

Modellierung mit mehrdimensionalen Netzwerken

Joachim Haase, Peter Trappe
Joachim.Haase | Peter.Trappe@eas.iis.fraunhofer.de
Fraunhofer-Institut für Integrierte Schaltungen/Außenstelle EAS
Zeunerstr. 38, 01069 Dresden

Kurzfassung

Kirchhoffsche Netzwerke stellen ein wichtiges Hilfsmittel für die Modellierung technischer Systeme dar. Netzwerke, bei denen Fluss- und Differenzgrößen Vektoren sind, werden im Folgenden als mehrdimensionale Netzwerke bezeichnet. Die Verhaltensbeschreibungssprache VHDL-AMS erlaubt eine einfache Beschreibung solcher Netzwerke. Wie diese Möglichkeiten bei der Modellierung von Switched-Capacitor (SC) -Schaltungen für die Bestimmung von Frequenzgängen genutzt werden können, wird im Beitrag vorgestellt.

1 Einleitung

Netzwerkmodelle werden seit langem für unterschiedliche Modellierungsaufgaben eingesetzt. Voraussetzung für ihre Anwendung ist die Möglichkeit, Signalen Fluss- und Differenzgrößen (*across* und *through quantities*) zuzuordnen. In elektrischen Netzwerken sind das üblicherweise Ströme und Spannungen, in mechanisch-translatorischen Netzwerken z. B. Kräfte und Verschiebungen. Die Struktur eines Systems wird durch einen Netzwerkgraphen modelliert. Jedem Zweig werden eine Fluss- und eine Differenzgröße zugeordnet. Die Summe der Flussgrößen an einem Knoten und die Summe der Differenzgrößen in einer Masche müssen Null sein, d. h. die Kirchhoffschen Gesetze müssen erfüllt sein.

Sind die Werte von Fluss- und Differenzgrößen vektorielle Größen wird im Folgenden von mehrdimensionalen Netzwerken gesprochen. In mechanischen Netzwerken können so beispielsweise die Bewegungen in alle Raumrichtungen und um alle Koordinatenachsen berücksichtigt werden. Modellierungssprachen wie VHDL-AMS [1] erlauben die Erstellung von Verhaltensmodellen für mehrdimensionale Netzwerke und die Verwendung darauf aufbauender Modellierungsansätze. Dies soll in diesem Beitrag am Beispiel der Frequenzgangberechnung für Switched-Capacitor (SC)-Schaltungen gezeigt werden.

SC-Schaltungen sind getaktete Systeme, mit denen eine analoge Signalverarbeitung im Wesentlichen unter Verwendung von (MOS)-Schaltern, Kapazitäten und Operationsverstärkern stattfindet [2, 3]. Mit diesem Schaltungskonzept können u. a. Filter unter Verwendung von Komponenten integrierter Schaltungen realisiert werden. Das Verhalten eines Widerstandes, eines typischen Elements in einem analogen System, wird dabei durch eine Kombination von idealen Schaltern mit einer Kapazität nachgebildet. Die Schalter werden durch nichtüberlappende Taktimpulse angesteuert. Die Taktfrequenz, mit der die Schalter aktiviert werden, muss wesentlich größer als die Grenzfrequenz der zu übertragenden Signale sein. Für die Simulation im Zeitbereich kann

der nahe liegende Ansatz genutzt werden, das Öffnen und Schließen der Schalter direkt zu beschreiben [4, 5].

Nicht ganz so einfach ist die Situation bei der Ermittlung der Übertragungsfunktion im Frequenzbereich. Eine Beschreibung der Schaltung mit den üblichen Modellen für Kapazitäten, ideale Schalter und Operationsverstärker durch Linearisierung der Schaltung am Arbeitspunkt und anschließende Anwendung der Kleinsignalanalyse (AC-Analyse in Spice-ähnlichen Simulatoren) führt nicht zum Ziel. Spezielle Ansätze sind zur Lösung dieser Aufgabe entwickelt und in speziellen SC-Simulationsprogrammen implementiert worden [4].

Unterschiedliche Interpretationen dieser Verfahren für die Anwendung in allgemeinen Netzwerkanalyseprogrammen existieren [6, 7]. Ihre Anwendung führt aber leider nach Auffassung der Verfasser schnell zu sehr unübersichtlichen Beschreibungen [3]. Sehr einfach lassen sich diese Ansätze zur Frequenzbereichsberechnung aber unter Verwendung moderner Verhaltensbeschreibungssprachen wie VHDL-AMS umsetzen. Damit eröffnet sich gleichzeitig eine einfache Möglichkeit zur Einbeziehung von Frequenzgang-charakterisierungen in üblichen Simulationsabläufe.

