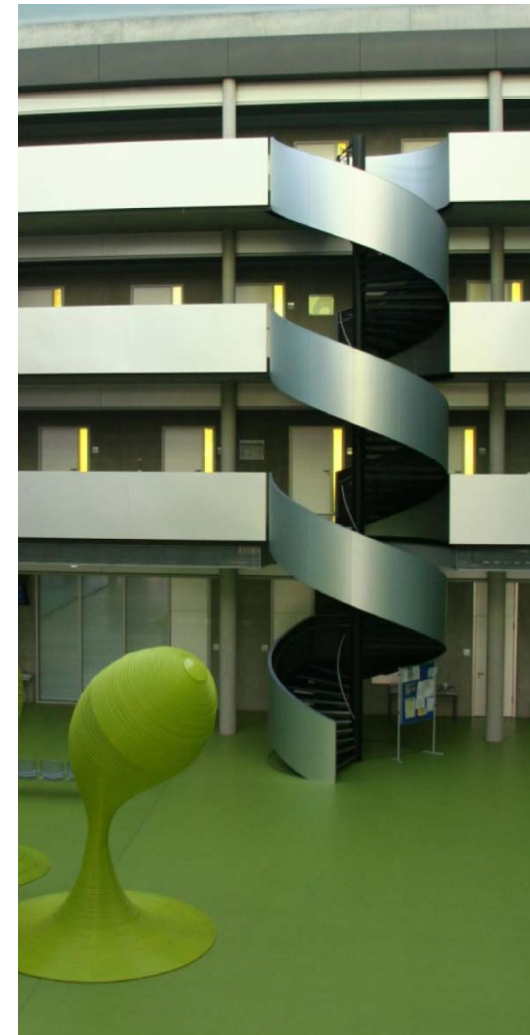
*Rainer G. Spallek*  
*Martin Zabel*

TU Dresden, 07.08.2013



## Gliederung

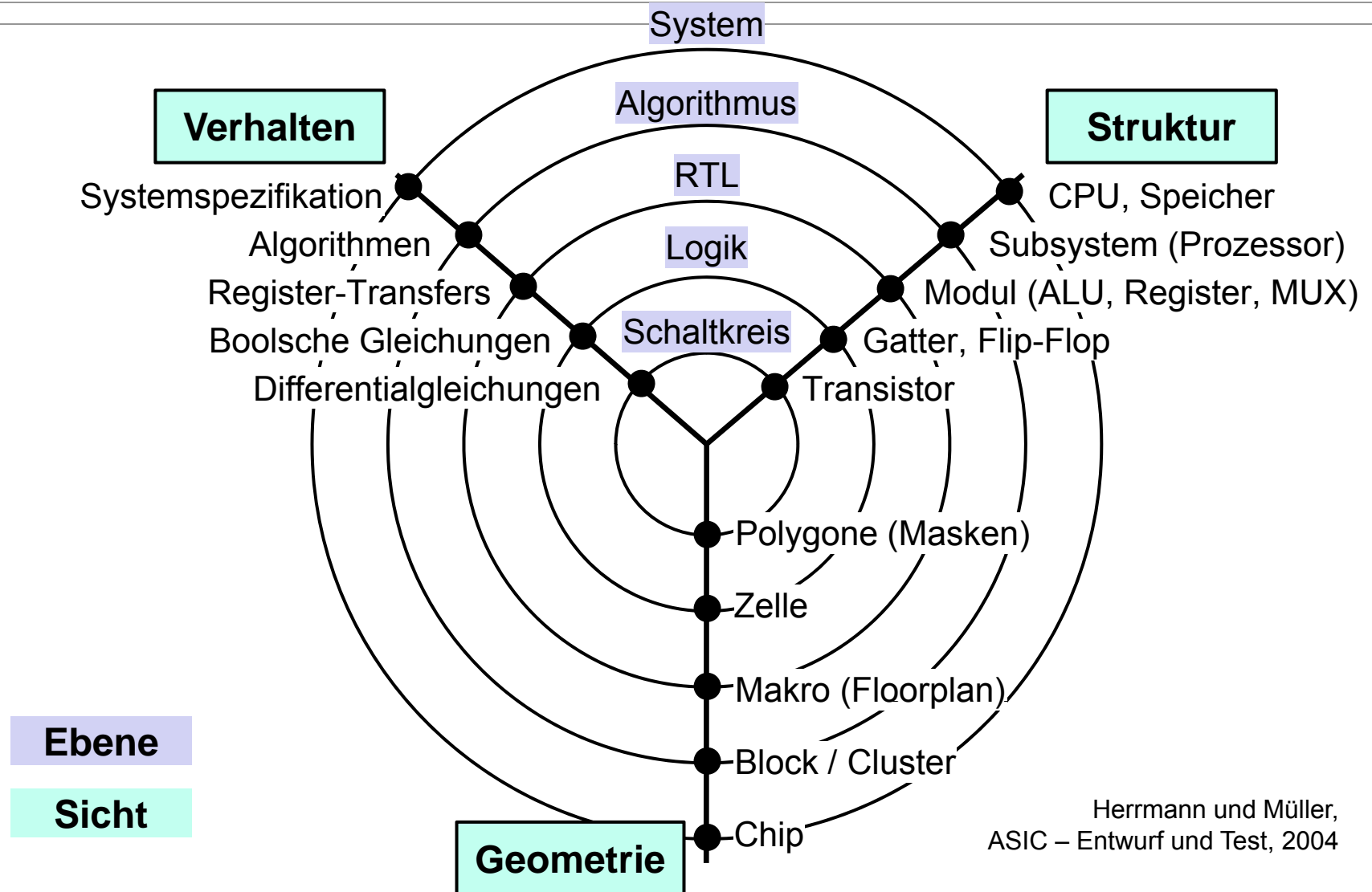
- 1 Wiederholung
- 2 Begriffe
- 3 M&S auf verschiedenen Ebenen
- 4 Modellvalidierung
- 5 Parameterextraktion
- 6 Effizienzanalysen und Optimierung
- 7 Zusammenfassung



# 1 Wiederholung

- Y-Diagramm nach Gajski
- Entwurfsablauf

Y-Diagramm nach Gajski



## Entwurfsablauf

**Allgemein:** Transformation einer Aufgabenstellung (Pflichtenheft) in einen fertigen Schaltkreis.

### Top-Down-Strategie:

- Systemebene → Schaltkreisebene.
- Vorteil: Parallele Entwicklung auf unteren Ebenen.
- Nachteil: Systemspezifikation zu Projektbeginn oft zu ungenau.

### Bottom-Up-Strategie:

- Analyse vorhandener Komponenten.
- Zusammensetzen von neuen Komponenten auf höherer Ebene im Sinne der Aufgabenstellung.
- Nachteil: Globales Ziel wird nicht immer erreicht.

