

Regeln für die Erstellung von VHDL-AMS-Modellen

Joachim Haase

Fraunhofer-Institut Integrierte Schaltungen/Außenstelle Entwurfsautomatisierung
Zeunerstr. 38, 01069 Dresden

E-Mail: Joachim.Haase@eas.iis.fhg.de

VHDL-AMS hat seit seiner Definition als IEEE-Standard im Jahr 1999 zunehmend Verbreitung bei der Beschreibung digitaler, analoger und gemischt analog-digitaler Systeme gefunden. Ein Vorteil der Sprache besteht in den sehr mächtigen Sprachkonstrukten zur Beschreibung nutzereigener Modelle. Die damit verbundenen Freiheiten bei der Modellerstellung können aber auch dazu führen, dass syntaktisch richtige Simulationsprobleme formuliert werden, für die keine Lösung existiert. Gesichtspunkte, die bei der Modellierung analoger Teilsysteme mit VHDL-AMS eine Rolle spielen, werden im folgenden Beitrag diskutiert. Darauf aufbauend werden Regeln für die Modellierung abgeleitet.

1. Einleitung

Die Definition der Hardwarebeschreibungssprache VHDL-AMS liegt seit 1999 als IEEE Standard 1076.1-1999 vor [1]. Der Teil VHDL in der informellen Bezeichnung VHDL-AMS steht dabei für VHSIC (Very High-Speed Integrated Circuits) Hardware Description Language. AMS ist eine Abkürzung für Analog and Mixed-Signal. VHDL-AMS ist eine Sprache zur Beschreibung und Simulation analoger, digitaler und gemischt analog-digitaler Systeme [2]. Der Sprachstandard erweitert das digitale VHDL 1076-1993 um Sprachkonstrukte zur Modellierung analogen und gemischt analog-digitalen Verhaltens. Alle Prinzipien des digitalen VHDL bleiben erhalten. VHDL-AMS ist eine Obermenge des digitalen VHDL (siehe Bild 1).

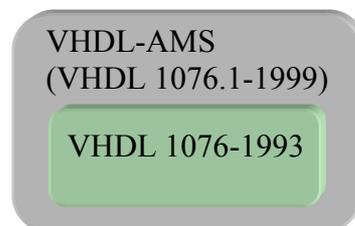


Bild 1: VHDL-AMS als Obermenge von VHDL 1076-1993

Seit der Standardisierung im Jahr 1999 erfreut sich VHDL-AMS zunehmender Beliebtheit. Eine Reihe kommerzieller Programme unterstützt die Simulation von VHDL-AMS-Beschreibungen (z. B. ADVance MS von Mentor Graphics, VeriasHDL von Synopsis, Smash von Dolpin). Teilweise werden auch abgerüstete oder kostenlose Versionen von VHDL-AMS-Simulatoren zur Verfügung gestellt, die sich für Lehr- und Lernzwecke eignen (z. B. SystemVision von Mentor Graphics, hAMSter von ANSOFT). Auch die Verfügbarkeit erster Lehrbücher spricht für eine weitere Verbreitung von VHDL-AMS [3], [4].

VHDL-AMS ist ebenso wie VHDL eine sehr mächtige Sprache. Gleiche Modellierungsprobleme können auf unterschiedliche Weise beschrieben werden. Um die Übersichtlichkeit und Austauschbarkeit von (digitalen) VHDL-Modellen zu erleichtern, gibt es seit längerem Bemühungen, Regeln für die Implementierung „guter“ VHDL-Modelle aufzustellen. Diese Regeln, die helfen sollen, den Modellaustausch zu erleichtern oder der weiteren Verwendung der Modelle, z. B. für Syntheseaufgaben, Rechnung zu tragen [5], zielen beispielsweise auf eine übersichtliche Gestaltung von Modellen. Daneben gibt es auch Empfehlungen die darauf ausgerichtet sind, Modelle so zu schreiben, dass unter Berücksichtigung des im VHDL-Standards beschriebenen Simulationszyklus und der Eigenschaften spezieller Simulatoren

eine effektive Simulation gewährleistet ist (siehe z. B. [6]). Derartige Regeln betreffen beispielsweise die günstige Gestaltung von Sensitivitätslisten von PROCESSEN, die Reduzierung von WAIT-Statements oder die Vermeidung überflüssiger Dateioperationen.

