Generation of the HDL-A-model of a Micromembrane from its Finite-element-description

Klaus Hofmann[1], Manfred Glesner[1], Nicu Sebe[2], A. Manolescu[2], Santiago Marco[3], Josep Samitier[3], Jean-Michel Karam[4], Bernard Courtois[4]

[1]:Darmstadt University of Technology, Institute of Microelectronic Systems, Karlstrasse 15, 64283 Darmstadt / Germany, E-Mail: hofmann@mes.th-darmstadt.de

[2]:„Politehnica“ University Bucharest / Romania

[3]:University of Barcelona / Spain

[4]:INPG / TIMA, Grenoble / France

Abstract

A CAD-tool for the automated generation of behavioral models in HDL-A is presented. This CAD-tool has been implemented in the frame of a project for automatical modeling of microsystem components for the co-simulation with VHDL- or Spice-Models. Starting from the Finite-Element-description of a microcomponent a nonlinear behavioral HDL-A-model is generated by successively adding or deleting effects to the HDL-A-model according to the observed differences between the two models. Using the example of a micromembrane the practicability of this approach will be demonstrated. This CAD-tool provides a method for decoupling the generation of behavioral models from the Finite-Element-simulation process.

1 Introduction

Simulation of heterogeneous systems on the system level implies the simulation of the whole system and that of particular devices with a special emphasis on points of interest like functional behavior, timing, power consumption and so on. Simulation is very closely connected to modeling because only the system- or device-behavior can be simulated that has been taken into account while developing the models. Functional simulation of microsystems and microcomponents can be done efficiently by using a single system description language suitable for a single simulator. The advent of analog HDLs like HDL-A by *Anacad,* MAST by *Analogy* and the future standard VHDL-AMS offer the possibility to create behavioral models of electrical and non-electrical devices without having the limitations of Spice. At VHDL-AMS´ stage of infancy not much work is known to the authors concerning the insertion of VHDL-AMS-like descriptions in the design flow for analog or heterogeneous systems (see [MOS95], [ELT93], [ARB91], [SEI95], [SAX95]).

The creation of behavioral models for microsystem components causes several problems:

- The engineer responsible for creating the behavioral models needs excellent theoretical knowledge about the component as well as excellent know-how about the simulation tools.
- No verification of the behavioral models with regard to the 2D/3D FEM-models is done.
- There is no quantitative number for the accuracy of the behavioral model for other waveforms than the one used during the creation.
- The behavioral models have to obey certain standards concerning system-interfaces, generics etc.. : difficult to manage from the FEM-designers view.

Some of the problems mentioned above need to be explained: The main problem for the global functional simulation of microsystems is to create accurate models of micromechanical and microoptical devices in the transient domain. Due to the fact that most system-engineers are not familiar with micromechanics and Finite-element-(FE) modeling and technology-experts do only scarcely know the needs for system-modeling on all levels of interest, in general a simple linear parameter extraction is performed to create a simple behavioral model of that device. With the miniaturization of mechanical and optical devices this will not be sufficient for an accurate simulation of system behavior. Furthermore, the fact that engineers have to be the experts for two CAD-tools in very different domains violates the idea of dividing the work to specialists.

ED&TC ’97 on CD-ROM

These problems can be overcome by the CAD-tool discussed here for supporting the automated creation of behavioral models. This CAD-tool will provide the following features:

- Fixed interface for checking in Finite-Element-models in a CAD-library
- Checkout of the FEM-models, alteration of the input stimuli and output-format according to the CAD-tools necessities, simulation of the FEM-models
- Support of reuse of already created behavioral models in the library, using the best-suited behavioral model as a start model
- Parameter optimization of the behavioral model
- Differential adaptation of the behavioral model effects by comparing the waveforms of the FEM- and the behavioral model and concluding the insertion and deletion of certain (nonlinear) effects
- Support of model verification by using up to 10 different input stimuli
- Checkin of the behavioral model in a format suitable to restart the model-creation-process
- Creation of a single HDL-A ARCHITECTURE/-ENTITY-pair that is optimized with regard to the number of internal states needed for representing the specific model, and therefore optimized with regard to simulation speed.

A side effect of the tool is that a well-defined interface for exchanging models from the 2/3D-structural to the behavioral model domain is provided for microsystem components.

