

„Multi-Architecture-Modeling“ Entwurfsmethode für Mixed-Signal- und Multi-Domain-Systemsimulation

Michael Schlegel, Göran Herrmann, Dietmar Müller
Professur Schaltungs- und Systementwurf, TU Chemnitz, Deutschland
e-mail: michael.schlegel@infotech.tu-chemnitz.de

Kurzfassung

Im Entwurfsprozeß von Mikrosystemen können digitale, analoge elektrische und nichtelektrische Modelle auf unterschiedlichen Abstraktionsebenen entstehen. Der Abstraktionsgrad eines Modells bestimmt in der Regel den Abstraktionsgrad der Schnittstellen. Dadurch kann es während des Systementwurfs dazu kommen, daß die Schnittstellen der Modelle mit jedem Entwicklungsschritt auf Grund der damit verbundenen Verringerung des Abstraktionsgrades angepaßt werden müssen. Das Ziel der in diesem Beitrag vorgestellten Arbeit ist es, einen methodischen Ansatz zur effizienten Schnittstellengestaltung von Modellen zu finden. Dieser Ansatz soll es ermöglichen, die Schnittstellen eines Modells während des Entwurfsprozesse unverändert zu lassen, womit die zeitaufwendige und fehleranfällige Anpassung der Schnittstellen entfallen kann.

1 Einführung

Beim Entwurf digitaler Systeme erlaubt VHDL die Beschreibung mehrerer Architekturen (*architecture*) für eine Schnittstellenbeschreibung (*entity*). Mit der Erweiterung von VHDL zu VHDL-AMS (*VHDL Analog and Mixed Signal*) steht diese Eigenschaft von VHDL auch für die Beschreibung analoger und heterogener Systeme zur Verfügung. In einem Top-Down-basierten Systementwurf entstehen zu Beginn Modelle auf einem hohen Abstraktionsgrad. Der Abstraktionsgrad der Modelle verringert sich mit Fortschreiten des Entwurfsprozesses. Durch die Eigenschaft von VHDL, für eine Entity mehrere Architekturen beschreiben zu können, ist es nun möglich, Architekturen von Komponenten heterogener Systeme auf unterschiedlichen Abstraktionsebenen für ein Modell zu hinterlegen. Dabei kann bisher das Problem auftreten, daß die Architekturen der Modelle auf den unterschiedlichen Abstraktionsebenen unterschiedliche Schnittstellen benutzen.

Bild 1 verdeutlicht diese Problematik anhand eines Beispiels, bei dem zunächst Modelle auf der Basis funktioneller Blöcke beschrieben werden. Als Interface kommen hier nichtkonservative Knoten (*QUANTITY*) zum Einsatz. Während des Entwurfsprozesses wird die rechte Komponente zu einer Netzwerkbeschreibung weiterentwickelt, die linke dagegen nicht. Die weiterentwickelte (verfeinerte) Komponente benutzt nun als Interface konservative Knoten (*TERMINAL*) während die unveränderte Komponente nach wie vor *QUANTITIES* benutzt. Beide Modell können dadurch nicht mehr miteinander verbunden werden (weitere Details s. Abschnitt 2.2).

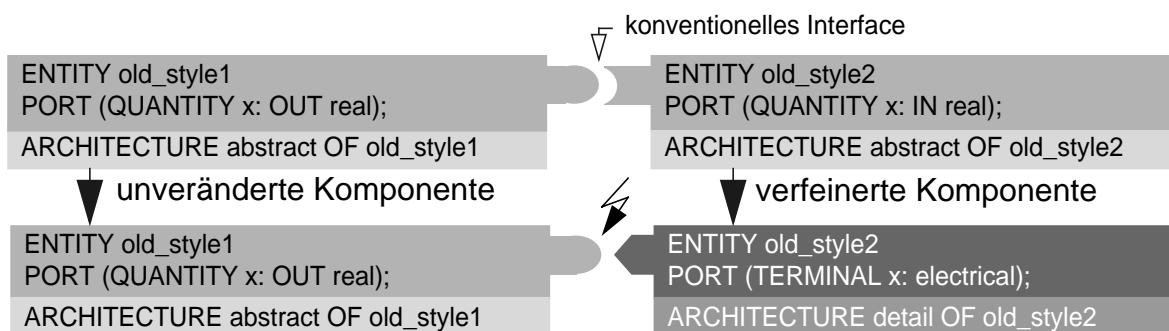


Bild 1 Problem beim Verbinden von Modellen mit unterschiedlichen Abstraktionsgraden

In den folgenden Abschnitten wird daher ein neuer methodischer Ansatz – das „Multi-Architecture-Modeling“ (MAM) – präsentiert. Wird dieser Ansatz bereits beim Entwurf der ersten abstrakten Modelle im Rahmen eines Top-Down Designs berücksichtigt, so können die später im Entwurfsprozeß entstehenden verfeinerten Modelle problemlos an Stelle der abstrakten Modelle eingesetzt werden. Dabei soll dieser Ansatz leicht anwendbar sein und nur einen geringen Mehraufwand bei der Modellierung hervorrufen.

