Chapter 9

COMPACT MODELING OF THE MOSFET IN VHDL-AMS

Christophe Lallement[1,a], François Pêcheux[2], Alain Vachoux[3] and Fabien Prégaldiny[1,b]

[1] *InESS (UMR 7163)/ ENSPS, Parc d'innovation, BP 10413, F-67412 Illkirch Cedex, France*

E-mail: [a] christophe.lallement@ensps.u-strasbg.fr,
[b] fabien.pregaldiny@iness.c-strasbourg.fr;

[2] *LIP6 Integrated Systems Architecture Department, Université Pierre et Marie Curie, 12, rue Cuvier, 75252 Paris Cedex 05, France*

E-mail: francois.pecheux@lip6.fr;

[3] *LSM Microelectronic Systems Laboratory, Ecole Polytechnique Fédérale de Lausanne, CH-1015 Lausanne, Switzerland*

E-mail: alain.vachoux@epfl.ch.

Abstract: In this chapter, we present the capabilities of the VHDL-AMS hardware description language for developing compact models. After a brief description of the VHDL-AMS language, we present two meaningful case studies on design oriented models of MOSFET.

The first study focuses on the EKV v2.6 MOSFET model and takes into account the thermo-electrical interaction and the extrinsic aspects. The EKV v2.6 model uses linearization with respect to surface potential, resulting in physically well-based expressions for the whole model.

The second study is a simplified version of the MM11 Philips model that takes into account the quantum mechanical effects. MM11 is a compact MOSFET model based on the formulation of the surface potential.

Key words: Hardware description language; VHDL-AMS; compact modeling; MOSFET model; thermo-electrical interactions; quantum effects; EKV; MM11.

W. Grabinski, B. Nauwelaers and D. Schreurs (eds.),
Transistor Level Modeling for Analog/RF IC Design, 243–269.

1. Introduction

For the past three decades, hardware description languages (HDLs) have been widely used to model and simulate systems belonging to various engineering fields, from digital and analog electronics to mechanics and chemistry. For a long time, all these fields have been completely separated, each scientific community having its own design methodologies, tools and idiosyncrasies. For example, in the electrical/electronic domain, the SPICE simulator and all its derivatives allow the description of the netlist of a circuit using electrical primitives such as resistors, capacitors, sources and transistors. In an attempt to support the modeling and simulation of non-electrical systems as well, several modeling methods using energy equivalences between the electrical domain and other domains such as mechanical, thermal or fluidic domains have been proposed. With the advent of nano-technologies, the design of innovative integrated devices, like Micro-Opto-Electro-Mechanical Systems (MOEMS), has shifted from vertical only to both vertical and horizontal integration. Using the benefit of all the experience acquired in incremental design, MOEMS design now involves strong "horizontal" interaction of different application-field parts on the very same chip (e.g., mechanical, electrical, thermal, fluidic parts), with partial close coupling between these fields. Neglecting the interaction effects or the cross coupling between parts may have disastrous consequences on the final design in terms of a loss in performance or an increase in design time.

One way of addressing this issue is to use a consistent modeling and simulation framework that allows for the description of systems from different disciplines and for the description of interactions between these systems. This is where the VHDL-AMS HDL comes in action.

2. VHDL-AMS: A Mixed-Signal HDL

VHDL-AMS [1–4] is the result of an IEEE effort to extend the VHDL language to support the modeling and the simulation of analog and mixed-signal systems. The effort culminated in 1999 with the release of the IEEE standard 1076.1-1999 [1].

VHDL-AMS supports the description of continuous-time behavior. For compact modeling, the most interesting feature of the language is that it provides a notation for describing Differential Algebraic Equations (DAE) in a fairly general way. The "==" operator and the way unknown variables are declared allow the designer to write equations in either implicit or explicit format.

VHDL-AMS supports the description of networks as conservative-law networks (Kirchhoff networks) and signal-flow networks (inputs with infinite impedance, outputs with zero impedance). As such, it supports the description

and the simulation of multi-discipline systems at these two levels of abstraction. As a companion standard, the IEEE 1076.1.1-2004 standard includes packages that define types, subtypes, natures and constants for modeling in multiple energy domains [2].[1]

The VHDL-AMS language has a canonical, tool independent, mixed-signal simulation cycle that defines how to simulate a mixed-signal description. It has in addition a formal definition of how to initialize a mixed-signal model. It supports continuous-time analyses such as time-domain, DC, small-signal AC and noise analyses.

VHDL-AMS does not provide any support of the SPICE netlist format, neither directly in the language nor in some standard library. It however provides all the necessary language elements to build libraries of SPICE models. This is certainly helpful when developing compact models.

Any VHDL-AMS design unit may be compiled separately and stored in a library. In addition, VHDL-AMS allows for a clear separation between the interface of a model and its internal description and provides a mechanism to select the submodels to use in a hierarchical description through the mechanism of configuration. Both capabilities hence allow for much flexibility when it comes to model large complex hierarchical systems.

Table 1 presents a synthetic view of the capabilities offered by VHDL-AMS.

Table 1. Key Features of VHDL-AMS [7].

Features class	Feature	VHDL-AMS
Language aspects	Definition	IEEE Std 1076.1-1999 Strict extension to IEEE Std 1076 (VHDL)
	Inheritance	Ada-like. Case insensitive
	Modularity	Separation of external/interface views (entities) and internal views (architectures), packages, configurations
	Genericity	Parameters, generate statements(1)
	Library management	Yes (pre-compiled design units)
	Analog subset	No(2)
Expression of structure	Ports	Event-driven and continuous Conservative and non conservative (signal-flow) Continuous ports are modeless
Expression of behavior	Composition	Hierarchical instantiation of components

(*Contd.*)

[1]However, as the new standard is not yet fully supported in the current tools, we shall use proprietary versions of the packages. The proprietary packages do not actually differ much from the standard ones.

Table 1. Continued.

Features class	Feature	VHDL-AMS
	Conservative semantics	Natures define energy domains, subtypes define nature attributes; no predefined natures. Terminal and branch quantities
	Objects	Terminals, quantities, signals, variables, constants
	Statements	Concurrent, sequential, continuous (simultaneous and procedural) Continuous statements can be freely mixed with concurrent statements
	Expression of DAEs(3)	Explicit and implicit form of equations(4) supported Simultaneous(5) and procedural(6) formulations Derivative attribute 'dot only possible on quantities. Attribute can be chained for higher order derivatives(7) Mathematical functions defined in separate standard IEEE 1076.2 Piecewise defined behavior supported(8)
	Discontinuity handling	Discontinuities must be explicitly announced in the model User-defined re-initialization after discontinuity supported
Conservative semantics	Energy domains	Natures define energy domains and subtypes define nature attributes. No predefined natures(9) Branch quantities(10)
	Formulation	Equation-oriented formulation with simultaneous statements(11) No specific circuit graph representation enforced
Signal-flow semantics	Model interface	Directional interface (free) quantities(12)
	Functional blocks	Laplace and z transforms
Mixed-signal aspects	Interfaces	A/D and D/A interface language attributes ('ramp, 'slew, 'above) No direct port association(13)
	Behavioral interactions	Access of discrete signals in continuous context Access of continuous quantities in discrete context

(*Contd.*)

Table 1. Continued.

Features class	Feature	VHDL-AMS
Simulation controls	Solvability	Solvability check done at design unit level(14)
	Timestep	Timestep size may be bounded
	Tolerances	Generic string annotation not formally linked to simulator(15)

(1) Generate statements offer macro-like capabilities in the text of the model.
(2) It is possible to develop packages to support SPICE level modeling. No standard packages exist yet.
(3) DAE = Differential Algebraic Equation.
(4) An equation in the explicit form looks roughly like an assignment, e.g. $x = f(y, z)$, while an equation in the implicit form typically requires iterations to compute the unknowns, e.g. $x = f(x, y, z)$.
(5) Simultaneous statements are basically equations that may be given in any order in the model.
(6) Procedural statements have to be given in a particular order. The VHDL-AMS tool used did not support simultaneous procedural statements yet, so we used functions instead.
(7) To maintain good numerical accuracy it is recommended to hold higher order derivatives in local quantities and to only use first order derivatives.
(8) Continuous behavior can be defined by regions of operation.
(9) A draft VHDL-AMS standard package for multiple energy domain support is currently under IEEE ballot.
(10) The direction of the flow in the branch and of the potential difference is defined in the branch quantity declaration.
(11) Quantities are the unknowns. As far as the language is defined, the order in which the simultaneous statements is not important. We anyway faced some non-convergence issues with "misplaced" simultaneous statements (tool issue).
(12) Direction is used for solvability checks.
(13) It is not allowed to associate formal and actual ports of different natures or types. Explicit interface code has to be added in the model when pre-defined attributes are not enough. It is also expected that tools may help in inserting proper interface code when working at schematic level.
(14) This basically checks that the number of unknowns matches the number of equations. Although the rules that define what is considered as an unknown and what is considered as an equation are clearly defined in the language reference manual, it may become pretty hard to figure out what is missing when a complex model such as the full EKV MOS model (with more than 100 quantities) does not comply with the solvability condition. In addition, the current implementation of the Mentor tool imposes to have the same number of simple simultaneous statements in each branch of a conditional or selective simultaneous statement.
(15) Current VHDL-AMS simulators are using their own tolerances that may be set in the tool's environment.

