
Modellierung des dynamischen Verhaltens nichtlinearer analoger Komponenten für die semi-formale Mixed-Signal-Verifikation

Martin Freibothe
Jens Döge
Torsten Coym
Stefan Ludwig
Bernd Straube

Fraunhofer-Institut Integrierte Schaltungen (IIS) Erlangen
Außenstelle Entwurfsautomatisierung Dresden

Ernst Kock

Infineon Technologies AG
München

Inhalt

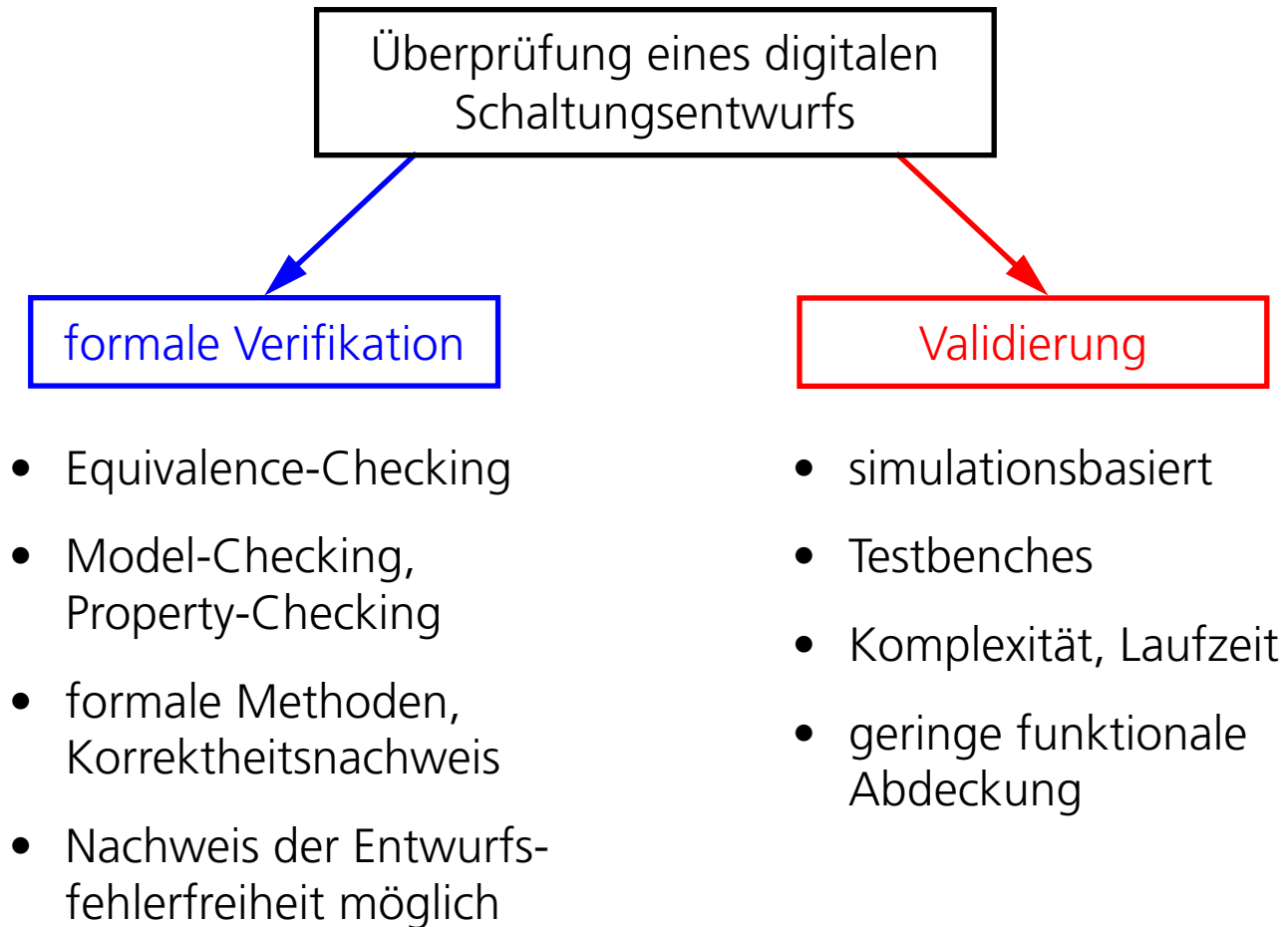
- **Verifikationsmethoden**
- **heutiger Verifikationsablauf für Mixed-Signal-Schaltungsentwürfe**
- **verifikationsgerechte Verhaltensmodellierung**
 - **Zeitdiskretisierung**
 - **Quantisierung**
 - **digitaler Automat**
- **semi-formale Verifikation**
- **Zusammenfassung**

