VHDL-AMS in MEMS Design Flow

Joachim Haase, Jens Bastian, and Sven Reitz

Fraunhofer Institute Integrated Circuits Erlangen
Branch Lab Design Automation Dresden, Zeunerstraße 38, D-01069 Dresden, Germany
Email: Joachim.Haase | Jens.Bastian | Sven.Reitz @eas.iis.fhg.de

Abstract. Behavioral modeling languages can be used in different steps of top-down design and bottom-up verification of MEMS design. The available language facilities of VHDL-AMS and the requirements of the applied methods in this process are confronted. Especially the application of Kirchhoffian networks to model 3D movements is taken into account. The decisions that should be done at the beginning of the modeling process are discussed. This is especially important if models from different sources shall be combined later on. Experiences using available simulation engines are presented. A micro mechanical accelerometer and an electrostatic beam actuator are investigated. Therefore a set of basic elements for MEMS simulation was created.

1 Introduction

MEMS technology requires CAD tools for support. Modeling and simulation play an important rule in this environment [Sen98], [FJ99]. Compared to the design of electronic systems the development of MEMS design tools is at the beginning. But a lot of well established ideas from EDA can be applied to MEMS design. Thus, a top-down methodology can be used to handle complex designs [Fe00]. Behavioral modeling facilities allow to specify components of micro electromechanical systems and to simulate the specified systems. After the design or decision concerning the re-use of existing subsystems a validation of the system behavior should be done. This can also be carried out by simulation. Therefore behavioral models of the components with calibrated parameters based on the realized subsystems must be available for the simulation. This step is usually called bottom-up verification in the electronic design flow [OVC00].

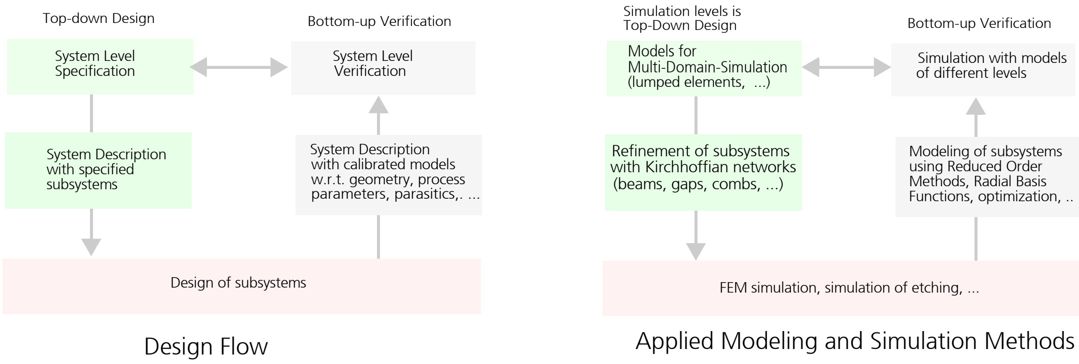

Fig. 1. Top-down design and bottom-up verification in a MEMS Design Flow

The usage of behavioral modeling languages in such a design flow is widely discussed. Figure 1 shows typical steps of this flow and the applied methods. Advantages of the application of a unified behavioral description language are among other things the possibilities to combine models of different levels of abstraction and from different sources. One language that covers these requirements and the applied methods in the design flow is VHDL-AMS [De99]. VHDL-AMS can be used to build complex analog and mixed-signal models. Differential equations, algebraic constraints and logic controls can be combined. This paper tries to confront the applied modeling with simulation methods. Examples use the language constructs that are available in ADVance MS [MGC]. With respect to the status of the standardization of the language and the scope of the language that is covered by commercially available tools, different implementation approaches are considered.