For this chapter, the EDA tools used for implementing and simulating the models are Advance MS from Mentor Graphics and Simplorer from Ansoft.

To be complete, it should be noticed that VHDL-AMS has a direct competitor: Verilog-AMS [5–7]. The Verilog-AMS language also supports the modeling and the simulation of analog and mixed-signal systems but has not been submitted yet to IEEE for standardization [7].

3. Compact Modeling of the MOSFET

Compact models for circuit simulation have been at the heart of CAD tools for circuit design over the past decades, and are playing an ever increasingly important role in the nanometer system-on-chip (SoC) era. The requirements for a competitive compact MOSFET model rely on a complex trade-off between accuracy, complexity and applicability for any advanced technology. To achieve this task, in particular for devices entering the sub-100-nm regime, it is essential to accurately model the physical effects that govern the MOSFET behavior. This is the reason why a new generation of MOSFET models (the 4th one) is being developed (see Figure 1). Conventional models of the 3rd generation like BSIM3 / BSIM4 [8] and MM9 [9] are based on the formulation of the threshold

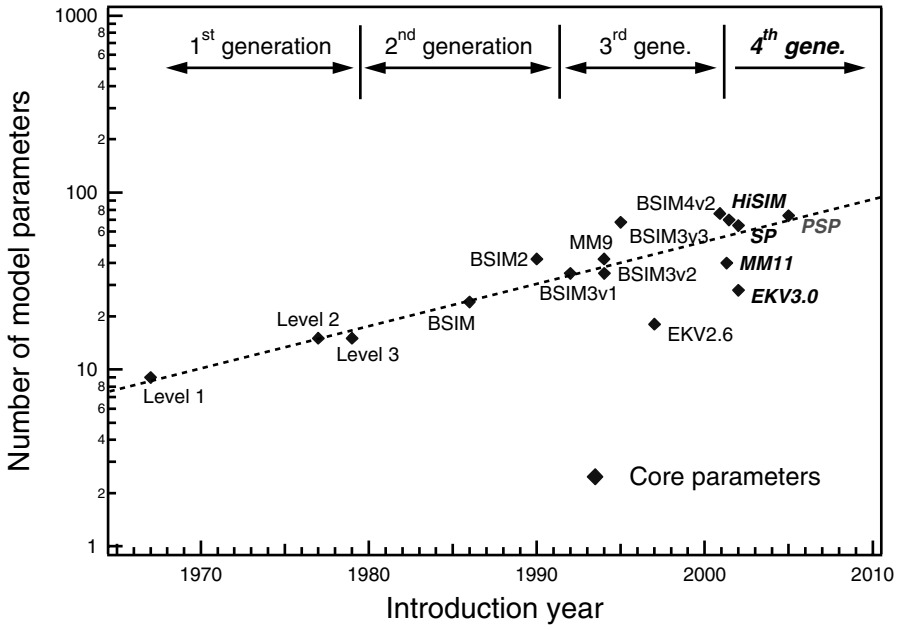

Figure 1. Number of core parameters (i.e., without geometrical or scaling parameters) for the major compact MOSFET models versus the introduction year of models.

voltage. The case of EKV v2.6 [10] is quite different since the model is based on the formulation of the inversion charge density. In fact, EKV v2.6 should already be considered as a 4th generation MOSFET model. For convenience, we use the term "inversion charge model" to refer to such a model. From the historical point of view, the major characteristics of the 3rd generation were [11]:

- the "original intent" to simplicity (in contrast to the 2nd generation models like BSIM and BSIM2),
- a small number of physically-based parameters,
- an improved mathematical conditioning,
- a single model equation for all regions of device operation,
- the use of smoothing functions.

Unfortunately, the most used model in the design community (BSIM3/4) has forgotten the original intent of both simplicity and small number of parameters, as depicted in Figure 1. All 3rd generation models (except EKV v2.6 which *is not* a threshold-voltage-based model) describe different operating regions with different equations. As a result, they are usually called "piece-wise" or "regional" models [11]. They often use unphysical parameters to smooth characteristics between the different operation modes. This artificial modeling may lead to unphysical behavior of the drain current and transconductance in the transition region between weak and strong inversion, the so-called moderate inversion region [12]. This region is however of crucial importance, not only for low-voltage and low-current analog applications, but also for digital circuits, owing to the reduction of supply voltage in modern CMOS technologies. Moreover, for most analog applications the device is typically biased in this region, i.e. just above threshold. Another drawback of regional models is that the drain current exhibits a discontinuity at the transition between linear and saturation regions due to the use of the drift approximation. Consequently, additional parameters are needed to get continuous characteristics through different operation modes, and the total number of parameters dramatically increases (see Figure 1).

In contrast to regional models, the compact models of the 4th generation like EKV 3.0 [13], HiSIM [14], MM11 [15], SP [16] and now PSP [17] are inherently single-piece and give an accurate and continuous description of characteristics in all regions of operation. They are generally charge sheet models based on the formulation of the surface potential, except EKV 3.0 which is a charge sheet model based on the formulation of the inversion charge density. Using the drift-diffusion approximation these models are more able to support future technology requirements.

In conclusion, and for the sake of completeness, it should be noted that a new compact MOSFET model called PSP is now available [17]. It has been developed by merging the best features of two surface-potential-based models: SP

(developed at The Pennsylvania State University) and MM11 (developed by Philips Research). The PSP model is a symmetrical model, and gives an accurate physical description of the transition from weak to strong inversion and includes an accurate description of all physical effects important for modern and future CMOS technologies. It is suitable for digital, analog and RF circuit design.

4. The EKV MOSFET Model v2.6

The EPFL EKV MOSFET model v2.6 is a scalable and compact simulation model built on fundamental physical properties of the MOS structure. This model is dedicated to the design and simulation of low-voltage, low-current analog, and mixed analog-digital circuits using submicron CMOS technologies.

4.1. Basic Version

The basic version of the EKV v2.6 MOSFET model [10] is a charge-based compact model. It consistently describes effects on charges, transcapacitances, drain current and transconductances in all regions of operation of the MOSFET (weak, moderate, strong inversion) as well as conduction to saturation. The effects modeled in this model include all the essential effects present in submicron technologies. For quasi-static dynamic operation, both a charge-based model for the node charges and transcapacitances, and a simpler capacitances model are available.

4.2. Features Specific to Submicron CMOS Technologies

The LDD regions in the sub- and deep- submicron CMOS technologies introduce additional parasitic resistances between the source/drain electrode and the channel, as well as parasitic capacitances.

A problem with all these parasitic elements is their non-linear and bias dependent behavior. An efficient MOSFET model dedicated to deep-submicron design must imperatively take into account these elements. These features specific to submicron technologies are not included in the basic version of the EKV MOSFET model v2.6. They have been added to the basic code of the EKV MOSFET model v2.6, in a modified version of this model, implemented in VHDL-AMS.[2]

[2]The full VHDL-AMS code of the model is available on a website [20].

