# Qucs-S and QucsStudio for Compact Device Modeling

Mike Brinson
Centre for Telecommunications Research
London Metropolitan University, UK
brinsonm@staff.londonmet.ac.uk

September 14-18, 2020

# Presentation Topics

- Qucs-S and QucsStudio compact device modelling and simulation features
- QucsStudio and Qucs-S: a combined modelling and simulation package
- QucsStudio Verilog-A module development: facilities and properties
- Built in Verilog-A modules: CMC and others
- Equation-Defined Device (EDD) modelling: principles and application
- Qucs-S Verilog-A module synthesis: facilities and link to QucsStudio
- The Verilog-A Equation-Defined Device (VAEDD): structure and properties
- Enhanced QucsStudio compact device modelling and simulation
- Enhanced Qucs-S/Xyce behavioural EDD modelling
- Onwards to the next generation package

**Linux and Windows compact device modelling and simulation package**

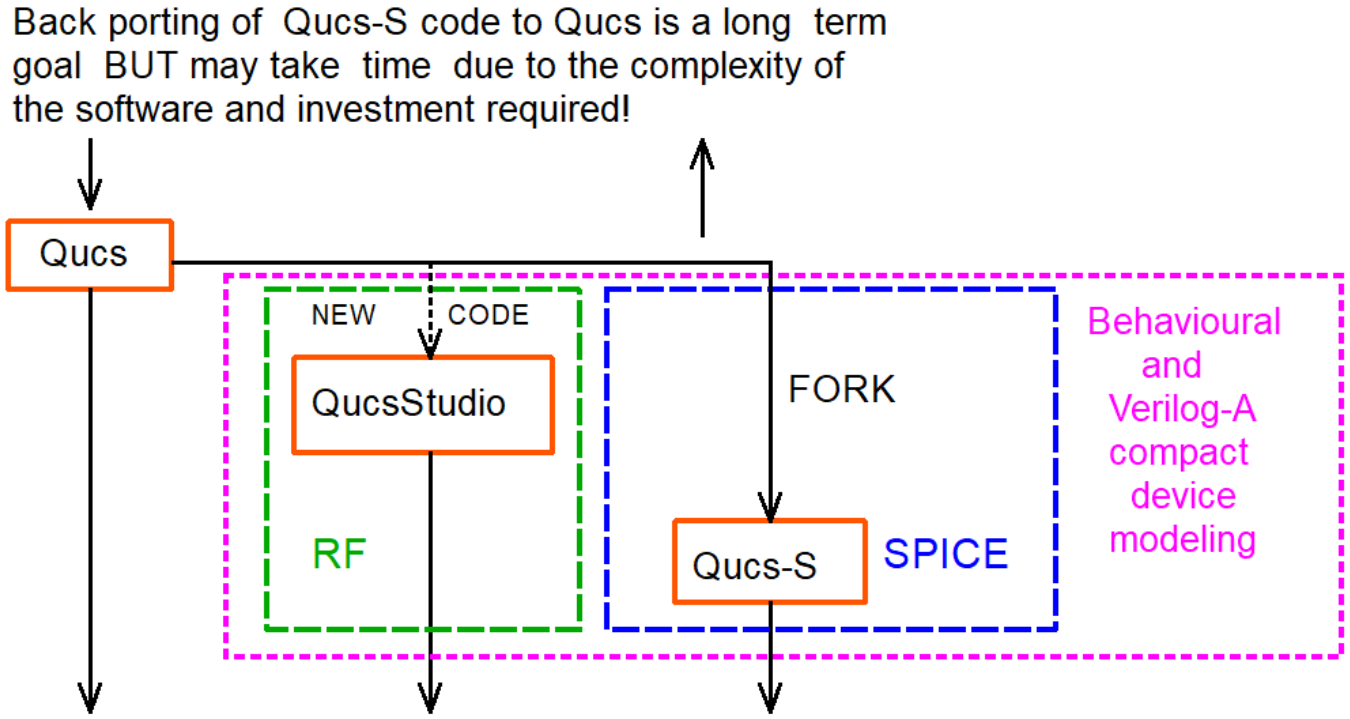
Release 0.0.1 2003

Release 1.0.0 2011

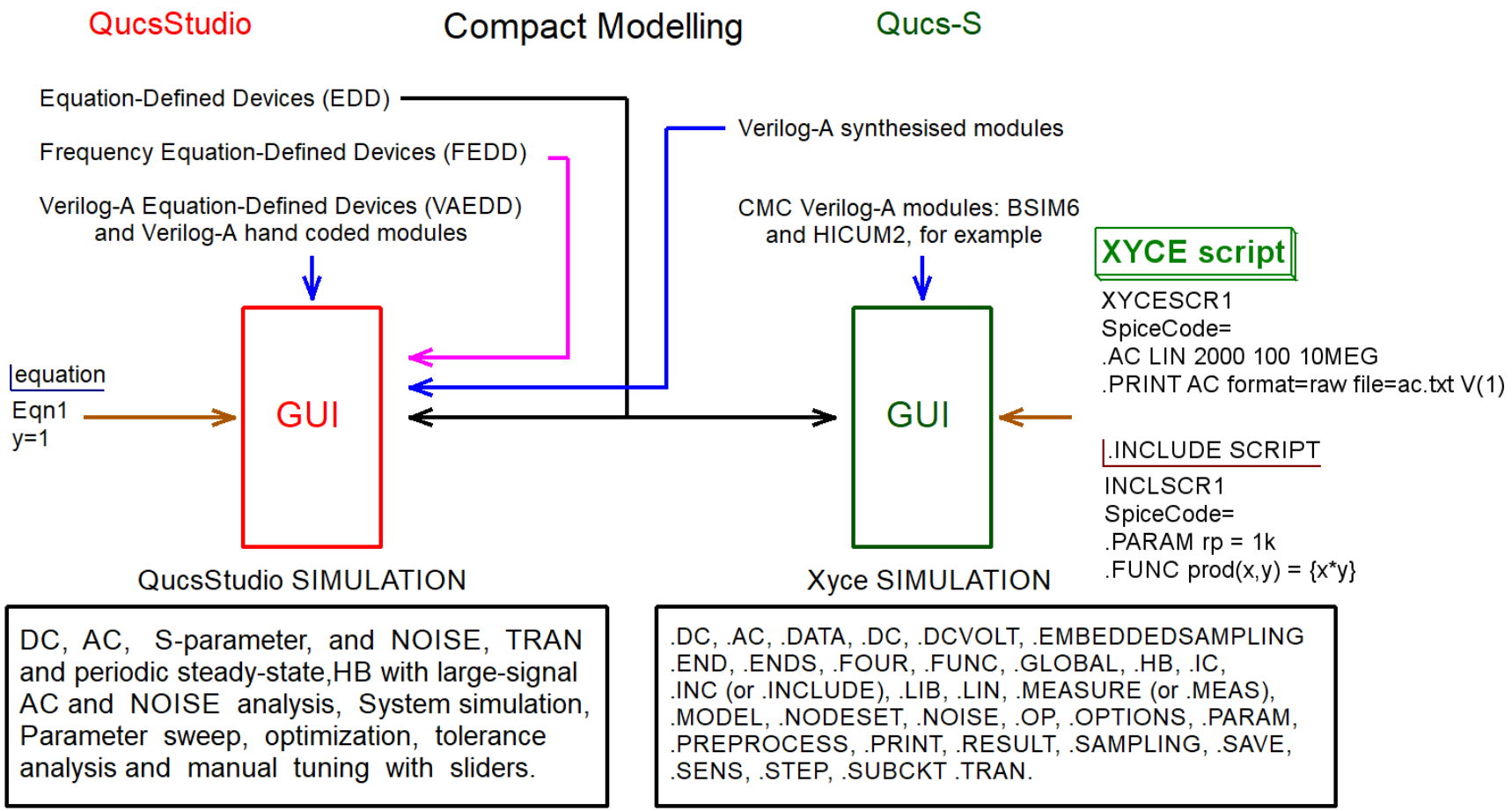
Release RC1 2015

Today

Around 60% of the Qucs and Qucs-S downloads are for the Windows operating system.