2 Probleme bei der Schnittstellengestaltung und deren Lösung

2.1 Digitale Schnittstellen

Für digitale Schnittstellen finden SIGNALS Verwendung. Der Abstraktionsgrad hat dabei einen Einfluß auf den verwendeten Datentyp. So können z. B. auf funktioneller oder algorithmischer Ebene Datentypen wie `integer` oder `real` verwendet werden, im Gegensatz dazu sind auf RTL-Ebene Vektoren der Datentypen `bit` oder `std_logic` üblich.

Für die MAM sind nun zwei Ansätze denkbar: Entweder man verwendet die Schnittstelle, die das Modell auf hohem Abstraktionsniveau besitzt auch für die weiteren Modelle auf niedrigem Abstraktionsniveau, oder man benutzt die Schnittstelle des detaillierteren Modells bereits auf hoher Abstraktionsebene. Dabei hat sich der erste Ansatz als nicht praktikabel erwiesen. Er hätte zur Folge, daß innerhalb eines Modells auf RTL-Ebene Werte aus einer mehrwertigen Logik (z. B. 'X' oder 'Z') entstehen, die über ein Interface vom Typ `integer` oder `real` nicht zum nächsten Modell transportiert werden können.

Dagegen ist der zweite Ansatz anwendbar und bedeutet für digitale Schnittstellen die Verwendung der Datentypen `bit_vector` oder eines Vektors einer mehrwertigen Logik bereits im ersten abstrakten Modell. Gegenüber der Verwendung der Datentypen `integer` oder `real` erzeugt dies nur einen sehr geringen Mehraufwand. Eine evtl. notwendige Konvertierung der Datentypen zu `integer` oder `real` kann u. U. zusätzliche Delta-Zyklen im Simulator verursachen. Zustände wie 'X', 'Z' usw. werden außerhalb des Modells korrekt propagiert, im Modell kann die Handhabung entsprechend der Anforderungen an das Modell frei implementiert werden.

Die MAM bezieht sich lediglich auf den zu verwendenden Objekt- bzw. Datentyp. Eine Transformation von Protokollen über Abstraktionsebenen hinweg [6] wird von der MAM nicht erfaßt.

2.2 Analoge elektrische und nichtelektrische Schnittstellen

Abstrakte analoge Komponenten werden oft als funktionelle Blöcke beschrieben [3]. Als Interface-Objekt kommen dabei in VHDL-AMS sog. QUANTITIES zum Einsatz. Diese repräsentieren einen zeit- und wertkontinuierlichen Informationsverlauf. Im Gegensatz dazu benutzt ein Modell auf niedrigerem Abstraktionsgrad TERMINALS als Interface. TERMINALS enthalten eine ACROSS und eine THROUGH QUANTITY (z. B. Spannung und Strom) und erfüllen die kirchhoffschen Gesetze. Da man TERMINALS nicht direkt mit QUANTITIES verbinden kann, müssen bislang während des Entwurfsprozesses die Schnittstellen der Modelle geändert werden, wenn z. B. funktionelle Blöcke mit Netzwerken verbunden werden sollen.

Wie schon bei den digitalen Schnittstellen hat es sich auch hier als ungünstig erwiesen, die Schnittstelle, die auf hohem Abstraktionsniveau verwendet wird, auch in dem Modell auf niedrigem Abstraktionsniveau zu benutzen. Der Einsatz von QUANTITIES auf niedrigem Abstraktionsniveau hätte zur Folge, daß ACROSS und THROUGH QUANTITY separat im Interface angelegt und die kirchhoffschen Knoten- und Maschenregeln explizit in jedem Modell beschrieben werden müßten.

Die Verwendung von TERMINALS für analoge Schnittstellen auf allen Abstraktionsebenen dagegen ist möglich.

Bild 2 zeigt anhand eines Beispiels die Anwendung der MAM beim Entwurf analoger Schnittstellen. Die z. B. als funktioneller Block beschriebenen Modelle erhalten als Interface im Gegensatz zu Bild 1 keine QUANTITY sondern gleich ein TERMINAL. Die Anpassung innerhalb des Modells an dieses Interface besteht lediglich aus der Definition von Branch-QUANTITIES (ACROSS und THROUGH QUANTITY). Wird nun wie in Bild 1 eine Komponente verfeinert, die andere aber nicht, so können beide Modelle immer noch problemlos miteinander verbunden werden.