2 Kleinsignalfrequenzgangsimulation von SC-Schaltungen

2.1 Modellierungsansatz

Betrachtet wird eine SC-Schaltung mit zwei Taktphasen. Eine Gruppe von (idealen) Schaltern wird in der ersten Taktphase PHI1, die andere Gruppe in der zweiten Taktphase PHI2 geschlossen. Die Topologie des SC-Netzwerkes ändert sich somit periodisch. Die Idee für die Modellierung besteht nun darin, jeden Zweig der ursprünglichen Beschreibung in zwei parallele Zweige eines zweidimensionalen Netzwerkes zu überführen. Die unterschiedlichen Topologien in den beiden Taktphasen werden durch die beiden Komponenten des so entstehenden Netzwerkes repräsentiert [4]. Für die Beschreibung des zweidimensionalen Netzwerkes für die Frequenzgangsimulation wird ein Ansatz mit Ladungen als Fluss- und Spannungen als Differenzgrößen im z-Bereich gewählt [6]. Bild 1 zeigt die Beschreibung für einen Kapazitätszweig.

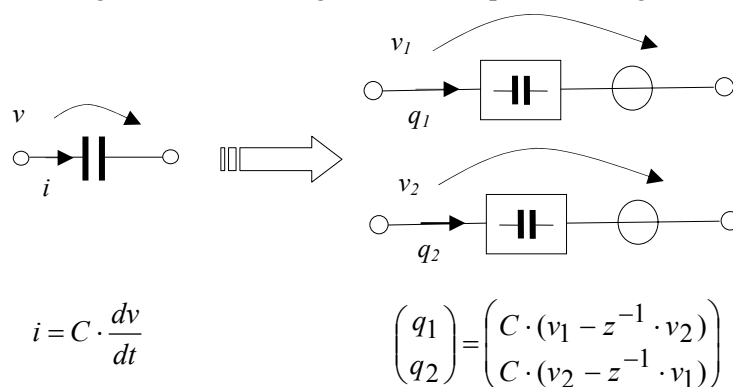


Bild 1: Beschreibung eines Kapazitätszweiges für die Frequenzgangsimulation

Die durch z^{-1} beschriebene Verzögerungszeit ergibt sich aus der halben Periode der Taktsignale für die Phasen PHI1 und PHI2 bei einem Duty Cycle von 0.5.

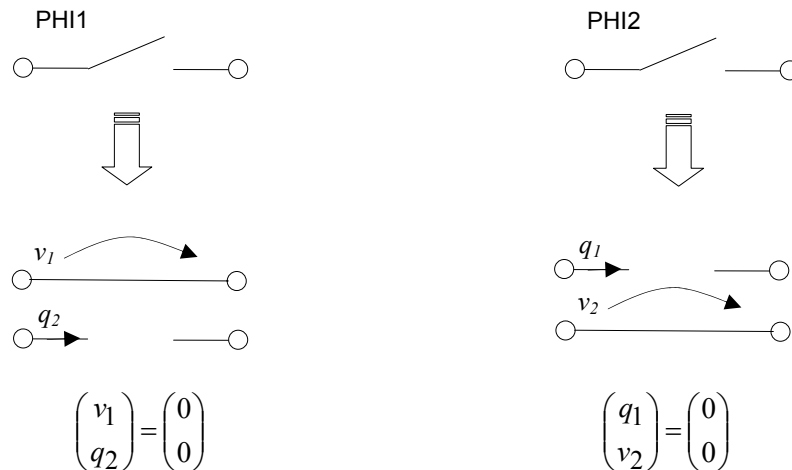


Bild 2: Beschreibung der Schalterzweige

Die Beschreibung der in den Taktphasen PHI1 bzw. PHI2 schließenden Schalter ist in Bild 2 angegeben. In analoger Weise können Modelle für Operationsverstärker erstellt werden. Der Ansatz lässt sich auf Schaltungen mit mehr Taktphasen durch analoge Überlegungen übertragen.

2.2 VHDL-AMS-Beschreibungen

Für die Beschreibung der Knoten im zweidimensionalen SC-Netzwerk wird in einem gegenüber der Darstellung in [8] erweiterten Package SC_DECLARATIONS eine Nature SC_NATURE_VECTOR deklariert. Die Periode für die Taktsignale in den Phasen PHI1 und PHI2 kann beispielabhängig im Package Body festgelegt werden. Das Package wird in eine logische Bibliothek SC_LIB übersetzt.

```
package SC_DECLARATIONS is
    nature SC_NATURE is
        REAL across
        REAL through
        SC_REF reference;
    nature SC_NATURE_VECTOR
        is array (NATURAL range <>) of SC_NATURE;
    subnature SC_2PHASE is SC_NATURE_VECTOR(1 to 2);
    constant T : REAL;    -- clock period for PHI1 and PHI2
end package SC_DECLARATIONS;
```