Diese Regeln verlieren natürlich nicht ihre Bedeutung bei der Erstellung von VHDL-AMS-Modellen. Daneben sind aber neue Probleme zu beachten. Digitale Teilsysteme werden in VHDL mit einem ereignisgesteuerten Simulationsalgorithmus ausgewertet. Signale können ihre Werte nur zu Zeitpunkten ändern, die in einem festen Zeitraster liegen. Zwischen diesen Zeitpunkten bleiben die Signalwerte konstant. Da nicht zu allen Rasterzeitpunkten Signalwechsel auftreten, ist eine Signalauswertung nur zu Zeiten potentieller Signaländerungen erforderlich. Anders ist die Situation bei der Beschreibung der analogen Teilsysteme. Aus der VHDL-AMS-Beschreibung der analogen Teilsysteme wird ein Algebraisdifferentialgleichungssystem (DAE-System; Differential Algebraic Equation-System) aufgestellt. Dieses wird diskretisiert. Die Zeitschrittweiten werden der geforderten Lösungsgenauigkeit angepasst. Probleme bei der Lösung ergeben sich aus den Eigenschaften der DAE-Lösungsalgorithmen [7]. Mit der zunehmenden Anwendung von VHDL-AMS ist auch das Interesse an Regeln und Methoden zur Vermeidung von Modellierungsproblemen gewachsen und hat seinen Niederschlag in einigen Veröffentlichungen zu dieser Thematik gefunden [8], [9]. Die Thematik sollte aber noch nicht als abgeschlossen behandelt betrachtet werden. Für den Anwender ist es sicher letztlich wünschenswert, dass ihm leistungsfähige Werkzeuge zur Verfügung gestellt werden, die ihm helfen, Probleme zu vermeiden und zu finden. Bevor es aber soweit ist, sollte versucht werden durch die Beachtung einiger Regeln, Schwierigkeiten zu vermeiden. Hierzu sollen im weiteren einige Anregungen gegeben werden.

Während sich digitale Signalwerte nur sprunghaft ändern können, erfolgt die Änderung analoger Zeitverläufe in der Regel stetig. Modelle sollten sprunghafte Veränderungen analoger Zeitverläufe während der Simulation, sogenannte Unstetigkeitsstellen, in der Regel vermeiden oder gezielt darauf reagieren. VHDL-AMS stellt für diesen Zweck das BREAK-Statement zur Verfügung. Ein anderer Problemkreis ergibt sich aus der Möglichkeit, Simulationsaufgaben zu formulieren, die zwar syntaktisch richtig gestellt sind, aber keine Lösung besitzen. An einem einfachen Beispiel wird das verständlich. Beispielsweise kann man ein System beschreiben, das zwei ideale parallel geschaltete Spannungsquellen enthält, die unterschiedliche Werte besitzen. Eine Lösung für ein derartiges System existiert nicht. Daneben gibt es Probleme, die aus der Wechselwirkung analoger und digitaler Teilsysteme entstehen. Digitale Signale können sich nur an fest vorgegebenen Rasterzeitpunkten ändern, analoge Zeitverläufe auch dazwischen. Das hat eventuell Konsequenzen für die Modellerstellung. In einem ersten Ansatz lassen sich mögliche Probleme mit VHDL-AMS-Modellen in drei Gruppen von Ursachen gliedern:

- Ungenügende Berücksichtigung der Lösungseigenschaften von Simulationsaufgaben
- Ungenügende Berücksichtigung des VHDL-AMS-Standards
- Unzureichende Berücksichtigung von Besonderheiten vorhandener VHDL-AMS-Simulatoren

Im folgenden wird versucht, Probleme anhand von Beispielen transparent zu machen und Regeln für ihre Vermeidung zu formulieren. Ausgangspunkt sind dabei die Vorgaben des VHDL-AMS-Standards, die einleitend noch einmal kurz zusammengefasst werden. Aufbauend auf bekannten mathematischen Aussagen wird versucht, die Regeln abzuleiten.

2. Simulationsaufgabe

2.1 Formulierung des Simulationsproblems

Analoger und digitaler Teil einer VHDL-AMS-Beschreibung werden getrennt ausgewertet. Während der Auswertung des Analogteils werden vom Simulator digitale Signale als konstant

angesehen. Entsprechend werden während der Auswertung des Digitalteils analoge Verläufe als konstant betrachtet. Attribute unterstützen den Zugriff auf Größen der jeweils anderen Beschreibungsebene (siehe Bild 2). Lösungseigenschaften des analogen Teils können bei dieser Betrachtungsweise weitgehend getrennt vom digitalen Teil betrachtet werden.

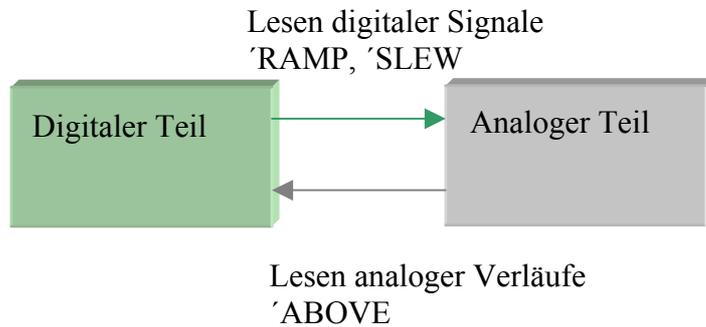


Bild 2: Attribute zur Unterstützung des Wertaustausches zwischen dem digitalen und analogen Teil einer VHDL-AMS-Beschreibung

Als Language Reference Manual (LRM) dient der IEEE Standard 1076.1 [1]. Die Beschreibung des analogen Verhaltens erfolgt mit charakteristischen Ausdrücken (*characteristic expressions*). Diese ergeben sich aus Forderungen aufgrund der Struktur (*structural set*) und der Modellgleichungen (*explicit set*) sowie zusätzlichen Bedingungen (*augmentation set*). Der analoge Lösungsalgorithmus bestimmt die Werte analoger Verläufe (*quantities*) so, dass an jedem analogen Lösungspunkt die charakteristischen Ausdrücke (Werte nahe) Null ergeben (LRM 12.1, 12.6.6).

