Adaptive EPFL-EKV Long and Short Channel MOS Device Models for Qucs, SPICE and Modelica Circuit Simulation

Mike E. Brinson, and Hassan Nabijou

***Abstract*—Equation-defined non-linear functional elements are important building blocks in the development of compact semiconductor device models. Current trends in compact device modelling suggest widespread acceptance among the modelling community of Verilog-A, for semiconductor device specification, model exchange and circuit simulation. This paper outlines techniques for the development of adaptive EPFL-EKV long and short channel MOS models which stress user selectable model features and diagnostic capabilities. Adaptive EPFL-EKV nMOS models based on Verilog-A and Modelica are introduced and their performance compared with simulation data obtained using the "Quite universal circuit simulator" (Qucs), SPICE and the Modelica simulation environment.**

***Index Terms*—Adaptive MOS models, Qucs, SPICE, Modelica, equation-defined device modelling, Verilog-A, EPFL-EKV MOSFET model, parameter and equation monitoring.**

I. Introduction

COMPACT semiconductor model development has in recent years advanced significantly through the use of equation defined non-linear functional elements [1] and the adoption of Verilog-A as the hardware description language of preference for model construction and model interchange between different circuit simulators [2]. The standardization of Verilog-A [3] has also accelerated its acceptance as the modelling tool of choice among the compact device modelling community. In many respects Verilog-A is a hardware description language which has arrived on the modelling scene at the right time, bringing to the art of compact modelling an array of positive features which greatly aid in the construction of complex models. Such models appear to be characterised by an ever increasing number of parameters, particularly when compared to the number associated with the SPICE legacy devices [4]. These parameters, plus built-in model equations, allow accurate modelling of I-V, dynamic and noise characteristics for devices with sub-micron feature sizes. However, model functionality is often achieved at the expense of model complexity. Previous and current generations of circuit simulator normally allow users to select semiconductor device model types through the use of a LEVEL parameter. Similarly, in some instances, model complexity can be controlled by setting one or more second order physical parameters to zero [5]. Unfortunately, both these approaches do not give a clear indication as to the calculation overhead incurred by a given model, or a restricted subset model, making it difficult to estimate circuit simulation run times for a specific model hierarchical level. This paper outlines techniques for the development of adaptive long and short channel nMOS models which stress user selectable model features and promote diagnostic features through the addition of signal probes to model interfaces. Adaptive EPFL-EKV nMOS models based on Verilog-A and Modelica are introduced and their performance compared using simulation data obtained with Qucs [6] and the Modelica simulation environment [7], [8].

II. The Basic Long Channel EPFL-EKV MOS Model

Qucs equation-defined device (EDD) models and Verilog-A code models for long channel EPFL-EKV 2.6 nMOS transistors [9] can be found in recently published literature [10]. A set of typical long channel nMOS I-V characteristics obtained by circuit simulation are presented in Fig. 1. As expected, both the Qucs EDD and the Verilog-A model simulation output data give identical results. However, one important difference between the two models is observed from the IV simulation tests, namely the model simulation speed. The EDD model, being an interpretive model, tends to be slower than the Verilog-A model after its code has been translated to the C/C++ language, compiled and linked to the main body of a circuit simulators C/C++ code. Qucs is not the only simulator which allows compact semiconductor

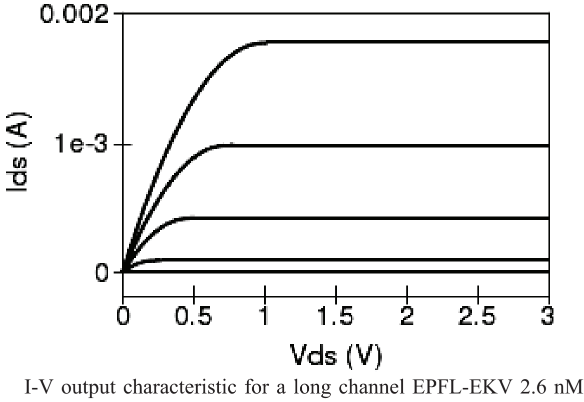


Fig. 1. I-V output characteristic for a long channel EPFL-EKV 2.6 nMOS. model with: L=0.5e-6m, W=10e-6m, VTO=0.6V, GAMMA=0.7√V, PHI=0.97V, KP=150e-6A/V^2, THETA = 50e-3V^{-1} and TEMP = 26.86 Celsius.

M.E. Brinson (email: mbrin72013@yahoo.co.uk) and H. Nabijou (email: h.nabijou@londonmet.ac.uk) are both with the Centre for Communications Technology Research, London Metropolitan University, London, UK.

device models to be constructed with equation-defined components. Shown in Fig. 2 is the SPICE code for an EPFL-EKV long channel nMOS transistor based on the SPICE extensions implemented in LTspice IV® [11]. Unfortunately, at this time, there appears to be little standardisation of the format for SPICE 3 extensions among the popular freely available General Public Licence (GPL) [12], [13] and the freely available commercial versions of SPICE, implying that the code shown in Fig. 2 would probably need some modification when run on different SPICE simulators. It is also possible to develop a version of the EPFL-EKV 2.6 long channel nMOS transistor model using the Modelica simulation language. Fig. 3 presents the Modelica code for such a model. Like Verilog-A most implementations of the Modelica modelling environment are compiled, rather than interpretive, which results in fast circuit simulation speeds. Note how similar in many respects the Modelica code is to Verilog-A hardware description language model definition.

