IOSR Journal of VLSI and Signal Processing (IOSR-JVSP)
Volume 8, Issue 1, Ver. I (Jan.-Feb. 2018), PP 73-86
e-ISSN: 2319 – 4200, p-ISSN No. : 2319 – 4197
www.iosrjournals.org

MOSFET EKV Verilog-A Model Implementation in Genesys

George P. Patsis[1,2]

[1]*Department of Electronic Engineering, Technological and Educational Institute of Athens,Aegaleo, Attica, Greece*

[2]*Nanometrisis private company, Scientific & Technological Park "Leukippos", NCSR Demokritos, Neapoleos 27 & Patr. Grigoriou Str., 15341, Agia Paraskevi, Attica, Greece*

Corresponding Author: George P. Patsis

Abstract: *A simple version of the EKV MOSFET model is implemented in Verilog-A and tested in Keysight's Genesys software suite. The basic physics of the model is briefly presented and the process of model development in Verilog-A and its integration into the software's library is discussed in detail. The aim of this work is to present the advantages of analog modeling with hardware description languages, especially when developing nonlinear device models, and to show some tools to accomplish such a task.*

Keywords –*MOSFET model, EKV, Verilog-A, Genesys simulator*

Date of Submission: 23-02-2018 Date of acceptance: 10-03-2018

I. INTRODUCTION

Analog modeling enables designers to capture high-level behavioral descriptions of components in a precise set of mathematical terms. The analog module's relation of input to output can be related by the external parameter description and the mathematical relations between the input and output ports.Analog models give the designer control over the level of abstraction with which to describe the action of the component.

This can provide higher levels of complexity to besimulated, allow faster simulation execution speeds, or can hide intellectual property. An analog model should ideally model the characteristics of the behavior as accurately as possible, with the trade-off of model complexity, which is usually manifested by reduced execution speed.

Hardware description languages (HDLs) were developed as a means to provide varying levels of abstraction to designers. Integrated circuits are too complex for an engineer to create by specifying the individual transistors and wires. HDLs allow the performance to be described at a high level and simulation synthesis programs can then take the language and generate the gate level description.Verilog and VHDL are the two dominant languages. In the current work the focus is with the Verilog language. As behavior beyond the digital performance was added, a mixed-signal language was created to manage the interaction between digital and analog signals. A subset of this, Verilog-A, was defined. Verilog-A describes analog behavior only; however, it has functionality to interface to some digital behavior.

Verilog-A provides a high-level language to describe the analog behavior of conservative systems. The disciplines and natures of the Verilog-A language enable designers to reflect the potential and flow descriptions of electrical, mechanical, thermal, and other systems. Verilog-A is a procedural language, with constructs similar to C and other languages. It provides simple constructs to describe the model behavior to the simulator program. The model effectively de-couples the description of the model from the simulator. The model creator provides the constitutive relationship of the inputs and outputs, the parameter names and ranges, while the Verilog-A compiler handles the necessary interactions between the model and the simulator. While the language does allow some knowledge of the simulator, most model descriptions should not need to know anything about the type of analysis being run.

In the current article, a simple version of the EKV MOSFET model is implemented in Verilog-A and tested in Keysight's Genesys software suite. In section II, the physics of the model is developed, along with the necessary steps for the model structuring in Verilog-A. Then in section III, the implementation of the model is presented along with basic simulations for testing its validity.

II. ENZ-KRUMMENACHER-VITTOZ (EKV) MODEL OF THE MOS TRANSISTOR

EKV MOS model [1] is a physics based model. It is based on the calculation of the charges in the terminal regions and the channel of the transistor structure. The modeling process can be divided into two parts: (a) one for the internal (intrinsic) structure of the MOS and (b) another for the external considering more macroscopic device characteristics. An advantage of EKV is that is uses the same set of equations for all transistor-operation regions, so the IV characteristic is uniquely and continuously determined for all regions of operation. The model symmetry is the other advantage of the EKV model. The electrical equivalent of the EKV model is seen in Fig. 1.A simplified version of the model implemented in the current work is seen in Fig. 2 and is briefly described below.

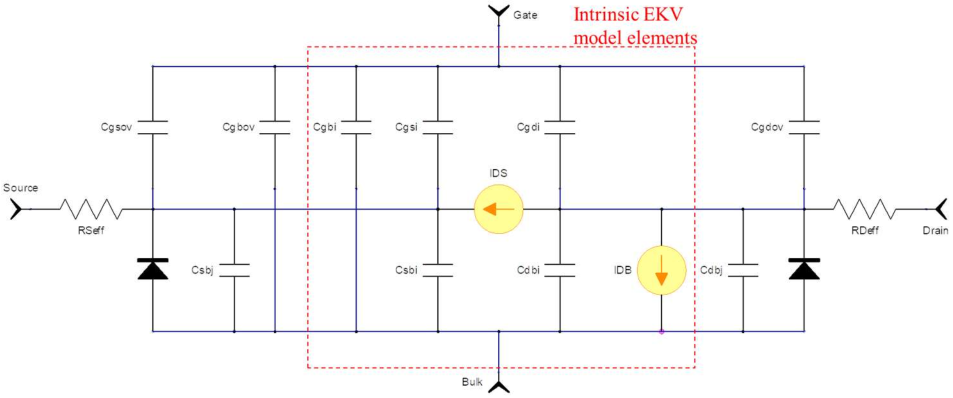


Figure 1. The basic EKV model electrical equivalent schematic of the MOS transistor. Refer to [1] for quantity definitions.

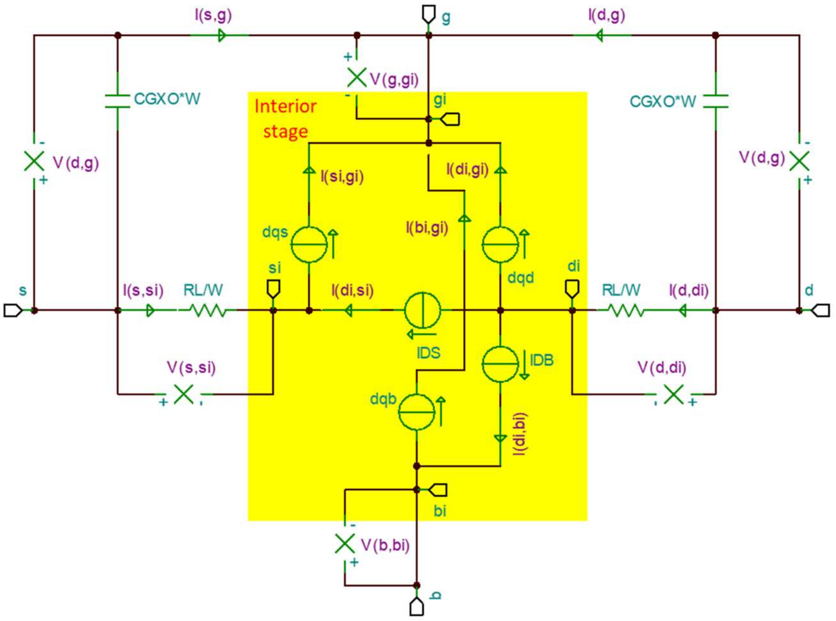


Figure 2. Simplified EKV equivalent model implemented in the current work. Various quantities shown are described in the Verilog-A code in Listing 8.

