Qucs

A Report

Verilog-A implementation of the EKV v2.6 long and short channel MOSFET models

Mike Brinson

Introduction

This report presents the background to the Qucs implementation of the EKV 2.6 long and short channel MOSFET models. During 2007 the Qucs development team employed the EKV v2.6 MOSFET model as a test case while developing the Qucs non-linear equation defined devices (EDD)[1]. More recently complete implementations of the long and short channel EKV v2.6 models have been developed using the Qucs Verilog-A compact device modelling route. This work forms part of the Verilog-A compact device modelling standardisation initiative[2]. The EKV v2.6 MOSFET model is a physics based model which has been placed in the public domain by its developers. It is ideal for analogue circuit simulation of submicron CMOS circuits. Since the models introduction and development between 1997 and 1999 it has been widely used in industry and by academic circuit design groups. Today the EKV v2.6 model is available with most of the major commercial simulators and a growing number of GPL simulators. The Verilog-A code for the Qucs ADMS[3] compiled version of the EKV v2.6 model is given in an appendix to this report.

Effects modelled

The EKV v2.6 MOSFET model includes the following effects:

- Basic geometrical and process related features dependent on oxide thickness, junction depth, effective channel length and width
- Effects of doping profile
- Modelling of weak, moderate and strong inversion behaviour
- Modelling of mobility effects due to vertical and lateral fields, velocity saturation
- Short channel effects including channel-length modulation, source and drain charge-sharing and reverse channel effect
- Modelling of substrate current due to impact ionization
- Thermal and flicker noise
- First order non-quasistatic model for the transconductances

---

[1] An example EDD macromodel of the short channel EKV 2.6 model can be found at `http://qucs.sourceforge.net/`.

[2] Stefan Jahn, Mike Brinson, Michael Margraf, Hélène Parruitte, Bertrand Ardouin, Paolo Nenzi and Laurent Lemaitre, GNU Simulators Supporting Verilog-A Compact Model Standardization, MOS-AK Meeting, Premstaetten, 2007, `http://www.mos-ak.org/premstaetten/papers/MOS-AK_QUCS_ngspice_ADMS.pdf`

[3] Lemaitre L. and GU B., ADMS - a fully customizable Verilog-AMS compiler approach, MOS-AK Meeting, Montreux. Available from `http://www.mos-ak.org/montreux/posters/17_Lemaitre_MOS-AK06.pdf`

- Short-distance geometry and bias dependent device matching

The Qucs implementation of the short channel EKV v2.6 model includes nearly all the features listed above[4]. A simpler long channel version of the model is also available for those simulations that do not require short channel effects. Both nMOS and pMOS devices have been implemented. No attempt is made in this report to describe the physics of the EKV v2.6 model. Readers who are interested in learning more about the background to the model, its physics and function should consult the following references:

- Matthias Bucher *et. al.*, The EPFL-EKV MOSFET Model Equations for Simulation, Electronics Laboratories, Swiss Federal Institute of Technology (EPFL), Lausanne, Switzerland, Model Version 2.6, Revision II, July 1998.
- Władysław Grabiński *et. al.* Advanced compact modelling of the deep submicron technologies, Journal of Telecommunications and Information Technology, 3-4/2000, pp. 31-42.
- Matthias Bucher *et. al.*, A MOS transistor model for mixed analog-digital circuit design and simulation, pp. 49-96, Design of systems on a chip - Devices and Components, KLUWER Academic Publishers, 2004.
- Trond Ytterdal *et. al.*, Chapter 7: The EKV model, pp. 209-220, Device Modeling for Analog and RF CMOS Circuit Design, John Wiley & Sons, Ltd, 2003.
- Patrick Mawet, Low-power circuits and beyond: a designer's perspective on the EKV model and its usage, MOS-AK meeting, Montreux, 2006, `http://www.mos-ak.org/montreux/posters/09_Mawet_MOS-AK06.pdf`
- Christian C. Enz and Eric A. Vittoz, Charge-based MOS transistor Modeling - The EKV model for low-power and RF IC design, John Wiley & Sons, Ltd, 2006.

The Qucs long channel EKV v2.6 model

A basic DC model for the long channel nMOS EKV v2.6 model is given at the EKV Compact MOSFET model website[5]. Unfortunately, this model is only of limited practical use due to its restricted modelling features[6]. It does however, provide a very good introduction to compact device modelling using the Verilog-A hardware description language. Readers who are unfamiliar with the Verilog-A hardware description language should consult the following references:

---

[4] This first release of the Qucs implementation of the EKV v2.6 MOSFET model does not include the first-order non-quasistatic model for transconductances.

[5] See `http://legwww.epfl.ch/ekv/verilog-a/` for the Verilog-A code.

[6] No dynamic, noise or temperature effects.

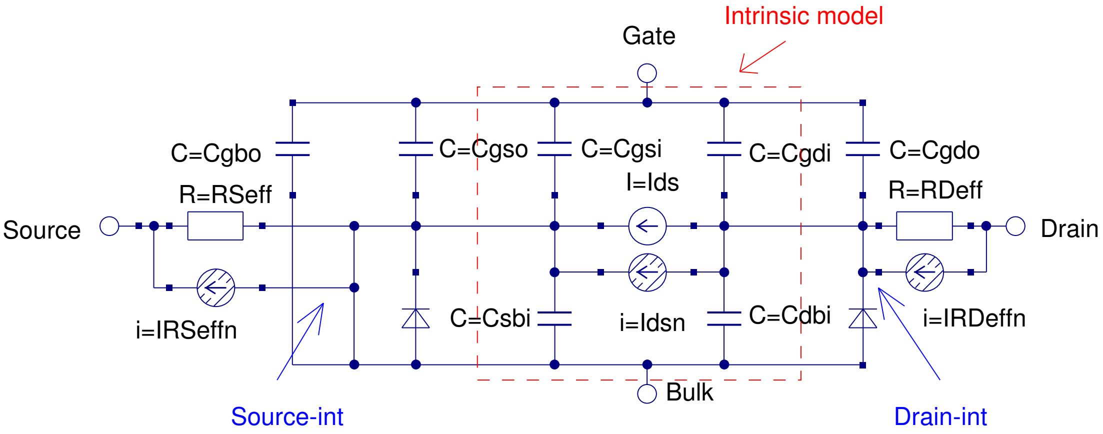


Figure 1: Equivalent circuit for the Qucs EKV v2.6 long channel nMOS model

- Accellera, Verilog-AMS Language Reference Manual, Version 2.2, 2004, Available from `http://www.accellera.org`.
- Kenneth S. Kundert and Olaf Zinke, The Designer's Guide to Verilog-AMS, Kluwer Academic Publishers, 2004.
- Dan Fitzpatrick and Ira Miller, Analog Behavioral Modeling with the Verilog-A Language, Kluwer Academic Publishers, 1998.
- Coram G. J., How to (and how not to) write a compact model in Verilog-A, 2004, IEEE International Behavioural modeling and Simulation Conference (BMAS2004), pp. 97-106.

The equivalent circuit of the Qucs EKV long channel n type MOSFET model is shown in Fig. 1. In this model the inner section, enclosed with the red dotted box, represents the fundamental intrinsic EKV v2.6 elements. The remaining components model extrinsic elements which represent the physical components connecting the intrinsic MOSFET model to its external signal pins. In the Qucs implementation of the EKV v2.6 long channel MOSFET model the drain to source DC current Ids is represented by the equations listed in a later section of the report, capacitors *Cgdi*, *Cgsi*, *Cdbi* and *Csbi* are intrinsic components derived from the charge-based EKV equations, capacitors *Cgdo*, *Cgso* and *Cgbo* represent external overlap elements, the two diodes model the drain to channel and source to channel junctions (including diode capacitance) and resistors *RDeff* and *RSeff* model series connection resistors in the drain and source signal paths respectively.

Long channel model parameters (LEVEL = 1)

Name	Symbol	Description	Unit	Default nMOS	Default pMOS
LEVEL		Model selector		1	1
L	L	length parameter	m	$10e-6$	$10e-6$

Name	Symbol	Description	Unit	Default nMOS	Default pMOS
W	W	width parameter	m	$10e-6$	$10e-6$
Np	Np	parallel multiple device number		1	1
Ns	Ns	series multiple device number		1	1
Cox	Cox	gate oxide capacitance per unit area	F/m^2	$3.4e-3$	$3.4e-3$
Xj	Xj	metallurgical junction length	m	$0.15e-6$	$0.15e-6$
Dw	Dw	channel width correction	m	$-0.02e-6$	$-0.02e-6$
Dl	Dl	channel length correction	m	$-0.05e-6$	$-0.05e-6$
Vto	Vto	long channel threshold voltage	V	0.5	-0.55
Gamma	$Gamma$	body effect parameter	$\sqrt{V}$	0.7	0.69
Phi	Phi	bulk Fermi potential	V	0.5	0.87
Kp	Kp	transconductance parameter	A/V^2	$50e-6$	$20e-6$
Theta	$Theta$	mobility reduction coefficient	$1/V$	$50e-3$	$50e-3$
Tcv	Tcv	threshold voltage temperature coefficient	V/K	$1.5e-3$	$-1.4e-3$
Hdif	$Hdif$	heavily doped diffusion length	m	$0.9e-6$	$0.9e-6$
Rsh	Rsh	drain-source diffusion sheet resistance	$\Omega/square$	510	510
Rsc	Rsc	source contact resistance	Ω	0.0	0.0
Rdc	Rdc	drain contact resistance	Ω	0.0	0.0
Cgso	$Cgso$	gate to source overlay capacitance	F	$1.5e-10$	$1.5e-10$
Cgdo	$Cgdo$	gate to drain overlay capacitance	F	$1.5e-10$	$1.5e-10$
Cgbo	$Cgbo$	gate to bulk overlay capacitance	F	$4e-10$	$4e-10$
N	N	diode emission coefficient		1.0	1.0
Is	Is	leakage current	A	$1e-1$	$1e-14$
Bv	Bv	reverse breakdown voltage	V	100	100
Ibv	Ibv	current at Bv	A	$1e-3$	$1e-3$
Vj	Vj	junction potential	V	1.0	1.0
Cj0	$Cj0$	zero bias depletion capacitance	F	$1e-12$	$1e-12$
M	M	grading coefficient		0.5	0.5
Area	$Area$	relative area		1.0	1.0
Fc	Fc	forward-bias depletion capcitance coefficient		0.5	0.5
Tt	Tt	transit time	s	$0.1e-9$	$0.1e-9$
Xti	Xti	saturation current temperature exponent		3.0	3.0
Kf	KF	flicker noise coefficient		$1e-27$	$1e-28$
Af	Af	flicker noise exponent		1.0	1.0
Tnom	$Tnom$	parameter measurement temperature	$^\circ C$	26.85	26.85
Temp	$Temp$	device temperature	$^\circ C$	26.85	26.85

Fundamental long channel DC model equations (LEVEL = 1)

<22> $Vg = V(Gate) - V(Bulk)$

<23> $Vs = V(Source) - V(Bulk)$

<24> $Vd = V(Drain) - V(Bulk)$

<33> $VGprime = Vg - Vto + Phi + Gamma \cdot \sqrt{Phi}$

<34> $$Vp = VGprime - Phi - Gamma \cdot \left(\sqrt{VGprime + \left[\frac{Gamma}{2}\right]^2} - \frac{Gamma}{2} \right)$$

<39> $$n = 1 + \frac{Gamma}{2 \cdot \sqrt{Vp + Phi + 4 \cdot Vt}}$$

<58, 64> $$\beta = Kp \cdot \frac{W}{L} \cdot \frac{1}{1 + Theta \cdot Vp}$$

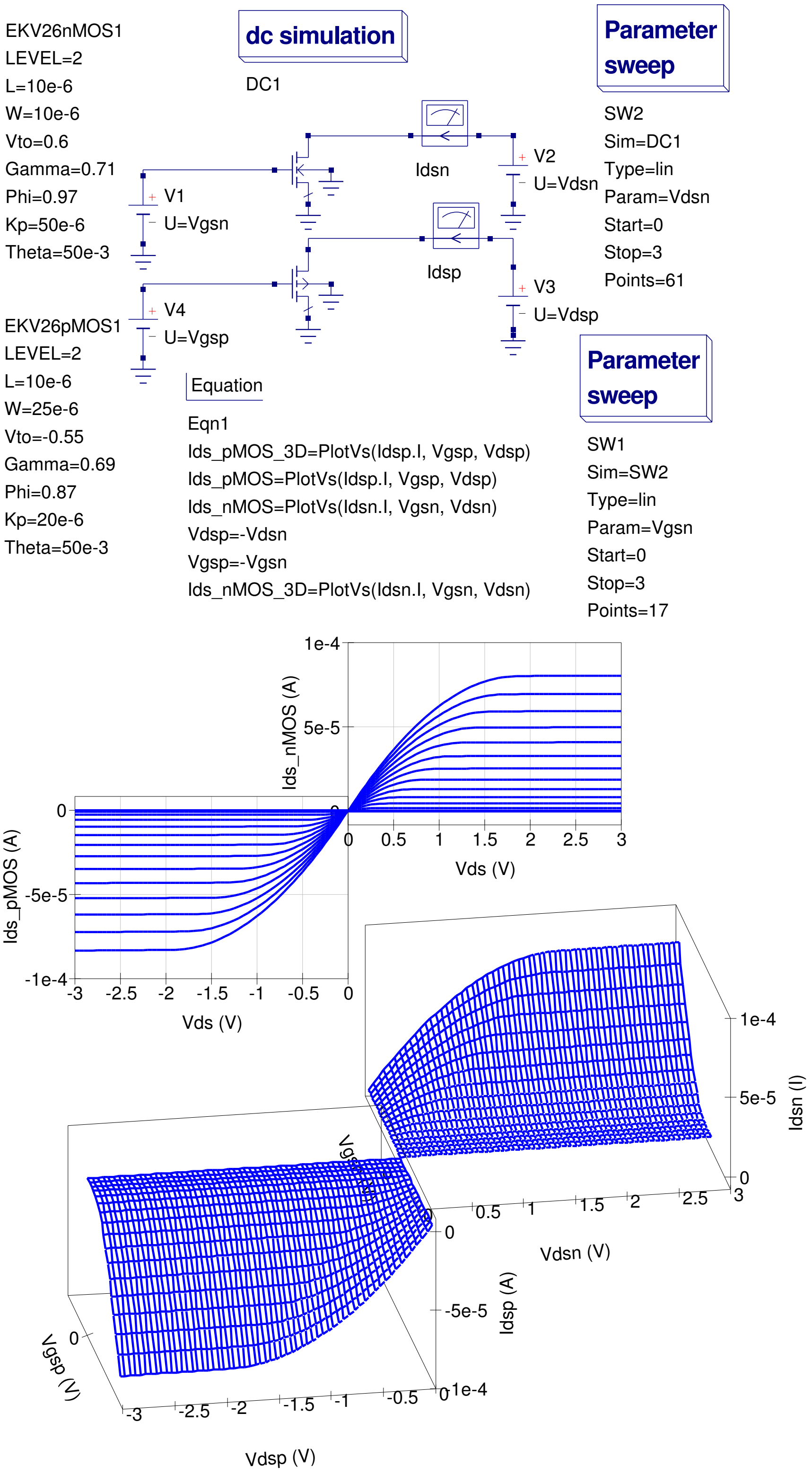
<44> $$X1 = \frac{Vp - Vs}{Vt} \qquad If = \left[ln \left\{ 1 + limexp\left(\frac{X1}{2}\right) \right\} \right]^2$$

<57> $$X2 = \frac{Vp - Vd}{Vt} \qquad Ir = \left[ln \left\{ 1 + limexp\left(\frac{X2}{2}\right) \right\} \right]^2$$

<65> $Ispecific = 2 \cdot n \cdot \beta \cdot Vt^2$

<66> $$Ids = Ispecific \cdot (If - Ir)$$

Where *VGprime* is the effective gate voltage, *Vp* is the pinch-off voltage, n is the slope factor, β is a transconductance parameter, *Ispecific* is the specific current, *If* is the forward current, *Ir* is the reverse current, *Vt* is the thermal voltage at the device temperature, and *Ids* is the drain to source current. EKV v2.6 equation numbers are given in "< >" brackets at the left-hand side of each equation. Typical plots of Ids against Vds for both the nMOS and pMOS long channel devices are given in Figure 2.