In diese Bibliothek SC_LIB werden weitere Grundmodelle für die SC-Simulation unter Verwendung des in Abschnitt 2.1 vorgestellten Ansatzes übersetzt. Genutzt werden können die folgenden Design Entitys:

- Kapazitätsmodell SC_CAP_2PHASE(A1) entsprechend Bild 2,
- Schaltermodelle SC_SWITCH_2PHASE(PHI1) und SC_SWITCH_2PHASE(PHI2) entsprechend Bild 3,
- Operationsverstärkermodell SC_OPAMP_2PHASE(A1),
- Modelle SC_VINPUT_2PHASE(A1) und SC_VOUTPUT_2PHASE(A2) zur Wandlung von elektrischen Spannungen in Spannungsbeschreibungen im SC-Netzwerk bzw. zur Wandlung in umgekehrter Richtung,
- Modell SC_REF_2PHASE(A1) zur Verbindung eines Knotens im SC-Netzwerk mit dem Bezugsknoten im SC-Netzwerk.

Exemplarisch ist im Folgenden die Beschreibung des Kapazitätsmodells entsprechend Bild 2 angegeben:

```

library SC_LIB;
use SC_LIB.SC_DECLARATIONS.all;

entity SC_CAP_2PHASE is
  generic ( CAP : REAL := 1.0);
  port (terminal P : SC_NATURE_VECTOR (1 to 2);
        terminal N : SC_NATURE_VECTOR (1 to 2));
end entity SC_CAP_2PHASE;

architecture A1 of SC_CAP_2PHASE is
  constant TSAMPL_2PHASE : REAL := SC_LIB.SC_DECLARATIONS.T/2.0;
  quantity V1 across Q1 through P(1) to N(1);
  quantity V2 across Q2 through P(2) to N(2);
  quantity V1_VAL, V2_VAL : REAL;
begin
  V1_VAL == V1;
  V2_VAL == V2;
  if DOMAIN = QUIESCENT_DOMAIN use
    Q1 == CAP*V1;
    Q2 == CAP*V2;
  else
    Q1 == CAP*(V1-V2_VAL'DELAYED(TSAMPL_2PHASE));
    Q2 == CAP*(V2-V1_VAL'DELAYED(TSAMPL_2PHASE));
  end use;
end architecture A1;

```

3 Beispiel: Nichtinvertierender Integrator

In Bild 3 ist die Schaltung eines nichtinvertierenden SC-Integrators angegeben. Das Verhältnis der Kapazitäten C1 und C2 beträgt 0.6283 : 1. Mit einer Taktfrequenz von 100 kHz ergibt sich damit eine Integrationsfrequenz von 10 kHz [3] (S. 542 f.). Die Frequenzgänge sind für unterschiedliche Werte der Verstärkung des Operations-

verstärkers ($gain = 10^2, 10^3, 10^4, \infty$) ermittelt worden. Die in Bild 4 dargestellten Ergebnisse entsprechen den Erwartungen. Die 0 dB-Durchgänge des Frequenzganges erfolgen bei 10 kHz, für die endlichen Verstärkungen des Operationsverstärkers ergeben sich bei kleinen Frequenzen die Werte 40 dB, 60 dB und 80 dB für die Amplitudenfrequenzgangwerte. Das ideale Verhalten ergibt sich bei unendlicher Verstärkung (GAIN = REAL'HIGH). Auf einem 750 kHz PC betrug die Rechenzeit für die Ermittlung des Frequenzganges für eine gegebene Verstärkung ungefähr eine Sekunde.

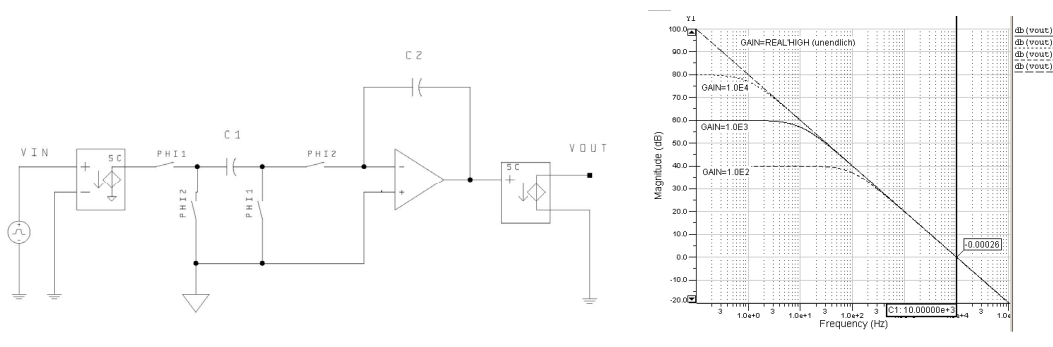


Bild 3: Nichtinvertierender Integrator und Amplitudenfrequenzgang für verschiedene Verstärkungen des Operationsverstärkers (GAIN=1.0E2, 1.0E3, 1.0E4, REAL'HIGH)