In allen Phasen der Simulation müssen die Beziehungen, die aufgrund der Modellstruktur aufgestellt werden (*structural set*), erfüllt sein. Bei der Zusammenschaltung an konservativen Klemmen entsprechen diese Gleichungen den Kirchhoffschen Gesetzen (LRM 12.2.4, 12.3.1.4). Die analogen Verläufe an Knoten, an denen nichtkonservative Klemmen miteinander verbunden werden, müssen gleich sein. Nichtkonservative Klemmen sind z. B. Signalflussklemmen regelungstechnischer Blöcke.

Die Beschreibung der zwischen Fluss- und Differenzgrößen von Zweigen, zusätzlichen Größen zur Beschreibung des analogen Verhaltens (*free quantities*) und Ein- und Ausgangsverläufen nichtkonservativer Klemmen bestehenden Beziehungen erfolgt mit sogenannten *simultaneous statements*. Bei elektrischen Netzwerken lassen sich so beispielsweise die Strom-Spannungs-Beziehungen von Netzwerkzweigen formulieren. Jeder charakteristische Ausdruck, der sich aus einem *simultaneous statement* ergibt, gehört zu den explizit definierten Modellgleichungen (*explicit set*, LRM Abschnitt 15). Als notwendige Bedingung für die (eindeutige) Lösbarkeit des analogen Modellteils wird gefordert, dass die charakteristische Anzahl von *Quantities* gleich der Anzahl der *simultaneous statements* ist (LRM 12.6.6).

Verwendung von	Initialisierungsphase ($t=0$)	Unstetigkeitsstelle ($t=ts$)
Q1'DOT	$Q1'DOT = 0$	$Q1(ts-0) = Q1(ts+0)$ Q1 stetig
Q2'INTEG	$Q2 = 0$	$Q2'INTEG(ts-0) = Q2'INTEG(ts+0)$ Q2'INTEG stetig

Tabelle 1 : Standardmäßige zusätzliche Bedingungen (*augmentation set*)

In der Initialisierungsphase müssen zusätzliche Bedingungen erfüllt sein (quiescent state *augmentation set*, LRM 12.6.5.1). Standardmäßig sind alle Ableitungen Q'DOT von Quanti-

ties Null (siehe Tabelle 1). Wird ein Integral Q'INTEG verwendet, muss Q in der Initialisierungsphase Null sein. Diese Default-Zuweisungen können mit BREAK Anweisungen ersetzt werden (LRM 8.14, 12.6.6.1). Ein verzögerter Verlauf Q'DELAYED(T) einer Quantity Q nimmt während der Initialisierungsphase den Wert von Q an. Ähnliche Beziehungen gelten an Unstetigkeitsstellen (LRM 12.6.5.3). Standardmäßig ändert sich Q nicht, wenn Q'DOT verwendet wird. Q'INTEG bleibt unverändert. Aber auch diese Voreinstellungen können mit BREAK Anweisungen außer Kraft gesetzt werden. Durch die Erweiterungsmenge (*augmentation set*) können somit Zusatzbedingung für die Anfangswertberechnung vom Anwender sehr flexibel formuliert werden.

Im Ergebnis führt die Aufstellung der charakteristischen Ausdrücke (mit unwesentlichen Einschränkungen die Verwendung von 'INTEG und 'DELAYED betreffend) für die Simulation im Zeitbereich auf ein DAE-System

$$F(x, \tilde{x}', t) = 0 \text{ mit } x: [0, T] \rightarrow R^n \text{ und } F: R^{n+q+1} \rightarrow R^n \quad (1)$$

Dieses DAE-System ergibt sich aus den Kirchhoffschen Gesetzen und den Modellgleichungen (*structural set* und *explicit set*). q ist dabei die Anzahl der analogen Verläufe (*quantities*), auf die das Attribut 'DOT (oder 'INTEG) angewendet werden. In \tilde{x} sind die zugehörigen Komponenten von x zusammengefasst. In der Initialisierungsphase ($t=0$) und an Unstetigkeitsstellen ($t=ts$) sind noch zusätzliche Beziehungen (*augmentation set*)

$$B(x(0), \tilde{x}'(0)) = 0 \text{ mit } B: R^{n+q} \rightarrow R^q \quad (2.1)$$

bzw.

$$B(x(ts), \tilde{x}'(ts)) = 0 \quad (2.2)$$

zu erfüllen.

Aus der Auswertung dieser Gleichungssysteme sind die Differenzgrößen zwischen jedem Knoten und dem zugehörigen Bezugsknoten, die Differenz- und Flußgrößen von Zweigen und die sogenannten *free quantities* zu bestimmen. Bei einem rein elektrischen Modell entsprechen diesen Größen die Knotenspannungen sowie Zweigspannungen und –ströme und die *free quantities*. Im weiteren werden der Einfachheit halber nur diese in Bezeichnungen verwendet.

