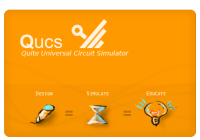
Equation-Defined Template and Synthesis Driven Compact Modelling of Semiconductor Devices

Mike Brinson [1], mbrin72043@yahoo.co.uk.

[1]Centre for Communications Technology, London Metropolitan University, UK

Presented at IEEE EDS Distinguished Lecturer Mini-Colloquium on "Nanoelectronics - Technology, Design and Modelling" held at Rzeszów, Poland, on June 26, 2019.



Introduction

- Current rapid developments in new semiconductor materials, device structures and circuit design techniques have intensified the need for improved circuit simulation and data analysis tools.
- For greatest impact on the compact modelling community these tools should be freely available, preferable under the FOSS licence.
- This presentation outlines a number of current trends in compact device modelling based on FOSS circuit simulation and device modelling packages, particular reference is made to interpreted and compiled Equation-Defined template and synthesis driven modelling techniques.
- All the circuit simulation software and compact modelling tools described are freely available for download from the Internet as either binary or source code packages. [A list of Internet Web addresses is given at the end of the presentation slides.]

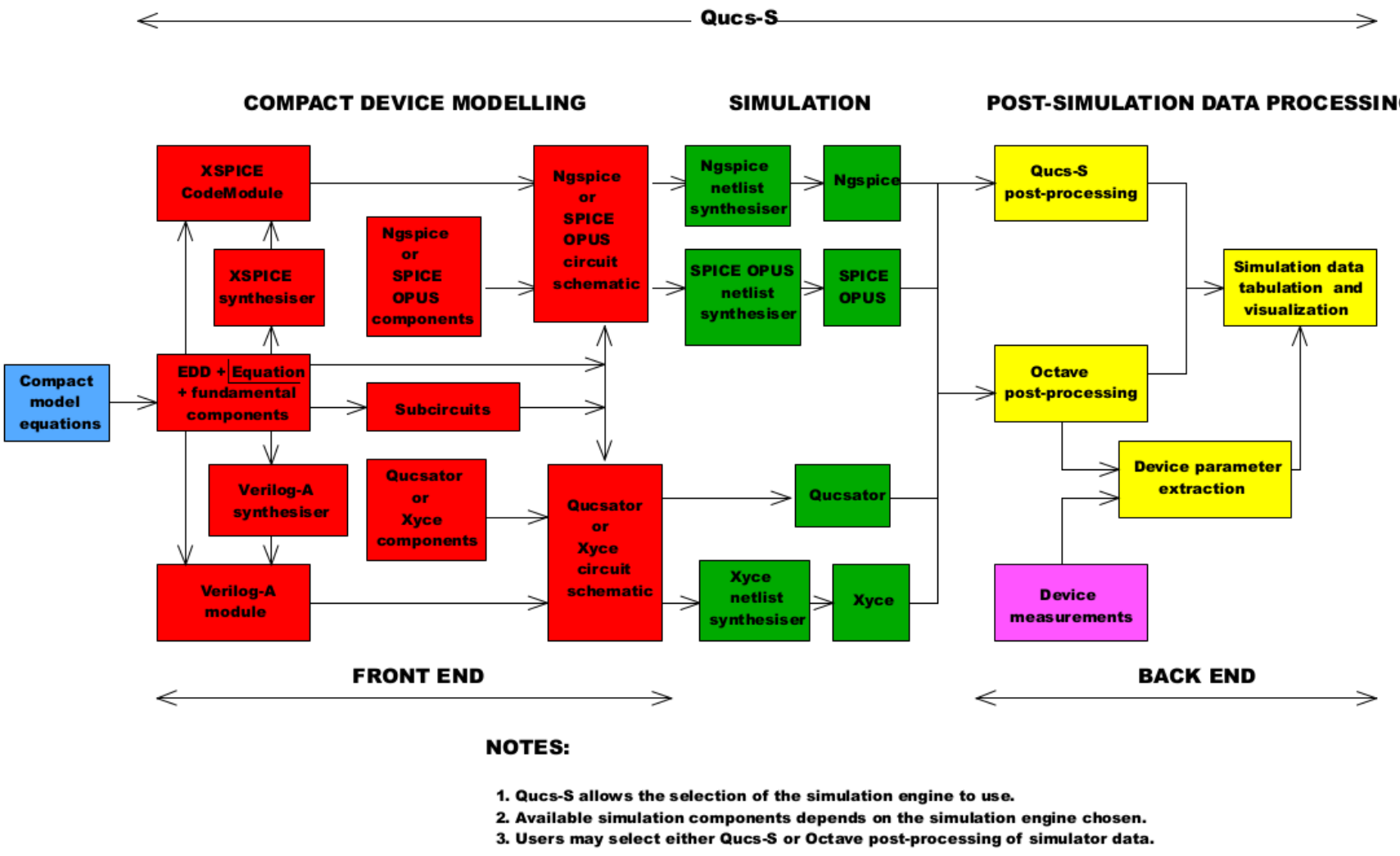
Established Compact Device Modelling and Circuit Simulation Tools

Simulator	Simulation and output commands	Components/models	Modelling features
Berkeley SPICE	.OP, .DC, .TF, .AC, .TRAN, .NOISE., .DISTO, .PZ, .SENS, .SAVE, .PRINT, .PLOT, .FOUR, .SUBCKT .... .ENDS Berkeley Nutmeg simulation data post processor.	R, C, L, M, S, W, V, I, T, LTRA, U, URC, D, BJT, J, M, Z, B	1. Fully expanded circuits, 2. Subcircuits[A], 3. Macromodels[B], 4. C compiled code models[C].
Qucs	DC, AC, transient, harmonic balance[D], Parameter sweep, equations, S-parameter analysis, Qucs and Octave post simulation data processors. Noise analysis. Optimization.	Similar to Berkely SPICE plus RF component and device models, VHDL and Verilog Digital models.	1., 2. and 3. the same as SPICE, 4. EDD and RFEDD behavioural modelling 5. Verilog-A code models[E].
QucsStudio	DC, AC, transient, harmonic balance (including large signal AC and noise), equations, S parameter analysis, Qucs and Octave post simulation data processors, Parameter sweep, transient shooting method periodic steady-state simulation, Monte Carlo analysis	Similar to Berkely SPICE plus RF component and device models, VHDL and Verilog Digital models. Communication system models	1., 2. and 3. the same as SPICE, 4. EDD and RFEDD behavioural modelling 5. Verilog-A and C++ code models[E].

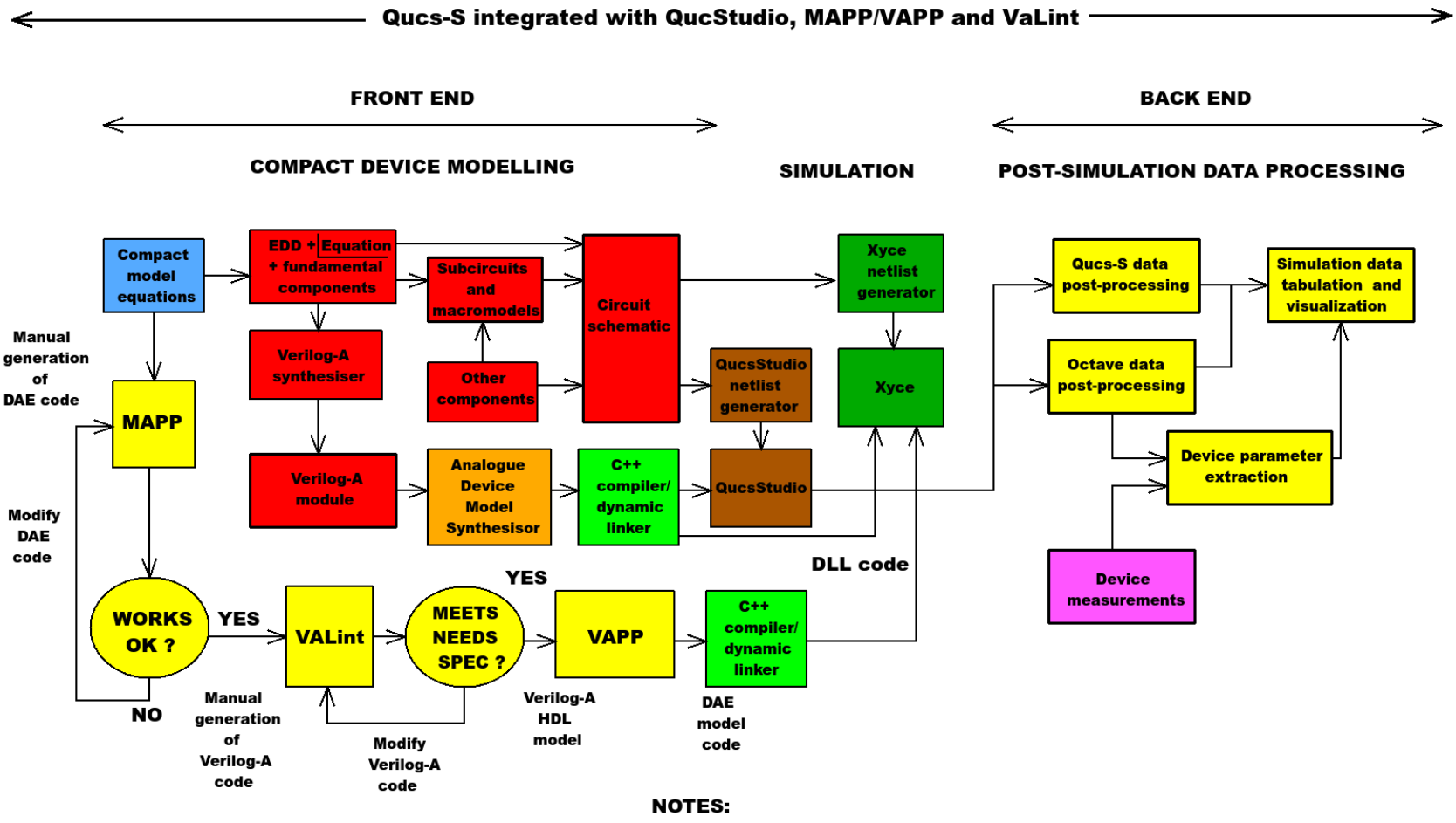
NOTES

A : Linear and non-linear device models based on subcircuits.
B: Linear and non-linear device models constructed from existing component models.
C: SPICE C code model API .
D: Incomplete single tone Harmonic Balance simulation feature.
E : ADMS Verilog-A compact modelling feature with "Turn-key" capabilities.

Qucs-S: A Qucs Fork for Compact Device Modelling and Circuit Simulation



Merging Qucs-S, QucsStudio and Xyce/MAPP/VAPP/VAlint Verilog-A Compact Modelling Tools



NOTES:

1. Qucs-S allows the selection of the simulation engine to use.
2. Available simulation components depends on the simulation engine chosen.
3. Users may select either Qucs-S or Octave post-processing of simulator data.
4. Berkeley MAPP/VAPP and VaLint requires that Octave 4.0 or greater is installed.

More detailed instructions for Verilog-A and DAE model generation and a number of examples are presented in later slides.

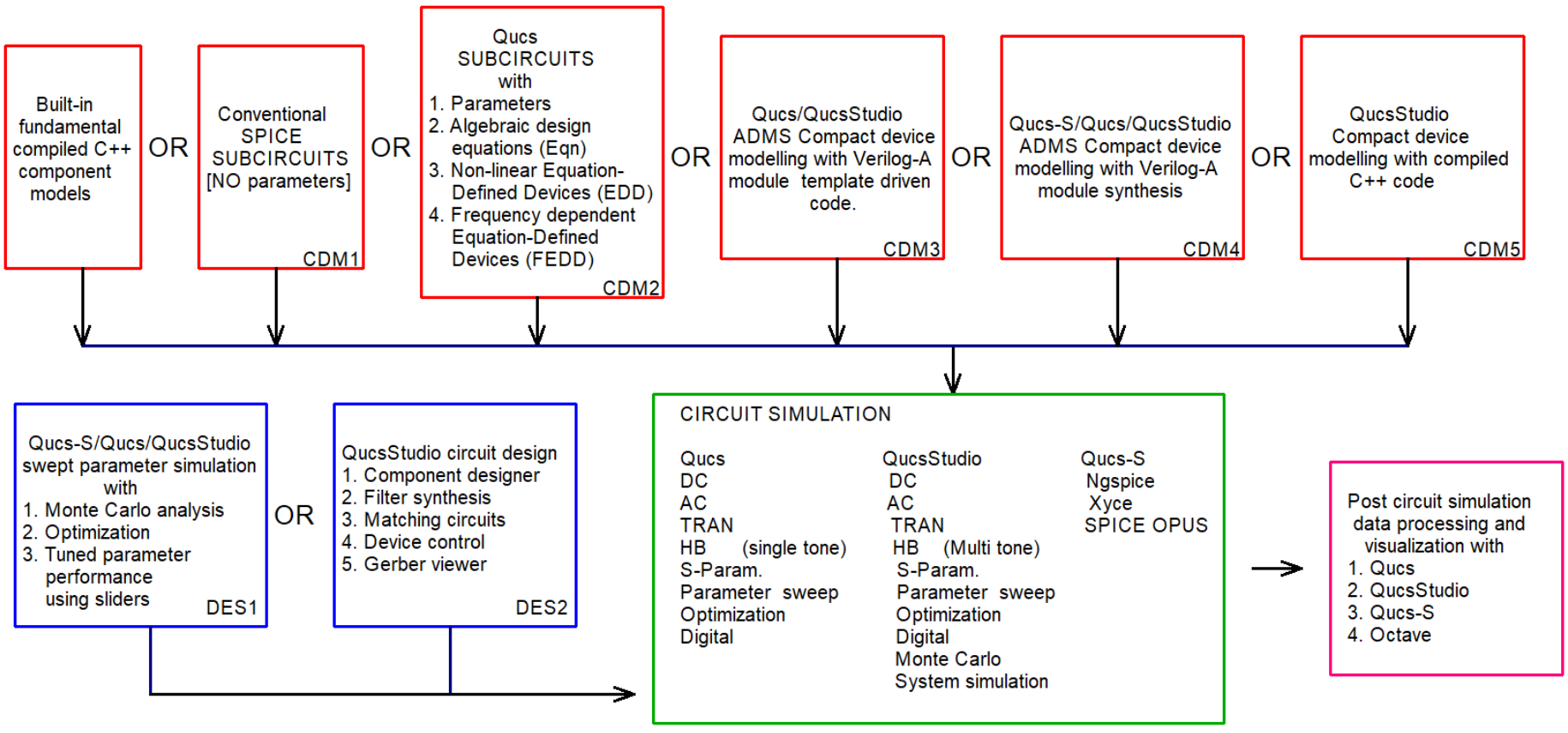
Qucs Timeline 2003 to 2019, Listing Compact Modelling Development

QucsStudio Timeline 2003 to 2018, Listing Primary Compact Modelling Development

New FOSS/GPL SPICE Based Circuit Simulators and Modelling Tools

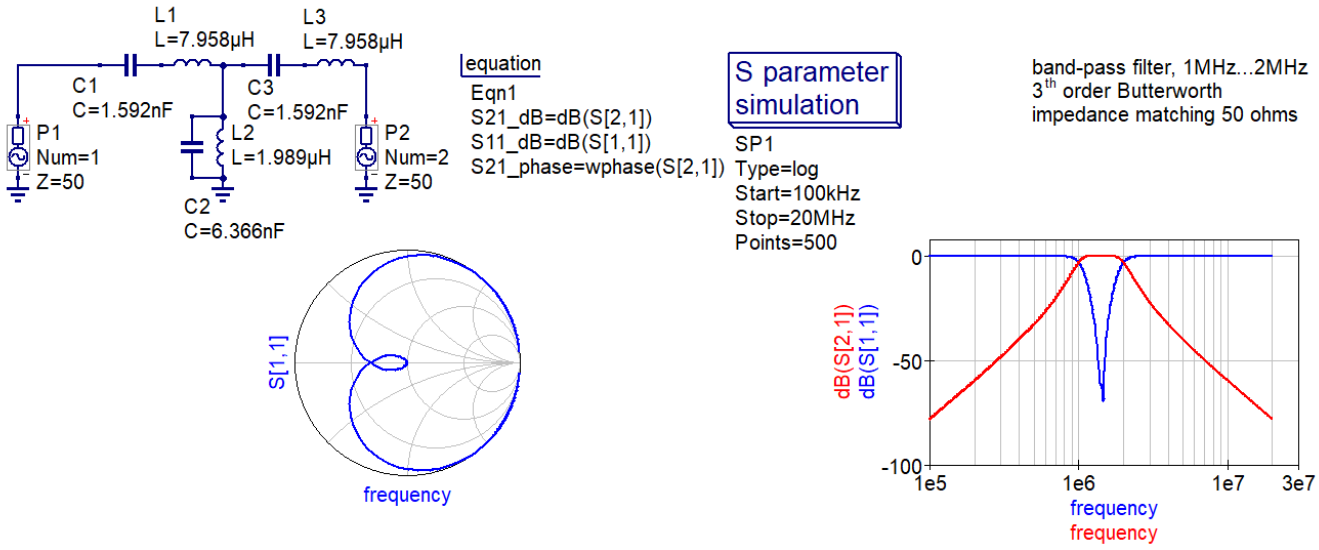
Simulator	Simulation and output commands	components/models	Modelling features
Xyce 6.10 (2018)	.AC, .DATA, .DC, .DCVOLT .END, .ENDS, .FOUR, .FUNC .GLOBAL_PARAM, .HB, .IC, .INC, .LIB, .MEASURE, .MODEL .NODESET, .NOISE, .OP, .OPTIONS .PARAM, .PRINT, .RESULT, .SAMPLING .SAVE, .SENS, .STEP, .SUBCIR, .TRAN	C, L, R, K, D, I, V, E, F, G, H B, Q, J,Z,M, LTR, S, SW, T Y, U, X, ADMS/VERILOG-A MODELS	Analogue behavioural modelling. Homotopy and continuation models. TCAD (PDE Dvices) simulation
WRspice 4.3.9 (2019)	.DC, .DC SWEEP, .AC, .NOISE TRANSFER FUNCTION, .TRAN OPERATING RANGE, MONTE CARLO	C, L, R, K, S, SW, V, E, H, G, B, Q, D, J, Josephson Junction, MOSFET, MESFET, Run-time loadable Verilog-A models	Xic/WRspice interface with graphical schematic editor Powerful grahical plotting facility Powerful scripting langage and interpreter Digital Verilog parser and language extensions for mixed mode simulaton CMC QA validation Extended precision arithmatic