EKV26nMOS1
LEVEL=2
L=10e-6
W=10e-6
Vto=0.6
Gamma=0.71
Phi=0.97
Kp=50e-6
Theta=50e-3

EKV26pMOS1
LEVEL=2
L=10e-6
W=25e-6
Vto=-0.55
Gamma=0.69
Phi=0.87
Kp=20e-6
Theta=50e-3

DC1

V1
U=Vgsn

V4
U=Vgsp

V2
U=Vdsn

V3
U=Vdsp

SW2
Sim=DC1
Type=lin
Param=Vdsn
Start=0
Stop=3
Points=61

SW1
Sim=SW2
Type=lin
Param=Vgsn
Start=0
Stop=3
Points=17

Eqn1
Ids_pMOS_3D=PlotVs(Idsp.I, Vgsp, Vdsp)
Ids_pMOS=PlotVs(Idsp.I, Vgsp, Vdsp)
Ids_nMOS=PlotVs(Idsn.I, Vgsn, Vdsn)
Vdsp=-Vdsn
Vgsp=-Vgsn
Ids_nMOS_3D=PlotVs(Idsn.I, Vgsn, Vdsn)

Figure 2: Ids versus Vds plots for the Qucs EKV v2.6 long channel nMOS and pMOS models

Testing model performance

Implementing advanced component models like the EKV v2.6 MOSFET model is a complex process, involving the translation of a set of equations into the Verilog-A hardware design language, conversion of the Verilog-A code into C++ code via the ADMS compiler, and finally compiling and linking the model code with the main body of Qucs code. At all stages in the process accuracy becomes an important issue. This section of the Qucs EKV v2.6 report introduces a number of test simulations which were used during the model development cycle to check the performance of the Qucs EKV v2.6 implementation. The tests also demonstrate how a circuit simulator can be used to extract model parameters. The values of which help to confirm correct model operation.

Extraction of Ispec

When a MOS transistor is operating in the saturation region, reverse current *Ir* approaches zero and the drain to source current is approximated by

$$Ids = Ispecific \cdot If = Ispecific \cdot \left[ln\{1 + limexp\left(\frac{Vp - Vs}{2 \cdot Vt}\right)\}\right]^2 \tag{1}$$

In saturation $limexp\left(\frac{Vp-Vs}{2 \cdot Vt}\right) >> 1$, yielding

$$\sqrt{Ids} = \sqrt{\frac{Ispecific}{2 \cdot Vt^2}} \cdot (Vp - Vs) \tag{2}$$

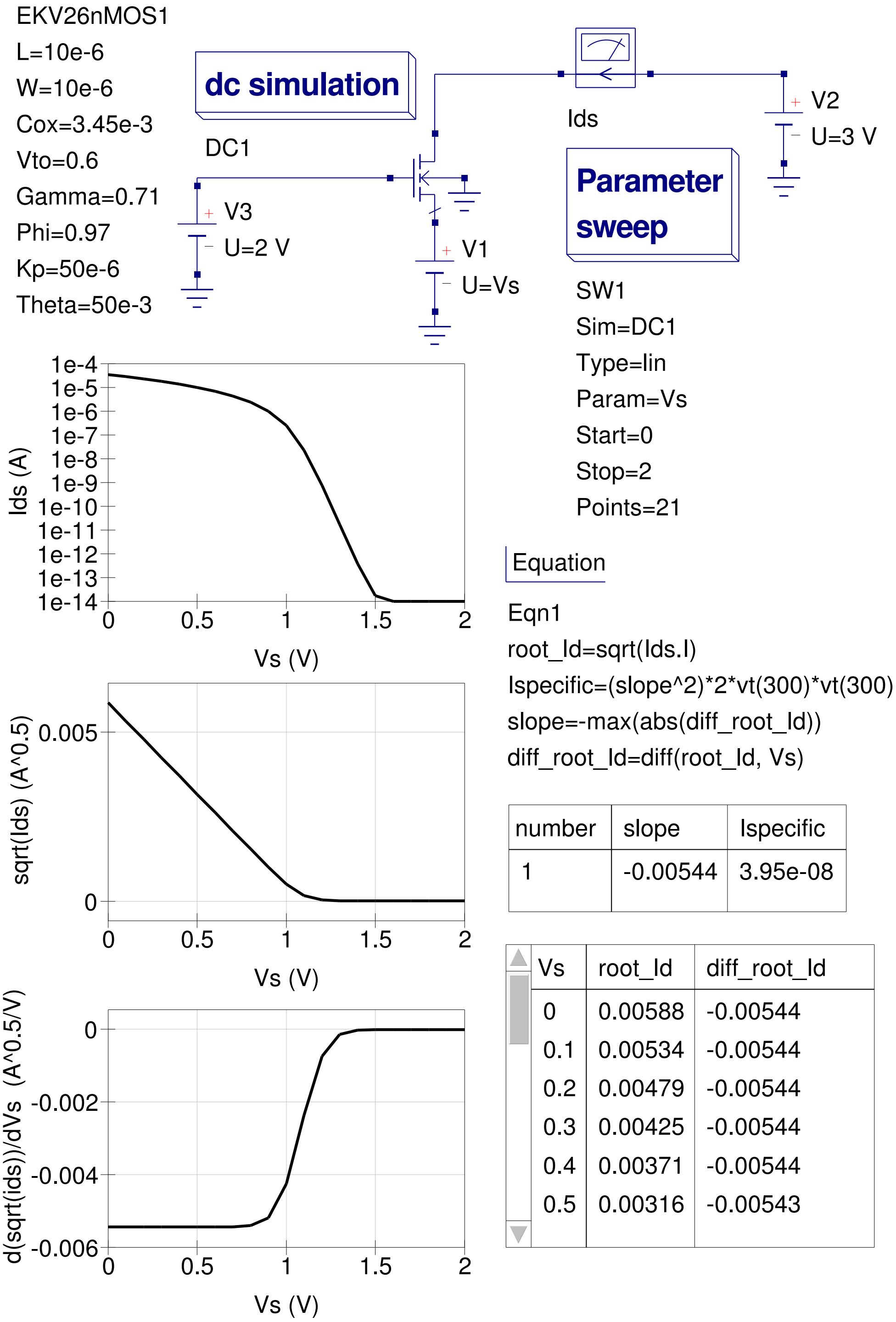
Hence

$$\frac{\partial(\sqrt{Ids})}{\partial Vs} = -\sqrt{\frac{Ispecific}{2 \cdot Vt^2}} = -slope \tag{3}$$

Or

$$Ispecific = 2 \cdot slope^2 \cdot Vt^2 \tag{4}$$

Figure 3 shows a typical test circuit configuration for measuring and simulating *Ids* with varying *Vs*. Qucs post-simulation functions in equation block Eqn1 are used to calculate the value for *Ispecific*. The value of *Ispecific* for the nMOS transistor with the parameters given in Fig. 3 is 3.95e-8 A. Figure 4 illustrates a test circuit for measuring *Vp* with the transistor in saturation. In this circuit *Is=Ispecific* and the threshold voltage corresponds to *Vg* when *Vp* = 0V. Notice also that $n = \partial Vg / \partial Vp$. In Fig. 4 Qucs post-simulation processing functions are also used to generate data for *Vp*, *VGprime* and *n*. The value of the threshold voltage for the device shown in Fig. 4 is 0.6V. At this voltage n = 1.37. The two test configurations illustrated in Figs. 3 and 4 go some way to confirming that the Qucs implementation of the EKV v2.6 long channel model is functioning correctly.



EKV26nMOS1
L=10e-6
W=10e-6
Cox=3.45e-3
Vto=0.6
Gamma=0.71
Phi=0.97
Kp=50e-6
Theta=50e-3

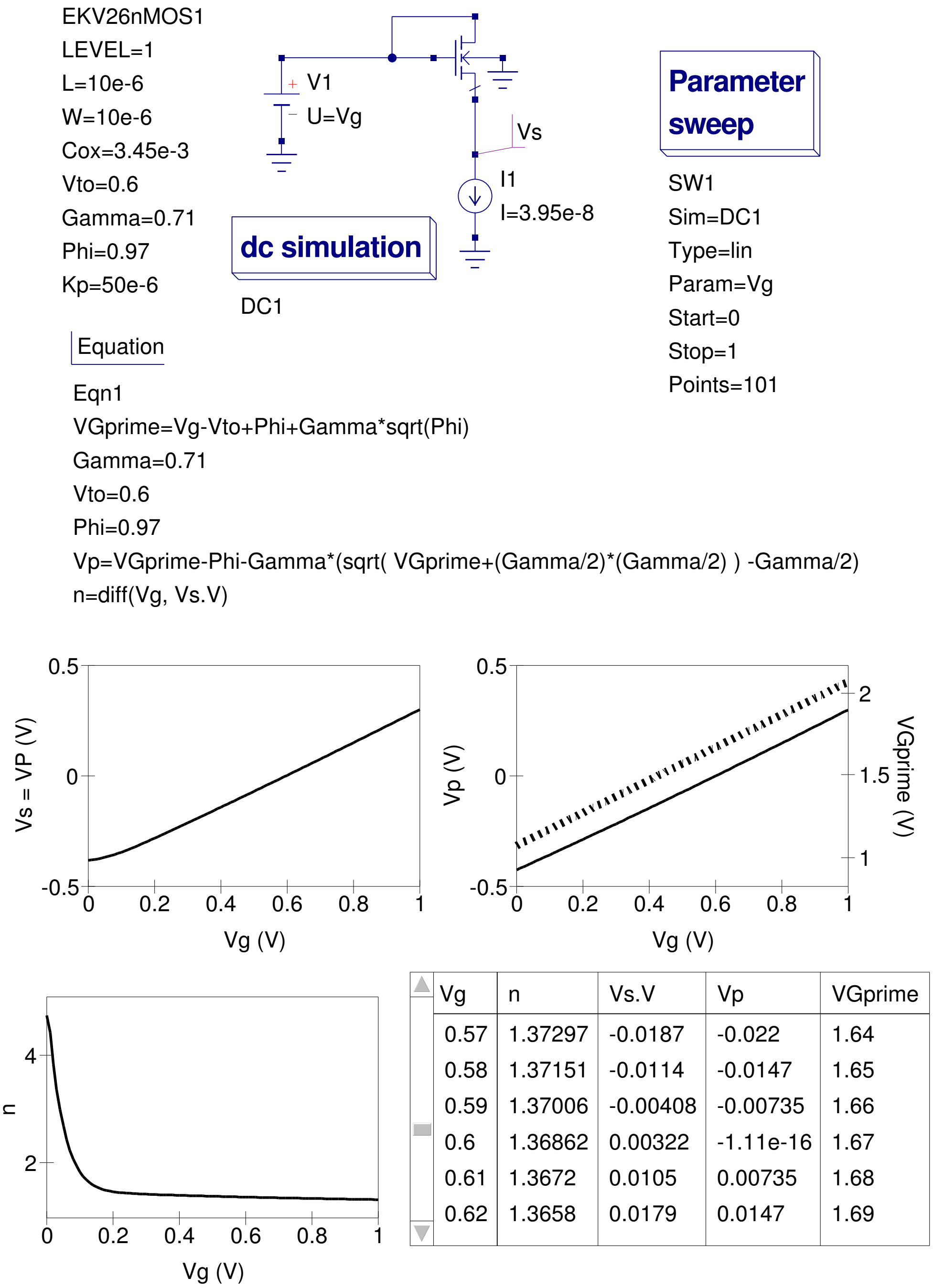
SW1
Sim=DC1
Type=lin
Param=Vs
Start=0
Stop=2
Points=21

Eqn1
root_Id=sqrt(Ids.I)
Ispecific=(slope^2)*2*vt(300)*vt(300)
slope=-max(abs(diff_root_Id))
diff_root_Id=diff(root_Id, Vs)

number	slope	Ispecific
1	-0.00544	3.95e-08

Vs	root_Id	diff_root_Id
0	0.00588	-0.00544
0.1	0.00534	-0.00544
0.2	0.00479	-0.00544
0.3	0.00425	-0.00544
0.4	0.00371	-0.00544
0.5	0.00316	-0.00543

Figure 3: Ispecific extraction test circuit and post simulation data processing results



EKV26nMOS1
LEVEL=1
L=10e-6
W=10e-6
Cox=3.45e-3
Vto=0.6
Gamma=0.71
Phi=0.97
Kp=50e-6

V1
U=Vg

I1
I=3.95e-8

DC1

SW1
Sim=DC1
Type=lin
Param=Vg
Start=0
Stop=1
Points=101

Eqn1
VGprime=Vg-Vto+Phi+Gamma*sqrt(Phi)
Gamma=0.71
Vto=0.6
Phi=0.97
Vp=VGprime-Phi-Gamma*(sqrt(VGprime+(Gamma/2)*(Gamma/2)) -Gamma/2)
n=diff(Vg, Vs.V)

Vg	n	Vs.V	Vp	VGprime
0.57	1.37297	-0.0187	-0.022	1.64
0.58	1.37151	-0.0114	-0.0147	1.65
0.59	1.37006	-0.00408	-0.00735	1.66
0.6	1.36862	0.00322	-1.11e-16	1.67
0.61	1.3672	0.0105	0.00735	1.68
0.62	1.3658	0.0179	0.0147	1.69

Figure 4: Vp extraction test circuit and post simulation data processing results

Extraction of model intrinsic capacitance

The Qucs implementation of the EKV v2.6 MOSFET model uses the charge-based model for transcapacitances. This model ensures charge-conservation during transient analysis. Both the long channel and short channel versions employ the quasi-static charge-based model. The EKV v2.6 charge equations for the long channel intrinsic device are:

$$nq = 1 + \frac{Gamma}{2 \cdot \sqrt{Vp + Phi + 1e - 6}} \tag{5}$$

$$Xf = \sqrt{\frac{1}{4} + If} \tag{6}$$

$$Xr = \sqrt{\frac{1}{4} + Ir} \tag{7}$$

$$qD = -nq \cdot \left\{ \frac{4}{15} \cdot \frac{3 \cdot Xr^3 + 6 \cdot Xr^2 \cdot Xf + 4 \cdot Xr \cdot Xf^2 + 2 \cdot Xf^3}{(Xf + Xr)^2} - \frac{1}{2} \right\} \tag{8}$$

$$qS = -nq \cdot \left\{ \frac{4}{15} \cdot \frac{3 \cdot Xf^3 + 6 \cdot Xf^2 \cdot Xr + 4 \cdot Xf \cdot Xr^2 + 2 \cdot Xr^3}{(Xf + Xr)^2} - \frac{1}{2} \right\} \tag{9}$$

$$qI = qS + qD = -nq \cdot \left\{ \frac{4}{3} \cdot \frac{3 \cdot Xf^3 + Xr \cdot Xf + Xr^2}{Xf + Xr} - 1 \right\} \tag{10}$$

$$qB = -Gamma \cdot \sqrt{Vp + Phi + 1e - 6} \cdot \frac{1}{Vt} - \left(\frac{nq - 1}{nq} \right) \cdot qI \quad \forall (VGprime > 0) \tag{11}$$

$$qB = -VGprime \cdot \frac{1}{Vt} \quad \forall (VGprime <= 0) \tag{12}$$

$$qG = -qI - qB \tag{13}$$

$$COX = Cox \cdot Np \cdot Weff \cdot Ns \cdot Leff \tag{14}$$

$$Q(I, B, D, S, G) = COX \cdot Vt \cdot q(I, B, D, S, G) \tag{15}$$

The first release of the Qucs EKV v2.6 MOSFET model assumes that the gate and bulk charge is partitioned between the drain and source in equal ratio[7]. Fifty percent charge portioning yields the following *Ids* current contributions:

$$I(Gate, Source\_int) < +0.5 \cdot p\_n\_MOS \cdot ddt(QG) \tag{16}$$

---

[7] For an example of this type of charge partitioning see F. Pregaldiny et. al., An analytic quantum model for the surface potential of deep-submicron MOSFETS, 10th International Conference, MIXDES 2003, Lodz, Poland, 26-28 June 2003.

$$I(Gate, Drain\_int) < +0.5 \cdot p\_n\_MOS \cdot ddt(QG) \tag{17}$$

$$I(Source\_int, Bulk) < +0.5 \cdot p\_n\_MOS \cdot ddt(QB) \tag{18}$$

$$I(drain\_int, Bulk) < +0.5 \cdot p\_n\_MOS \cdot ddt(QB) \tag{19}$$

Where p_n_MOS = 1 for nMOS devices or -1 for pMOS devices. Charge associated with the extrinsic overlap capacitors, Cgs0, Cgd0 and Cgb0, is represented in the Qucs EKV v2.6 implementation by the following equations:

$$Qgs0 = Cgs0 \cdot Weff \cdot Np \cdot (VG - VS) \tag{20}$$

$$Qgd0 = Cgd0 \cdot Weff \cdot Np \cdot (VG - VD) \tag{21}$$

$$Qgb0 = Cgb0 \cdot Leff \cdot Np \cdot VB \tag{22}$$

The drain to bulk and source to bulk diodes also introduce additional components in the extrinsic capacitance model. The default value of CJ0 being set at 300fF. Analysis of the y-parameters[8] for the EKV v2.6 equivalent circuit shown in the test circuit illustrated in Fig. 5 yields

$$y_{11} = \frac{j \cdot \omega \cdot Cg}{1 + \omega^2 \cdot (Rgn \cdot Cg)^2} \tag{23}$$