Verifikationsmethoden

- **Ziel: Finden von Entwurfsfehlern**

- **industrielle Praxis: hauptsächlich Simulation**

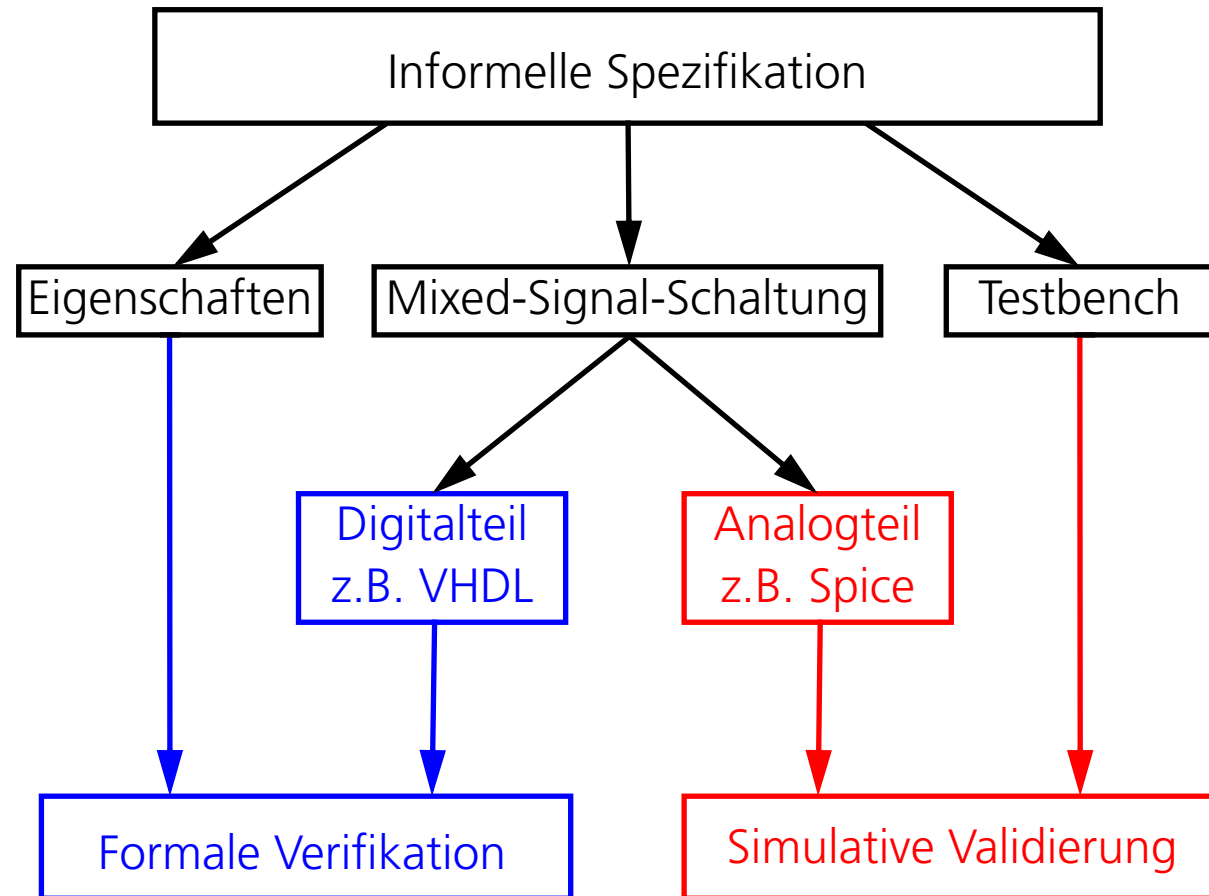
- **Verifikation analoger und Mixed-Signal-Entwürfe ist Gegenstand der Forschung**



Verifikationsablauf für Mixed-Signal-Schaltungen (1)

- **Mixed-Signal-Schaltung**

- **simulative Überprüfung**
- **Trennung in Digital- und Analogteil**
- **Gesamtverhalten wird simuliert**
- **lange Simulationszeiten**
- **manueller Aufwand**
- **geringe funktionale Abdeckung**



Verifikationsablauf für Mixed-Signal-Schaltungen (2)