Without going into the many details of the physics of the model, only the basic steps of the formulation of a Verilog-A model will be mentioned next.

A. SPECIFICATION OR CALCULATION OF BODY FACTOR COEFFICIENT

The body factor coefficient γ is calculated when it is not specified, by equation (1), where q is the electron charge, ε_{Si} the silicon dielectric constant, C_{OX}, the oxide capacitance per unit of area, and N_{sub} the substrate impurity concentration. Generally, γ is technology dependent.

$$\gamma = \frac{\sqrt{2q\varepsilon_{\mathrm{Si}}N_{sub}}}{C_{OX}} \tag{1}$$

B. CALCULATION OF PINCH-OFF VOLTAGE

The pinch-off voltage V_P is calculated through threshold voltage V_{T0}, parameter Φ_0 (twice the Fermi potential) and the body factor coefficientγ, by equation (2).

$$V_P = V_G - V_{T0} + \gamma\Phi_0 - \gamma\left\{\sqrt{V_G - V_{T0} + \Phi_0 + \gamma\Phi_0 + \frac{\gamma^2}{4}} - \frac{\gamma}{2}\right\} \quad (2)$$

It is used in the determination of current. It is a function of the gate voltage. It is zero when the charge in the channel is zero. The difference (V_S and V_P are the source and drain voltage respectively) V_P-V_S and V_P-V_D is used to calculate the source and drain current respectively (having opposite directions). The difference between these currents provides the total current in the channel. The process of obtaining (2) is described now. Looking from the gate down, there are several interfaces that can be identified: the metal-semiconductor (MS) interface, the oxide (OX), the inversion layer (INV) and the bulk (B) of the substrate's body. Equation (3) can be written for these interfaces, where V_G is the gate voltage Φ_{MS} is the MS voltage, Φ_{OX} the oxide voltage and Φ_S the semiconductor surface voltage. Due to the structure being electrically neutral, eq. (4) will hold for the corresponding charges. Q_C is the channel charge for which eq. (5) can be written. The EKV model is based on the charges, so these must be expressed in terms of the voltages that are the quantities manipulated outside of the device. The flat-band voltage V_{FB} (6) is the voltage applied to the gate that results in zero charge in the channel, zero semiconductor surface-voltage and opposite charge between oxide and gate material. Equation (7), which defines the built in oxide-voltage, holds in this case. Using (3), eq. (8) can also be considered for the flat-band voltage. In the general case (9) holds, and (10) is obtained, which when used in (3) gives (11) for the gate voltage in terms of the channel charge.

$$V_G = \Phi_{MS} + \Phi_{OX} + \Phi_S \quad (3)$$

$$Q_G + Q_{OX} + Q_C = 0 \quad (4)$$

$$Q_C = Q_{INV} + Q_B \quad (5)$$

$$V_{FB} = V_G(Q_C = 0) : Q_{OX} = -Q_G, \Phi_S = 0 \quad (6)$$

$$\Phi_{OX} = \frac{Q_G}{C_{OX}} = -\frac{Q_{OX}}{C_{ox}} \quad (7)$$

$$V_{FB} = \Phi_{MS} - \frac{Q_{OX}}{C_{OX}} \quad (8)$$

$$Q_{OX} = -Q_G - Q_C \quad (9)$$

$$\Phi_{MS} = V_{FB} + \frac{Q_{OX}}{C_{OX}} = V_{FB} - \frac{Q_G}{C_{OX}} - \frac{Q_C}{C_{OX}} \quad (10)$$

$$V_G = V_{FB} - \frac{Q_G}{C_{OX}} - \frac{Q_C}{C_{OX}} + \frac{Q_G}{C_{OX}} + \Phi_S \Leftrightarrow V_G = V_{FB} + \Phi_S - \frac{Q_C}{C_{OX}} \quad (11)$$

Using Poisson equation (12) (y is the direction vertical from the gate towards substrate), where the thermal voltage U_T and the Fermi potentialΦ_F are defined by (13) and (14) respectively, it becomes possible to obtain the charge in the channel, in the bulk and in the inversion layer through equations (15), (16), and (17) respectively.

$$\frac{\partial^2\Phi}{\partial y^2} = -\frac{qN_A}{\varepsilon_{\mathrm{Si}}}\left[\exp\left(-\frac{\Phi}{U_T}\right) - 1 - \exp\left(-\frac{2\Phi_F}{U_T}\right)\left(\exp\left(\frac{\Phi - V_C}{U_T}\right) - 1\right)\right] \quad (12)$$

$$U_T = \frac{k_B T}{q} \quad (13)$$

$$\Phi_F = U_T \ln\frac{N_{sub}}{N_i} \quad (14)$$

$$Q_C = -\gamma C_{OX}\sqrt{U_T}\sqrt{\frac{\Phi_S}{U_T} + \exp\left(\frac{\Phi_S - 2\Phi_F - V_C}{U_T}\right)} \quad (15)$$

$$Q_B = -\gamma C_{OX}\sqrt{U_T}\sqrt{\frac{\Phi_S}{U_T}} \quad (16)$$

$$Q_{INV} = -\gamma C_{OX}\sqrt{U_T}\left[\sqrt{\frac{\Phi_S}{U_T} + \exp\left(\frac{\Phi_S - 2\Phi_F - V_C}{U_T}\right)} - \sqrt{\frac{\Phi_S}{U_T}}\right] \quad (17)$$

Returning to eq. (11), it can be written as in (18), which, if the last term is assumed negligible, can be solved for the semiconductor surface potentialΦ_S (19). Then, in strong inversion, the semiconductor surface potential is approximately related with the channel voltage through a relation such as (20) new potential parameter $\Phi_0$is equal to twice the Fermi potential plus a few thermal voltages. It is consider as a model parameter depending on the fabrication technology. Now, the pinch-off voltage V_P is the channel voltage just before the channel is formed, so eq. (21) should hold. The threshold voltageV_{T0}could also be defined now, as the gate voltage for zero minority carrier concentration and zero channel voltage, though eq. (22). Finally, using the auxiliary definition (23), the final pinch-off voltage implemented in the Verilog-A code is given by eq. (24).