Ein Auszug aus der Netzliste zeigt die einfache Beschreibung der Struktur für die Frequenzgangberechnung:

```
architecture arch_INTEGRATOR of INTEGRATOR is

    terminal N1, N2, N3, N4, NOUT : SC_NATURE_VECTOR(1 to 2);
    terminal SC_GROUND           : SC_NATURE_VECTOR(1 to 2);
    ...

begin

    S1: entity SC_LIB.SC_SWITCH_2PHASE(PHI1)
        port map (P => N1, N => N2);

    S2: entity SC_LIB.SC_SWITCH_2PHASE(PHI2)
        port map (P => N2, N => SC_GROUND);

    GR: entity SC_LIB.SC_REF_2PHASE(A1) port map (P => SC_GROUND);

    C1: entity SC_LIB.SC_CAP_2PHASE(A1)
        generic map (CAP => 0.6283) port map (P => N2, N => N3);
    ...

    OV1 : entity SC_LIB.SC_OPAMP_2PHASE
        generic map (GAIN => REAL'HIGH)
        port map (CP => SC_GROUND, CN => N4, P => NOUT);
    ...

end architecture arch_INTEGRATOR;
```

4 Zusammenfassung

Die Verwendung mehrdimensionaler Netzwerke erlaubt die einfache Umsetzung von Ansätzen zur Frequenzgangbestimmung für SC-Schaltungen unter Verwendung von VHDL-AMS. Die Grundidee des Vorgehens und ein einfaches Beispiel wurden skizziert. Weitere Beispiele wurden unter Verwendung der Simulationsprogramme ADVance MS und SystemVision der Firma Mentor Graphics [9] gerechnet, und es zeigte sich eine gute Übereinstimmung der Simulationsergebnisse mit den erwarteten Resultaten. Die Rechenzeiten lagen im Sekundenbereich. Auch eine Simulation im Zeitbereich war mit dem vorgestellten Ansatz möglich. Der Vorteil des vorgestellten Weges besteht unter anderem darin, dass auf einfache Weise das Frequenzgangverhalten einer SC-Schaltung in die Kleinsignalfrequenzganganalyse eines größeren Systems einbezogen werden kann. Daneben eröffnet sich die Möglichkeit zur Anwendung der gesamten Funktionalität eines allgemeinen VHDL-AMS-Simulationsprogramms wie beispielsweise Variantenrechnungen, Monte-Carlo-Simulation und eventuell Optimierung für die Frequenzganguntersuchungen von SC-Schaltungen.

Literatur

- [1] *Christen, E.; Bakalar, K.*: VHDL-AMS – A Hardware Description Language for Analog and Mixed-Signal Applications. IEEE Trans. Circuits Syst. II 46 (1999) 10, pp. 1263–1272.
- [2] *von Wangenheim, L.*: Aktive Filter in RC- und SC-Technik. Heidelberg: Hüthig Buch Verlag, 1991.
- [3] *Allen, P. E.; Holberg, D. R.*: CMOS Analog Circuit Design. New York/Oxford: Oxford University Press, 2002.
- [4] *Vandewalle, J.; de Man, H. J.; Rabaey, J.*: Time, Frequency, and z-Domain Modified Nodal Analysis of Switched-Capacitor Networks. IEEE Trans. Circuits Syst. 28 (1981) 3, pp. 186-195.
- [5] *Yuan, F.; Opal, A.*: Computer Methods for Switched Circuits. IEEE Trans. Circuits Syst. I 50 (2003) 8, pp. 1013-1024.
- [6] *Kurth, C. F.; Moschytz, G. S.*: Two-Port Analysis of Switched-Capacitor Networks Using Four-Port Equivalent Circuits in the z-Domain. IEEE Trans. Circuits Syst. 26 (1976) 3, pp. 166-179.
- [7] *Kuo, Y.-L.; Liou, M. L.; Kasinskas, J. W.*: An Equivalent Circuit Approach to the Computer-Aided Analysis of Switched Capacitor Circuits. IEEE Trans. Circuits Syst. 26 (1979) 9, pp. 708-714.
- [8] *Haase, J.; Alber, J.; Jores, P.; Schwarz, P.; Trappe, P.*: Modellierung von SC-Schaltungen für die Systemsimulation. in GMM-Fachbericht 46 ANALOG '05 – Fachbeiträge der 8. GMM/ITG-Diskussionssitzung vom 16. bis 18. März 2005 in Hannover. Berlin/Offenbach: VDE Verlag GmbH, 2005, S. 229-234.
- [9] Informationen zu den Programmen ADVance MS und SystemVision. Verfügbar: http://www.mentor.com/products/ic_nanometer_design/simulation, <http://www.mentor.com/products/sm/systemvision>