In the following sections we will present and discuss the features mentioned above. Section 2 will present the work-flow implemented in the CAD-tool, section 3 will provide the algorithms implemented for model optimization in a very brief way, since this has been discussed in previous papers. In section 4 we will discuss the application of this CAD-tool to the Finite-Element-description of a micromembrane and we will present the created HDL-A-model in detail. Section 5 will contain an extensive discussion.

This CAD-tool has been created in the frame of the Esprit-Basic-Research Project 8173 BARMINT (Basic Research for Microsystems Integration).

2 Work flow within the CAD-tool

One of the key features of this CAD-tool is the decoupling of the FEM-simulation-process from the creation of the behavioral model. Therefore a global interface for storing and retrieving the FEM-model in a library format must be defined. For this CAD-tool the engineer responsible for FEM-model creation and simulation has to provide only the following 4 files: the database-file (.db), the ASCII-file describing the transient simulation plus the loadstepfiles describing a ´0´ resp. a ´1´. These 4 files (plus a file containing information about the model) are archived, compressed, uuencoded and stored in the Cadence-library format. It must be pointed out that providing these 4 files is the only action required by the engineer responsible for FEM-simulation. It is this engineers responsibility to provide a batchfile that ensures a stable, convergent and appropriate simulation of the component. This clearly defined interface is described in Fig. 1 as step 1. Step 2 (as well as the following steps except step 7) can be done by an engineer who need not to be an expert in FEM-simulation.

Step 7 requires the creation of a rulebase for a certain class of microcomponent. This is a step that requires deep knowledge of the (nonlinear) effects that may be necessary for describing the behavior of the microcomponent, but also theoretical based solutions for simpler, but related 2/3D structures may be included. The rulebase is constructed as a fuzzy-rulebase, whereby its operators are traditional identification techniques (simple parameter-estimation) and observational operators (like "Slewrate too low" or "Behavioral model has no oscillations"). These operators may be connected by Fuzzy-operators to weighted rules. It must be stressed that the creation of such a database is a work necessary only once for a certain type of component. Internal studies at Darmstadt University show that rule-databases may be created which can be used for component-classes such as membranes, operational amplifiers and acceleration sensors. Furthermore it should be pointed out that these rule-databases are almost independent of the geometry of the 2/3D-component, but of course certain expert knowledge is necessary for creating them. Step 8 describes the choice of the parameter optimization techniques that may be used for finding the parameters. Simple gradient techniques are in general not sufficient since nonlinear effects may cause the existence of several local optima. The selections required are easy to perform and are supported by the GUI. After the completion of the global optimization process the user has to decide if the generated model meets the required accuracy. If necessary, the translation has to be repeated, otherwise the new behavioral model may be checked in the Cadence-library being accessible for future runs. The compiled object as well as the ASCII-version of the HDL-A-model is available and may be used for further simulations.

3 Brief Description of the CAD-tool for automatical modeling of microsystem components based on Finite-Element-(FE)-models

In this section a brief overview on the CAD-tool for automatical modeling of microsystem components based

Action		Description	User-Input	Tool
1. Perform the FE-Simulation		Simple transient simulation with a step excitation at the input. Integral outputs must be defined + a .log-File written	FE-Database (model), Simulation	Ansys
2. Checkin of the model in database	MCS-Library	Library-Checkin of the .log-, .db + Loadstep-files	Negligible	FEM->HDL-A-translator, Cadence
3. Checkout of the model + automatic adaption of bitlength	MCS-Library	Checkout of the model, adaption of the bitlength in order to get 15-20% change of the output-variable	Negligible	FEM->HDL-A-translator, Cadence
4. Simulation of bitpatterns	Ansys	Creation of the loadsteps, FE-simulation	Bitpattern (01101...)	FEM->HDL-A-translator, Ansys
5. Restriction of the search-space within the behavioral model library		Decide, which model-categories may serve as a basis for the translation	Selection of Model-categories	FEM->HDL-A-translator
6. Library Search + Checkout	? MCS-Library	The Cadence-library is searched for the best-matching behavioral model. Best selection can be checked out.	Negligible	FEM->HDL-A-translator, Cadence
7. Choice or creation of rulebase	IF .. THEN	The structural evolution of the behavioral model depends on the programmed database.	Rules	FEM->HDL-A-translator
8. Choice of parameter optimization techniques	Opt.	The tool offers different parameter optimization techniques within a roundrobin-scheduler	Setup of parameter optimization techniques	FEM->HDL-A-translator
9. Perform the translation	HDL-A	Parameter and structural optimization of the behavioral model	Negligible	FEM->HDL-A-translator, Eldo
10. Accept or reject the new model	HDL-A ?	After a successful translation, accept (i.e. store the new model in the library) or reject it (start the translation again using different optimization parameters or starting models)	Negligible	FEM->HDL-A-translator
11. Store the new model in the library	HDL-A MCS-Library	The internal format of the new behavioral model must be stored in the library if it should serve as a possible new starting model	Negligible	FEM->HDL-A-translator, Cadence

