

# Graphentheoretischer Ansatz zur Initialdimensionierung analoger Schaltungen

Dr.-Ing. Volker Boos  
IMMS Ilmenau, Außenstelle Erfurt  
[volker.boos@imms.de](mailto:volker.boos@imms.de)

## Kurzfassung

Beim analogen Schaltungsentwurf werden zur Dimensionierung der Bauelemente Optimierungstools eingesetzt, um eine optimale Performance und maximale Robustheit zu erreichen. Beginnend mit einer Anfangslösung berechnen diese Tools iterativ bessere Lösungen. Dabei kann eine gute Anfangslösung die Rechenzeit stark verkürzen und den Optimierungserfolg verbessern. Untersuchungen haben gezeigt, dass die Optimierung wesentlich leichter zu beherrschen ist, wenn an den Bauelementen bestimmte DC-Bedingungen (Constraints) erfüllt sind. In diesem Beitrag wird gezeigt, wie durch graphentheoretische Ansätze die optimalen Knotenspannungen und Zweigströme mit geringem Rechenaufwand ermittelt werden und daraus die Dimensionierung der Bauelemente als gute Startlösung für Optimierungstools berechnet wird.

## 1 Einleitung

Im Designflow des analogen Schaltungsentwurfs erfolgt nach der Auswahl einer geeigneten Schaltungstopologie die Dimensionierung der Bauelemente [1]. Diese umfasst im Allgemeinen zwei Schritte, die Nominaloptimierung für optimale Performance und die Designzentrierung für maximale Robustheit gegenüber Prozesstoleranzen.

Die aktuellen Optimierungstools arbeiten iterativ, das heißt, sie beginnen mit einer Anfangslösung und errechnen schrittweise bessere Lösungen bis keine Verbesserung mehr eintritt. Beginnt die Optimierung mit einer ungünstigen Anfangslösung, steigt die Rechenzeit stark an, oft wird auch die optimale Lösung wegen der starken Nichtlinearität des Problems nicht erreicht. Deshalb ist es wichtig, mit einer guten Initiallösung zu starten.

Untersuchungen bei der Entwicklung des Optimierungs- und Designzentrierungstools „WiCkeD“ haben

gezeigt, dass das Optimierungsproblem nahezu linear wird, wenn die DC-Arbeitspunkte der Transistorstrukturen bestimmte Bedingungen (Constraints) erfüllen. Wird der Arbeitspunkt jedes Bauelements durch seine Spannungen und Ströme vorgegeben, kann beispielsweise beim MOS-Transistor  $W$  und  $L$  berechnet werden.

## 2 Sizing Strategien

Wenn man von der manuellen Dimensionierung absieht, arbeiten die meisten Verfahren iterativ mit den beschriebenen Nachteilen. Einen neuen Ansatz beschreibt F. Leyn [2] als „Operating-Point Driven“. Die Idee ist, dass für ein gewünschtes elektrisches Verhalten bestimmte Bedingungen (Constraints) für Spannungen und Ströme an den Anschlüssen des Bauelements eingehalten werden müssen.

Werden für jedes Bauelement diese Werte festgelegt, lässt sich daraus die Dimensionierung errechnen. In einem sinnvollen elektrischen Netzwerk gibt es Knoten mit fester Span-

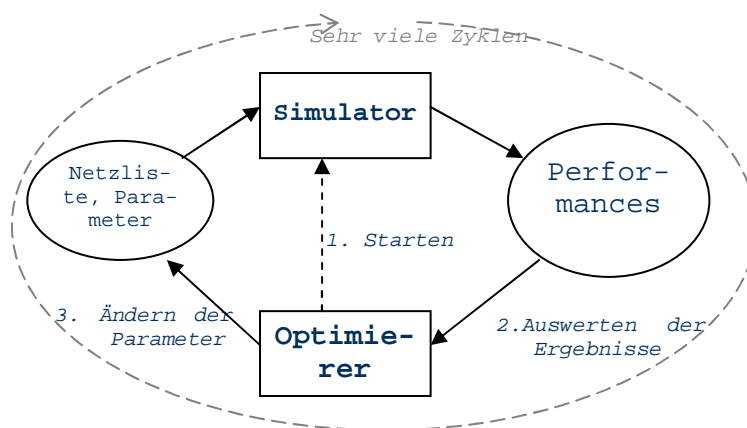


Abb. 1 : Iterative Optimierung

nung (Betriebs- und Referenzspannungen) und Knoten, deren Spannung wegen den Constraints in einem bestimmten Intervall liegen dürfen. Leyn [2] verwendet einem MinMax-Algorithmus, um die Spannungsreserven an den freien Knoten gleichmäßig zu verteilen.