### Meet-in-the-Middle

**Entwurfsschritt:**

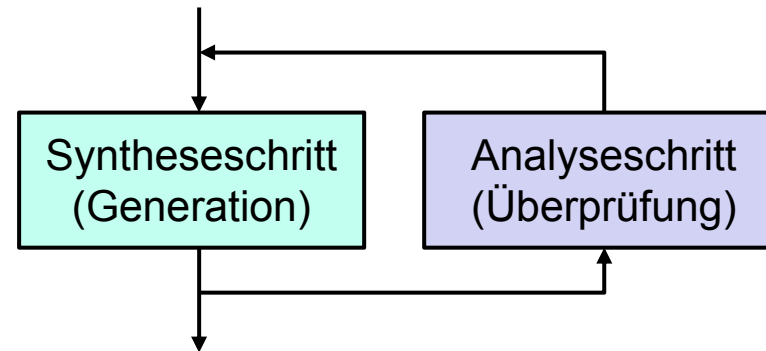
- Generierende Aktivität.
- Überprüfende Aktivität.

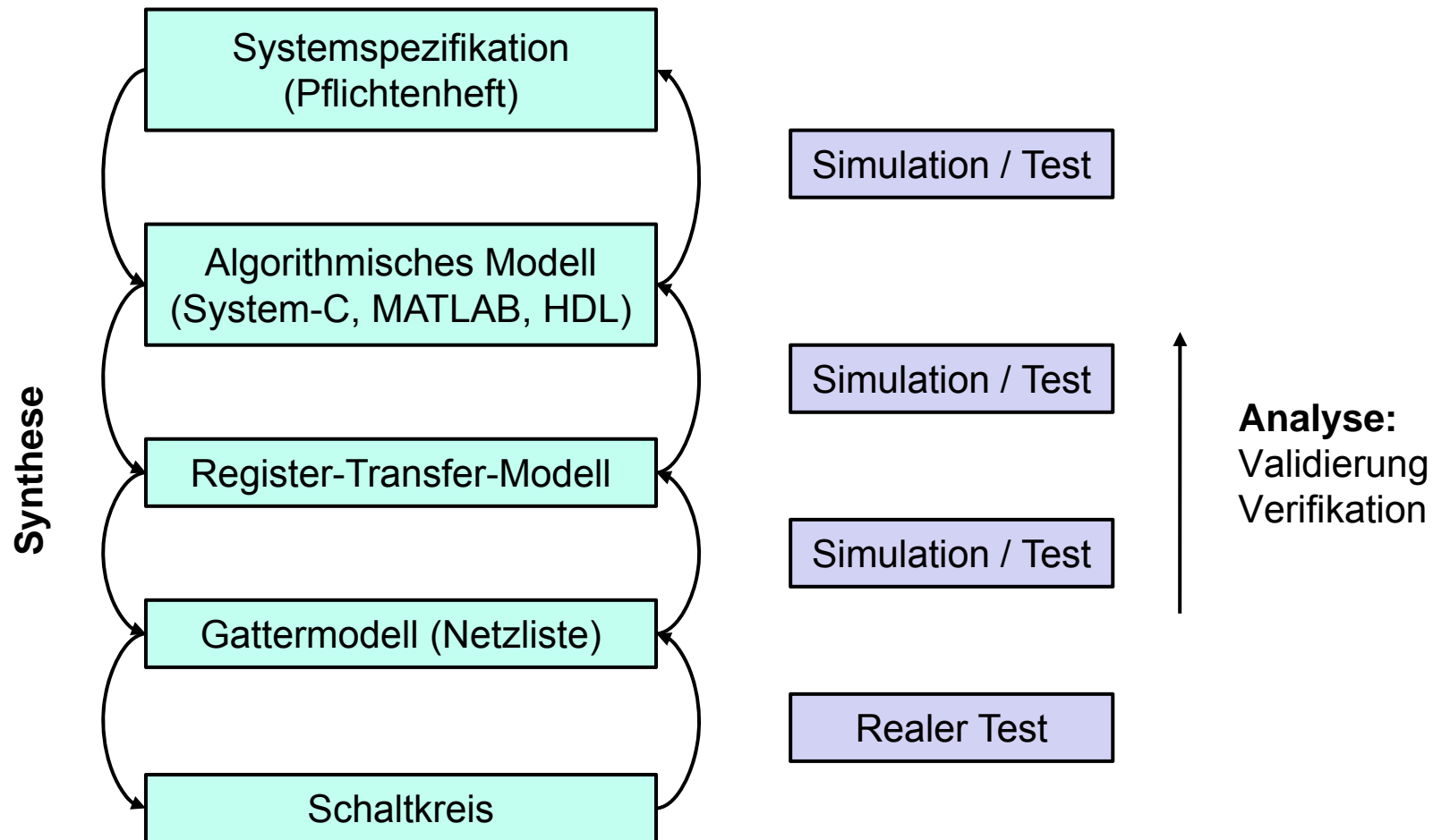
**Syntheseschritt:**

- Abbildung eines Entwurfsschrittes in Richtung auf das Entwurfsziel.
- Abstraktionsgrad sinkt, Detailliertheitsgrad steigt.
- Einbringung neuer Informationen.

**Analyseschritt:**

- Abbildung eines Entwurfsschrittes in umgekehrter Richtung zum Syntheseschritt.
- Gewinnung abstrakter Informationen durch Zusammenfassen und Generalisieren von Details (Extraktionsprozess).
- Beispiel: Validierung eines Syntheseschrittes.





## 2 Begriffe

### System:

- Beobachtungsgegenstand
- Zusammenschaltung mehrere Komponenten, die zusammen agieren und interagieren, um eine Aufgabe zu erfüllen.
- Umfang des Systems ist abhängig vom zu untersuchenden Aspekt.