2 Modeling Methods and their Application using VHDL-AMS

2.1 General Approach using Kirchhoffian Networks

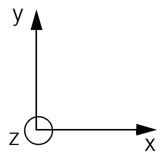

Fig. 2. Global coordinate system

Modeling with Kirchhoffian networks is well established in MEMS simulation [NBL98], [CPZ98]. The following considerations base on this approach. The fundamental idea consists in the combination of subsystem models to describe a more complex behavior. The connection points carry across and flow quantities. The across quantities describe translational displacements in a global coordinate system, rotations about global axes, and electrical voltages. The flow quantities characterize forces in the direction of the coordinate axes, torques about axes, and currents resp. The sums of the mechanical flow quantities at a connection point have to be zero for each axis of the coordinate system (see Fig. 2). The connectivity of the components is described by a netlist. The simulation algorithm is responsible for the fulfillment of the Kirchhoff Laws. Refinement of the specification of mechanical subsystems should take into account the movement described in a global coordinate system. In the simpler case movements in x and y directions and rotations about the z axis are considered. That means only movements in the xy-plane are allowed in the simpler case. In the general case movements in all directions and about all axes must be taken into account [NBL98], [CBZ00].

2.1.1 VHDL-AMS Declaration of MEMS Connection Points

From the VHDL-AMS language point of view appropriate `NATURE` and `TERMINAL` declarations have to be applied to take into account the special MEMS requirements. Thus, a connection point of a MEMS component has to be characterized by a collection of different terminal declarations.

	Displacement in xy-plane / rotation about z-axis (I)	6 degrees of freedom (II)
A	**terminal** tx : kinematic; **terminal** ty : kinematic; **terminal** rz : rotational; **terminal** e : electrical;	**terminal** tx : kinematic; **terminal** ty : kinematic; **terminal** tz : kinematic; **terminal** rx : rotational; **terminal** ry : rotational; **terminal** rz : rotational; **terminal** e : electrical;
B	**terminal** t : kinematic_vector (1 **to** 2); **terminal** r : rotational_vector(1 **to** 1); **terminal** e : electrical;	**terminal** t : kinematic_vector (1 **to** 3); **terminal** r : rotational_vector(1 **to** 3); **terminal** e : electrical;

Fig. 3. Different possibilities for terminal declarations

Different possibilities to define connection points are shown in Fig. 3. The simpler case is considered in column (I), the general case in column (II). The declarations in row A do not require the implementation of multi-dimensional terminals in a VHDL-AMS simulation engine. The declarations in row B allow an easy change from a lower dimensional description to a higher dimensional one. In both cases different lines have to be used to connect associated terminals.

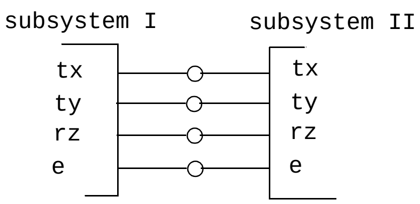

Fig. 4. Example for connection

Fig. 4 demonstrates the connection of two “MEMS points“ that among other things describe the displacement in the xy-plane as pointed out in the first column of row A of Fig. 3. Unfortunately it is not possible to combine the scalar and array natures that characterize a “MEMS point” in a composite record nature. Nature record fields have to be of the same type (see [IEEE99], section 3.5.2.2). Thus the current VHDL-AMS standard in principle only allows to handle “MEMS” points in the way suggested in Fig. 3. Furthermore if the system of equations is established using SI units the solutions vary across several orders of magnitude [BB01]. This fact has to be taken into consideration during the solution of the simulation problem. One way would be to define appropriate natures and `TOLERANCE` groups for this purpose. At the moment, such declarations are not part of the draft packages of the standard package working group [WG]. Another way is to equilibrate across and through quantities for the mechanical part using scaling factors.

2.1.2 Applied Rules in VHDL-AMS Models

The following rules were applied for the investigations:

- A multi-dimensional connection "MEMS point" is characterized in accordance with row A/column (I) of Fig. 3. The corresponding declaration should be allowd in most current VHDL-AMS simulation engines. In the case of six degrees of freedom the declarations were done in accordance with row B/column (II). In addition, simple lumped elements as mass, spring, and damping elements only take into consideration displacements in one direction or rotations about one axis.
- If a subsystem is a pure mechanical one as for instance a beam only the mechanical part of the "MEMS point" description is used in the description. Also in other case only the necessary parts of the description are used.
- Forces and displacements of the model descriptions can be transformed to the SI system by multiplication with `scale_pos`. In the case of angles and torques the values have to be multiplied by `scale_ang`.
- External forces are modeled by a through source that is directed from the concerning node to the corresponding reference node.