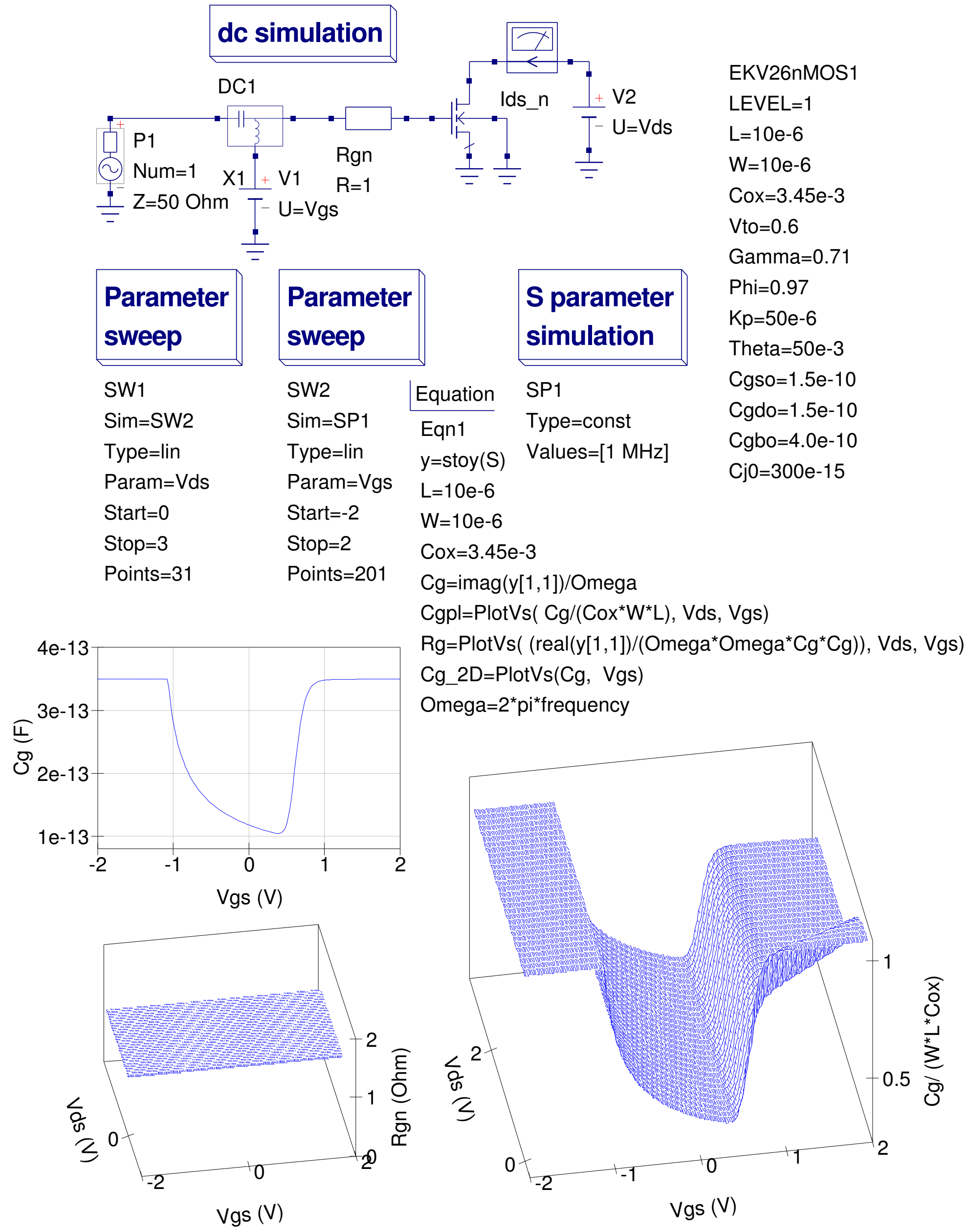
Or

$$y_{11} \cong \omega^2 \cdot Rg \cdot Cg^2 + j \cdot \omega \cdot Cg, \quad \text{when} \quad \omega \cdot Rg \cdot Cg << 1. \tag{24}$$

Hence, $Cg = imag(y_{11}/\omega)$ and $Rgn = real(y_{11}/(\omega^2 \cdot Cg^2))$, where $\omega = 2 \cdot \pi \cdot f$, and f is the frequency of y-parameter measurement, Rg is a series extrinsic gate resistance and $Cg \cong Cgs + Cgd + Cgb$. With equal partitioning of the intrinsic gate charge Cgb approximates to zero and $Cg \cong Cgs + Cgd$. The data illustrated in Figures 5 and 6 shows two features which are worth commenting on; firstly the values of Cg are very much in line with simple hand calculations (for example in the case of the nMOS device $Cg(max) = W \cdot L \cdot Cox = 10e - 6 * 10e - 6 * 3.45e - 3 = 3.45e - 13$F) and secondly both sets of simulation data indicate the correct values for the nMOS and pMOS threshold voltages (for example -0.55 V for the pMOS device and 0.6 V for the nMOS device), reinforcing confidence in the EKV v2.6 model implementation.

---

[8] A more detailed anlysis of the EKV v2.6 y-parameters can be found in F. Krummenacher *et. al.*, HF MOSET MODEL parameter extraction, European Project No. 25710, Deliverable D2.3, July 28, 2000.



EKV26nMOS1
LEVEL=1
L=10e-6
W=10e-6
Cox=3.45e-3
Vto=0.6
Gamma=0.71
Phi=0.97
Kp=50e-6
Theta=50e-3
Cgso=1.5e-10
Cgdo=1.5e-10
Cgbo=4.0e-10
Cj0=300e-15

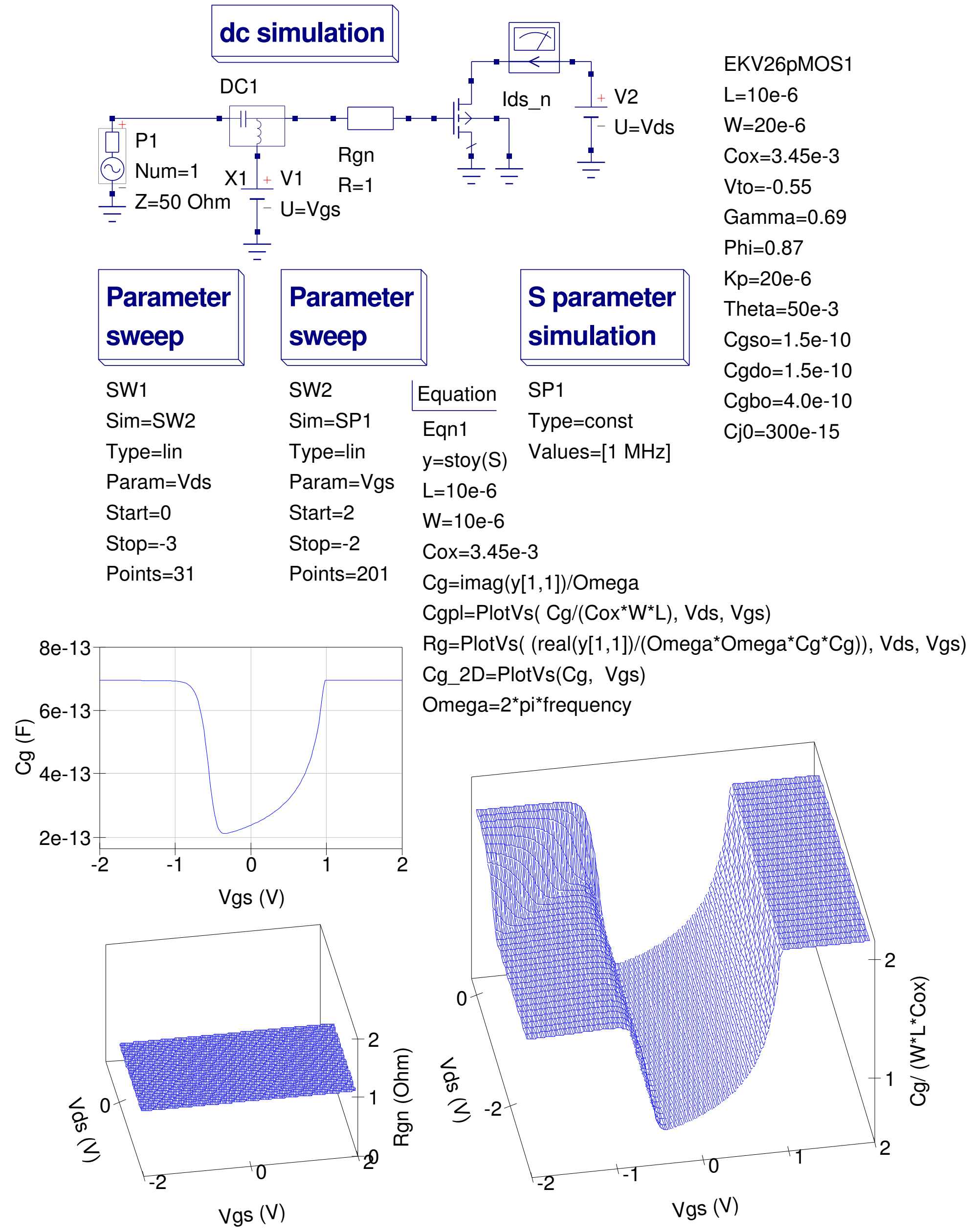
SW1
Sim=SW2
Type=lin
Param=Vds
Start=0
Stop=3
Points=31

SW2
Sim=SP1
Type=lin
Param=Vgs
Start=-2
Stop=2
Points=201

Eqn1
y=stoy(S)
L=10e-6
W=10e-6
Cox=3.45e-3
Cg=imag(y[1,1])/Omega
Cgpl=PlotVs(Cg/(Cox*W*L), Vds, Vgs)
Rg=PlotVs((real(y[1,1])/(Omega*Omega*Cg*Cg)), Vds, Vgs)
Cg_2D=PlotVs(Cg, Vgs)
Omega=2*pi*frequency

SP1
Type=const
Values=[1 MHz]

Figure 5: y_{11} test circuit and values of Cg for the long channel EKV v2.6 nMOS model



DC1

P1
Num=1
Z=50 Ohm

X1

V1
U=Vgs

Rgn
R=1

Ids_n

V2
U=Vds

EKV26pMOS1
L=10e-6
W=20e-6
Cox=3.45e-3
Vto=-0.55
Gamma=0.69
Phi=0.87
Kp=20e-6
Theta=50e-3
Cgso=1.5e-10
Cgdo=1.5e-10
Cgbo=4.0e-10
Cj0=300e-15

SW1
Sim=SW2
Type=lin
Param=Vds
Start=0
Stop=-3
Points=31

SW2
Sim=SP1
Type=lin
Param=Vgs
Start=2
Stop=-2
Points=201

Eqn1
y=stoy(S)
L=10e-6
W=10e-6
Cox=3.45e-3
Cg=imag(y[1,1])/Omega
Cgpl=PlotVs(Cg/(Cox*W*L), Vds, Vgs)
Rg=PlotVs((real(y[1,1])/(Omega*Omega*Cg*Cg)), Vds, Vgs)
Cg_2D=PlotVs(Cg, Vgs)
Omega=2*pi*frequency

SP1
Type=const
Values=[1 MHz]

Figure 6: y_{11} test circuit and values of Cg for the long channel EKV v2.6 pMOS model

Extraction of extrinsic diode capacitance and drain resistance

The extrinsic section of the EKV v2.6 model includes diodes which in turn are modelled by conventional DC characteristics and parallel capacitance. This capacitance is represented by depletion layer capacitance in the diode reverse bias region of operation. In the diode forward bias section of the I-V characteristic diffusion capacitance dominates. Figure 7 illustrates a test circuit that allows the diode capacitance to be extracted as a function of *Vds*. In Fig. 7 the nMOS device is turned off and the drain to bulk diode reverse biased. Simple analysis indicates that

$$y_{11} \cong \omega^2 \cdot RDeff \cdot Cd^2 + j \cdot \omega \cdot Cd, \quad \text{when} \quad \omega \cdot RDeff \cdot Cd << 1. \tag{25}$$

Hence, $Cd = imag(y_{11}/\omega)$ and $Rdeff = real(y_{11}/(\omega^2 \cdot Cd^2))$, where $\omega = 2 \cdot \pi \cdot f$, f is the frequency of y-parameter measurement, and Cd is the diode capacitance. The data shown in Fig. 7 indicate good agreement with the expected values for *Cd* and *RDeff*; which are expected to be *Cd* = 300fF at *Vds*=0V, and *Rdeff* = 46Ω.

Simulating EKV v2.6 MOSFET noise

The EKV v2.6 intrinsic device noise is modelled by a noise current source connected between the internal drain and source terminals. The noise current source *Idsn*, see Fig. 1, is composed of a thermal noise component and a flicker noise component. The Power Spectral Density (S_{PSD}) of these components are given by:

$$S_{PSD} = S_{thermal} + S_{flicker} \tag{26}$$

Where

- Thermal noise

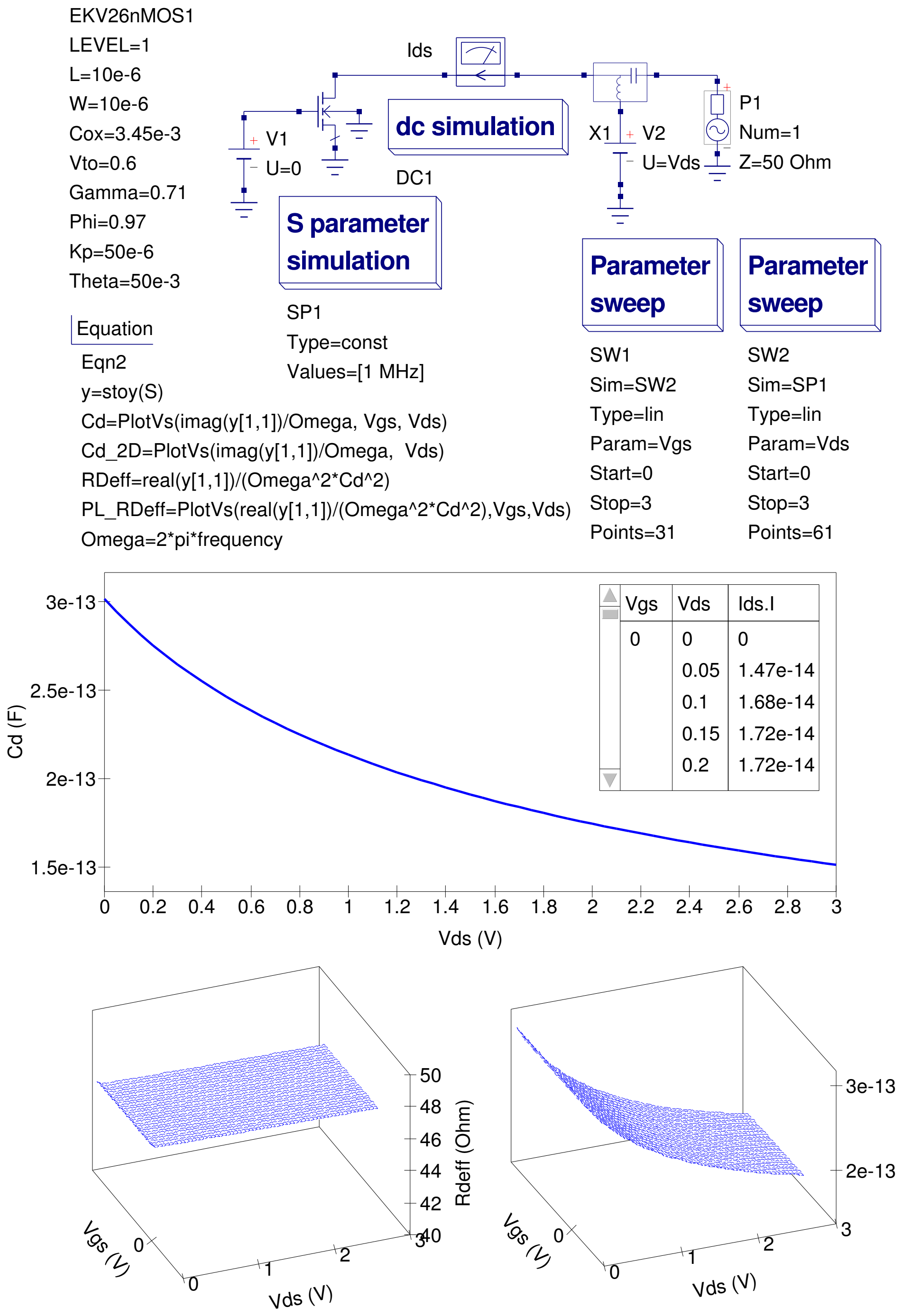
$$S_{thermal} = 4 \cdot k \cdot T \cdot \beta \cdot \mid qI \mid \tag{27}$$

- Flicker noise

$$S_{flicker} = \frac{KF \cdot g_{mg}^2}{Np \cdot Weff \cdot Ns \cdot Leff \cdot Cox \cdot f^{Af}}, \tag{28}$$

$$g_{mg} = \frac{\partial Ids}{\partial Vgs} = \beta \cdot Vt \cdot \left(\sqrt{\frac{4 \cdot If}{Ispecific} + 1} - \sqrt{\frac{4 \cdot Ir}{Ispecific} + 1} \right) \tag{29}$$

Where β is a transconductance factor, $qI = qD + qS$, and the other symbols are defined in the EKV v2.6 long channel parameter list or have their usual meaning. Noise has been implemented in both the Qucs long channel and short channel EKV v2.6 models. In addition to the intrinsic device noise the Qucs EKV v2.6 model includes the thermal noise components for both extrinsic resistors *RDeff* and *RSeff*. Figure 8 presents a typical noise test circuit and simulated noise currents. In Figure 8 four nMOS devices are biased under different DC conditions and their noise current simulated for a range of *W* values