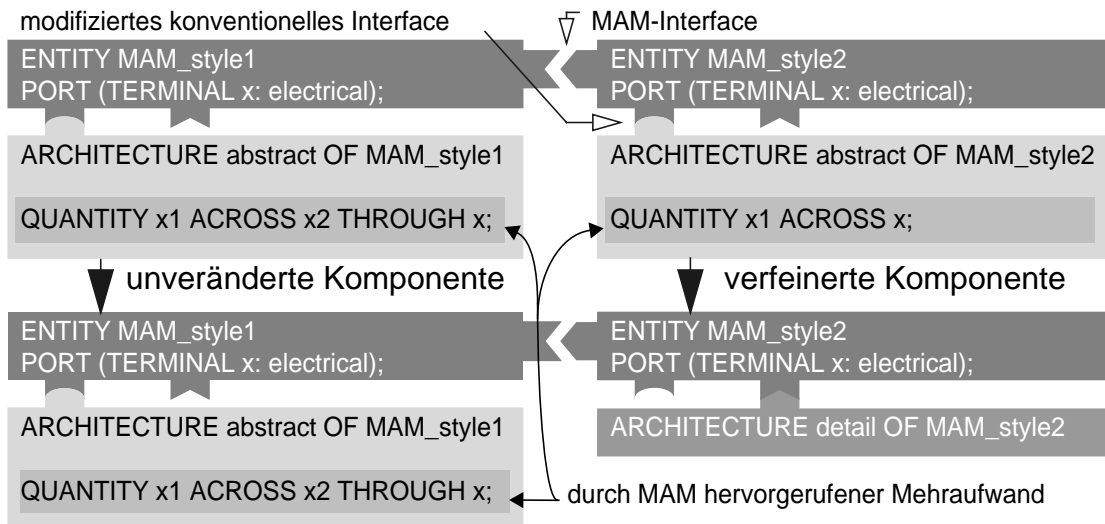


Bild 2 Verbindung analoger Modelle mittels MAM

Eine zu übertragende QUANTITY wird dabei als ideale Spannungs- oder Stromquelle zwischen dem TERMINAL und dem Masseknoten electrical_ground beschrieben. In der Empfängerkomponente wird die übertragene QUANTITY als Spannung bzw. Strom zwischen dem TERMINAL und electrical_ground ermittelt. Dieser Ansatz verursacht nur einen geringen Mehraufwand in der Modellierung und ist leicht anzuwenden.

2.3 Digital/analoge Schnittstellen

Im Rahmen einer Variantendiskussion (z. B. Ausführung eines Filters analog, digital oder mechanisch) oder in zeitkritischen digitalen Systemen kann es erforderlich sein, daß Schnittstellen auf hohem Abstraktionsniveau digital, auf niedrigem Niveau aber analog beschrieben werden müssen. Da digitale Schnittstellen aber SIGNALS benutzen und analoge Schnittstellen TERMINALS, muß ein gemeinsames Interface-Objekt gefunden werden, das die Eigenschaften beider Schnittstellenobjekte nachbilden kann. Da es nicht möglich ist, die Eigenschaften von TERMINALS auf SIGNALS abzubilden, mußte auch hier ein TERMINAL als gemeinsames Objekt zum Einsatz kommen.

Für die Verbindung zwischen SIGNALS und TERMINALS finden spezielle D/A und A/D-Wandler Verwendung. Diese ermöglichen das Abbilden der Zustände einer mehrwertigen Logik als Spannung auf ein TERMINAL. Eine geeignete Beschreibung der D/A und A/D-Wandler erlaubt außerdem die Rückgewinnung der events auf den digitalen Signalen. Die Parameter R_s , R_w , R_i , U_x , U_1 , U_0 , U_z , U_{i1} und U_{i0} (s. Bild 3) der D/A und A/D-Wandler können dabei auf technologiebezogene oder auf abstrakte Werte gesetzt werden, so daß sich entweder ein realitätsnahes Verhalten ergibt oder eine VHDL resolution function nachgebildet werden kann.