$$V_G - V_{FB} = \Phi_S + \gamma\sqrt{\Phi_S} - \frac{Q_{INV}}{C_{OX}} \quad (18)$$

$$\Phi_S \simeq V_G - V_{FB} - \gamma\left(\sqrt{V_G - V_{FB} + \frac{\gamma^2}{4}} - \frac{\gamma}{2}\right) \quad (19)$$

$$\Phi_S = \Phi_0 + V_C \quad (20)$$

$$V_P = V_C(Q_{INV} = 0) = V_G - V_{FB} - \gamma\left(\sqrt{V_G - V_{FB} + \frac{\gamma^2}{4}} - \frac{\gamma}{2}\right) - \Phi_0 \quad (21)$$

$$V_{T0} = V_G(Q_{INV} = 0, V_C = 0) = V_{FB} + \Phi_0 + \gamma\Phi_0 \quad (22)$$

$$V'_G = V_G - V_{FB} = V_G - V_{T0} + \Phi_0 + \gamma\Phi_0 \quad (23)$$

$$V_P = V_{G'} - \Phi_0 - \gamma\left[\sqrt{V'_G + \frac{\gamma^2}{4}} - \frac{\gamma}{2}\right] \quad (24)$$

C. CALCULATION OF SLOPE COEFFIENT

The slope factor *n* is calculated with eq. (25) and expressed the slope of the gate voltage in terms of the pinch-off voltage. Practically, it is observed that the pinch-off voltage is liner in respect to gate voltage and in a first approximation eq. (26) holds.

$$n = \frac{dV_G}{dV_P} = 1 + \frac{\gamma}{2\sqrt{\Phi + V_p}} \quad (25)$$

$$V_P \simeq \frac{V_G - V_{T0}}{n(V_G)} \quad (26)$$

D. CALCULATION OF DRAIN AND SOURCE CHARGES AND CHANNEL CURRENT INTERELATIONS

The determination of drain and source charges is crucial in EKV model because the current calculation is based on them. So, their calculation is linked with the current calculation. The channel current is defined through eq. (27), where the slope of the channel voltage V_C with the position along the channel is given by eq. (28). Combining these two equations, (29) occurs, and using (30) and integrating along the channel length *L,* eq. (31) occurs, where the forward I_F and reverse I_R currents are given by (32) and (33) respectively. These two equations can also be solved for the charge in the source $Q_{INV}(S)$ and the drain $Q_{INV}(D)$, resulting in equations (34) and (35). Now, defining the charge normalization factorQ_0 using eq. (36) and the current normalization factorI_0 using eq. (37) and performing normalizations as seen in eqs. (38), one gets eqs. (39)-(43), which are the ones implemented in Verilog-A as part of the simple EKV model presented here. Obviously a method is required to determine the charges without the knowledge of current. This is considered next.

$$I_D = \mu W(-Q_{INV})\frac{dV_C}{dx} \quad (27)$$

$$\frac{dV_C}{dx} = \frac{d\Phi_S}{dx} + \frac{U_T}{Q_{INV}}\frac{dQ_{INV}}{dx} \quad (28)$$

$$I_D = \mu W\left(-Q_{INV}\frac{d\Phi_S}{dx} - U_T\frac{dQ_{INV}}{dx}\right) \quad (29)$$

$$\frac{d\Phi_S}{dx} = \frac{1}{nC_{OX}}\frac{dQ_{INV}}{dx} \quad (30)$$

$$I_D = \int_0^L I_D dx = \mu W\int_0^L\left(-Q_{INV}\frac{1}{nC_{OX}}\frac{dQ_{INV}}{dx} + U_T\frac{dQ_{INV}}{dx}\right)dx = I_F - I_R \quad (31)$$

$$I_F = \mu\frac{W}{L}\left[\frac{Q_{INV}(S)}{nC_{OX}} - U_T Q_{INV}(S)\right] \quad (32)$$

$$I_R = \mu\frac{W}{L}\left[\frac{Q_{INV}(D)}{nC_{OX}} - U_T Q_{INV}(D)\right] \quad (33)$$

$$Q_{INV}(S) = -2nC_{OX}U_T\left[\sqrt{\frac{1}{4} + \frac{I_F}{2nC_{ox}\mu\frac{W}{L}U_T^2}} - \frac{1}{2}\right] \quad (34)$$

$$Q_{INV}(D) = -2nC_{OX}U_T\left[\sqrt{\frac{1}{4} + \frac{I_R}{2nC_{ox}\mu\frac{W}{L}U_T^2}} - \frac{1}{2}\right] \quad (35)$$

$$Q_0 = -2nC_{OX}U_T \quad (36)$$

$$I_0 = 2n\mu C_{ox}U_T^2\frac{W}{L} \quad (37)$$

$$q_{inv} = \frac{Q_{INV}}{Q_0}, i_f = \frac{I_F}{I_0}, i_r = \frac{I_R}{I_0}, v_s = \frac{V_S}{U_T}, v_d = \frac{V_D}{U_T} \quad (38)$$

$$I_D = I_0(i_f - i_r) \tag{39}$$

$$i_f = q_s^2 + q_s \tag{40}$$

$$i_r = q_d^2 + q_d \tag{41}$$

$$q_s = \sqrt{\tfrac{1}{4} + i_f} - \tfrac{1}{2} \tag{42}$$

$$q_d = \sqrt{\tfrac{1}{4} + i_r} - \tfrac{1}{2} \tag{43}$$

E. CALCULATION OF CHARGES THROUGH VOLTAGES

In the previous paragraph the connection between current and charge is developed. If the voltage is related to the charge, then the voltage-current relation could be realized. In the EKV model this is accomplished considering the channel transconductance. Without going into many more details (refer to [1]-[4]), the charges could be obtained through (44) and (45) where the function *F* is given by eq. (46). The *q*s are the normalized charges and the *v*s are the normalized voltages. By numerically inverting (46), (using iterative Newton-Raphson method) the charges could be obtained and then the current, using (39)-(41).

$$q_s = F^{-1}(v_p - v_s) \tag{44}$$

$$q_d = F^{-1}(v_p - v_d) \tag{45}$$

$$v_p - v_c = 2q_{inv} + \ln q_{inv} = F(q_{inv}) \tag{46}$$

Unfortunately eq. (46) cannot be inverted analytically. However it can be inverted using a Newton-Raphson iterative scheme. A simplification of this algorithm that avoids iteration is used, leading to a continuous expression for the large signal interpolation function. The (inverted) large signal interpolation function has the following asymptotes in strong and weak inversion respectively, (47) [4].

$$F(v) = \left\{ \begin{array}{c} (v/2)^2, v >> 0 \\ \exp(v), v << 0 \end{array} \right\} \tag{47}$$

F. CALCULATION OF IMPACT IONIZATION CURRENT

Besides drain current, the impact ionization current I_{DB} is modeled in this work. This component is due to the electron impact from the drain to the bulk. It increases with V_{DS}. Due to their increased velocity in the drain, electrons there can ionize the bulk and create this parasitic current. The corresponding equations are easy to implement in Verilog-A. Equation (47) gives the velocity saturation voltage V_C, where U_{CRIT} is the longitudinal critical field. I_{BA} is the first impact ionization coefficient, and I_{BB} the second impact ionization coefficient. Equations (48)-(50) are required to finally determine I_{DB} with eq. (51).