2.2 Lösungseigenschaften

Einige Aussagen zu den Lösungseigenschaften der durch einen VHDL-Simulator zu lösenden Gleichungssysteme, deren Aufstellung im Abschnitt 2.1 erläutert wurde, bei der Auswertung des Analogteils ergeben sich aus

- einem Satz über die Lösungseigenschaften linearer Gleichungssysteme [10],
- dem Satz über inverse Funktionen [11]
- und Aussagen über die Eigenschaften konsistenter Anfangswerte [12], [13].

Diese Sätze sind im Anhang zusammengestellt. Im folgenden werden die Konsequenzen für die Initialisierungsphase und die Zeitbereichssimulation diskutiert. Im Abschnitte 3 soll dann versucht werden, diese Konsequenzen in eine den Anwendern von Elektroniksimulationsprogrammen vertraute Sprache zu „übersetzen“. Um den Umfang dieser Darstellung nicht zu sprengen, wird auf die Diskussion des Verhaltens an Unstetigkeitsstellen verzichtet. Die Überlegungen für die Initialisierungsphase können in Prinzip auf die Berechnung rechtsseitiger Anfangswerte an Unstetigkeitsstellen übertragen werden.

2.2.1 Initialisierungsphase

In der Initialisierungsphase sind $x(0)$ und $\tilde{x}'(0)$ so zu bestimmen, dass sie das sich aus (1) und (2.1) ergebende nichtlineare Gleichungssystem

$$\begin{aligned} F(x(0), \tilde{x}'(0), 0) &= 0 \\ B(x(0), \tilde{x}'(0)) &= 0 \end{aligned} \quad (3)$$

lösen. Die Beziehungen (3) lassen sich in p lineare und $n+q-p$ nichtlineare Beziehungen trennen:

$$A \cdot (x(0), \tilde{x}'(0)) - b(0) = 0 \quad (4.1)$$

$$f(x(0), \tilde{x}'(0), 0) = 0 \quad (4.2)$$

mit $A \in R^{p \times (n+q)}$, $b \in R^p$ und $f: R^n \times R^q \times [0, T] \rightarrow R^{n+q-p}$. Damit ergeben sich folgende Folgerungen:

- Voraussetzung für die **Existenz** von *gültigen Anfangswerten* in der Initialisierungsphase ist die Widerspruchsfreiheit der Beziehungen von (3).
- Notwendige Bedingung für die **Existenz einer Lösung** ist bei Darstellung von (3) in der Form (4.1) und (4.2) aufgrund der Lösungseigenschaften linearer Gleichungssysteme (siehe Anhang)

$$\text{Rang}(A) = \text{Rang}(A|b(0)) \quad (5)$$

- Notwendige Bedingung für die **Eindeutigkeit von Anfangswerten** (in einer Umgebung der Lösung von (3)) ist die Regularität der Jacobi-Matrix von (3) am Lösungspunkt

$$\text{Rang} \begin{pmatrix} \frac{\partial F}{\partial x} & \frac{\partial F}{\partial \tilde{x}} \\ \frac{\partial B}{\partial x} & \frac{\partial B}{\partial \tilde{x}} \end{pmatrix} = n + q \quad (6)$$

- Bei DAE-Systemen (1) mit höherem Index m ergibt sich die Forderung, dass konsistente Anfangswerte neben (3) auch die ersten m Ableitungen des DAE-Systems erfüllen müssen (siehe Anhang).

2.2.2 Simulation im Zeitbereich

Ausgehend von den ermittelten Anfangswerten wird eine Lösung von (1) numerisch bestimmt. Dazu wird in der Regel ein lineares Mehrschrittverfahren verwendet [7]. Nach Diskretisierung von (1) ist ein nichtlineares Gleichungssystem zur Bestimmung der Lösungspunkte zu jedem analogen Simulationszeitpunkt zu lösen.

Notwendige Bedingung für die **Eindeutigkeit der Lösung** (in einer Umgebung eines Lösungspunktes) ist die Regularität der Jacobi-Matrix. Diese Forderung sollte beispielsweise bei Anwendung des impliziten Euler-Verfahrens erfüllt sein

$$\text{Rang} \left(\frac{\partial F}{\partial x} + \frac{1}{h} \cdot \frac{\partial F}{\partial x'} \right) = n \quad (7)$$

h ist die Schrittweite. Die Ableitung von F nach x' ergibt sich aus der Ableitung von F nach \tilde{x}' und der Ergänzung der zu entstandenen Matrix durch Nullspalten für nicht in \tilde{x} berücksichtigte Komponenten von x .