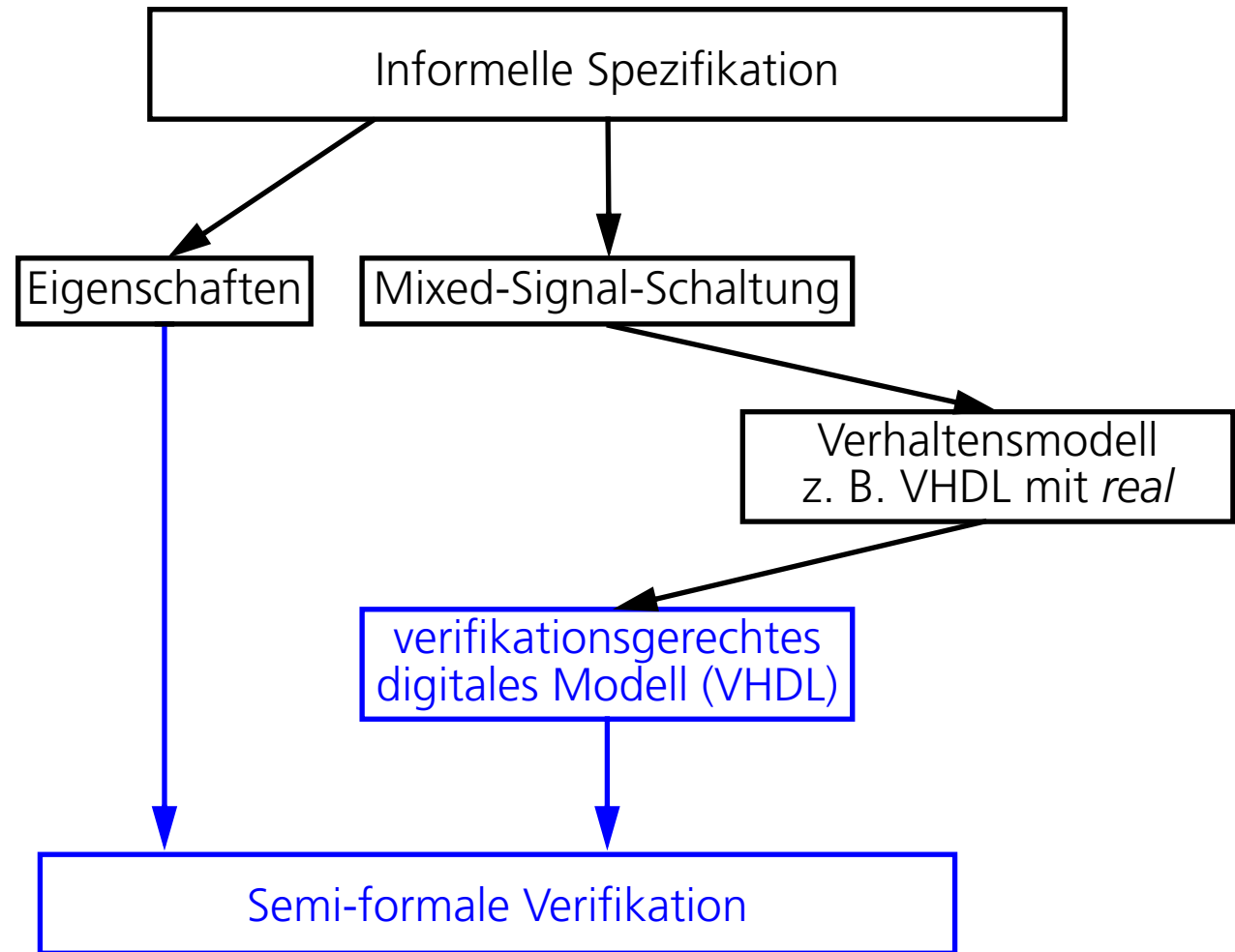
- **digitales Verhaltensmodell: Quantisierung, endlicher Wertebereich**

- Quantisierungsfehler
- mögliche Überläufe

→ **semi-formales Verfahren**

- **offene Probleme:**

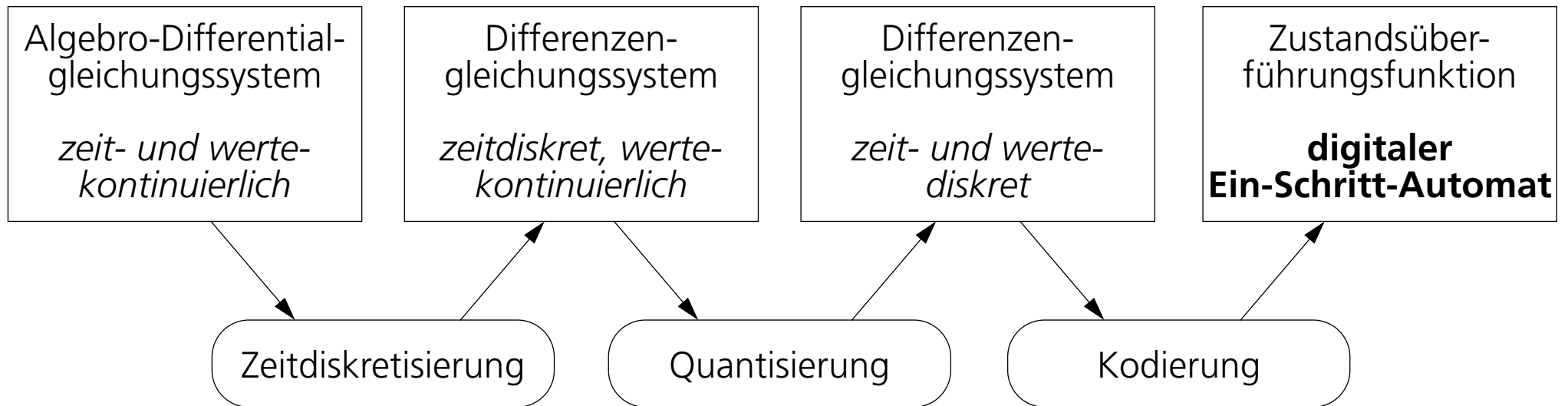
- Bereitstellung der Verhaltensmodelle
- Verifikation dynamischen Verhaltens



Verhaltensmodellierung analoger Komponenten

- **Ziel**
 - Erweiterung der Klasse von Mixed-Signal-Schaltungen, für die formale Verifikationsmethoden eingesetzt werden kann
 - Modellierung von dynamischem Verhalten durch endliche deterministische Automaten
 - digitale Verhaltensmodelle (Implementierung in VHDL)
 - Nutzung vorhandener digitaler Verifikationswerkzeuge
- **Probleme**
 - Verhaltensbeschreibungen analoger Komponenten in Form nichtlinearer Algebra-Differentialgleichungen
 - im Allgemeinen keine geschlossene Lösung