$$V_C = U_{CRIT}L \tag{47}$$

$$V_{DSS} = V_C\left[\sqrt{\tfrac{1}{4} + \tfrac{U_T}{V_C}\sqrt{i_f}} - \tfrac{1}{2}\right] \tag{48}$$

$$L_C = \sqrt{\tfrac{\varepsilon_{\mathrm{Si}}}{C_{OX}} x_j} \tag{49}$$

$$V_{IB} = V_D - V_S - 2I_{BN}V_{DSS} \tag{50}$$

$$I_{DB} = \left\{ \begin{array}{c} I_{DS}\tfrac{I_{BA}}{I_{BB}}V_{IB}\exp\left(-\tfrac{I_{BB}L_C}{V_{IB}}\right), V_{IB} > 0 \\ 0, V_{IB} < 0 \end{array} \right\} \tag{51}$$

G. QUASI-STATIC MODEL EQUATIONS

Finally the quasi-state region of the MOS behavior on ac signals is modeled. It is based on the determination of the absolute charge on the device nodes and then by its time derivative the corresponding current component is determined. Based on the equations specified in [4], eqs. (52) – (58) are formulated. The time derivatives on (55)-(58) provide current components between drain-gate, source-gate, and bulk-gate respectively, as seen in Fig. 2.

$$n_q = 1 + \frac{\gamma}{2\sqrt{V_P+\Phi+10^{-6}}} \qquad (52)$$

$$x_f = \sqrt{\frac{1}{4} + i_f} \qquad (53)$$

$$x_r = \sqrt{\frac{1}{4} + i_r} \qquad (54)$$

$$q_d = -n_q\left[\frac{4}{15}\frac{3x_f^3+6x_r^2x_f+4x_rx_f^2+2x_f^3}{(x_f+x_r)^2} - \frac{1}{2}\right] \qquad (55)$$

$$q_s = -n_q\left[\frac{4}{15}\frac{3x_f^3+6x_f^2x_r+4x_fx_r^2+2x_r^3}{(x_f+x_r)^2} - \frac{1}{2}\right] \qquad (56)$$

$$q_{inv} = q_s + q_d = -n_q\left[\frac{4}{3}\frac{x_f^2+x_fx_r+x_r^2}{x_f+x_r} - 1\right] \qquad (57)$$

$$q_b = -\gamma\sqrt{V_P + \Phi} - \frac{n_q-1}{n_q}q_{inv} \qquad (58)$$

III. Verilog-A Coding Of The Model

The simple version of the EKV model described above, is implemented here in Verilog-A [2], and then the model is simulated within the Keysight's Genesys software suite. Genesys Advanced Modeling Kit gives the ability to develop custom nonlinear device models such as transistors, diodes, Micro-electro-mechanical Systems (MEMs) or electro-optical devices using Verilog-A. The Verilog-A models can then be used in the Genesys frequency- or time-domain nonlinear circuit simulators (i.e. Harbec or Cayenne respectively) to design first-to-market circuit or system components.

The code begins with the calling of several header files. Specifically the "disciplines.vams", the "constants.vams", and the "compact.vams" files (Listing 1). These files contain definition of quantities that are useful in the following code. For example "constants.vams" contains Verilog-A definition of mathematical and physical constants (Listing 2). The "disciplines.vams" contains Verilog-A definition of natures and disciplines (Listing 3). Finally, the "compact.vams" file contains various useful, common macro definition and utilities (Listing 4). Using these header files it is easy to develop the code for the MOSFET. For example it is now easier to define the Silicon dielectric constant as shown in Listing 5.

Listing 1. Calling header files.

```
`include "disciplines.vams"
`include "constants.vams"
`include "compact.vams"
```

Listing 2. Contents of constants.vams file.

```
// The constants.vams File
/*
 Verilog-A definition of Mathematical and physical constants
 $RCSfile: constants.vams,v $ $Revision: 1.1 $ $Date: 2003/09/22 01:36:17 $
*/
`ifdef CONSTANTS_VAMS
`else
`define CONSTANTS_VAMS 1
// M_ indicates a mathematical constant
`define M_E 2.7182818284590452354
`define M_LOG2E 1.4426950408889634074
`define M_LOG10E 0.43429448190325182765
`define M_LN2 0.69314718055994530942
`define M_LN10 2.30258509299404568402
`define M_PI 3.14159265358979323846
`define M_TWO_PI 6.28318530717958647652
`define M_PI_2 1.57079632679489661923
`define M_PI_4 0.78539816339744830962
`define M_1_PI 0.31830988618379067154
```

```
`define M_2_PI 0.63661977236758134308
`define M_2_SQRTPI 1.12837916709551257390
`define M_SQRT2 1.41421356237309504880
`define M_SQRT1_2 0.70710678118654752440
// P_ indicates a physical constant
// charge of electron in coulombs
`define P_Q 1.6021918e-19
// speed of light in vacuum in meters/sec
`define P_C 2.997924562e8
// Boltzmann’s constant in joules/kelvin
`define P_K 1.3806226e-23
// Plank's constant in joules*sec
`define P_H 6.6260755e-34
// permittivity of vacuum in farads/meter
`define P_EPS0 8.85418792394420013968e-12
// permeability of vacuum in henrys/meter
`define P_U0 (4.0e-7 * `M_PI)
// zero Celsius in kelvin
`define P_CELSIUS0 273.15
`endif
```

Listing 3. Part of disciplines.vams file.

```
…
// Electrical
// Current in amperes
natureCurrent
units = "A";
access = I;
idt_nature = Charge;
`ifdef CURRENT_ABSTOL
abstol = `CURRENT_ABSTOL;
`else
abstol = 1e-12;
`endif
endnature
// Charge in coulombs
natureCharge
units = "coul";
access = Q;
ddt_nature = Current;
`ifdef CHARGE_ABSTOL
abstol = `CHARGE_ABSTOL;
`else
abstol = 1e-14;
`endif
endnature
// Potential in volts
natureVoltage
units = "V";
access = V;
idt_nature = Flux;
`ifdef VOLTAGE_ABSTOL
abstol = `VOLTAGE_ABSTOL;
`else
abstol = 1e-6;
`endif
endnature
…
```

Listing 4. Part of "compact.vams" file.

```
…
// SPICE-specificdifferentvalues:
`define SPICE_GMIN 1.0e-12
`define SPICE_K 1.3806226e-23
`define SPICE_Q 1.6021918e-19
`define LARGE_REAL 1.0e38
`define MIN_CONDUCTANCE 1.0e-3
`define DEFAULT_TNOM 27
…
```

Listing 5. Definition of silicon dielectric constant.

```
`define   MY_P_ESI   11.7*`P_EPS0
```

The body of the code should begin with a module definition with the name of the device and the terminal names, specifying the input-output nodes and the nature (electrical in this case) of these nodes (d,g,s,b). Any other intrinsic node not seen on the outside world is also defined here (di, gi, si, bi)(Listing 6). Several parameters, characteristic of the model, should be defined with the "parameter" keyword, while any other variable used for holding intermediate calculations and data should be defined with an appropriate data type, e.g. as real variable(Listing 7). All the quantities used in the model equations are normalized depending on their nature. This way simpler equations occur, which are easier to handle computationally. For model parameter definition and more information refer to Table 1.