EKV26nMOS1
LEVEL=1
L=10e-6
W=10e-6
Cox=3.45e-3
Vto=0.6
Gamma=0.71
Phi=0.97
Kp=50e-6
Theta=50e-3

V1
U=0

DC1

SP1
Type=const
Values=[1 MHz]

X1

V2
U=Vds

P1
Num=1
Z=50 Ohm

Equation
Eqn2
y=stoy(S)
Cd=PlotVs(imag(y[1,1])/Omega, Vgs, Vds)
Cd_2D=PlotVs(imag(y[1,1])/Omega, Vds)
RDeff=real(y[1,1])/(Omega^2*Cd^2)
PL_RDeff=PlotVs(real(y[1,1])/(Omega^2*Cd^2),Vgs,Vds)
Omega=2*pi*frequency

SW1
Sim=SW2
Type=lin
Param=Vgs
Start=0
Stop=3
Points=31

SW2
Sim=SP1
Type=lin
Param=Vds
Start=0
Stop=3
Points=61

Vgs	Vds	Ids.I
0	0	0
	0.05	1.47e-14
	0.1	1.68e-14
	0.15	1.72e-14
	0.2	1.72e-14

Figure 7: Test circuit for extracting EKV v2.6 extrinsic diode capacitance

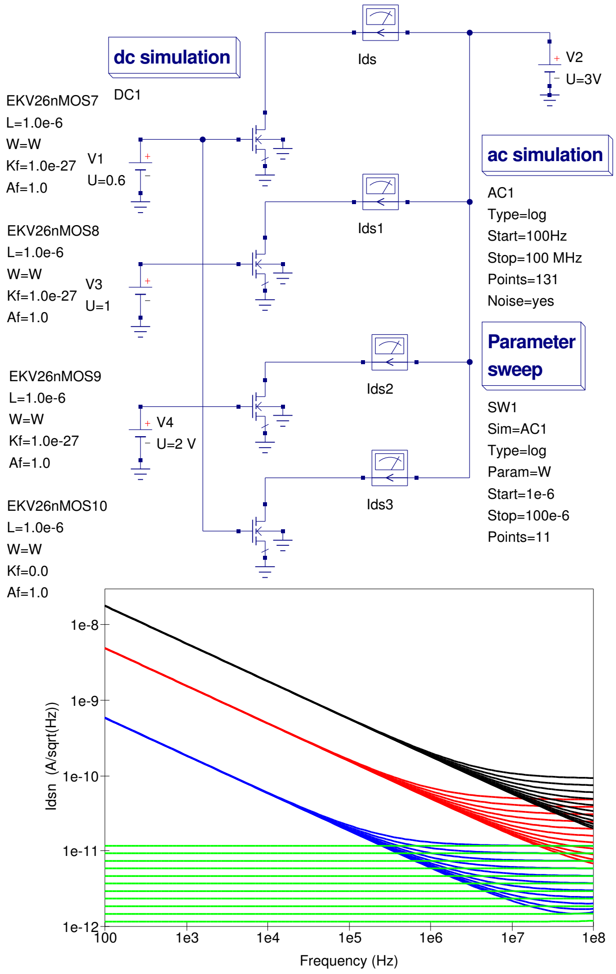


Figure 8: Test circuit for simulating EKV v2.6 noise: Ids.in blue curve, Ids1.in red curve, Ids2.in black curve and Ids3.in green curve

between 1e-6 m and 100e-6 m. The first three devices include both thermal and flicker noise components (KF = 1e-27) while the fourth device has it's flicker component set to zero. The resulting current noise curves clearly demonstrate the effect of summing intrinsic thermal and flicker components on the overall performance of the EKV v2.6 noise model.

The Qucs short channel EKV v2.6 model

The Qucs implementation of the short short channel EKV v2.6 MOSFET model contains all the features implemented in the long channel version of the model plus a number of characteristics specific to short channel operation. However, the short channel version of the model does not use parameter *Theta*. Parameter LEVEL set to 2 selects the short channel model. Both pMOS and nMOS versions of the model are available for both long and short channel implementations. The entire short channel EKV v2.6 MOSFET model is described by roughly 94 equations. Readers who are interested in the mathematics of the model should consult “The EPFL-EKV MOSFET Model Equations for Simulation” publication cited in previous text. Appendix A lists the complete Verilog-A code for the first release of the Qucs EKV v2.6 MOSFET models. Additional Verilog-A code has been added to the model equation code to (1) allow interchange of the drain and source terminals, and (2) select nMOS or pMOS devices.

Short channel model parameters (LEVEL = 2)

Name	Symbol	Description	Unit	Default nMOS	Default pMOS
LEVEL		Model selector		2	2
L	L	length parameter	m	$10e-6$	$10e-6$
W	W	width parameter	m	$10e-6$	$10e-6$
Np	Np	parallel multiple device number		1	1
Ns	Ns	series multiple device number		1	1
Cox	Cox	gate oxide capacitance per unit area	F/m^2	$3.4e-3$	$3.4e-3$
Xj	Xj	metallurgical junction length	m	$0.15e-6$	$0.15e-6$
Dw	Dw	channel width correction	m	$-0.02e-6$	$-0.02e-6$
Dl	Dl	channel length correction	m	$-0.05e-6$	$-0.05e-6$
Vto	Vto	long channel threshold voltage	V	0.5	-0.55
Gamma	$Gamma$	body effect parameter	$\sqrt{V}$	0.7	0.69
Phi	Phi	bulk Fermi potential	V	0.5	0.87
Kp	Kp	transconductance parameter	A/V^2	$50e-6$	$20e-6$
EO	EO	mobility reduction coefficient	V/m	$88e-6$	$51e-6$
Ucrit	$Ucrit$	longitudinal critical field	V/m	$4.5e-6$	$18e-6$
Lambda	$Lambda$	depletion length coefficient		0.23	1.1
Weta	$Weta$	narrow channel effect coefficient		0.05	0.0
Leta	eta	short channel effect coefficient		0.28	0.45
Q0	$Q0$	reverse short channel effect peak charge density		$280e-6$	$200e-6$
Lk	Lk	reverse short channel effect characteristic length	m	$0.5e-6$	$0.6e-6$
Tcv	Tcv	threshold voltage temperature coefficient	V/K	$1.5e-3$	$-1.4e-3$
Bex	Bex	mobility temperature coefficient		-1.5	-1.4
Ucex	$Ucex$	longitudinal critical field temperature exponent		1.7	2.0
Ibbt	$Ibbt$	temperature coefficient for Ibb	$1/K$	0.0	0.0
Hdif	$Hdif$	heavily doped diffusion length	m	$0.9e-6$	$0.9e-6$
Rsh	Rsh	drain-source diffusion sheet resistance	$\Omega/square$	510	510
Rsc	Rsc	source contact resistance	Ω	0.0	0.0
Rdc	Rdc	drain contact resistance	Ω	0.0	0.0
Cgso	$Cgso$	gate to source overlay capacitance	F	$1.5e-10$	$1.5e-10$
Cgdo	$Cgdo$	gate to drain overlay capacitance	F	$1.5e-10$	$1.5e-10$
Cgbo	$Cgbo$	gate to bulk overlay capacitance	F	$4e-10$	$4e-10$
N	N	diode emission coefficient		1.0	1.0
Is	Is	leakage current	A	$1e-1$	$1e-14$
Bv	Bv	reverse breakdown voltage	V	100	100
Ibv	Ibv	current at Bv	A	$1e-3$	$1e-3$
Vj	Vj	junction potential	V	1.0	1.0
Cj0	$Cj0$	zero bias depletion capacitance	F	$1e-12$	$1e-12$
M	M	grading coefficient		0.5	0.5
Area	$Area$	relative area		1.0	1.0
Fc	Fc	forward-bias depletion capcitance coefficient		0.5	0.5
Tt	Tt	transit time	s	$0.1e-9$	$0.1e-9$
Xti	Xti	saturation current temperature exponent		3.0	3.0
Kf	KF	flicker noise coefficient		$1e-27$	$1e-28$
Af	Af	flicker noise exponent		1.0	1.0
Avto	$Avto$	area related threshold mismatch parameter		0	0
Akp	Akp	area related gain mismatch parameter		0	0
Agamma	$Agamma$	area related body effect mismatch parameter		0	0
Iba	Iba	first impact ionization coefficient	$1/m$	$2e8$	0.0
Ibb	Ibb	second impact ionization coefficient	V/m	$3.5e8$	$3.5e8$

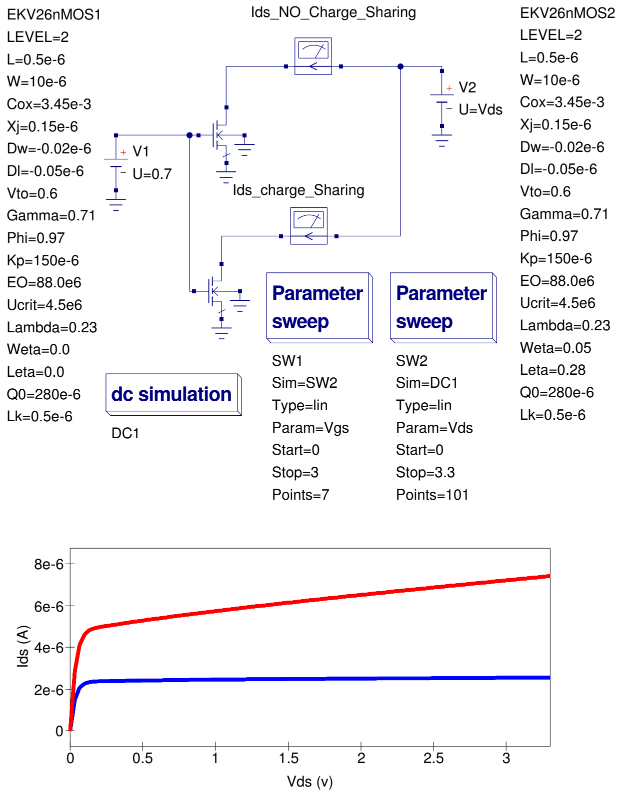


Figure 9: Test circuit for simulating EKV v2.6 charge sharing effects in short channel devices: *Ids* blue curve; NO charge sharing ($Weta = 0.0$, $Leta = 0.0$), *Ids* red curve; charge sharing ($Weta = 0.05$, $Leta = 0.28$)

Name	Symbol	Description	Unit	Default nMOS	Default pMOS
Ibn	Ibn	saturation voltage factor for impact ionization		1.0	1.0
Tnom	$Tnom$	parameter measurement temperature	$^\circ C$	26.85	26.85
Temp	$Temp$	device temperature	$^\circ C$	26.85	26.85

Simulating short channel charge sharing effects

A simple test circuit for demonstrating the effects of charge sharing is given in Figure 9. With charge sharing disabled, by setting Weta and Leta to zero, the magnitude and slope of the *Ids* vs. *Vds* curve shows a marked difference to that where charge sharing is enabled. One point to note with this test: charge sharing in short channel devices significantly reduces the device output resistance which could have, of course, important consequences on circuit performance.

End note

This report outlines some of the background to the Qucs implementation of the EKV v2.6 MOSFET model. A series of test results demonstrate a range of results that have been achieved with this new Qucs compact device model. Although the test results give data similar to what is expected in all cases it must be stressed that this is the first release of this MOSFET model and as such it will probably contain bugs. A great deal of work has gone into providing this new Qucs model. However, all the effort has been worthwhile because for the first time Qucs now has a submicron MOSFET model. Please use the model and report bugs to the Qucs development team. Much work still remains to be done in the development of MOSFET models for Qucs. In future releases both bug fixes and new models are likely to feature strongly. Once again I would like to thank Stefan Jahn and Władysław Grabiński (of MOS-AK) for their encouragement and support during the period I have been working on developing the Qucs implementation of the EKV v2.6 model and writing this report.