### 3 Graphentheoretische Ansatz

Der hier gezeigte Ansatz löst das Problem der Initialdimensionierung graphentheoretisch ohne zeitaufwändige Iterationsschleifen. Der verwendete Graph besteht aus Knoten, die den elektrischen Knoten entsprechen, und gerichteten Kanten, die mit ihrer Gewichtung in Richtung steigender Spannung die Mindestspannung und in Richtung fallender Spannung die Maximalspannung zwischen zwei Knoten charakterisieren. Nach der topologischen Sortierung wird mit einem modifizierten Algorithmus aus der Terminplanung für jeden Knoten die minimal und maximal mögliche Spannung berechnet. Ist das Spannungsintervall für mindestens einen Knoten negativ, ist die Schaltung nicht realisierbar. Im nächsten Schritt wird für jeden Knoten eine optimale Spannung aus dem errechneten Intervall so bestimmt, dass die Reserven für die Mindestspannungen möglichst gleichmäßig verteilt sind.

In einer analogen CMOS-Verstärkerschaltung werden die meisten Transistoren im Sättigungsbereich betrieben. Daher soll dieser Fall genauer beschrieben werden. Bei einem N-MOS Transistor im Sättigungsbe-  
reich gelten folgende Beziehungen:

$$V_{GS} > V_{th} \text{ und } V_{DS} > V_{GS} - V_{th}.$$

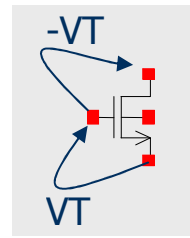
Um die Spannungsintervalle mit dem graphentheoretischen Algorithmus zu lösen, sind auf der rechten Seite der Ungleichung Konstanten erforderlich. Für die Threshold-Spannung wird eine Abschätzung verwendet:

$$V_{th} \approx V_{T0} + DVT = VT \text{ mit } DVT \approx 0.1V \text{ je nach Technologie. Die zweite Ungleichung wird nach Subtraktion von } V_{GS} \text{ und wegen } V_{DS} - V_{GS} = V_{DG} \text{ zu}$$

$$V_{DG} \geq -V_{th} .$$

Diese beiden Beziehungen werden als Kanten in den Graph eingefügt:

$$V_{GS} \geq VT \text{ und } V_{DG} \geq -VT .$$



Die Zweigströme werden im einfachsten Fall pauschal vorgegeben, z.B. mit mindestens 50µA pro Transistor, für einzelne Zweige ist aber auch die Vorgabe spezieller Werte möglich. Anschließend werden aus den Strombedingungen die Zweigströme so errechnet, dass die Kirchoffsche Knotenregel für jeden Netzwerkknoten erfüllt ist.

Sind alle Knotenspannungen und Zweigströme festgelegt, kann daraus die Dimensionierung der Bauelemente berechnet werden. Bei MOS-Transistoren sind z.B. Weite (W) und Länge (L) zu bestimmen.