`NATURE`, library and package identifiers are used in accordance with ADVance MS.

2.2 Modeling Methods

2.2.1 Models of Primitives for Top-Down Design

The behavior of the (linear) mechanical part of the subsystems can be described by differential equations

$$M \cdot \ddot{x} + D \cdot \dot{x} + K \cdot x = -f \quad \text{with } M, D, K \in R^{N \times N} \text{ and } x, f : [0, Tend] \to R^N \tag{1}$$

x is e. g. the vector of displacements and angles of rotation in a global coordinate system. *f* is the contribution vector of forces and torques at the connection points of the subsystem that are responsible for the across quantities *x*. *M* is the analytically given mass, *D* the damping, and *K* the stiffness matrix.

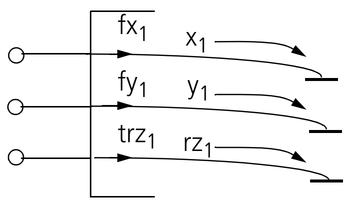

Fig. 5. MEMS connection point and terminal quantities

From the viewpoint of a description language equation (1) requires that vector and matrix operations with one and multi-dimensional arrays should be available to model the behavior. These operations are not part of VHDL-AMS but it is possible to overload the basic operations for real-valued operands by matrix operations.Fig. 5 shows a part of the internal structure of a model described by (1). The (scalar mechanical) terminals are connected by branches with the associated reference nodes. x_1, y_1, and rz_1 are components of a vector *x*. The through quantities fx_1, fy_1, and trz_1 are components of a vector *f*. A typical element that can be modelled in this way is a mechanical beam (see e. g. [NBL98]). In addition the electrical resistance of the beam can be taken into consideration. A more complex element is built up by two parallel beams that take into account the electrostatic force between these beams. A good choice of fundamental MEMS elements was proposed in [CZP98] and following papers. In principle all these elements can also be expressed with VHDL-AMS. The advantage of a VHDL-AMS solution is the possibility to combine all these models with other models and extend user defined models in an easy way.

Example

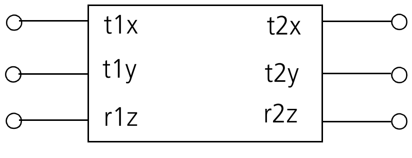

Fig. 6. MEMS connection point and terminal quantities

Fig. 6 shows the interface of a mechanical beam that can be moved in the xy-plane. The entity declaration is given in Fig. 7. General technology data are stored as constants in the package `mems_technology`. The default values of the generic parameters are taken from this package. They can be overwritten during instantiation. The package `mems_technology`, especially the package body where the values are assigned to the constants, has to be replaced in the library `mems` to change from one technology to another one. Mass, stiffness, and damping matrices depend analytically on geometric and material parameters (see e. g. [CZP98]).

```
library disciplines, mems;
use disciplines.kinematic_system.all;
use disciplines.rotational_system.all;
use mems.mems_technology.all;

entity beam2d is
    generic
     (l              : real;           -- beam length in meters
      w              : real;           -- beam width  in meters
     h             : real := mems.mems_technology.h; -- thickness of the beam in meters
      density        : real := mems.mems_technology.density;
      fluid          : real := mems.mems_technology.fluid;
      viscosity      : real := mems.mems_technology.viscosity;
      Youngsmodulus  : real := mems.mems_technology.youngsmodulus;
      oz             : real := 0.0);    -- initial rotation about beam's z-axis
    port (terminal t1x, t1y :  kinematic;
          terminal r1z      :  rotational;
          terminal t2x, t2y :  kinematic;
          terminal r2z      :  rotational);
end entity beam2d;
```