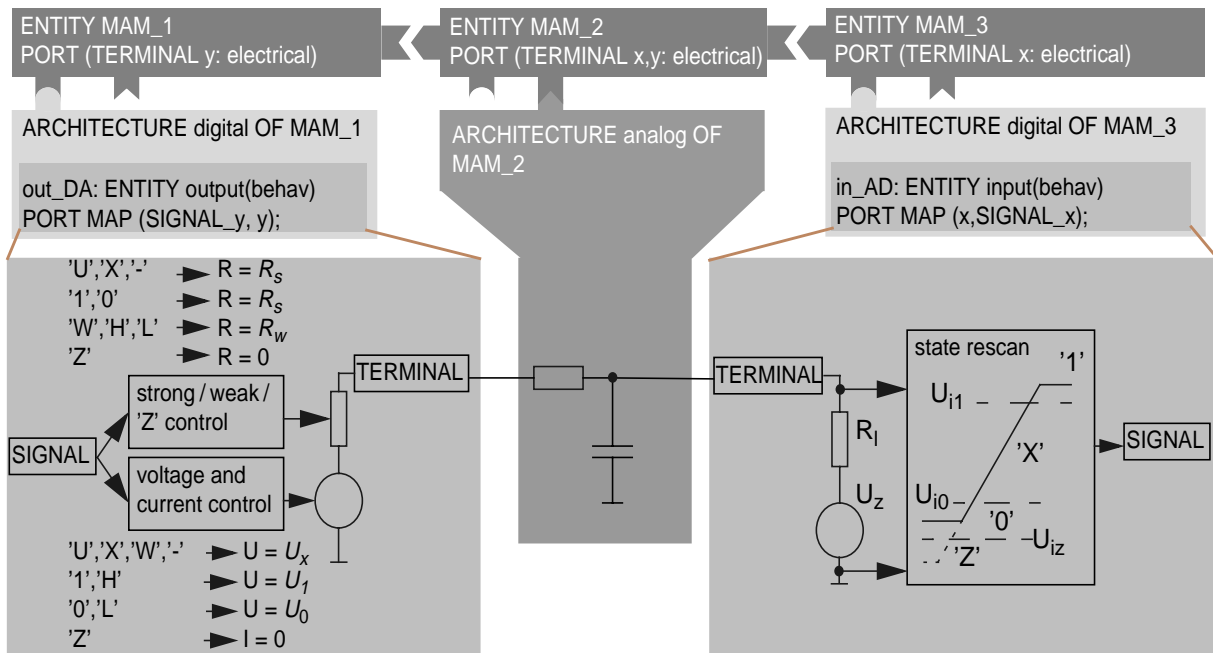


Bild 3 Verbindung analoger und digitaler Ports

Für die Parameter der D/A und A/D-Wandler gelten folgende Beziehungen:

n =maximale Anzahl von Treibern für ein TERMINAL

Parameter	technologieabhängig	technologieunabhängig
U_{i1} : minimale Eingangsspannung, so daß der Zustand '1' erkannt wird, wenn kein Treiber den Zustand 'X' oder '0' hat. U_{i0} : maximale Eingangsspannung, so daß der Zustand '0' erkannt wird, wenn kein Treiber den Zustand 'X' oder '1' hat.	U_{i1} und U_{i0} sind durch die Technologie vorgegeben.	$\frac{U_x + U_1 \cdot (n-1)}{n} < U_{i1} < U_1$ $\frac{U_x + U_0 \cdot (n-1)}{n} > U_{i0} > U_0$
R_s : Ausgangswiderstand für starke Treiber, R_w : Ausgangswiderstand für schwache Treiber, R_l : Eingangswiderstand	R_s, R_w und R_l sind durch die Technologie vorgegeben. $0 < R_s < R_w \ll R_l$	$0 < R_s < R_w \ll R_l$ $R_w \geq \max \left[[n-1] \cdot R_s \cdot \frac{U_0 - U_{i1}}{U_{i1} - U_1}, \right.$ $\left. (n-1) \cdot R_s \cdot \frac{U_1 - U_{i0}}{U_{i0} - U_0} \right]$
Maximales Verhältnis von Treibern mit den Zuständen 'W', 'L' oder 'H' zu Treibern mit den Zuständen '1' oder '0' bei dem am Eingang die Zustände '1' und '0' wiedererkannt werden.	$\frac{N_{w,l,h}}{N_{1,0}} = \frac{R_w}{R_s} \cdot \frac{U_{i1,0} - U_{1,0}}{U_{x,0,1} - U_{i1,0}}$	durch eine <i>resolution function</i> vorgegeben

Eine Besonderheit beim A/D-Wandler stellt die Spannungsquelle U_z dar. Sind alle Treiber im Zustand 'Z', so würde am Eingang des A/D-Wandlers eine Spannung von 0 V anliegen. Der Zustand 'Z'

könnte so nicht von '0' unterschieden werden. Mit Hilfe der Quelle U_z speist der A/D-Wandler über R_i einen kleinen Strom in das `TERMINAL` ein. Ist nur ein Treiber nicht im Zustand 'Z', so wird der minimale Teststrom überlagert und trägt praktisch nicht zum Verhalten auf dem `TERMINAL` bei. Sind dagegen alle Treiber im Zustand 'Z', so stellt sich am Eingang des Wandlers U_z ein. Möchte man den Zustand 'Z' erkennen, so muß U_z entweder über U_{i1} oder unter U_{i0} liegen. Soll dagegen z. B. ein TTL-System betrachtet werden, so kann man U_z auf 4 V einstellen und der A/D-Wandler erkennt eine '1', wenn alle Treiber auf 'Z' sind. Bei einem 5 V-CMOS-System kann U_z auf 2,5 V gelegt werden und der A/D-Wandler erkennt in diesem Fall ein 'X'.