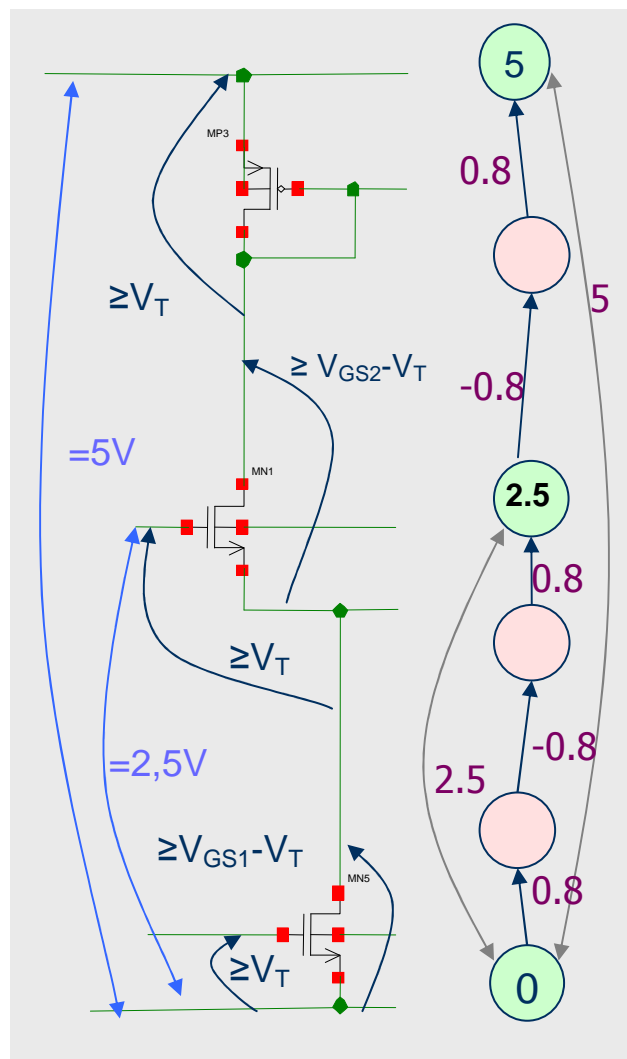


Abb. 2: Spannungsbeziehungen an einer Eingangsstufe und Graph mit  $V_T = 0.8V$

## 4 Dimensionierung von MOS-Transistoren

Der Drainstrom eines MOS-Transistors in einer vorgegebenen Technologie ist abhängig von den Design-Parametern ( $W$ ,  $L$ ) und den Anschlussspannungen:

$$I_D = f(V_{GS}, V_{DS}, V_{SB}, L, W) \quad (1)$$

$V_{GS}$  und  $V_{DS}$  sind gegeben.  $V_{SB}$  wird vorläufig auf 0 gesetzt, der relativ geringe Einfluss wird durch das verwendete  $V_T > V_{th}$  kompensiert.  $L$  wird zunächst wegen des geringsten Platzbedarfs auf das technologische Minimum  $L_{min}$  gesetzt. Dann ergibt sich eine fast lineare Beziehung zwischen  $I_D$  und  $W$ . Für gegebenen  $I_D$  muss nun  $W$  berechnet werden. Die verschiedenen Lösungsmöglichkeiten werden weiter unten diskutiert. Ist das berechnete  $W_0$  kleiner als das technologische Minimum  $W_{min}$ , wird  $W=W_{min}$  gesetzt und mit der Gleichung (1)  $L_0$  berechnet.

### 4.1 Berechnung mit Modellgleichungen

Die Formeln des Level 1 Modells lassen sich einfach nach  $W$  umstellen:

$$I_D = \frac{\mu \cdot C_{ox}}{2} \cdot \frac{W}{L} \cdot (V_{GS} - V_{th})^2 (1 + \lambda V_{DS})$$

(in Sättigung)

Das Problem ist, dass nicht alle technologischen Konstanten in den Modellkarten direkt enthalten sind, sondern vom Simulator mit

sehr komplexen Formeln berechnet werden [3]. Unter Verwendung der Parameter aus den Modellkarten ( $\mu_0$  statt  $\mu_{eff}$ ,  $V_{T0}$  statt  $V_{th}$ ), und des Bauelements ( $W$  statt  $W_{eff}$ ,  $L$  statt  $L_{eff}$ ) entsteht somit nur eine Näherung, die aber als Startlösung für andere Verfahren benutzt werden kann.

### 4.2 Berechnung mit OCEAN Skript

Im CADENCE Design Framework besteht die Möglichkeit, Simulationsaufgaben mit Skripten zu programmieren. Die Skriptsprache OCEAN [4] enthält den Befehl **root**, mit dem für eine durch Simulation entstandene Funktion  $y = f(x)$  der  $x$ -Wert für ein gegebenes  $y$  berechnet wird. Das verwendete Skript wird aus einem Template (Listing 1) erzeugt und durchläuft die DC-Analyse mit mehreren Sweeplisten. Die Parameter in den Sweeplisten müssen so gewählt werden, dass alle möglichen Weiten und Längen berücksichtigt werden, die Kurve zwischen den Werten etwa linear ist, und auch keine Modellwechsel stattfinden.