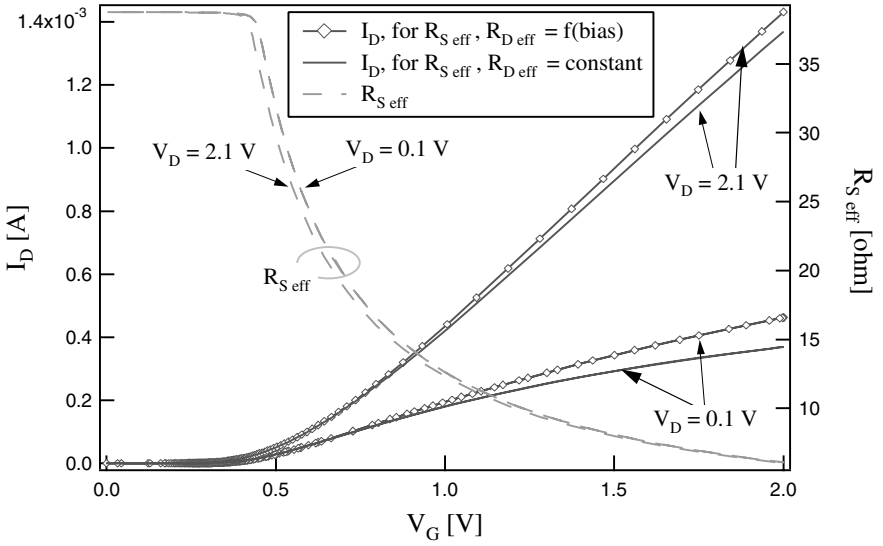

Figure 2. Simulation of I_D and $R_{S\text{eff}}$ versus V_G characteristics for two drain bias.

In Figure 2 and Figure 3 are plotted some results showing the influence of these parasitic elements. All these simulation results in VHDL-AMS are made with a n-channel transistor of $\mathrm{W/L} = 1.5\,\mu\mathrm{m}/0.15\,\mu\mathrm{m}$.

More information of the modeling of these effects and of the simulation results can be found in [18, 19].

4.2.1. Series parasitic resistance

A typical characteristic of series parasitic resistance can be observed in Figure 2, for two different drain bias.

Not taking into account the bias dependent of this resistance introduces some important errors on the drain current level, mainly for small V_D bias. These variations can considerably affect the parameters extraction procedures where, classically, channel length and series resistance are extracted altogether, at small V_D bias [21].

4.2.2. Parasitic capacitances

The dynamic behavior of a MOSFET in deep submicron technology is strongly affected by its extrinsic capacitance formed by the overlap capacitance

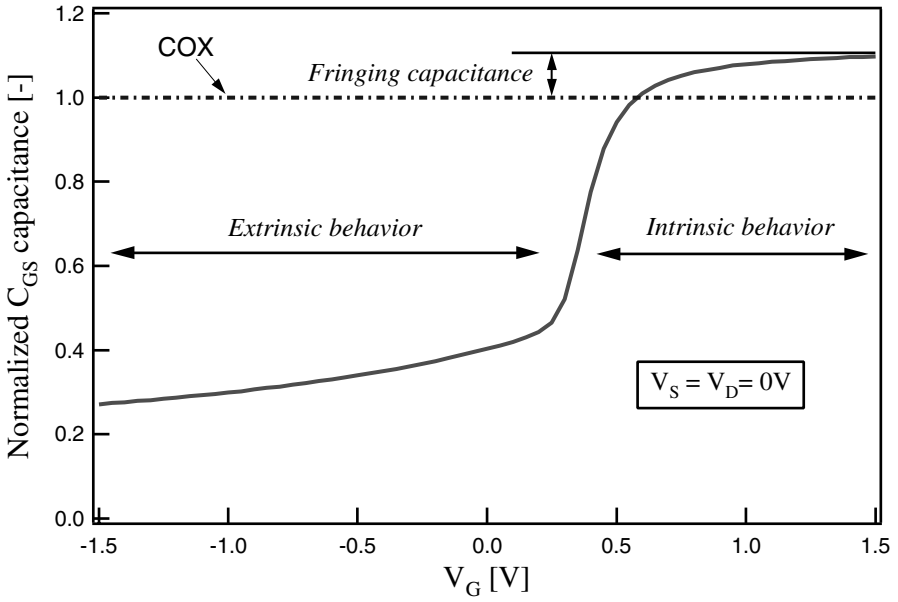

Figure 3. Simulation of the normalized global gate-source capacitance (CGS/COX).

and the fringing capacitance (see Figure 3) [22]. The fringing capacitance is constant, but the overlap capacitance is bias dependent.

As seen in Figure 3, the parasitic capacitances represent a more and more important part of the global capacitance of the MOSFET (more 35% for a 0.15 μm technology) as observed in the accumulation region. The influence of the fringing capacitance can be observed in the strong inversion region; it represents the additional capacitance to COX (COX = W.L.Cox).

4.3. Modeling of an Inverter with Thermo-Electrical Interactions

As the transistor size decreases, thermal interactions between devices on the same chip increase. These thermal effects are constantly amplified by the growing power density, and a failure in their estimation at an early development stage of the design often means extra costs and delays.

For the system designer, one of the major interests of VHDL-AMS is the simplicity with which models involving various physical domains (electronical, thermal, optical, mechanical, etc) can be interconnected. We illustrate this with an example: a CMOS inverter with thermo-electrical interactions.

4.3.1. VHDL-AMS implementation of the EKV model v2.6

In this inverter, the pMOS and nMOS transistor behaviors are described using the EKV MOSFET model v2.6. Several electrical parameters of this model are highly dependent on temperature, namely the threshold voltage, the mobility, the thermal voltage, etc... Their respective temperature variations are taken into account by appropriate coefficients in the model equations [10].

To take the self heating in the MOSFET into account, its packaging must also be considered. Classically, we have modeled the heat diffusion through solid materials by sourcing dissipated power into a thermal RC network [18, 23], which represents the material properties of the different layers. The temperature profile is the result of a heat flow in the thermal network.

Figure 4 shows how thermal-electronic interactions between an n-MOSFET and its direct environment can be modeled.

In such networks, energy conservation states that the sum of all power contributions at a thermal node equals zero, and that the temperature at all terminals connected to a thermal node be identical. Thermal evolution of a system is thus ruled by the very same Kirchhoff laws dictating the behavior of conservative systems: voltage becomes the across quantity temperature and current becomes the through quantity heat flow.

The IEEE standard 1076.1.1-2004 includes the `thermal_system` package that defines the `thermal` nature and its related characteristics. The principle is to introduce a thermal `terminal` and a thermal branch with associated `through` and `across` quantities respectively bound to heat flow (or power) and temperature.

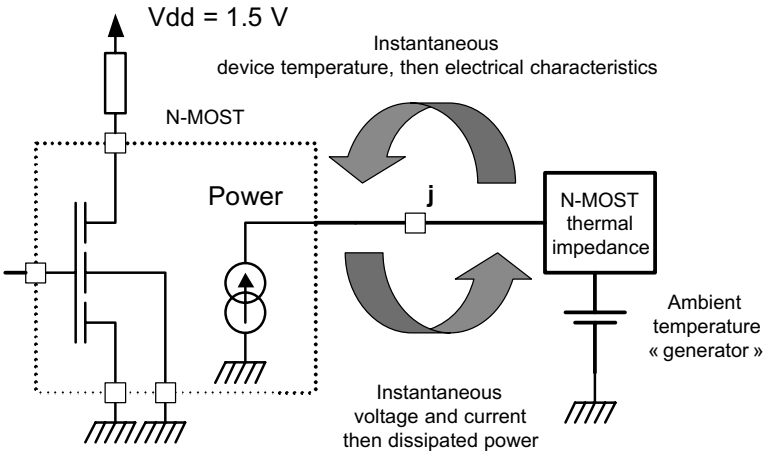

Figure 4. Modeling electro-thermal interactions.