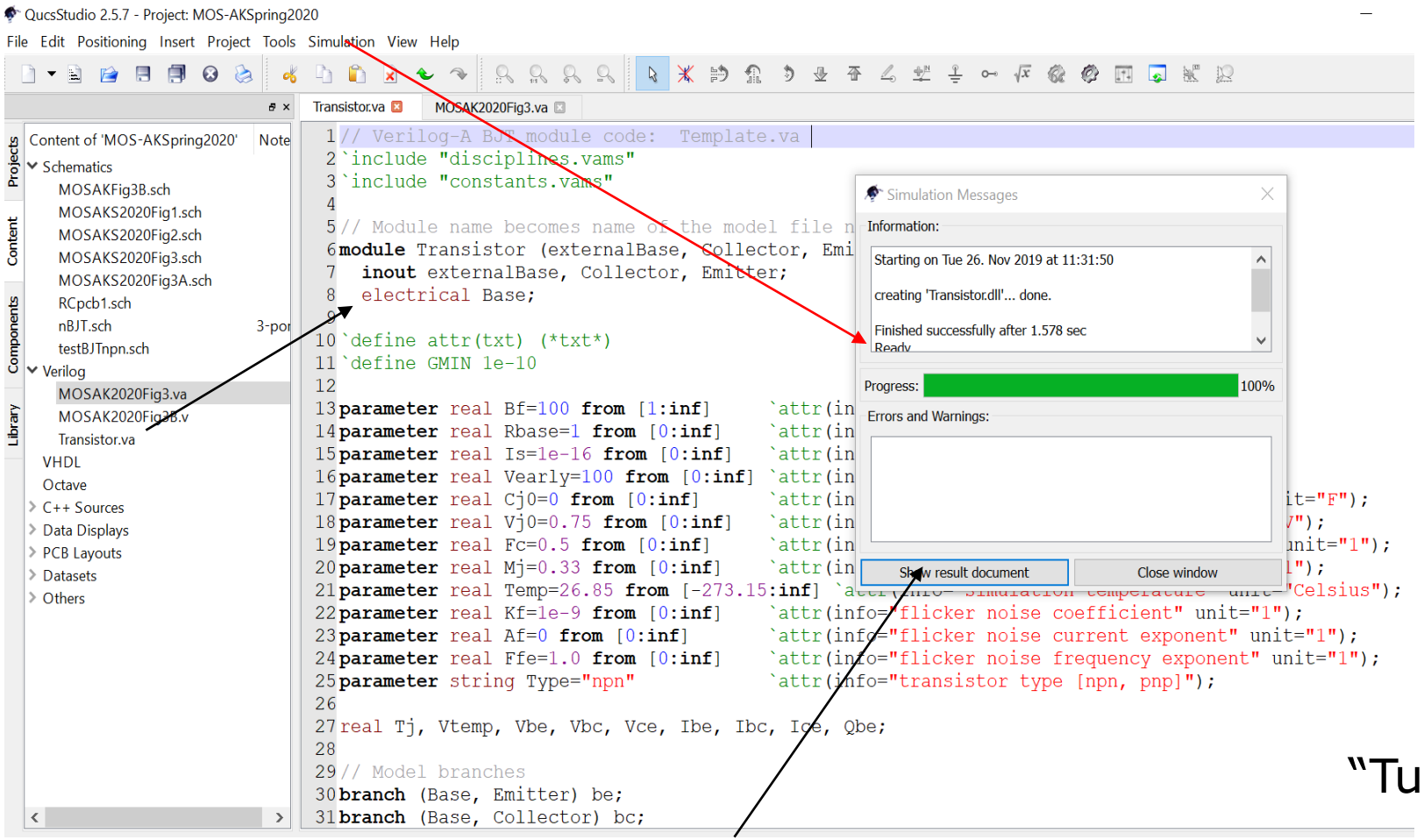



QucsStudio with Qucs-S provide advanced RF design, compact device modelling and simulation linked to industrial level Xyce SPICE circuit simulation "glued" together by behavioural compact device models and Verilog-A modules.


QucsStudio
Compact Modelling
Qucs-S
Equation-Defined Devices (EDD)
Frequency Equation-Defined Devices (FEDD)
Verilog-A Equation-Defined Devices (VAEDD)
and Verilog-A hand coded modules
Verilog-A synthesised modules
CMC Verilog-A modules: BSIM6
and HICUM2, for example
XYCE script
XYCESCR1
SpiceCode=
.AC LIN 2000 100 10MEG
.PRINT AC format=raw file=ac.txt V(1)
.INCLUDE SCRIPT
INCLSCR1
SpiceCode=
.PARAM rp = 1k
.FUNC prod(x,y) = {x*y}
equation
Eqn1
y=1
GUI
GUI
QucsStudio SIMULATION
Xyce SIMULATION
DC, AC, S-parameter, and NOISE, TRAN and periodic steady-state,HB with large-signal AC and NOISE analysis, System simulation, Parameter sweep, optimization, tolerance analysis and manual tuning with sliders.
.DC, .AC, .DATA, .DC, .DCVOLT, .EMBEDDEDSAMPLING
.END, .ENDS, .FOUR, .FUNC, .GLOBAL, .HB, .IC,
.INC (or .INCLUDE), .LIB, .LIN, .MEASURE (or .MEAS),
.MODEL, .NODESET, .NOISE, .OP, .OPTIONS, .PARAM,
.PREPROCESS, .PRINT, .RESULT, .SAMPLING, .SAVE,
.SENS, .STEP, .SUBCKT .TRAN.

# Verilog-A Development

Edit Verilog-A module code

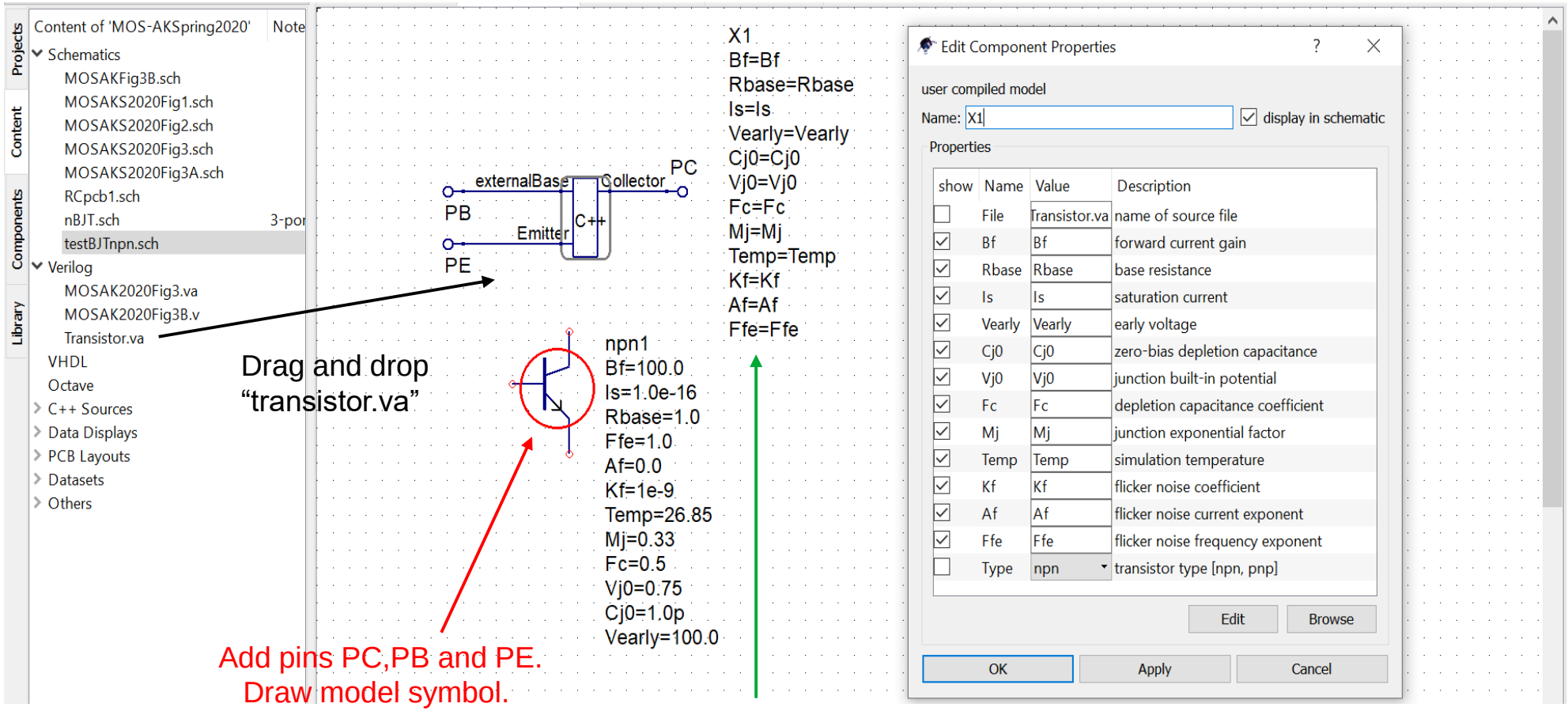



Synthesize Verilog-A Module Code [with ADMS] To form "Transistor.va.cpp" and Compile C++ code to form "Transistor.dll"

"Turn-Key" system

Click to display generated C++ code

# Verilog-A Development



# Built in Verilog-A modules

Windows “turn key” ADMS Verilog-A module development system

QucsStudio
Standard SPICE models:
Diode, BJT, MOSFET, JFET and MESFET

CMC and other models:
BJT – HICUM/L2/L0
MOSFET – EKV2.6

The Windows serial (single processor) version of Xyce is compiled under Cygwin64 using static libraries. Hence, a “turn key” Verilog-A module development system is NOT implemented.

Qucs-S/Xyce
Standard SPICE models:
Diode, BJT, MOSFET, JFET and MESFET

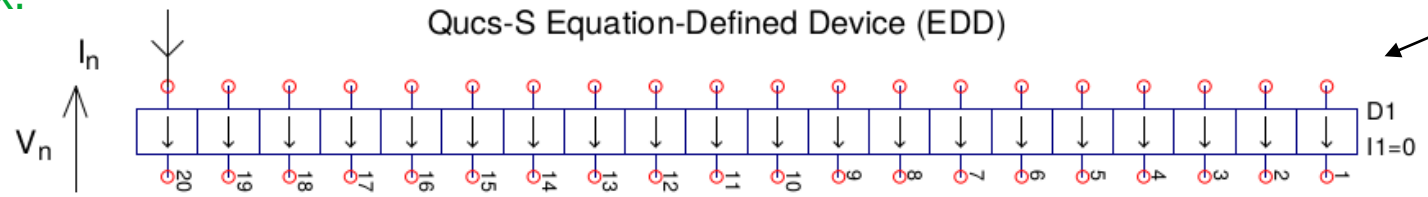
CMC and other models:
BJT – VBIC 1.3, FBH HBT_X, HICUML0/L2, MEXTRAM.
MOSFET – BSIM3, BSIM4, BSIM6, BSIM_SOI, BSIM_CMG, MVS, PSP

**Verilog-A ADMS generated models**

CMC = Compact Modelling Coalition

QucsStudio: 8 two port Terminals max.

Qucs-S: 20 two port terminals max.