III. Adding Charge Storage Effects to the Long Channel EPFL-EKV 2.6 MOS Model

To be able to simulate the dynamic performance of MOS transistors a model for the stored charge or device capacitance must be added to a DC MOS model. In the case of the EPFL-EKV 2.6 MOS model the stored charge is represented by equations (69) to (78) listed in publication [9]. Fig. 4 presents the Verilog-A code for a long channel nMOS transistor with

```
*
.subckt EKV26nMOSLC 1 2 3 4  L=0.5e-6 W=10e-6 VTO=0.6 GAMMA=0.7
+                          PHI=0.97 KP=150e-6 THETA=50e-3
+                          TEMP=26.85
* Node order -->   gate drain source bulk
* Initialisation code
.param P1={-VTO+PHI+GAMMA*sqrt(PHI)},  P2={GAMMA/2}, P3={P2*P2}
.param PK=1.3806503e-23, PQ=1.602176472e-19, T2={TEMP+273.15}
.param VTT2={PK*T2/PQ}, P5={PHI+4*VTT2+1e-12}, P6={KP*W/L}
.param P7={1/(2*VTT2)}, P8={2*VTT2*VTT2}
*End initialisation code
*Current contributions representing model equations
Bvgprime 0 5 I=V(1,4)+P1
Rvgprime 5 0 1
Bvp 0 6 I=if(V(5) >= 0.0, V(5)-PHI-GAMMA*(sqrt(V(5)+P3)-P2), -PHI)
Rvp 6 0 1
Bn  0 7  I=1+P2/sqrt(V(6)+P5)
Rn  7 0 1
Bbeta  0 8 I=P6/(1+THETA*V(6))
Rbeta  8 0 1
BIff1 0 9 I=ln(1+exp((V(6)-V(3,4))*P7))
RIff1 9 0 1
BIr   0 10 I=ln(1+exp((V(6)-V(2,4))*P7))
RIr 10 0 1
BIds 2 3 I=P8*V(7)*V(8)*(V(9)*V(9)-V(10)*V(10))
* End current contributions
.ends
* nMOS long channel I-V characteristics test circuit
Vgs 1 0 dc 1
Vds 99 0 dc 1
Vm  99 2 dc 0
X1 1 2 0 0 EKV26nMOSLC  L=0.5e-6 W=10e-6 VTO=0.6 GAMMA=0.7
+                    PHI=0.97 KP=150e-6 THETA=50e-3
+                    TEMP=26.85
.op
.dc  Vds 5 0 -0.05 Vgs 0 3 0.5
.end
```

Fig. 2. LTspice IV equation-defined non-linear subcircuit and I-V test circuit for a long channel EPFL-EKV 2.6 nMOS model: parameter values are the same as those listed in Fig. 1.

```
model EKV26nMOSLC
  Modelica.Electrical.Analog.Interfaces.Pin D;
  Modelica.Electrical.Analog.Interfaces.Pin G;
  Modelica.Electrical.Analog.Interfaces.Pin S;
  Modelica.Electrical.Analog.Interfaces.Pin B;
// Initialisation code
  parameter Real L=5e-07; parameter Real W=1e-05;
  parameter Real VTO=0.6; parameter Real PHI=0.97;
  parameter Real GAMMA=0.71; parameter Real KP=150e-6;
  parameter Real THETA=0.05; parameter Real TEMP=26.85;
  constant Real PQ=1.602176462e-19; constant Real PK=1.3806503e-23;
  parameter Real P1=-VTO+PHI+GAMMA*sqrt(PHI);
  parameter Real P2=GAMMA/2; parameter Real P3=P2*P2;
  parameter Real T2=TEMP+273.15; parameter Real VTT2=(PK*T2)/PQ ;
  parameter Real P5=PHI+4*VTT2; parameter Real P6=(KP*W)/L;
  parameter Real P7=1/(2*VTT2); parameter Real P8=2*VTT2*VTT2;

 Real vgprime,vp, n, beta, iff1, iff, ir1, ir, Id;
// Model equations and current contributions
equation
 vgprime = G.v + P1;
 vp=if gprime>0.0 then vgprime-PHI-GAMMA*(sqrt(vgprime+P3)-P2) else -PHI;
 n = 1 + P2 / sqrt(vp + P5);
 iff1 = log(1 + exp((vp - S.v) * P7));  iff = iff1 * iff1;
 ir1 = log(1 + exp((vp-D.v) * P7));     ir = ir1 * ir1;
 beta= P6/(1+THETA*vp); Id = P8 * n * beta * (iff - ir);
 G.i = 0; D.i = Id; S.i = -Id; B.i = 0;
end EKV26nMOSLC;
```