3 Entwurfsablauf mit MAM

Im Vergleich zu einem herkömmlichen Top-Down-Entwurf ergibt sich mit MAM der folgende Entwurfsablauf:

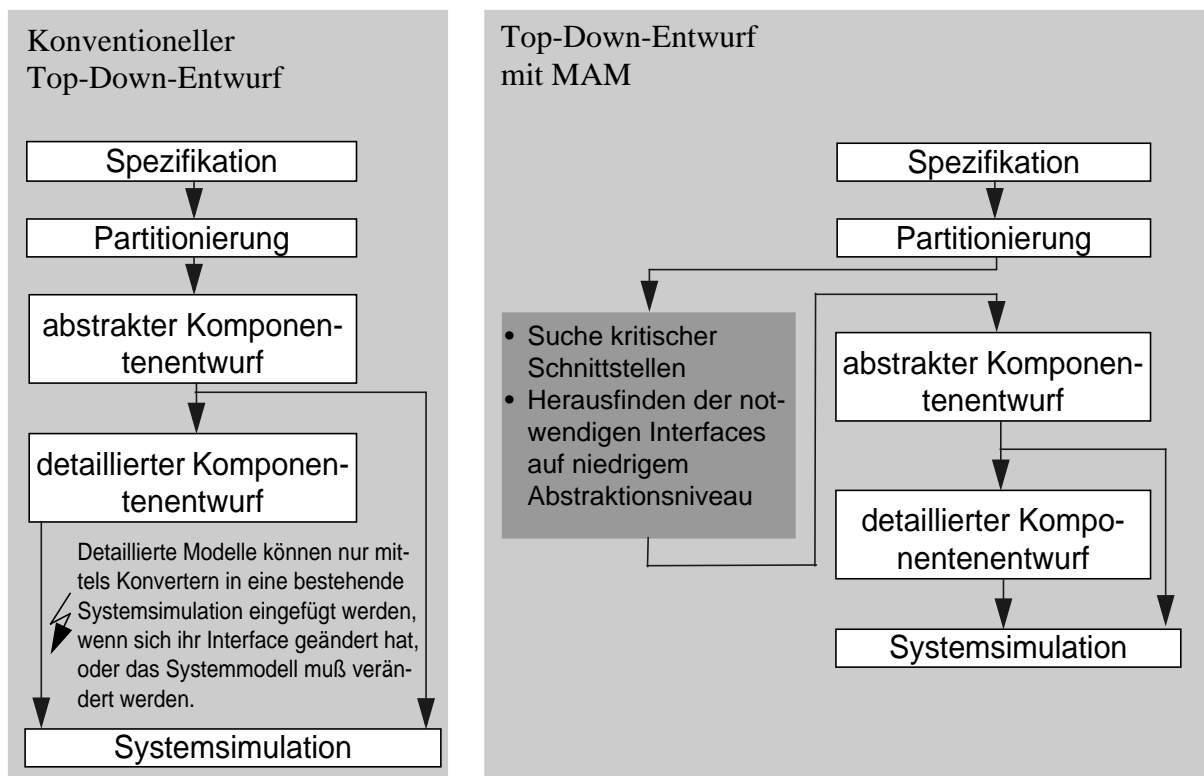


Bild 4 Top Down Entwurf ohne und mit MAM

Die MAM sollte nicht angewandt werden bei rein digitalen Systemen oder Systemen bei denen Modelle oder Schnittstellen automatisch generiert werden, da bei solchen Systemen keine Steigerung der Entwurfs-effizienz durch die MAM zu erwarten ist. Besteht jedoch nicht die Möglichkeit, Modelle durch Synthese zu erzeugen, wie dies bei heterogenen Systemen oft der Fall ist, so ist der Einsatz der MAM sinnvoll. Da solche Entwürfe durch ein hohes Maß an manuellen Arbeiten geprägt sind, wurde im Rahmen MAM zunächst auf die Untersuchung der Realisierbarkeit von Werkzeugen zur Automatisierung verzichtet.