Die Beziehung (1) lässt sich in \bar{p} lineare und $n - \bar{p}$ nichtlineare Beziehungen trennen:

$$\bar{A} \cdot (x(t), \tilde{x}'(t)) - b(t) = 0 \quad (8.1)$$

$$\bar{f}(x(0), \tilde{x}'(0), 0) = 0 \quad (8.2)$$

mit $\bar{A} \in R^{\bar{p} \times (n+q)}$, $b(t) \in R^{\bar{p}}$ und $\bar{f} : R^n \times R^q \times [0, T] \rightarrow R^{n-\bar{p}}$. Notwendige Voraussetzung für die Gültigkeit von (7) ist dann

$$\text{Rang}(\bar{A}) = \bar{p} \quad (9).$$

3. Regeln für die Modellerstellung

3.1 Allgemeine Regeln

Voraussetzung für die eindeutige Lösung der Simulationsaufgabe (1) (vgl. Abschnitt 2.1) ist die Möglichkeit zur eindeutigen Bestimmung aller Knotenspannungen, Zweigspannungen und –ströme und aller *free quantities*. Notwendige Bedingungen dafür sind die Einhaltung folgender Regeln

- Von jedem Knoten muß es einen Weg von Zweigen zum zur entsprechenden NATURE (vgl. LRM 3.5) gehörenden Bezugsknoten geben. Andernfalls können die Knotenspannungen nicht eindeutig bestimmt werden. *Anmerkung:* In einer SPICE-Beschreibung entspricht diese Forderung der mindestens einmaligen Verwendung des Knotens 0 in einer Netzliste.
- Bei einer idealen Spannungsquelle ist die Zweigspannung unabhängig vom Zweigstrom. Maschen von idealen Spannungsquellen sind zu vermeiden. *Anmerkung:* Bei Überführung der in (7) verwendeten Jacobi-Matrix in eine *row echelon*-Form ergibt sich andernfalls für die Ströme in den entsprechenden Zweigen eine Nullspalte. Die zugehörigen Zweigströme können nicht eindeutig bestimmt werden.
- Analog sind Schnitte aus idealen Stromquellen zu vermeiden.
- Kirchhoffsche Gesetze dürfen nicht als *simultaneous statements* noch einmal formuliert werden. *Anmerkung:* (1) ergibt sich aus *structural* und *explicit set* von charakteristischen Gleichungen. Eine Verletzung der Regel würde eine Dopplung von Gleichungen in (1) zur Folge haben. Daraus folgt ein Rangabfall bei der in (7) verwendeten Jacobi-Matrix. Die Simulationsaufgabe ist nicht mehr eindeutig lösbar.

3.2 Simulation im Zeitbereich

Eine notwendige Bedingung für die eindeutige Lösbarkeit der Simulationsaufgabe ergibt sich aus (7) in Abschnitt 2.2.2. Es folgt:

- Die Jacobi-Matrix des in der Zeitbereichssimulation diskretisierten Gleichungssystems darf keinen Rangabfall (Rang kleiner n) haben.

Beispiel 3.2-1:

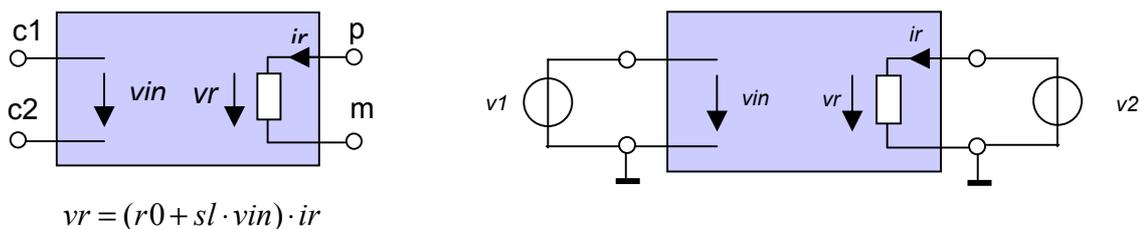


Bild 3: Spannungsgesteuerter Widerstand und Testschaltung

Betrachtet wird das in Bild 3 dargestellte resistive Netzwerk. Die Ableitung von F nach x' ist Null. Die obige Regel ist für die Testschaltung für $v1 = -\frac{r0}{sl}$ verletzt. In diesem Fall kann keine eindeutige Lösung für die Testschaltung ermittelt werden. Ob überhaupt eine Lösung existiert, hängt vom Wert von $v2$ ab. Wie leicht zu überprüfen ist, gibt es bei dem genannten Wert für $v1$ unendliche viele Lösungen, wenn $v2=0$ ist. Ansonsten existiert gar keine Lösung.

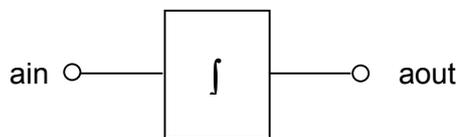
3.3 Initialisierungsphase

Es ergibt sich aufgrund der Überlegungen aus Abschnitt 2.2.1 die Regel

- Notwendige Bedingung für die Ermittlung von Anfangswerten ist die **Widerspruchsfreiheit** der sich aus *structural set*, *explicit set* und *augmentation set* ergebenden Beziehungen der Form (3).

Anmerkung: Notwendig für die eindeutige Lösbarkeit ist Gültigkeit von (6) (siehe auch [9]).