Qucs/QucsStudio Compact Device Modelling Capabilities: CDM1 to CDM5 and DES1 to DES2

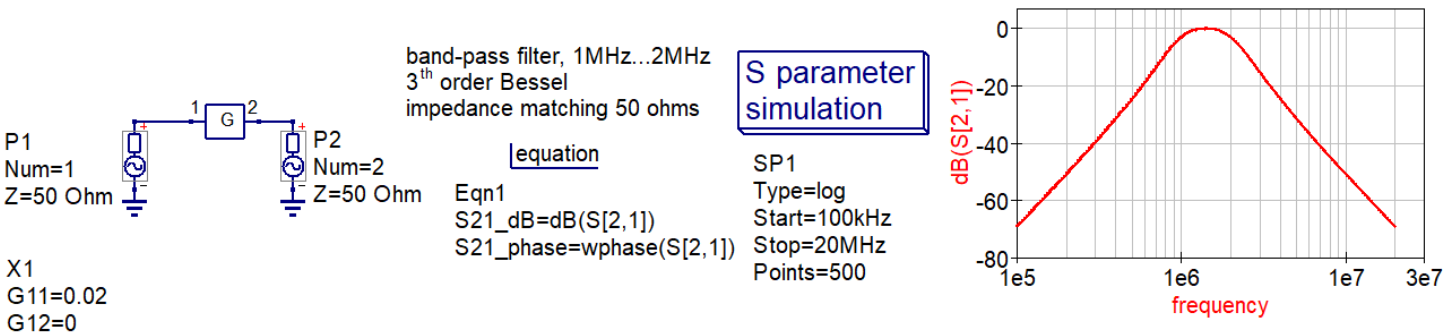


Qucs/QucsStudio Modelling Examples: 1. buit-in components, 2. FEDD models

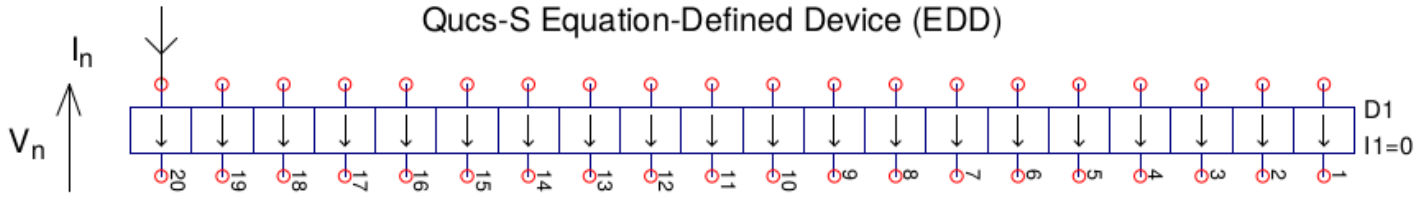
- With fundamental built-in component models



- With Frequency Equation-Defined Device (FEDD) Models



Building Compact Device Models with Qucs/Qucs-S/Qucsstudio: Compact Device Modelling with Equation-Defined Devices (EDD) - Part 1



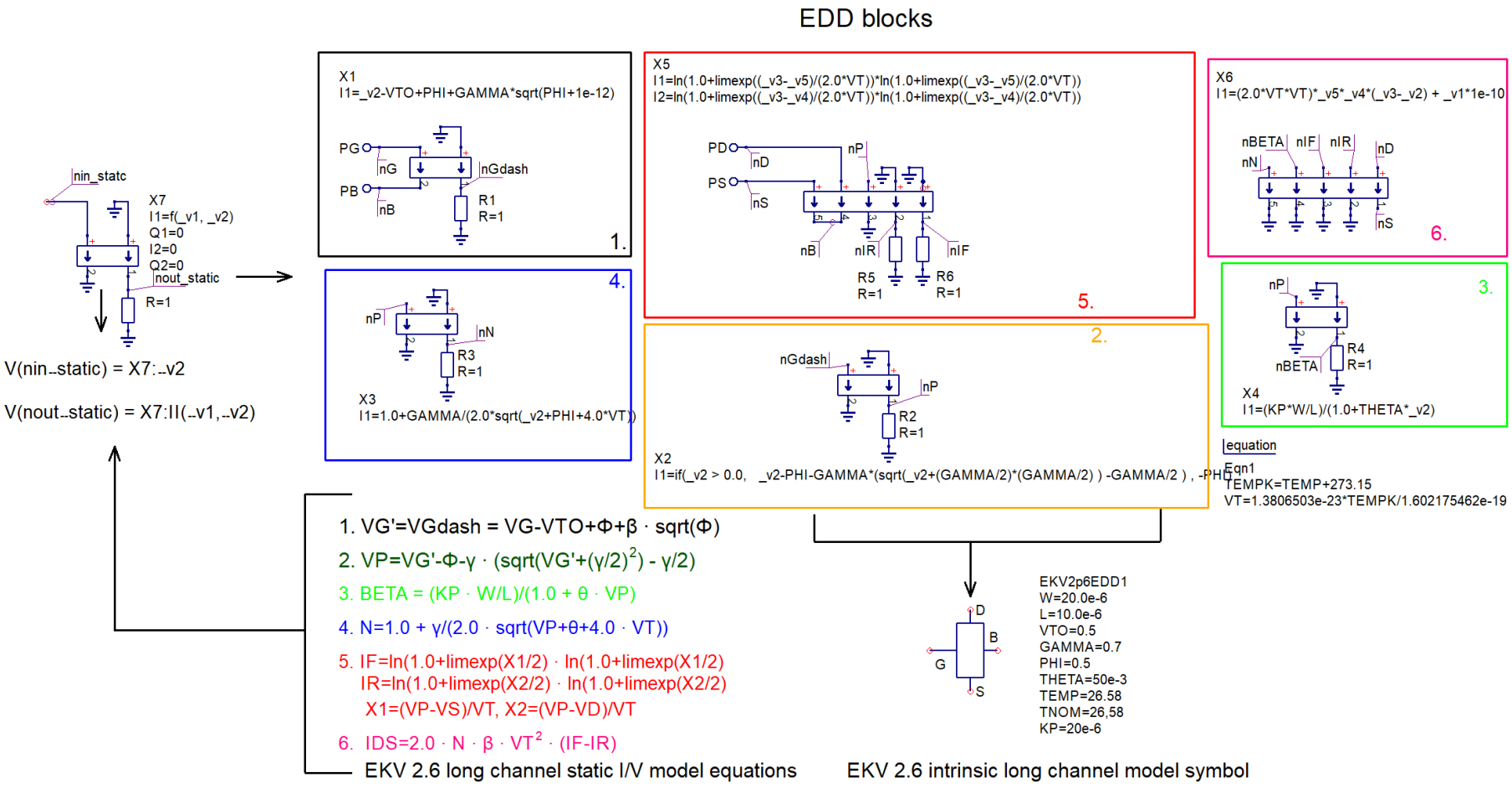
$I = I(V)$, $g = dI/dV$

$Q = Q(V,I)$, $C = dQ/dV = \partial Q(V)/\partial V + \partial Q(I)/\partial I \cdot g$, where

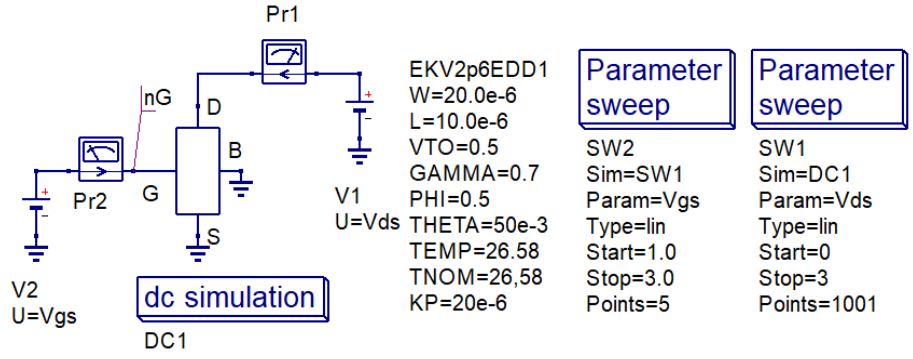
the current flowing in branch n is $I_n = I(V_n) + d/dt(Q_n)$, and $1 <= n <= 20$.

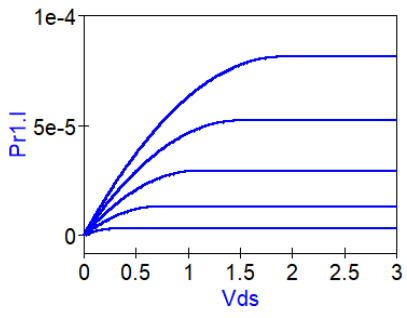
- EDD is a multiterminal nonlinear component with branch currents that can be functions of EDD branch voltage, and stored charge that can be a function of both EDD branch voltages and currents
- EDD is similar, but more advanced to the SPICE 3f5 B type I or V controlled sources
- EDD can be combined with conventional circuit components and Qucs-S equation blocks when constructing compact device models and subcircuit macromodels
- EDD is an advanced component, allowing users to construct prototype experimental models from a set of equations derived from physical device properties
- EDD operator d/dt is undertaken internally by Qucs-S
- Qucs-S EDD can have a maximum of 20 two terminal branches

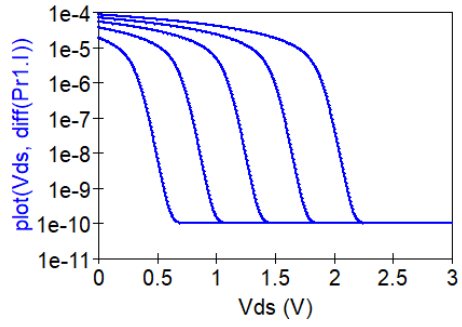
Building Compact Device Models with Qucs/Qucs-S/QucsStudio EDD - Part 2

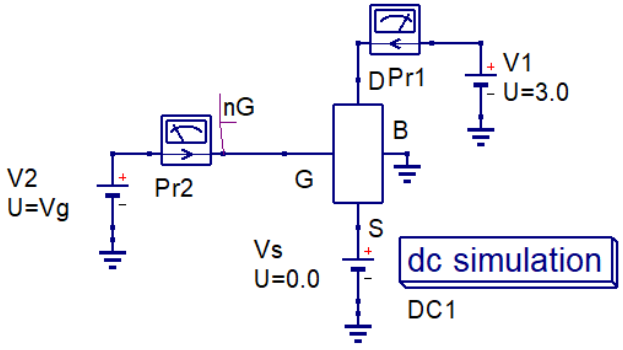


Building Compact Device Models with Qucs/Qucs-S/QucsStudio EDD - Part 3



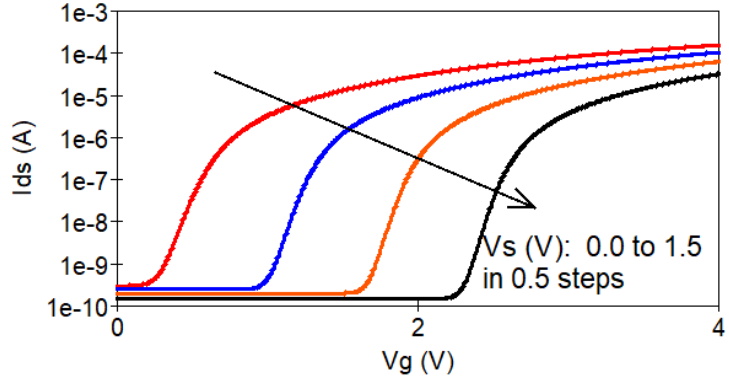




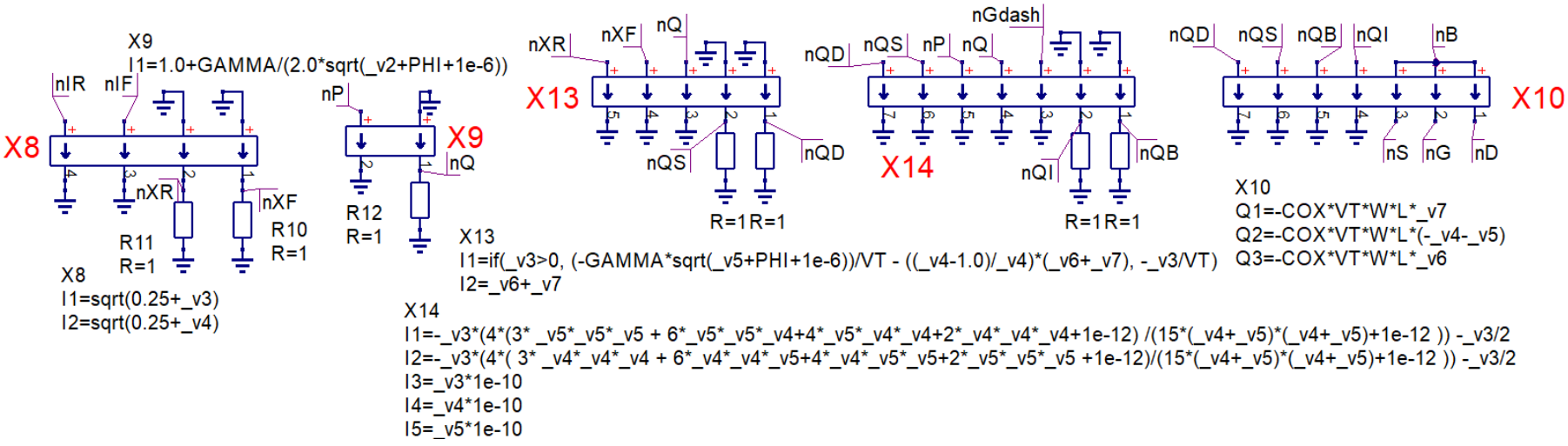


EKV2p6EDD1
W=20.0e-6
L=10.0e-6
VTO=0.5
GAMMA=0.7
PHI=0.5
THETA=50e-3
TEMP=26.58
TNOM=26,58
KP=20e-6

Parameter sweep
SW1
Sim=DC1
Param=Vg
Type=lin
Start=-1.5
Stop=4
Points=1001



Building Compact Device Models with Qucs/Qucs-S/QucsStudio EDD - Part 4



X9 ---> nq (69), X13 ---> qD, qS (72, 73)
X8 ---> Xf, Xr (70, 71), X14 ---> ql, qB (74, 75)

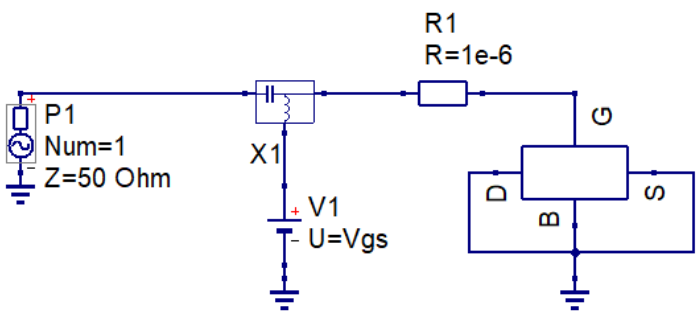
X10 --> I1 = ddt(Qdb), I2 = ddt(Qgb), I3 = ddt(Qsb) (76, 77, 78)

EKV 2.6 model equations in (...), from

Matthias Bucher et al., The EPFL-EKV MOSFET Model Equations for Simulation, Technical Report, Model Version 2.6, June 1997,

Electronics Laboratories, Swiss Federal Institute of Technology (EPFL), Lausanne, Switzerland.