Qucs Verilog-A code for the EKV v2.6 MOSFET model

nMOS: EKV equation numbers are given on the right-hand side of code lines

```
//    Qucs EPFL-EKV 2.6 nMOS model:
//
//    The structure and theoretical background to the EKV 2.6
//    Verilog-a model is presented in the Qucs EPFL-EKV 2.6 report.
//    Typical parameters are for 0.5um CMOS (C) EPLFL-LEG 1999.
//    Geometry range: Short channel W>= 0.8um, L >= 0.5um
//                    Long channel  W>= 2um,    L >= 2um
//    Voltage range:  |Vgb| < 3.3V, |Vdb| < 3.3V, |Vsb| < 2V
//
//    This is free software; you can redistribute it and/or modify
//    it under the terms of the GNU General Public License as published by
//    the Free Software Foundation; either version 2, or (at your option)
//    any later version.
//
//    Copyright (C), Mike Brinson, mbrin72043@yahoo.co.uk, May 2008.
//
‘include ”disciplines.vams”
‘include ”constants.vams”

//
 module EKV26nMOS (Drain, Gate, Source, Bulk);
 inout Drain, Gate, Source, Bulk;
electrical Drain, Gate, Source, Bulk;
// Internal nodes
electrical Drain_int, Source_int;
‘define attr(txt) (*txt*)
// Device dimension parameters
 parameter real LEVEL = 1 from [1 : 2]
        ‘attr(info=”long␣=␣1,␣short␣=␣2”);
 parameter real L = 0.5e-6 from [0.0 : inf]
        ‘attr(info=”length␣parameter” unit = ”m” );
 parameter real W = 10e-6 from [0.0 : inf]
        ‘attr(info=”Width␣parameter” unit = ”m”);
 parameter real Np = 1.0 from [1.0 : inf]
        ‘attr(info=”parallel␣multiple␣device␣number”);
 parameter real Ns = 1.0 from [1.0 : inf]
        ‘attr(info=”series␣multiple␣device␣number”);
// Process parameters
```

```
 parameter real Cox = 3.45e-3 from [0 : inf]
        `attr(info="gate␣oxide␣capacitance␣per␣unit␣area" unit = "F/m**2" );
 parameter real Xj = 0.15e-6 from [0.01e-6 : 1.0e-6]
        `attr(info="metallurgical␣junction␣depth" unit = "m");
 parameter real Dw = -0.02e-6 from [-inf : 0.0]
        `attr(info="channel␣width␣correction" unit = "m");
 parameter real Dl = -0.05e-6 from [-inf : 0.0]
        `attr(info="channel␣length␣correction" unit = "m");
// Basic intrinsic model parameters
 parameter real Vto = 0.6 from [1e-6 : 2.0]
        `attr(info="long␣channel␣threshold␣voltage" unit="V" );
 parameter real Gamma = 0.71 from [0.0 : 2.0]
        `attr(info="body␣effect␣parameter" unit="V**(1/2)");
 parameter real Phi = 0.97 from [0.3 : 2.0]
        `attr(info="bulk␣Fermi␣potential" unit="V");
 parameter real Kp = 150e-6 from [10e-6 : inf]
        `attr(info="transconductance␣parameter" unit = "A/V**2");
 parameter real Theta = 50e-3 from [0.0 : inf]
        `attr(info="mobility␣reduction␣coefficient" unit = "1/V");
 parameter real EO = 88.0e6 from [1.0e6 : inf]
        `attr(info="mobility␣coefficient" unit="V/m");
 parameter real Ucrit = 4.5e6 from [2.0e6 : 25.0e6]
        `attr(info="longitudinal␣critical␣field" unit="V/m");
// Channel length and charge sharing parameters
 parameter real Lambda = 0.23 from [0.1 : inf]
        `attr(info="depletion␣length␣coefficient");
 parameter real Weta = 0.05 from [0.0 : inf]
        `attr(info="narrow-channel␣effect␣coefficient");
 parameter real Leta = 0.28 from [0.0 : inf]
        `attr(info="longitudinal␣critical␣field");
// Reverse short channel effect parameters
 parameter real Q0 = 280e-6 from [0.0 : inf]
        `attr(info="reverse␣short␣channel␣charge␣density" unit="A*s/m**2");
 parameter real Lk = 0.5e-6 from [0.0 : inf]
        `attr(info="characteristic␣length" unit="m");
// Intrinsic model temperature parameters
 parameter real Tcv = 1.5e-3
        `attr(info="threshold␣voltage␣temperature␣coefficient" unit="V/K");
 parameter real Bex = -1.5
        `attr(info="mobility␣temperature␣coefficient");
 parameter real Ucex = 1.7
        `attr(info="Longitudinal␣critical␣field␣temperature␣exponent");
 parameter real Ibbt = 0.0
        `attr(info="Ibb␣temperature␣coefficient"  unit="1/K");
// Series resistance calculation parameters
 parameter real Hdif = 0.9e-6 from [0.0 :inf]
        `attr(info="heavily␣doped␣diffusion␣length" unit = "m");
 parameter real Rsh = 510.0 from [0.0 : inf]
        `attr(info="drain/source␣diffusion␣sheet␣resistance" unit="Ohm/square");
 parameter real Rsc = 0.0 from [0.0 : inf]
        `attr(info="source␣contact␣resistance" unit="Ohm");
 parameter real Rdc = 0.0 from [0.0 : inf]
        `attr(info="drain␣contact␣resistance" unit="Ohm");
// Gate overlap capacitances
 parameter real Cgso = 1.5e-10 from [0.0 :inf]
        `attr(info="gate␣to␣source␣overlap␣capacitance" unit = "F/m");
 parameter real Cgdo = 1.5e-10 from [0.0 : inf]
        `attr(info="gate␣to␣drain␣overlap␣capacitance"  unit=  "F/m");
 parameter real Cgbo = 4.0e-10 from [0.0 : inf]
        `attr(info="gate␣to␣bulk␣overlap␣capacitance"   unit=  "F/m");
// Impact ionization related parameters
 parameter real Iba = 2e8 from [0.0 :inf]
        `attr(info="first␣impact␣ionization␣coefficient" unit = "1/m");
 parameter real Ibb = 3.5e8 from [1.0e8 : inf]
        `attr(info="second␣impact␣ionization␣coefficient" unit="V/m");
```

```
 parameter real Ibn = 1.0 from [0.1 : inf]
        `attr(info="saturation␣voltage␣factor␣for␣impact␣ionization");
// Flicker noise parameters
 parameter real Kf = 1.0e-27 from [0.0 :inf]
        `attr(info="flicker␣noise␣coefficient");
 parameter real Af = 1.0 from [0.0 : inf]
        `attr(info="flicker␣noise␣exponent" );
// Matching parameters
 parameter real Avto = 0.0 from [0.0 :inf]
        `attr(info="area␣related␣theshold␣voltage␣mismatch␣parameter" unit = "V*m");
 parameter real Akp = 0.0 from [0.0 : inf]
        `attr(info="area␣related␣gain␣mismatch␣parameter" unit="m");
 parameter real Agamma = 0.0 from [0.0 : inf]
        `attr(info="area␣related␣body␣effect␣mismatch␣parameter" unit="sqrt(V)*m");
// Diode parameters
 parameter real N=1.0 from [1e-6:inf]
        `attr(info="emission␣coefficient");
 parameter real Is=1e-14 from [1e-20:inf]
        `attr(info="saturation␣current" unit="A" );
 parameter real Bv=100 from [1e-6:inf]
        `attr(info="reverse␣breakdown␣voltage" unit="V");
 parameter real Ibv=1e-3 from [1e-6:inf]
        `attr(info="current␣at␣reverse␣breakdown␣voltage" unit="A");
 parameter real Vj=1.0 from [1e-6:inf]
        `attr(info="junction␣potential" unit="V");
 parameter real Cj0=300e-15 from [0:inf]
        `attr(info="zero-bias␣junction␣capacitance" unit="F");
 parameter real M=0.5 from [1e-6:inf]
        `attr(info="grading␣coefficient");
 parameter real Area=1.0 from [1e-3:inf]
        `attr(info="diode␣relative␣area");
 parameter real Fc=0.5 from [1e-6:inf]
        `attr(info="forward-bias␣depletion␣capcitance␣coefficient");
 parameter real Tt=0.1e-9 from [1e-20:inf]
        `attr(info="transit␣time" unit="s" );
 parameter real Xti=3.0 from [1e-6:inf]
        `attr(info="saturation␣current␣temperature␣exponent");
// Temperature parameters
 parameter real Tnom = 26.85
        `attr(info="parameter␣measurement␣temperature" unit = "Celsius");
// Local variables
real epsilonsi, epsilonox, Tnomk, T2, Tratio, Vto_T, Ucrit_T, Egnom, Eg, Phi_T;
real Weff, Leff, RDeff, RSeff, con1, con2, Vtoa, Kpa,Kpa_T,Gammaa, C_epsilon, xi;
real nnn, deltaV_RSCE, Vg, Vs, Vd, Vgs, Vgd, Vds, Vdso2, VG, VS, VD;
real VGprime, VP0, VSprime, VDprime, Gamma0, Gammaprime, Vp;
real n, X1, iff, X2, ir, Vc, Vdss, Vdssprime, deltaV, Vip;
real Lc, DeltaL, Lprime, Lmin, Leq, X3, irprime, Beta0, eta;
real Qb0, Beta0prime, nq, Xf, Xr, qD, qS, qI, qB, Beta, Ispecific, Ids, Vib, Idb, Ibb_T;
real A, B, Vt_T2, Eg_T1, Eg_T2, Vj_T2, Cj0_T2, F1, F2, F3, Is_T2;
real Id1, Id2, Id3, Id4, Is1, Is2, Is3, Is4, V1, V2, Ib_d, Ib_s, Qd, Qs, Qd1, Qd2, Qs1, Qs2;
real qb, qg, qgso, qgdo, qgbo, fourkt, Sthermal, gm, Sflicker, StoDswap, p_n_MOS;
//
analog begin
// Equation initialization
p_n_MOS = 1.0; // nMOS
A=7.02e-4;
B=1108.0;
epsilonsi = 1.0359e-10;       // Eqn 4
epsilonox = 3.453143e-11;     // Eqn 5
Tnomk = Tnom+273.15;                  // Eqn 6
T2=$temperature;
Tratio = T2/Tnomk;
Vto_T = Vto-Tcv*(T2-Tnomk);
Egnom = 1.16-0.000702*Tnomk*Tnomk/(Tnomk+1108);
Eg =     1.16-0.000702*T2*T2/(T2+1108);
```

```
Phi_T = Phi*Tratio - 3.0*$vt*ln(Tratio)-Egnom*Tratio+Eg;
Ibb_T = Ibb*(1.0+Ibbt*(T2 -Tnomk));
Weff = W + Dw;     // Eqn 25
Leff = L + Dl;     // Eqn 26
RDeff = ( (Hdif*Rsh)/Weff)/Np + Rdc;
RSeff = ( (Hdif*Rsh)/Weff)/Np + Rsc;
con1 = sqrt(Np*Weff*Ns*Leff);
Vt_T2=`P_K*T2/`P_Q;
Eg_T1=Eg-A*Tnomk*Tnomk/(B+Tnomk);
Eg_T2=Eg-A*T2*T2/(B+T2);
Vj_T2=(T2/Tnomk)*Vj-(2*Vt_T2)*ln(pow((T2/Tnomk),1.5))-((T2/Tnomk)*Eg_T1-Eg_T2);
Cj0_T2=Cj0*(1+M*(400e-6*(T2-Tnomk)-(Vj_T2-Vj)/Vj));
F1=(Vj/(1-M))*(1-pow((1-Fc),(1-M)));
F2=pow((1-Fc), (1+M));
F3=1-Fc*(1+M);
Is_T2=Is*pow( (T2/Tnomk), (Xti/N))*limexp((-Eg_T1/Vt_T2)*(1-T2/Tnomk));
con2 = (Cox*Ns*Np*Weff*Leff);
fourkt = 4.0*`P_K*T2;
//
if (LEVEL == 2)
 begin
   Ucrit_T = Ucrit*pow(Tratio, Ucex);
   Vtoa = Vto+Avto/con1;                 // Eqn 27
   Kpa = Kp*(1.0+Akp/con1);              // Eqn 28
   Kpa_T = Kpa*pow( Tratio, Bex);        // Eqn 18
   Gammaa = Gamma+Agamma/con1;           // Eqn 29
   C_epsilon = 4.0*pow(22e-3, 2);        // Eqn 30
   xi = 0.028*(10.0*(Leff/Lk)-1.0);      // Eqn 31
   nnn = 1.0+0.5*(xi+ sqrt(pow(xi,2) + C_epsilon));
   deltaV_RSCE = (2.0*Q0/Cox)*(1.0/pow(nnn,2)); // Eqn 32
 end
//
// Model branch and node voltages
//

Vg = p_n_MOS*V(Gate, Bulk);
Vs = p_n_MOS*V(Source, Bulk);
Vd = p_n_MOS*V(Drain, Bulk);
VG=Vg;         // Eqn 22
if ( (Vd-Vs) >= 0.0)
        begin
          StoDswap = 1.0;
          VS=Vs;    // Eqn 23
          VD=Vd;    // Eqn 24
        end
else
        begin
          StoDswap = -1.0;
          VD=Vs;
          VS=Vd;
        end
if (LEVEL == 2)
    VGprime=VG-Vto_T-deltaV_RSCE+Phi_T+Gamma*sqrt(Phi_T);   // Eqn 33  nMOS equation
else
    VGprime=Vg-Vto_T+Phi_T+Gamma*sqrt(Phi_T);

//
if (LEVEL == 2)
 begin
   if (VGprime > 0)
        VP0=VGprime-Phi_T-Gammaa*(sqrt(VGprime+(Gammaa/2.0)*(Gammaa/2.0))
              -(Gammaa/2.0));  // Eqn 34
   else
        VP0 = -Phi_T;
   VSprime=0.5*(VS+Phi_T+sqrt(pow( (VS+Phi_T),2) + pow( (4.0*$vt),2)));  // Eqn 35
```

```
  VDprime=0.5*(VD+Phi_T+sqrt(pow( (VD+Phi_T),2) + pow( (4.0*$vt),2))); // Eqn 35
  Gamma0=Gammaa-(epsilonsi/Cox)*((Leta/Leff)*(sqrt(VSprime)+sqrt(VDprime))
              -(3.0*Weta/Weff)*sqrt(VP0+Phi_T)); // Eqn 36
  Gammaprime = 0.5*(Gamma0+sqrt( pow(Gamma0,2) +0.1*$vt )); // Eqn 37
  if (VGprime > 0.0 )
           Vp = VGprime-Phi_T-Gammaprime*(sqrt(VGprime+(Gammaprime/2.0)*
                (Gammaprime/2.0)) - (Gammaprime/2.0)); // Eqn 38
  else
        Vp = -Phi_T;
  n = 1.0 +Gammaa/(2.0*sqrt(Vp+Phi_T+4.0*$vt)); // Eqn 39
 end
else
  begin
   if (VGprime > 0)
        Vp=VGprime-Phi_T-Gamma*(sqrt(VGprime+(Gamma/2.0)*(Gamma/2.0))
           -(Gamma/2.0)); // Eqn 34
   else
        Vp = -Phi_T;
  n = 1.0 +Gamma/(2.0*sqrt(Vp+Phi_T+4.0*$vt)); // Eqn 39
end
//
X1 = (Vp-VS)/$vt;
iff = ln(1.0+limexp(X1/2.0))*ln(1.0+limexp(X1/2.0)); // Eqn 44
X2 = (Vp-VD)/$vt;
ir = ln(1.0+limexp(X2/2.0))*ln(1.0+limexp(X2/2.0)); // Eqn 57
//
if (LEVEL == 2)
 begin
   Vc = Ucrit_T*Ns*Leff; // Eqn 45
   Vdss = Vc*(sqrt( 0.25 + (($vt/(Vc))*sqrt(iff)))-0.5); // Eqn 46;
   Vdssprime = Vc*(sqrt( 0.25 + ($vt/Vc)*(sqrt(iff)-0.75*ln(iff))) - 0.5)
              +$vt*(ln(Vc/(2.0*$vt)) - 0.6 ); // Eqn 47
   if (Lambda*(sqrt(iff) > (Vdss/$vt) ) )
          deltaV = 4.0*$vt*sqrt(Lambda*(sqrt(iff) -(Vdss/$vt))
                   + (1.0/64.0) ); // Eqn 48
   else
          deltaV = 1.0/64.0;
   Vdso2 = (VD-VS)/2.0; // Eqn 49
   Vip = sqrt( pow(Vdss, 2) + pow( deltaV,2)) - sqrt( pow( (Vdso2 - Vdss), 2)
        + pow (deltaV, 2)); // Eqn 50
   Lc = sqrt( (epsilonsi/Cox)*Xj); // Eqn 51
   DeltaL = Lambda*Lc*ln(1.0+((Vdso2-Vip)/(Lc*Ucrit_T))); // Eqn 52
   Lprime = Ns*Leff - DeltaL + ( (Vdso2+Vip)/Ucrit_T); // Eqn 53
   Lmin = Ns*Leff/10.0; // Eqn 54
   Leq = 0.5*(Lprime + sqrt( pow(Lprime, 2) + pow(Lmin, 2))); // Eqn 55
   X3 = (Vp-Vdso2-VS-sqrt( pow(Vdssprime, 2) + pow( deltaV, 2))
        + sqrt( pow( (Vdso2-Vdssprime), 2) + pow(deltaV,2)))/$vt;
   irprime = ln(1.0+limexp(X3/2.0))*ln(1.0+limexp(X3/2.0)); // Eqn 56
   Beta0 = Kpa_T*(Np*Weff/Leq); // Eqn 58
   eta = 0.5; // Eqn 59 - nMOS
   Qb0 = Gammaa*sqrt(Phi_T); // Eqn 60;
   Beta0prime = Beta0*(1.0 +(Cox/(EO*epsilonsi))*Qb0); // Eqn 61
   nq = 1.0 +Gammaa/(2.0*sqrt(Vp+Phi_T+1e-6)); // Eqn 69
 end
else
 nq = 1.0 +Gamma/(2.0*sqrt(Vp+Phi_T+1e-6)); // Eqn 69
//
Xf = sqrt(0.25+iff); // Eqn 70
Xr = sqrt(0.25+ir); // Eqn 71
qD = -nq*( (4.0/15.0)*((3.0*pow( Xr,3) + 6.0*pow( Xr, 2)*Xf + 4.0*Xr*pow( Xf, 2)
      + 2.0*pow(Xf, 3))/(pow( (Xf+Xr), 2) ) ) -0.5); // Eqn 72
qS = -nq*( (4.0/15.0)*((3.0*pow( Xf,3) + 6.0*pow( Xf, 2)*Xr + 4.0*Xf*pow( Xr, 2)
      + 2.0*pow(Xr, 3))/(pow( (Xf+Xr), 2) ) ) -0.5); // Eqn 73
qI = -nq*( (4.0/3.0)*( (pow(Xf,2)+(Xf*Xr)+pow(Xr,2))/(Xf+Xr)) - 1.0); // Eqn 74
if (LEVEL == 2)
```

```
  if (VGprime > 0)
        qB = (-Gammaa*sqrt(Vp+Phi_T+1e-6))*(1.0/$vt) - ( (nq-1.0)/nq)*qI;  // Eqn 75
  else
        qB = -VGprime/$vt;
else
    if (VGprime > 0)
        qB = (-Gamma*sqrt(Vp+Phi_T+1e-6))*(1.0/$vt) - ( (nq-1.0)/nq)*qI;  // Eqn 75
  else
        qB = -VGprime/$vt;
//
if (LEVEL == 2)
   Beta = Beta0prime/(1.0 + (Cox/ (EO*epsilonsi))*$vt*abs(qB+eta*qI));      // Eqn 62
else
   Beta = Kp*(Weff/Leff)/(1+Theta*Vp);
//
Ispecific = 2.0*n*Beta*pow( $vt, 2);    // Eqn 65
//
if (LEVEL == 2)
  begin
   Ids = Ispecific*(iff-irprime);          // Eqn 66
   Vib = VD-VS-Ibn*2.0*Vdss;               // Eqn 67
   if ( Vib > 0.0)
    Idb = Ids*(Iba/Ibb_T)*Vib*exp( (-Ibb_T*Lc)/Vib);   // Eqn 68
   else
   Idb = 0.0;
  end
else
   Ids = Ispecific*(iff-ir);          // Eqn 66
//
Sthermal = fourkt*Beta*abs(qI);
gm = Beta*$vt*(sqrt( (4.0*iff/Ispecific) +1.0) - sqrt( (4.0*ir/Ispecific) + 1.0) );
Sflicker = (Kf*gm*gm)/(Np*Weff*Ns*Leff*Cox);
//
qb = con2*$vt*qB;
qg = con2*$vt*(-qI-qB);
qgso = Cgso*Weff*Np*(VG-VS);
qgdo = Cgdo*Weff*Np*(VG-VD);
qgbo = Cgbo*Leff*Np*VG;
// Drain and source diodes
if (StoDswap > 0.0)
     begin
        V1=p_n_MOS*V(Bulk, Drain_int);
        V2=p_n_MOS*V(Bulk, Source_int);
     end
else
     begin
        V2=p_n_MOS*V(Bulk, Drain_int);
        V1=p_n_MOS*V(Bulk, Source_int);
     end
Id1= (V1>-5.0*N*$vt) ? Area*Is_T2*(limexp( V1/(N*Vt_T2) )-1.0) : 0;
Qd1=(V1<Fc*Vj)? Tt*Id1+Area*(Cj0_T2*Vj_T2/(1-M))*(1-pow((1-V1/Vj_T2),(1-M))):0;
Id2=(V1<=-5.0*N*$vt) ? -Area*Is_T2 : 0;
Qd2=(V1>=Fc*Vj)? Tt*Id1+Area*Cj0_T2*(F1+(1/F2)*(F3*(V1-Fc*Vj_T2)+(M/(2.0*Vj_T2))
     *(V1*V1-Fc*Fc*Vj_T2*Vj_T2))):0;
Id3=(V1 == -Bv) ? -Ibv  : 0 ;
Id4=(V1<-Bv) ?-Area*Is_T2*(limexp(-(Bv+V1)/Vt_T2)-1.0+Bv/Vt_T2) : 0;
Ib_d = Id1+Id2+Id3+Id4;
Qd = Qd1+Qd2;
//
Is1= (V2>-5.0*N*$vt) ? Area*Is_T2*(limexp( V2/(N*Vt_T2) )-1.0) : 0;
Qs1=(V2<Fc*Vj)? Tt*Is1+Area*(Cj0_T2*Vj_T2/(1-M))*(1-pow((1-V2/Vj_T2),(1-M))):0;
Is2=(V2<=-5.0*N*$vt) ? -Area*Is_T2 : 0;
Qs2=(V2>=Fc*Vj)? Tt*Is1+Area*Cj0_T2*(F1+(1/F2)*(F3*(V2-Fc*Vj_T2)+(M/(2.0*Vj_T2))
     *(V2*V2-Fc*Fc*Vj_T2*Vj_T2))):0;
Is3=(V2 == -Bv) ? -Ibv  : 0 ;
```

```
Is4=(V2<-Bv) ?-Area*Is_T2*(limexp(-(Bv+V2)/Vt_T2)-1.0+Bv/Vt_T2) : 0;
Ib_s = Is1+Is2+Is3+Is4;
Qs = Qs1+Qs2;
// Current and noise contributions
if ( StoDswap > 0.0)
 begin
      if (RDeff > 0.0)
         I(Drain, Drain_int) <+ V(Drain, Drain_int)/RDeff;
      else
         I(Drain, Drain_int) <+ V(Drain, Drain_int)/1e-7;
      if (RSeff > 0.0)
         I(Source, Source_int) <+ V(Source, Source_int)/RSeff;
      else
         I(Source, Source_int) <+ V(Source, Source_int)/1e-7;
    I(Drain_int, Source_int) <+ p_n_MOS*Ids;
    if (LEVEL == 2)
        I(Drain_int, Bulk) <+ p_n_MOS*Idb;
    I(Gate, Drain_int) <+ p_n_MOS*0.5*ddt(qg);
    I(Gate, Source_int) <+ p_n_MOS*0.5*ddt(qg);
    I(Drain_int, Bulk) <+ p_n_MOS*0.5*ddt(qb);
    I(Source_int, Bulk) <+ p_n_MOS*0.5*ddt(qb);
    I(Gate, Source_int) <+ p_n_MOS*ddt(qgso);
    I(Gate, Drain_int)  <+ p_n_MOS*ddt(qgdo);
    I(Gate, Bulk)    <+ p_n_MOS*ddt(qgbo);
    I(Bulk, Drain_int) <+ p_n_MOS*Ib_d;
    I(Bulk, Drain_int) <+ p_n_MOS*ddt(Qd);
    I(Bulk, Source_int) <+ p_n_MOS*Ib_s;
    I(Bulk, Source_int) <+ p_n_MOS*ddt(Qs);
    I(Drain_int, Source_int) <+ white_noise(Sthermal,"thermal");
    I(Drain_int, Source_int) <+ flicker_noise(Sflicker, Af, "flicker");
    I(Drain, Drain_int) <+ white_noise(fourkt/RDeff, "thermal");
    I(Source, Source_int) <+ white_noise(fourkt/RSeff, "thermal");
end
else
 begin
      if (RSeff > 0.0)
         I(Drain, Drain_int) <+ V(Drain, Drain_int)/RSeff;
      else
         I(Drain, Drain_int) <+ V(Drain, Drain_int)/1e-7;
      if (RDeff > 0.0)
         I(Source, Source_int) <+ V(Source, Source_int)/RDeff;
      else
         I(Source, Source_int) <+ V(Source, Source_int)/1e-7;
    I( Source_int, Drain_int) <+ p_n_MOS*Ids;
    if (LEVEL == 2)
        I(Source_int, Bulk) <+ p_n_MOS*Idb;
    I( Gate, Source_int) <+ p_n_MOS*0.5*ddt(qg);
    I( Gate, Drain_int) <+ p_n_MOS*0.5*ddt(qg);
    I( Source_int, Bulk) <+ p_n_MOS*0.5*ddt(qb);
    I( Drain_int, Bulk) <+ p_n_MOS*0.5*ddt(qb);
    I( Gate, Drain_int) <+ p_n_MOS*ddt(qgso);
    I( Gate, Source_int)  <+ p_n_MOS*ddt(qgdo);
    I( Gate, Bulk)    <+ p_n_MOS*ddt(qgbo);
    I( Bulk, Source_int) <+ p_n_MOS*Ib_d;
    I( Bulk, Source_int) <+ p_n_MOS*ddt(Qd);
    I( Bulk, Drain_int) <+ p_n_MOS*Ib_s;
    I( Bulk, Drain_int) <+ p_n_MOS*ddt(Qs);
    I( Source_int, Drain_int) <+ white_noise(Sthermal,"thermal");
    I( Source_int, Drain_int) <+ flicker_noise(Sflicker, Af, "flicker");
    I( Source_int, Source) <+ white_noise(fourkt/RDeff, "thermal");
    I( Drain_int, Drain) <+ white_noise(fourkt/RSeff, "thermal");
 end
end
endmodule
```