Beispiel 3.3-1:



$$a_{in} = \frac{1}{ki} \cdot \frac{da_{out}}{dt}$$

```
entity integrator is
  generic (ki : real := 1.0;
           ic : real := real'low);
  port    (quantity ain  : in real;
           quantity aout : out real);
end entity integrator;

architecture a1 of integrator is
  quantity ahelp : real;
begin
  ahelp == aout;
  ain   == 1.0/ki*ahelp'dot;
end architecture a1;
```

Bild 4: Integrator und VHDL-AMS-Modell

Da standardmäßig in der Initialisierungsphase bei Verwendung der Architecture *a1* die zeitliche Ableitung von a_{out} zur Zeit Null verschwindet, muß auch a_{in} zur Zeit Null Null sein. Somit kann eine Lösung bei Verwendung dieses Modells **nur** ermittelt werden, wenn das Modell so verwendet wird, dass der Wert am analogen Eingang *ain* bei der Arbeitspunktbestimmung in der Initialisierungsphase Null ist. a_{out} ist nicht eindeutig bestimmt.

```
architecture a3 of integrator is
  quantity ahelp : real;
begin
  ahelp == aout;
  if domain = quiescent_domain use
    if ic /= real'low use
      aout == ic;
    else
      aout == 0.0;
    end use;
  else
    ain == 1.0/ki*ahelp'dot;
  end use;
end architecture a3;
```

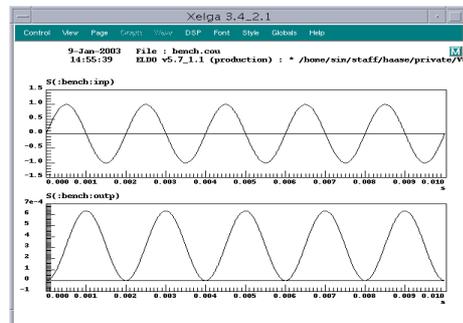
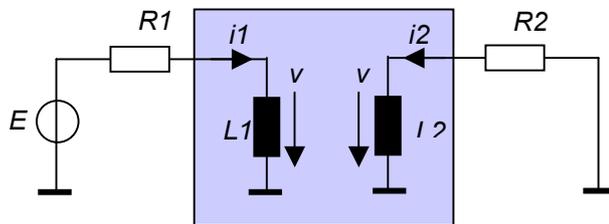


Bild 5: Simulationsergebnisse bei $a_{in}(0)=0$

Bild 5 zeigt die Ergebnisse einer Simulation, bei der an *ain* mit dem analogen Verlauf *inp* (Diagramm oben) angeregt wird. Dieses Problem wird bei Anwendung der Architecture *a3* umgangen. In diesem Fall ist auch unter Verwendung des Anfangswertes *ic* die Ausgangsgröße *aout* eindeutig bestimmt. Für eine Testschaltung, in der der Integrator mit einer analogen Eingangsgröße ergänzt wird, ist die Beziehung (6) stets erfüllt.

Beispiel 3.3-2:

In Bild 6 ist eine Testschaltung mit ideal gekoppelten Spulen dargestellt. Bei beliebiger Vorgabe der Induktivitätsströme i_1 und i_2 zum Zeitpunkt Null, mit denen die Standardbedingungen (vgl. Tabelle 1) ersetzt werden, können Anfangswerte ermittelt werden. Die oben angegebene Regel ist erfüllt.



$$v_1 = L_1 \cdot \frac{di_1}{dt} + \sqrt{L_1 \cdot L_2} \cdot \frac{di_2}{dt}$$

$$v_2 = \sqrt{L_1 \cdot L_2} \cdot \frac{di_1}{dt} + L_2 \cdot \frac{di_2}{dt}$$

Bild 6: Gekoppelte Spulen

Allerdings sind diese Anfangswerte nur konsistent, wenn zusätzlich die Bedingung $i_1(0) = \frac{L_1}{\sqrt{L_1 \cdot L_2}} \cdot \frac{R_2}{R_1} \cdot i_2(0) + \frac{E(0)}{R_1}$ erfüllt ist. Diese Problematik wird bereits in [14] in Zusammenhang mit entarteten Systemen diskutiert. Das Beispiel zeigt, dass die Erfüllung der Beziehungen (3) für die Konsistenz von Anfangswerten notwendig, aber nicht in jedem Fall hinreichend ist.

3.4 Weitere Regeln

Weitere Regeln ergeben sich aus Besonderheiten des VHDL-AMS-Standards [1]. Diese hängen beispielsweise mit der Initialisierung von Signalen zusammen. Werden diese nicht explizit bei der Deklaration initialisiert, wird Signalen vom Typ T der Wert T'LEFT zugewiesen (LRM 4.3.1.2). Analoge Verläufe (*quantities*) werden standardmäßig mit Null initialisiert (LRM 4.3.1.6). Daraus resultierende Folgerungen sind z. B. in [8] diskutiert. Die Signalaktualisierung erfolgt nur zu Ereigniszeitpunkten. Zu beachten sind deshalb folgende Regeln

- Die explizite Initialisierung reellwertiger Signale, die in *simultaneous statements* gelesen werden, ist erforderlich.
- *Simultaneous statements* müssen auch für Werte von analogen Verläufen und Signalen im Zuge des Lösungsalgorithmus ausgewertet werden können, die letztlich keine Lösungen der Simulationsaufgabe sind.
- Es ist zu beachten, dass die Aktualisierung von Signalen nur zu Ereigniszeitpunkten erfolgt (siehe dazu Beispiel 3.4-1).