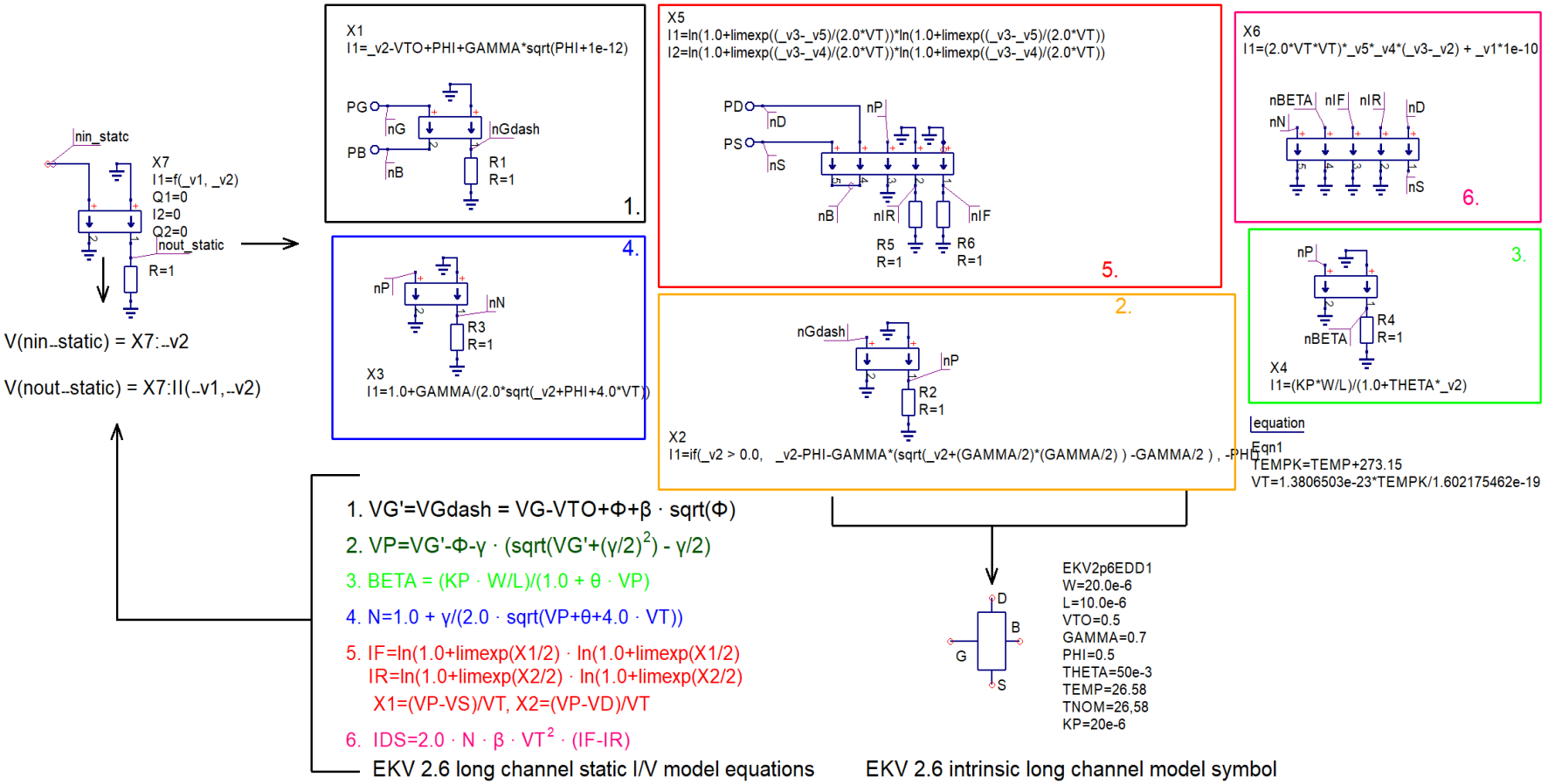

$I = I(V),\ g = dI/dV$

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

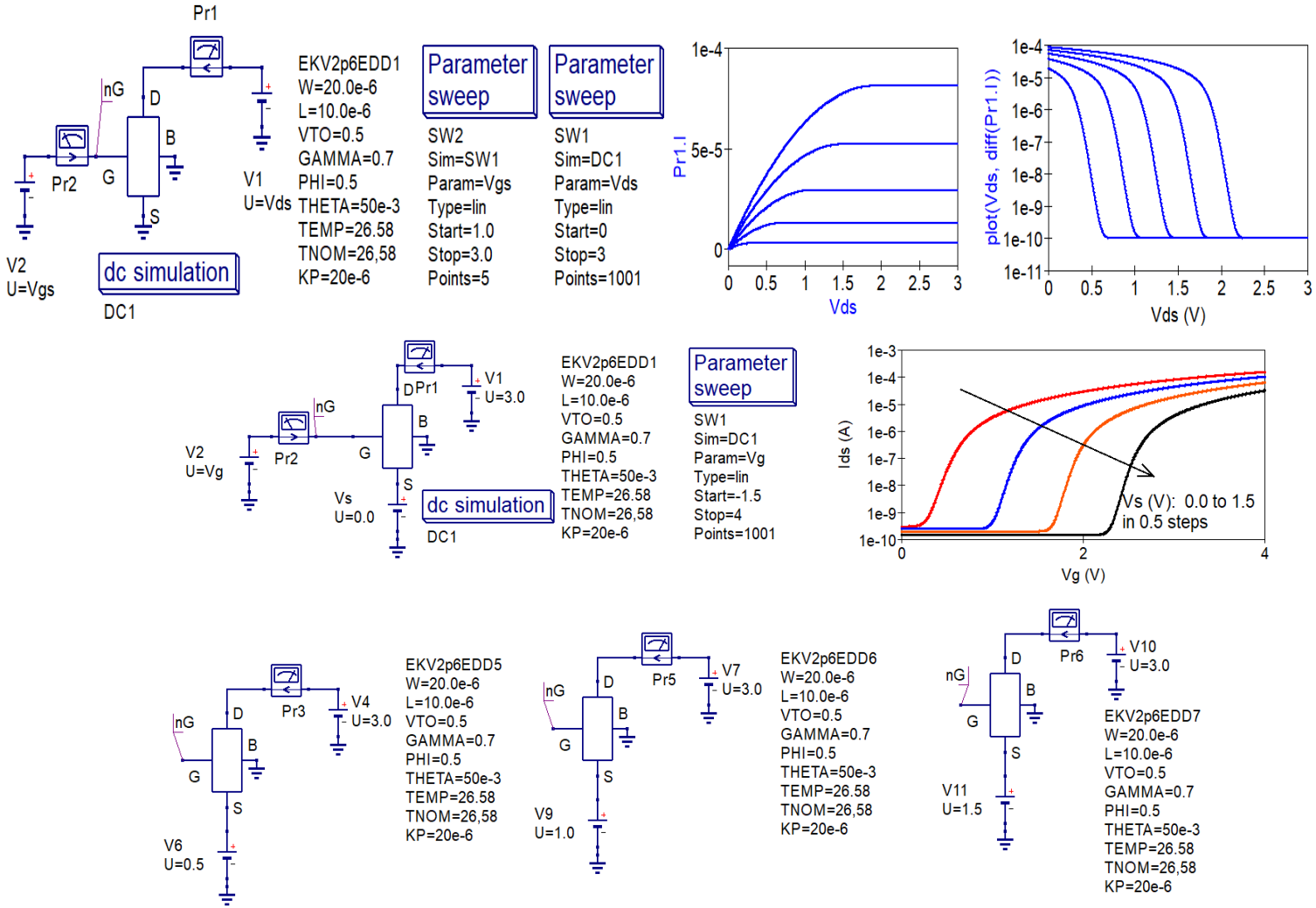
the current flowing in branch n is $I_n = I(V_n) + d/dt(Q_n)$, and $1 <= n <= 20$.

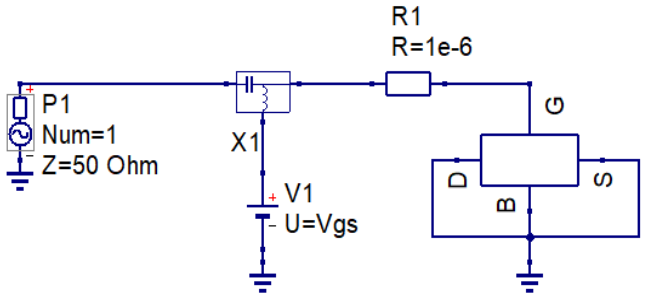
- EDD is a multiterminal nonlinear component with branch currents that can be functions of EDD branch voltage, and stored charge that can be a function of both EDD branch voltages and currents
- EDD is similar, but more advanced to the SPICE 3f5 B type I or V controlled sources
- EDD can be combined 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

EDD blocks


nin_statc
X7
I1=f(_v1, _v2)
Q1=0
I2=0
Q2=0
nout_static
R=1
V(nin_static) = X7:_v2
V(nout_static) = X7:II(_v1,_v2)
X1
I1=_v2-VTO+PHI+GAMMA*sqrt(PHI+1e-12)
PG
nG
nGdash
PB
nB
R1
R=1
1.
X5
I1=ln(1.0+limexp((_v3-_v5)/(2.0*VT))*ln(1.0+limexp((_v3-_v5)/(2.0*VT))
I2=ln(1.0+limexp((_v3-_v4)/(2.0*VT))*ln(1.0+limexp((_v3-_v4)/(2.0*VT))
PD
nD
nP
PS
nS
nB
nIR
nIF
R5
R=1
R6
R=1
5.
X6
I1=(2.0*VT*VT)*_v5*_v4*(_v3-_v2) + _v1*1e-10
nBETA
nIF
nIR
nD
nN
nS
6.
4.
nP
nN
R3
R=1
X3
I1=1.0+GAMMA/(2.0*sqrt(_v2+PHI+4.0*VT))
2.
nGdash
nP
R2
R=1
X2
I1=if(_v2 > 0.0, _v2-PHI-GAMMA*(sqrt(_v2+(GAMMA/2)*(GAMMA/2)) -GAMMA/2 ) , -PHI)
3.
nP
R4
nBETA
R=1
X4
I1=(KP*W/L)/(1.0+THETA*_v2)
equation
Eqn1
TEMPK=TEMP+273.15
VT=1.3806503e-23*TEMPK/1.602175462e-19
1. VG'=VGdash = VG-VTO+Φ+β · sqrt(Φ)
2. VP=VG'-Φ-γ · (sqrt(VG'+(γ/2)^2) - γ/2)
3. BETA = (KP · W/L)/(1.0 + θ · VP)
4. N=1.0 + γ/(2.0 · sqrt(VP+θ+4.0 · VT))
5. IF=ln(1.0+limexp(X1/2) · ln(1.0+limexp(X1/2)
IR=ln(1.0+limexp(X2/2) · ln(1.0+limexp(X2/2)
X1=(VP-VS)/VT, X2=(VP-VD)/VT
6. IDS=2.0 · N · β · VT^2 · (IF-IR)
EKV 2.6 long channel static I/V model equations
D
B
G
S
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
EKV 2.6 intrinsic long channel model symbol