pMOS: EKV equation numbers are given on the right-hand side of code lines

```
//   Qucs EPFL-EKV 2.6 pMOS model:
//
//   The structure and theoretical background to the EKV 2.6
//   Verilog-a model is presented in the Qucs EPL-EKV 2.6 report.
//   Typical parameters are for 0.5um CMOS (C) EPLFL-LEG 1999.
//   Geometry range: Short channel: W>= 0.8um, L >= 0.5um
//                   Long channel:  W>= 2um,    L >= 2um
//   Voltage range:  |Vgb| < 3.3V, |Vdb| < 3.3V, |Vsb| < 2V
//
//   This is free software; you can redistribute it and/or modify
//   it under the terms of the GNU General Public License as published by
//   the Free Software Foundation; either version 2, or (at your option)
//   any later version.
//
//   Copyright (C), Mike Brinson, mbrin72043@yahoo.co.uk, May 2008.
//
‘include ”disciplines.vams”
‘include ”constants.vams”

//
 module EKV26pMOS (Drain, Gate, Source, Bulk);
 inout Drain, Gate, Source, Bulk;
electrical Drain, Gate, Source, Bulk;
// Internal nodes
electrical Drain_int, Source_int;
‘define attr(txt) (*txt*)
// Device dimension parameters
 parameter real LEVEL = 1 from [1 : 2]
        ‘attr(info=”long␣=␣1,␣short␣=␣2”);
 parameter real L = 0.5e-6 from [0.0 : inf]
        ‘attr(info=”length␣parameter” unit = ”m” );
 parameter real W = 10e-6 from [0.0 : inf]
        ‘attr(info=”Width␣parameter” unit = ”m”);
 parameter real Np = 1.0 from [1.0 : inf]
        ‘attr(info=”parallel␣multiple␣device␣number”);
 parameter real Ns = 1.0 from [1.0 : inf]
        ‘attr(info=”series␣multiple␣device␣number”);
// Process parameters
 parameter real Cox = 3.45e-3 from [0 : inf]
        ‘attr(info=”gate␣oxide␣capacitance␣per␣unit␣area” unit = ”F/m**2” );
 parameter real Xj = 0.15e-6 from [0.01e-6 : 1.0e-6]
        ‘attr(info=”metallurgical␣junction␣depth” unit = ”m”);
 parameter real Dw = -0.03e-6 from [-inf : 0.0]
        ‘attr(info=”channel␣width␣correction” unit = ”m”);
 parameter real Dl = -0.05e-6 from [-inf : 0.0]
        ‘attr(info=”channel␣length␣correction” unit = ”m”);
// Basic intrinsic model parameters
 parameter real Vto = -0.55 from [-inf : -1e-6]
        ‘attr(info=”long␣channel␣threshold␣voltage” unit=”V” );
 parameter real Gamma = 0.69 from [0.0 : 2.0]
        ‘attr(info=”body␣effect␣parameter” unit=”V**(1/2)”);
 parameter real Phi = 0.87 from [0.3 : 2.0]
        ‘attr(info=”bulk␣Fermi␣potential” unit=”V”);
 parameter real Kp = 35e-6 from [10e-6 : inf]
        ‘attr(info=”transconductance␣parameter” unit = ”A/V**2”);
 parameter real Theta = 50e-3 from [0.0 : inf]
        ‘attr(info=”mobility␣reduction␣coefficient” unit = ”1/V”);
 parameter real EO = 51.0e6 from [1.0e6 : inf]
        ‘attr(info=”mobility␣coefficient” unit=”V/m”);
 parameter real Ucrit = 18.0e6 from [2.0e6 : 25.0e6]
```

```
        ‘attr(info="longitudinal␣critical␣field" unit="V/m");
// Channel length and charge sharing parameters
 parameter real Lambda = 1.1 from [0.1 : inf]
        ‘attr(info="depletion␣length␣coefficient");
 parameter real Weta = 0.0  from [0.0 : inf]
        ‘attr(info="narrow-channel␣effect␣coefficient");
 parameter real Leta = 0.45 from [0.0 : inf]
        ‘attr(info="longitudinal␣critical␣field");
// Reverse short channel effect parameters
 parameter real Q0 = 200e-6 from [0.0 : inf]
        ‘attr(info="reverse␣short␣channel␣charge␣density" unit="A*s/m**2");
 parameter real Lk = 0.6e-6 from [0.0 : inf]
        ‘attr(info="characteristic␣length" unit="m");
// Intrinsic model temperature parameters
 parameter real Tcv = -1.4e-3
        ‘attr(info="threshold␣voltage␣temperature␣coefficient" unit="V/K");
 parameter real Bex = -1.4
        ‘attr(info="mobility␣temperature␣coefficient");
 parameter real Ucex = 2.0
        ‘attr(info="Longitudinal␣critical␣field␣temperature␣exponent");
 parameter real Ibbt = 0.0
        ‘attr(info="Ibb␣temperature␣coefficient" unit = "1/K");
// Series resistance calculation parameters
 parameter real Hdif = 0.9e-6 from [0.0 :inf]
        ‘attr(info="heavily␣doped␣diffusion␣length" unit = "m");
 parameter real Rsh = 990.0 from [0.0 : inf]
        ‘attr(info="drain/source␣diffusion␣sheet␣resistance" unit="Ohm/square");
 parameter real Rsc = 0.0 from [0.0 : inf]
        ‘attr(info="source␣contact␣resistance" unit="Ohm");
 parameter real Rdc = 0.0 from [0.0 : inf]
        ‘attr(info="drain␣contact␣resistance" unit="Ohm");
// Gate overlap capacitances
 parameter real Cgso = 1.5e-10 from [0.0 :inf]
        ‘attr(info="gate␣to␣source␣overlap␣capacitance" unit = "F/m");
 parameter real Cgdo = 1.5e-10 from [0.0 : inf]
        ‘attr(info="gate␣to␣drain␣overlap␣capacitance"  unit=  "F/m");
 parameter real Cgbo = 4.0e-10 from [0.0 : inf]
        ‘attr(info="gate␣to␣bulk␣overlap␣capacitance"   unit=  "F/m");
// Impact ionization related parameters
 parameter real Iba = 0.0 from [0.0 :inf]
        ‘attr(info="first␣impact␣ionization␣coefficient" unit = "1/m");
 parameter real Ibb = 3.0e8 from [1.0e8 : inf]
        ‘attr(info="second␣impact␣ionization␣coefficient" unit="V/m");
 parameter real Ibn = 1.0 from [0.1 : inf]
        ‘attr(info="saturation␣voltage␣factor␣for␣impact␣ionization");
// Flicker noise parameters
 parameter real Kf = 1.0e-28 from [0.0 :inf]
        ‘attr(info="flicker␣noise␣coefficient");
 parameter real Af = 1.0 from [0.0 : inf]
        ‘attr(info="flicker␣noise␣exponent" );
// Matching parameters
 parameter real Avto = 0.0 from [0.0 :inf]
        ‘attr(info="area␣related␣theshold␣voltage␣mismatch␣parameter" unit = "V*m");
 parameter real Akp = 0.0 from [0.0 : inf]
        ‘attr(info="area␣related␣gain␣mismatch␣parameter" unit="m");
 parameter real Agamma = 0.0 from [0.0 : inf]
        ‘attr(info="area␣related␣body␣effect␣mismatch␣parameter" unit="sqrt(V)*m");
// Diode parameters
 parameter real N=1.0 from [1e-6:inf]
        ‘attr(info="emission␣coefficient");
 parameter real Is=1e-14 from [1e-20:inf]
        ‘attr(info="saturation␣current" unit="A" );
 parameter real Bv=100 from [1e-6:inf]
        ‘attr(info="reverse␣breakdown␣voltage" unit="V");
 parameter real Ibv=1e-3 from [1e-6:inf]
```

```
        ‘attr(info=”current␣at␣reverse␣breakdown␣voltage” unit=”A”);
 parameter real Vj=1.0 from [1e-6:inf]
        ‘attr(info=”junction␣potential” unit=”V”);
 parameter real Cj0=300e-15 from [0:inf]
        ‘attr(info=”zero-bias␣junction␣capacitance” unit=”F”);
 parameter real M=0.5 from [1e-6:inf]
        ‘attr(info=”grading␣coefficient”);
 parameter real Area=1.0 from [1e-3:inf]
        ‘attr(info=”diode␣relative␣area”);
 parameter real Fc=0.5 from [1e-6:inf]
        ‘attr(info=”forward-bias␣depletion␣capcitance␣coefficient”);
 parameter real Tt=0.1e-9 from [1e-20:inf]
        ‘attr(info=”transit␣time” unit=”s” );
 parameter real Xti=3.0 from [1e-6:inf]
        ‘attr(info=”saturation␣current␣temperature␣exponent”);
// Temperature parameters
 parameter real Tnom = 26.85
        ‘attr(info=”parameter␣measurement␣temperature” unit = ”Celsius”);
// Local variables
real epsilonsi, epsilonox, Tnomk, T2, Tratio, Vto_T, Ucrit_T, Egnom, Eg, Phi_T;
real Weff, Leff, RDeff, RSeff, con1, con2, Vtoa, Kpa,Kpa_T,Gammaa, C_epsilon, xi;
real nnn, deltaV_RSCE, Vg, Vs, Vd, Vgs, Vgd, Vds, Vdso2, VG, VS, VD;
real VGprime, VP0, VSprime, VDprime, Gamma0, Gammaprime, Vp;
real n, X1, iff, X2, ir, Vc, Vdss, Vdssprime, deltaV, Vip;
real Lc, DeltaL, Lprime, Lmin, Leq, X3, irprime, Beta0, eta;
real Qb0, Beta0prime, nq, Xf, Xr, qD, qS, qI, qB, Beta, Ispecific, Ids, Vib, Idb, Ibb_T;
real A, B, Vt_T2, Eg_T1, Eg_T2, Vj_T2, Cj0_T2, F1, F2, F3, Is_T2;
real Id1, Id2, Id3, Id4, Is1, Is2, Is3, Is4, V1, V2, Ib_d, Ib_s, Qd, Qs, Qd1, Qd2, Qs1, Qs2;
real qb, qg, qgso, qgdo, qgbo, fourkt, Sthermal, gm, Sflicker, StoDswap, p_n_MOS;
//
analog begin
// Equation initialization
p_n_MOS = -1.0; // pMOS
A=7.02e-4;
B=1108.0;
epsilonsi = 1.0359e-10;      // Eqn 4
epsilonox = 3.453143e-11;    // Eqn 5
Tnomk = Tnom+273.15;               // Eqn 6
T2=$temperature;
Tratio = T2/Tnomk;
Vto_T = -Vto+Tcv*(T2-Tnomk); // Signs of Vto and Tcv changed for pMOS
Egnom = 1.16-0.000702*Tnomk*Tnomk/(Tnomk+1108);
Eg =     1.16-0.000702*T2*T2/(T2+1108);
Phi_T = Phi*Tratio - 3.0*$vt*ln(Tratio)-Egnom*Tratio+Eg;
Ibb_T = Ibb*(1.0+Ibbt*(T2 -Tnomk));
Weff = W + Dw;    // Eqn 25
Leff = L + Dl;    // Eqn 26
RDeff = ( (Hdif*Rsh)/Weff)/Np + Rdc;
RSeff = ( (Hdif*Rsh)/Weff)/Np + Rsc;
con1 = sqrt(Np*Weff*Ns*Leff);
Vt_T2=‘P_K*T2/‘P_Q;
Eg_T1=Eg-A*Tnomk*Tnomk/(B+Tnomk);
Eg_T2=Eg-A*T2*T2/(B+T2);
Vj_T2=(T2/Tnomk)*Vj-(2*Vt_T2)*ln(pow((T2/Tnomk),1.5))-((T2/Tnomk)*Eg_T1-Eg_T2);
Cj0_T2=Cj0*(1+M*(400e-6*(T2-Tnomk)-(Vj_T2-Vj)/Vj));
F1=(Vj/(1-M))*(1-pow((1-Fc),(1-M)));
F2=pow((1-Fc), (1+M));
F3=1-Fc*(1+M);
Is_T2=Is*pow( (T2/Tnomk), (Xti/N))*limexp((-Eg_T1/Vt_T2)*(1-T2/Tnomk));
con2 = (Cox*Ns*Np*Weff*Leff);
fourkt = 4.0*‘P_K*T2;
//
if (LEVEL == 2)
 begin
  Ucrit_T = Ucrit*pow(Tratio, Ucex);
```

```
  Vtoa = Vto+Avto/con1;                  // Eqn 27
  Kpa = Kp*(1.0+Akp/con1);               // Eqn 28
  Kpa_T = Kpa*pow( Tratio, Bex);         // Eqn 18
  Gammaa = Gamma+Agamma/con1;            // Eqn 29
  C_epsilon = 4.0*pow(22e-3, 2);         // Eqn 30
  xi = 0.028*(10.0*(Leff/Lk)-1.0);       // Eqn 31
  nnn = 1.0+0.5*(xi+ sqrt(pow(xi,2) + C_epsilon));
  deltaV_RSCE = (2.0*Q0/Cox)*(1.0/pow(nnn,2)); // Eqn 32
 end
//
// Model branch and node voltages
//

Vg = p_n_MOS*V(Gate, Bulk);
Vs = p_n_MOS*V(Source, Bulk);
Vd = p_n_MOS*V(Drain, Bulk);
VG=Vg;          // Eqn 22
if ( (Vd-Vs) >= 0.0)
        begin
          StoDswap = 1.0;
          VS=Vs;    // Eqn 23
          VD=Vd;    // Eqn 24
        end
else
        begin
          StoDswap = -1.0;
          VD=Vs;
          VS=Vd;
        end
if (LEVEL == 2)
    VGprime=VG-Vto_T-deltaV_RSCE+Phi_T+Gamma*sqrt(Phi_T);   // Eqn 33  nMOS equation
else
    VGprime=Vg-Vto_T+Phi_T+Gamma*sqrt(Phi_T);

if (LEVEL == 2)
 begin
  if (VGprime > 0)
        VP0=VGprime-Phi_T-Gammaa*(sqrt(VGprime+(Gammaa/2.0)*(Gammaa/2.0))-(Gammaa/2.0));   // Eqn 34
  else
        VP0 = -Phi_T;
  VSprime=0.5*(VS+Phi_T+sqrt(pow( (VS+Phi_T),2) + pow( (4.0*$vt),2)));   // Eqn 35
  VDprime=0.5*(VD+Phi_T+sqrt(pow( (VD+Phi_T),2) + pow( (4.0*$vt),2)));   // Eqn 35
  Gamma0=Gammaa-(epsilonsi/Cox)*((Leta/Leff)*(sqrt(VSprime)+sqrt(VDprime))
        -(3.0*Weta/Weff)*sqrt(VP0+Phi_T));   // Eqn 36
  Gammaprime = 0.5*(Gamma0+sqrt( pow(Gamma0,2) +0.1*$vt ));   // Eqn 37
  if (VGprime > 0.0 )
           Vp = VGprime-Phi_T-Gammaprime*(sqrt(VGprime+(Gammaprime/2.0)*(Gammaprime/2.0))
                - (Gammaprime/2.0)); // Eqn 38
  else
        Vp = -Phi_T;
  n = 1.0 +Gammaa/(2.0*sqrt(Vp+Phi_T+4.0*$vt));   // Eqn 39
 end
else
 begin
  if (VGprime > 0)
        Vp=VGprime-Phi_T-Gamma*(sqrt(VGprime+(Gamma/2.0)*(Gamma/2.0))-(Gamma/2.0));   // Eqn 34
  else
        Vp = -Phi_T;
  n = 1.0 +Gamma/(2.0*sqrt(Vp+Phi_T+4.0*$vt));   // Eqn 39
end
//
X1 = (Vp-VS)/$vt;
iff = ln(1.0+limexp(X1/2.0))*ln(1.0+limexp(X1/2.0));    // Eqn 44
X2 = (Vp-VD)/$vt;
ir = ln(1.0+limexp(X2/2.0))*ln(1.0+limexp(X2/2.0));     // Eqn 57
```

```
//
if (LEVEL == 2)
 begin
  Vc = Ucrit_T*Ns*Leff;  // Eqn 45
  Vdss = Vc*(sqrt( 0.25 + (($vt/(Vc))*sqrt(iff)))-0.5);  // Eqn 46;
  Vdssprime = Vc*(sqrt( 0.25 + ($vt/Vc)*(sqrt(iff)-0.75*ln(iff))) - 0.5)
              +$vt*(ln(Vc/(2.0*$vt)) - 0.6 ); // Eqn 47
  if (Lambda*(sqrt(iff) > (Vdss/$vt) ) )
          deltaV = 4.0*$vt*sqrt(Lambda*(sqrt(iff) -(Vdss/$vt)) + (1.0/64.0) );  // Eqn 48
  else    deltaV = 1.0/64.0;
  Vdso2 = (VD-VS)/2.0;         // Eqn 49
  Vip = sqrt( pow(Vdss, 2) + pow( deltaV,2)) - sqrt( pow( (Vdso2 - Vdss), 2)
        + pow (deltaV, 2));  // Eqn 50
  Lc = sqrt( (epsilonsi/Cox)*Xj);  // Eqn 51
  DeltaL = Lambda*Lc*ln(1.0+((Vdso2-Vip)/(Lc*Ucrit_T)));  // Eqn 52
  Lprime = Ns*Leff - DeltaL + ( (Vdso2+Vip)/Ucrit_T); // Eqn 53
  Lmin = Ns*Leff/10.0;  // Eqn 54
  Leq = 0.5*(Lprime + sqrt( pow(Lprime, 2) + pow(Lmin, 2)));   // Eqn 55
  X3 = (Vp-Vdso2-VS-sqrt( pow(Vdssprime, 2) + pow( deltaV, 2))
        + sqrt( pow( (Vdso2-Vdssprime), 2) + pow(deltaV,2)))/$vt;
  irprime = ln(1.0+limexp(X3/2.0))*ln(1.0+limexp(X3/2.0));  // Eqn 56
  Beta0 = Kpa_T*(Np*Weff/Leq);      // Eqn 58
  eta = 0.3333333;  // Eqn 59 - pMOS
  Qb0 = Gammaa*sqrt(Phi_T);  // Eqn 60;
  Beta0prime = Beta0*(1.0 +(Cox/(EO*epsilonsi))*Qb0);  // Eqn 61
  nq = 1.0 +Gammaa/(2.0*sqrt(Vp+Phi_T+1e-6));           // Eqn 69
 end
else
  nq = 1.0 +Gamma/(2.0*sqrt(Vp+Phi_T+1e-6));           // Eqn 69
//
Xf = sqrt(0.25+iff);      // Eqn 70
Xr = sqrt(0.25+ir);       // Eqn 71
qD = -nq*( (4.0/15.0)*((3.0*pow( Xr,3) + 6.0*pow( Xr, 2)*Xf + 4.0*Xr*pow( Xf, 2)
        + 2.0*pow(Xf, 3))/(pow( (Xf+Xr), 2) ) ) -0.5);  // Eqn 72
qS = -nq*( (4.0/15.0)*((3.0*pow( Xf,3) + 6.0*pow( Xf, 2)*Xr + 4.0*Xf*pow( Xr, 2)
        + 2.0*pow(Xr, 3))/(pow( (Xf+Xr), 2) ) ) -0.5);  // Eqn 73
qI = -nq*( (4.0/3.0)*( (pow(Xf,2)+(Xf*Xr)+pow(Xr,2))/(Xf+Xr)) - 1.0);     // Eqn 74
if (LEVEL == 2)
  if (VGprime > 0)
        qB = (-Gammaa*sqrt(Vp+Phi_T+1e-6))*(1.0/$vt) - ( (nq-1.0)/nq)*qI;  // Eqn 75
  else
        qB = -VGprime/$vt;
else
    if (VGprime > 0)
        qB = (-Gamma*sqrt(Vp+Phi_T+1e-6))*(1.0/$vt) - ( (nq-1.0)/nq)*qI;  // Eqn 75
  else
        qB = -VGprime/$vt;
//
if (LEVEL == 2)
  Beta = Beta0prime/(1.0 + (Cox/ (EO*epsilonsi))*$vt*abs(qB+eta*qI));        // Eqn 62
else
  Beta = Kp*(Weff/Leff)/(1+Theta*Vp);
//
Ispecific = 2.0*n*Beta*pow( $vt, 2);     // Eqn 65
//
if (LEVEL == 2)
  begin
    Ids = Ispecific*(iff-irprime);             // Eqn 66
    Vib = VD-VS-Ibn*2.0*Vdss;                  // Eqn 67
    if ( Vib > 0.0)
         Idb = Ids*(Iba/Ibb_T)*Vib*exp( (-Ibb_T*Lc)/Vib);    // Eqn 68
    else
         Idb = 0.0;
  end
else
```

```
   Ids = Ispecific*(iff-ir);            // Eqn 66
//
Sthermal = fourkt*Beta*abs(qI);
gm = Beta*$vt*(sqrt( (4.0*iff/Ispecific) +1.0) - sqrt( (4.0*ir/Ispecific) + 1.0) );
Sflicker = (Kf*gm*gm)/(Np*Weff*Ns*Leff*Cox);
//
qb = con2*$vt*qB;
qg = con2*$vt*(-qI-qB);
qgso = Cgso*Weff*Np*(VG-VS);
qgdo = Cgdo*Weff*Np*(VG-VD);
qgbo = Cgbo*Leff*Np*VG;
// Drain and source diodes
if (StoDswap > 0.0)
      begin
         V1=p_n_MOS*V(Bulk, Drain_int);
         V2=p_n_MOS*V(Bulk, Source_int);
      end
else
      begin
         V2=p_n_MOS*V(Bulk, Drain_int);
         V1=p_n_MOS*V(Bulk, Source_int);
      end
Id1= (V1>-5.0*N*$vt) ? Area*Is_T2*(limexp( V1/(N*Vt_T2) )-1.0) : 0;
Qd1=(V1<Fc*Vj)? Tt*Id1+Area*(Cj0_T2*Vj_T2/(1-M))*(1-pow((1-V1/Vj_T2),(1-M))):0;
Id2=(V1<=-5.0*N*$vt) ? -Area*Is_T2 : 0;
Qd2=(V1>=Fc*Vj)? Tt*Id1+Area*Cj0_T2*(F1+(1/F2)*(F3*(V1-Fc*Vj_T2)+(M/(2.0*Vj_T2))
      *(V1*V1-Fc*Fc*Vj_T2*Vj_T2))):0;
Id3=(V1 == -Bv) ? -Ibv   : 0 ;
Id4=(V1<-Bv) ?-Area*Is_T2*(limexp(-(Bv+V1)/Vt_T2)-1.0+Bv/Vt_T2) : 0;
Ib_d = Id1+Id2+Id3+Id4;
Qd = Qd1+Qd2;
//
Is1= (V2>-5.0*N*$vt) ? Area*Is_T2*(limexp( V2/(N*Vt_T2) )-1.0) : 0;
Qs1=(V2<Fc*Vj)? Tt*Is1+Area*(Cj0_T2*Vj_T2/(1-M))*(1-pow((1-V2/Vj_T2),(1-M))):0;
Is2=(V2<=-5.0*N*$vt) ? -Area*Is_T2 : 0;
Qs2=(V2>=Fc*Vj)? Tt*Is1+Area*Cj0_T2*(F1+(1/F2)*(F3*(V2-Fc*Vj_T2)+(M/(2.0*Vj_T2))
      *(V2*V2-Fc*Fc*Vj_T2*Vj_T2))):0;
Is3=(V2 == -Bv) ? -Ibv   : 0 ;
Is4=(V2<-Bv) ?-Area*Is_T2*(limexp(-(Bv+V2)/Vt_T2)-1.0+Bv/Vt_T2) : 0;
Ib_s = Is1+Is2+Is3+Is4;
Qs = Qs1+Qs2;
// Current contributions
if ( StoDswap > 0.0)
 begin
      if (RDeff > 0.0)
         I(Drain, Drain_int) <+ V(Drain, Drain_int)/RDeff;
      else
         I(Drain, Drain_int) <+ V(Drain, Drain_int)/1e-7;
      if (RSeff > 0.0)
         I(Source, Source_int) <+ V(Source, Source_int)/RSeff;
      else
         I(Source, Source_int) <+ V(Source, Source_int)/1e-7;
   I(Drain_int, Source_int) <+ p_n_MOS*Ids;
   if (LEVEL == 2)
      I(Drain_int, Bulk) <+ p_n_MOS*Idb;
   I(Gate, Drain_int) <+ p_n_MOS*0.5*ddt(qg);
   I(Gate, Source_int) <+ p_n_MOS*0.5*ddt(qg);
   I(Drain_int, Bulk) <+ p_n_MOS*0.5*ddt(qb);
   I(Source_int, Bulk) <+ p_n_MOS*0.5*ddt(qb);
   I(Gate, Source_int) <+ p_n_MOS*ddt(qgso);
   I(Gate, Drain_int)  <+ p_n_MOS*ddt(qgdo);
   I(Gate, Bulk)    <+ p_n_MOS*ddt(qgbo);
   I(Bulk, Drain_int) <+ p_n_MOS*Ib_d;
   I(Bulk, Drain_int) <+ p_n_MOS*ddt(Qd);
   I(Bulk, Source_int) <+ p_n_MOS*Ib_s;
```

```
   I(Bulk, Source_int) <+ p_n_MOS*ddt(Qs);
   I(Drain_int, Source_int) <+ white_noise(Sthermal,"thermal");
   I(Drain_int, Source_int) <+ flicker_noise(Sflicker, Af, "flicker");
   I(Drain, Drain_int) <+ white_noise(fourkt/RDeff, "thermal");
   I(Source, Source_int) <+ white_noise(fourkt/RSeff, "thermal");
end
else
 begin
     if (RSeff > 0.0)
        I(Drain, Drain_int) <+ V(Drain, Drain_int)/RSeff;
     else
        I(Drain, Drain_int) <+ V(Drain, Drain_int)/1e-7;
     if (RDeff > 0.0)
        I(Source, Source_int) <+ V(Source, Source_int)/RDeff;
     else
        I(Source, Source_int) <+ V(Source, Source_int)/1e-7;
   I( Source_int, Drain_int) <+ p_n_MOS*Ids;
   if (LEVEL == 2)
       I(Source_int, Bulk) <+ p_n_MOS*Idb;
   I( Gate, Source_int) <+ p_n_MOS*0.5*ddt(qg);
   I( Gate, Drain_int) <+ p_n_MOS*0.5*ddt(qg);
   I( Source_int, Bulk) <+ p_n_MOS*0.5*ddt(qb);
   I( Drain_int, Bulk) <+ p_n_MOS*0.5*ddt(qb);
   I( Gate, Drain_int) <+ p_n_MOS*ddt(qgso);
   I( Gate, Source_int)  <+ p_n_MOS*ddt(qgdo);
   I( Gate, Bulk)    <+ p_n_MOS*ddt(qgbo);
   I( Bulk, Source_int) <+ p_n_MOS*Ib_d;
   I( Bulk, Source_int) <+ p_n_MOS*ddt(Qd);
   I( Bulk, Drain_int) <+ p_n_MOS*Ib_s;
   I( Bulk, Drain_int) <+ p_n_MOS*ddt(Qs);
   I( Source_int, Drain_int) <+ white_noise(Sthermal,"thermal");
   I( Source_int, Drain_int) <+ flicker_noise(Sflicker, Af, "flicker");
   I( Source_int, Source) <+ white_noise(fourkt/RDeff, "thermal");
   I( Drain_int, Drain) <+ white_noise(fourkt/RSeff, "thermal");
 end
end
endmodule
```