Beispiel 3.4-1:

```
library ieee;
  use ieee.math_real.all;

entity bench is end entity bench;

architecture does_not_go of bench is
  signal vs : real := 0.0;
  quantity vq : real;
begin
  vq == 5.0*sin(math_2_pi*1.0E3*now);
  vs <= vq;
end architecture does_not_go;
```

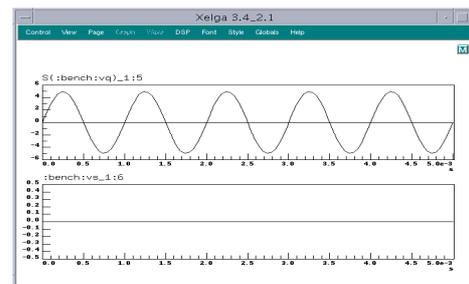


Bild 7: VHDL-AMS-Beschreibung ohne Signalaktualisierung

```
architecture goes of bench is
    signal vs : real := 0.0;
    signal clk : boolean;
    quantity vq : real;
begin
    vq == 5.0*sin(math_2_pi*1.0E3*now);
    clk <= not clk after 100 us;
    vs <= vq when clk'event;
end architecture goes;
```

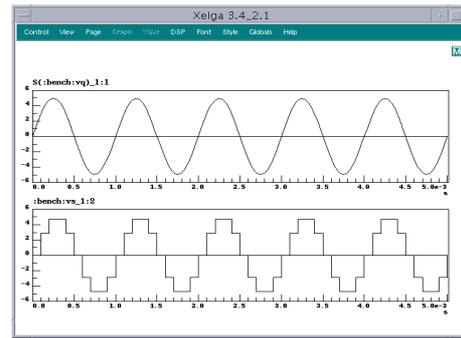


Bild 8: VHDL-AMS-Beschreibung mit Signalaktualisierung zu Ereigniszeitpunkten

Das Beispiel demonstriert, dass bei Zuweisung einer Quantity zu einem Signal Signalwechsel nur zu Ereigniszeitpunkten erfolgen. Gibt es keine digitalen Ereignisse bleibt das Signal konstant (Bild 7). Ansonsten erfolgt die Zuweisung, zu den Zeitpunkten, an denen sie durch ein Signal aktiviert wird (Bild 8).

4. Zusammenfassung

Ausgehend vom VHDL-AMS-Standard und mathematischen Sätzen über die Lösungseigenschaften von Gleichungssystemen und DAE-System wurden Regeln zur Formulierung von VHDL-AMS-Modelle abgeleitet. Eine formale Prüfung dieser Regeln würde durch Zugriff auf die von VHDL-AMS-Simulatoren aufgestellten Gleichungssysteme erleichtert. Dies ist bei kommerziellen Simulatoren für den normalen Anwender nicht möglich. So bleibt es erst einmal dem Anwender überlassen, die vorgestellten Regeln bei der Modellierung zu beachten. Das hilft aber, eine Reihe von Modellierungs- und Simulationsproblemen zu vermeiden.

Nicht diskutiert wurden in diesem Rahmen Besonderheiten, die sich aus den verwendeten Simulatoren ergeben. Ein gegenwärtig noch vorhandenes Problem ist der unterschiedliche Grad der Abdeckung des VHDL-AMS-Standards durch verfügbare Simulatoren. Diesen Unterschieden sollte insbesondere dann Beachtung geschenkt werden, wenn es um den Austausch von Modellen zwischen unterschiedlichen Simulatoren geht.

Die Beispielrechnungen erfolgten mit dem Simulator ADVance MS Version 1.5 der Firma Mentor Graphics.

Literatur

- [1] IEEE Standard VHDL Analog and Mixed-Signal Extensions. IEEE Std 1076.1-1999. Design Automation Standards Committee of the IEEE Computer Society, March 18, 1999.
- [2] Christen, E.; Bakalar, K.: VHDL-AMS – A Hardware Description Language for Analog and Mixed-Signal Applications. IEEE Transactions on CAD-II 46(1999)10, 1263-1272.
- [3] Hervé, Y.: VHDL-AMS Application et enjeux industriels. Paris : Dunod, 2002.
- [4] Ashenden, P. J.; Peterson, G. D.; Teegarden, D. A.: The System Designer's Guide to VHDL-AMS. Morgan Kaufmann Publishers, 2002.
- [5] Keating, M.; Bricaud, P.: Reuse Methodology Manual for System-on-a-Chip-Designs. Boston: Kluwer Academic Publishers, 2002.
- [6] VHDL Style Guidelines for Performance by Model Technology Product Support. Last modified: Oct 01, 1999. (http://www.model.com/resources/appnotes/vhdl_style_guidelines.asp)