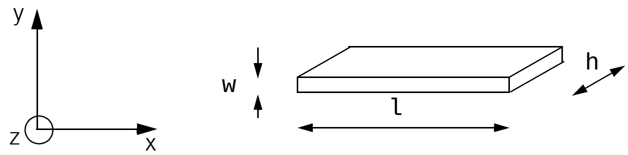

Fig. 7. Entity declaration for mechanical beam model

2.2.2 Reduced Order Models for Bottom-Up Verification

The analytical models of predefined primitives as masses, springs, beams, gaps and so on can be easily combined. Thus, the system behavior can be evaluated and the components can be specified. In many cases the real design of the nonelectrical components is finally carried out with the help of simulation programs that can solve systems of partial differential equations such as the FEM tool ANSYS. For bottom up verification models of the designed nonelectrical components are needed. Order reduction methods allow to derive descriptions from linear PDEs that can be used to establish behavioral models. If reduced order models are applied, the matrices in contrast to (1) cannot be expressed by analytical formulas that depend on generic parameters. They are built up by fixed values. The equations of a reduced order model are given by

$$\tilde{M} \cdot \ddot{\tilde{x}} + \tilde{D} \cdot \dot{\tilde{x}} + \tilde{K} \cdot \tilde{x} = -\tilde{B} \cdot f \qquad (2)$$

$$x = \tilde{B}_a^{\,T} \cdot \tilde{x} \qquad (3)$$

$\tilde{x}$ is a vector of auxiliary quantities of the reduced order model. $\tilde{M}$, $\tilde{D}$ and $\tilde{K}$ are the reduced mass, damping, and stiffness matrices. The matrices of the reduced order model can be derived using projection methods from the system of ordinary differential equations that results from the semi-discretization of the PDEs [Sh99]. $\tilde{B}$ is the incidence matrix of connection points with applied forces, torques etc., *f*. $\tilde{B}_a$ is the incidence matrix to observe across quantities *x* at the connection points. Usually $\tilde{B}_a$ equals $\tilde{B}$. In a VHDL-AMS model $\tilde{x}$ can be declared as a vector of free quantities. *x* and *f* can be declared in accordance with Fig. 5. A natural way would be to read the values of $\tilde{M}$, $\tilde{D}$, $\tilde{K}$ and $\tilde{B}$ from a file. That means the I/O-facilities of VHDL-AMS are very helpful especially during the bottom up verification.

Example

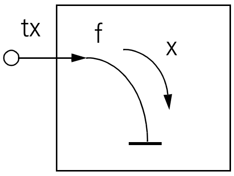

Fig. 8. Interface of sensor(reduced)

The reduced order model shall only take into consideration the translational movement in one direction. Fig. 8 shows the block diagram. Fig. 9 shows the VHDL-AMS model. The matrices $\tilde{M}$, $\tilde{D}$, $\tilde{K}$ and $\tilde{B}$ are read from a file with the user-defined function `read` into arrays `mt`, `dt`, `kt` and `bt` of type `real_vector` and length `n`. The function `read` was declared in the package `encapsulation` and compiled into the library `mems`. In ADVance MS it is implemented using the foreign C language interface. The same model can be used for different dimensions of the matrices. The free quantities `xt, dxt, d2xt` correspond to $\tilde{x}$, $\dot{\tilde{x}}$ and $\ddot{\tilde{x}}$ resp. The local function `reduced_model` realizes equation (2). This function uses a user-defined overloaded operator from the package `operator` of library `mems` to implement the matrix vector multiplications of (2). The model was successfully applied in time domain simulation.

```
library disciplines, mems;
use disciplines.kinematic_system.all;
use mems.operator.all;

entity sensor is
  generic (n           : integer := 6;
           file_name   : string  := "tilde.dat");
  port    (terminal t1 : kinematic);
end entity sensor;

architecture reduced of sensor is
   constant mt  : real_vector (1 to n*n) := mems.encapsulation.read ("m"&file_name);
   constant dt  : real_vector (1 to n*n) := mems.encapsulation.read ("d"&file_name);
   constant kt  : real_vector (1 to n*n) := mems.encapsulation.read ("k"&file_name);
   constant bt  : real_vector (1 to n)   := mems.encapsulation.read ("b"&file_name);

   quantity xt, dxt, d2xt       : real_vector (1 to n);
   quantity x across f through  t1;

   constant lhs : real_vector (1 to n) := (others => 0.0);

   function reduced_model (n : integer;
                           mt, dt, kt, d2xt, dxt, xt, bt : real_vector; f : real)
   return real_vector is
      variable i1, i2 : integer;
      variable r  : real_vector (1 to n);
   begin
      for i in 1 to n loop
          i1   := 1 + (i-1)*n;
          i2   := i*n;
         r(i) := mt(i1 to i2) * d2xt + dt(i1 to i2) * dxt + kt(i1 to i2) * xt  + bt(i)*f;
      end loop;
      return r;
   end function reduced_model;

begin
   dxt  == xt'dot;
   d2xt == dxt'dot;
   lhs  == reduced_model (n, mt, dt, kt, d2xt, dxt, xt, bt, f);
   x    == bt*xt;
end architecture reduced;
```