Fig. 3. Modelica equation-defined non-linear model for a long channel EPFL-EKV 2.6 nMOS model: with identical parameter values to the ones listed in Fig. 1.

```
`include "disciplines.vams"
`include "constants.vams"
module EKV26nMOSSC1(drain,gate,source,bulk);
inout drain,gate,source,bulk; electrical drain,gate,source,bulk;
parameter real L=0.5e-6 from [1e-20:inf];
parameter real W=10e-6 from [1e-20:inf];
parameter real VTO=0.6 from [1e-20:2.0];
parameter real GAMMA=0.71 from [0:2.0];
parameter real PHI=0.97 from [0.3:2.0];
parameter real KP=150e-6 from [1e-20:inf];
parameter real THETA=50e-3 from [0:inf];
parameter real DW=-0.02e-6 from [-5e-6:0];
parameter real DL=-0.05e-6 from [-5e-6:0];
parameter real COX=3.45e-3 from [1e-20:inf];
parameter real TEMP=26.85 from [-100: 100];
parameter integer CHARGESWITCH=0 from [0:1];
parameter real Xpart=0.4 from [0:1];
real P1,P2,P3,P5,P6,P7,P8,P9, P10; real vgprime,vp,n,beta,iff1,ir1,iff,ir, weff, leff;
real nq,P21,xf,xr,qi,qb,qg,coxide,Qi,Qb,Qg,Spart;
branch (gate,bulk) Bgb; branch (source,bulk) Bsb; branch (drain,bulk) Bdb;
branch (drain,source) Bds; branch (gate, drain) Bgd; branch (gate,source) Bgs;
analog begin
@(initial_model) begin
 P1=-VTO+PHI+GAMMA*sqrt(PHI); P2=GAMMA/2; P3=P2*P2;  P10=4*$vt;
 P5=PHI+4*$vt+1e-12; P6=KP*W/L; P7=1/(2*$vt); P8=2*$vt*$vt;
 weff=W+DW; leff=L+DL; P21=PHI+1e-6;
 Spart=1-Xpart; coxide=COX*weff*leff;
end
vgprime=V(Bgb)+P1;
if (vgprime>0.0) vp=vgprime-PHI-GAMMA*(sqrt(vgprime+P3)-P2); else vp=-PHI;
n=1+P2/sqrt(vp+P5);
iff1=ln(1+limexp((vp-V(Bsb))*P7)); iff=iff1*iff1;
ir1=ln(1+limexp((vp-V(Bdb))*P7));  ir=ir1*ir1;
// Charge equations
if ((CHARGESWITCH==1)  begin
 nq=1+GAMMA/(2*sqrt(vp+P21)); xf=sqrt(0.25+iff); xr=sqrt(0.25+ir);
 qi=-nq*(1.333333*(xf*xf+xf*xr+xr*xr)/(xf+xr+1e-20)-1);
 if (vgprime > 0) qb=(-GAMMA*sqrt(vp+P21))/$vt-((nq-1)/nq)*qi;
 else qb=-vgprime/$vt;
 qg=-qi-qb;
end
if (CHARGESWITCH==1) begin
 Qi=coxide*$vt*qi; Qb=coxide*$vt*qb; Qg=coxide*$vt*qg;
 I(Bgd) <+ Spart*ddt(Qg); I(Bgs) <+ Xpart*ddt(Qg);
I(Bdb) <+ Spart*ddt(Qb);  I(Bsb) <+ Xpart*ddt(Qb);
end;
vpdash=0.5*(vp+sqrt(vp*vp+P8)); beta=P6/(1+THETA*vpdash);
I(Bds) <+ P8*n*beta*(iff-ir);
end
endmodule
```

Fig. 4. Verilog-A non-linear long channel EPFL-EKV 2.6 nMOS DC model plus device capacitance: default device parameters are listed in the initial section of the model.

additional charge equation code. Device control parameter CHARGESWITCH is set to one if the charge calculation code is to be included in a simulation, otherwise it is set to zero and the charge code is ignored. The use of this type of switch allows a model to be adapted to fit the requirements of a given simulation, for example at DC or very low AC frequencies the stored charge calculations can be removed from the model without loss of accuracy. The approach adopted in this paper ensures that switched out sections of Verilog-A code are not evaluated during simulation and hence improves the overall simulation speed of a model. It is also worth noting that the proposed code selection technique is ideal in the sense that it only adds a simulation speed penalty equivalent to the run time associated with a C/C++ “if-then-else” statement and a simple variable assignment statement. The test circuit and simulation results shown in Fig. 5 indicate firstly how capacitance values can be extracted from a Qucs model, either the equation defined device or Verilog-A forms, using small signal S parameter simulation techniques, and secondly how Qucs visualisation procedures can be used to plot intrinsic capacitance *Cg = Cgs + Cgd + Cgb* as a function of voltage *Vgs.* Capacitor *Cgb* contributes to the gate capacitance *Cg* in the accumulation region of device operation. In contrast, in the inversion region, *Cgb* decreases to zero, or strictly speaking to the overlap/fringing capacitance along the channel edges. The long channel EPFL-EKV 2.6 nMOS model listed in Fig.4. does not include overlap/fringing capacitance. If required, these can be added as additional external fixed capacitors. The same basic procedure can be adopted when developing a Modelica version of the EPFL-EKV 2.6 model. Presented in Fig.6 is example Modelica code for a long channel EPFL-EKV 2.6 nMOS transistor model. In contrast to the Verilog-A model the Modelica model is implemented with code statements representing device capacitance rather than device charge: equations 79 to 85 cited in publication [9]. Note that in the Modelica model capacitance *Cgb* is not calculated. Although, the code for the two different hardware description languages looks similar a number of points are worth commenting on. Firstly, in the Verilog-A code the = and < + operators represent different forms of assignment statement. However, in the Modelica code the = operator is equivalent to the “equals“ found in the notation for a mathematical equation. Secondly, again in the case of Modelica, the number of equations and the number of variables must balance for the code to simulate without error. This implies that there must only be one equation for each variable in a model. In the Modelica code given in Fig. 6 Verilog-A control variable CHARGESWITCH is replaced by Modelica control variable CAPSWITCH and the Verilog-A time differential operator *ddt* by the equivalent Modelica operator *der*. Presented in Fig.7 is an example plot of the nMOS transistors capacitance obtained by Modelica simulation.