Fig. 1 Workflow inside the CAD-tool (users view)

on FE-models is given. A more detailed description can be found in [HOF95], a description of the CAD-framework suitable for integrated microsystems in [POP95a]. Due to the fact that realistic microsystem components can only be described by inclusion of nonlinear effects, whereby the kind and impact of such effects varies very much, standard system identification (as in [ISE88]) cannot be used efficiently. The CAD-tool discussed here uses a global optimization scheme, which can be subdivided in parameter and structural optimization. The lacks in the "structure" of the behavioral model are identified by operators, which are the arguments in a fuzzy-rule based environment for the representation of the knowledge concerning the used component. The basic algorithm is the following:

I: Select a startmodel from the database
II: Select the input bitpattern which yields the worst costvalue (the worst correspondence between the 2/3D and the behavioral model)
III: Optimize the parameters by applying simulation procedures (back to II if an internal counter has not exceeded a certain predefined value).
IV: If the targeted accuracy is not met, identify lacks by operators and insert or delete a certain effect in the behavioral model. (back to II until an internal counter for the number of allowed structural optimizations has not exceeded a certain predefined value).

Fig. 2 gives an overview about the functional blocks of the CAD-tool. Starting from the best HDL-A-like model which can be found in a library and the initial parameter values, a variety of optimization methods (gradient methods, heuristical methods such as simulated annealing) are used to find the model-parameters in such a way that the output-functions of the behavioral model are almost identical to the one of the FE-model. Very often it is the case that the considered effects in the behavioral model are not sufficient to meet the desired accuracy. In this case the module for structural optimization analyses and alters the behavioral model in such a way that effects which are not effective for the conformity of both models

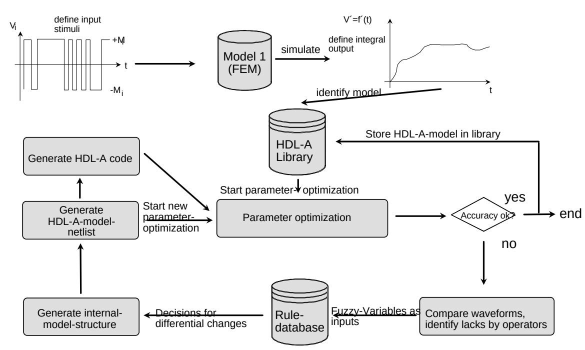

Fig. 2 Function blocks of the FEM->HDL-A translator

are deleted and other effects that may improve the accuracy are inserted. A traditional expert-system approach based on observational operators is used, the values of the operators are connected by fuzzy-operators such as AND, OR, NOT in fuzzy-rules. These rules represent the knowledge of the component-specialists concerning the effects which must be considered while modeling the device, so that different rule-databases are provided for different devices.

4 Generation of the behavioral model of a micromembrane

In the frame of the Esprit-project BARMINT one of the existing realistic benchmarks is the 3D-model of a micromembrane with the pressure as the input and the membrane position as the output. As the effects of the starting model for the model generation a single pole, a positive gain and a positive offset have been selected. These effects are too simple to describe the realistic behavior of the structural model, therefore a rulebase containing 7 rules for the possible insertion of additional poles and zeros plus nonlinear saturation effects (such as *arctan* and *slewrate*-limitations) has been created. The main goal set forth at the beginning of the generation process was to create a model that yields an error of less than 1% for all 4 chosen bitpatterns. The following settings have been chosen:

- parameter optimization: three succeeding optimization slices (gradient, simulated annealing, gradient) targeting 1% accuracy
- gradient optimization slice to run on 2 CPUs in parallel
- four bitpatterns with 76 bits each
- HDL-A-codegeneration enabled

The final result has an accuracy of 2.49%, which is less accurate than projected, but acceptable for the designers eye (see Fig. 3). The resulting model is a single HDL-A ENTITY/ARCHITECTURE model (see Fig.4). The generated code is very dense and hard to read, but optimal with regard to the minimum of required internal states.