Fig. 9. Example of a reduced order model

2.3 Language Requirements

Besides the discussed and used language statements also other facilities of VHDL-AMS are helpful for MEMS modelling. For instance micro electromechanical systems often consist of regular structures. The GENERATE statement of VHDL-AMS that can be applied on concurrent and simultaneous statements may help to describe such structures (see example in the next section). Comprising we can say that the following VHDL-AMS language constructs are helpful to support modeling and simulation in the MEMS design flow (especially if Kirchhoffian networks are used)

- Concept of entity and architectures to describe a system on different levels of abstraction
- Declaration of packages with typical technology data
- Declaration of functions that are widely used in MEMS models
- Overloading of at least addition and multiplication for real-valued one and multidimensional arrays and their application in simultaneous statements
- Usage of file I/O operations
- Description of regular structures with the GENERATE statement

It depends on the target simulators which of these facilities can be used. Models can be exchanged if the corresponding language constructs are available in the target simulation engines.
Furthermore some general developments would support MEMS modelling

- Declaration of mixed RECORD natures
- Declaration of the across and through subtypes with special tolerance aspects for MEMS applications

Compared to other solutions the great advantage of VHDL-AMS is the availability of the usual simulation algorithms (DC, AC and transient) in a dedicated simulation engine. Typical MEMS question as e. g. the determination of resonance frequencies and values of pull-in voltages can answered by the formulation of adequate simulation problems.

3 Examples and Experiences

On this basis some elementary Kirchhoffian elements (similar to those used in SUGAR) were implemented using VHDL-AMS. Beam models describe linear mechanical beams that are in-plane and can rotate out-of-plane. Gap models describe electrostatic gaps, which consist of two mechanical beams that are the movable electrodes. Models of external forces, anchors, electrostatic comb, mass, spring, and damping are also available. Besides these elements all other facilities of VHDL-AMS can be used.
With these elements some subsystem specifications were simulated in correspondence to the step “System Description with specified subsystems” of Fig. 2. The pictures that describe the geometric structures are created with the help of SUGAR [BB01]. The simulations were carried out using ADVance MS.

3.1 Accelerometer

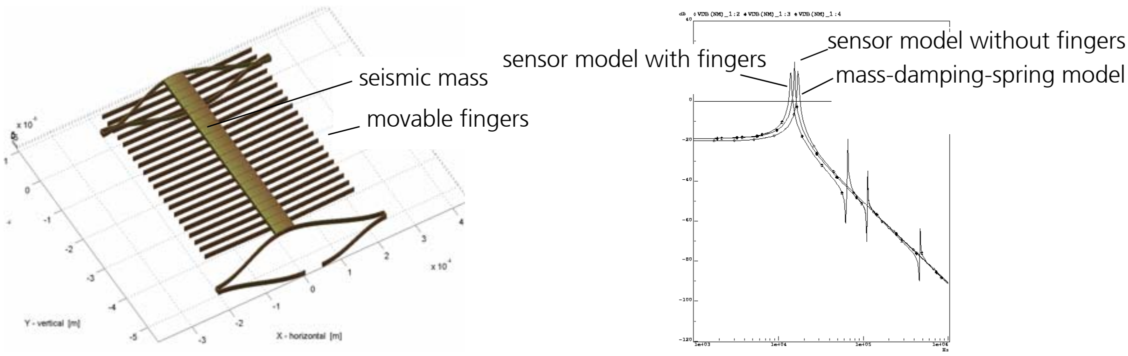

Fig. 10. Representation of the movable parts of the sensor and transfer characteristics