Eine Schnittstelle einer Komponente ist entsprechend Bild 4 „kritisch“, wenn für die Komponente eines oder mehrere der folgenden Kriterien gelten:

- Eine digitale Komponente wird zunächst algorithmisch beschrieben, soll später aber durch ein synthetisierbares oder synthetisiertes Modell ersetzt werden bzw. die Komponente steht in Verbindung mit einer digitalen Komponente auf niedrigem Abstraktionsniveau.
- Eine analoge Komponente wird zunächst als funktioneller Block beschrieben, soll aber später

gegen eine Netzwerkbeschreibung ausgetauscht werden bzw. die Komponente steht in Verbindung mit einem Netzwerk.

- Die Realisierungsart einer Komponente unterliegt einer Variantendiskussion.
- Das exakte zeitliche Verhalten der Komponente ist bedeutsam.
- Für die Verbindung von Komponenten erscheint ein Leitungsmodell notwendig.
- Eine Schnittstelle ist ebenfalls kritisch, wenn sie mit einer der zuvor genannten Komponenten in Verbindung steht.

Bei der Anwendung der MAM haben sich einige Modellierungsregeln ergeben, die beim Entwurf eingehalten bzw. beachtet werden müssen:

- Das Abbilden einer digitalen Schnittstelle auf TERMINALS sollte nur für kritische Schnittstellen durchgeführt werden. Da für jedes SIGNAL ein TERMINAL angelegt werden und für jede THROUGH QUANTITY eines TERMINALS eine Gleichung vorhanden sein muß, ergibt sich bei einer Vielzahl so abgebildeter SIGNALS ein großes vom analogen Simulator zu lösendes Gleichungssystem, was die Simulation verlangsamt.
- Schnittstellen, die man sowohl im abstrakten als auch im detaillierten Modell mit QUANTITIES beschreiben kann, sind keine kritischen Schnittstellen und sollten ebenfalls nicht als TERMINAL beschrieben werden. Informationen auf QUANTITIES kann ein Simulator u. U. von Komponente zu Komponente ohne Rückwirkung propagieren, während für ein Terminal stets ein Eintrag im Gleichungssystem zu lösen ist. Daraus kann je nach Modellstruktur und verwendetem Simulator ein Verlust an Simulationsgeschwindigkeit und -stabilität resultieren.
- Das Abbilden einer QUANTITY erfolgt üblicherweise auf den ACROSS-Zweig eines TERMINALS. Dies bedeutet, daß der Eingangswiderstand des nächsten Modells größer als Null sein muß. Ist der Eingangswiderstand gleich Null, so muß die QUANTITY im sendenden Modell den THROUGH-Zweig des TERMINALS benutzen.
- Werden abstrakte und detaillierte Modelle miteinander verbunden, so ist zu beachten, daß die abstrakten Modelle auf den TERMINALS ein ideales Verhalten zeigen und bezüglich Rückwirkungen aus der detaillierten Komponente unempfindlich sind.
- Weiterhin ist zu beachten, daß in abstrakten Komponenten Informationen aus vorhergehenden detaillierten Komponenten verloren gehen können (z. B. digitale Zustände wie 'U', 'X')

Werden diese Modellierungsregeln beachtet, ist nur mit einer geringfügigen Verringerung der Simulationsgeschwindigkeit zu rechnen. Dafür erspart man sich beim Entwurf die zeitaufwendige und fehleranfällige Konvertierung der Schnittstellen beim Einfügen von Modellen oder Architekturen auf unterschiedlichen Abstraktionsebenen. Dies ist insbesondere dann von Vorteil, wenn z. B. für eine oder mehrere Komponenten Verhaltensmodelle und Makromodelle (durch Ordnungsreduktion aus FEM-Modellen gewonnene Verhaltensbeschreibung [4], [5]) zur Verfügung stehen. Während der Simulation kann so für jede Komponente, für die Architekturen auf unterschiedlichen Abstraktionsebenen existieren, entweder eine abstrakte oder eine detaillierte Version eingesetzt werden, je nach Erfordernissen bezüglich Simulationsgeschwindigkeit und Genauigkeit.

4. Ergebnisse

Die in den Abschnitten 2 und 3 beschriebene Methode wurde und wird beim Systementwurf des Demonstrators „Vibrationssensorarray“, eines Systems zur Schwingungsanalyse mittels mikro-mechanischem Vibrationssensorarray, im Rahmen des SFB 379 angewandt und hat sich dabei bewährt. Auch das Bestimmen der notwendigen Interfaces der detaillierteren Komponenten auf hohem Abstraktionsniveau erwies sich als praktikabel.