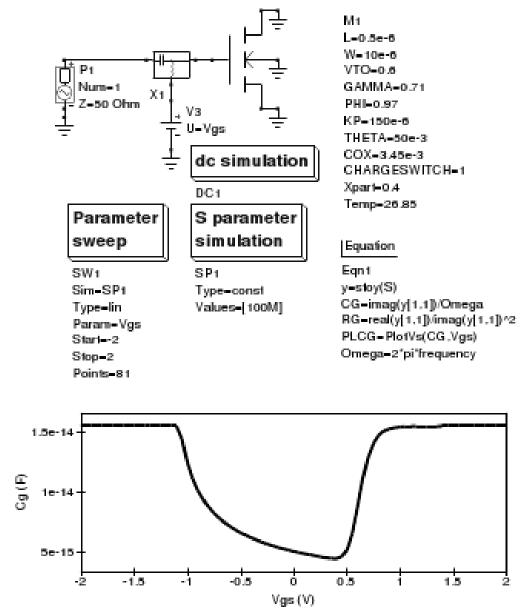


Fig. 5. Gate capacitance extraction test circuit with device drain, source and bulk terminals connected to ground and waveform Cg = Cgs+Cgd+Cgb (F) plotted against Vgs (V) at a test frequency of 100MHz and -2v <= Vgs <= 2V.

```
model EKV26nMOSCap
Modelica.Electrical.Analog.Interfaces.Pin D;
Modelica.Electrical.Analog.Interfaces.Pin G;
Modelica.Electrical.Analog.Interfaces.Pin S; Modelica.
Electrical.Analog.Interfaces.Pin B;
parameter Real L = 5e-07; parameter Real W = 1e-05;
parameter Real VTO = 0.6; parameter Real PHI = 0.97;
parameter Real GAMMA = 0.71;
parameter Real KP = 0.00015;
parameter Real THETA = 0.05; parameter Real DW = -2e-08;
parameter Real DL = -5e-08; parameter Real COX = 0.00345;
parameter Real TEMP = 26.85; parameter Real Xpart = 0.4;
parameter Integer CAPSWITCH = 0;
constant Real PQ = 1.602176462e-19; constant Real PK = 1.3806503e-23;
parameter Real P1 = -VTO + PHI + GAMMA * sqrt(PHI);
parameter Real P2 = GAMMA / 2; parameter Real P3 = P2 * P2;
parameter Real T2 = TEMP + 273.15; parameter Real VTT2 = (PK * T2) / PQ;
parameter Real P5 = PHI + 4 * VTT2; parameter Real P6 = (KP * W) / L;
parameter Real P7 = 1 / (2 * VTT2); parameter Real P8 = 2 * VTT2 * VTT2;
parameter Real P21 = PHI + 1e-06; parameter Real weff = W+DW;
parameter Real leff = L+DL; parameter Real P31 = COX*weff*leff;
parameter Real Spart = 1.0 - Xpart;
Real vgprime,vp,n,beta,iff1,iff,ir1,ir,Id, nq,xf,xr,qi,qg,qb,cgs,cgd,cdb,Px,csb,vpdash;
equation
vgprime = G.v + P1;
vp = if vgprime > 0.0 then
        vgprime - PHI - GAMMA * (sqrt(vgprime + P3) – P2)
      else -PHI;
n = 1 + P2 / sqrt(vp + P5); iff1 = log(1 + exp((vp - S.v) * P7));  iff = iff1 * iff1;
ir1 = log(1 + exp((vp - D.v) * P7)); ir = ir1 * ir1;
if CAPSWITCH == 1 then
 nq = 1 + GAMMA / (2 * sqrt(vp + P21)); xf = sqrt(0.25 + iff);  xr = sqrt(0.25 + ir);
 qi = nq * ((1.333333 * (xf * xf + xf * xr + xr * xr)) / (xr + xf) - 1.0);
if   vgprime > 0.0 then qb = (-GAMMA * sqrt(vp + P21)) / VTT2 - (nq - 1) / nq * qi;
else qb = -vgprime / VTT2;
end if;
qg = -qi - qb;
else
 nq = 0.0; xf = 0.0; xr = 0.0; qi = 0.0; qb = 0.0; qg = 0.0;
end if;
if CAPSWITCH == 1 then
Px = (xf + xr) * (xf + xr) + 1e-20;  cgs = P31 * 0.666666 * (1 - (xr * xr + xr + 0.5 * xf) / Px);
cgd = P31 * 0.666666 * (1 - (xf * xf + xf + 0.5 * xr) / Px);
csb = (nq - 1) * cgs; cdb = (nq - 1) * cgd;
else
 Px = 0.0; cgs = 0.0; cgd = 0.0; csb = 0.0; cdb = 0.0;
end if;
vpdash = 0.5 * vp + sqrt(vp * vp + P8); beta = P6 / (1 + THETA * vpdash);
Id = P8 * n * beta * (iff – ir); G.i = cgs * der(G.v - S.v) + cgd * der(G.v - D.v);
D.i = Id - cgd * der(G.v - D.v) + cdb * der(D.v - B.v);
S.i = -Id - cgs * der(G.v - S.v) + csb * der(S.v - B.v);
B.i = 0.0 - cdb * der(D.v - B.v) - csb * der(S.v - B.v);
end EKV26nMOSCap;
```