Building Compact Device Models with Qucs/Qucs-S/QucsStudio EDD - Part 5




```
EKV2p6EDD1
W=W
L=L
VTO=0.5
GAMMA=0.7
PHI=0.5
THETA=50e-3
TEMP=26.58
TNOM=26,58
KP=20e-6
COX=COX
```

S parameter simulation

```
SP1
Type=list
Points=100e6
```

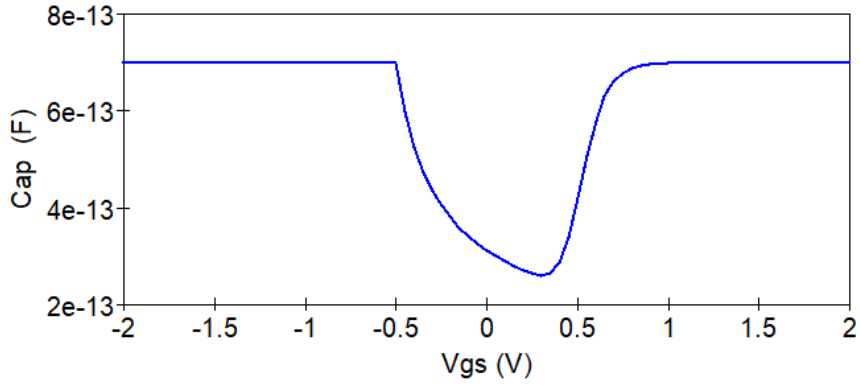
Parameter sweep

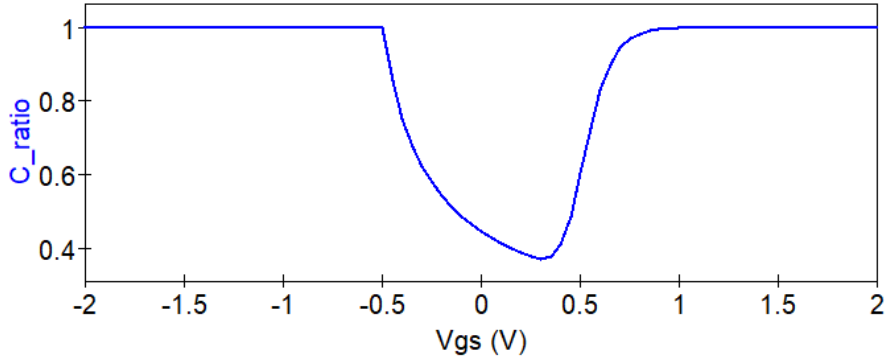
```
SW1
Sim=SP1
Param=Vgs
Type=lin
Start=-2
Stop=2
Points=81
```

Equation

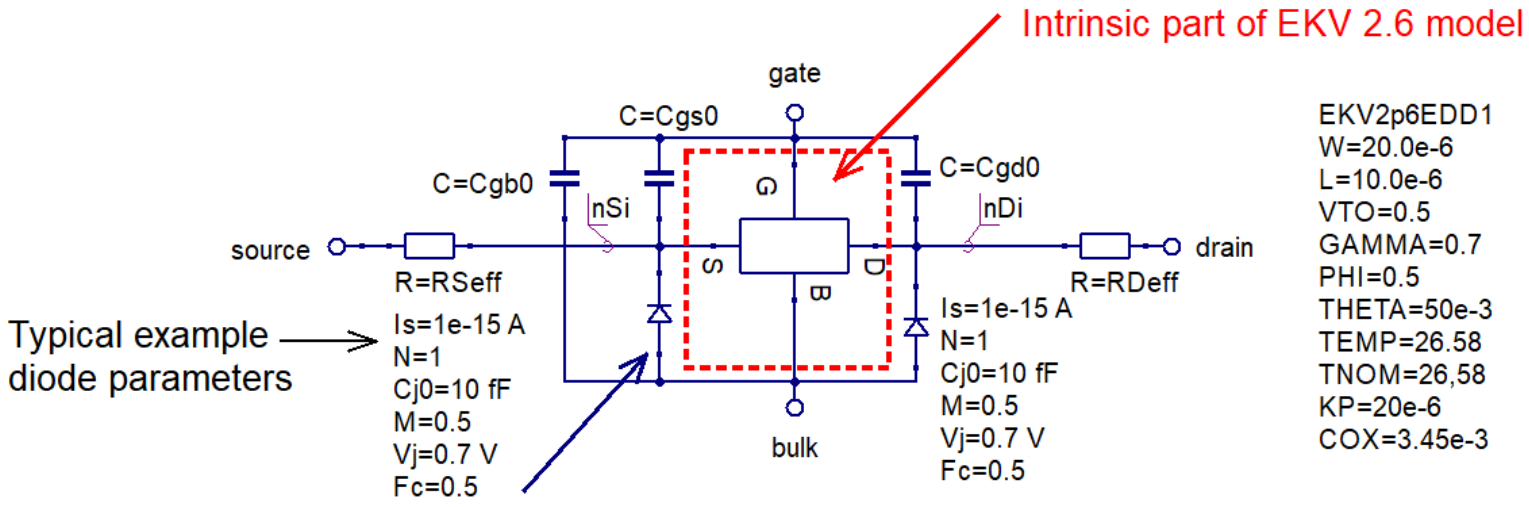
```
Eqn1
y=stoy()
W=20e-6
L=10e-6
COX=3.5e-3
Omega=2*pi*frequency
Cap=imag(y[1,1])/Omega
C_parallel_plate=W*L*COX
C_ratio=Cap/C_parallel_plate
```

number	C_parallel_plate
1	7e-13

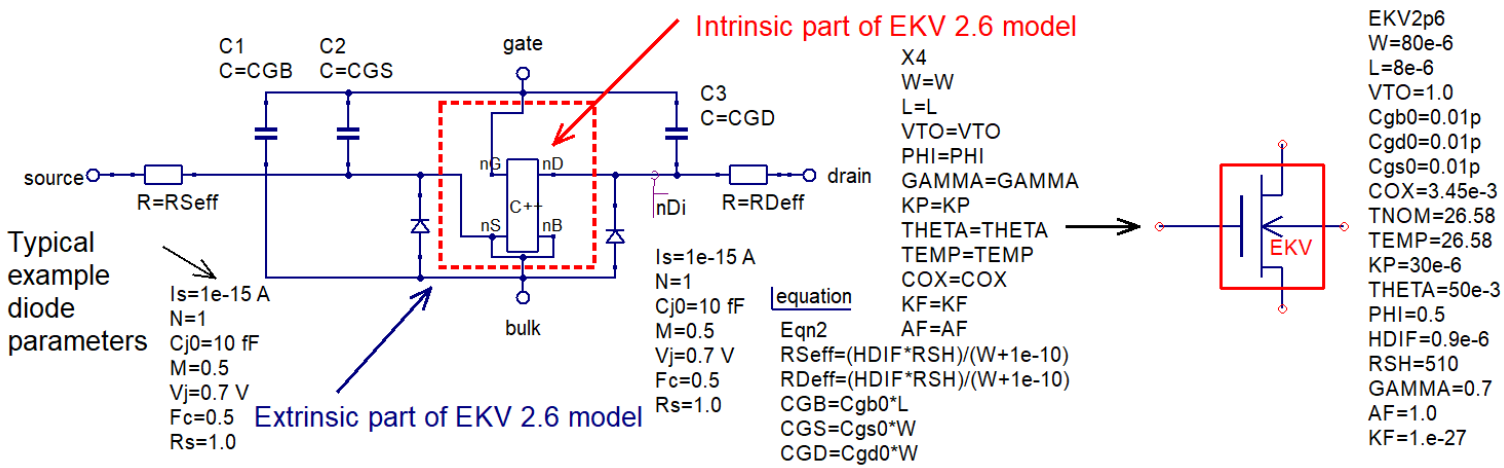




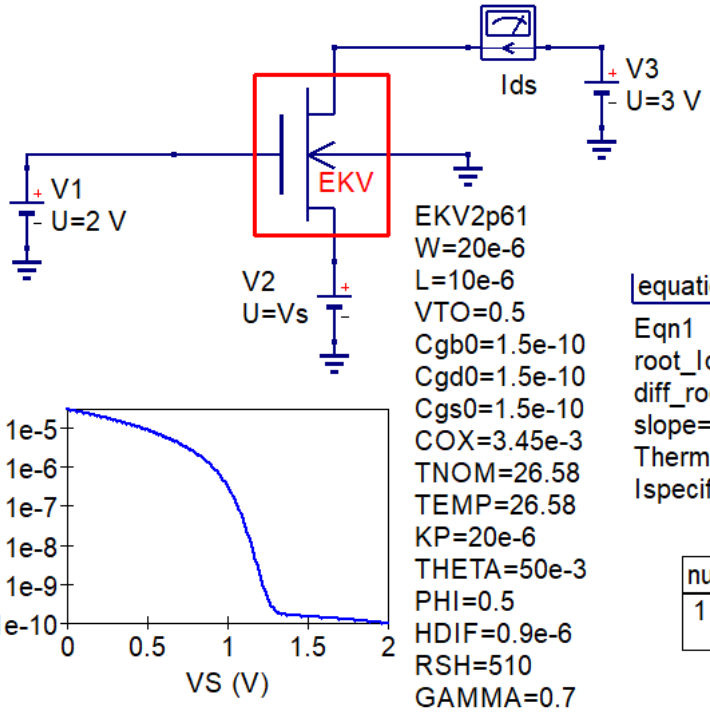
Building Compact Device Models with Qucs/Qucs-S/QucsStudio - the Combined Intrinsic and Extrinsic EDD Long Channel Model



Extrinsic part of EKV 2.6 model



Building Compact Device Models with Qucs/Qucs-S/QucsStudio - Extraction of Ispecific

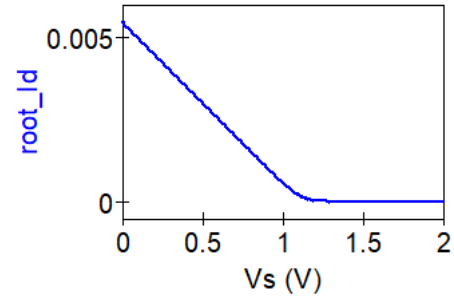


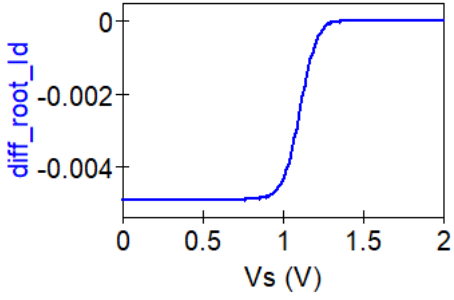
dc simulation
DC1

Parameter sweep
SW1
Sim=DC1
Param=Vs
Type=lin
Start=0
Stop=2
Points=201

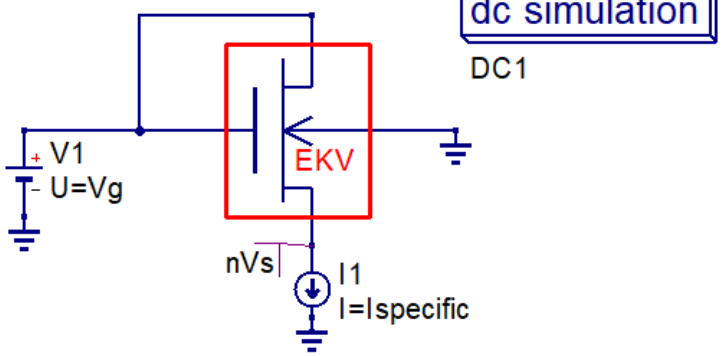
equation
Eqn1
root_Id=sqrt(Ids.I)
diff_root_Id=diff(root_Id)
slope=-max(abs(diff_root_Id))
ThermalVolt=1.3806503e-23*300.0/1.602175462e-19
Ispecific=real((slope^2)*2*ThermalVolt*ThermalVolt)

number	slope	Ispecific
1	-0.0049	3.20678e-08





Building Compact Device Models with Qucs/Qucs-S/QucsStudio - Extraction of VP and Estimation of N

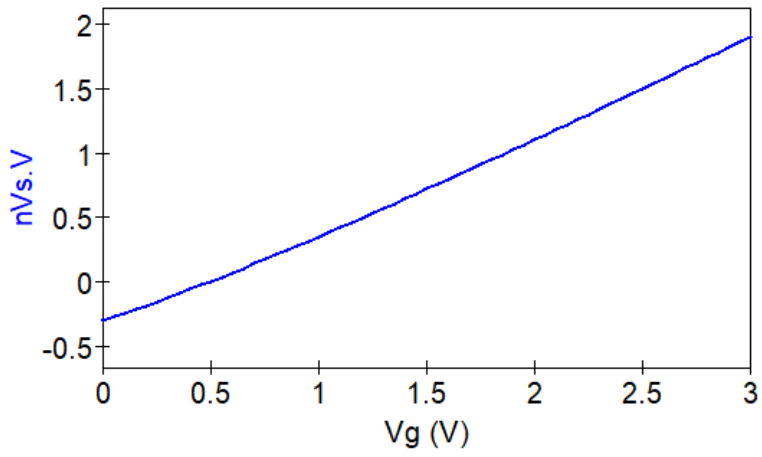


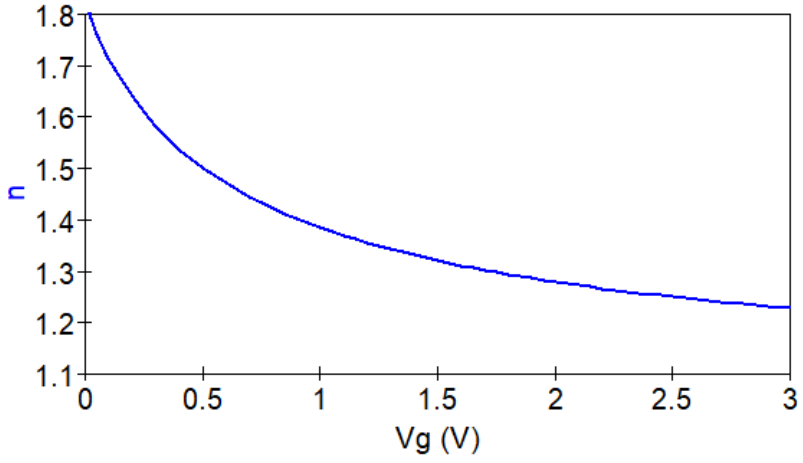
dc simulation
DC1

Parameter sweep
SW1
Sim=DC1
Param=Vg
Type=lin
Start=0
Stop=3
Points=61

equation
Eqn1
Ispecific=3.2e-8
n=1/diff(nVs.V)

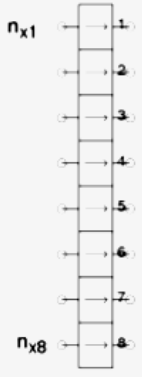
EKV2p61
W=20e-6
L=10e-6
VTO=0.5
Cgb0=1.5e-10
Cgd0=1.5e-10
Cgs0=1.5e-10
COX=3.45e-3
TNOM=26.58
TEMP=26.58
KP=20e-6
THETA=50e-3
PHI=0.5
HDIF=0.9e-6
RSH=510
GAMMA=0.7

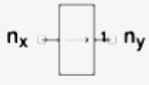
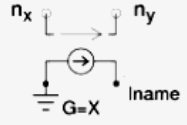
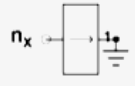




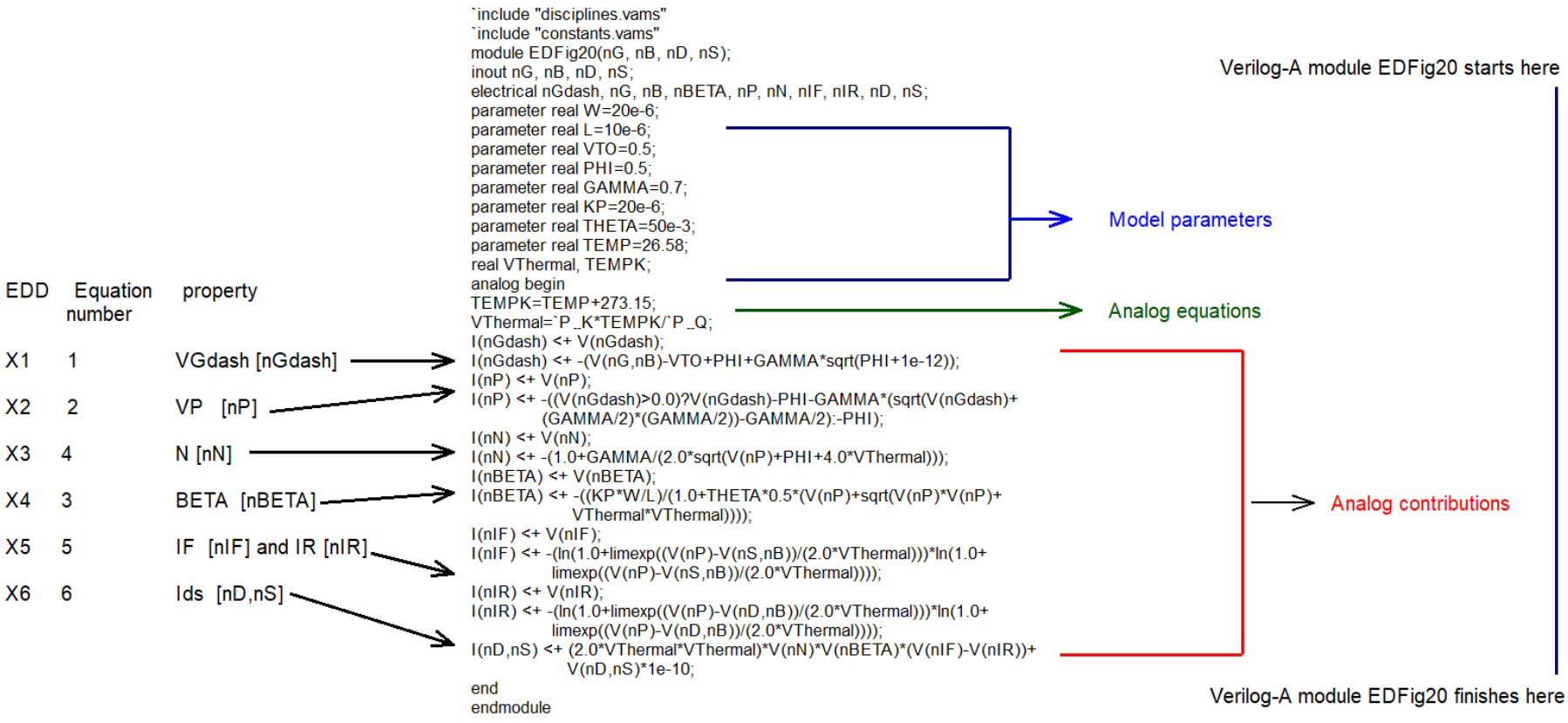
Building Compact Device Models with Qucs/Qucs-S/QucsStudio - Schematic to Verilog-A Link