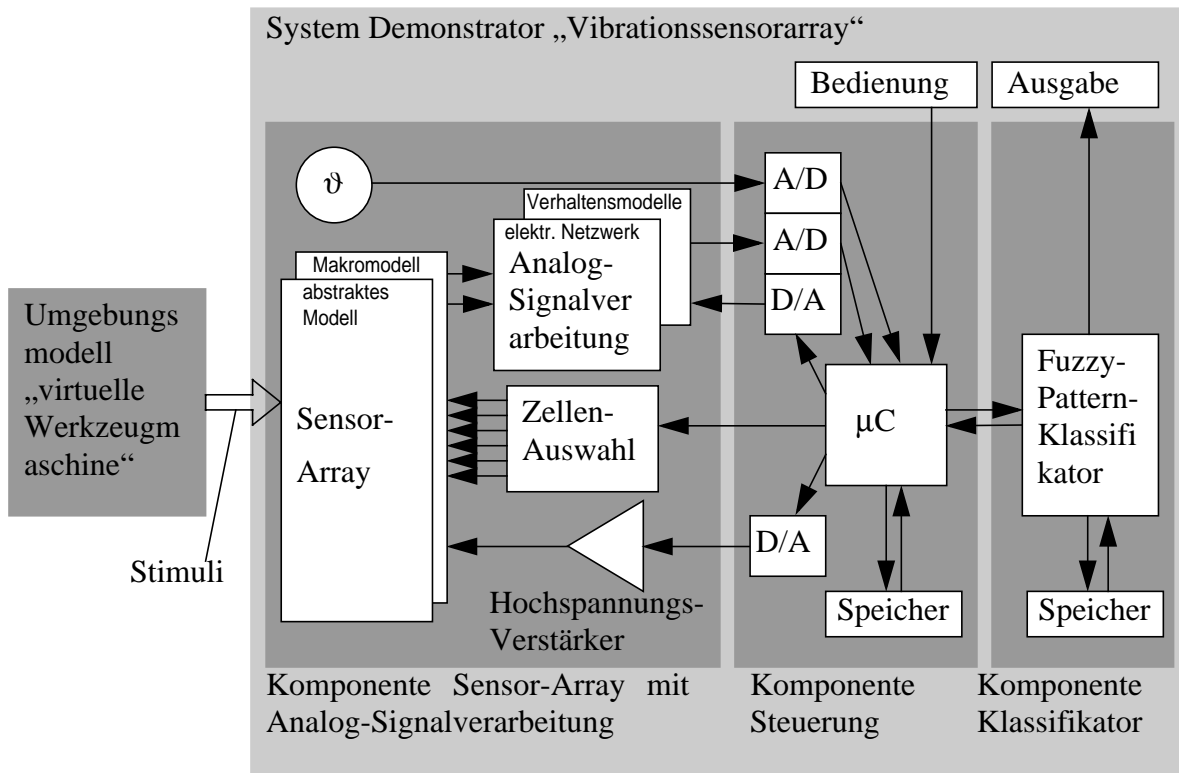


Bild 5 Systemkonzept Demonstrator „Vibrationssensorarray“

Das System besteht aus:

- einem Umgebungsmodell, welches die Vibrationen und die Erwärmung einer Werkzeugmaschine nachbildet,
- dem Sensor, der aus mehreren Feder-Masse-Schwingern gebildet wird, die in ihrer Resonanzfrequenz abgestimmt werden können. Damit ist es möglich, ein Vibrationsspektrum von 1 kHz bis 10 kHz frequenzselektiv zu überwachen.
- einer Analog-Signalverarbeitung, die das entstehende Ausgangssignal aufbereitet,
- einem steuernden Mikrocontroller und einem Fuzzy-Pattern-Klassifikator, der anhand der Vibrationen Fehlerzustände erkennen soll.

Das Umgebungsmodell „virtuelle Werkzeugmaschine“, das Modell des Sensorarrays und der Analog-Signalverarbeitung wurden bereits mit Hilfe der MAM erstellt. Die Modelle für Steuerung und Fuzzy-Pattern-Klassifikator befinden sich noch in der Entwicklung.

Für das Sensorarray ist ein abstraktes Modell auf der Basis linearer Zusammenhänge geschaffen worden. Dieses Modell benutzt jedoch TERMINALS als Interface, so daß dieses Modell sofort durch ein mittels Ordnungsreduktion aus einem FEM-Modell erzeugtes Makromodell ersetzt werden kann, sobald dieses zur Verfügung steht. Die Analogsignalverarbeitung besteht aus mehreren Komponenten, die teils als elektrische Schaltung, teils als Verhalten beschrieben sind. Für einige Komponenten stehen im Rahmen der MAM mehrere Modelle zur Verfügung, so z. B. für einen Tiefpaßfilter, der als Netzwerk, Übertragungsfunktion mit dem Verhalten des Netzwerkes und einer Übertragungsfunktion mit gleicher Grenzfrequenz aber anderer Filtercharakteristik beschrieben ist.