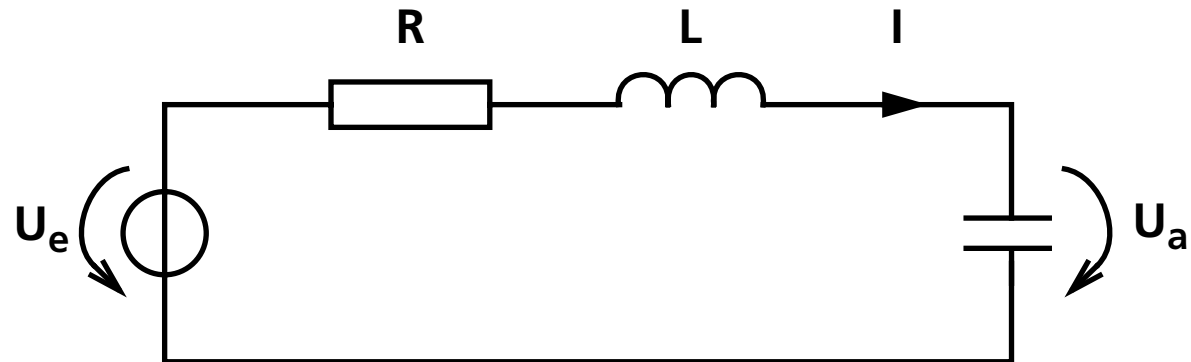
Ablaufschema des Modellierungsansatzes



- **Integrationsformeln**
- **DGL-Löser, Schrittweite**
- **Genauigkeit**
- **Komplexität**
- **Auflösung**
- **Genauigkeit**
- **Komplexität**
- **endlicher Wertebereich**
- **Komplexität**
- **Berechnung der Wertetabelle**

Illustrierendes Beispiel

- RLC-Schaltung
 - Schaltbild



- Kenngrößen für die Modellierung

- Zwei Speichervariablen:
 - Kapazitätsspannung $U_c = U_a$
 - Induktivitätsstrom I
- Eingangsvariable U_e
- Ausgabevariable U_a

Aufstellen des Automaten (1)

- **Ausgangspunkt**
 - Verhaltensbeschreibung durch Zustandsgleichungen (allgemein: Algebro-Differentialgleichungssystem)

$$\begin{bmatrix} \dot{I}(t) \\ \dot{U}_a(t) \end{bmatrix} = \begin{bmatrix} -\frac{R}{L} & -\frac{1}{L} \\ \frac{1}{C} & 0 \end{bmatrix} \cdot \begin{bmatrix} I(t) \\ U_a(t) \end{bmatrix} + \begin{bmatrix} \frac{1}{L} \\ 0 \end{bmatrix} \cdot U_e(t)$$

- **Zeitdiskretisierung**
 - z-Transformation, Anwendung von numerischen Integrationsformeln; hier: Trapezregel
 - Repräsentation der Zeit: Schrittweite h
 - Ergebnis: Differenzengleichungssystem

$$\begin{bmatrix} I(k+1) \\ U_a(k+1) \end{bmatrix} = \frac{1}{K_1} \cdot \left[\begin{bmatrix} K_2 & -\frac{1}{L} \cdot h \\ \frac{1}{C} \cdot h & K_3 \end{bmatrix} \cdot \begin{bmatrix} I(k) \\ U_a(k) \end{bmatrix} + \begin{bmatrix} \frac{1}{L} \cdot \frac{h}{2} \\ \frac{1}{L \cdot C} \cdot \frac{h^2}{4} \end{bmatrix} \cdot (U_e(k+1) + U_e(k)) \right]$$

Aufstellen des Automaten (2)

- **Quantisierung**
 - Genauigkeit der Approximation
 - Komplexität des Modells
 - Komplexität der Verifikationsaufgabe
 - geeignete Wahl der Auflösung und Schrittweite h
 - neues Quantisierungsintervall muß nach einem Zeitschritt h erreicht sein

- **endlicher Wertebereich**
 - endliche Menge von Eingabe-, Zustands- und Ausgabesymbolen für den Automten
 - Komplexität des Modells
 - Überdeckung des praxisrelevanten Wertebereichs mit mehreren Teilmodellen
 - »Teile und Herrsche« —

Aufstellen des Automaten (3)

- **Parameter für die Beispielschaltung**
 - **Spannungen:**
-6,4 V . . . 6,0 V mit 0,4 V / bit
 - **Strom:**
-2,0 mA . . . 1,875 mA mit 0,125 mA / bit
 - **Repräsentation durch Bitvektoren der Länge 5, Schrittweite $h = 0,7$ ns**

- **Zustandsüberföhrungs-
funktion als Wertetabelle**
 - **Berechnung des Differenzgleichungssystems für jede Kombination von Eingangs- und Anfangszustandsbelegung**
 - **auch technisch nicht mögliche Kombinationen**
 - **vollständige Beschreibung in Bezug auf gewählte Parameter (Quantisierung, Wertebereich)**

Zustandsüberföhrungsfunktion

- Auswahl einer endlichen Anzahl von Eingangsverläufen; hier:

0,0 V → 0,0 V

0,0 V → 3,2 V

3,2 V → 3,2 V

3,2 V → 0,0 V

$U_e(k)$	$U_e(k+1)$	$I(k)$	$U_a(k)$	$I(k+1)$	$U_a(k+1)$
0,0	0,0	-1,250	0,8	-0,750	0,0
0,0	0,0	-1,250	1,2	-1,000	0,4
0,0	0,0	-1,375	2,8	-1,750	1,6
...
3,2	3,2	-1,375	2,4	-0,125	1,6

- Berechnung der Wertetabelle der Zustandsüberföhrungsfunktion
- 4096 Tabellenzeilen
- vollständige Beschreibung des Ein-Schritt-Automaten
 - durch die Ausgabefunktion λ und
 - die Zustandsübergangsfunktion δ

Implementierung des Ein-Schritt-Automaten

- **synthesefähige Unter-
menge von VHDL**
- **voltage_ut** und **current_ut**
intern als Bitvektoren
- **Takteingang**

```
architecture behavior of rlc is
begin

    p1: process (clk, U_e)
        variable U_intern : voltage_ut;
        variable I_intern : current_ut;
        variable U_e_prev : voltage_ut;
    begin

        if clk'event and clk = '1' then

            if( U_e_prev = voltage_ut( 0.0 )    and
               U_e      = voltage_ut( 3.2 ) ) then

                if( I_intern = current_ut( 0.001500 ) ) then
                    if( U_intern = voltage_ut( -2.400000 ) ) then
                        I_intern := current_ut( 0.001875 ) ;
                        U_intern := voltage_ut( -0.800000 ) ;
                    end if;
                end if;
            end if;
            ...
        end process;
    end behavior;

entity rlc is

    port (
        -- Takt
        clk: in std_ulogic;

        -- Ein- und Ausgangs-
        -- spannung
        U_e: in voltage_ut;
        U_a: out voltage_ut);

end rlc;
```

Repräsentation des dynamischen Verhaltens

- **Komposition des Ein-Schritt-Automaten**
- **Ausführung über mehreren Takten**
 - **Taktperiode entspricht der Schrittweite h**
 - **beliebige Folge von Eingangsbelegungen in Bezug auf die beim Erstellen berücksichtigten Werte**
 - **Folge der Ausgangsbelegungen stellt das approximierte, zeit- und wertediskretisierte dynamische Verhalten dar**

Nachweis von Eigenschaften

- **Eigenschaft für den Einschwingvorgang:**

property example is

```
assume:  
  during[t, t+1]:  
    either  
      U_e = VDD_0V0; or  
      U_e = VDD_3V2;  
    end either;
```

- **bounded model checking**
 - **endliches Intervall**
 - **hier: [t; t+14]**

```
during[t+2, t_last]: U_e = prev( U_e );
```

```
prove:  
  at t+14: U_a = prev( U_a );  
  at t+14: I_intern = prev( I_intern );
```