In this paper, we present a simplified version of the EKV MOSFET model with thermo-electrical interactions (Figure 5) [7] as the full version[3] would have needed several pages of code. This single transistor model is valid for both pMOST and nMOST.

```
(1)   library ieee; use ieee.math_real.all;
(2)   library ieee_proposed;
(3)     use ieee_proposed.energy_systems.all;
(4)     use ieee_proposed.electrical_systems.all;
(5)     use ieee_proposed.thermal_systems.all;
(6)   entity mos is
(7)     generic (
(8)       MTYP : real := 1.0; – NMOS: 1.0, PMOS: -1.0
(9)       – geometrical parameters
(10)      WEFF : real := 1.0*MICRO;      – effective channel width
(11)      LEFF : real := 0.15*MICRO;     – effective channel length
(12)      – threshold voltage and substrate body effect parameters
(13)      VT0 : real := 0.4;             – long channel threshold voltage (NMOS!)
(14)      PHI : real := 0.97;            – bulk Fermi potential
(15)      GAMMA: real := 0.71;           – body effect parameter
(16)      – mobility parameters
(17)      KP : real := 453.0*MICRO;      – transconductance parameter
(18)      THETA: real := 50.0*MILLI;     – mobility reduction coefficient
(19)      – temperature coefficients
(20)      TCV : real := 1.5*MILLI;       – temp. coef. of threshold voltage
(21)      BEX : real := -1.5);           – temp. coef. of transcond. parameter
(22)    port (
(23)      terminal td, tg, ts, tb: electrical;
(24)      terminal tj: thermal);
(25)  end entity mos;
(26)  architecture ekv_simple of mos is
(27)    constant KOQ: real := K/Q;
(28)    constant TEMPREF: real := 300.15;
(29)    – electrical branch quantities
(30)    quantity vg across tg to tb;
(31)    quantity vd across td to tb;
(32)    quantity vs across ts to tb;
(33)    quantity ids through td to ts;
(34)    – thermal branch quantities
(35)    quantity gpower through thermal_ref to tj;
(36)    quantity temp across tj to thermal_ref;
(37)
(38)  function i_v (constant v: real) return real is
(39)    variable x: real;
(40)  begin
(41)     return (log(1.0 + 0.5*exp(v)))**2;
(42)  end function i_v;
```

[3]Different VHDL-AMS simulation results of the thermo-electrical interactions in a MOSFET (with the full version of the EKV v2.6 MOSFET model [20]) can be found in [19].

```
(43)
(44)  function f_id (temp, vg, vs, vd: real) return real is
(45)    variable id, vt, ratio, eg, egref: real;
(46)    variable vto_th, kp_th: real;
(47)    variable vgp_0, vgp, vp, iff, irr, beta, n: real;
(48)  begin
(49)    vt := KOQ*temp + 1.0e-6;
(50)    ratio := abs(temp/TEMPREF + 1.0e-6);
(51)    vto_th := MTYP*(VT0 - TCV*(temp - TEMPREF));
(52)    kp_th := KP*(ratio**BEX);
(53)    vgp_0 := vg - vto_th + PHI + GAMMA*sqrt(PHI);
(54)    vgp :=0.5*(vgp_0+sqrt(vgp_0*vgp_0+1.0e-3));
(55)    vp := vgp - PHI - GMA*(sqrt(vgp+0.25*GMA*GMA)-0.5*GMA);
(56)    iff := i_v((vp - vs)/vt);
(57)    irr := i_v((vp - vd)/vt);
(58)    beta := kp_th*(WEFF/LEFF)*(1.0/(1.0 + THETA*vp));
(59)    n := 1.0;
(60)    return 2.0*n*beta*vt*vt*(iff - irr) + 1.0e-10;
(61)  end function f_id;
(62)
(63)  begin
(64)      ids == MTYP*f_id(temp, MTYP*vg, MTYP*vs, MTYP*vd);
(65)      gpower == abs(ids*(vd - vs));
(66)  end architecture ekv_simple;
```

Figure 5. VHDL-AMS model of a simple EKV MOST model. (n- and p- channel).

As we can see in Figure 5, a traditional VHDL-AMS file first contains references to the used libraries (lines 1–5). In this EKV model, electrical and thermal domains are requested. As previously mentioned in Section 2, we still use a proprietary version of the `thermal_system` package that is available in the `ieee_proposed` library.

For the circuit designer, the most important part is the interface, known as an `entity` in VHDL-AMS (lines 6–25). The interface ports are the four standard electrical pins of a MOSFET (line 23), plus an additional thermal pin to account for dynamic thermal exchanges between the transistor and its environment (line 24). The order in which terminals are specified in a branch quantity declaration defines the direction of the flow.

In VHDL-AMS, the `generic` statement in the `entity` allows the designer to define parameters whose values can be overridden during instantiation of the sub-model. Typically here, the geometrical parameters (`weff` and `leff`)and the electrical parameters (`VTO, PHI, ...`) of the transistor are defined as generic.

The `MTYP` generic parameter allows defining the type of the MOS transistor and also the sign of some relevant voltages and parameters (defined in line 8, used in lines 51 and 64). Note that some actual parameters in the MOS instances must anyway have the right sign (e.g., `VT0` in Figure 5).

The MOSFET behavior is defined in a separate architecture called `EKV_simple` (lines 26–66). Lines 30–36 declare a number of *branch quantities* that correspond in VHDL-AMS to the unknowns of the system of equations to be solved by the analog solver. These branch quantities are defined between two terminals and represent `across` and `through` aspects.

These branch quantities are either `electrical` (lines 30–33) or `thermal` (lines 35–36).

Considering the thermal `port`, the temperature `temp` is measured between the port and the thermal reference (line 36), while the heat `gpower` is flowing out the device from the thermal reference to the port (line 35). This way, the thermal interaction is really bi-directional and the self-heating behavior of the device is properly taken into account with the power computation (line 65).

As the electrical behavior of the EKV MOSFET model is naturally procedural, it is more efficient to use the sequential statements provided in VHDL-AMS. The simultaneous `procedural` statement could be used, but, as it is not yet supported in the Mentor graphics tool, we had to use a function instead, namely the `f_id` function, to implement the computation of the drain current (lines 44 to 61). The equation of the drain current is then implemented in a single simultaneous statement with the appropriate signs for the function arguments to account for the actual model type (line 64). Note that all terminal potentials are defined relatively to the bulk terminal, a specificity of the EKV MOSFET model.

To illustrate the interest of such a model, the simulation of the charge (*QI*) and of the transconductance (*Gm*) versus the gate voltage (*VG*), with and without thermal coupling, is given in Figure 6[4].

As we can see, not taking the thermal coupling into account in a transistor (or a circuit) can lead to some important errors in the estimation of the electrical performances of the device.

4.3.2. VHDL-AMS implementation of the CMOS inverter

The CMOS inverter is composed of one nMOS and one pMOS transistor and is connected to its direct environment as shown in Figure 7.

When located on the same substrate, thermo-electronic interactions take place between the nMOS transistor and pMOS transistor. In this CMOS inverter, the nMOS and the pMOS are thermally interconnected through a coupling thermal resistance: `Rcoupling`. The inverter is excited by a squarewave stimulus. The two thermal networks represent the thermal constants of the various material layers of each transistor.

Figure 8 shows the hierarchical tree of sub-models in the VHDL-AMS testbench.

[4]These results are obtained with the full VHDL-AMS code [20].

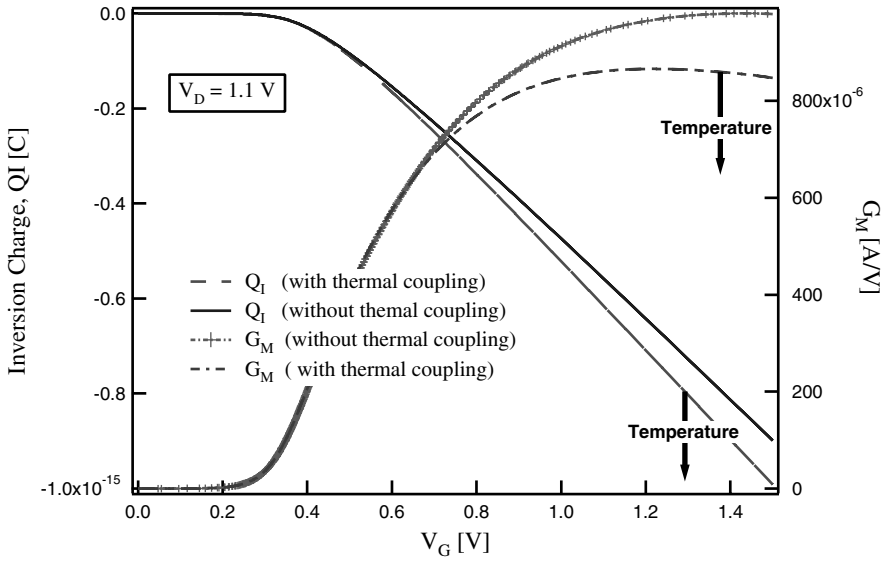

Figure 6. Characteristics of the inversion charge *QI* and the transconductance *Gm* vs. *VG*.