Fundamental EDD blocks

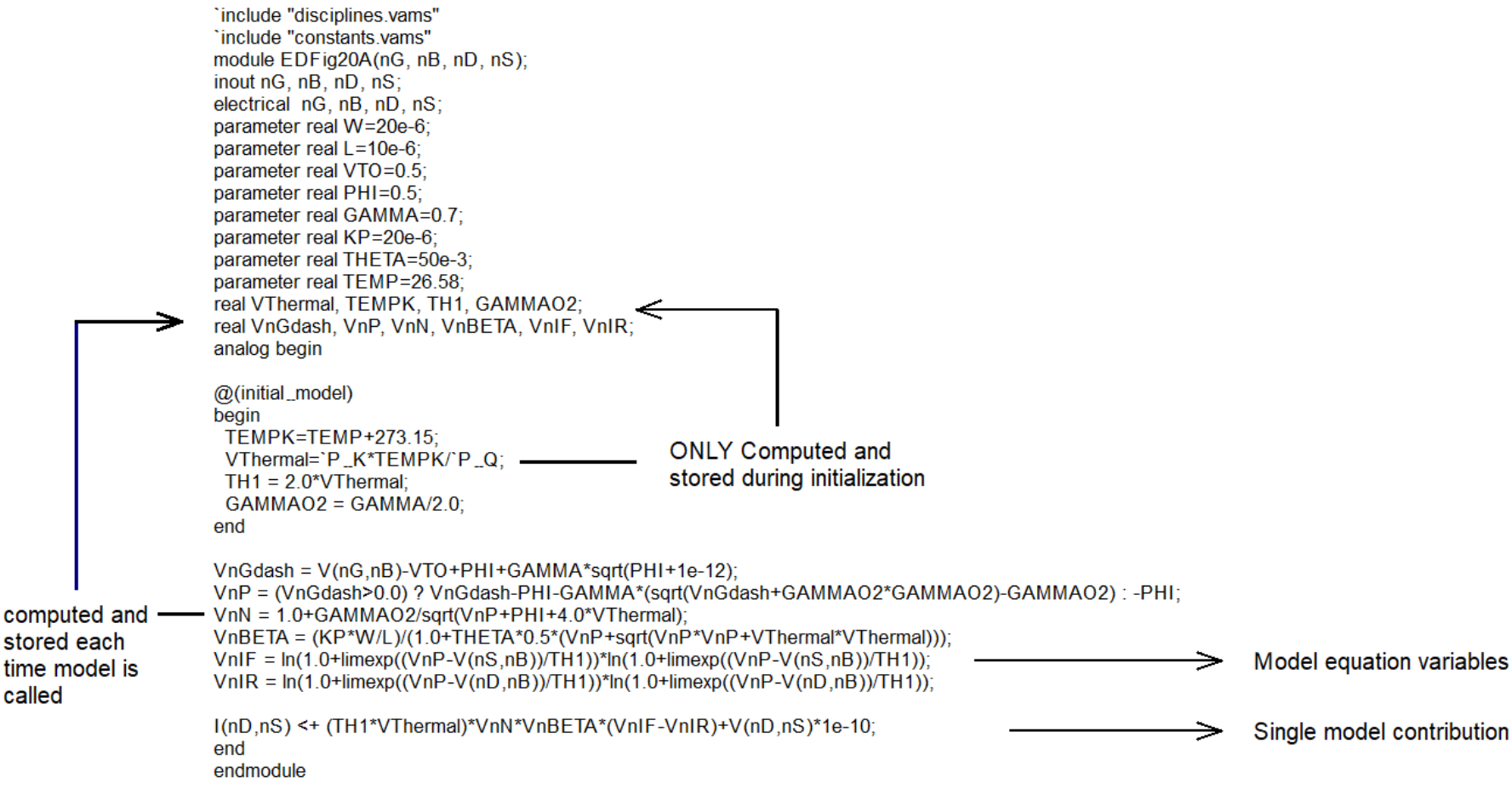
Qucs symbol	Quantity equations	Verilog-A code fragment
	Iname = I1 = f(V2, V3, ... V8) I2, I3, ... I8 = 0 and Q1, Q2, ... Q8 = 0. Where Vm = V(n_{xm}, n_{ym}) or Vm = V(n_{xm}), and 2 ≤ m ≤ 8.	Iname = f(V2, V3, ... V8); Or Iname <+ f(V2, V3, ... V8);
	Q1 = f(V1, V2, ... V8, I1, I2, ... I8) Q2, ... Q8 = 0. Where Vm = V(n_{xm}, n_{ym}) or Vm = V(n_{xm}), and 1 ≤ m ≤ 8.	I(n_{x1}, n_{y1}) <+ ddt(Q1); Or I(n_{x1}, n_{y1}) = ddt(Q1);

	Quantity equations	Verlog-A code fragment
(a) Model initialisation block	Equation Eqn1 con1= ..... con2= ..... con3=	@(initial_model) begin con1 = ...; con2 = ...; con3 = ...; end
(b) Standard resistors		I(n_x, n_y) <+ V(n_x, n_y)/R; I(n_x, n_y) <+ white_noise((FourKT/R, "thermal"); Where FourKT = 4.0 · `P_K · $temperature, and `P_K = 1.3806505e-23 K^{-1}, $temperature is the resistor temperature in Kelvin.
(c) Noise free resistors	 I1 = V1/R = V(n_x, n_y)/R	I(n_x, n_y) <+ V(n_x, n_y)/R;
(d) Voltage controlled current block	 Iname = X · V(n_x, n_y)	Iname = X*V(n_x, n_y);
(e) current to voltage conversion block	 Iname = I1 = V1	Iname = V(n_x);

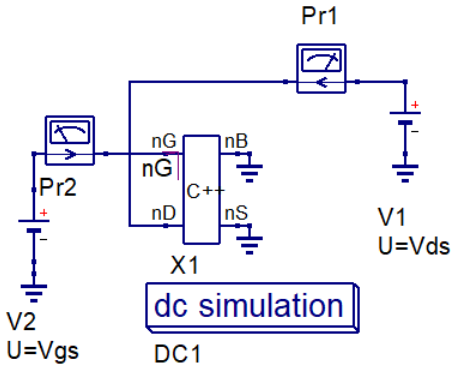
Building Compact Device Models with Qucs/Qucs-S/QucsStudio - Verilog-A EDD Style Template



Building Compact Device Models with Qucs/Qucs-S/QucsStudio - Verilog-A Equation Style Template



Building Compact Device Models with Qucs/Qucs-S/QucsStudio - Relative Model Timing

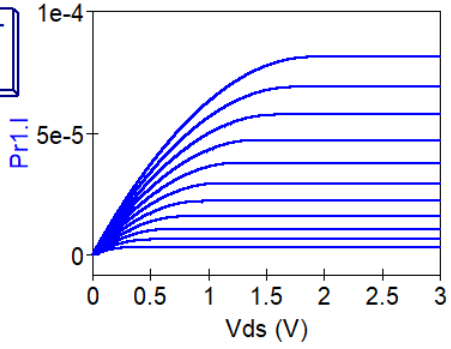


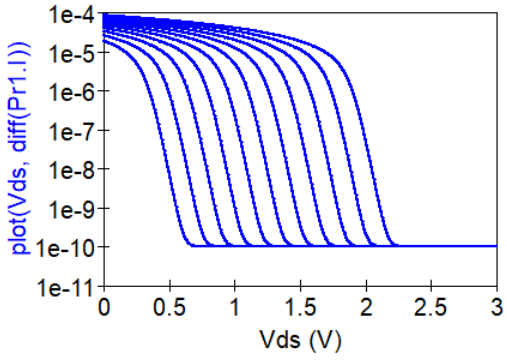
Parameter sweep

SW2
Sim=SW1
Param=Vgs
Type=lin
Start=1.0
Stop=3.0
Points=11

Parameter sweep

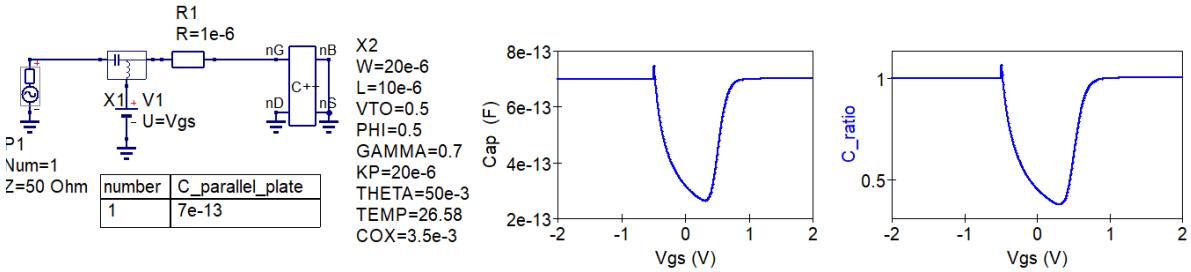
SW1
Sim=DC1
Param=Vds
Type=lin
Start=0
Stop=3
Points=10001





Relative timing information [indicative data]	
1 Verilog-A equation template	1
2 Verilog-A EDD template	1.3
3. EDD	194

Building Compact Device Models with Qucs/Qucs-S/QucsStudio - Verilog-A Dynamic Charge Current Contributions to Equation Style Template




```
`include "disciplines.vams"
`include "constants.vams"
module EDFig20B(nG, nB, nD, nS);
inout nG, nB, nD, nS;
electrical  nG, nB, nD, nS;
parameter real W=20e-6;  parameter real L=10e-6;  parameter real VTO=0.5;  parameter real PHI=0.5; parameter real GAMMA=0.7;
parameter real KP=20e-6;  parameter real THETA=50e-3;  parameter real TEMP=26.58;  parameter real COX=0.7e-3;
real VThermal, TEMPK, TH1, GAMMAO2,  VnGdash, VnP, VnN, VnBETA, VnIF, VnIR, VnQ, VXF, VXR, VnQD, VnQS, VnQI, VnQB;
analog begin
  @(initial_model)
    begin   TEMPK=TEMP+273.15;  VThermal=`P_K*TEMPK/`P_Q;  TH1 = 2.0*VThermal;  GAMMAO2 = GAMMA/2.0;  end
  VnGdash = V(nG,nB)-VTO+PHI+GAMMA*sqrt(PHI+1e-12);
  VnP = (VnGdash>0.0) ? VnGdash-PHI-GAMMA*(sqrt(VnGdash+GAMMAO2*GAMMAO2)-GAMMAO2) : -PHI;
  VnN = 1.0+GAMMAO2/sqrt(VnP+PHI+4.0*VThermal);
  VnBETA = (KP*W/L)/(1.0+THETA*0.5*(VnP+sqrt(VnP*VnP+VThermal*VThermal)));
  VnIF = ln(1.0+limexp((VnP-V(nS,nB))/TH1))*ln(1.0+limexp((VnP-V(nS,nB))/TH1));
  VnIR = ln(1.0+limexp((VnP-V(nD,nB))/TH1))*ln(1.0+limexp((VnP-V(nD,nB))/TH1));
  I(nD,nS) <+ (TH1*VThermal)*VnN*VnBETA*(VnIF-VnIR)+V(nD,nS)*1e-10;
  VnQ = 1.0 + GAMMA/(2.0*sqrt(VnP+PHI+1e-6));
  VXF = sqrt(0.25+VnIF);      VXR = sqrt(0.25+VnIR);
  VnQD = (-VnQ*(  (4.0/15.0)*(3.0*VXR*VXR*VXR + 6.0*VXR*VXR*VXF + 4.0*VXR*VXF*VXF + 2.0*VXF*VXF*VXF +1e-12)/((VXF+VXR)*(VXR+VXF)+1e-12) ) -0.5);
  VnQS = (-VnQ*(  (4.0/15.0)*(3.0*VXF*VXF*VXF + 6.0*VXF*VXF*VXR + 4.0*VXF*VXR*VXR + 2.0*VXR*VXR*VXR +1e-12)/((VXF+VXR)*(VXR+VXF)+1e-12) ) -0.5);
  VnQI =  VnQD + VnQS;
  VnQB =  (VnGdash > 0.0) ? (-GAMMA*sqrt(VnP+PHI+1e-6)/VThermal) - ( (VnQ - 1.0)/VnQ )* VnQI :  -VnGdash/VThermal;
  I(nS,nB) <+ ddt(COX*VThermal*W*L*VnQS);    I(nD,nB) <+ ddt(COX*VThermal*W*L*VnQD);    I(nG,nB) <+ ddt(COX*VThermal*W*L*(-VnQI-VnQB));
end
endmodule
```

Static current I(nD,nS)

Dynamic charge currents

Qucs/Qucs-S/QucsStudio Basic Verilog-A subset - Part 1

- Verilog-A is a case sensitive language.
- Comments: (1) single line comments start with **//**,
  (2) block comments begin with **/*** and end with ***/**
- Identifiers are sequences of letters, digits, dollar signs '**$**' and the underscore '_'; the first letter of an identifier must not be a digit.
- Qucs/ADMS version 2.34 keywords: **parameter, aliasparameter, aliasparam, module, endmodule, function, endfunction, discipline, potential, flow, domain, ground, enddiscipline, nature, endnature, input, output, inout, branch, analog, begin, end, if, while, case, endcase, default, for, else, integer,real, string, from, exclude, inf, INF.**
- Compiler directives: **`define,`undef, `ifdef, `else, `endif, `include.**
- Data types: **integer**s, **real**s and **string**s.
- Predefined constants in “constants.vams”: **`M_PI, `M_TWO_PI, `M_PI_2, `M_PI_4, `M_1_PI, `M_2_PI, `M_2_SQRTPI, `M_E, `M_LOG2E, `M_LOG10E, `M_LN2, `M_LN10, `M_SQRT2, `M_SQRT1_2, `P_Q, `P_C, `P_K, `P_H,`P_EPS0, `P_U0, `P_CELSIUS0.**
- Variables are named objects that contain a value of a particular type. They are initialised to zero or uknown. They retain their value until changed by an assignment statement.

Qucs/Qucs-S/QucsStudio Basic Verilog-A subset - Part 2

- Parameters are declared with statements of the form:
  parameter integer size=16; parameter real period = 1.0 from (0:inf);
  parameter integer dir = 1 from [-1:1] exclude 0;
- Verilog-A natures and disciplines ae listed in file “disciplines.vams”
- Port, net and node examples in Verilog-A:

```
module amp(out1, in1);
  input in1;
  output out1;
  electrical out1, in1;
```

- Branches are declared with statement **branch (n1,n2) b1;**
- Signal access function examples: **V(n2), I(n), V(b1), I(b1), V(n,m), I(n,m)**
- Current contribution examples:

```
I(diode) <+ Is*(limexp(V(diode)/$vt)-1);
I(diode) <+ ddt(-2*cj0*phi*sqrt(1-V(diode)/phi));
```

- Qucs/ADMS allows an extensive range of Verilog-AMS operators and mathematical functions.
- Environmental Functions: **$temperature, $vt, $strobe, $finish, $given, $parameter_given.**
- Analogue operators: **@(initial_step), @(final_step), @(initial_model), @(initial_instance).**

Qucs/Qucs-S/QucsStudio Basic Verilog-A subset - Part 3

Analogue behavioural statements:

1. Analog process/procedural block;

```
analog begin
   I(diode) <+ Is*(limexp(V(diode)/$vt)-1);
   qd       =  tf*I(diode) -2*cj0*phi*sqrt(1-V(diode)/phi);
   I(diode) <+ ddt(qd);
end
```

2. Assignment statements consist of a variable followed by = and an expression
3. Conditional operator cond ? Val1 : Val2, for example

```
State = (V(d) > 0.0) ? 1 : -1;
```

TRUE = non-zero value

4. if-else statement:

```
If (V(ps,ns) > thresh)
    I(p,n) <+ 1;
else
    I(p,n) <+ 0;
```

5. Case statement:

```
case (select)
  0 : out = I(in0);
  1 : out = I(in1);
  2 : out = I(in2);
  default : out = 0;
endcase
```