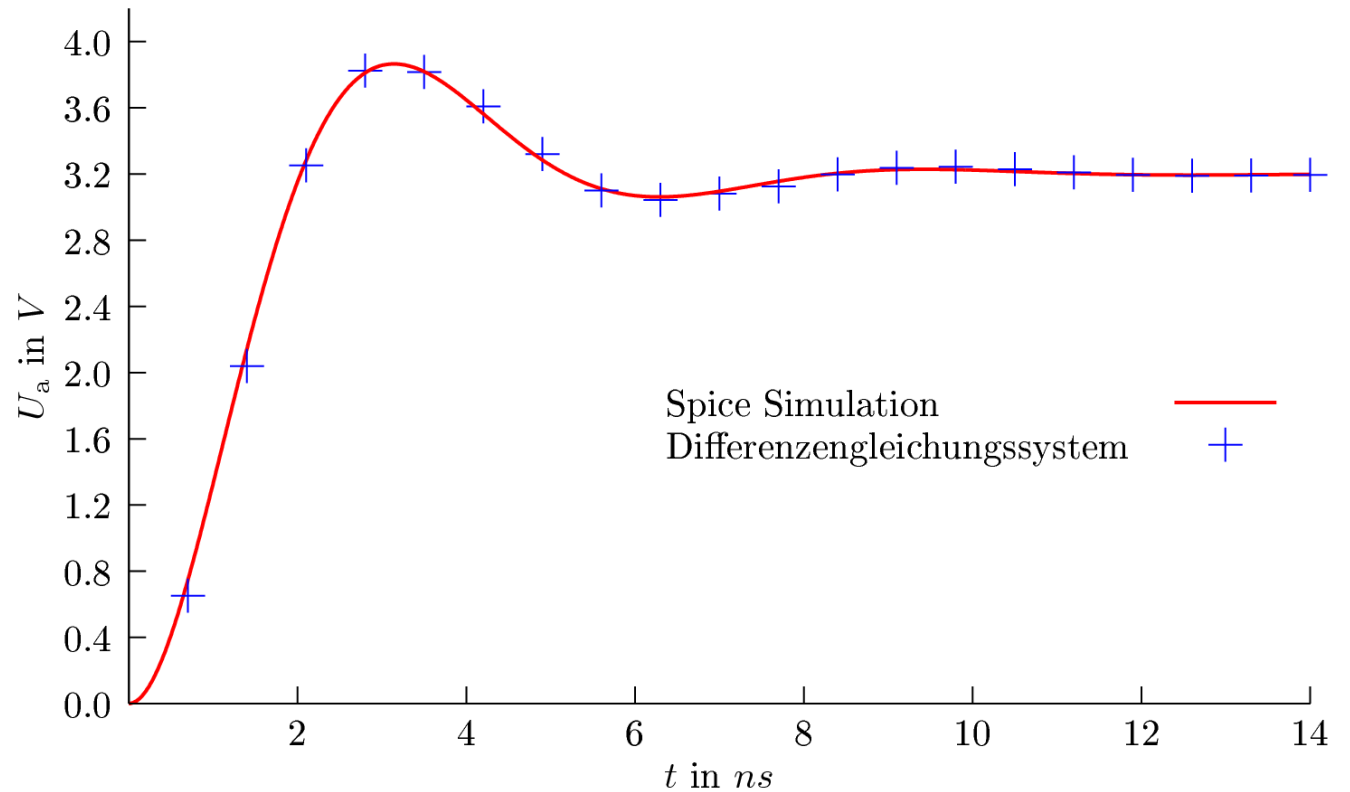
```
end property;
```

- **Ressourcenverbrauch**
 - **10 Minuten CPU**
 - **750 MB RAM**

- **unabhängig vom Anfangszustand ist die Schaltung nach 12 Takten des Ein-Schritt-Automaten eingeschwungen**
- **12 Takte des Ein-Schritt-Automaten entsprechen mit der gewählten Schrittweite 8,4 ns**

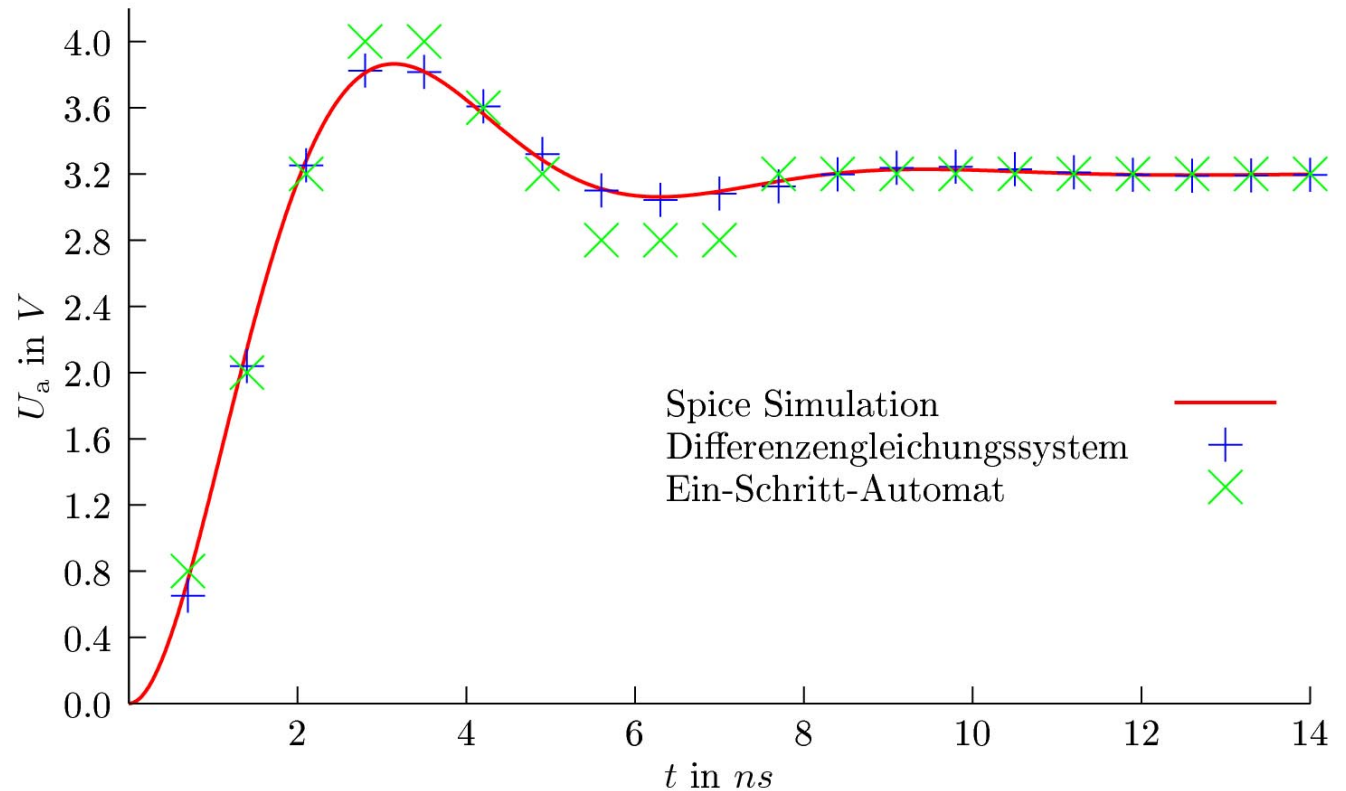
Approximationsfehler

- Modell repräsentiert approximiertes Verhalten
 - Diskretisierungsfehler



Approximationsfehler

- **Modell repräsentiert approximiertes Verhalten**
 - Diskretisierungsfehler
 - Quantisierungsfehler
- **Berücksichtigung in Eigenschaften**
- **semi-formale Methode**