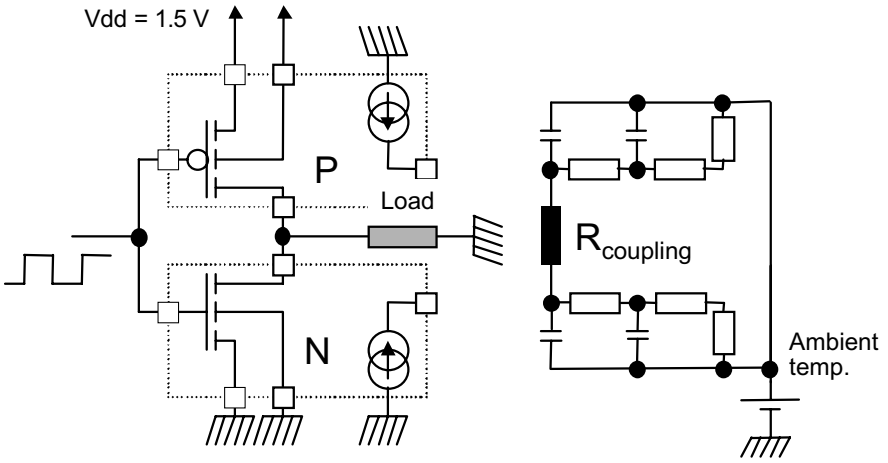

Figure 7. The CMOS inverter and its direct environment.

The thermal network is modeled by thermal resistor and thermal capacitors. The VHDL-AMS models of the thermal capacitance, the thermal resistance, and the ambient heat source (thermal generator) are given in Figure 9 [7, 24, 25]. The thermal resistor and capacitor models are straightforward equivalents of their electrical counterparts.

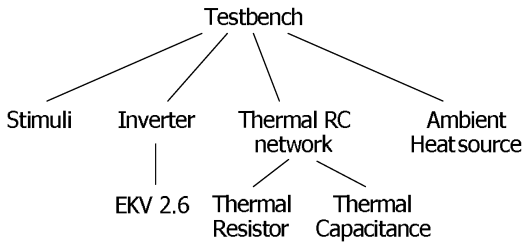

Figure 8. The VHDL-AMS testbench hierarchical tree.

```
(1)   – Behavioural model of a thermal capacitor
(2)   library ieee_proposed;
(3)      use ieee_proposed.energy_systems.all;
(4)      use ieee_proposed.thermal_systems.all;
(5)
(6)   entity capth is
(7)      generic (CVAL: real := 0.1*PICO);
(8)      port (terminal tp, tm: thermal);
(9)   end entity capth;
(10)
(11)  architecture bce of capth is
(12)        quantity temp across hfl through tp to tm;
(13)  begin
(14)        hfl == CVAL*temp'dot;
(15)  end architecture bce;
(16)
(17)  – Behavioural model of a thermal resistor
(18)  entity resth is
(19)        generic (RVAL: real := 1.0*KILO);
(20)        port (terminal tp, tm: thermal);
(21)  end entity resth;
(22)
(23)  architecture bce of resth is
(24)        quantity temp across hfl through tp to tm;
(25)  begin
(26)        temp == RVAL*hfl;
(27)  end architecture bce;
(28)
(29)  – Behavioural model of a thermal generator
(30)  entity genetherm is
(31)        generic ( tambient : real := 300.0);
(32)        port (terminal tlp : thermal);
(33)  end;
(34)
(35)  architecture equ of genetherm is
(36)        quantity temp across power through tlp to thermal_ground;
(37)  begin
(38)        temp == tambient ;
(39)  end;
```

Figure 9. VHDL-AMS models of the thermal components.

```
(1)   library ieee_proposed;
(2)     use ieee_proposed.energy_systems.all;
(3)     use ieee_proposed.electrical_systems.all;
(4)     use ieee_proposed.thermal_systems.all;
(5)   entity cmos_inv is
(6)     generic (
(7)       WN: real := 15.0*MICRO; – NMOS channel width
(8)       LN: real := 0.15*MICRO;  – NMOS channel length
(9)       WP: real := 15.0*MICRO; – PMOS channel width
(10)      LP: real := 0.15*MICRO); – PMOS channel length
(11)    port (
(12)      terminal tin, tout, tvdd, tvss: electrical;
(13)      terminal tjn, tjp: thermal);
(14)  end entity cmos_inv;
(15)
(16)  architecture str of cmos_inv is
(17)  begin
(18)    PMOS: entity work.mos(ekv_simple)
(19)      generic map (
(20)        MTYP => -1.0, WEFF => WP, LEFF => LP,
(21)        VT0 => -0.4, TCV => -1.5*MILLI)
(22)      port map (
(23)        td => tout, tg => tin, ts => tvdd, tb => tvdd, tj => tjp);
(24)    NMOS: entity work.mos(ekv_simple)
(25)      generic map (
(26)        MTYP => 1.0, WEFF => WN, LEFF => LN,
(27)        VT0 => 0.4, TCV => 1.5*MILLI)
(28)      port map (
(29)        td => tout, tg => tin, ts => tvss, tb => tvss, tj => tjn);
(30)  end architecture str;
```

Figure 10. VHDL-AMS structural model of the CMOS inverter.

The CMOS inverter is a structural model that instantiates two components: one pMOS transistor called `PMOS` (Figure 10, lines 18 to 23) and one nMOS transistor called `NMOS` (Figure 10, lines 24 to 29). Both the generic parameters and the port associations use the named association mechanism for improved readability.

Figure 11 shows the temperature evolution in the inverter for two different values of `Rcoupling`. As expected, for a small value of `Rcoupling`, the temperature in the N and P transistors are tightly linked (curves 1 and 2). For a higher value, Figure 11 shows the free temperature evolution of each transistor (curves 3 and 4).

For simulation purpose, the values of the capacitances and resistances of the thermal network have voluntary been overstated to emphasize the thermal effects.

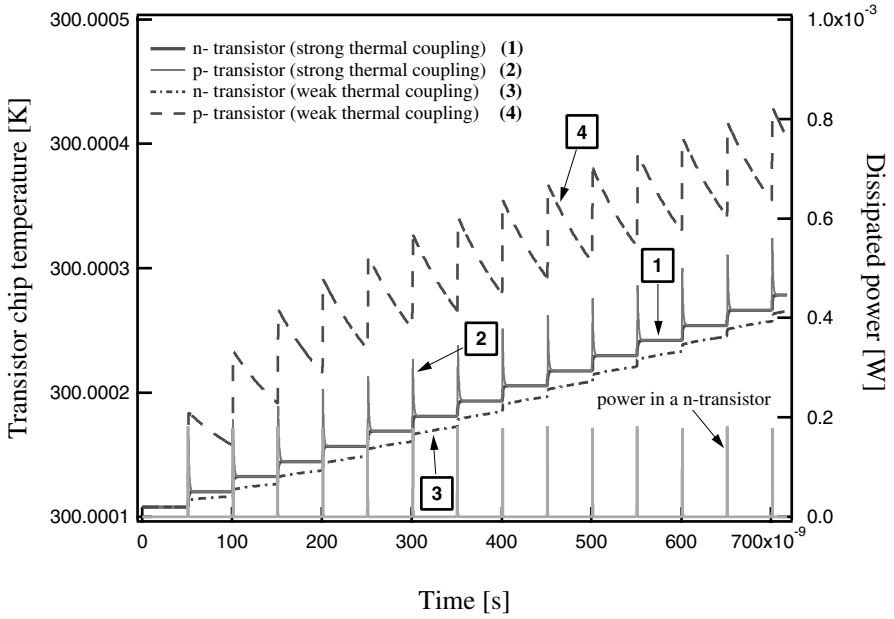

Figure 11. Simulation of the n/p – channel transistor chip temperature variation versus time during commutation (in an inverter), for two values of thermal coupling.

5. Accounting for the Quantum Effects in an Advanced MOSFET Model

As CMOS technology tends towards ever thinner gate oxide and higher substrate doping concentration, the quantum effects are more and more significant. From a physical point of view, they result in a change in the relationship between charges and applied voltages. In previous works, we have shown that in the context of a surface potential model, we only need to compute the exact value (i.e. the quantum value) of the surface potential to get a coherent model [26, 27]. The proposed model fully accounts for the quantum effects and is able to accurately describe all major characteristics of MOSFET. It does not require either definition of an effective oxide thickness or use of additional parameters.

Based on the core of the MM11 model [28], we have developed new concepts to compute the exact value of the surface potential, i.e. accounting for the quantum effects. The model covers all operating regions from accumulation to inversion and is valid for all bias conditions. This section is organized as follows. First, we describe the physical basis of our analytical and quantum surface potential model. The straightforward use of a charge sheet model

(drift-diffusion approximation) is then discussed. Next, we detail the VHDL-AMS implementation of the quantum model for an n-MOS transistor. Finally, the simulation results obtained with the VHDL-AMS model are presented.