Fig. 10 represents a micro mechanical accelerometer. The sensor consists of a movable beam (seismic mass), suspended by two spring tethers on either end. Movable fingers are attached to the mass. The fingers establish together with fixed plates (not represented in Fig. 10) capacitances that are evaluated by an electronic circuit. If the seismic mass is moved by an external force, the capacitances depend on this force. The structure is highly regular. The model consists of mechanical beams of different dimensions. The electrostatic forces are modeled by comb models. Fig. 10 shows the displacement of the sensing element at the first resonance frequency. The transfer characteristics are simulated for different modeling approaches using AC analysis. The transfer characteristics describe the displacement of the seismic mass for different frequencies.

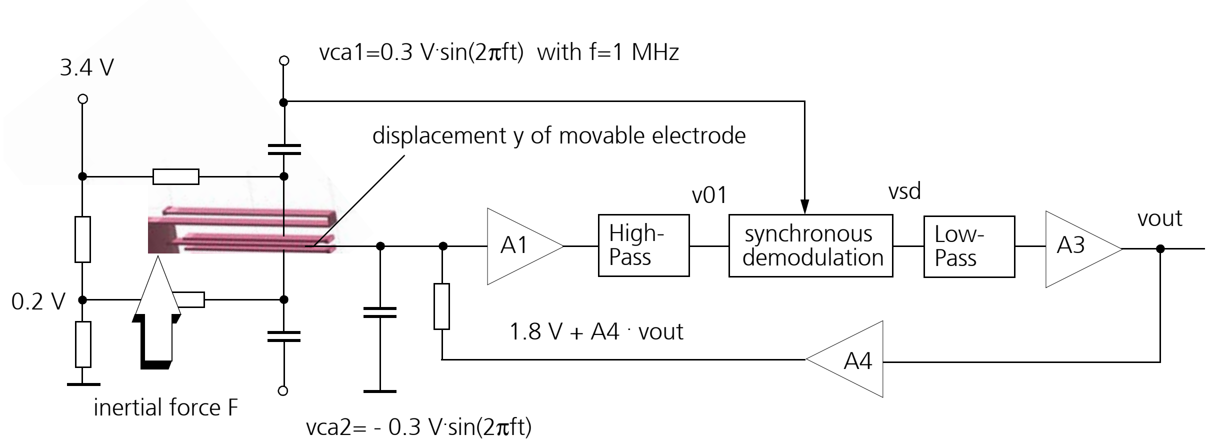

Fig. 11. Representation of the movable parts of the sensor and transfer characteristics [Bao00]

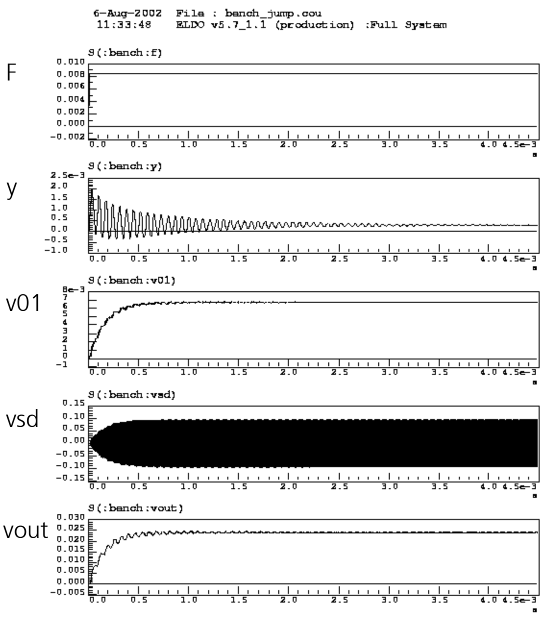

Fig. 12. Simulation results

Similar micro mechanical devices are used in force-balanced accelerometers like ADXL series from Analog Devices and Siemens. Fig. 11 shows a schematic drawing of the sensing element and the electronic circuitry of such an accelerometer. The inertial force F causes the movable fingers to displace. This displacement y causes a voltages in phase with the carrier vca1 at the input terminal of the buffer amplifier A1. The signal is amplified. The output of the high pass filter v01 is fed to the synchronous demodulator. The output vsd is filtered by a low pass filter and amplified. The output of the amplifier A3 is the output vout of the circuit. The output signal is also fed back to the seismic mass. In more detail the circuit is described in [Bao00].