Fig. 6. Modelica non-linear long channel EPFL-EKV 2.6 nMOS DC model plus device capacitance: default device parameters are listed in the initialisation section of the model.

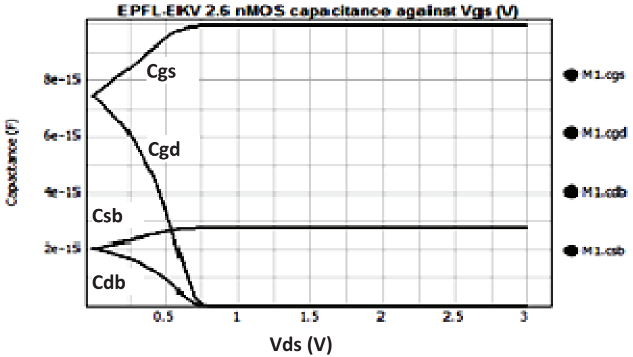


Fig. 7. Individual values for Modelica simulated nMOS capacitances Cgs, Cgd, Cdb and Csb plotted as a function of Vds over the range 0 <= Vds <= 3V: default device parameters as listed in the initialisation section of the model given in Fig.6.

IV. Versions of the EPFL-EKV 2.6 MOS Model to Include Short Channel Effects

Verilog-A and Modelica short channel versions of the long channel EPFL-EKV 2.6 MOS model previously described are introduced in this section. The proposed short channel model includes the four switches listed in Table I. The purpose of these switches is to select the short channel effects that are to be included in the model during simulation: when the switches are set to one the corresponding short channel effect is active. The reverse is true when the switches are set to zero. Using the Verilog-A and Modelica code presented in Fig.4 and Fig.6 as a template allows short channel MOS models to be easily constructed from the equations listed in the reference cited in Table I. The set of nMOS transistor output curves drawn in Figs. 8 to 11 gives an indication of the effects that adding short channel features to the EPFL-EKV 2.6 MOS model have on nMOS device characteristics. The data shown in Figs. 8 to 11 applies to both the Verilog-A and Modelica short channel EPFL-EKV 2.6 models.

TABLE I
Control Switches for Selecting Short Channel Effects

Switch	Short channel effect	Equations (in publication [9])
CLMSWITCH	Channel length modulation	58 to 62
TFMRSWITCH	Transconductance factor and mobility reduction due to vertical field	46, 48 to 55
CHSHSWITCH	Charge sharing for short narrow channels	34 to 38
RSCESWITCH	Reverse short channel effect	30 to 32

V. Adding Monitoring Probes to Verilog-A Compact Semiconductor Device Models

One of the most striking differences between Verilog-A and current implementations of the Modelica hardware description language is the ability of the latter to automatically record, during simulation, values for all the variables that contribute

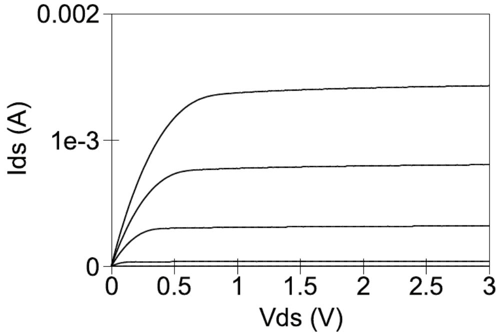


Fig. 8. Typical nMOS transistor output characteristics showing the effects caused by setting short channel control switches: CLMSWITCH=1, TFRMRSWITCH=0, CHSHSWITCH=0, RSCESWITCH=0 and CHARGESWITCH or CAPSWITCH=0.

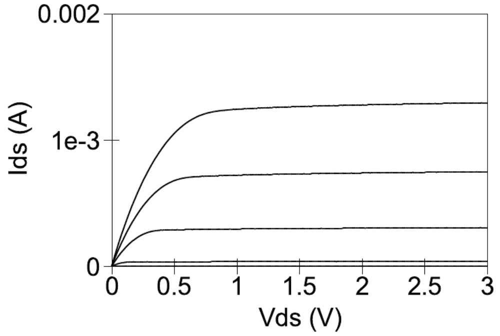


Fig. 9. Typical nMOS transistor output characteristics showing the effects caused by setting short channel control switches: CLMSWITCH=1, TFRMRSWITCH=1, CHSHSWITCH=0, RSCESWITCH=0 and CHARGESWITCH or CAPSWITCH=0.