Update number one: September 2008

The first version of the Qucs EPFL-EKV v2.6 model provided Qucs users with reasonably complete long and short channel models for nMOS and pMOS devices. In no respect were these models optimized for minimum simulation run time or were they flexible enough to allow users to select the style of charge partitioning employed by the EKV model. Recent work on the Qucs implementation of the EKV v2.6 MOSFET model and the Qucs ADMS/XML interface has resulted in a significant reduction in simulation run time overhead, particularly in transient and small signal analysis. The addition of a SPICE BSIM style partition parameter Xpart to the Qucs version of the EKV v2.6 model now allows users to set the style of charge partitioning employed by the EKV model. These notes explain the function of the first EKV v2.6 update and introduce a series of test simulations that demonstrate the effects these changes have on the operation of the Qucs port of the EKV V2.6 model.

Model initialisation

Readers who have looked through the EKV v2.6 Verilog-A code listed in the previous sections of these notes will probably have been struck by the quantity of calculations involved each time the code is evaluated during simulation. In the case of transient analysis it is calculated at least once per time step, often resulting in many thousands of passes through the code. The more MOS devices included in a circuit the greater the time overhead becomes. Obviously, a sensible approach would be to minimize the amount of calculation by only evaluating once those parts of the EKV model equations which result in constant values during simulation. The Verilog-A hardware description language provides a model initialisation feature which selects those parts of a device model code which are to be evaluated prior to the start of a simulation. The resulting calculated variables are then available for use by other sections of the Verilog-A model code during simulation. In transient and small signal analysis this is particularly important as it significantly reduces simulation calculation time. Verilog-A employs the "at" (@(`initial_step`) or @(`initial_model`)) language construction coupled with a begin ... end block to signify the Verilog-A code that is to be evaluated only at model initialisation. Although this technique does greatly improve model simulation speed it does imply significantly more work for the model developer in that the Verilog-A device code has to be split into initialisation and dynamic simulation sections. Readers interested in the detail of how this split can be achieved should compare the latest EKV v2.6 Verilog-A CVS code given at the Qucs Web site with that presented in previous sections of these notes.