### Modell:

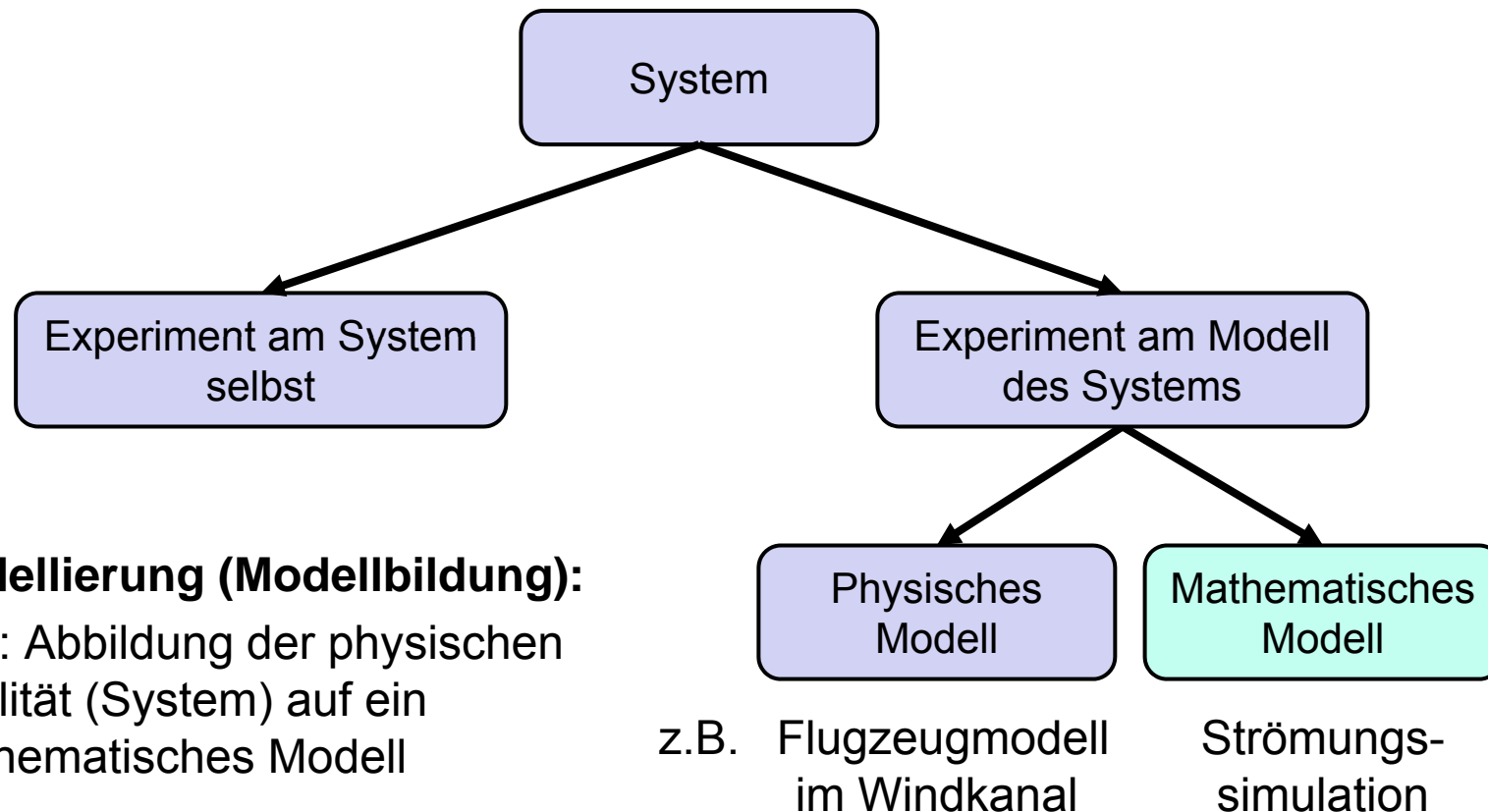
- Menge von Annahmen mit dem Ziel die Funktionsweise des Systems zu verstehen.
- Annahmen: i.d.R. in mathematischer oder logischer<sup>1</sup> Form  
→ mathematisches / formales Modell

<sup>1</sup>z.B. Aussagenlogik, Prädikatenlogik

A.M. Law: *Simulation, Modeling & Analysis*, Mc Graw Hill, 2007. – ISBN 978-007-125519-6



## Untersuchung eines Systems

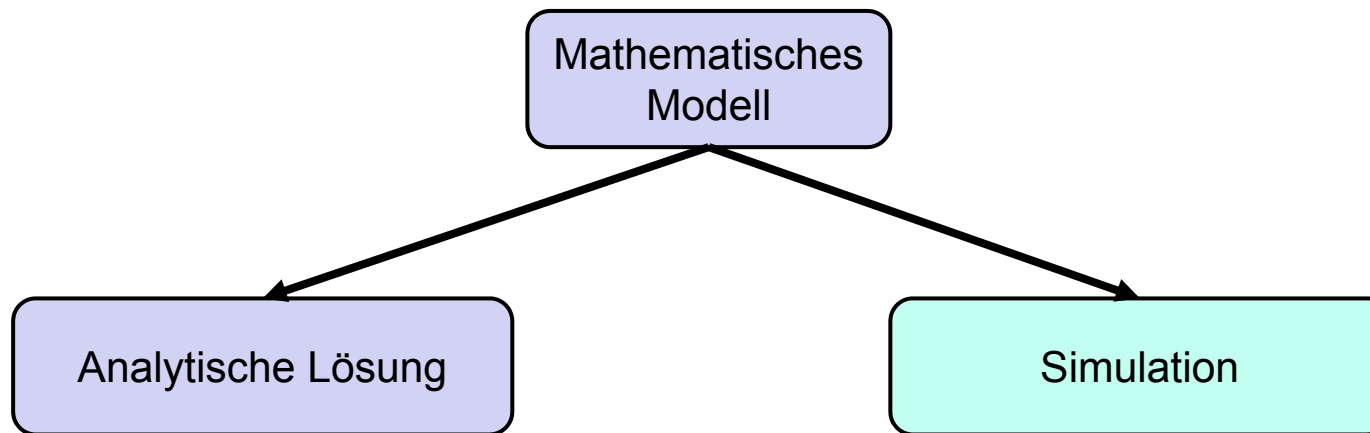


### Modellierung (Modellbildung):

Hier: Abbildung der physischen Realität (System) auf ein mathematisches Modell

A.M. Law: *Simulation, Modeling & Analysis*, Mc Graw Hill, 2007. – ISBN 978-007-125519-6

## Untersuchung am mathematischen Modell



Mathematische Verfahren  
(Algebra, Analysis,  
Wahrscheinlichkeitstheorie)  
→ Exakte Informationen

Numerische Verfahren  
(Iterative Berechnung des Modells)  
→ Näherungslösung

A.M. Law: *Simulation, Modeling & Analysis*, Mc Graw Hill, 2007. – ISBN 978-007-125519-6

## Simulation

### Definition in VDI-Richtlinie 3633:

„Simulation ist das Nachbilden eines Systems mit seinen dynamischen Prozessen in einem experimentierfähigen Modell, um zu Erkenntnissen zu gelangen, die auf die Wirklichkeit übertragbar sind.“

### Anwendung:

- Komplexe Modelle, für die analytische Lösung nicht praktikabel ist.
- In der Technischen Informatik z.B.:  
Netzwerksimulation, Logiksimulation, Fehlersimulation,  
Verlustleistungssimulation

**Simulationsmodell:** Modell, welches mittels Simulation untersucht wird.

## Klassifikation von Simulationsmodellen

### Statisch vs. dynamisch:

- Statisch: Betrachtung eines Zeitpunkts oder ohne zeitlichen Bezug
- Dynamisch: Betrachtung der Entwicklung über die Zeit

### Deterministisch vs. stochastisch:

- Deterministisch: Modell ohne Zufallsgrößen
- Stochastisch: Modell mit Zufallsgrößen, z.B. zur Abdeckung vieler verschiedener Eingaben (auch zu unterschiedlichen Zeitpunkten)

### Kontinuierlich vs. diskret:

- Zeitkontinuierlich: Zustandsänderung zu beliebigen Zeitpunkten (Differentialgleichungen)
- Zeitdiskret: Zustandsänderung nur zu diskreten Zeitpunkten (Differenzengleichungen)

A.M. Law: *Simulation, Modeling & Analysis*, Mc Graw Hill, 2007. – ISBN 978-007-125519-6

## Anwendung in Technischen Informatik

### **Ereignisorientierte Simulation** (engl.: discrete-event simulation)

- Simulationsmodell ist dynamisch, stochastisch und diskret.
- Schließt deterministische Modelle ein.
- Beispiel: Logiksimulation, Fehlersimulation

### **Kontinuierliche Simulation** (engl.: continuous simulation)

- Simulationsmodell ist dynamisch, deterministisch und kontinuierlich.
- Differentialgleichungen (DGL)
- Beispiel: Simulation elektrischer Netzwerke
- Analytische Lösung theoretisch möglich, wenn DGL-System lösbar.