6. While statement:

```
test = 4;
While( test) begin
  A = A+1;
  B = B-1;
  test = test-1;
end
```

7. For loops:

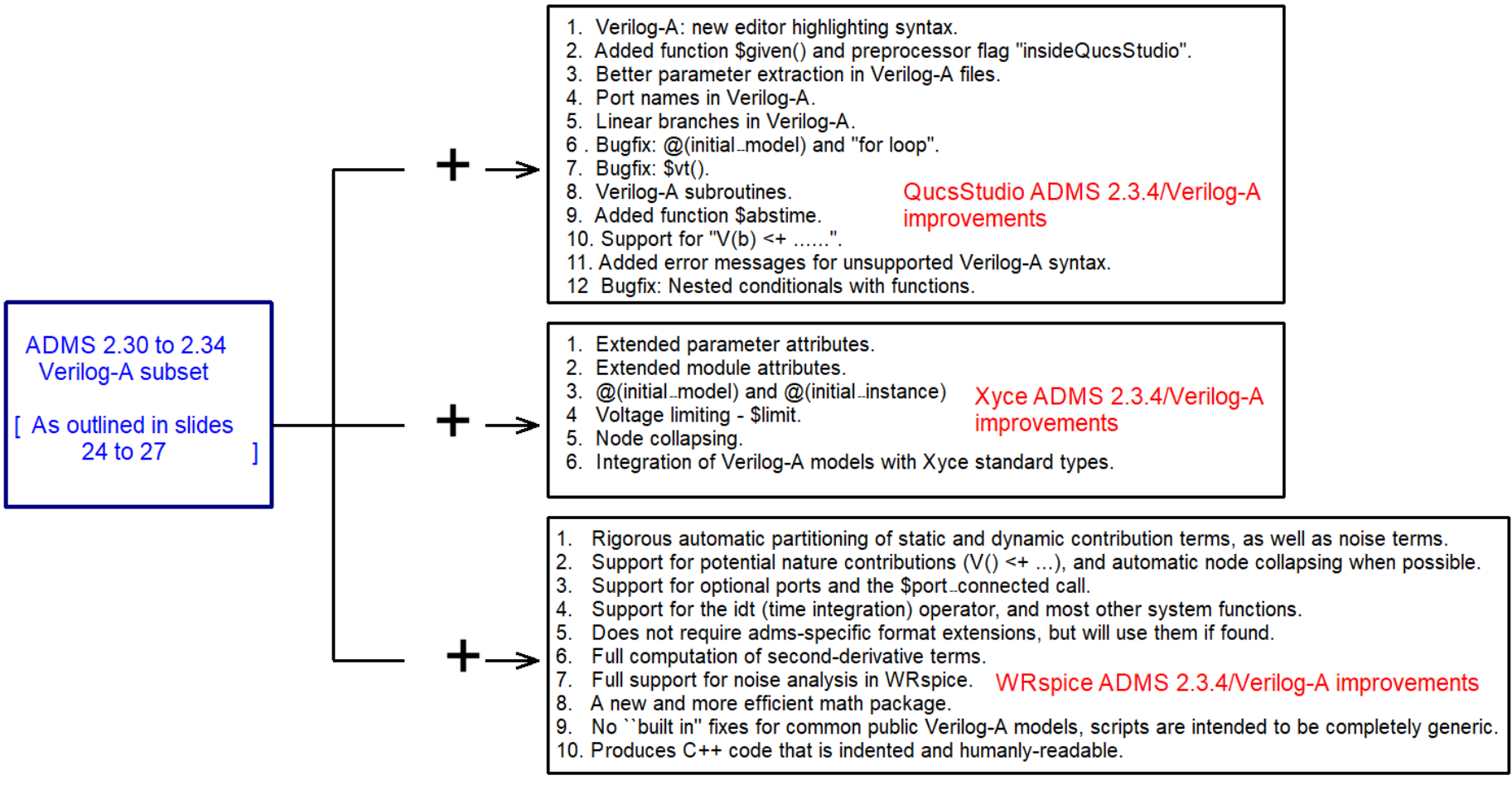
```
for (i=0; I <10; i=i+1) begin
                ..
end
```

8. User defined functions and function calls:

Qucs/Qucs-S/QucsStudio Basic Verilog-A subset - Part 4

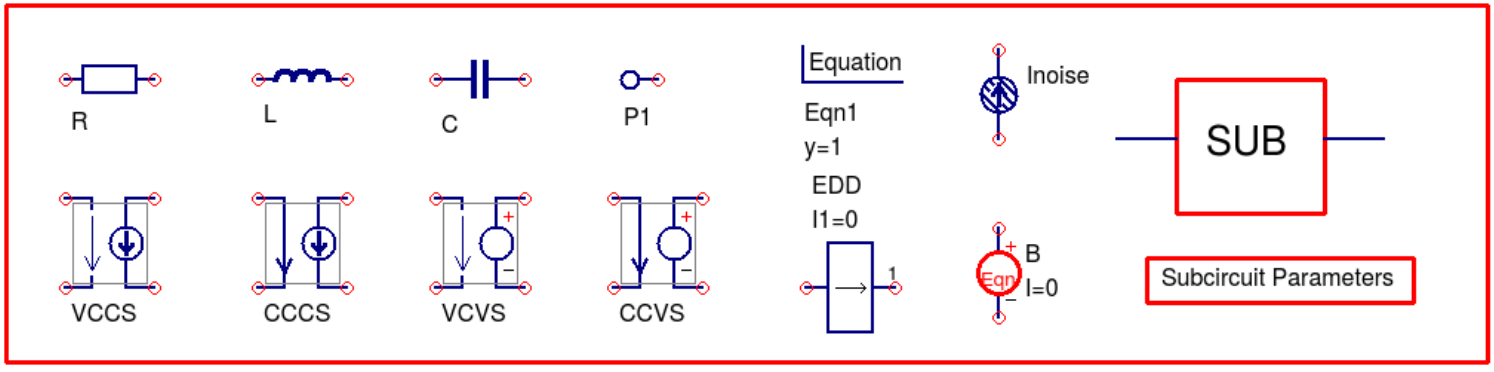
- Although Verilog-A is standardised, published versions of ADMS implement different subsets of verilog-A.
- Simulator interface facilities may vary: for example **$strobe** is replaced/added to by **$error** and **$warning** in QucsStudio.
- The inclusion mechanism for Verilog-A predefined constants, natures and discipines files can change from simulator to simulator: for example use of **`include constants.h** and **`include disciplines.h** by SymicaDE.
- Parameter statement descriptive elements vary significantly amoung circuit simulator implementations: for example Qucs and QucsStudio use **`attr(info = "description")**, or **`attr(desc="description")** and **unit="value"** other simulators **use`P(.....)**.
- Verilog-A compact modelling extensions: special features taylored to compact device modelling have been added to the MOT-ADMS 2.30 software, including **desc, unit, $parameter_given, $given**, **aliasparameter** and **string** type in **parameters**. In addition the function **ddx** is now adopted in the language standard, where **ddx** takes a variable and a node potential (for example **gm = ddx(Ids,V(g));**) and returns the symbolic partial derivative of the variable with respect to the node potential, holding all other potentials constant. Qucs and QucsStudio implement these extensions but other circuit simulators may not.
- Each circuit simulator uses one or more backend XML script files for C++ code generation and model interfacing to the main body of C++ code: as these XML scripts are specific to each simulator API there are likely to be differences in the C++ model generated. One example being electrical noise: Verilog-A statements **white_noise(pwr, <name>), flicker_noise(pwr, <exp>,<name>)** are implemented in Qucs and QucsStudio but not, for example, in ngspice.
- In addition to the above are the results from new types of circuit simulation, like for example
- Harmonic Balance, correct or not?

Qucs/Qucs-S/QucsStudio Basic Verilog-A subset - Part 5



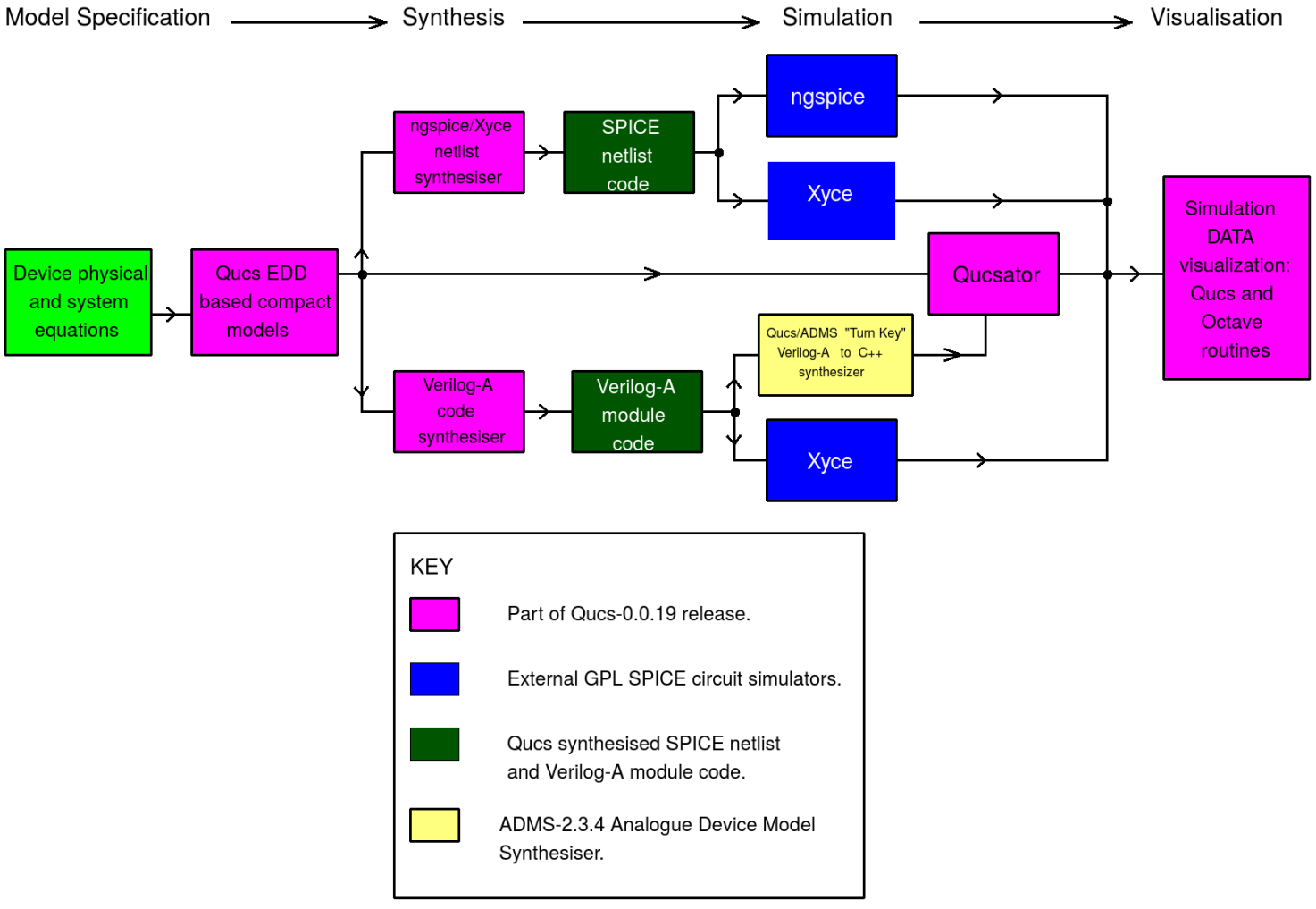
Qucs-0.0.21-S includes the Qucs GPL Verilog-A synthesis tool for compact device modelling.

- The Qucs-0.0.21-S Verilog-A synthesizer is a fully working version of this new open source ECAD tool.
- Verilog-A device models and circuit macromodels can be synthesized from the following Qucs/SPICE built in components:

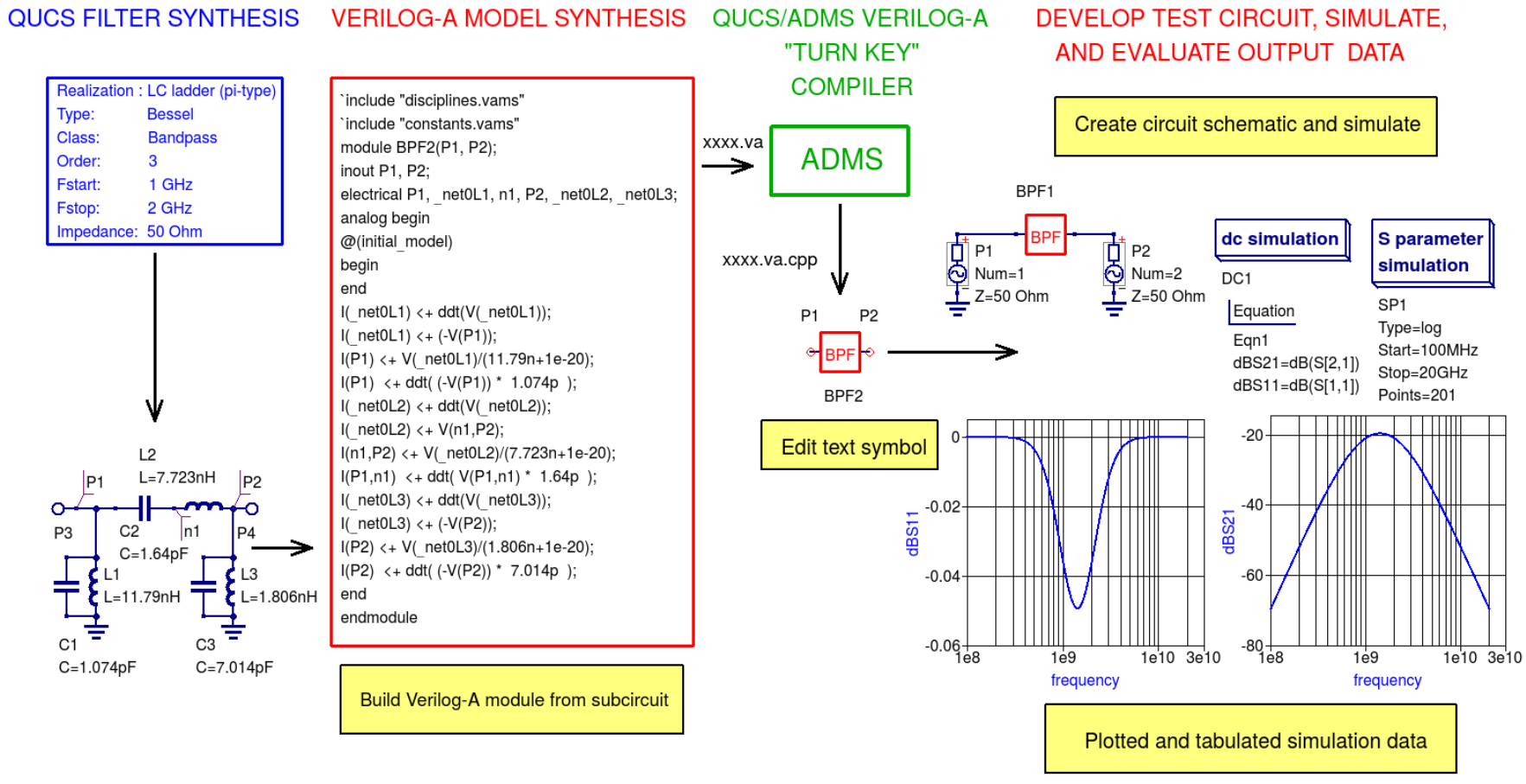


Introduction to the Qucs-S GPL Verilog-A Module Synthesizer: Part 2

Structure:

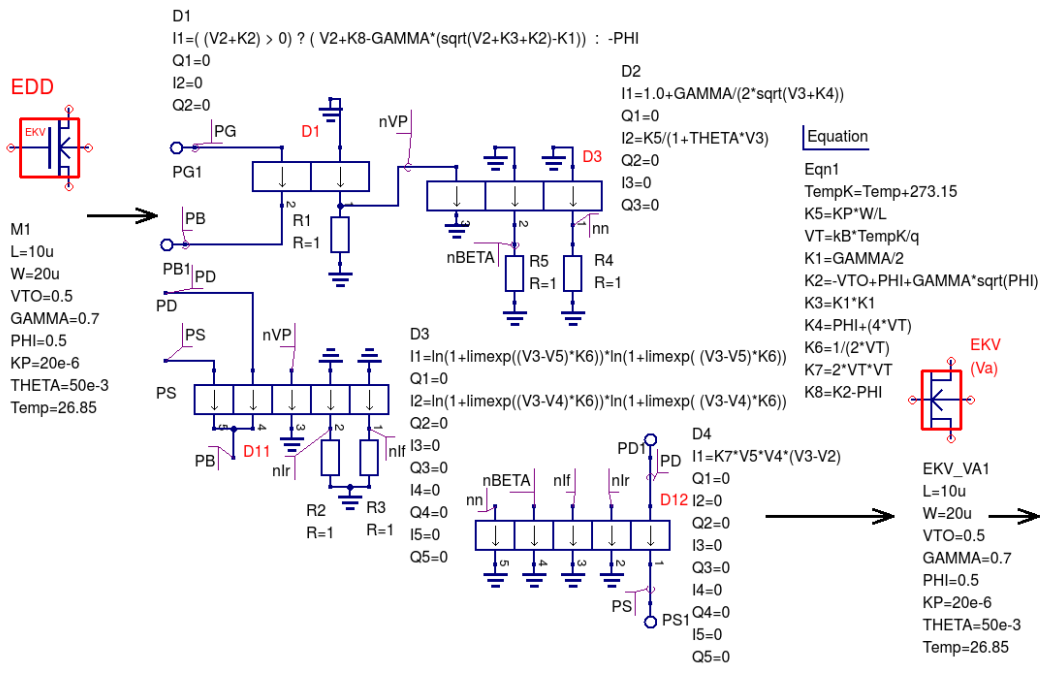


Data flow through the Qucs GPL compact device modelling tool set.



Verilog-A synthesis of multi-EDD models: EKV2p6 nMOS

$I_{ds} = f(V_d, V_g, V_s, V_b)$ model for a transistor operating in long channel mode.