Listing 6. Beginning module definition.

```
module ekv_mos (d,g,s,b);
inout d,g,s,b;
electrical d ,g ,s ,b ; //input-output nodes
electrical di,gi,si,bi; //intrinsic nodes
…
endmodule
```

Listing 7. Parameter definition and real variable declaration.

```
…
parameter integer SIGN=1.0      from [-1:1] exclude 0;
parameter real    W=1e-5        from [0:inf];
parameter real    L=1e-5        from [0:inf];
parameter real    VTO=0.5       from [0:inf];
parameter real    PHI=0.9       from [0:inf];
parameter real    GAMMA=0.9     from [0:inf];
parameter real    KP=100e-6     from [0:inf];
parameter real    COX=11.50e-3  from [0:inf];
parameter real    UCRIT=3.8e6 from [0:inf];
parameter real    XJ=50.00e-9   from [0:inf];
parameter real    IBA=0.0       from [0:inf];
parameter real    IBB=300e6     from [0:inf];
parameter real    IBN=1.0       from [0:inf];
parameter real    RL=1.0e-6     from [0:inf];
parameter real    CGXO=1.0e-4   from [0:inf];

real VC,VDSS, VD,VG,VS,VG_PRIME,VP,n,BETA,Ispec,vps,vpd;
real qr,qf,z0,zk,yk,if_,ir,IDS,xf,xr;
real Q0,QS,QD,QI,QG,QB,nq,dqd,dqs,dqb,dqg;
real LC,V_IB,IDB;
```

Table 1. EKV model parameter and variable definitions. The Verilog-A code of several equations is also shown.

SIGN	Defines the MOS type. Specifically, SING=1 corresponds to nMOS and SING=-1 to pMOS.
W, L	W is the channel width and L is the channel length.
VTO	Threshold voltage of the transistor.
PHI	Twice the Fermi potential of the substrate plus a few UT.
GAMMA	Body factor coefficient. Its value can be given or calculated from other process parameters.
KP	It is the product of carrier mobility by the W/L ratio, KP = mu*W/L;
COX	The capacitance per surface are of the gate oxide. Its value is inversely proportional to the oxide thickness.
XJ	Junction depth
RL	It is a parameter used to calculate the ohmic resistance seen at the drain and source terminals. Its value is given in Ohm*m.
CGXO	Represents the external capacitance seen between the gate and the drain and source terminals. Its value is expressed in Farad/m.
UT	Thermodynamic potential, UT=k*T/q; $vtin Verilog-A.
VS	Source voltage, VS = SIGN*V(si,bi);
VD	Drain voltage, VD = SIGN*V(di,bi);
VG	Gate voltage, VG = SIGN*V(gi,bi);
VC	Velocity saturation voltage, VC = UCRIT*L;
VG_PRIME	VG_PRIME = VG - VTO + PHI +GAMMA*sqrt(PHI);
VP	Pinch-off voltage. VP = VG_PRIME - PHI - GAMMA*(sqrt(VG_PRIME+GAMMA*GAMMA/4)-GAMMA/2);
VDSS	VDSS = VC*(sqrt(0.25+($vt*sqrt(if_)/VC))-0.5);
n	Slope factor, n=1+GAMMA/(2.0*sqrt(PHI+VP));
KP	Transconductance parameter
BETA	BETA = KP*W/L;
nq	Slope factor used in the quasi-static model, nq = 1 + 0.5*GAMMA/sqrt(PHI+0.5*VP+1e-6);
Q0	Normalization constant for charges, Q0 = 2*nq*$vt*COX;
QS	Source charge. In quasi-static modeling: QS= -(0.5*Q0*W*L)*((4.0/15.0)* ((3*pow(xf,3)+6*pow(xf,2)*xr+4*pow(xr,2)*xf+2*pow(xr,3))/pow(xf+xr,2)-0.5));
QD	Drain charge, In quasi-static modeling: QD= -(0.5*Q0*W*L)*((4.0/15.0)* ((3*pow(xr,3)+6*pow(xr,2)*xf+4*pow(xf,2)*xr+2*pow(xf,3))/pow(xf+xr,2)-0.5));
QI	Inversion region charge, QI = QS+QD;
QG	Gate charge, QG= - QI - QB;
QB	Bulk charge. In quasi-static modeling: QB= - GAMMA*COX*W*L***sqrt**(VP+PHI) - (nq-1)*QI/nq;
dqd	Charge time-derivative in drain, dqd=ddt(QB);
dqs	Charge time-derivative in source, dqs=ddt(QS);
dqb	Charge time-derivative in bulk, dqb=ddt(QB);
LC	LC = sqrt(`MY_P_ESI*XJ/COX);
V_IB	V_IB = VD - VS - IBN*2*VDSS;

IDB	if (V_IB>0) IDB=IDS*V_IB*exp(-IBB*LC/V_IB)*IBA/IBB;
	else IDB=0;

Ispec	Normalization constant for currents,Ispec = 2*n*BETA*$vt*$vt;
vps	Normalized VP-VS,vps = (VP-VS)/$vt;if (vps<-18) vps=-18;
vpd	Normalized VP-VD, vpd = (VP-VD)/$vt;if (vpd<-20) vpd=-20
qr	Normalized drain charge
qf	Normalized source charge
if_	Normalized forward current, if_ = qf*qf+qf;
ir	Normalized reverse current, ir = qr*qr+qr;
IDS	IDS = Ispec * (if_-ir);
xf	xf= sqrt(0.25+if_);
xr	xr= sqrt(0.25+ir);

Finally and most importantly, Verilog-A equations connecting the device model with the rest of the circuit are presented by (59)-(64) for the extrinsic part, and by (65)-(69) for the intrinsic part of the device. Listing 8, shows the complete model developed for the purpose of the current work.