- [7] Christen, E.: Selected Topics in Mixed-Signal Simulation. Proc. FDL '02, September 24-27, 2002, Marseille.
- [8] Ruan, K. G.: Initialization of Mixed-Signal Systems in VHDL-AMS. Proc. BMAS 2001, October 10-12, 2001, Fountain Grove Inn Santa Rosa, CA.
(<http://www.esat.kuleuven.ac.be/~bmas/web-docs/Repository/Y2001/main.html>)
- [9] Mades, J.; Glesner, M.: Regularization of hierarchical VHDL-AMS models using bipartite graphs. Proc. DAC 2002, June 10-14, 2002, New Orleans, LA, 548-551.
(<http://portal.acm.org/citation.cfm?doid=513918.514056#>)
- [10] Meyberg, K. ; Vachenaer, P. : Höhere Mathematik 1. Berlin-Heidelberg-New York: Springer-Verlag, 1990.
- [11] Schwetlick, H.: Numerische Lösung nichtlinearer Gleichungen. Berlin: Deutscher Verlag der Wissenschaften, 1979.
- [12] Pantelides, C. C. : The Consistent Initialization of Differential-Algebraic Systems. SIAM J. Sci. Stat. Comp. 9(1988)2, 213-231.
- [13] Leimkuhler, B.; Petzold, L. R.; Gear, C. W.: Approximation Methods for the Consistent Initialization of Differential-Algebraic Equations. SIAM J. Numer. Anal. 28(1991)1, 205-226.
- [14] Berg, L.: Einführung in die Operatorenrechnung. Berlin: Deutscher Verlag der Wissenschaften, 1962.

Anhang

Satz (Lösbarkeitstest für lineare Gleichungssysteme, [10, S. 260])

Sei $A \in R^{m \times n}$, $b \in R^m$. Das inhomogene Gleichungssystem $A \cdot x = b$ ist genau dann lösbar, wenn gilt

$$\text{Rang}(A|b) = \text{Rang}(A) \quad .$$

Folgerung zum Satz über die inverse Funktion [11, S. 52]

Die Funktion $F : D \subset R^n \rightarrow R^n$ sei in einer Umgebung von $x^* \in \text{int}(D)$ differenzierbar, F' sei in x^* stetig, $F'(x^*)$ sei regulär, und es gelte $F(x^*) = 0$. Dann gibt es Zahlen $M > 0$, $\xi_0 > 0$, so dass $X_0 := \bar{S}(x^*, \xi_0) \subset D$ gilt, und für alle $x \in X_0$ gilt

- (i) $\|F'(x)^{-1}\| \leq M$
- (ii) $\|x - x^*\| \leq M \cdot \|F(x)\|$.

Anmerkung: Aus (ii) folgt insbesondere, dass x^* die einzige Nullstelle von F in X_0 ist.

Definition (konsistenter Anfangswertvektor, [13, S. 210])

Ein konsistenter Anfangswertvektor (x_0, x_0') für die DAE (differential-algebraic equation) mit Index m

$$F(x, x', t) = 0 \text{ mit } x : [t_0, T] \rightarrow R^n \text{ und } F : R^{2n+1} \rightarrow R^n \text{ (} C^l \text{-Abbildung)}$$

bei $t=t_0$ ist ein solcher, bei dem die DAE eine C^{m+1} Lösung besitzt, die durch diesen Vektor verläuft.

1. *Anmerkung:* Im allgemeinen besitzt der Anwender Informationen über den Anfangszustand in Form von algebraischen Gleichungen

$$(A.1) \quad B(x_0, x_0') = 0 .$$

Zusätzlich zu diesen anwenderspezifischen Beziehungen muss jeder gültige Anfangswert die DAE selbst erfüllen:

$$(A.2) \quad F(x_0, x_0', t_0) = 0 .$$

Ebenso müssen in Abhängigkeit von der Größe des Index die ersten m Ableitungen der DAE die folgenden Beziehungen erfüllen:

$$(A.3) \quad DF(x_0, x_0', x_0'', t_0) = \frac{\partial F}{\partial x} \cdot x_0' + \frac{\partial F}{\partial x'} \cdot x_0'' + \frac{\partial F}{\partial t} = 0 ,$$

...

$$(A.4) \quad D^m F(x_0, x_0', x_0'', \dots, x_0^{(m+1)}, t_0) = 0 .$$

Die partiellen Ableitungen von F werden an der Stelle (x_0, x_0') bei $t=t_0$ für eine Wahl von $(x_0'', x_0''', \dots, x_0^{(m+1)})$ ermittelt. Die $m+2$ Beziehungen (A.1) bis (A.4) werden als Konsistenzgleichungen bezeichnet.

2. *Anmerkung:* Bei VHDL-AMS-Beschreibungen ist t_0 stets 0 (LRM 12.6.4).

Satz (Anfangswertproblem, [13, S. 211])

Es werden angenommen, das Problem

$$(I) \quad \begin{aligned} F(x(t), x'(t), t) &= 0 \\ B(x(t_0), x'(t_0)) &= 0 \end{aligned}$$

habe genau eine stetige Lösung $\bar{x}(t)$ und diese sei glatt: $\bar{x}(t) \in C^{m+1}[t_0, T]$. Dann besitzen die Konsistenzgleichungen (A.1) bis (A.4) eine Lösung $(x_0, x_0', x_0'', x_0''', \dots, x_0^{(m+1)})$ und die ersten beiden Komponenten sind eindeutig bestimmt: $x_0 = \bar{x}(t_0)$, $x_0' = \bar{x}'(t_0)$.