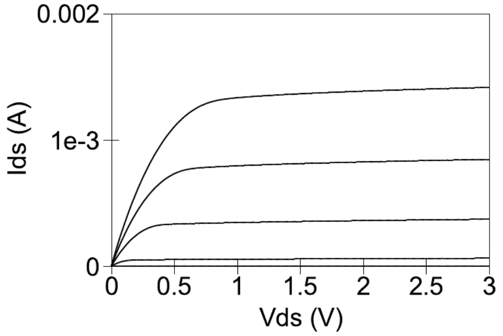


Fig. 10. Typical nMOS transistor output characteristics showing the effects caused by setting short channel control switches: CLMSWITCH=1, TFRMRSWITCH=1, CHSHSWITCH=1, RSCESWITCH=0 and CHARGESWITCH or CAPSWITCH=0.

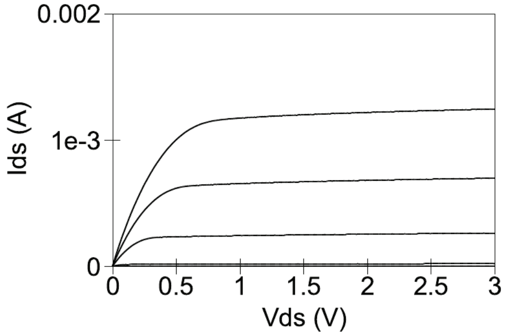
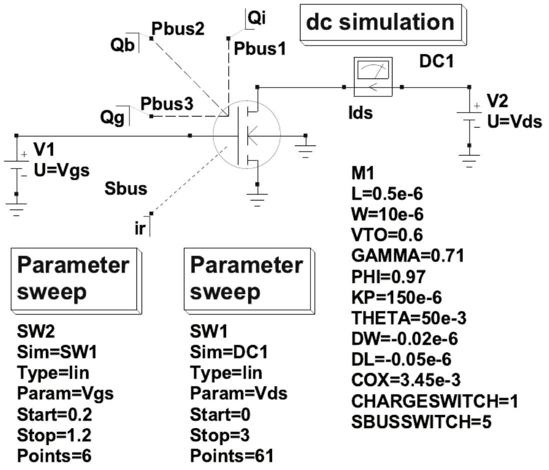


Fig. 11. Typical nMOS transistor output characteristics showing the effects caused by setting short channel control switches: CLMSWITCH=1, TFRMRSWITCH=1, CHSHSWITCH=1, RSCESWITCH=1 and CHARGESWITCH or CAPSWITCH=0.

to a set of model equations. This allows a complete post simulation analysis of the performance of a model to be undertaken easily as part of a model validation sequence. One approach with Verilog-A to obtain values for model internal signals, equations, and variables is to add extra signal nets to a model interface as part of the Verilog-A module statement. These nets act as data highways for internal model probes. Two obvious types of signal probe structure can be built into Verilog-A compact device models: firstly parallel signal probes with one probe per signal or internal variable, and secondly serial signal probes which can be switched to monitor specific signals by setting the value of a control variable prior to simulation. In practice, a combination of both parallel and serial probes are often employed to give coverage of a range of internal device signals and variable values, with priority signals connected to parallel probes. Figure 12 illustrates a typical combined parallel and series model probe configuration, here different line styles are employed to indicate the parallel and serial probes. Figure 13 shows the Verilog-A code for the parallel and serial probe signal highways and the internal model quantities being monitored. Notice that setting variable SBUSSWITCH to zero disables the serial monitoring bus and setting CHARGESWITCH to zero also disables the parallel monitoring bus. Figure 14 shows a typical group of internal model signals obtained from transistor output characteristic simulation tests.

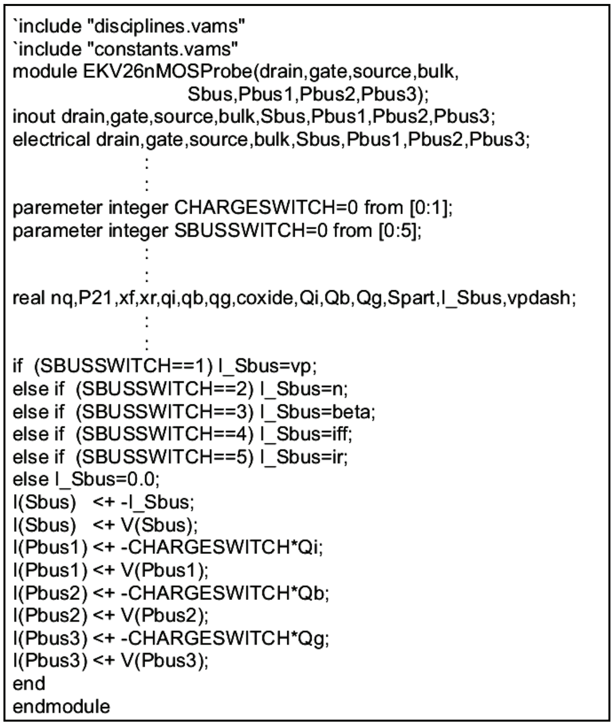


SW2
Sim=SW1
Type=lin
Param=Vgs
Start=0.2
Stop=1.2
Points=6

SW1
Sim=DC1
Type=lin
Param=Vds
Start=0
Stop=3
Points=61

M1
L=0.5e-6
W=10e-6
VTO=0.6
GAMMA=0.71
PHI=0.97
KP=150e-6
THETA=50e-3
DW=-0.02e-6
DL=-0.05e-6
COX=3.45e-3
CHARGESWITCH=1
SBUSSWITCH=5