I(d,di) <+ (W/RL)*V(d,di);   (59)

```
I(s,si) <+ (W/RL)*V(s,si);                        (60)
V(g,gi) <+ 0;   //zero resistance                 (61)
V(b,bi) <+ 0;   //zero resistance                 (62)
I(d,g) <+ (CGXO*W)*ddt(V(d,g));                   (63)
I(s,g) <+ (CGXO*W)*ddt(V(s,g));                   (64)
I(di,si) <+ IDS;                                  (65)
I(di,bi) <+ IDB;                                  (66)
I(di,gi) <+ dqd;                                  (67)
I(si,gi) <+ dqs;                                  (68)
I(bi,gi) <+ dqb;                                  (69)
```

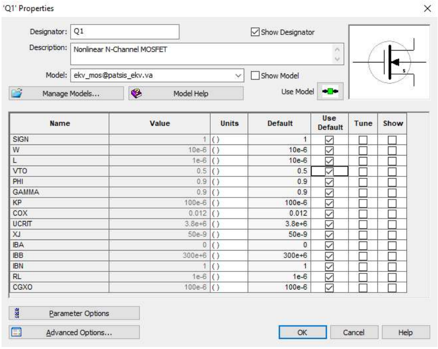
Listing 8. Complete Verilog-A code of the simple EKV MOS model.

```
`include "disciplines.vams"
`include "constants.vams"
`include "compact.vams"
`define   MY_P_ESI   11.7*`P_EPS0

module ekv_mos (d,g,s,b);
inoutd,g,s,b;
electrical d ,g ,s ,b ; //input-output nodes
electrical di,gi,si,bi; //intrinsic nodes

parameter integer SIGN=1.0      from [-1:1] exclude 0;
parameter real    W=1e-5        from [0:inf];
parameter real    L=1e-5        from [0:inf];
parameter real    VTO=0.5       from [0:inf];
parameter real    PHI=0.9       from [0:inf];
parameter real    GAMMA=0.9     from [0:inf];
parameter real    KP=100e-6     from [0:inf];
parameter real    COX=11.50e-3  from [0:inf];
parameter real    UCRIT=3.8e6 from [0:inf];
parameter real    XJ=50.00e-9   from [0:inf];
parameter real    IBA=0.0       from [0:inf];
parameter real    IBB=300e6     from [0:inf];
parameter real    IBN=1.0       from [0:inf];
parameter real    RL=1.0e-6     from [0:inf];
parameter real    CGXO=1.0e-4   from [0:inf];

real VC,VDSS, VD,VG,VS,VG_PRIME,VP,n,BETA,Ispec,vps,vpd;
real qr,qf,z0,zk,yk,if_,ir,IDS,xf,xr;
//qs
real Q0,QS,QD,QI,QG,QB,nq,dqd,dqs,dqb,dqg;
//IDB
real LC,V_IB,IDB;

analog

begin

//extrinsic part
I(d,di) <+ (W/RL)*V(d,di);
I(s,si) <+ (W/RL)*V(s,si);
V(g,gi) <+ 0;   //zero resistance
V(b,bi) <+ 0;   //zero resistance
I(d,g) <+ (CGXO*W)*ddt(V(d,g));
I(s,g) <+ (CGXO*W)*ddt(V(s,g));

//intrinsic part
VD = SIGN*V(di,bi);
```

```
VG = SIGN*V(gi,bi);
VS = SIGN*V(si,bi);

//CALCULATE PINCH-OFF VOLTAGE VP
VG_PRIME = VG - VTO + PHI +GAMMA*sqrt(PHI);
VP = VG_PRIME - PHI - GAMMA*(sqrt(VG_PRIME+GAMMA*GAMMA/4)-GAMMA/2);

//CALCULATE SLOPE COEFFICIENT n
n=1+GAMMA/(2.0*sqrt(PHI+VP));

//CALCULATE NORMALIZING CURRENT COEFFICIENT
BETA = KP*W/L;
Ispec = 2*n*BETA*$vt*$vt;

//CALCULATION OF FORWARD CURRENT COMPONENT (SOURCE TERMINAL)
//INVERSION OF FUNCTION F, USING THE ITERATIVE NEWTON-RAPHSON METHOD
vps = (VP-VS)/$vt;
if (vps<-18) vps=-18; //PREVENT OPERATION WITH VERY SMALL ARGUMENTS
z0 = (vps>-0.35) ? 2/(1.3+vps-ln(vps+1.6)) : 1.55+exp(-vps);
zk = (2+z0)/(1+vps+ln(z0));
yk = (1+vps+ln(zk))/(2+zk);
qf = yk;
if_ = qf*qf+qf;

//CALCULATION OF REVERSE CURRENT COMPONENT (DRAIN TERMINAL)
//INVERSION OF FUNCTION F, USING THE ITERATIVE NEWTON-RAPHSON METHOD
vpd = (VP-VD)/$vt;
if (vpd<-20) vpd=-20; //PREVENT OPERATION WITH VERY SMALL ARGUMENTS
z0 = (vpd>-0.35) ? 2/(1.3+vpd-ln(vpd+1.6)) : 1.55+exp(-vpd);
zk = (2+z0)/(1+vpd+ln(z0));
yk = (1+vpd+ln(zk))/(2+zk);
qr = yk;
ir = qr*qr+qr;

//IDS – CHANNEL CURRENT CALCULATION
IDS = Ispec * (if_-ir);

// IDB-IMPACT IONIZATION CURRENT CALCULATION
//velocity saturation voltage
VC = UCRIT*L;
VDSS = VC*(sqrt(0.25+($vt*sqrt(if_)/VC))-0.5);
LC = sqrt(`MY_P_ESI*XJ/COX);
V_IB = VD - VS - IBN*2*VDSS;
if (V_IB>0) IDB=IDS*V_IB*exp(-IBB*LC/V_IB)*IBA/IBB;
else        IDB=0;

//QUASI-STATICE BEHAVIOR
xf= sqrt(0.25+if_);
xr= sqrt(0.25+ir);
nq = 1 + 0.5*GAMMA/sqrt(PHI+0.5*VP+1e-6);
Q0 = 2*nq*$vt*COX;
QD= -(0.5*Q0*W*L)*((4.0/15.0)*
((3*pow(xr,3)+6*pow(xr,2)*xf+4*pow(xf,2)*xr+2*pow(xf,3))/pow(xf+xr,2)-0.5));
QS= -(0.5*Q0*W*L)*((4.0/15.0)*
((3*pow(xf,3)+6*pow(xf,2)*xr+4*pow(xr,2)*xf+2*pow(xr,3))/pow(xf+xr,2)-0.5));
QI = QS+QD;
QB= - GAMMA*COX*W*L*sqrt(VP+PHI) - (nq-1)*QI/nq;
// QG= - QI - QB;
dqd=ddt(QB);
```

```
dqs=ddt(QS);
dqb=ddt(QB);

I(di,si) <+ IDS;
I(di,bi) <+ IDB;
I(di,gi) <+ dqd;
I(si,gi) <+ dqs;
I(bi,gi) <+ dqb;

end
endmodule
```

The model parameters are the quantities that can be user-changed and tuned after the model is integrated into the simulator (Genesys in this case), and they will appear when the model icon is double-clicked, permitting the user to change their values as seen in Fig. 3. The model is saved in a file (e.g., patsis_ekv.va). This file will be loaded into Genesys library for compilation in order to be used as a functional model of the MOSFET. The process to do so is relatively easy. From the menu Tools, click on Library manager (Fig. 4) and then select Add From File… button to navigate to the file where the model is saved (partis_ekv.va in this case) as seen in Fig. 5. If the code is error free then the compilation will be successful and a directory named "Compiled" will be created in the same directory as the "*.va" file, containing a "cml" and a "xml" file with the same name as the "va" file. These files could be copied in the "Complied" directory of the "VerilogA" directory of the Genesys installation directory. Doing so, open then a schematic sheet in a Genesys workspace, insert a MOS symbol, e.g. the nMOS Q1 shown in Fig. 4 and double click it. The library manager will open, listing all the available MOS models including the newly created one (patsis_ekv.va) as seen in Fig. 6. The model is ready for use in simulations. One DC simulation is to test the I_{DS} vs. V_{GS} for various V_{DS} (Fig. 7a,b) and another one to test I_{DS} vs. V_{DS} for various V_{GS} (Fig. 7c,d), verify the validity of the developed model.