Betrachtete Klasse von Analogkomponenten

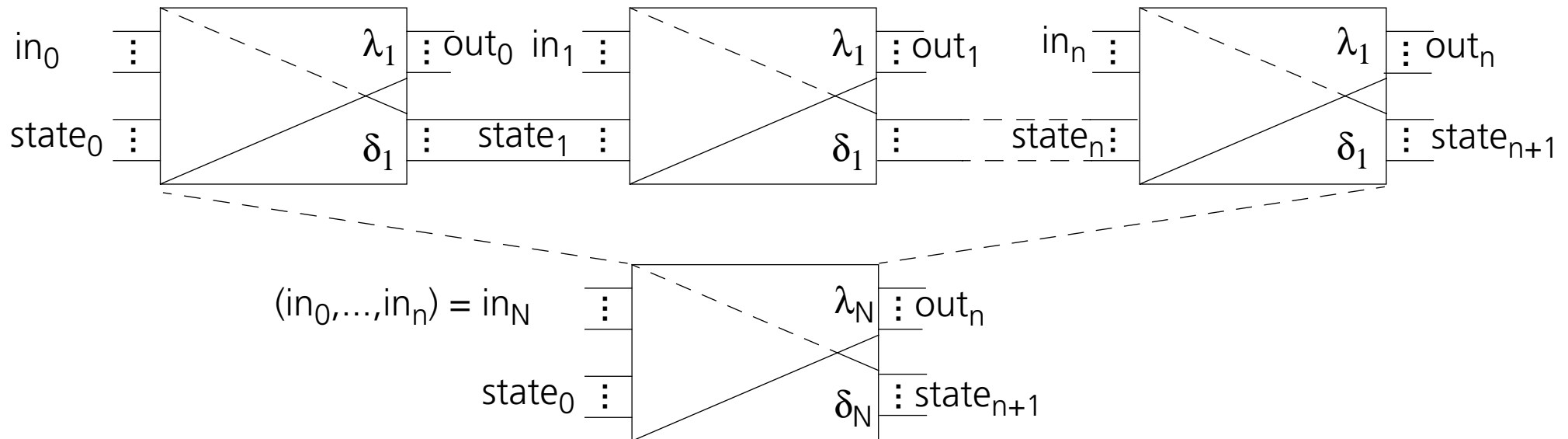
- beliebige nichtlineare Schaltungen
- beliebige Verläufe der Belegung der Eingangsvariablen
- sehr große und sehr kleine Anstiege
 - Beachtung bei Wahl der Auflösung für die Quantisierung und Schrittweite h
- begrenzte Anzahl von Speichervariablen (Kapazitäten, Induktivitäten)

n-Schritt-Automat

- **Ziel**
 - Modell zur Repräsentation analogen Verhaltens bezüglich des digitalen Umgebungstaktes
 - digitales Modell ersetzt kontinuierliches Verhaltensmodell
 - digitales Modell der gesamten Mixed-Signal-Schaltung

- **Vorgehen**
 - weiterer Abstraktionsschritt ausgehend vom Ein-Schritt-Automaten
 - Aufstellen eines n-Schritt-Automaten
 - Taktperiode T ist ganzzahliges Vielfaches der Schrittweite h mit $n \cdot h = T$

Aufstellen eines n-Schritt-Automaten



- **n-Schritt-Automat**

- Eingabevariablen

- Ausgabefunktionen: λ_N

- Zustandsübergangsfunktionen: δ_N

- $\text{out}_n = \lambda_1 (\delta_1 (\dots \delta_1 (\delta_1 (\text{state}_0, \text{in}_0), \text{in}_1), \dots, \text{in}_{n-1}), \text{in}_n)$

- $= \lambda_N (\text{state}_0, (\text{in}_0, \dots, \text{in}_n))$

- $\text{state}_{n+1} = \delta_1 (\dots \delta_1 (\delta_1 (\text{state}_0, \text{in}_0), \text{in}_1), \dots, \text{in}_n)$