Fig. 12. EPFL-EKV 2.6 nMOS transistor output characteristic test circuit with parallel and serial data monitoring bus structures.



```
`include "disciplines.vams"
`include "constants.vams"
module EKV26nMOSProbe(drain,gate,source,bulk,
                      Sbus,Pbus1,Pbus2,Pbus3);
inout drain,gate,source,bulk,Sbus,Pbus1,Pbus2,Pbus3;
electrical drain,gate,source,bulk,Sbus,Pbus1,Pbus2,Pbus3;
                  :
                  :
paremeter integer CHARGESWITCH=0 from [0:1];
parameter integer SBUSSWITCH=0 from [0:5];
                  :
                  :
real nq,P21,xf,xr,qi,qb,qg,coxide,Qi,Qb,Qg,Spart,I_Sbus,vpdash;
                  :
                  :
if  (SBUSSWITCH==1) I_Sbus=vp;
else if  (SBUSSWITCH==2) I_Sbus=n;
else if  (SBUSSWITCH==3) I_Sbus=beta;
else if  (SBUSSWITCH==4) I_Sbus=iff;
else if  (SBUSSWITCH==5) I_Sbus=ir;
else I_Sbus=0.0;
I(Sbus)   <+ -I_Sbus;
I(Sbus)   <+ V(Sbus);
I(Pbus1) <+ -CHARGESWITCH*Qi;
I(Pbus1) <+ V(Pbus1);
I(Pbus2) <+ -CHARGESWITCH*Qb;
I(Pbus2) <+ V(Pbus2);
I(Pbus3) <+ -CHARGESWITCH*Qg;
I(Pbus3) <+ V(Pbus3);
end
endmodule
```

Fig. 13. Serial and Parallel signal monitoring Verilog-A code for a long channel EPFL-EKV 2.6 nMOS transistor model: Only those lines of code which need to be modified or added to the Verilog-A code given in Fig. 4 are listed.

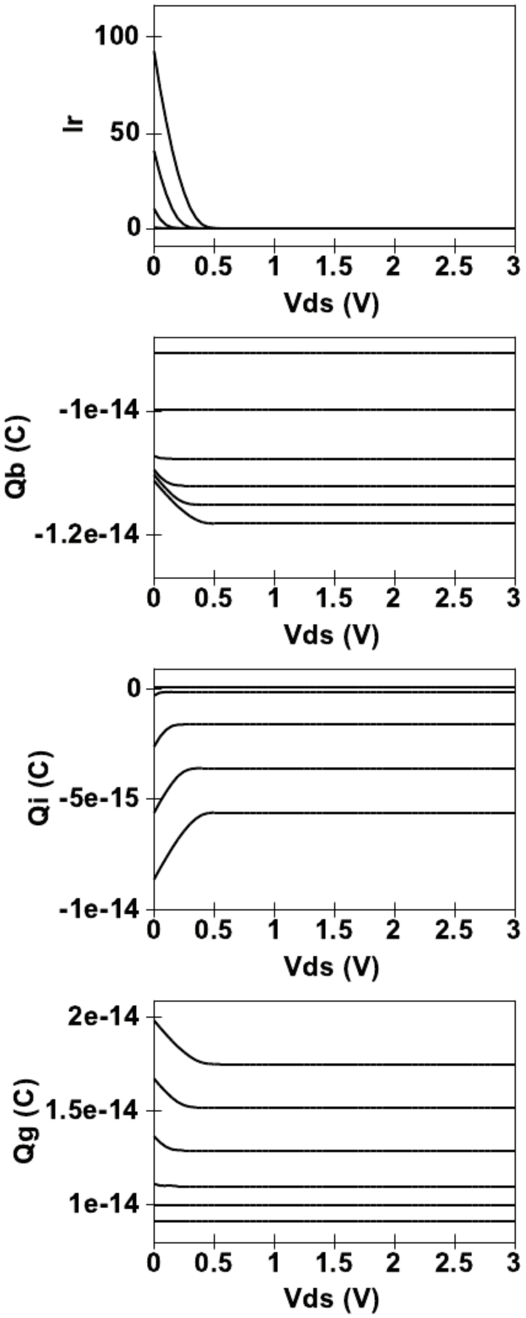


Fig. 14. A typical set of serial and parallel monitoring bus signals plotted as a function of Vds: SBUSSWITCH=5, CHARGESWITCH=1 and other parameters the same as those given in Fig. 12.