Charge partitioning

The MOSFT is a four terminal device with a dynamic performance that requires accurate calculation of the charge at each terminal. Previous notes indicated that the intrinsic channel charge equals the sum of the drain and source charges. However, the exact proportion of intrinsic channel charge that belongs to the drain or to the source is often not known. The assignment of the proportion of the channel charge to the drain and source charges is called charge partitioning. The first release of the Qucs EKV v2.6 model used the 50/50 partitioning scheme where 50% of the channel charge is arbitrarily assigned to both drain and source. It's interesting to note that this partitioning scheme has no physical basis but depends entirely on convenience. A second partitioning scheme, called the 40/60 partitioning, does however, have a strong physical basis[9]. Yet a third charge partitioning is often employed for digital circuit simulation; this is known as the 0/100 partition. The second release of the Qucs EPFL-EKV v2.6 model includes an extra parameter called Xpart which allows users to set the partitioning scheme for dynamic simulation calculations. Xpart default is set at 0.4 which corresponds to the 40/60 partitioning scheme. Figure 10 illustrates a test circuit for determining the S-Parameters of an nMOS device connected as a capacitance. Both the device capacitance and associated series resistance can be extracted from S[1,1]. Qucs equation block Eqn1 gives the equations for extracting these properties. Other equations in Eqn1 show how the extracted capacitance can be represented as a ratio of the basic parallel capacitance given by

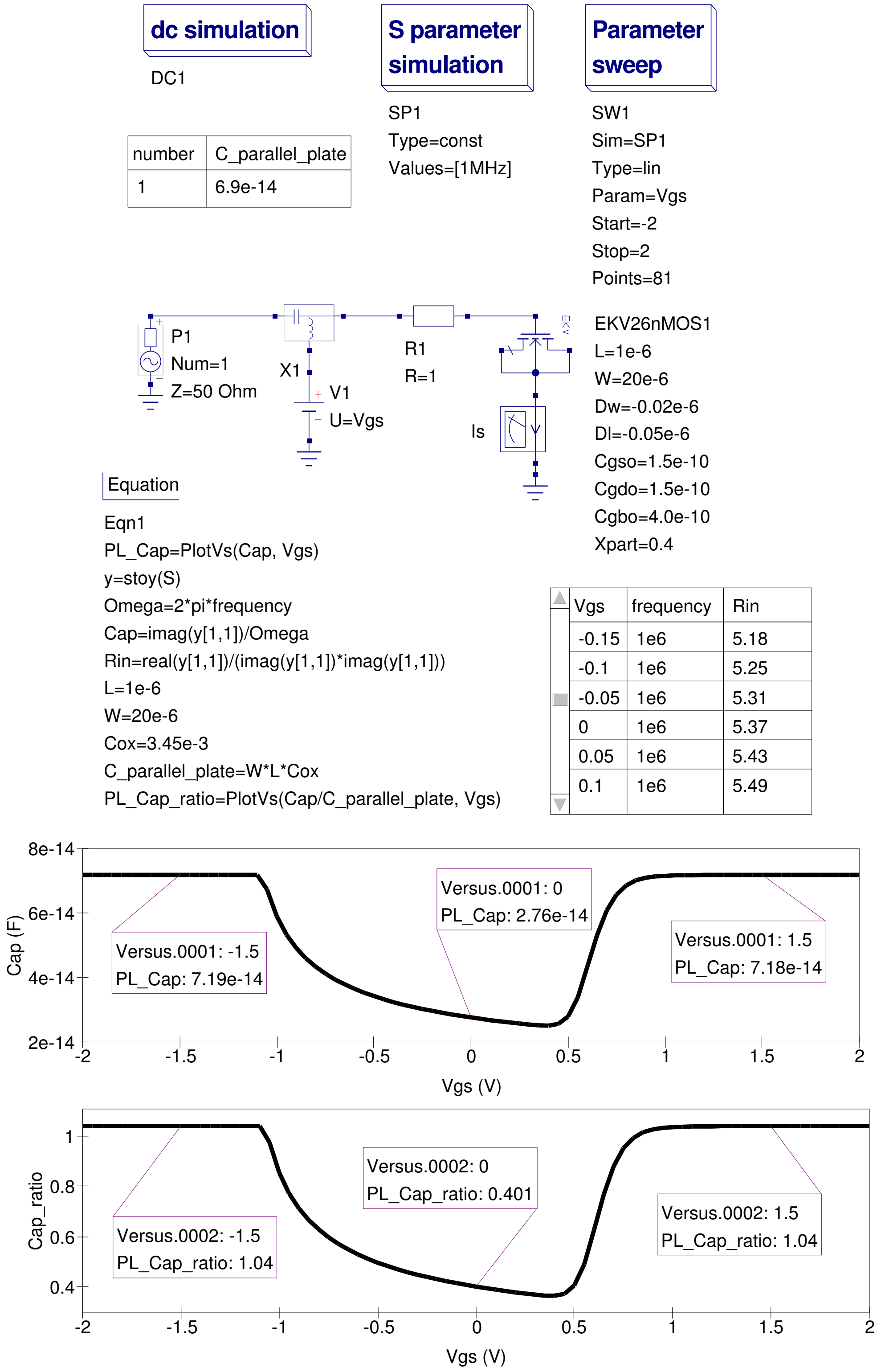
$$C\_parallel\_plate = W \cdot L \cdot Cox. \tag{30}$$

Modelling EKV v2.6 charge partitioning using Qucs EDD

Complex simulation results like those shown in Fig 10 suggest the question "How do we check the accuracy of the model being simulated?". One possible approach is to develop a second model of the same device based on the same physical principles and equations but using a different approach like the Qucs EDD/subcircuit modelling route shown in Fig. 11. It is an EDD/subcircuit model of a long channel EKV v2.6 nMOS device which includes charge partitioning. Figure 12 illustrated the same test circuit as Fig. 10 and the extracted capacitance and resistance values for the EDD model of the long channel nMOS device. A number of features observed from Fig. 10 and Fig. 11 are worth commenting on; firstly that good agreement is recorded between the two sets of results, secondly that the Verilog-A model includes both overlap capacitance and drain and gate source resistances. Hence the slight difference in the capacitance ratio and the recorded values of Rin above one Ohm for the Verilog-A model.

---

[9] William Liu, MOSFET models for SPICE simulation, including BSIM3v3 and BSIM4, 2001, Wiley-Interscience publications, ISBN 0-471-39697-4.



dc simulation

DC1

number	C_parallel_plate
1	6.9e-14

S parameter simulation

SP1
Type=const
Values=[1MHz]

Parameter sweep

SW1
Sim=SP1
Type=lin
Param=Vgs
Start=-2
Stop=2
Points=81

P1
Num=1
Z=50 Ohm

X1

V1
U=Vgs

R1
R=1

Is

EKV26nMOS1
L=1e-6
W=20e-6
Dw=-0.02e-6
Dl=-0.05e-6
Cgso=1.5e-10
Cgdo=1.5e-10
Cgbo=4.0e-10
Xpart=0.4

Equation

Eqn1
PL_Cap=PlotVs(Cap, Vgs)
y=stoy(S)
Omega=2*pi*frequency
Cap=imag(y[1,1])/Omega
Rin=real(y[1,1])/(imag(y[1,1])*imag(y[1,1]))
L=1e-6
W=20e-6
Cox=3.45e-3
C_parallel_plate=W*L*Cox
PL_Cap_ratio=PlotVs(Cap/C_parallel_plate, Vgs)

Vgs	frequency	Rin
-0.15	1e6	5.18
-0.1	1e6	5.25
-0.05	1e6	5.31
0	1e6	5.37
0.05	1e6	5.43
0.1	1e6	5.49

Figure 10: Test circuit for simulating EKV v2.6 charge partitioning effects: Xpart = 0.4 or QD/QS = 40/60

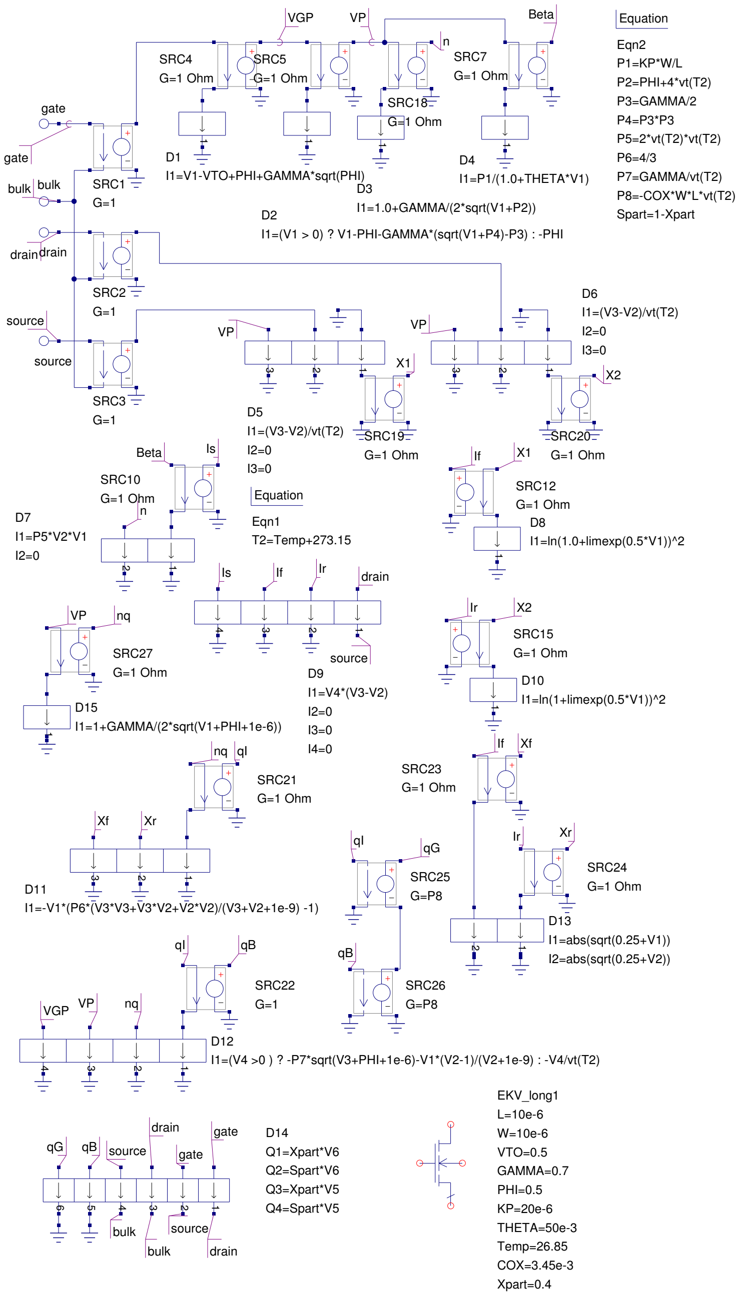
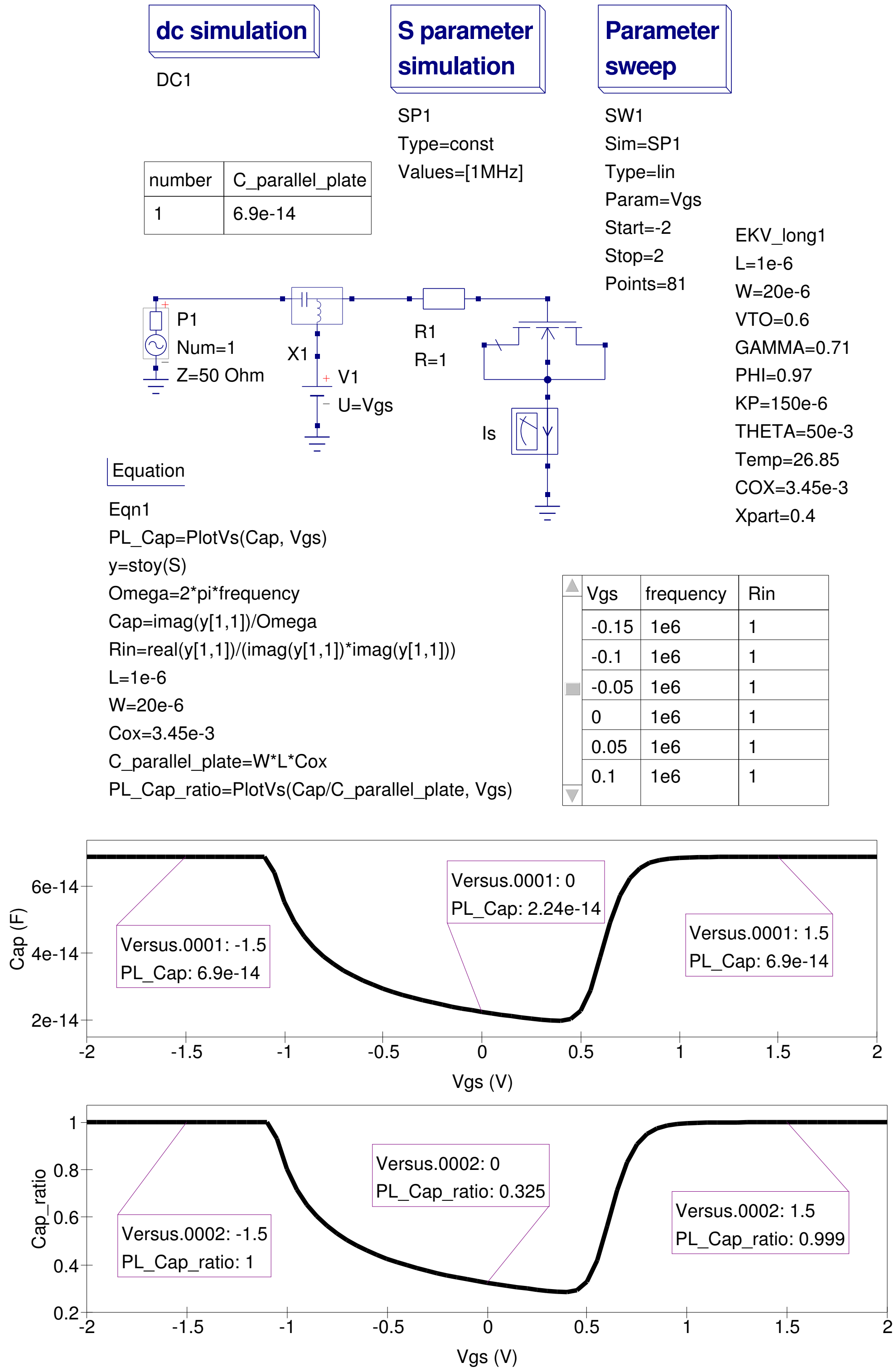


Figure 11: Qucs EDD model for a EKV v2.6 long channel nMOS device with charge partitioning



dc simulation

DC1

S parameter simulation

SP1
Type=const
Values=[1MHz]

Parameter sweep

SW1
Sim=SP1
Type=lin
Param=Vgs
Start=-2
Stop=2
Points=81

number	C_parallel_plate
1	6.9e-14

EKV_long1
L=1e-6
W=20e-6
VTO=0.6
GAMMA=0.71
PHI=0.97
KP=150e-6
THETA=50e-3
Temp=26.85
COX=3.45e-3
Xpart=0.4

P1
Num=1
Z=50 Ohm

V1
U=Vgs

R1
R=1

Equation

Eqn1
PL_Cap=PlotVs(Cap, Vgs)
y=stoy(S)
Omega=2*pi*frequency
Cap=imag(y[1,1])/Omega
Rin=real(y[1,1])/(imag(y[1,1])*imag(y[1,1]))
L=1e-6
W=20e-6
Cox=3.45e-3
C_parallel_plate=W*L*Cox
PL_Cap_ratio=PlotVs(Cap/C_parallel_plate, Vgs)

Vgs	frequency	Rin
-0.15	1e6	1
-0.1	1e6	1
-0.05	1e6	1
0	1e6	1
0.05	1e6	1
0.1	1e6	1

Figure 12: Test circuit for simulating EKV v2.6 EDD model charge partitioning effects: Xpart = 0.4 or QD/QS = 40/60

End note

The first update of the Qucs EKV v2.6 model provides users with a more optimised model, with improved simulation performance and a more complete charge partitioning scheme. Even with these changes the model is still not complete. The nMOS and pMOS Verilog-A code needs to be unified and a number of optional parameters need to be added to the Qucs implementation of the EKV v2.6 model. The next update of the model is scheduled for the near future, following correction of bug reports sent in by Qucs users. Once again my thanks to Stefan Jahn for all his help and support during the first EKV v2.6 update development phase.