# EKV 2.6 Id/Vd Example






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

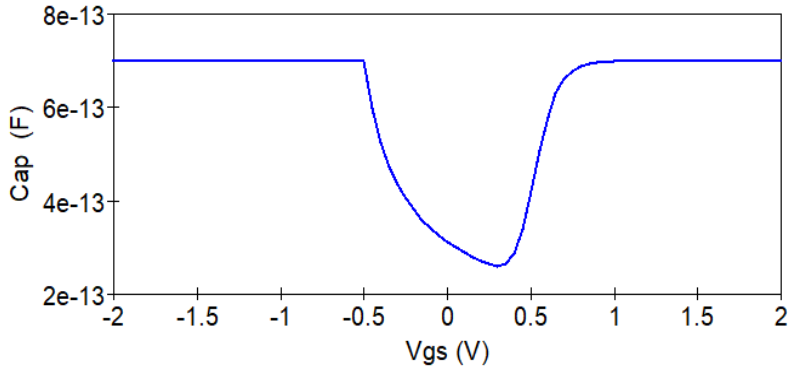
**Parameter sweep**

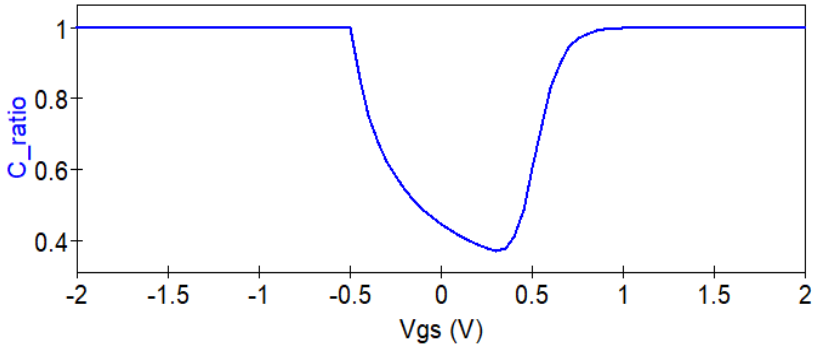
SW1
Sim=SP1
Param=Vgs
Type=lin
Start=-2
Stop=2
Points=81

**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 |



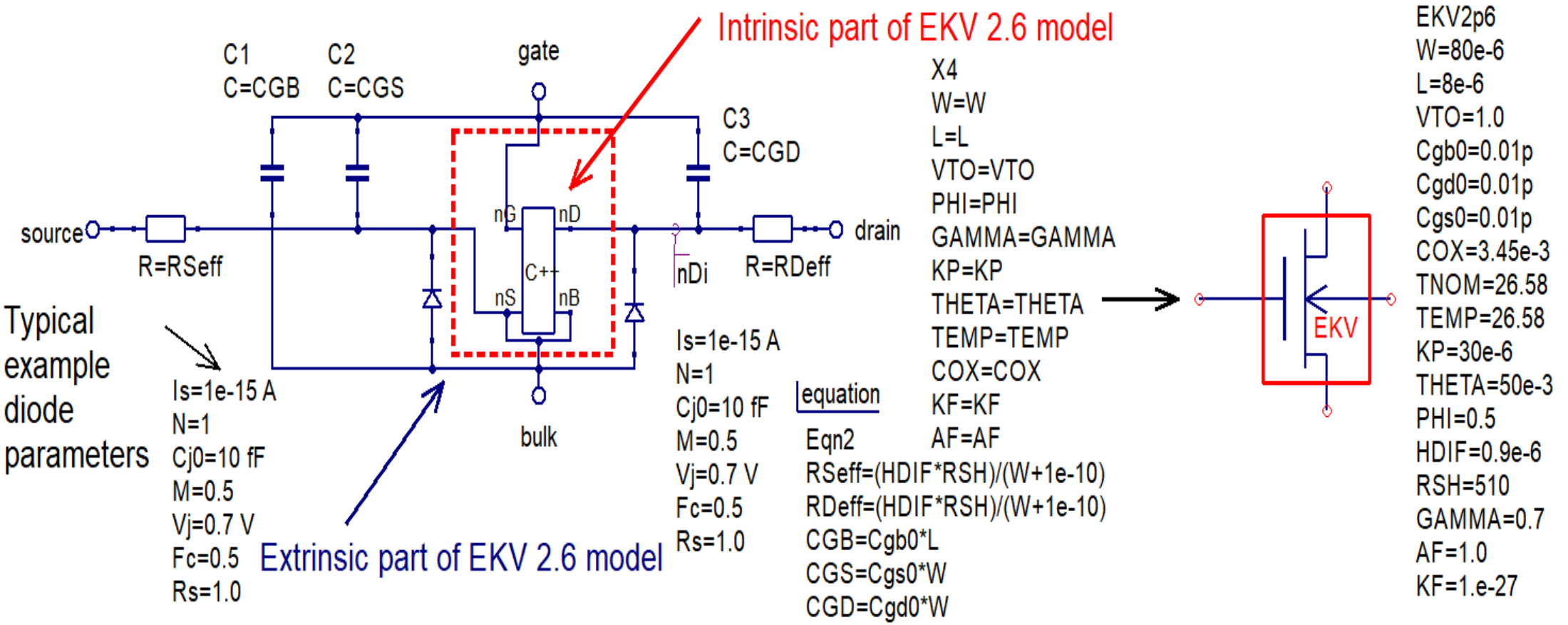

X9 ---> nq (69), X13 ---> qD, qS (72, 73)
X8 ---> Xf, Xr (70, 71), X14 ---> ql, qB (74, 75)

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

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

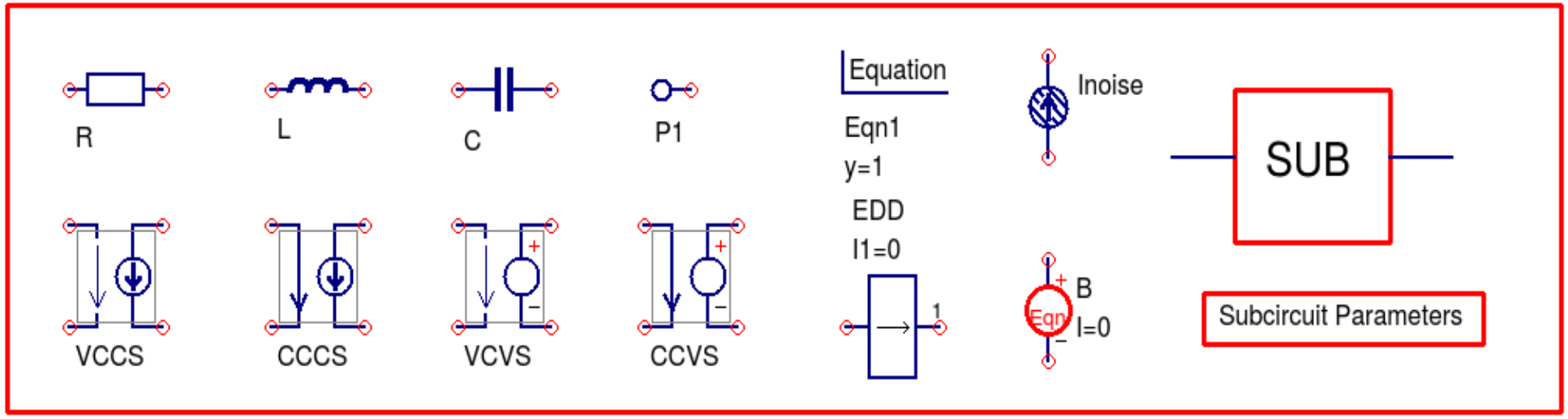
Matthias Bucher et al., The EPFL-EKV MOSFET Model Equations for Simulation, Technical Report, Model Version 2.6, June 1997,