5.1. Modeling the Quantum Mechanical Effects

In advanced CMOS technologies, the use of thin gate oxides and high substrate doping levels results in a very high normal field at the Si–SiO_2 interface, so that the energy spectrum consists of a set of discrete energy levels, where the first allowed energy level does no longer coincide with the bottom (top) of the conduction (valence) band. Figure 12 shows the energy band diagram of an n-MOS transistor biased in strong inversion. It appears that the quantum mechanical effects (QME) increase the apparent bandgap of the semiconductor $(\Delta E_g = E_0 - E_c)$. The same reasoning is valid for the accumulation region as well. Within the context of surface-potential-based models, we have demonstrated that the quantum effects can be fully taken into account by the new concept of pseudo bandgap widening. The reader is referred to references [26, 27, 29] for full details of the procedure.

Once the explicit relationship between the quantum increment/decrement of the surface potential and the gate and source/drain voltages is known, incorporating this relationship into the core of the MM11 model (i.e. a classical

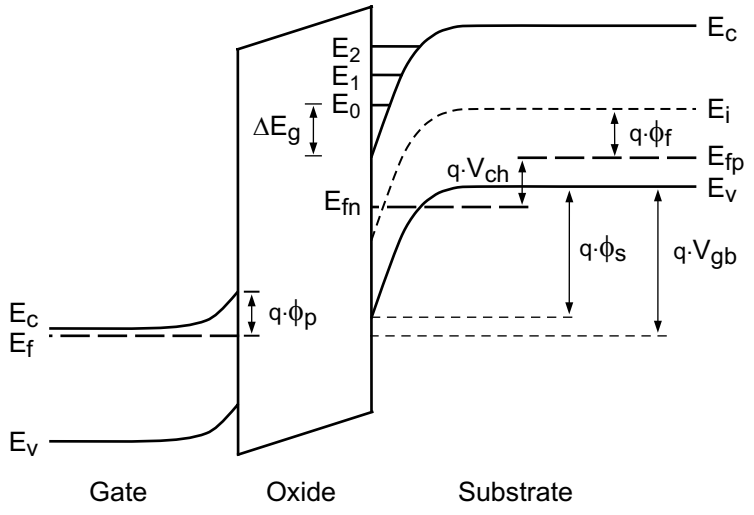

Figure 12. Energy band diagram of an n-channel MOSFET (strong inversion mode). ϕ_s is the surface potential, ϕ_p is the band bending in the gate due to the polydepletion effect, V_{ch} is the channel potential (electron quasi-Fermi potential) and ϕ_f the intrinsic Fermi potential.

description of the surface potential without QME) [28] allows us to obtain an accurate analytical and quantum surface potential model valid from accumulation to strong inversion, and from linear to saturation region [29].

Finally, taking into account the quantum effects does not make the new model more computational demanding and does not introduce any additional parameter with respect to a classical description of the surface potential. Figure 13 shows a comparison between the surface potential computed with the new model and the results obtained by a self-consistent resolution of the Schrödinger and Poisson equations.

A major interest of a surface potential model is that it enables a straightforward use of a charge sheet model since all charges in the latter explicitly depend on the surface potential value [30]. As our model computes the surface potential analytically, the use of the drift-diffusion approximation does not require time consuming iterations to calculate the surface potential (at the source and drain ends). This means that the major advantage of common piece-wise models does no longer hold, and consequently the surface-potential-based MOSFET models are the best candidates to be chosen as new standard compact MOSFET models.

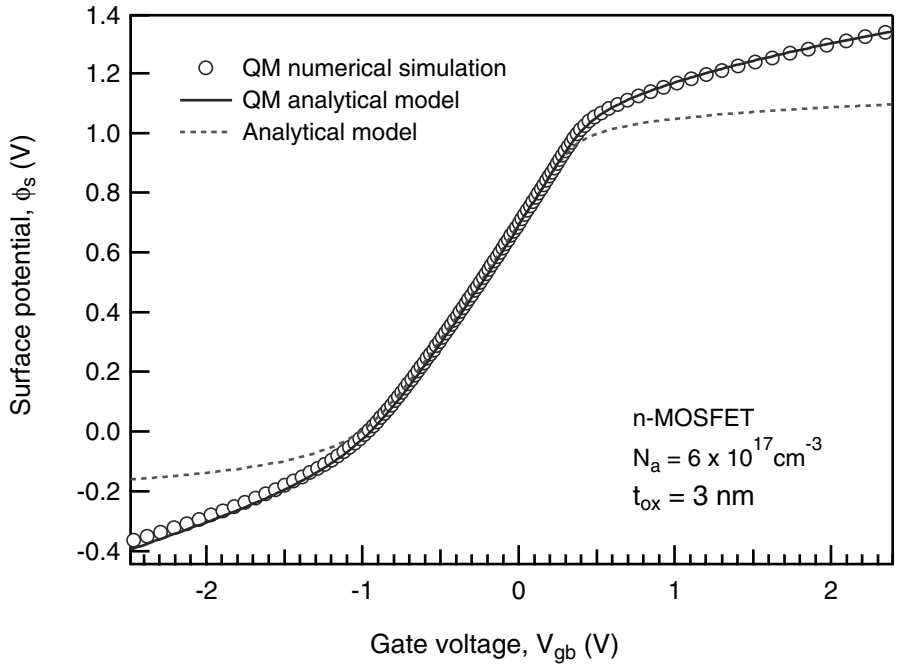

Figure 13. Surface potential analytically computed as a function of gate voltage. The channel potential V_{ch} is set to zero in this simulation.

5.2. VHDL-AMS Implementation

In Figure 14, we present the skeleton of the VHDL-AMS code for the quantum surface potential model (the full code is available on a website [20]). The VHDL-AMS code first contains references libraries needed to parse the model (lines 1–2). For the model end-user (circuit designer), the most important part of the model is the interface, contained in what is called the VHDL-AMS `entity` (lines 33–39). The model interface is decomposed into the specification of generic parameters (lines 34–37) and of the interface ports (line 38). The `generic` statement allows the designer to define its own values for the model parameters. Typically, geometrical W and L transistor parameters are defined as generic. The `mosfet` entity contains four terminals (G, D, S and B stand for the gate, drain, source and bulk terminal respectively), all of electrical type. All the terminals are part of a `port` statement.

```
(1)   library ieee;
(2)     use ieee.electrical_systems.all; use ieee.math_real.all;
(3)
(4)   – Functions declaration
(5)   package mm11_functions is
(6)     pure function phis1_qm_pd(Cox,Vg,…,PDE:real) return real; –acc.
(7)     pure function phis2_qm_pd(Cox,Vg,Vch,…,PDE:real) return real; –inv.
(8)     …/…
(9)   end;
(10)  – Functions definitions
(11)  package body mm11_functions is
(12)  – Quantum description of the surface potential (accumulation)
(13)    pure function phis1_qm_pd(Cox,Vg,…,PDE:real) return real is
(14)      variable ret :real;
(15)    begin
(16)      ret := …; return ret;
(17)    end phis1_qm_pd;
(18)  – Quantum description of the surface potential (inversion)
(19)    pure function phis2_qm_pd(Cox,Vg,Vch,…,PDE:real) return real is
(20)      variable ret :real;
(21)    begin
(22)      ret := f_qm_pd(Cox,Vg,Vch,…,PDE) +
(23)      phit*log(((((2.0/gamma)*(Vg-Vfb-
(24)      psiB(Cox,Vg,Vch,…,PDE)))
(25)      / (1.0+sqrt(1.0+(4.0/gamma_p**2)*
(26)      (Vg-Vfb-psiB(Cox,Vg,Vch,…,PDE)))))**2 -
(27)      f_qm_pd(Cox,Vg,Vch,…,PDE)+phit) / phit);
(28)      return ret;
(29)    end phis2_qm_pd;
(30)      …/…
(31)  end mm11_functions;
(32)
(33)  entity mosfet is
(34)    generic(W :real := 1.0e-6; – Gate width
```

```
(35)        L :real := 1.0e-6; – Gate length
(36)        Na :real := 5.0e23; – Substrate doping level
(37)        …/…);
(38)     port (terminal G,D,S,B :electrical);
(39)  end entity mosfet;
(40)
(41)  architecture quantum_polydep of mosfet is
(42)     constant T :real := 300.0; …/…
(43)     quantity Qg1,Qg2,Qg,Cgg1,Cgg2,Cgg :real;
(44)     quantity Idiff,Idrift :real; …/…
(45)     quantity Ids through D to S;
(46)     quantity Vdb across D to B; quantity Vsb across S to B;
(47)     quantity Vgb across G to B;
(48)  begin
(49)     …/…
(50)  – Gate charge & gate transcapacitance
(51)     Qg1 == W*L*Cox*(Vgb-Vfb-phip-phis1_qm_pd(Cox,Vgb,…,PDE));
(52)     Qg2 == 0.4*W*L*Cox*(Vgb-Vfb-phip-phis2_qm_pd(Cox,Vgb,Vdb,…,PDE))
(53)              +0.6*W*L*Cox*(Vgb-Vfb-phip-phis2_qm_pd(Cox,Vgb,Vsb,…,PDE));
(54)     Cgg1 == Qg1'dot;
(55)     Cgg2 == Qg2'dot;
(56)        if Vgb'above(0.0) use
(57)        Qg == Qg2; Cgg == Cgg2/(W*L*Cox);
(58)        else
(59)        Qg == Qg1; Cgg == Cgg1/(W*L*Cox);
(60)        end use;
(61)  – Drain current
(62)     Idrift == …; Idiff == …; Ids == Idrift + Idiff;
(63)     …/…
(64)  end architecture quantum_polydep;
```