## Modellierung in der Praxis

### Begrenzung des Modellierungsaufwandes durch Vereinfachung:

- System mit konzentrierten Parametern statt verteilten Parametern (z.B. nur Massepunkte, punktförmige Bauelemente)  
→ gewöhnliche statt partielle Differentialgleichungen
- Idealisierte Komponenten:
  - Physik: starre statt elastische Körper, masselose statt massebehaftete Federn
  - Technische Informatik: ideale Bauelemente, ideales Schaltverhalten

**Ziel:** Modell soll nur so genau wie nötig sein!

Beispiel: Charakterisierung von Bauelementen mittels Kennlinien.

H. Scherf: *Modellbildung und Simulation dynamischer Systeme*, Oldenbourg-Verlag, 2010. – ISBN 978-3-486-59655-7

## 3 M&S auf verschiedenen Ebenen

Analog Ebenen des Schaltkreisentwurfs:

- Systemebene
- Algorithmenebene
- Register-Transfer-Ebene
- Gatterebene
- Schaltkreisebene (Schalterebene)
- Schaltkreisebene (Schaltungsebene)

## 3.1 Systemebene

**Modell:** formale Spezifikation mittels

- Aussagenlogik
- Prädikatenlogik
- Temporaler Logik

**Formaler Nachweis** bestimmter Eigenschaften, z.B.

- Vollständigkeit
- Widerspruchsfreiheit
- ...

**Beispiel:** SAT-Solver



## 3.2 Algorithmenebene

**Merkmal:** Beschreibung des funktionalen und teils auch zeitlichen Verhaltens mittels Hochsprachen

**Ansätze:**

- Modellierung des Datenpfades:
  - Gleichungssysteme, z.B. MatLab / Simulink
  - Unterstützung für High-Level-Synthese nach RTL-Modell, z.B. Xilinx System Generator for DSP, Altera DSP Builder
- Modellierung der Kommunikation:
  - Transaction-Level Modeling
  - Bus Functional Model
- Beides: High-Level-Synthese, z.B. AutoESL

# Transaction-Level Modeling

Genauer: Transaction-Based Modeling

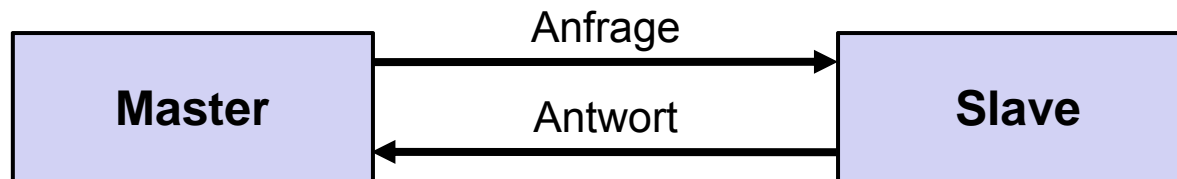
## Merkmale:

- Trennung von Kommunikation und Verarbeitung
- Hochsprachen, z.B. SystemC:
  - Klasse = HW-Modul
  - Klassen definieren Schnittstellen
  - Methodenaufruf führt Transaktion aus
  - Verarbeitung innerhalb Methode (C++ -Code)
  - Simulator = Kompilat des SystemC-Programms
- Optional: Modellierung eines Taktsignals
  - ➔ Taktzyklen-akkurate Simulation

## Kommunikation in SystemC

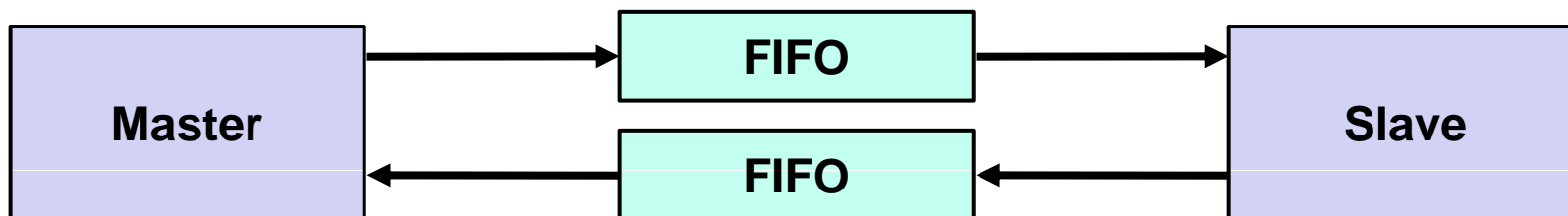
### Signalbasierte Kommunikation:

- Direkte Verbindung der Ein- und Ausgänge von Komponenten
- Eigenes „Protokoll“



### Kanalbasierte Kommunikation:

- Zusätzliche Komponente repräsentiert Verbindungskanal, z.B. FIFO
- Protokoll definiert durch Kanal



## Beispiel: Signalbasierte Kommunikation in SystemC

```
#include "systemc.h"
SC_MODULE(adder)          // module (class) declaration
{
    sc_in<int> a, b;        // ports
    sc_out<int> sum;

    void do_add()           // process
    {
        sum.write(a.read() + b.read()); //or just sum = a + b
    }

    SC_CTOR(adder)          // constructor
    {
        SC_METHOD(do_add);  // register do_add to kernel
        sensitive << a << b; // sensitivity list of do_add
    }
};
```

## Bus Functional Model

auch: Transaction Verification Model

### Merkmale:

- Typisch: Vereinfachte Modelle von Mikroprozessoren mit einem oder mehreren externen Bussen
  - Modellierung der Kommunikation auf dem Bus
  - Befehlsabarbeitung nur vereinfacht
- Zweck: Überprüfung der Funktionalität und des Zeitverhaltens von Bussystemen
- Als Zwischenschritt zum vollständigen Modell auf Algorithmenebene oder detaillierteren Ebenen