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

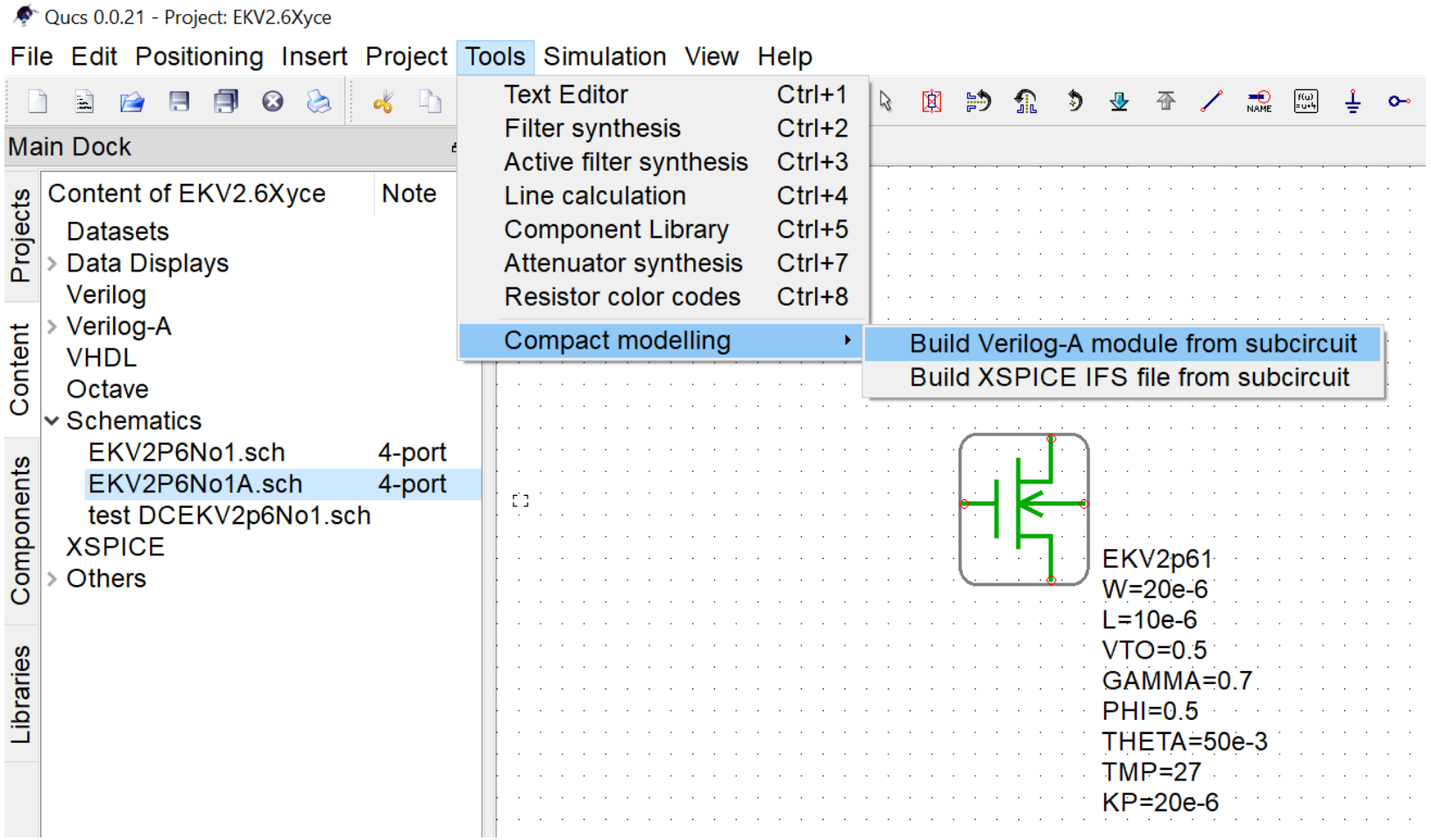

Intrinsic part of EKV 2.6 model
C1
C=CGB
C2
C=CGS
gate
C3
C=CGD
source
R=RSeff
nG
nD
C++
nS
nB
nDi
R=RDeff
drain
bulk
Typical example diode parameters
Is=1e-15 A
N=1
Cj0=10 fF
M=0.5
Vj=0.7 V
Fc=0.5
Rs=1.0
Extrinsic part of EKV 2.6 model
Is=1e-15 A
N=1
Cj0=10 fF
M=0.5
Vj=0.7 V
Fc=0.5
Rs=1.0
X4
W=W
L=L
VTO=VTO
PHI=PHI
GAMMA=GAMMA
KP=KP
THETA=THETA
TEMP=TEMP
COX=COX
KF=KF
AF=AF
equation
Eqn2
RSeff=(HDIF*RSH)/(W+1e-10)
RDeff=(HDIF*RSH)/(W+1e-10)
CGB=Cgb0*L
CGS=Cgs0*W
CGD=Cgd0*W
EKV
EKV2p6
W=80e-6
L=8e-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

Qucs-0.0.22-S includes a GPL Verilog-A synthesis tool for compact device modeling.

- The Verilog-A synthesizer is a fully working version of this open source ECAD tool,
- Verilog-A device/subcircuit models can be synthesized from the following built in components:



# Verilog-A module synthesis



# Verilog-A module synthesis

| EDD | Equation number | property |
|---|---|---|
| X1 | 1 | VGdash [nGdash] |
| X2 | 2 | VP [nP] |
| X3 | 4 | N [nN] |
| X4 | 3 | BETA [nBETA] |
| X5 | 5 | IF [nIF] and IR [nIR] |
| X6 | 6 | Ids [nD,nS] |

```
`include "disciplines.vams"
`include "constants.vams"
module EDFig20(nG, nB, nD, nS);
inout nG, nB, nD, nS;
electrical nGdash, nG, nB, nBETA, nP, nN, nIF, nIR, 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;
real VThermal, TEMPK;
analog begin
TEMPK=TEMP+273.15;
VThermal=`P_K*TEMPK/`P_Q;
I(nGdash) <+ V(nGdash);
I(nGdash) <+ -(V(nG,nB)-VTO+PHI+GAMMA*sqrt(PHI+1e-12));
I(nP) <+ V(nP);
I(nP) <+ -((V(nGdash)>0.0)?V(nGdash)-PHI-GAMMA*(sqrt(V(nGdash)+
      (GAMMA/2)*(GAMMA/2))-GAMMA/2):-PHI);
I(nN) <+ V(nN);
I(nN) <+ -(1.0+GAMMA/(2.0*sqrt(V(nP)+PHI+4.0*VThermal)));
I(nBETA) <+ V(nBETA);
I(nBETA) <+ -((KP*W/L)/(1.0+THETA*0.5*(V(nP)+sqrt(V(nP)*V(nP)+
      VThermal*VThermal))));
I(nIF) <+ V(nIF);
I(nIF) <+ -(ln(1.0+limexp((V(nP)-V(nS,nB))/(2.0*VThermal)))*ln(1.0+
      limexp((V(nP)-V(nS,nB))/(2.0*VThermal))));
I(nIR) <+ V(nIR);
I(nIR) <+ -(ln(1.0+limexp((V(nP)-V(nD,nB))/(2.0*VThermal)))*ln(1.0+
      limexp((V(nP)-V(nD,nB))/(2.0*VThermal))));
I(nD,nS) <+ (2.0*VThermal*VThermal)*V(nN)*V(nBETA)*(V(nIF)-V(nIR))+
      V(nD,nS)*1e-10;
end
endmodule
```

Verilog-A module EDFig20 starts here

Model parameters

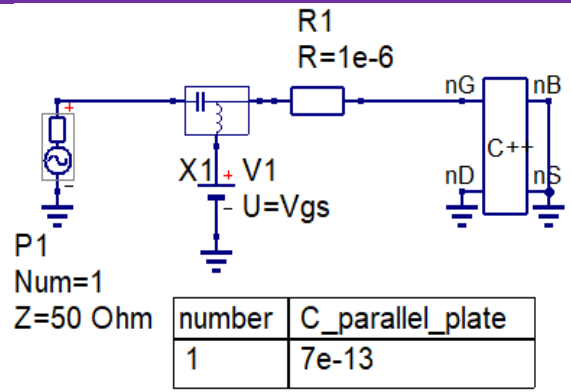
Analog equations

Analog contributions

Verilog-A module EDFig20 finishes here

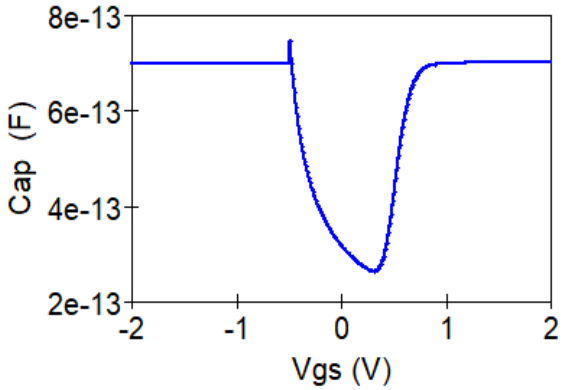
Internal nodes - two I(xx) <+ … contributions per EDD equation
External nodes- one I(yy) <+ … contribution per node pair.