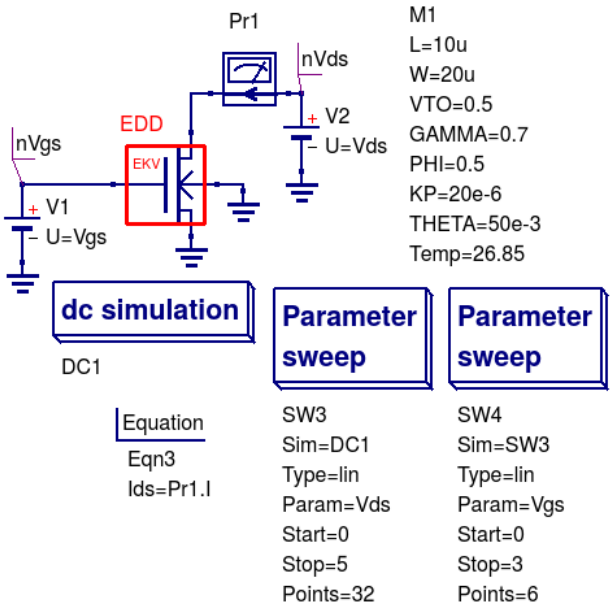


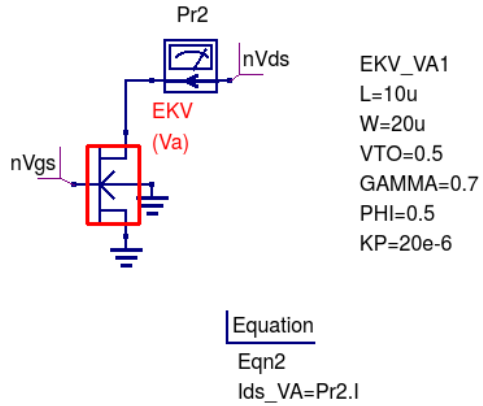
Qucs EDD EKV2p6 Ids=f(Vd, Vg, Vs, Vb) model

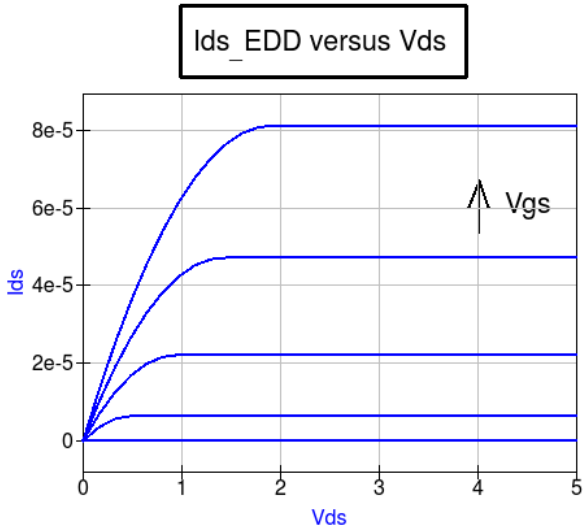
```
`include "disciplines.vams"
`include "constants.vams"
module EKV_VA(PB, PG, PD, PS);
inout PB, PG, PD, PS;
electrical PD, PS, nIr, nIf, nBETA, nn, nVP, PG, PB;
parameter real L=10u;    parameter real W=20u;    parameter real VTO=0.5;
parameter real GAMMA=0.7;    parameter real PHI=0.5;    parameter real KP=20e-6;
parameter real THETA=50e-3;    parameter real Temp=26.85;
real TempK, K5, VT, K1, K2, K3, K4, K6, K7, K8;
analog begin
@(initial_model)
begin
TempK=Temp+273.15;    K5=KP*W/L;    VT=`P_K*TempK/`P_Q;
K1=GAMMA/2;    K2=-VTO+PHI+GAMMA*sqrt(PHI);
K3=K1*K1;    K4=PHI+(4*VT);
K6=1/(2*VT);    K7=2*VT*VT;    K8=K2-PHI;
end
I(PD,PS) <+ K7*V(nn)*V(nBETA)*(V(nIf)-V(nIr));
I(nVP) <+ -(((V(PG,PB)+K2)>0)?(V(PG,PB)+K8-GAMMA*(sqrt(V(PG,PB)+K3+K2)-K1)):-PHI);
I(nVP) <+ V(nVP)/( 1 );
I(nBETA) <+ -((-V(nBETA))/( 1 ));
I(nn) <+ -(1.0+GAMMA/(2*sqrt(V(nVP)+K4)));
I(nBETA) <+ -(K5/(1+THETA*(-V(nVP))));
I(nn) <+ -((-V(nn))/( 1 ));
I(nIr) <+ -((-V(nIr))/( 1 ));
I(nIf) <+ -((-V(nIf))/( 1 ));
I(nIf) <+ -(ln(1+limexp((V(nVP)-V(PS,PB))*K6))*ln(1+limexp((V(nVP)-V(PS,PB))*K6)));
I(nIr) <+ -(ln(1+limexp((V(nVP)-V(PD,PB))*K6))*ln(1+limexp((V(nVP)-V(PD,PB))*K6)));
end
endmodule
```

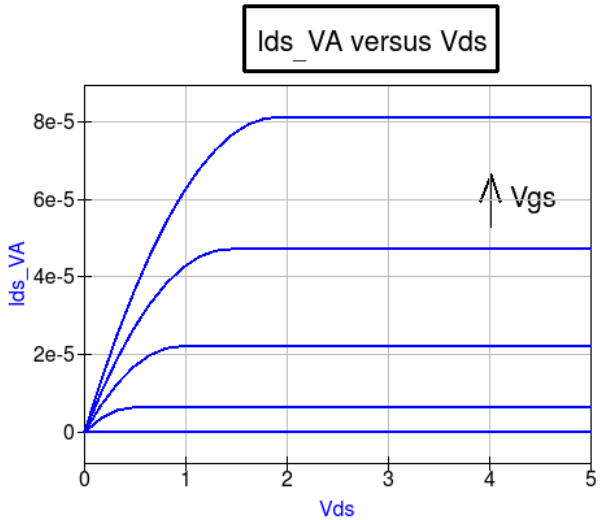
Synthesized EKV2p6 Ids=f(Vd, Vg, Vs, Vb) Verilog-A code

Verilog-A synthesis of multi-EDD models: EKV2p6 nMOS
$I_{ds} = f(V_d, V_g, V_s, V_b)$ swept DC simulation data.

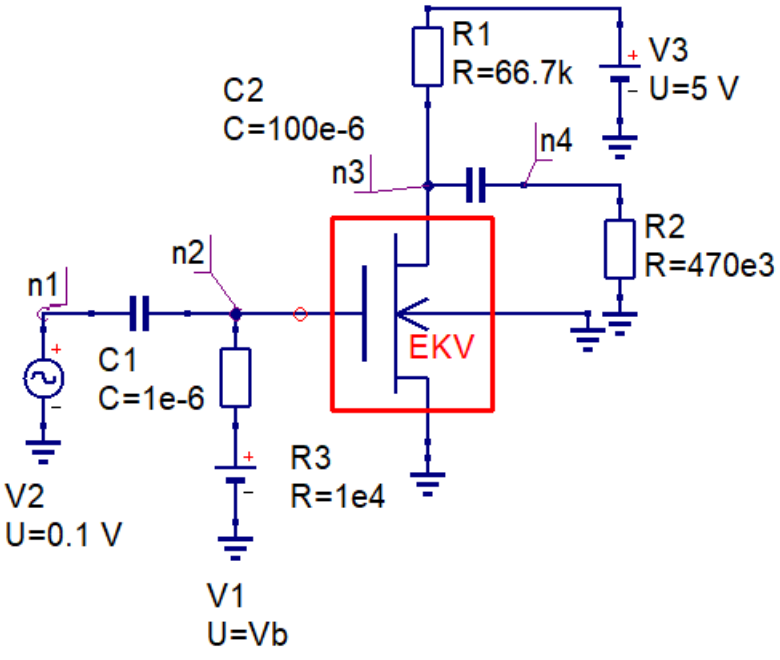








EKV 2.6 Simulation Examples - Part 1: DC



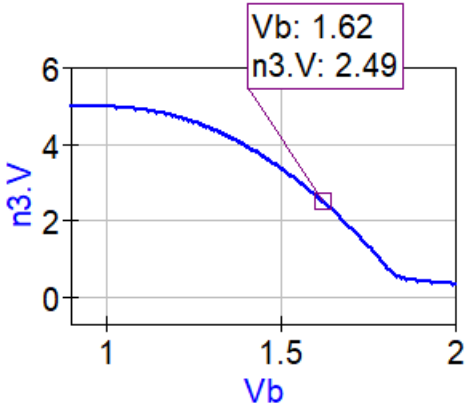
EKV2p61
W=80e-6
L=8e-6
VTO=1.0
Cgb0=1.5e-10
Cgd0=1.5e-10
Cgs0=1.5e-10
COX=3.45e-3
TNOM=26.58
TEMP=26.58
KP=30e-6
THETA=50e-3
PHI=0.5
HDIF=0.9e-6
RSH=510
GAMMA=0.7
KF=1e-27
AF=1.0

Parameter sweep

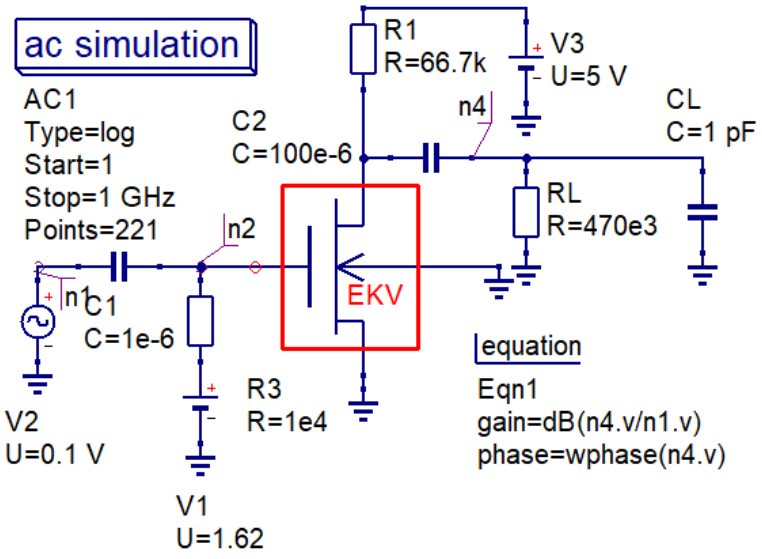
SW1
Sim=DC1
Param=Vb
Type=lin
Start=0.9
Stop=2.0
Points=221

dc simulation

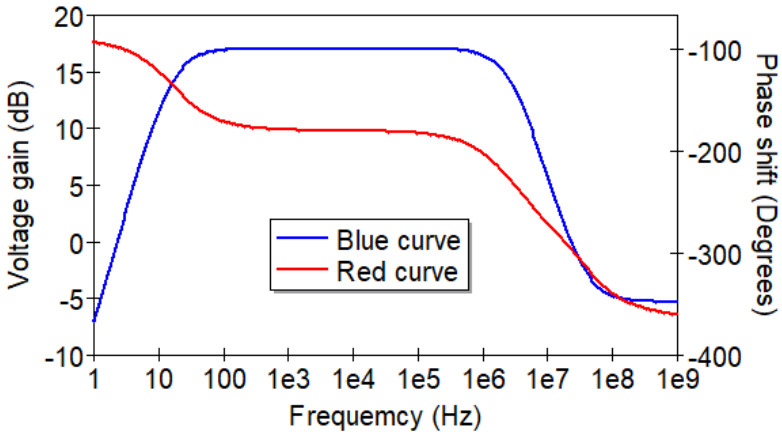
DC1



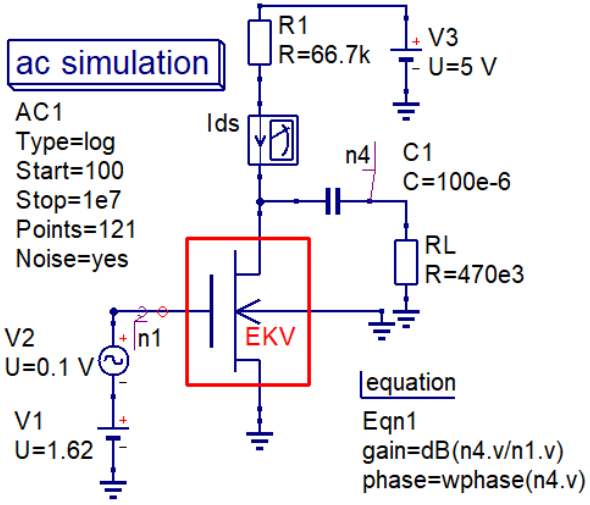
EKV 2.6 Simulation Examples - Part 2: AC



EKV2p61
W=80e-6
L=8e-6
VTO=1.0
Cgb0=1.5e-10
Cgd0=1.5e-10
Cgs0=1.5e-10
COX=3.45e-3
TNOM=26.58
TEMP=26.58
KP=30e-6
THETA=50e-3
PHI=0.5
HDIF=0.9e-6
RSH=510
GAMMA=0.7
KF=1e-27
AF=1.0



EKV 2.6 Simulation Examples - Part 3: AC noise



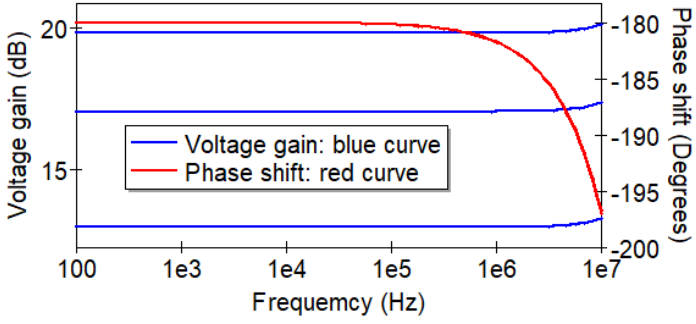
EKV2p61
W=Wscan
L=8e-6
VTO=1.0
Cgb0=1.5e-10
Cgd0=1.5e-10
Cgs0=1.5e-10
COX=3.45e-3
TNOM=26.58
TEMP=26.58
KP=30e-6
THETA=50e-3
PHI=0.5
HDIF=0.9e-6
RSH=510
GAMMA=0.7
KF=1e-27
AF=1.0

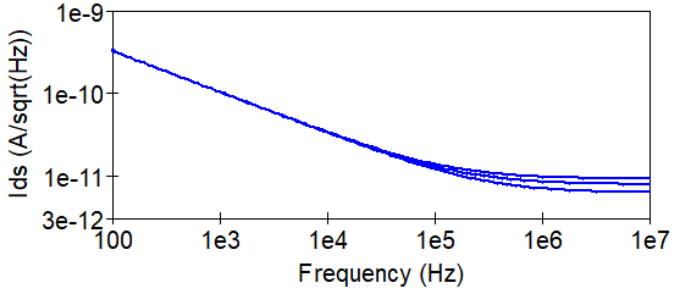
```
....
module EKV2p6R4(nG, nD, nS, nB);
.....
parameter real KF=1.0e-27;
parameter real AF=1.0;
....
real FOURKT, Ispecific, GM,Sthermal, Sflicker;
analog begin

@(initial_model)
begin
....
 FOURKT = 4.0*`P_K*TEMPK;
.....
end
.....
Ispecific = (TH1*VThermal)*VnN*VnBETA;      I(nD,nS) <+ Ispecific*(VnIF-VnIR)+V(nD,nS)*1e-10;
Sthermal = FOURKT*VnBETA*abs(VnQI);
GM = VnBETA*VThermal*( sqrt( (4.0*VnIF/(Ispecific+1e-10)) + 1.0) - sqrt( (4.0*VnIR/(Ispecific+1e-10)) + 1.0) );
Sflicker = (KF*GM*GM)/(W*L*COX);
I(nS,nD) <+ white_noise(Sthermal, "thermal");      I(nS,nD) <+ flicker_noise(Sflicker, AF, "flicker");
end
endmodule
```

Verilog-A white and flicker noise code.