Durch Nutzung der MAM mußte bei der Modellierung des Systems „Vibrationssensorarray“ kein Interface nachträglich geändert werden. Eine Verschlechterung der Simulationsstabilität wurde nicht beobachtet. Die Simulationsdauer konnte allein durch den Einsatz der MAM für die Analog-Signalverarbeitung bei etwas verringerter Genauigkeit halbiert werden. Wird die volle Genauigkeit gefordert, so kann sofort für abstrakte Komponenten wieder eine detailliertere Komponente

eingesetzt werden. Durch den Einsatz des linearen Modells für den Sensor entsteht derzeit noch ein Fehler von 20 %. Steht das Makromodell zur Verfügung, reduziert sich dieser sofort erheblich.

5. Zusammenfassung und Ausblick

Wird das Interface eines Komponentenmodells auf hohem Abstraktionsniveau mit dem Interface entworfen, das auf niedrigem Abstraktionsniveau notwendig ist, so kann während des Systementwurfes für jedes Modell wahlweise ein abstraktes und damit schnelles oder ein detailliertes und damit genaues Modell eingesetzt werden. Durch die Benutzung der MAM ist es somit möglich, ein Systemmodell flexibel zwischen Schnelligkeit und Genauigkeit zu skalieren. Außerdem entfallen zeitaufwendige und fehleranfällige Anpassungen der Schnittstellen bei Änderungen des Abstraktionsgrades der Modelle.

Die MAM läßt sich auch auf Bottom-Up-Ansätze anwenden, indem das Interface des detaillierten Modells in den abstrakteren Modellen weiterverwendet wird. Die MAM ermöglicht es ebenso, Designalternativen aus digitalen und analogen Modellen für eine Komponente zu evaluieren und zu optimieren. Der Ansatz der MAM verursacht nur wenig Overhead, ist 100 %ig konform zum IEEE-Standard für VHDL-AMS [1] und in der Praxis unter Beachtung der im Abschnitt 3 beschriebenen Modellierungsregeln leicht anwendbar.

Nachteilig für einen automatisierten Entwurf ist die Tatsache, daß diejenigen Modelle manuell, auf Basis der Systempartitionierung und der Erfahrung des Entwerfers, gefunden werden müssen, bei denen ein Einsatz der MAM notwendig und sinnvoll ist. Die Bestimmung der notwendigen Interface-Objekte und Datentypen für diese Modelle kann dann formell entsprechend Abschnitt 2 und 3 erfolgen. Durch geschickte Partitionierung des Systems besteht die Möglichkeit, Teilsysteme, auf die die MAM nicht angewandt werden muß, wie gewohnt mit automatisierten Methoden zu erstellen oder zu bearbeiten.

Die hier vorgestellte Methode der MAM hat sich bereits im Entwurf des Demonstrators „Vibrationssensorarray“ prinzipiell bewährt und wird auch bei zukünftigen Arbeiten eingesetzt. Die sich dabei ergebenden Erkenntnisse fließen in die MAM ein, wobei insbesondere die Modellierungsregeln vervollständigt und präzisiert werden.

Danksagung

Die hier vorgestellte Arbeit entstand im Rahmen des Teilprojekts A2 „Systementwurf“ des SFB 379, der durch die DFG (Deutsche Forschungsgemeinschaft) gefördert wird.

Literatur

- [1] The Institute of Electrical and Electronics Engineers, Inc.: *IEEE Standard VHDL Language Reference Manual (Integrated with VHDL-AMS changes)*. IEEE Std 1076.1, 1999.
- [2] Schlegel, M.; Herrmann, G.; Müller, D.: *Multi Architecture Modeling Design Method for Mixed Signal and Multi Domain System Simulation - First Solutions*. The International MEMS Workshop 2001, Singapore, July 4-6, 2001, pp. 662-667, ISBN 981-04-4165-7
- [3] Rosenberger, R.; Huss, S. A.: *A Systems Theoretic Approach to Behavioral Modeling and Simulation of Analog Functional Blocks*. IEEE/ACM Design, Automation and Test Conference Europe, Paris, 1998
- [4] Hanna, J. P.; Hillman, R. G.: *A Common Basis for Mixed-Technology Micro-System Modeling*. International Conference on Modeling and Simulation of Microsystems MSM 99, San Juan, Puerto Rico, U.S.A., Apr. 19-21, 1999
- [5] Bennini, F.; Mehner, J.; Dötzel, W.: *A modal decomposition technique for fast harmonic and transient simulation of MEMS*. The International MEMS Workshop 2001, Singapore, July 4-6, 2001
- [6] Siegmund, R.; Müller, D.: *SystemC^{SV}-Extension of SystemC for Mixed Multi-Level Communication Modeling and Interface-based System Design*. Design, Automation and Test in Europe (DATE 2001), Munich, Germany, 13-16 March