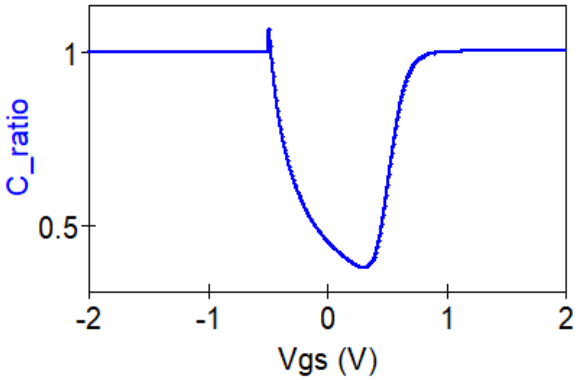
# Verilog-A module synthesis




| number | C_parallel_plate |
|---|---|
| 1 | 7e-13 |

X2
W=20e-6
L=10e-6
VTO=0.5
PHI=0.5
GAMMA=0.7
KP=20e-6
THETA=50e-3
TEMP=26.58
COX=3.5e-3



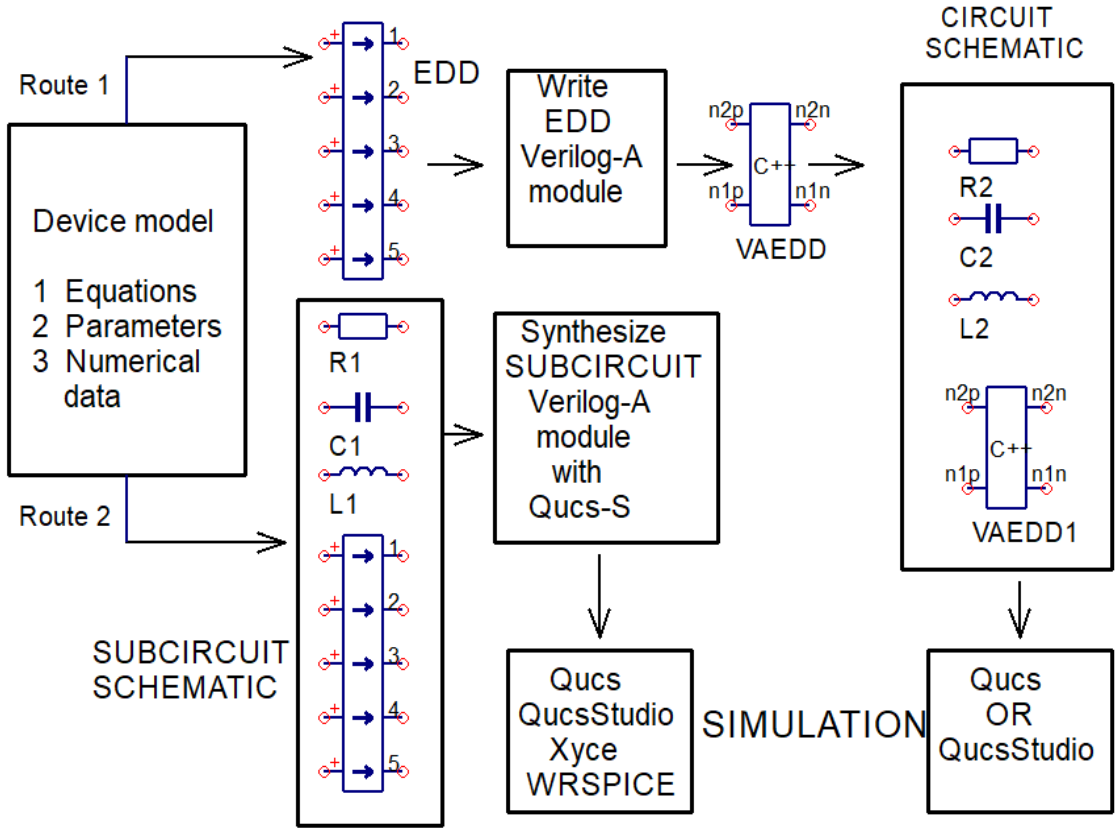

```
`include "disciplines.vams"
`include "constants.vams"
module EDFig20B(nG, nB, nD, nS);
inout nG, nB, nD, nS;
electrical  nG, nB, nD, nS;
parameter real W=20e-6;  parameter real L=10e-6;  parameter real VTO=0.5;  parameter real PHI=0.5; parameter real GAMMA=0.7;
parameter real KP=20e-6;  parameter 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

Only one I(yy) <+ … contribution per external node pair

- Simulation of complex EDD compact models can be very slow.
- Constructing complex Verilog-A models often requires considerable time and effort.
- Introduce a compromise: construct a compact device model from a mixture of EDD blocks and very simple Verilog-A modules where each Verilog-A module models at most two or three EDD equations.

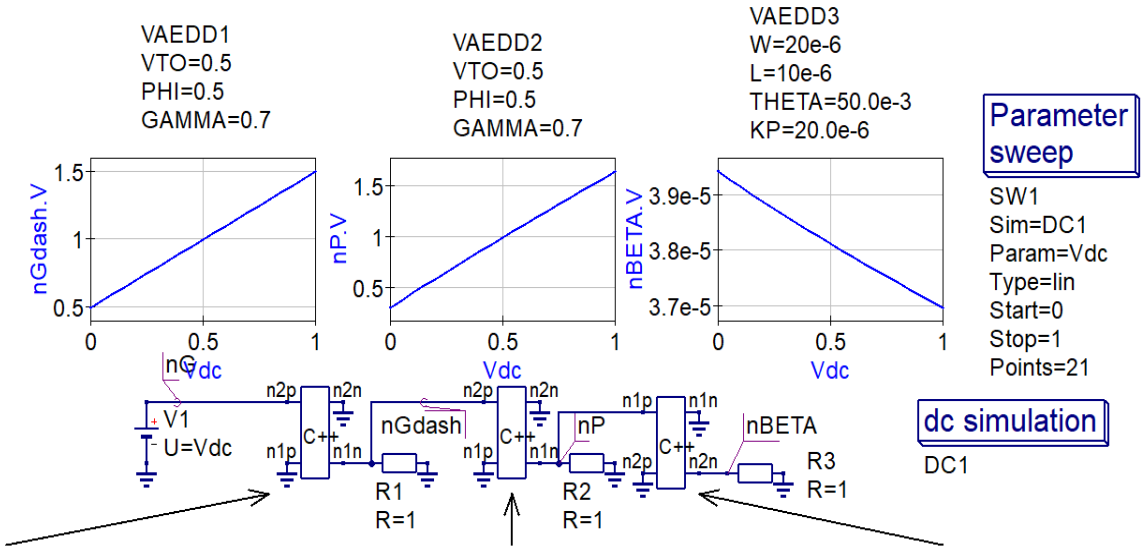



**These simplified Verilog-A blocks are called Verilog-A Equation-Defined devices or VAEDD**

# VAEDD Example

One or more VAEDD blocks can be mixed with EDD to construct a compact device model

EKV2.6 :
Vgdash
VP
BETA




```
`include "disciplines.vams"
`include "constants.vams"
module VAEDDnGdash(n2p,n2n,n1p,n1n);
inout n2p, n2n, n1p, n1n;
electrical n2p, n2n, n1n, n1n;
parameter real VTO = 0.5;
parameter real PHI = 0.5;
parameter real GAMMA = 0.7;
branch (n2p, n2n) B1;
branch (n1p, n1n) B2;
analog begin
 I(B1) <+ V(B1)*1e-9;
 I(B2) <+ V(B1)-VTO+PHI+GAMMA*sqrt(PHI+1e-10);
 end
endmodule
```

```
`include "disciplines.vams"
`include "constants.vams"
module VAEDDnP(n2p,n2n,n1p,n1n);
inout n2p, n2n, n1p, n1n;
electrical n2p, n2n, n1n, n1n;
parameter real VTO = 0.5;  parameter real PHI = 0.5;
parameter real GAMMA = 0.7;
real GAMMAO2;
branch (n2p, n2n) B1;  branch (n1p, n1n) B2;
analog begin
 GAMMAO2=GAMMA/2;   I(B1) <+ V(B1)*1e-9;
 if (V(B1) > 0.0)  begin I(B2) <+ V(B1)-PHI+
   GAMMA*(sqrt(V(B1)+GAMMAO2*
   GAMMAO2)-GAMMAO2); end
 else begin I(B2) <+ -PHI; end
end
endmodule
```

```
`include "disciplines.vams"
`include "constants.vams"
module VAEDDnBETA(n1p, n1n, n2p, n2n);
inout n2p, n2n, n1p, n1n;
electrical n2p, n2n, n1p, n1n;
parameter real W = 20e-6;
parameter real L = 10e-6;
parameter real THETA = 50.0e-3;
parameter real KP = 20.0e-6;
branch (n1p, n1n) B1;
branch (n2p, n2n) B2;
analog begin
 I(B1) <+ V(B1)*1e-9;
 I(B2) <+ (KP*W/L)/(1.0+THETA*V(B1));
 end
endmodule
```

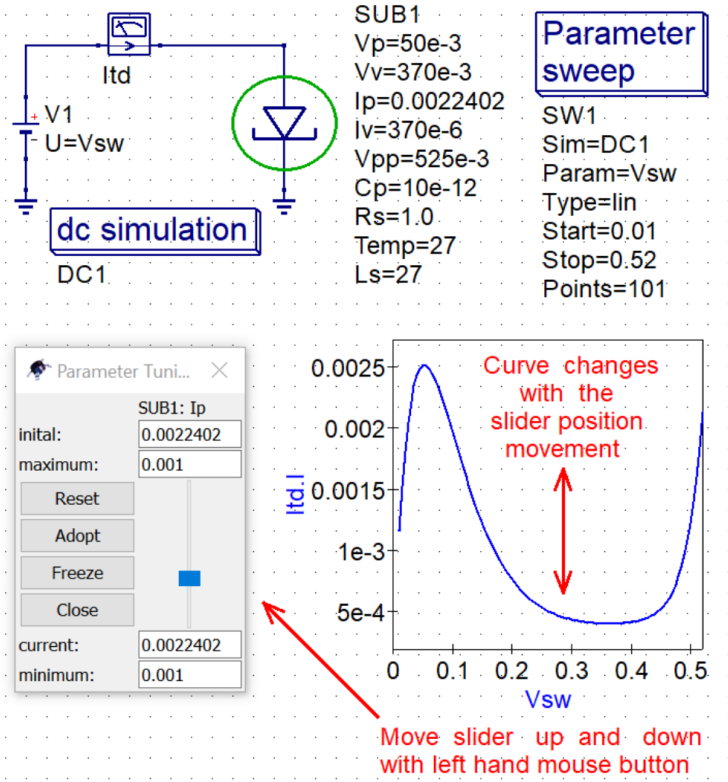
QucsStudio 3.3.2 was released on the 15 July 2020 two years after version 2.5.7.

**New features: NOT in any specific order**

- ❑ EM field simulation using openEMS,
- ❑ New diagram: equation,
- ❑ HICUM L0 model implemented,
- ❑ Improved random-number generation,
- ❑ Smith chart: enable impedance and admittance circles,
- ❑ New document type: PCB layout,
- ❑ Differentiate name.i and name.v in simulator equations,
- ❑ Complex source and load impedance possible in matching dialogue.

**Plus others and many bug fixes**

- This package has reached an advanced stage of development in that it offers an almost complete set of circuit simulation routines covering the d.c. to transient domains with significant additions beyond SPICE 3f5, like multi-tone Harmonic Balance analysis, Monte Carlo analysis, parameter sweep, multi-port S parameter and noise simulation, optimization and system simulation.
- Full "turn-key" Verilog-A compact modelling is also offered via the ADMS software.
- In terms of development QucsStudio is particularly interesting in that it is the first of the Qucs series of circuit simulators to introduce interactive animation as a tool for advanced circuit simulation.
- QucsStudio allows one or more parameter values to be simultaneously controlled by sliders.
- With the computational power of a modern PC changes in simulation output data can be observed as movements in plotted curves as the sliders are moved.

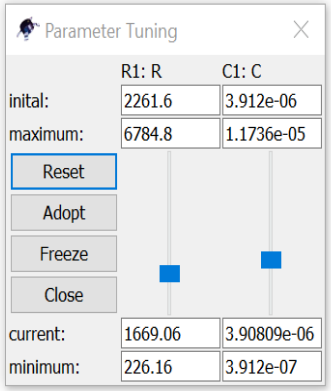


**Measurements: Digilent "Analog Discovery 2" [16]**

Measured RC data in CSV format