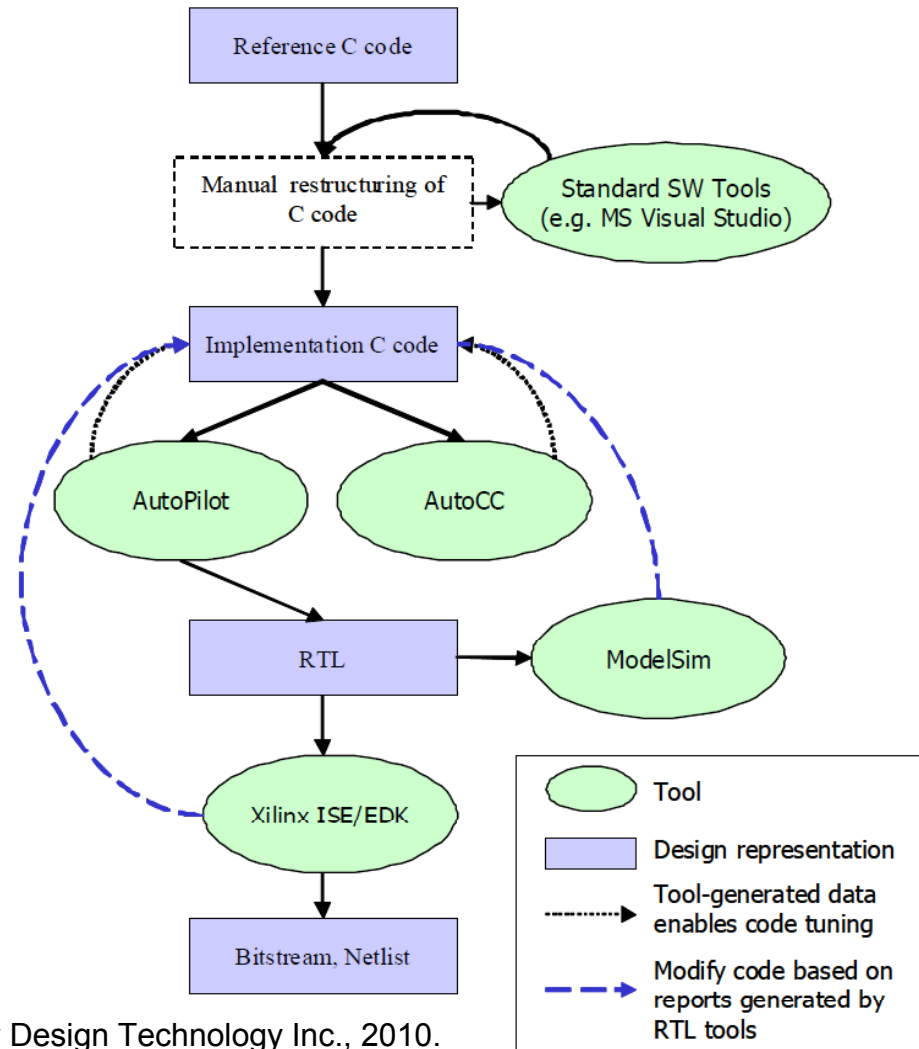
## High-Level-Synthese

### AutoESL:

- Modell in C, C++, SystemC
- Zielplattform: FPGAs
- Automatische Nutzung von IP-Cores
- Simulation mit ModelSim und Aldec

### Angestrebte Vorteile:

- Reduktion Verifikationszeit
- Bessere Performance als bei manuellem RTL-Design (bei Datenpfad-intensiven Designs)
- Portabilität



BDTi: *The AutoESL AutoPilot High-Level Syntheses Tool*, Berkely Design Technology Inc., 2010.

## 3.3 Register-Transfer-Ebene

### Modell:

- Hardwarebeschreibung mit VHDL, Verilog u.a.
- Verhaltensbeschreibung mit Abweichungen in Synthese möglich

### Simulation:

- nur logische Funktion (je Taktperiode)
- taktgenau
- Signaländerung → Ereignis → ereignisorientierte Simulation

## Ereignisorientierte Simulation: Allgemeines

### Ereignisse:

- Signaländerung = Ereignis
- Abarbeitung der Ereignisse entsprechend ihrer zeitlichen Reihenfolge  
→ Ereignisgesteuerte Simulation

### Zeitsteuerung der Simulation: Fortschreibung der Simulationszeit

- bis zum nächsten Ereignis (*next-event time-advance*), oder
- in festen Zeitabständen (*fixed-increment time-advance*)  
= Bearbeitung der Ereignisse zum Ende der Zeitscheibe.

### VHDL:

- 2-dimensionale Simulationszeit mit Delta-Zyklen
- Signalzuweisung nach  $\Delta t$  oder fester Zeit  $t$



## Ereignisorientierte Simulation: Ablauf

### Initialisierung:

- Simulationszeit = 0
- Setze System in Startzustand
- Initialisiere Ereignisliste

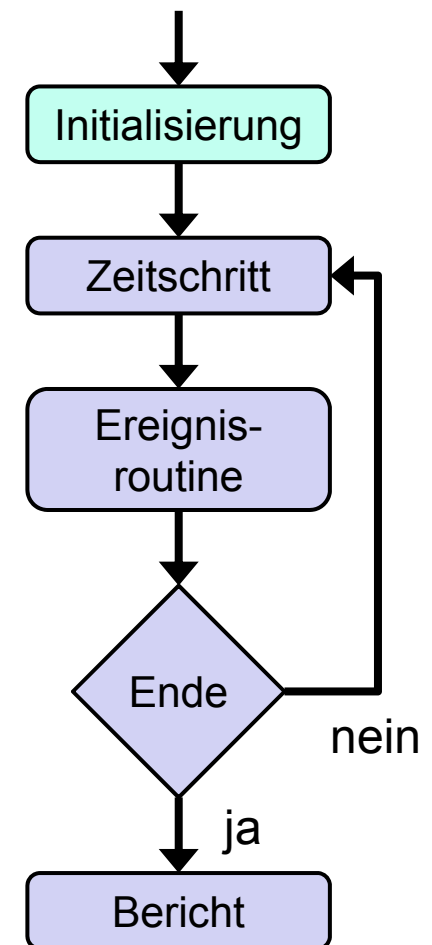
### Zeitschritt:

- Hole nächstes Ereignis  $E_i$  aus Liste
- Schreibe Simulationszeit bis zum nächsten Ereignis fort

### Ereignisroutine $E_i$ :

- Aktualisiere Systemzustand
- Erzeuge neue Ereignisse

### Bericht: Werte Simulation aus



A.M. Law: *Simulation, Modeling & Analysis*, Mc Graw Hill, 2007. – ISBN 978-007-125519-6

## 3.4 Gatterebene

### Modell:

- aus RTL-Modell synthetisierte Netzliste
- Berücksichtigung Schaltverhalten einzelner Gatter auf Basis Fan-out möglich

### Bauelementemodellierung:

- Schaltfunktion entsprechend Hardware
- Schaltverhalten, approximativ; VITAL

### Abweichung zu RTL-Simulation: aufgrund Synthese

- Schaltfunktion von integrierten Speichern
- teils asynchrone Eingänge
- fehlerhafte Sensitivitätslisten → abweichende Synthese

## 3.5 Schaltkreisebene (Schalterebene)

### Modell:

- Schalter, Widerstände, Kapazitäten
- Variante: Back-Annotation der Netzliste durch Technologiesynthese
- Berücksichtigt Schaltverhalten auf Basis der konkreten Verdrahtung  
→ Analyse Verlustleistung möglich

**Abweichung zu Simulation auf Gatterebene** aufgrund zusätzlicher Verzögerungszeit durch konkrete Verdrahtung; betrifft:

- Verhalten asynchroner Eingänge
- Cross-Clock-Designs

## 3.6 Schaltkreisebene (Schaltungsebene)

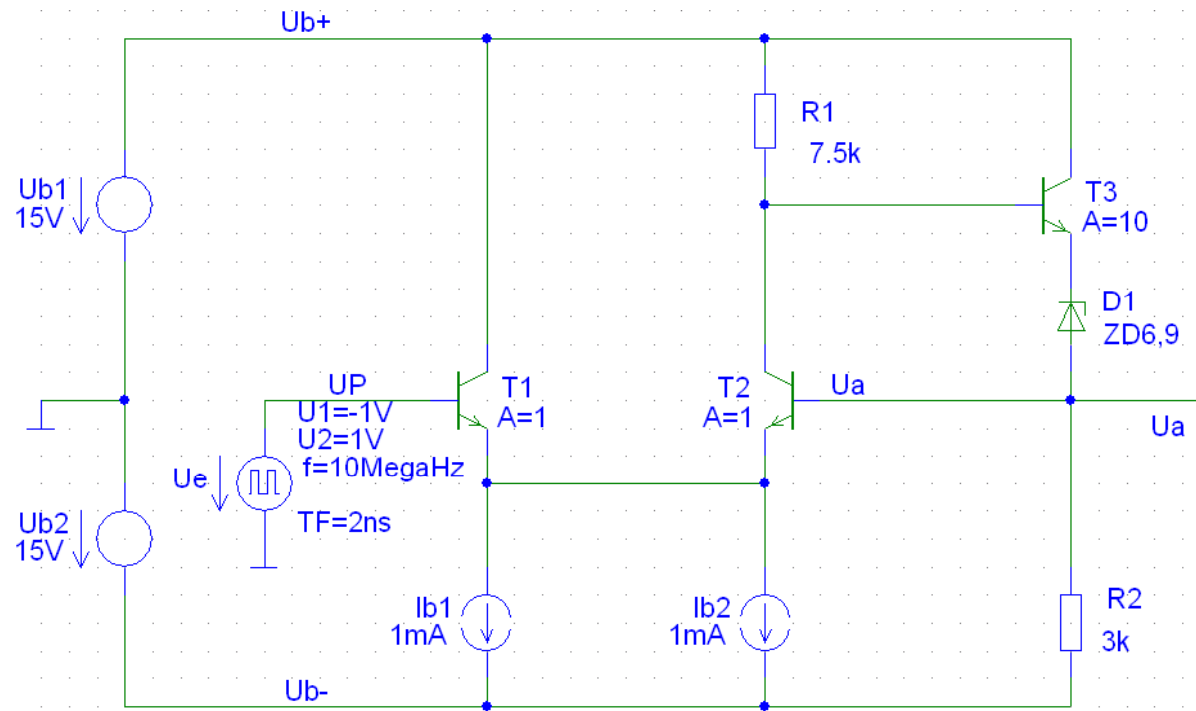
**SPICE** (Simulation Program with Integrated Circuit Emphasis)

- Statische und kontinuierliche Simulation elektrischer Netzwerke
- Eingabe: Schaltplan (Schematic) → Netzliste
- Ausgabe: Tabelle mit Strömen / Spannungen an den Messpunkten, ggf. in Abhängigkeit von Zeit und Frequenz → Diagramme, Statistik
- Typische Analysen:
  - Eingeschwungener Zustand: DC Analysis, DC Transfer-Curve Analysis, Transfer-Function Analysis
  - Frequenzbereich: AC Analysis
  - Zeitbereich: Transient Analysis
  - Rauschen: Noise Analysis
- Vertreter: PSpice, LTspice, TINA, Multisim, ngspice und weitere

## Schaltplan

- Grafische Eingabe
- Auswahl aus einer Bauelementebibliothek

### Beispiel: OPV als Impedanzwandler



## Netzliste

- Manuelle Eingabe oder Generierung aus Schaltplan
- Baulement: Anschlussknoten und Parameter
- Implizite Definition der Knotenmenge, Knoten 0 = Masse

### Beispiel: OPV als Impedanzwandler

```
R_R1      $N_0001 Ub+ 7.5k
V_Ub2     0 Ub- DC 15V
V_Ub1     Ub+ 0 DC 15V
Q_T2      $N_0001 Ua $N_0002 [Ub-] N1 1
D_D1      Ua $N_0003 ZD69
Q_T3      Ub+ $N_0001 $N_0003 [Ub-] N1 10
R_R2      Ub- Ua 3k
I_Ib2     $N_0002 Ub- DC 1mA
I_Ib1     $N_0002 Ub- DC 1mA
Q_T1      Ub+ UP $N_0002 [Ub-] N1 1
V_Ue      UP 0 DC 0 AC 1
+ PULSE -1V 1V 0 2ns 2ns {0.5/10MHz-2ns} {1/10MHz}
```

## Bauelementemodellierung

### **DGL für Grundkomponenten:**

Widerstände, unabhängige und gesteuerte Spannungs-/ Stromquellen, Induktivitäten (auch gekoppelt), Kapazitäten

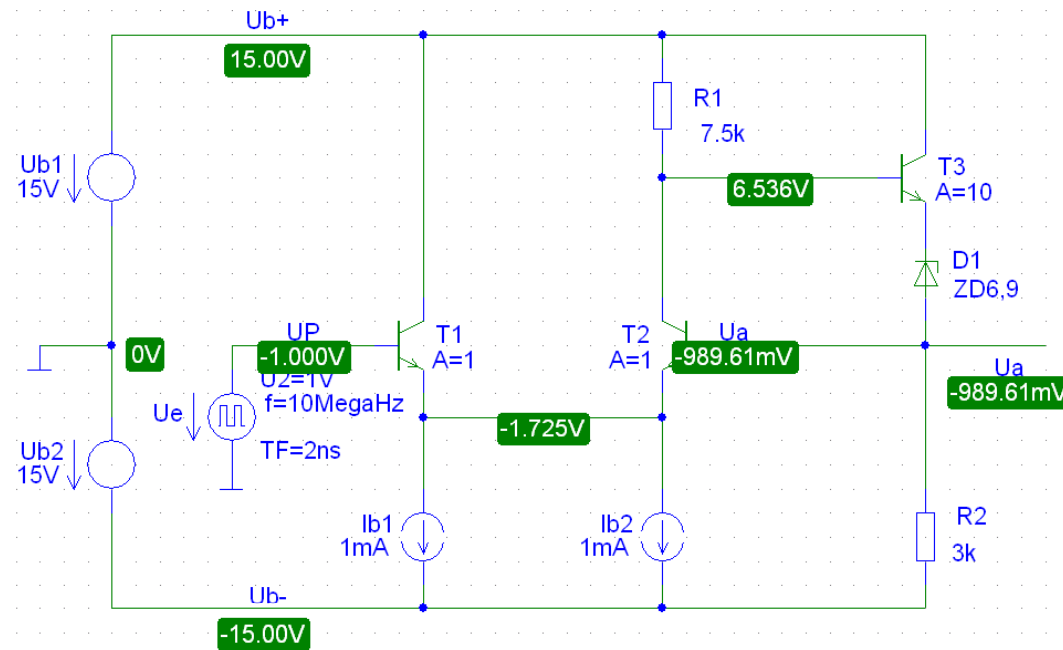
### **Komplexe Modelle für Halbleiterbauelemente:**

- Bestehend aus Grundkomponenten
- Parametrierbare Modelle: Standardwerte hinterlegt in Bauelementebibliothek, Abweichungen in Netzliste spezifiziert
- Diode
- Bipolartransistor: kombiniertes Ebers-Moll + Gummel-Poon-Modell
- MOSFET: Level 1 bis 3, BSIM und weitere
- JFET