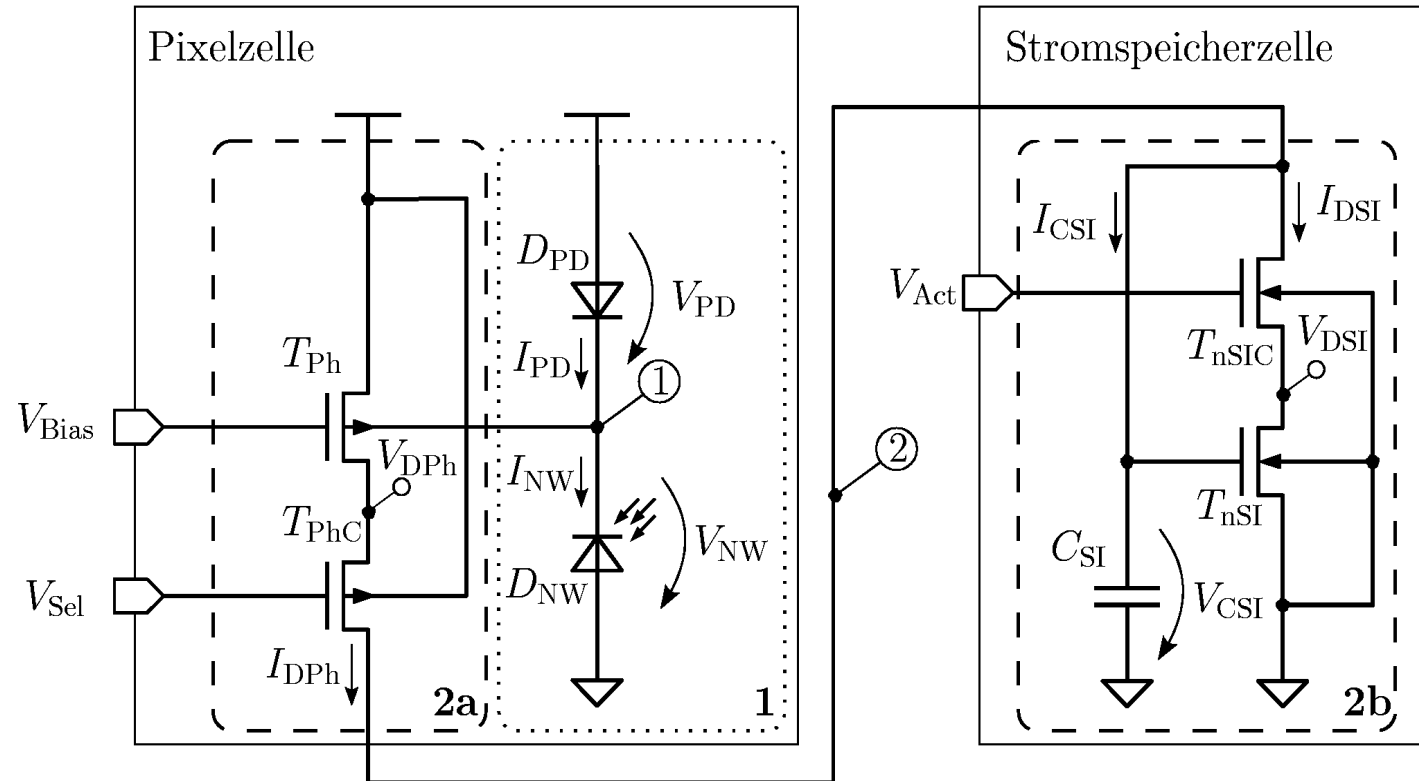
- $= \delta_N (\text{state}_0, (\text{in}_0, \dots, \text{in}_n))$

Anwendungen des digitalen Modells

- **Ein-Schritt-Automat**
 - Nachweis praxisrelevanter Eigenschaften der dynamischen Vorgänge von analogen Komponenten
 - Komposition mehrerer Ein-Schritt-Automaten
 - Weitere Abstraktion durch Aufstellen eines n-Schritt-Automaten
- **n-Schritt-Automat**
 - Zusammenschaltung mit digitaler Umgebung
 - Komposition mehrerer n-Schritt-Automaten

Anwendungsbeispiel: Pixelzelle (1)

- Pixelzelle mit Stromspeicher
- Zwei Modi:
 - integrierend
 - logarithmierend
- Eingangsvariable:
 - Photostrom I_{NW}
- Ausgangsvariable:
 - Spannung über C_{SI}



Anwendungsbeispiel: Pixelzelle (2)

- **Partitionierung**
 - Unterscheidung der charakteristischen Zeitkonstanten um sechs Größenordnungen
 - Sensitivitätsanalyse
 - Annahme der Rückwirkungsfreiheit
 - Modellierung mit Hilfe von zwei Ein-Schritt-Automaten

- **Wertebereich**
 - integrierender Modus der Schaltung
 - kleines Intervall des Eingangspdstroms
 - hohe Auflösung

Anwendungsbeispiel: Pixelzelle (3)

- **Zustandsüberföhrungs-
funktion**
 - Nichtlinearitäten → keine analytische Lösung
 - numerische Differentialgleichungslöser
 - Matlab, Mathematica (interpretierend)
 - SUNDIALS Bibliothek (C-Funktionen)

- **Verifizierte Eigenschaften**
 - Überschwingen: Nachweis der Grenzen mit Hilfe der Ein-Schritt-Automaten
 - nachfolgende Schaltungskomponente übernimmt gültige Werte
 - Ausgangsspannung ist im gültigen Intervall

Zusammenfassung

- **Anwendung formaler Verifikationsmethoden für eine größere Klasse von Mixed-Signal-Schaltungen**
- **Nachweis von praxisrelevanten Eigenschaften über dynamisches Zeitverhalten analoger Komponenten**
- **Bereitstellung von digitalen Verhaltensmodellen analoger Komponenten, die kontinuierliche Modelle in der Zusammenschaltung ersetzen**
- **exponentieller Aufwand der Modellierung in der Anzahl der Eingangs- und Zustandsvariablen sowie der Anzahl der Quantisierungsstufen**
- **Verifikationsverfahren ist aufgrund der Modellierung trotz der Anwendung formaler Techniken semi-formal**