```
Frequency (Hz), V(nCH1) (dB), V(nCH2) (dB)
2, 0.0024752594790344 67 ,-0.0534246328867990 9
2.4045288692348 26 ,0.0027373575656619841, -0.0669157344350223 3
2.8908795414918 56, 0.0026234651932572 26 ,-0.0858036602431495
3.4756016574987 52, 0.0024624672368685 76, -0.1129364847879198
4.1785922261708078 ,0.0025311264485912 ,-0.1518179252403876
5.0237728630191 59, 0.0025937545319882 87 ,-0.2075493458479011
6.0399034408040 31, 0.0026833068876765 68 , -0.2867653490052169
7.2615610954020 28, 0.0027652123076593 83 , -0.3985765557863726
8.7303166448033 22, 0.0026653674287242 74, -0.5555438274209732
10.496149204995 45, 0.0027734941331188 48, -0.7721405324716 02
12.619146889603 86, 0.0024794777898935 6 , -1.0673101054129 36
.....
.....
.....
20000 ,0.0576995283080102 6, -58.3185526060593 3
```

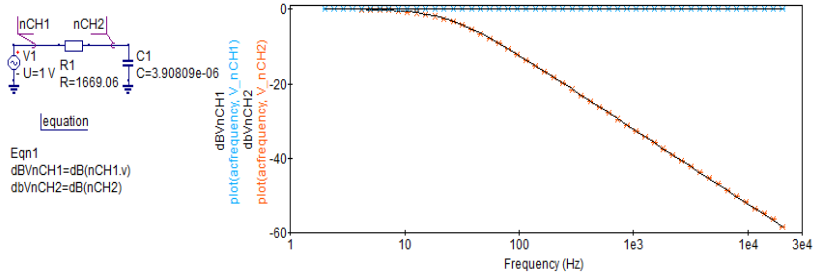
Octave script to convert XXXX.CSV data to simulation control lists




**R and C Parameter tuning**

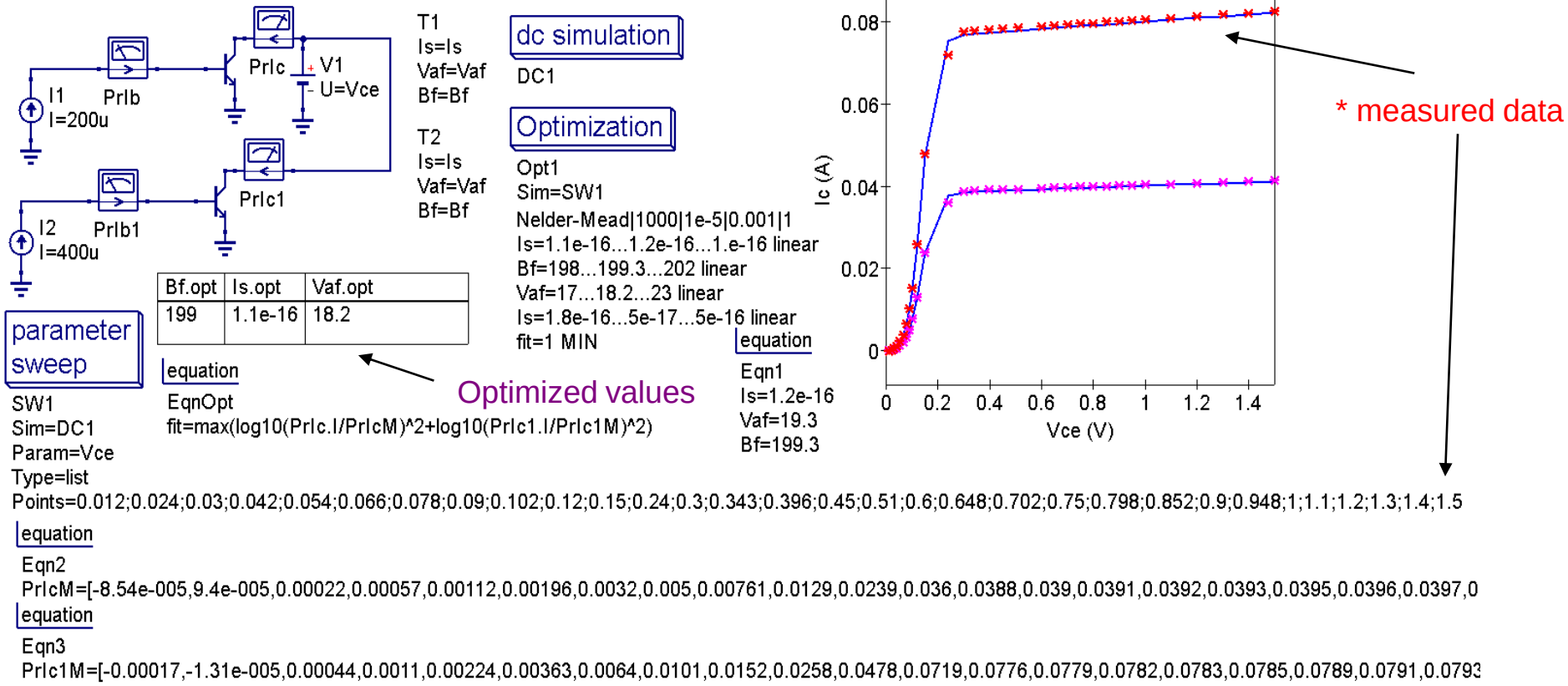
```
ac simulation
AC1
Type=list
Points=2;2.4045;2.8909;3.4756;4.1786;5.0238;6.0399;7.2616;8.7303;10.496;12.619;15.172;18.24;21.93;26.365;31.698;38.109;45.817;55.085;66.226;79.621;95.726;115.09;138.37
equation
Eqn2
V_nCH1=[0.0024753,0.0027374,0.0026235,0.0024625,0.0025311,0.0025938,0.0026833,0.0027652,0.0026654,0.0027735,0.0024795,0.0022577,0.0024353,0.0023488,0.00236(
equation
Eqn3
V_nCH2=[-0.053425,-0.066916,-0.085804,-0.11294,-0.15182,-0.20755,-0.28677,-0.39858,-0.55554,-0.77214,-1.0673,-1.4611,-1.9741,-2.6216,-3.4144,-4.3522,-5.4268,-6.6212,-7.
```

**Synthesized simulation a.c. control Icon and measured data Icons**

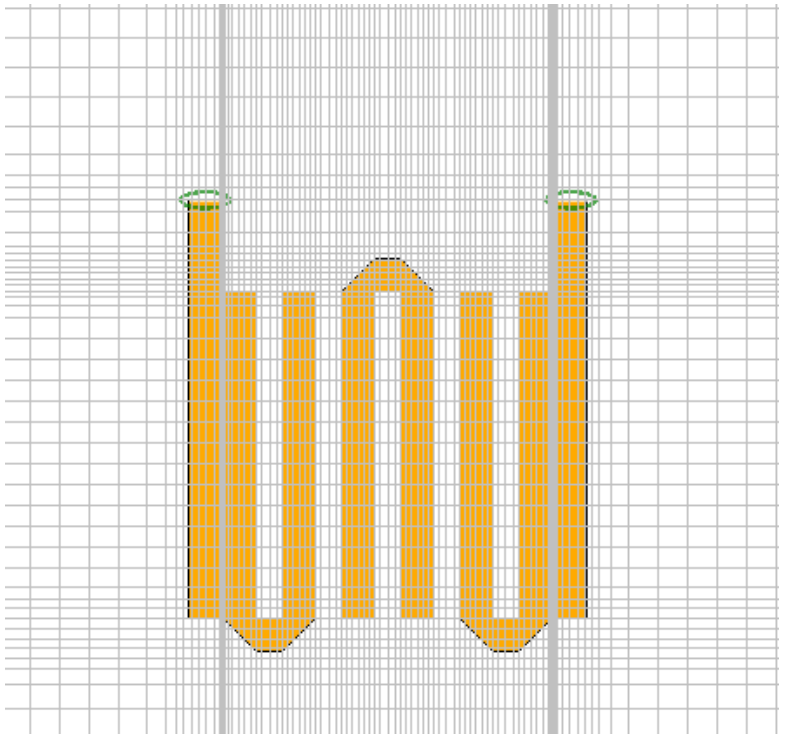


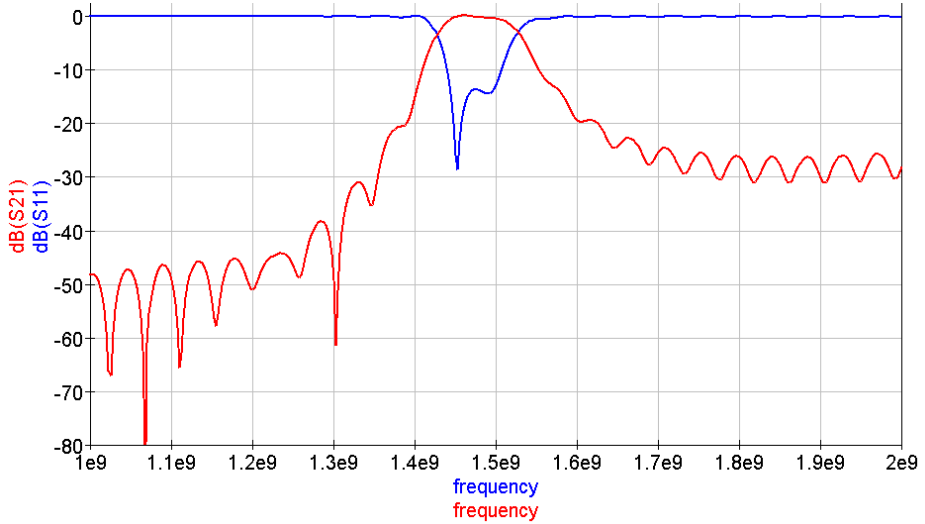

**Measured data and simulation output visualization**

**A simple RC low pass passive filter example showing an advanced test bench schematic with a computer controlled transfer function measurement system where the measured output data is converted from CSV format to Qucs simulation control ICONS.**




Three parameter optimization (Bf, Is, Vaf ) with parallel IC/Vce BJT test bench using a single Vce parameter sweep and (1) Ib=200u, PricM measured Ic data and (2) Ib= 400u, Pric1M measured Ic data.






Open EMS simulated response

EM field simulation using openEMS: 1.5 GHz bandpass filter (imported from HyperLynx file by Koen De Vleeschauwer). Example developed by Dr M. Margraf as part of the Qucs-Studio 3.3.2 release.

Xyce 7.1 was released on 8 June 2020.

**New features: NOT in any specific order**

- Non-linear solution dependent capacitors
  Cap1 1 2 q= {ca*(c1*v1*ln(cosh((v(1,2)-v0)/v1)+c0*v(1,2))}, or
  Cap1 1 2 c= {ca*(c0+c1*tanh((v(1,2)-v0)/v1))}, where both forms are implemented to ensure charge conservation,
- C-style ternary conditional operator,
- .LIN transfer analysis and extraction of S, Y and Z parameters from a general multiport network,
- .SAMPLING : calculates a full analysis (.DC, .TRAN, .AC etc.) over a distribution of parameter values,
- Sweep loops can now use .DATA command

**Plus others and many bug fixes**

Adding new Xyce features to Qucs-S.

- These Xyce specific icons allow new features to be placed on a circuit schematic and interpreted during simulation.
- Qucs-S has in fact a two level GUI system; items common to SPICE 3f5, and other equivalent simulators, operate via built-in Icons or a XYCE script, while the less used or recently added features, can only be accessed via a XYCE script.
- For example, since 2018 approximately 20 important additions to Xyce functionality have been implemented, including Monte Carlo analysis and Lattice Hypercube sampling via a new *.SAMPLING* feature, transient simulation direct sensitivity analysis that supports *.FOUR*, *.LIN* for *S* parameter multiport analysis with *Y* and *Z* output data in Touchstone level 1 and level 2 format, and a new charge expression variant for capacitors that is similar to the *EDD* branch charge implementation.

**XYCE script** (a)