'Q1' Properties

Designator: Q1 ☑ Show Designator

Description: Nonlinear N-Channel MOSFET

Model: ekv_mos@patsis_ekv.va ☐ Show Model

Manage Models... | Model Help | Use Model

Name	Value	Units	Default	Use Default	Tune	Show
SIGN	1	()	1	☑	☐	☐
W	10e-6	()	10e-6	☑	☐	☐
L	1e-6	()	10e-6	☑	☐	☐
VTO	0.5	()	0.5	☑	☐	☐
PHI	0.9	()	0.9	☑	☐	☐
GAMMA	0.9	()	0.9	☑	☐	☐
KP	100e-6	()	100e-6	☑	☐	☐
COX	0.012	()	0.012	☑	☐	☐
UCRIT	3.8e+6	()	3.8e+6	☑	☐	☐
XJ	50e-9	()	50e-9	☑	☐	☐
IBA	0	()	0	☑	☐	☐
IBB	300e+6	()	300e+6	☑	☐	☐
IBN	1	()	1	☑	☐	☐
RL	1e-6	()	1e-6	☑	☐	☐
CGXO	100e-6	()	100e-6	☑	☐	☐

Parameter Options | Advanced Options... | OK | Cancel | Help

Figure 3. Integrated EKV model MOS shows its parameters, which can be modified and tuned in simulations.

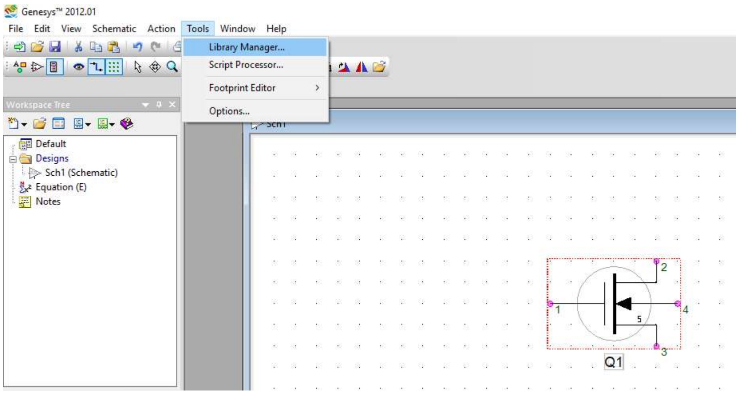
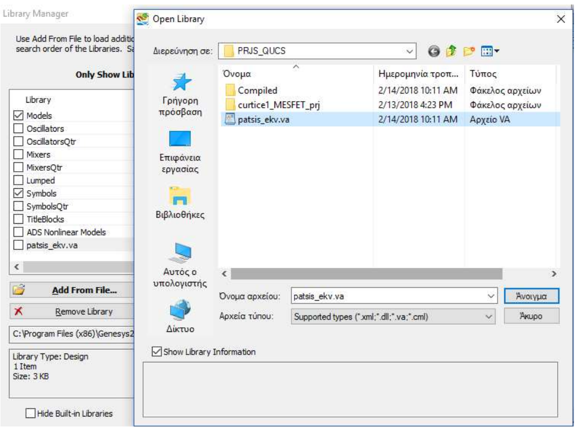


Figure 4. Place a MOS instance in schematic, selected, and call Tools➔Library Manager to set the model.



Library Manager

Use Add From File to load additic search order of the Libraries. Sa

Only Show Lib

Library
☑ Models
□ Oscillators
□ OscillatorsQtr
□ Mixers
□ MixersQtr
□ Lumped
☑ Symbols
□ SymbolsQtr
□ TitleBlocks
□ ADS Nonlinear Models
□ patsis_ekv.va

Add From File...
Remove Library

C:\Program Files (x86)\Genesys2

Library Type: Design
1 Item
Size: 3 KB

□ Hide Built-in Libraries

Open Library

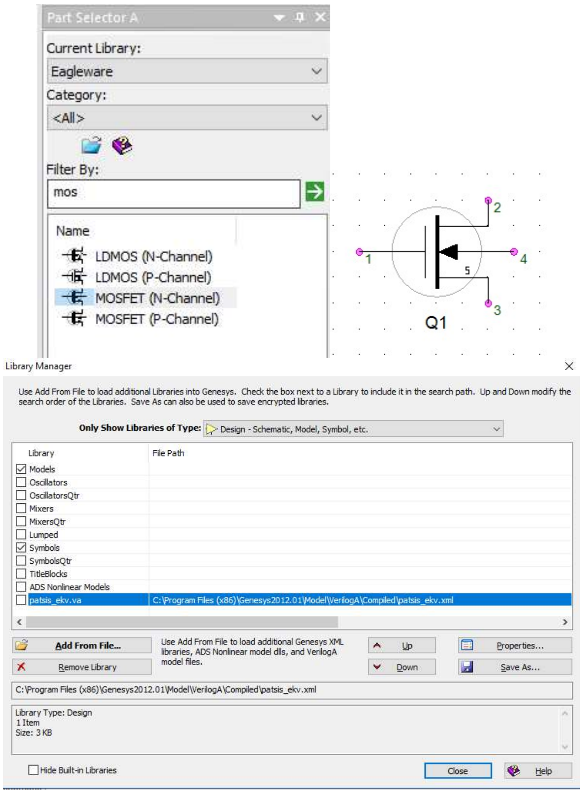
Διερεύνηση σε: PRJS_QUCS

Όνομα	Ημερομηνία τροπ...	Τύπος
Compiled	2/14/2018 10:11 AM	Φάκελος αρχείων
curtice1_MESFET_prj	2/13/2018 4:23 PM	Φάκελος αρχείων
patsis_ekv.va	2/14/2018 10:11 AM	Αρχείο VA

Όνομα αρχείου: patsis_ekv.va
Αρχεία τύπου: Supported types (*.xml;*.dll;*.va;*.cml)

☑ Show Library Information

Figure 5. In Library Manager, select Add From File to navigate to the va file containing the Verilog-A code of the model.



Part Selector A

Current Library: Eagleware
Category: <All>
Filter By: mos

Name
LDMOS (N-Channel)
LDMOS (P-Channel)
MOSFET (N-Channel)
MOSFET (P-Channel)

Library Manager

Use Add From File to load additional Libraries into Genesys. Check the box next to a Library to include it in the search path. Up and Down modify the search order of the Libraries. Save As can also be used to save encrypted libraries.

Only Show Libraries of Type: Design - Schematic, Model, Symbol, etc.