The basic functionality can easy be simulated using VHDL-AMS. For test purposes behavioral models of the electronic subsystems and a model of the sensing element using basic MEMS primitives (see section 2.2.1) were used. The advantage of this approach is an easy combination of these MEMS primitives with other user-defined models.

3.2 Electrostatic Beam Actuator

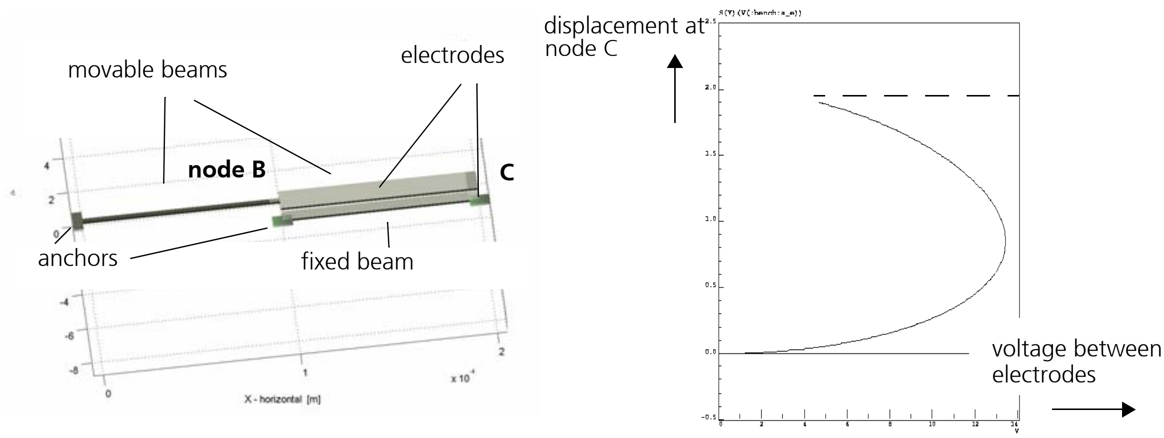

Fig. 13. Electrostatic beam actuator and characteristic of tip displacement

For case studies of the electrostatic forces we use the beam actuator shown in Fig. 13. It is taken from [BB01]. The top actuator plate moves as result of the voltage between the electrodes to the fixed beam. The displacement of the tip w.r.t. the voltage between the electrodes is shown by the characteristic. In the rest position the displacement is zero. It begins to increase with increasing the voltage until the pull-in voltage is achieved. The pull-in voltage is characterized by the coordinates of the turning point of the static characteristic. Then the characteristic turns back. Because of the nonlinearities the pull-in phenomena do widely exist in electrostatic actuators. The effect may be either derogatory or useful [NB01].

Those characteristics can be calculated using path following algorithms [UC84]. It can be shown that these algorithms can be interpreted in a lot of cases as a network analysis problem [Ha82]. These network analysis problems can also be expressed using VHDL-AMS.

Using this way the characteristic from Fig. 13 was determined. The model of the system is built up by beam and gap models. The anchors fix across quantities. That means they connect terminals to the associated reference nodes. For the curve tracing the voltage source between movable and fixed beam has to be replaced by a special circuit element. For details see [HaR02].

4 Conclusion

VHDL-AMS covers the requirements of different methods applied in the MEMS design process. The standard offers very good possibilities that support these objectives. With respect to the implementation of the standard in available simulation engines some of the facilities cannot be applied at the moment to obtain models that can be exchanged between different simulators. This has to be taken into consideration during the creation of models.
Nevertheless practical examples can be modeled and simulated using the available language constructs.

Parts of this work were founded by German Bundesministerium für Bildung, Wissenschaft und Technologie (Project EKOSAS "Entwurf komplexer Sensor-Aktor-Systeme", Label 16 SV 1161). The authors would like to thank one of the reviewers for an important hint concerning composite record natures.