Figure 14. Skeleton of the VHDL-AMS code for the modified MM11 model.

The MOSFET behavior is defined in a separate architecture called `quantum_polydep` (lines 41–64). Lines 43–47 declare *quantities*.

In the lines 45–47, a number of *branch quantities* are declared. A number of so-called *free quantities* (i.e. quantities not bound to any terminal) are also declared (lines 43–44). These quantities are mainly used to break down complex relationships into more manageable and understandable pieces.

The electrical behavior of the surface potential model is actually procedural so it is more efficient to use the sequential statements proposed by VHDL-AMS. However, as the simultaneous `procedural` statement is not yet supported in the Simplorer tool, we use different functions instead. All the needed functions are defined in a package called `mm11_functions` (lines 5–31). This package is divided into two units. The first unit includes the declaration of the functions prototypes (lines 5–9) while the second unit includes the functions bodies (lines 11–31).

For instance, the `phis2_qm_pd` function given in lines 19–29 corresponds to the following mathematical relationship:

$$\phi_{s2\_qm\_pd} = f_{\_qm\_pd} + \phi_t \cdot \ln\left\{\left[\left(\frac{\frac{2}{\gamma}\cdot(V_g - V_{fb} - \psi_B)}{1+\sqrt{1+\frac{4}{\gamma_p^2}\cdot(V_g - V_{fb} - \psi_B)}}\right)^2 - f_{\_qm\_pd} + \phi_t\right]/\phi_t\right\}$$

Next, making use of the functions, we can easily implement useful quantities such as, for example, the gate charge/transcapacitance (lines 51–60) and the drain current (lines 62). In our model, all the free quantities (simultaneous statements) are functions of the surface potential. They just depend on two functions, namely `phis1_qm_pd` and `phis2_qm_pd`. For example, lines 50–60 detail the implementation of the gate transcapacitance. The simultaneous `if` statement has been used to select the `phis1_qm_pd` or `phis2_qm_pd` function which corresponds to the formulation of the surface potential in accumulation or inversion. Note the use of the `'dot` attribute to denote a first-order time derivative of the quantity prefix.

5.3. Simulation Results

The VHDL-AMS model previously discussed has been implemented using Simplorer 6.0 from Ansoft. Four different architectures for the `mosfet` entity have been implemented. They allow the user to choose between classical or quantum surface potential models and give the possibility to take into account or not the polydepletion effect (PDE). In fact, only the architecture called `quantum_polydep` (see Figure 14) is useful since it describes an n-MOS transistor using the full model (QME+PDE). The three others have been written for comparison purpose.

We have tested the different n-MOSFET architectures by applying a 1V/s ramp on the gate terminal in a transient simulation. Both source and bulk terminals of the device are connected to the ground and the drain-to-bulk voltages are set to 0.1V. Figure 15 shows the simulated gate transcapacitance ($C_{gg} = dQ_g/dV_{gb}$) for different architectures of the `mosfet` entity.

Since the derivative over time of the gate-to-bulk voltage equals one, the quantity `Qg'dot` (lines 54–55 in Figure 14) is simply equal to the gate transcapacitance ($dQ_g/dt = C_{gg} \times dV_{gb}/dt = C_{gg}$).

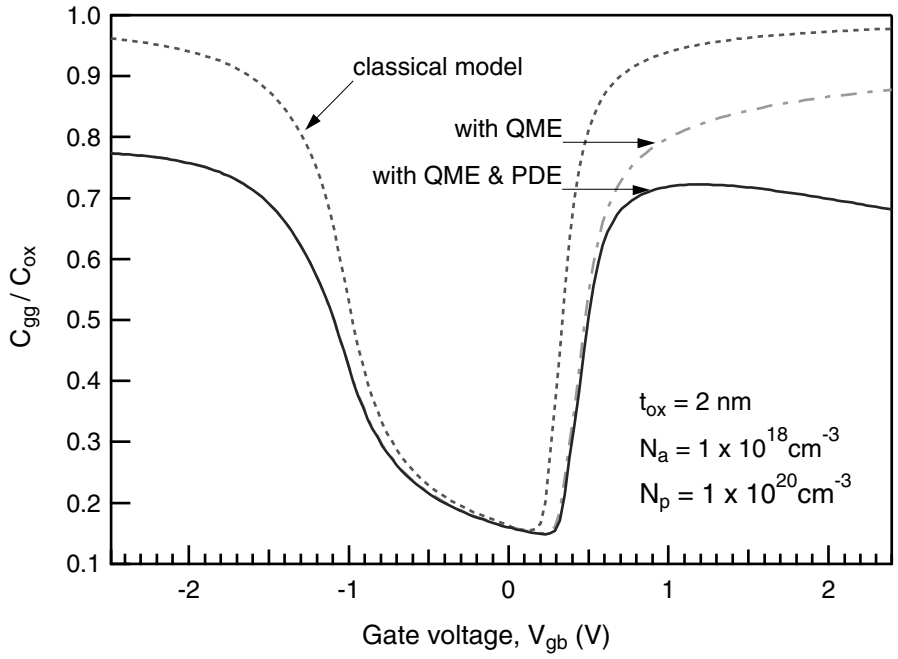

Figure 15. Normalized gate transcapacitance as a function of gate voltage for different architectures (classical, quantum = QME; quantum with polydepletion effect = QME + PDE).

Figure 16 gives the *I–V* simulation extracted from the same transient simulation. The drain current I_{ds} is obtained as the sum of the drift and diffusion currents and exhibits an excellent behavior from weak to strong inversion.

6. Conclusions

The presented case studies show that VHDL-AMS can be successfully used to implement low-level models, such as EKV 2.6 and MM11 models of MOSFET devices. The main point is that the physical equations of the models can actually be written "as is" in the model source code. The only limitations in this straightforward translation do not come from the language itself, but from the available simulation tools, Advance-MS from Mentor Graphics, and Simplorer from Ansoft. The lack of support in these commercial tools for a procedural statement forces us to use functions with numerous parameters to avoid a whole set of simultaneous statements.

Taking the various domains involved in the modeling task is not complicated either. We have shown that the basic models can be easily enhanced to include major physical effects like self-heating, extrinsic aspects and quantum

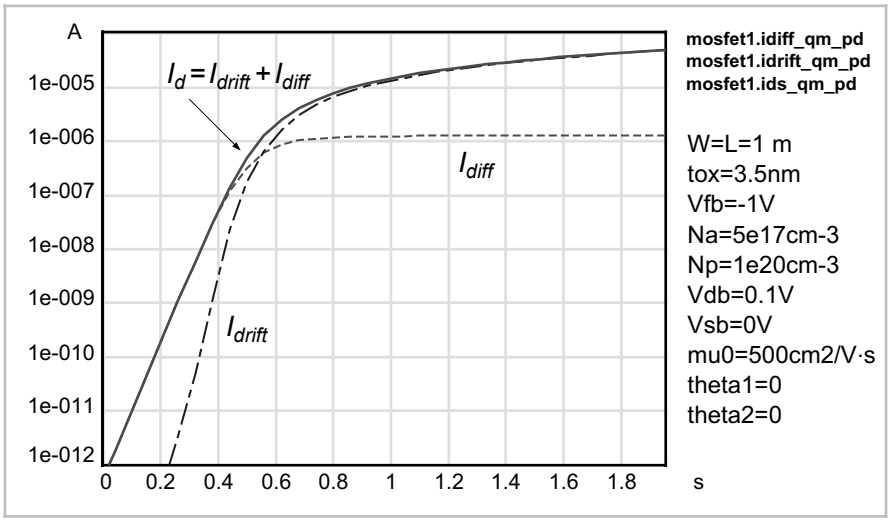

Figure 16. Screen dump of a VHDL-AMS simulation showing both drift and diffusion components of the drain current for an n-channel MOSFET.

effects, as the VHDL-AMS language naturally supports multi-domain (thermal, mechanical, fluidic, etc) modeling.