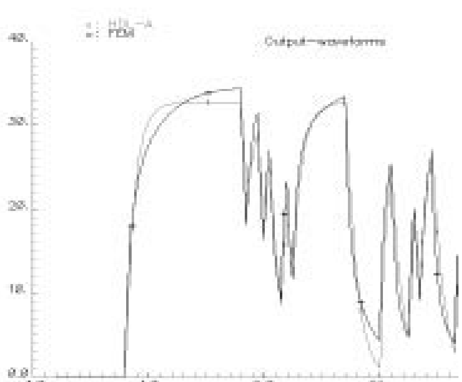

Fig. 3 Output waveforms

5 Final discussion

The described CAD-tool works well within the research laboratories of Darmstadt University of Technology and has been tested during the last 6 month (beside artificial examples) on various microelectronical and micromechanical components such as operational amplifiers, microaccelerometers and micromembranes. The usage of the tool has shown to be adequate even for nonexpert users due to the GUI.

```
ENTITY ww IS
GENERIC (y2pole1_fpole , y3gain_pos_value ,
y5pos_off_value, y4_atan_neg_value : REAL);
PIN ( in0 , out0 :ELECTRICAL );
END ENTITY ww;

ARCHITECTURE behavioral OF ww IS
STATE V_primey2 : ANALOG  ;
BEGIN
 RELATION
PROCEDURAL FOR INIT =>
y2pole1_fpole := 10000.0;
y3gain_pos_value := 10000.0;
y5pos_off_value := 1000000.0;
y4_atan_neg_value := 10.0;
PROCEDURAL FOR DC, AC, TRANSIENT =>
out0.i %= out0.v*1000000.0 -
V_primey2*6280000.0*y2pole1_fpole*y3gain_pos_value -
1000000.0*y5pos_off_value ;
EQUATION ( V_primey2 ) FOR AC, TRANSIENT =>
 - in0.i*9.999999975e-07 + atan(y4_atan_neg_value*in0.v)
- V_primey2*6.28000021*y2pole1_fpole - ddt(V_primey2) ==
0.0 ;
END RELATION ;
END ARCHITECTURE behavioral ;
```

Fig. 4 Generated HDL-A-code

The experience gained during this period can be summarized as following:

- The tool works well with a success-rate of more than 75% (generating a usable nonlinear behavioral model meeting the desired accuracy)
- There is no need to use this tool if the 2/3D-model is known to be linear (in this case parameter identification techniques are much faster)
- The rule-database has to be consistent in the sense of not providing two actions for one observation
- Known analytical solutions of simpler, but related structural objects drastically speed up the generation of the behavioral model
- The computer resources needed are huge: several hours on several computers are needed for the transient simulation of the FEM-models. The generation-process occupies a single machine (with up to 4 CPUs) for approx. 5 CPU-hours.
- The usage of the CAD-tool causes only little problems even for non-experienced users (e.g. undergrad students): Enhancement of the HDL-A-core-database and the rule-database requires only little experience.
- Problems still exist with non-steady nonlinearities such as deadband-functions or hysteresis effects

- The current version allows only the generation of components with 1 input and 1 output (but *through* and *across*-values (e.g. voltage and current) are allowed for the pins), providing more than 1 input is a current research topic.
- The implicit verification (i.e. the usage of different input-stimuli during the generation process) yields much better accuracy even for trapezoidal input stimuli than the usage of a single input-stimulus.
- Generation of true HDL-A-code provides a speedup of the generation by a factor of 3, but is currently possible only for a limited set of effects.

Future work will be in the domain of creating consistent rule-databases for several microcomponents, in the domain of observational identifiers as the variables of the rule-database and in the domain of HDL-A-codegeneration.

References

[ANA96] HDL-A Language and Users Reference Manual, ANACAD, 1996

[ARB91] B. Arbegard: Functional modeling of analog blocks in VHDL, Proc. of the EuroVHDL 1991, Stockholm 8-11 September 1991

[BAR95b] R. Baraniecki, A. Rosinski: An approach to the VHDL modeling of operational amplifiers, MixVLSI´95 Conference, Krakau, 29-31 May 1995, page 205ff

[BER93] J.M. Berge, A. Vachoux: VHDL-A Extensions, Euro-DAC´93 with EuroVHDL´93, Tutorial

[BOR91] J. Bortolazzi, K.D. Müller-Glaser: "Framework Integration of an environment for Microsystems", Micro System Technologies 91, p. 153-160