### 4.3 Berechnung mit symbolischer Analyse

Diese Möglichkeit wurde nur theoretisch in Betracht gezogen, da die Großsignalmodelle für MOS-Transistoren noch nicht zur Verfügung standen.

```
sweep1ists = list( list("0.5u", "1u", "2u", "4u", "8u", "16u")
                  list("0.35u", "0.6u", "1u", "2u", "4u", "8u"))
sweepparams = list( "W", "L" )

foreach (sweep1ist, sweepparam) sweep1ists sweepparams

analysis('dc ?saveOppoint t ?param sweepparam ?values sweep1ist )
save( 'i "/M0/D" )

...
for each instance not yet sized
  desVar( "VDS" { $Vds } )
  desVar( "VGS" { $Vgs } )
  desVar( "IDSoll" { $Ids } )
  idsoll = evalstring( desVar("IDSoll") )
  run()
  selectResult( 'dc )
  idwf = i("/M0/D")
  S0 = root(idwf idsoll 1)
  /* if S0 is a valid W or L the device is sized, otherwise the
  device will sized with the next sweep1ist */
```

Listing 1 : Auszug aus dem OCEAN-Skript Template zur Berechnung von  $W$  und  $L$

## 5 Dimensionierung weiterer Bauelemente

### Bipolartransistoren

Das „Operating Point Driven“ Prinzip ist für Bipolartransistoren nicht direkt geeignet. In der Literatur gibt es aber Lösungsansätze [6].

### Widerstände

Im Graphenmodell wird in Richtung steigender Spannung eine Kante mit 0 Volt Mindestspannung eingefügt. Sind Knotenspannungen und der Strom durch den Widerstand gegeben, kann der Wert mit dem Ohmschen Gesetz direkt berechnet werden.

### Spannungsquellen

An DC-Spannungsquellen werden zwei Kanten in den Graph eingefügt, eine für die Mindest- und eine für die Maximalspannung. So wird gesichert, dass genau die gewünschte Spannung an den Knoten anliegt.

### Stromquellen

An Stromquellen wird wie bei Widerständen eine Kante mit der Gewichtung 0 eingefügt.

### Kapazitäten

Kapazitäten werden nicht berücksichtigt.

## 6 Ergebnisse

Die beschriebenen Verfahren wurden in einem Testprogramm implementiert. Die Dimensionierung wurde an Schaltungen mit bis zu 20 Transistoren getestet. Die Simulation zeigte das gewünschte Verhalten. Eine anschließende Optimierung mit WiCkeD erreichte in kürzerer Zeit bessere Ergebnisse.

## 7 Ausblick

Der „Operating-Point Driven“ Ansatz erfordert beim Erstentwurf einer Topologie interaktive Eingriffsmöglichkeiten, z.B. um den Strom zu justieren. Daher ist der Einbau in den Designflow mit einer grafischen Oberfläche sinnvoll.

Ein weiteres Einsatzgebiet ist die Dimensionierung von technologieunabhängigen Topologien aus entsprechenden Bibliotheken [7].

## 8 Literatur

- [1] R. Sommer u.a., „Reuse-orientierter Ananalogschaltungsentwurf mit automatischer Dimensionierung und Ausbeuteoptimierung“, ANALOG 2003
- [2] F. Leyn u.a. „An Efficient DC Root Solving Algorithm with Guaranteed Convergence for Analog Integrated CMOS Circuits“, ICCAD, 1998
- [3] Hu u.a., „BSIM3v3.3 MOSFET Model“, University of California, Berkeley 2005
- [4] Cadence, „OCEAN Reference“, Version 5.1.41, June 2004
- [5] WiCkeD, [www.muneda.com](http://www.muneda.com)
- [6] F. Leyn u.a. „A dc solution algorithm with attractive numerical properties applicable for designing BJT circuits“, ProRISC, 2000
- [7] V. Boos, „Wiederverwendung von Schaltungstopologien“, DASS 2006