In a near future, we plan to use these compact multi-domain models to design analog circuits like an operational amplifier, and digital subsystems like a nand gate and flip-flop, that take into account thermo-electrical interactions. We also plan to use the bond-graph modeling approach experienced in [31] to propose accurate models of isFET (ion sensitive FET), that could eventually lead to the development of biosensor simulation models. Some models of advanced devices such as DG-MOSFET are also under development [32].

Acknowledgements

The authors would like to thank Dr. Wladek Grabinski for his strong implication in MOS-AK activities (see MOS-AK web site: http://www.mos-ak.org/).

References

[1] 1076.1-1999 IEEE Standard VHDL Analog and Mixed-Signal Extensions Language Reference Manual, *IEEE Press, ISBN 0-7381-1640-8*, **1999**.

[2] 1076.1.1-2004 IEEE Standard VHDL Analog and Mixed-Signal Extensions – Packages for Multiple Energy Domain Support, *IEEE Press, ISBN 0-7381-4646-5*, **2005**.

[3] Ashenden, P.; Peterson, G.D.; Teegarden, D.A. "The System Designer's Guide to VHDL-AMS", Morgan Kaufman Publishers, **2002**, *ISBN 1-55860-749-8*.
[4] Christen, E.; Bakalar, K. "VHDL-AMS-a hardware description language for analog and mixed-signal applications", *IEEE Trans. Circ. & Syst. - Part II*, **1999**, *46(10)*, 1263–1272.
[5] Frey, P.; O'Riordan, D. "Verilog-AMS: Mixed-signal simulation and cross domain connect modules," *Proc. of the IEEE/ACM Int. Workshop Behav. Modeling Simul. (BMAS)*, **2000**, 103–108.
[6] http://www.eda.org/verilog-ams/
[7] Pêcheux, F.; Lallement, C.; Vachoux, A. "VHDL-AMS and Verilog-AMS as Alternative Hardware Description Languages for Efficient Modeling of Multi-Discipline Systems", *IEEE Trans. Comput.–Aided Design Integrated Circ. and Syst.*, **2005**, *24*, 204–225.
[8] BSIM 4.2 Manual. http://www-device.eecs.berkeley.edu/∼bsim (**April 2001**).
[9] The MM9 model. http://www.semiconductors.philips.com/philips_models/mos models (**2001**).
[10] Bucher, M.; Lallement, C.; Enz, C.C.; Théodoloz, F.; Krummenacher, F. "The EPFL-EKV MOSFET Model, Version 2.6", *Technical Report*, **1999**, [Online] http://legwww.epfl.ch/ekv/
[11] Foty, D. "MOSFET modelling with Spice". *Principles and Practice*, Englewood Cliffs, NJ: Prentice-Hall, *ISBN: 0-13-227935-5*, **1997**.
[12] Tsividis, Y.P. "Operation and Modelling of the MOS Transistor", second edition, New York: McGraw-Hill Book Company, *ISBN: 0070655235*, **1999**.
[13] Bucher, M.; Enz, C.C.; Krummenacher, F.; Sallese, J.-M.; Lallement, C.; Porret, A.-S. "The EKV 3.0 compact MOS transistor model: accounting for deep-submicron aspects", *Proc. MSM Int. Conf., Nanotech 2002*, **2002**, 670–673.
[14] Miura-Mattausch, M.; Mattausch, H.J.; Arora, N.D.; Yang, C.Y. "MOSFET modelling gets physical", *IEEE Circ. and Dev.*, **2001**, *17(6)*, 29–36.
[15] van Langevelde, R.; Scholten, A.J.; Havens, R.J. "Advanced compact MOS modelling", *Proc. ESSDERC*, **2001**, 81–90.
[16] Chen, T.L.; Gildenblat, G. "Analytical approximation for the MOSFET surface potential", *Solid-State Electron.*, **2001**, *45*, 335–339.
[17] Gildenblat, G.; Li, X.; Wang, H.; Lu, W.; van Langevelde, R.; Scholten, A.J.; Smit, G.D.J.; Klaassen, D.B.M. "Introduction to PSP MOSFET Model", *Proc. the MSM 2005 Int. Conf., Nanotech 2005*, **2005**.
[18] Lallement, C.; Pêcheux, F.; Hervé, Y. "VHDL-AMS Design of a MOST Model Including Deep Submicron and Thermal-Electronic Effects", *Proc. IEEE/ACM Int. Workshop Behav. Modeling Simul. (BMAS)*, **2001**, *91–96*.
[19] Lallement, C.; Pêcheux, F.; Hervé, Y. "A VHDL-AMS Case study: The incremental Design of an Efficient 3rd generation MOS Model of Deep Sub Micron Transistor", In *SOC Design Methodologies*, M. Robert, B. Rouzeyre, C. Piguet and M.-L. Flottes, Eds., Boston, Hardbound, Kluwer Academic Publishers, ISBN 1-4020-7148-5, **July 2002**, 349–360.
[20] http://lsmwww.epfl.ch/models/compact/
[21] Arora, N. "Mosfet Models for VLSI Circuit Simulation: Theory and Practice", *Computational Microelectronics*, Wien, New York: Springer Verlag, **1993**.
[22] Prégaldiny, F.; Lallement, C.; Mathiot, D. "A simple efficient model of parasitic capacitances of deep-submicron LDD MOSFETs", *Solid-State Electron.*, **2002**, *46*, 2191–2198.
[23] Lallement, C.; Bouchakour, R.; Maurel, T. "One-dimensional Analytical Modeling of the VDMOS Transistor Taking Into Account the Thermoelectrical Interactions", *IEEE Trans. Circ. & Sys., Part. I*, **1997**, *44(2)*, 103–111.

[24] Lallement, C.; Pêcheux, F.; Grabinski, W. "High Level Description of Thermodynamical effects in the EKV 2.6 MOST Model", *9th Int. Conf. Mixed Design of Integrated Circ. & Sys. (MIXDES 2002)*, Wroclaw/Poland, **June 2002**, 45–50.

[25] http://lsmwww.epfl.ch/tcad_paper/.

[26] Prégaldiny, F.; Lallement, C.; van Langevelde, R.; Mathiot, D. "An advanced explicit surface potential model physically accounting for the quantization effects in deep-submicron MOSFETs", *Solid-State Electron.*, **2004**, *48(3)*, 427–435.

[27] Prégaldiny, F.; Lallement, C.; Mathiot, D. "Accounting for quantum mechanical effects from accumulation to inversion, in a fully analytical surface-potential-based MOSFET model", *Solid-State Electron.*, **2004**, *48(5)*, 781–787.

[28] van Langevelde, R.; Klaassen, F.M. "An explicit surface-potential-based MOSFET model for circuit simulation", *Solid-State Electronics*, **2000**, *44*, 409–418.

[29] Prégaldiny, F.; Lallement, C. "Fourth generation MOSFET model and its VHDL-AMS implementation", *Int. J. Numerical Modelling: Electronic Networks, Dev. and Fields*, **2005**, *18*, 39–48.

[30] Brews, J.R. "A charge sheet model of the MOSFET", *Solid-State Electron.*, **1978**, *21*, 345–355.

[31] Pêcheux, F.; Allard, B.; Lallement, C.; Vachoux, A.; Morel, H. "Modeling and simulation using bond graphs and VHDL-AMS", *Int. Conf. Bond Graph Modeling & Simul. (ICBM'2005)*, New Orleans - USA, **2005**, 149–155.

[32] Prégaldiny, F.; Krummenacher, F.; Diagne, B.; Pêcheux, F.; Sallese, J.-M.; Lallement, C. "Explicit modeling of the double-gate MOSFET with VHDL-AMS", *submitted to Int. J. Numerical Modelling: Electronic Networks, Dev. and Fields*, **July 2005**.