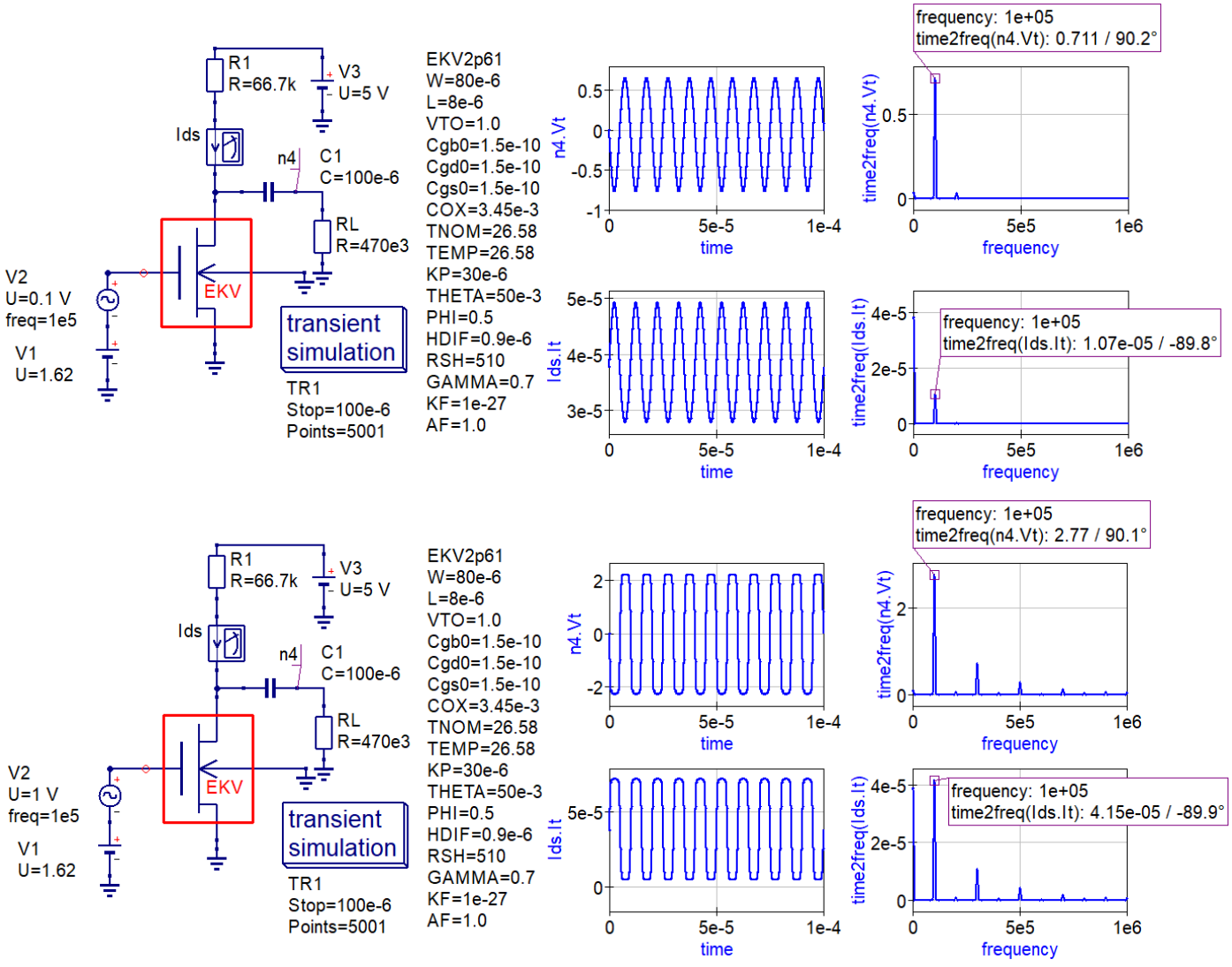
Parameter sweep

SW1
Sim=AC1
Param=Wscan
Type=list
Points=50e-6; 80e-6; 110e-6

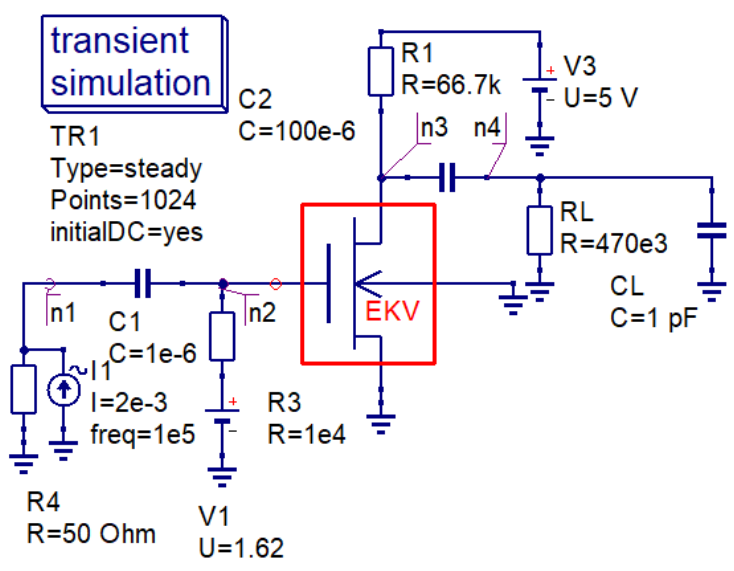




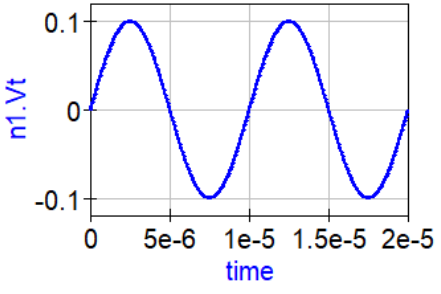
EKV 2.6 Simulation Examples - Part 4: TRAN

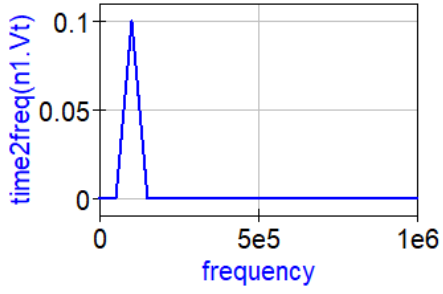


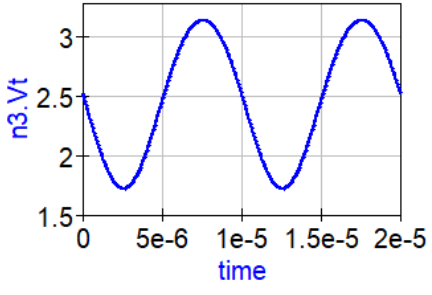
EKV 2.6 Simulation Examples - Part 5: TRAN Shooting Method Small Signal Input

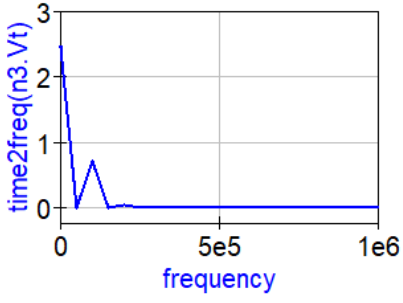


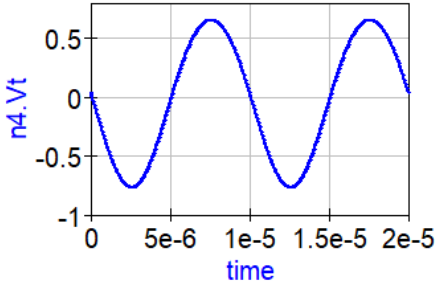

```
EKV2p61
W=80e-6
L=8e-6
VTO=1.0
Cgb0=1.5e-10
Cgd0=1.5e-10
Cgs0=1.5e-10
COX=3.45e-3
TNOM=26.58
TEMP=26.58
KP=30e-6
THETA=50e-3
PHI=0.5
HDIF=0.9e-6
RSH=510
GAMMA=0.7
KF=1e-27
AF=1.0
```

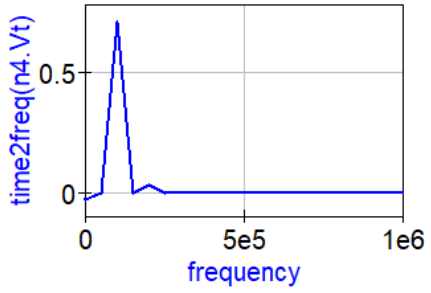




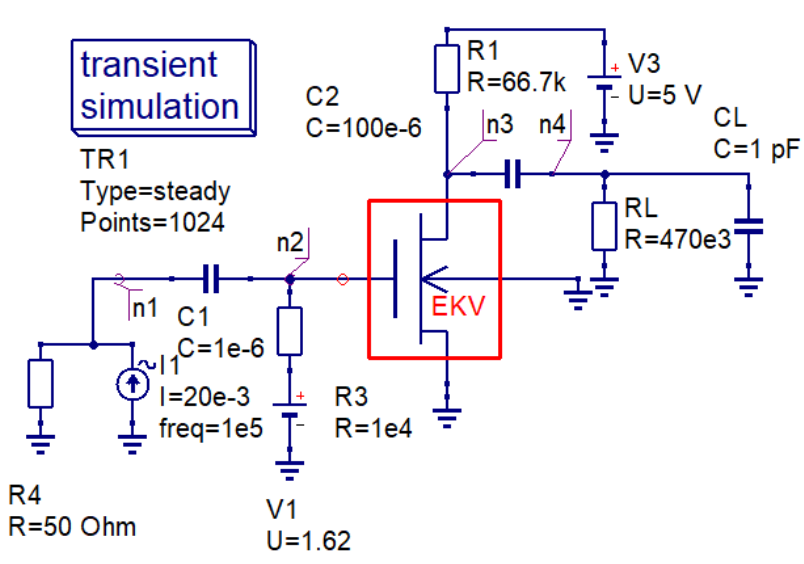




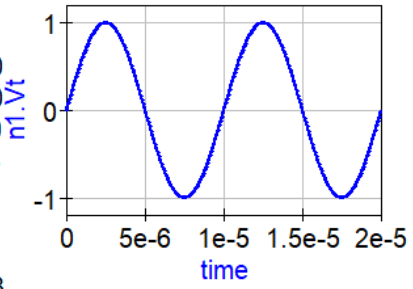


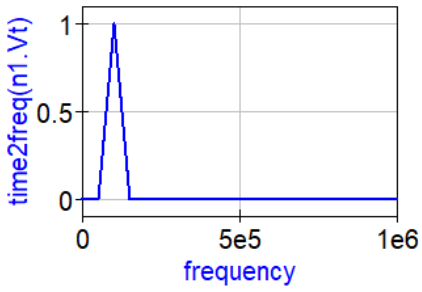


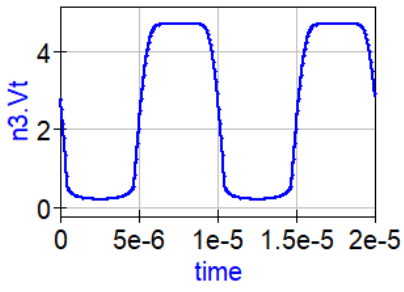
EKV 2.6 Simulation Examples - Part 6: TRAN Shooting Method Large Signal Input

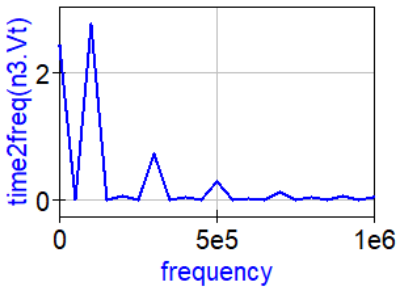


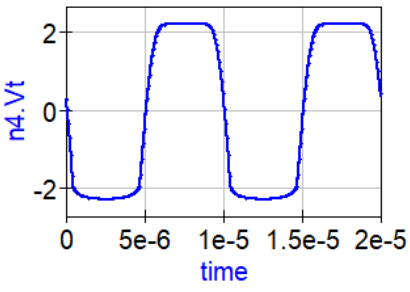

```
EKV2p61
W=80e-6
L=8e-6
VTO=1.0
Cgb0=1.5e-10
Cgd0=1.5e-10
Cgs0=1.5e-10
COX=3.45e-3
TNOM=26.58
TEMP=26.58
KP=30e-6
THETA=50e-3
PHI=0.5
HDIF=0.9e-6
RSH=510
GAMMA=0.7
KF=1e-27
AF=1.0
```

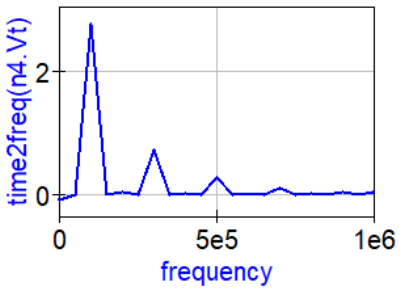




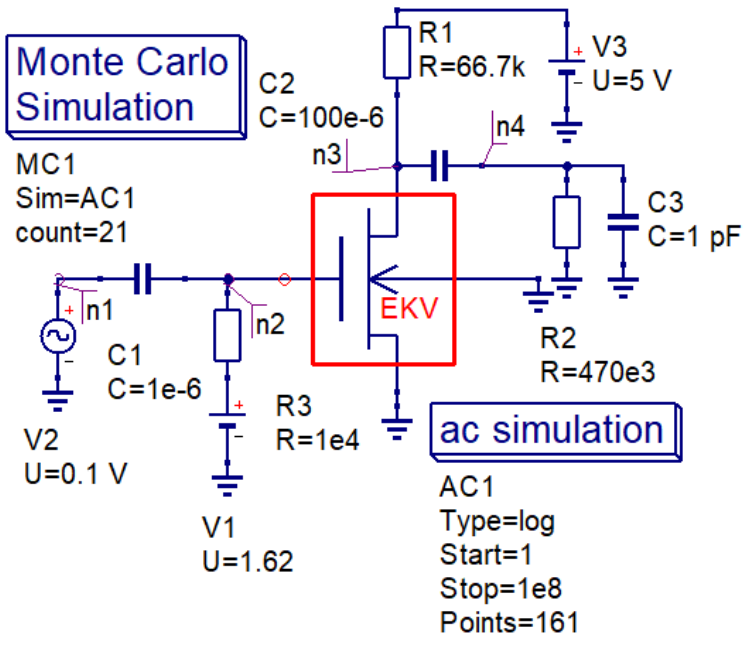




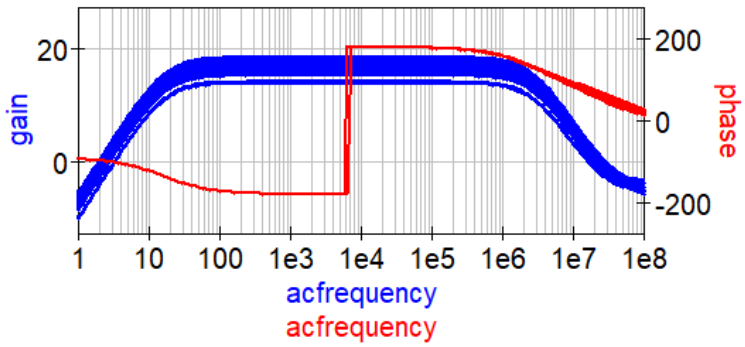




EKV 2.6 Simulation Examples - Part 7: Monte Carlo Analysis

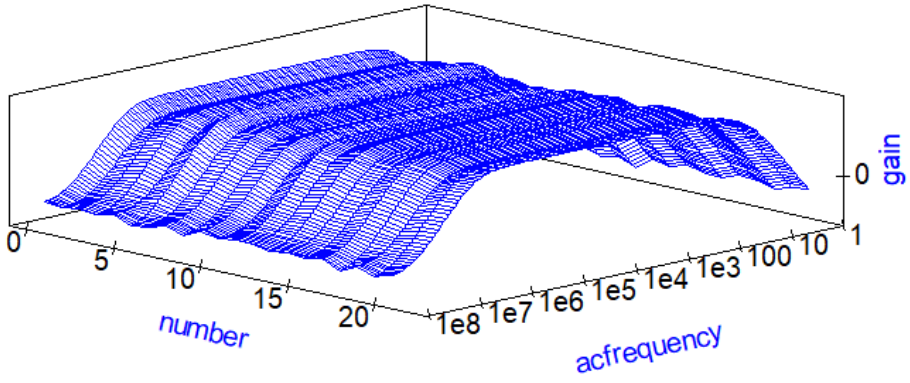


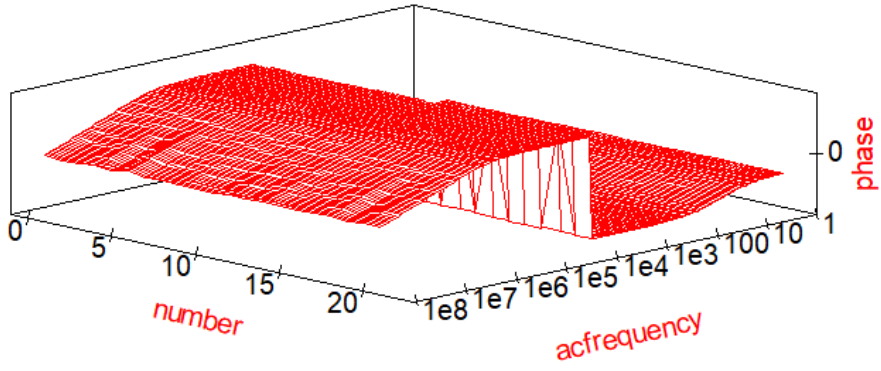
EKV2p61
W=WMC
L=LMC
VTO=1.0
Cgb0=1.5e-10
Cgd0=1.5e-10
Cgs0=1.5e-10
COX=3.45e-3
TNOM=26.58
TEMP=26.58
KP=30e-6
THETA=50e-3
PHI=0.5
HDIF=0.9e-6
RSH=510
GAMMA=0.7
KF=1e-27
AF=1.0