Library	File Path
☑ Models	
□ Oscillators	
□ OscillatorsQtr	
□ Mixers	
□ MixersQtr	
□ Lumped	
☑ Symbols	
□ SymbolsQtr	
□ TitleBlocks	
□ ADS Nonlinear Models	
□ patsis_ekv.va	C:\Program Files (x86)\Genesys2012.01\Model\VerilogA\Compiled\patsis_ekv.xml

Add From File...
Remove Library

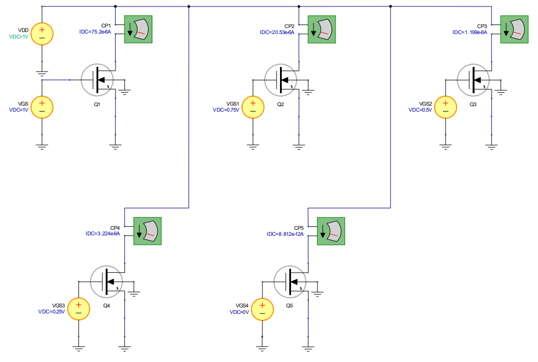
Use Add From File to load additional Genesys XML libraries, ADS Nonlinear model dlls, and VerilogA model files.

C:\Program Files (x86)\Genesys2012.01\Model\VerilogA\Compiled\patsis_ekv.xml

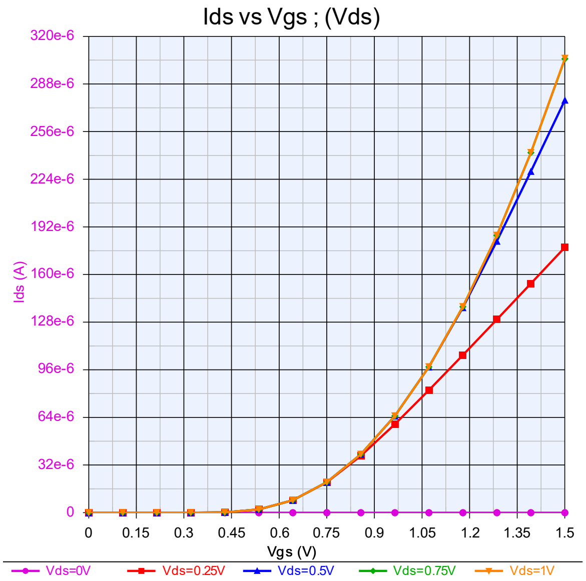
Library Type: Design
1 Item
Size: 3 KB

□ Hide Built-in Libraries

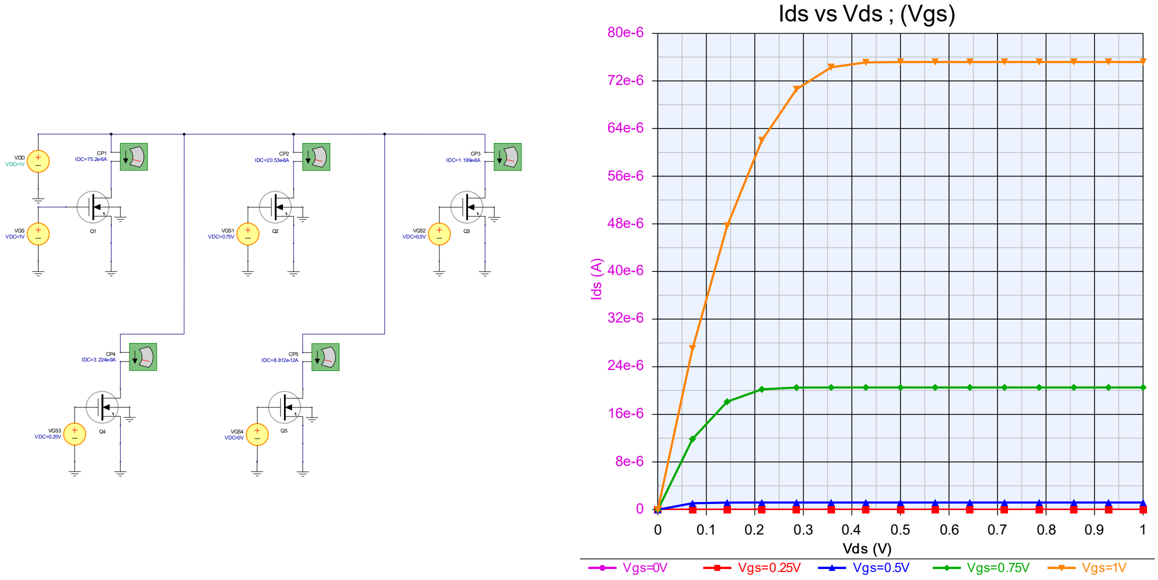
Figure 6. After model compilation, this can be selected to represent the MOS instance behavior.



(a)



(b)



(c) **(d)**

Figure 7. Simulation to test the I_{DS} vs. V_{GS} for various V_{DS} (Fig. 7a,b) and another one to test I_{DS} vs. V_{DS} for various V_{GS} (Fig. 7c,d).

Conclusions And Future Work

A simple version of the EKV MOSFET model is implemented in Verilog-A and tested in Keysight's Genesys software suite. Using the presented methodology, various model parameter effect on the *IV* curves, can be tested. In the simple model developed here, all of its parameters can be set to test by keeping each time all constant but one, and creating instances of the transistor with this parameter having different value. Then a sweep analysis can give the corresponding *IV* curves showing the effect of the parameter's variation. Specifically, the following parameters can be varied in the current EKV model of the MOS transistor: channel width, channel length, long-channel threshold voltage, bulk Fermi potential(x2), body effect parameter, transconductance parameter, gate oxide capacitance per unit area, longitudinal critical field, junction depth, first impact ionization coefficient, second impact ionization coefficient, and saturation voltage factor for impact ionization. Future work could advance this simple EKV model with the inclusion of several parasitic capacitances in order to test its frequency response.

References

[1] . C. C. Enz, F. Krummenacher, E. A. Vittoz, *An Analytical MOS Transistor Model Valid in All Regions of Operation and Dedicated to Low-Voltage and Low-Current Applications,* Analog Integrated Circuits and Signal Processing, 8, 1994, 83-114.

[2] . A. S. Bazigos, *Implementation of EKV MOSFET model in Verilog-A*, degree thesis, National Technical University of Athens, 2003.

[3] . C. C. Enz, E. A. Vittoz, *Charge-based MOS Transistor Modeling. The EKV model for low-power and RFIC design*, Wiley, 2006.

[4] . M. Bucher, C. Lallement, C. Enz, F. Theodoloz, F. Krummenancher, *The EPFL-EKV MOSFET Model Equations for Simulation,* Technical Report, Model Revision 2.6, 1997.

George P. Patsis "MOSFET EKV Verilog-A Model Implementation in Genesys." IOSR Journal of VLSI and Signal Processing (IOSR-JVSP), vol. 8, no. 1, 2018, pp. 73-86.