References

[Bao00] Bao, M.-H.: Micro Mechanical Transducers. Pressure Sensors, Accelerometers, and Gyroscopes. Amsterdam: Elsevier Science, 2000.

[BB01] Bai, Z.; Bindel, D. et al: New Numerical Techniques and Tools in SUGAR for 3D MEMS Simulation. Technical Proceedings of the International Conference on Modeling and Simulation of Microsystems (MSM 2001), Head Island, March 19-21, 2001 (http://bsac.berkeley.edu/cadtools/sugar/sugar/).

[CZB00] Clark, J. V.; Bindel, D. et al: 3D MEMS Simulation Modeling Using Modified Nodal Analysis. Proc. of the Microscale Systems: Mechanics and Measurements Symposium, Orlando, June 8, 2000, pp. 68-75 (/http://bsac.berkeley.edu/cadtools/sugar/sugar/).

[CZP98] Clark, J.V.; Zhou, N.; Pister, K.S.J.: MEMS Simulation using SUGAR v0.5. Tech. Digest, Solid-State and Actuator Workshop, Hilton Head Island SC, pp. 191-196, June 8-11, 1998.

[De99] Dewey, A.; Dussault, H. et al: Energy-Based Characterization of Micromechanical Systems (MEMS) and Component Modeling Using VHDL-AMS. Proc. Modeling and Simulation of Microsystems, Semiconductors, Sensors, and Actuators (MSM 99), San Juan, Puerto Rio, April 19-21, 1999.

[Fe00] Fedder, G. K.: Top-Down Design of MEMS. Technical Proceedings of the International Conference on Modeling and Simulation of Microsystems (MSM 2000), San Diego, CA, March 27-29, 2000.

[FJ99] Fedder, G. K.; Jing, Q.: A Hierachical Circuit-Level Design Methodology for Micromechanical Systems. IEEE Trans. on CAS-II 46(1999)10, pp. 1309-1315.

[Ha82] Haase, J.: Computation of Transfer Characteristics of Multivalued Resistive Nonlinear Networks. Short Contributions SSCT '82, July 12-16, 1982, Prag, pp. 268-272.

[HaR02] Haase, J.; Reitz, S.: Pull-Spannungsberechnung mit VHDL-AMS. Proc. ASIM, Rostock, September 10-13, 2002.

[IEEE99] IEEE Std 1076.1-1999 Standard VHDL Analog and Mixed-Signal Extensions. The Institute of Electrical and Electronic Engineers, 1999.

[MGC] ADVance MS simulation engine of Mentor Graphics, Inc. (http://www.mentor.com/ams/adms.html)

[NB01] Nemirovsky, Y.; Bochobza-Degani, O.: Methodology and Model for Pull-In Parameters of Electrostatic Actuators. Journal of Microelectromechanical Systems 10 (2001) 4, pp. 601-615.

[NBL98] Neul, R.; Becker, U.; Lorenz, G.; Schwarz, P.; Haase, J.; Wünsche, S.: A modeling approach to include mechanical microsystem components into system simulation. Proc. Design, Automation and Test in Europe (DATE'98), Paris, February 1998, pp. 510-517.

[OVC00] Oudinot, J.; Vaganay, C.; Robbe, M.; Radja, P.: Mixed-Signal ASIC top-down and bottom-up design methodology using VHDL-AMS. Proc. XV Conference on Design of Circuits and Integrated Systems, Montpellier, November 21-24, 2000.

[Sen98] Senturia, S. D.: CAD Challenges for Microsensors, Microactuators, and Microsysrems. Proceedings of the IEEE 86 (1998) 8, pp. 1611-1625.

[Sh99] Sheehan, B. N.: ENOR: Model Order Reduction of RLC Circuits Using Nodal Equations for Efficient Factorization. Proc. 36th Design Automation Conference, 1999, pp. 17-21.

[UC84] Ushida, A.; Chua, L.O.: Tracing Solution Curves of Non-Linear Equations with Sharp Turning Points. Circuit Theory and Applications 12 (1984), pp. 1-21.

[WG] Information of Standard Package IEEE Working Group (http://www.vhdl.org/analog/)