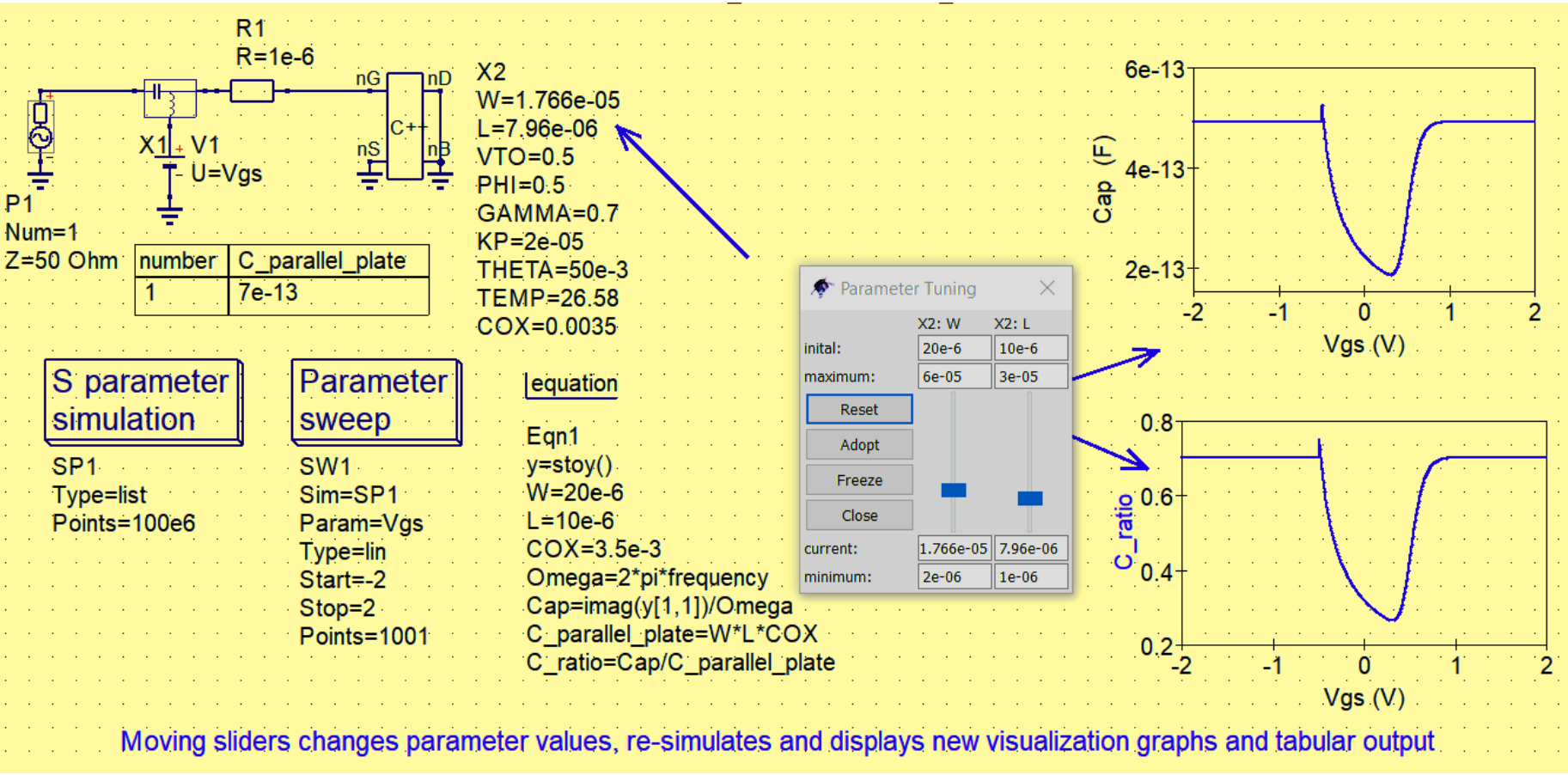
equation
Eqn1
WMC=tol(80e-6, 10)
LMC=tol(8e-6, 10)

equation
Eqn2
gain=dB(n4.v/n1.v)
phase=phase(n4.v)



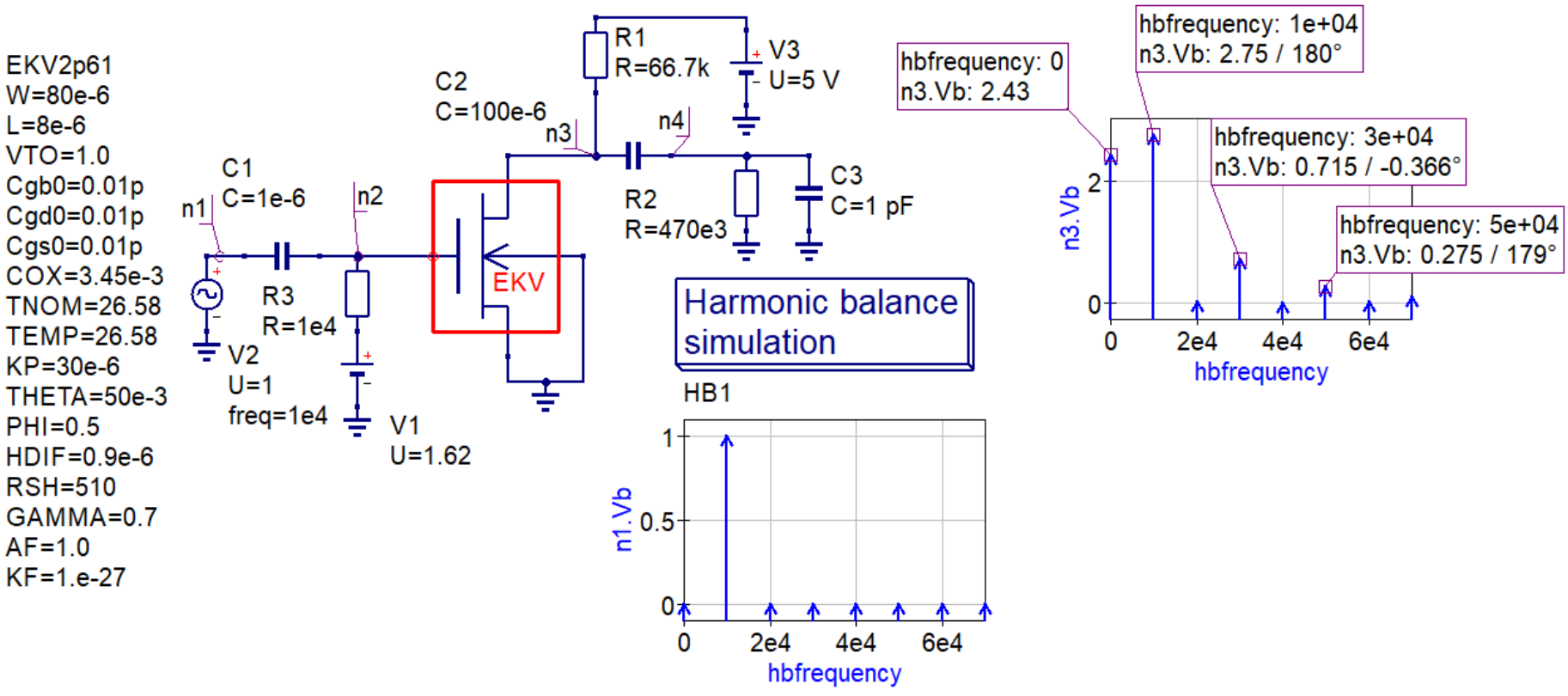


EKV 2.6 Simulation Examples - Part 8: Tuning

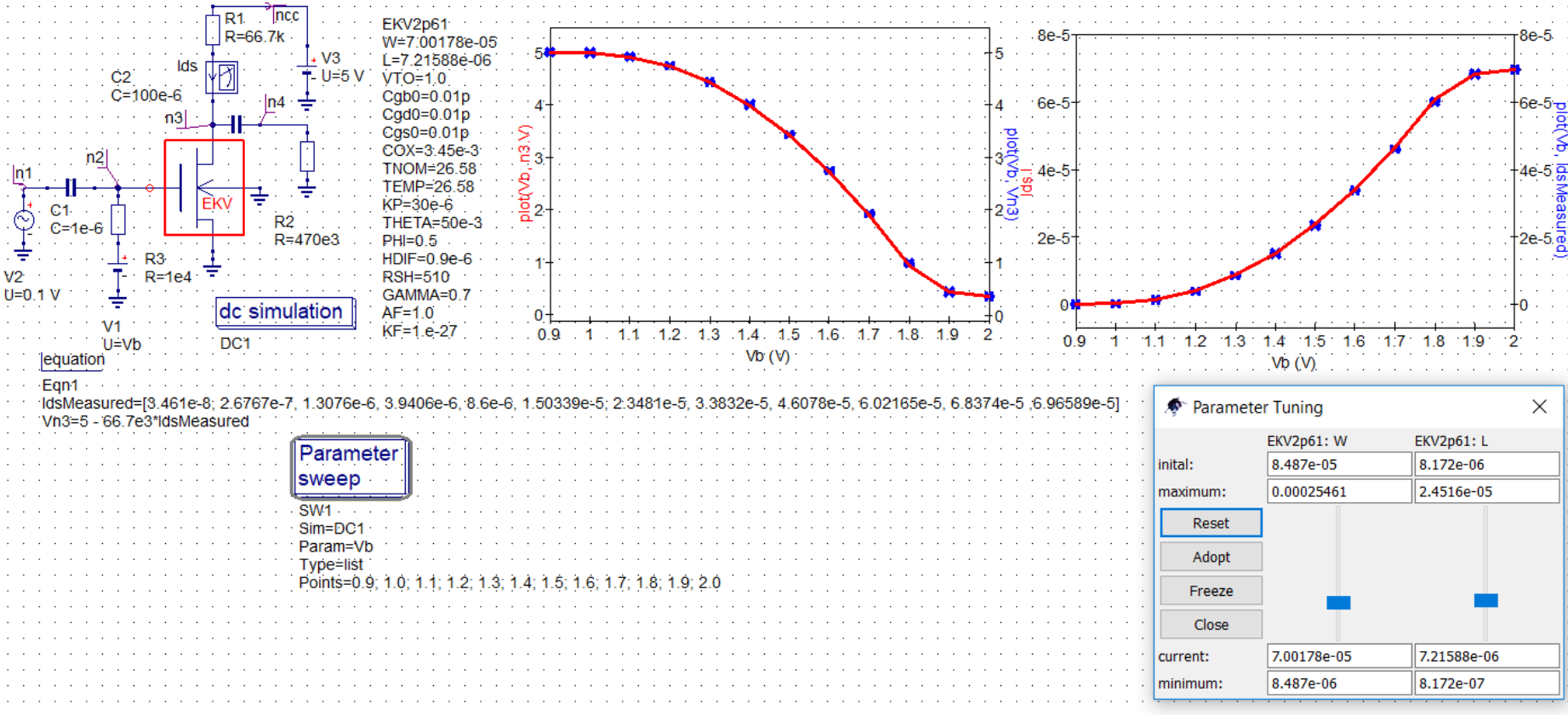


Moving sliders changes parameter values, re-simulates and displays new visualization graphs and tabular output

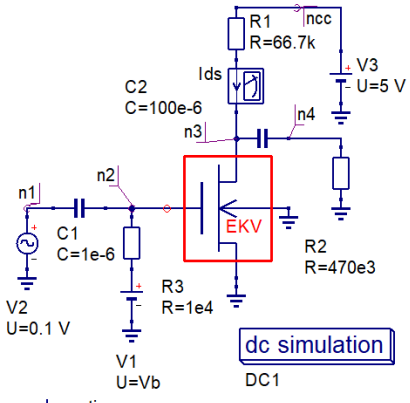
EKV 2.6 Simulation Examples - Part 9: EKV 2.6 Intrinsic Verilog-A Model Harmonic Balance (HB) Simulation



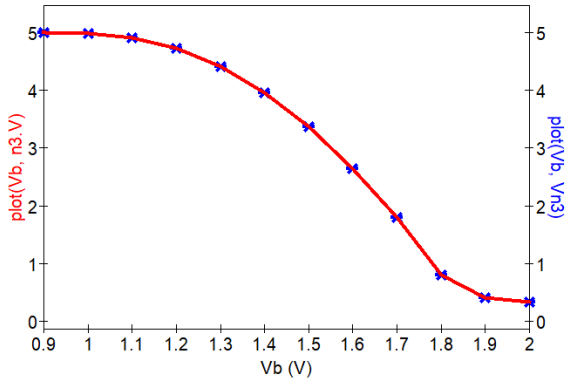
EKV 2.6 Parameter extraction - Part 1: Using Manual Tuning with Sliders

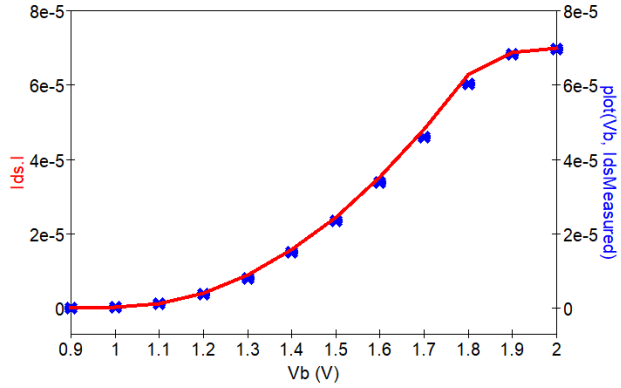


EKV 2.6 Parameter extraction - Part 2: Using Optimization



EKV2p61
W=W
L=7e-6
VTO=1.0
Cgb0=0.01p
Cgd0=0.01p
Cgs0=0.01p
COX=3.45e-3
TNOM=26.58
TEMP=26.58
KP=30e-6
THETA=50e-3
PHI=0.5
HDIF=0.9e-6
RSH=510
GAMMA=0.7
AF=1.0
KF=1.e-27





equation
Eqn1
IdsMeasured=[3.461e-8, 2.6767e-7, 1.3076e-6, 3.9406e-6, 8.0e-6, 1.50339e-5, 2.3481e-5, 3.3832e-5, 4.6078e-5, 6.02165e-5, 6.8374e-5 ,6.96589e-5]
Vn3=5 - 66.7e3*Ids.I

number	W.opt	fit
1	7e-05	0.0906

Parameter sweep
SW1
Sim=DC1
Param=Vb
Type=list
Points=0.9; 1.0; 1.1; 1.2; 1.3; 1.4; 1.5; 1.6; 1.7; 1.8; 1.9; 2.0

Optimization
Opt1
Sim=SW1
grid|10000|1e-5|0.1|2
W=70e-6...90e-6...90e-6 linear
fit=1 MIN

equation
EqnOpt
fit=max(log10((Ids.I/IdsMeasured)^2))

The Way Forward - Possible improvements to FOSS Compact Modelling and Simulation Tools

- This presentation has attempted to introduce the fundamental principles of compact device modelling using the Verilog-A hardware description language distributed with the Qucs series of compact modelling and simulation tools. In the limited time available only a selection of the most important topics have been included. Compact device modelling using FOSS tools is still very much in its infancy. Future developments are likely concentrate on:
  - An evaluation of the current limitations of the ADMS Verilog-A software and it's implementation in FOSS circuit simulators.
  - Development of improved and new Verilog-A compact models for existing semiconductor devices and emerging technologies.
  - The development of compact model parameter extraction procedures based on measurements (using low cost measuring instruments - Digilent Analog Discovery 2 and Analog Devices ADALM1000 and ADALM2000)
  - Add an electromagnetic simulator to the FOSS Qucs based tools.

FOSS circuit Simulation and Compact Modelling software - Web Home Page Links

QucsStudio http://www.dd6um.darc.de/QucsStudio/qucsstudio.html; Download Windows version QucsStudio-2.5.7.zip from http://www.dd6um.darc.de/QucsStudio/download.html.

Qucs-S: Qucs-S: Qucs with SPICE - https://ra3xdh.github.io/; Download - version (Linux or Windows) as required from home page.

Ngspice: http://ngspice.sourceforge.net/; Download - version (Linux or Windows) as required from home page.

Xyce: https://xyce.sandia.gov/; Download - version (Linux, Windows or Mac) as required from home page.

WRspice: http://wrcad.com/wrspice.html; Download - version (Linux or Windows) as required from home page.

References

- Jahn S., and Brinson M., Interactive compact device modelling using Qucs equation defined devices, International Journal of Numerical Modelling: Electronic Networks, Devices and Fields, September/October 2008, 21(5), pp 335-349, DOI:10.1002/jnm.676.
- Brinson M., and Jahn, Qucs: A GPL software package for circuit simulation, compact device modeling and circuit macromodeling from DC to RF and beyond, International Journal of Numerical Modelling: Electronic Networks, Devices and Fields, July/August 2009, 22(4), pp 207-319, DOI:10.1002/jnm.702.
- Mike Brinson and Michael Margraf, Verilog-A compact semiconductor device modelling and circuit macromodelling with the QucsStudio-ADMS "Turn-Key" modelling system, International journal of Microelectronics and Computer Science, Vol. 3, No. 1, pp. 32-40, Jan. 2012. ISSN 2080-8755
- Wladek Grabinski, Mike Brinson, Paolo Nenzi, Francesco Lannutti, Nikolaos Makris, Angelos Antonopoulos and Matthias Bucher, Open-source circuit simulation tools for RF compact semiconductor device modelling, International Journal of Numerical Modelling: Electronic Networks, Devices and Fields, Volume 27, Issue 5-6, September- December 2014, Pages: 761–779, DOI:10.1002/jnm.1973.
- Mike Brinson and Vadim Kuznetsov, A new approach to compact semiconductor device modelling with Qucs Verilog-A analogue module synthesis, International Journal of Numerical Modelling: Electronic Networks, Devices and Fields, Volume 29, Issue 6 November-December 2016, Pages 1070–1088, DOI: 10.1002/jnm.2166.
- Mike Brinson and Vadim Kuznetsov, Extended behavioural device modelling and circuitsimulation with Qucs-S, International Journal of Electronics, Published online on 29 July 2017. http://dx.doi.org/10.1080/00207217.2017.1357764.