### **Komplexe Modelle für eigene Komponenten:** Teilschaltungen

# DC Analysis

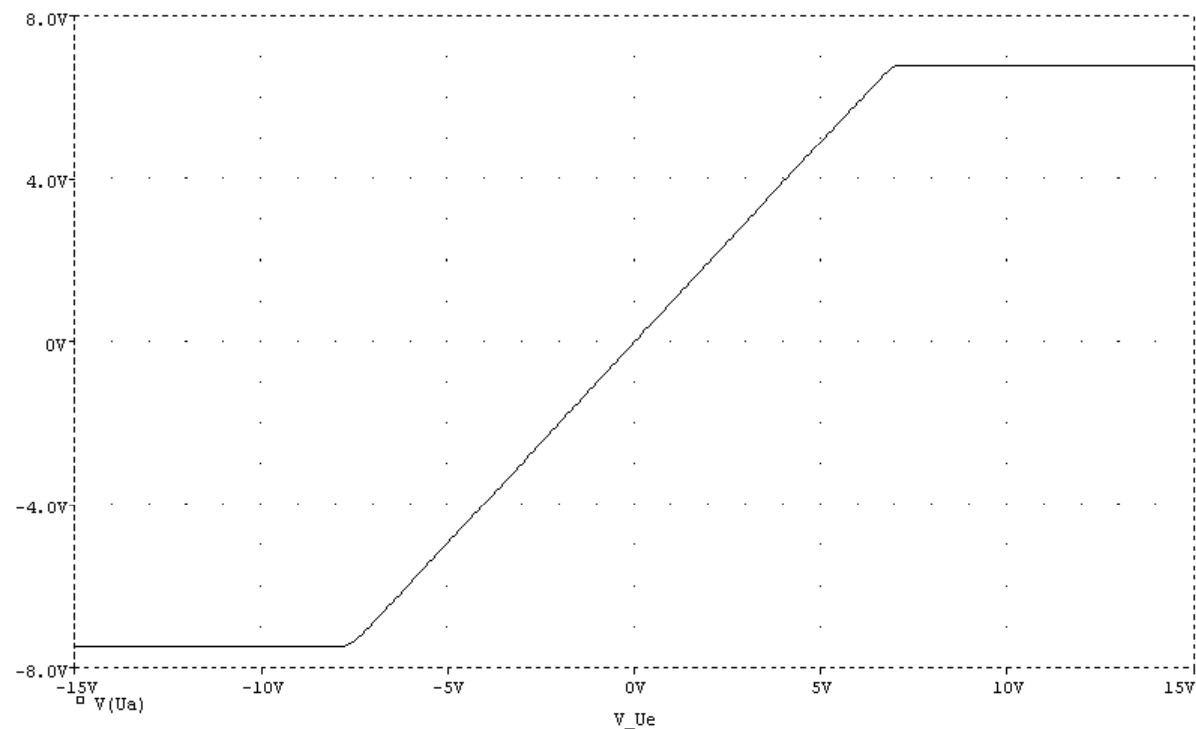
- Nichtlineare Berechnung des eingeschwungenen Zustands (Arbeitspunkt)
- Statisches, deterministisches Modell
- Lösung eines nichtlinearen Gleichungssystems zur Berechnung von Knotenspannungen und Zweigströmen





## DC Transfer-Curve Analysis

- Wiederholte (nichtlineare) Arbeitspunktberechnung unter Variation von Bauelementeparametern
- **Beispiel:** Variation Eingangsspannung → Großsignalkennlinie



## Transfer Function Analysis (DC)

### Ablauf:

1. Berechnung des Arbeitspunktes (AP)
2. Linearisierung der Kennlinien aller Bauelemente entsprechend AP
3. Berechnung Verstärkung und Ein-/Ausgangswiderstand bei angenommener kleiner Variation von Eingangsgrößen (Strom, Spannung)

### Unterschied zu DC Transfer-Curve Analysis:

- Nur Kleinsignalverhalten
- Schneller, da Schritt 3 nur lineare Berechnung statt wiederholter Lösung eines nichtlinearen Gleichungssystems umfasst.

**Anwendung:** Gleichstrom-/Gleichspannungsverstärker

## AC Analysis (1)

### Ablauf:

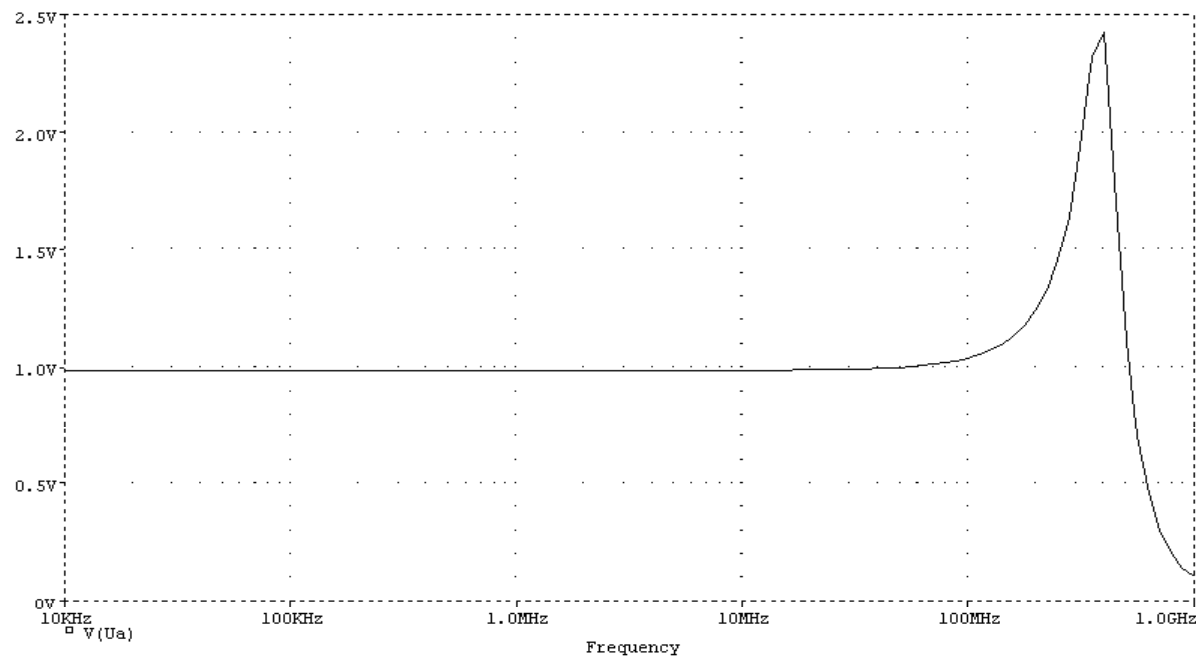
1. Berechnung des Arbeitspunktes (AP)
2. Linearisierung der Kennlinien aller Bauelemente entsprechend AP
3. Wiederholte Berechnung des frequenzabhängigen Kleinsignalverhaltens für einen Frequenzbereich (statisch, deterministisches Modell)

**Anwendung:** Signalverstärker im NF-Bereich

## AC Analysis (2)

**Beispiel:** OPV als Impedanzwandler

- Lineares Verhalten nur bis 100 MHz
- Mittkopplung bei 400 MHz aufgrund Transistorlaufzeiten
- Ab 600 MHz Tiefpassverhalten



## Transient Analysis (1)

### Eigenschaften:

- Nichtlineare Berechnung des zeitlichen Großsignalverhaltens
- Dynamisches, deterministisches, kontinuierliches Modell
- Nichtlineares Differentialgleichungssystem
  - ➔ allg. nicht geschlossen lösbar
- Numerische Simulation von Knotenspannungen und Zweigströmen
  - ➔ automatische Bestimmung der Zeitschritte