VI. Conclusions

Adaptive versions of complex compact semiconductor models promote simplified device modelling via user selective equation-defined model structures. This type of compact model provides users with the opportunity to select device features which are ”custom built” to meet the requirements of a specific type of circuit simulation. In reality, when selecting model features, there is a trade off between model complexity and simulation speed, which in turn encourages users to select models with minimum model complexity that offer improved simulation speed. This paper outlines the adaptation of the well known EPFL-EKV 2.6 MOS model as a structured semiconductor device model, while simultaneously presenting Verilog-A and Modelica simulation data which illustrates some of properties, and advantages, of modular compact device models.

Acknowledgement

The material reported in this paper forms part of the work done in response to the MOS Modeling and Parameter Extraction Working Group (MOS-AK) Verilog-A compact modelling standardization initiative [14]. The authors would like to thank ITI, GmbH Germany for providing copies of their SimulationX [8] software for part of the Modelica study reported in this paper.

References

[1] M.E. Brinson and S. Jahn, “Qucs: A GPL software package for circuit simulation, compact device modelling and circuit macromodelling from DC to RF and beyond”, International Journal of Numerical Modelling: Networks, Devices and Fields, vol. 22, pp. 297-319, 2009.

[2] L.Lemaitre, W. Grabinski and C. McAndrew, “Compact device modelling using Verilog-A and ADMS”, Electron Technology Internet Journal, , Vol 35, pp1-5, 2003.

[3] Accellera, “Verilog-AMS Language Reference Manual, Version 2.3’, 2008, http://www.accellera.org/activities/verilog-ams [accessed July 2012].

[4] W. Grabinski, M. Brinson, S. Mijalkovic, D. Pattullo and M.Pronath, ”Panel Discussion: Better analog modeling and integration with iPDKs“, GSA European Analog/Mixed-Signal and Modeling Technical Workshop, GSA and IET International Semiconductor Forum, London, UK, May 18-19, 2010.

[5] B. Johnson, T. Quarles, A.R. Newton, D.O. Pederson and A. Sangiovanni-Vincentelli, “SPICE3 Version 3f User’s Manual, Department of Electrical Engineering and Computer Science, University of California, Berkeley, CA, 1992.

[6] M. Margraf, S. Jahn,J. Flucke, R. Jacob, V. Habchi, T. Ishikawa, A. Gopala Krishna, M. Brinson, H. Parruitte, B. Roucaries, and G. Kraut, ”Qucs (Quite universal circuit simulator)“, 2003, http://qucs.sourceforge.net/index.html, [accessed March 2011].

[7] Open Source Modelica Consortium (OSCM), ”OpenModelica“, http://www.openmodelica.org/, [accessed July 2012].

[8] ITI GmbH,Germany, ”SimulationX 3.3“, http://www.simulationx.com, [accessed July 2012].

[9] M. Bucher, C. Lallement, C. Enz, F. Th´eodoloz and F. Krummenacher, ”The EPFL-EKV model equations for simulation, v2.6“ Electronics Laboratories, Swiss Federal Institute of Technology, Lausanne (EPFL), Switzerland, 1997.

[10] M.E. Brinson and S. Jahn, ”Compact device modeling for established and emerging technologies with the Qucs GPL circuit simulator“, MIXDES 2009, 16 International Conference, Mixed Design of Integrated Circuits, Lódx, Poland, June 25-27, 2009.

[11] Linear Technology, ”LTspice IV“, http://www.linear.com/designtools/software/, [accessed July 2012].

[12] Spice OPUS Homepage, http://www.fe.uni-lj.si/spice, [accessed July 2011].

[13] P. Nenzi, et al., ”ngspice“, http://www.ngspice.sourceforge.net/index.html, [accessed July 2012].

[14] S. Jahn, M. Brinson, H. Parruitte, B. Ardouin, P. Nenzi and L. Lemaitre, ”GNU simulators supporting Verilog-A compact model standardization“, MOS-AK meeting, Premstaetten 2007, http://www.mosak.org/premstaetten/papers/MOSAK_QUCS_ngspice_ADMS.pdf, [accessed July 2012].

M.E. Brinson received a first class honours BSc degree in the Physics and Technology of Electronics from the United Kingdom Council for National Academic Awards in 1965, and a PhD in Solid State Physics from London University in 1968. Since 1968 Dr. Brinson has held academic posts in Electronics and Computer Science. From 1997 till 2000 he was a visiting Professor of Analogue Microelectronics at Hochschule, Breman, Germany. Currently, he is a visiting Professor at the Centre for Communication Technology Research, London Metropolitan University, UK. He is a Chartered Engineer (CEng) and a Fellow of the Institution of Engineering and Technology (FIET), a Chartered Physicist (CPhys), and a member of the Institute of Physics (MInstP). Prof. Brinson Joined the Qucs project development team in 2006, specialising in device and circuit modelling, testing and document preparation.

H. Nabijou received his BSc in Electronic Communications, his MSc in Digital Systems and Instrumentation and his PhD in Statistical Signal Processing from UK universities. His research interests include modelling of non-linear stochastic systems. He is currently an academic member of staff in the Faculty of Computing, and a member of the Centre for Communication Technology Research, London Metropolitan University, UK. He is a member of the IEEE and a Fellow of the Royal Statistical Society. Dr Nabijou has recently joined the Qucs development project, as an associate, working on the modelling of signal processing components and algorithms.