[CAD94] SpectreHDL: Cadence press release, August 1994

[CAS88] G. Casinovi: "Macromodelling for the simulation of large scale analog Integrated Circuits", Ph.D.-Thesis, University of California, Berkeley, 1988

[CAS91] G. Casinovi, A. Sangiovanni-Vincentelli: "A Macromodeling Algorithm for Analog Circuits", IEEE Transactions on Computer-Aided Design, No.2, February 1991, pp 150-160

[CHU75] L.-O. Chua, P.-M. Lin: "Computer Aided analysis of electronic circuits: Algorithms and computational techniques", Prentice-Hall, 1975

[CON92] J.A. Conelly, P. Choi: "Macromodelling with Spice", Prentice-Hall 1992

[COU93] B. Courtois, "The state of the art in CAD", Invited presentation at the workshop on CAD Methods and Technical Management in Electrical Engineering Education, Budapest, Hungary, June 16-18 1993

[ELT93] H. El Tahawy et al.: "VHDeLDO: A New Mixed Mode Simulation", EuroDAC Okt. 93, Hamburg, S.546ff

[HOF95] K. Hofmann, J. M. Karam, M. Schulze, M. Theisen, B. Courtois, M. Glesner:"Automatische Übersetzung von FEM-Modellen in eine analoge Hardwarebeschreibungssprache, Mikrosystem-technik & Mikroelektronik 95, 16-17. Oktober 1995, Chemnitz

[HOF96a] K. Hofmann, M. Lang, J.M. Karam, M. Glesner, B. Coutois: Generation of a behavioral model of an acceleration sensor from its Finite-Element description, to be published in the Proceedings of the 3nd France-Japan Congress (1st Europe-Asia Congress) Mecatronics, Besancon, October 1-3 1996

[HOF96b] K. Hofmann, M. Lang, M. Glesner: Generierung eines HDL-A-Modells eines Beschleunigungs-sensors ausgehend von einer FEM-Beschreibung. 3. Workshop Methoden und Werkzeuge für den Entwurf von Mikrosystemen in Frankfurt a. M. am 27. und 28.6.1996

[ISE88] R. Isermann: "Identifikation dynamischer Prozesse Band I + II", Springer, 1988

[KAR96] J.M. Karam, B. Courtois, K. Hofmann, M. Glesner, A. Poppe, M. Rencz, V. Szekely: Applied Design and Analysis of Microsystems, ED&TC 96, Paris, France, 11-14 March 1996

[MOS95] V. Moser, H.P. Amann, P. Nussbaum, F. Pellandini: "Generating VHDL-A-like Models Using ABSynth". In: Proceedings of the EURO-DAC & EURO-VHDL 95, 18.-22. September 1995, Brighton, UK

[MÜL93] K.D. Müller-Glaser, J. Bortolazzi, Y. Tanhuran, J. Ernst: CAE in Requirements Definition and Specification for complex Microelectronic Systems. In: Conf. on VLSI Design 1993, Bombay, India, 1993, S. 305-310

[PEL94] G. Pelz, J. Bielefeld, G. Zimmer: "MEXEL: Simulation of Microsystems in a Circuit Simulator Using Automatic Electromechanical Modelling", Micro System Technologies 94, Berlin, 19-21 Oktober 1994

[POP95] A. Poppe, J.M. Karam, K. Hofmann, M. Rencz, B. Courtois, M. Glesner, V. Szekely: "CAD framework concept for the design of integrated microsystems", Micromachining and Micro-fabrication 95, 23-24 October 1995, Austin, TX

[POP95a] A. Poppe, J.M. Karam, K. Hofmann, M. Rencz, B. Courtois, M. Glesner, V. Szekely: A general CAD concept and design framework architecture for integrated microsystems, Proceedings of the MicroSim 95 Conference, Southampton UK, 1995

[SAX95] E. Sax, Y. Tanurhan, K.D. Müller-Glaser: "Unterstützung des durchgängigen Systement-wurfs durch VHDL-A". In: Proceedings des 7. E.I.S.-Workshops am 7.+8. November 1995, Chemnitz

[SEI95] R. Seidel, D. Müller.: Beschreibung von heterogenen Mikrosystemen mit VHDL-A, Beitrag zur Informationstagung Mikroelektronik 1995, Wien, 27.-28. September 1995

[VAC95] A. Vachoux: LESs & LAT White Papers, IEEE DASC 1076.1 group meeting, April 1995, available via anonymous ftp from nestor.epfl.ch