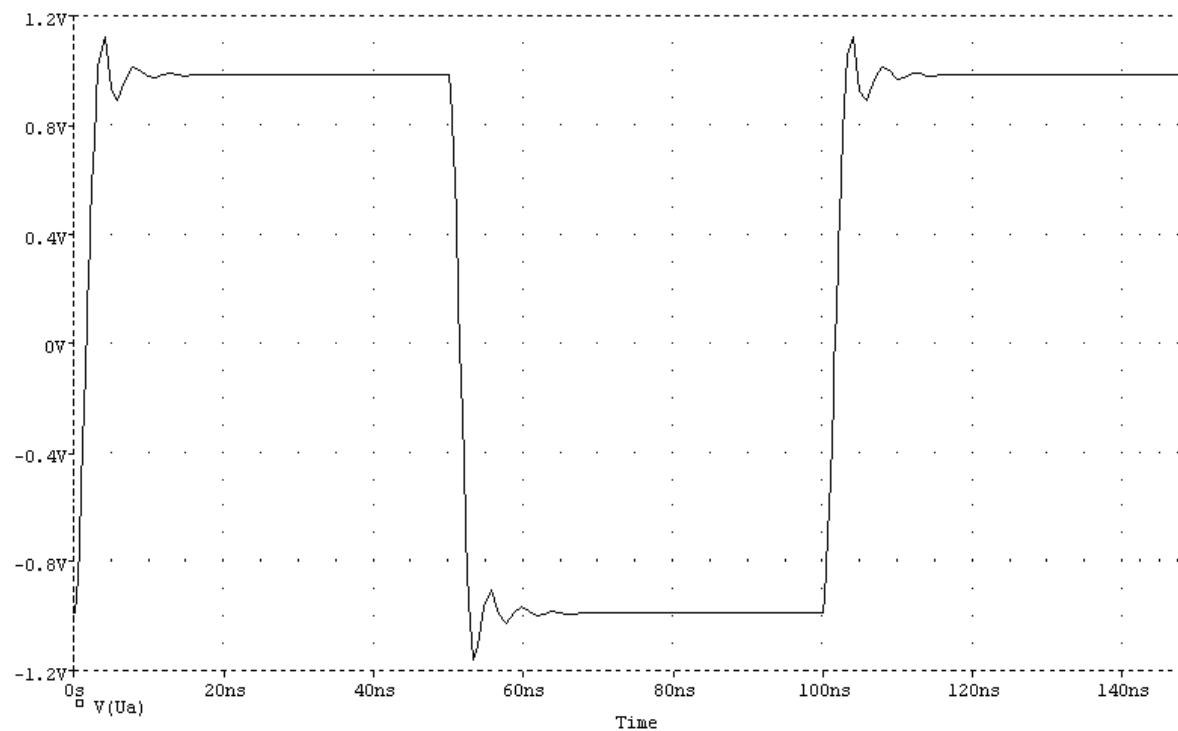
### Anwendung:

- Einschwingvorgänge
- Schaltverhalten allgemein

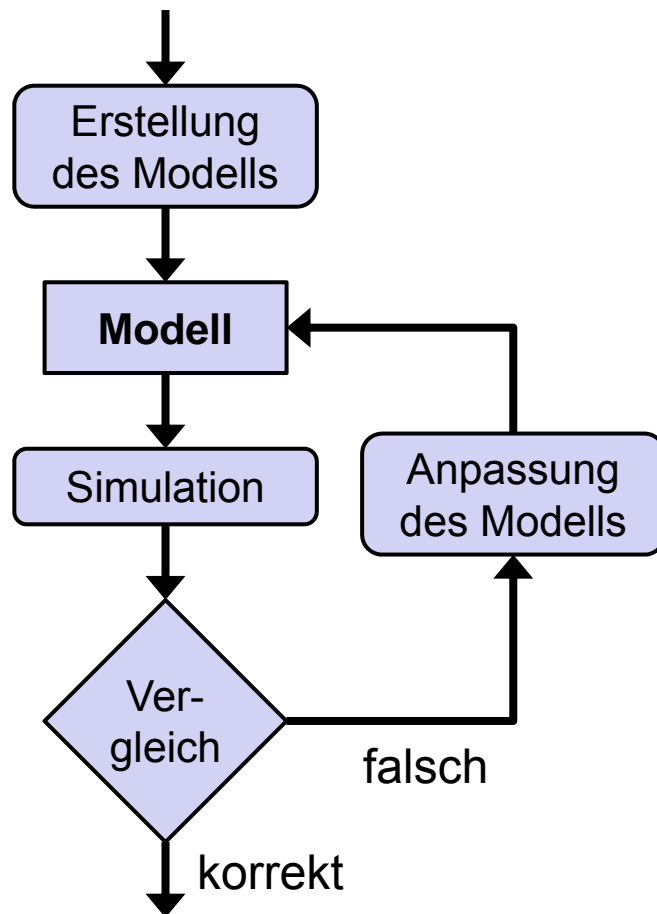
## Transient Analysis (2)

**Beispiel:** OPV als Impedanzwandler

Rechteckimpuls am Eingang → Einschwingvorgang in Sprungantwort



## 4 Modellvalidierung



**Vergleich** der Simulationsergebnisse mit:

- Messungen am System
- Spezifikation

**Anpassung** des Modells:

- Weitere Detaillierung
- Anpassen von Modellparametern

**Automatisierung:**

- Berechnung der Abweichung von Sollkurven
- Assertions (z.B. VHDL)
- Kalibrierung von Modellparametern

## 5 Parameterextraktion

**Ziel:** Vereinfachung von Modellen

➔ Gewinnung von Parametern für einfache Modelle aus der Simulation von Modellen desselben Systems mit komplexen Parametern

**Ausgangspunkt:** validiertes Simulationsmodell

**Ansatz:** Approximation vieler Simulationsergebnisse ➔ Kennlinie

**Beispiele:**

- OPV als Impedanzwandler ➔ RC-Tiefpass
- RTL-Modell eines Prozessors ➔ zyklengenaues Befehlssatzmodell



## 6 Effizienzanalysen und Optimierung

**Ziel:** Optimierung des Systems

**Ausgangspunkt:** validiertes Simulationsmodell

**Ansatz:** Analyse der Effizienz auf Basis der Simulationsergebnisse

→ Iterative Optimierung des Modells

→ Optimierung des Systems

**Beispiele:**

- Optimierung des Frequenzgangs von Verstärkern
- Optimierung von Prozessoren bzgl. Verlustleistung, Zeitverhalten, Funktionales Verhalten (Verarbeitungsleistung)

## 7 Zusammenfassung

### **Begriffe:**

- System, physisches Modell, mathematisches Modell
- Analytische Lösung, Simulation (statisch oder dynamisch, deterministisch oder stochastisch, kontinuierlich oder diskret)

### **Modellierung und Simulation:** auf den Ebenen des Schaltkreisentwurfs

- Formale Spezifikation
- Transaction-Level-Modeling, Bus Functional Model, High-Level-Synthese
- RTL-Modell und ereignisorientierte Simulation
- Gattermodell mit vereinfachtem Zeitverhalten
- Schaltkreismodell mit detailliertem Zeitverhalten

### **Modellvalidierung:** Überprüfung des Modells

### **Parameterextraktion:** Vereinfachung von Modellen

### **Effizienzanalysen:** Optimierung des Systems