```
XYCESCR1
SpiceCode=
.DC V1 0.0 0.55 0.01
.PRINT DC format=raw file=dc.txt
+ I(VITD)
```

.INCLUDE SCRIPT

```
INCLSCR1
SpiceCode=
.PARAM rp = 1k
.FUNC prod(x,y) = {x*y}
```

.FUNC

```
SpiceFunc1
prod(x,y)={x*y}
```

(b)

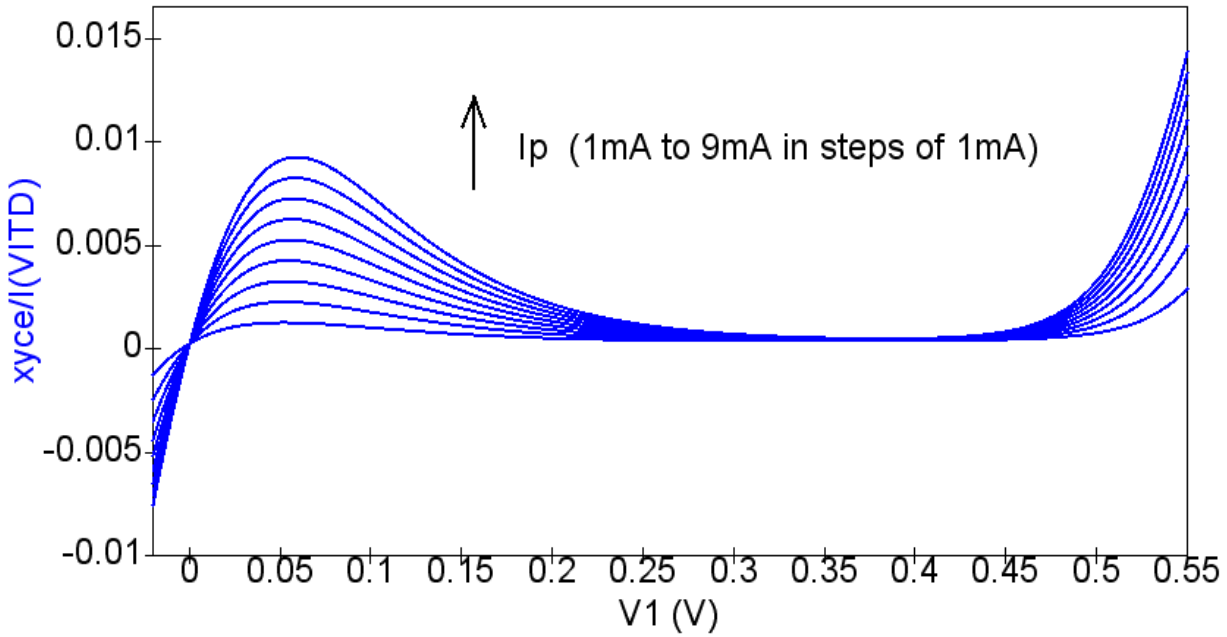
.spiceinit

```
SPICEINIT1
.spiceinit contents=
```

.INCLUDE

```
SpiceInclude1
File=~/home/user/library.inc
```

## A Xyce SPICE style tunnel diode compact model with a test bench for investigating the effects of stepped device parameters

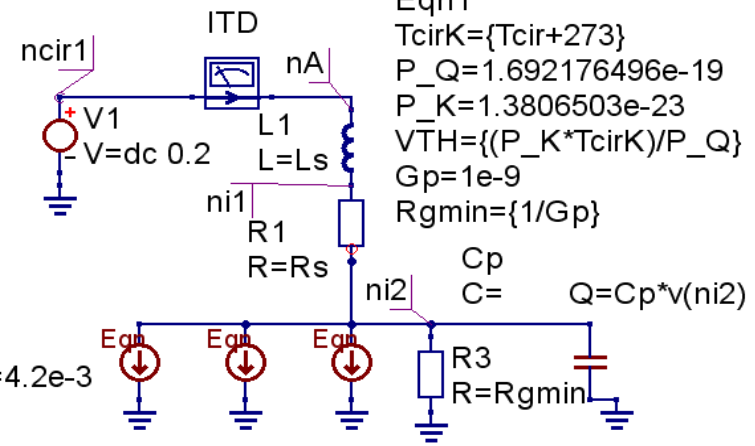



.INCLUDE SCRIPT

```
INCLSCR1
SpiceCode=
.PARAM Vp=50e-3
.PARAM Vv=370e-3
.PARAM Vpp=525e-3
.PARAM Iv=370e-6
.PARAM Tcir=27
.PARAM Ls=1e-9
.PARAM Cp=20.0e-12
.PARAM Rs=1.0
.GLOBAL_PARAM Ip=4.2e-3
```

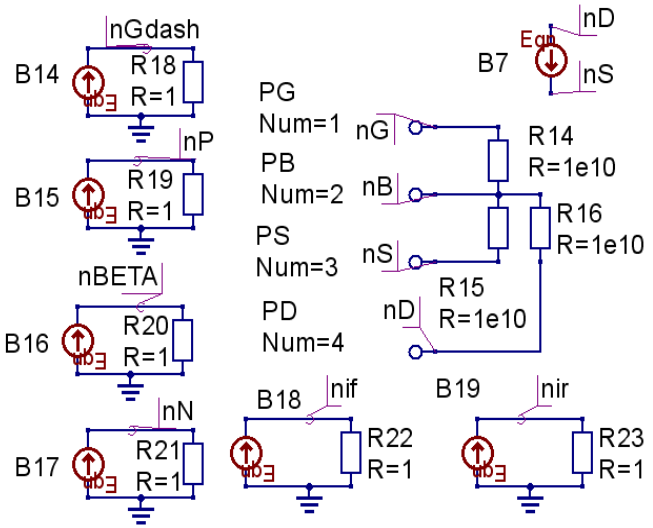
**XYCE script**

```
XYCESCR1
SpiceCode=
.DC LIN V1 -0.02 0.55 0.001
.STEP Ip 1e-3 9e-3 1e-3
.PRINT DC format=raw file=dc.txt
+ I(VITD) V(nA) V(ncir1)
```




Equation

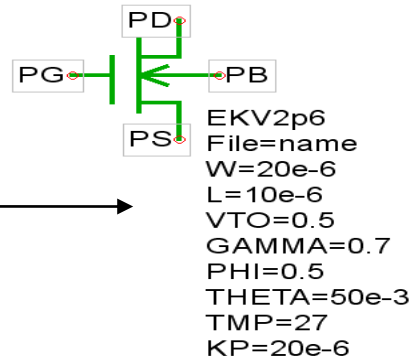
```
Eqn1
TcirK={Tcir+273}
P_Q=1.692176496e-19
P_K=1.3806503e-23
VTH={(P_K*TcirK)/P_Q}
Gp=1e-9
Rgmin={1/Gp}
```




.INCLUDE SCRIPT

INCLSCR1
SpiceCode=
.PARAM W = {W}
.PARAM L = {L}
.PARAM VTO = {VTO}
.PARAM GAMMA = {GAMMA}
.PARAM PHI = {PHI}
.PARAM THETA = {THETA}
.PARAM TMP = {TMP}
.PARAM KP = {KP}
.PARAM GAMMAO2 = {GAMMA/2}
.PARAM VTH = {1.3806503e-23*(TMP+273)/1.602157462e-19}
.PARAM VTHD2 = {1.0/(2.0*VTH)}

Much improved SPICE style behavioral modelling facilities




B14
I={V(nG)-VTO + PHI+GAMMA*sqrt(PHI+1e-12)}

B7
I=2.0*V(nN)*V(nBETA)*VTH*VTH*(V(nIf)-V(nIr))

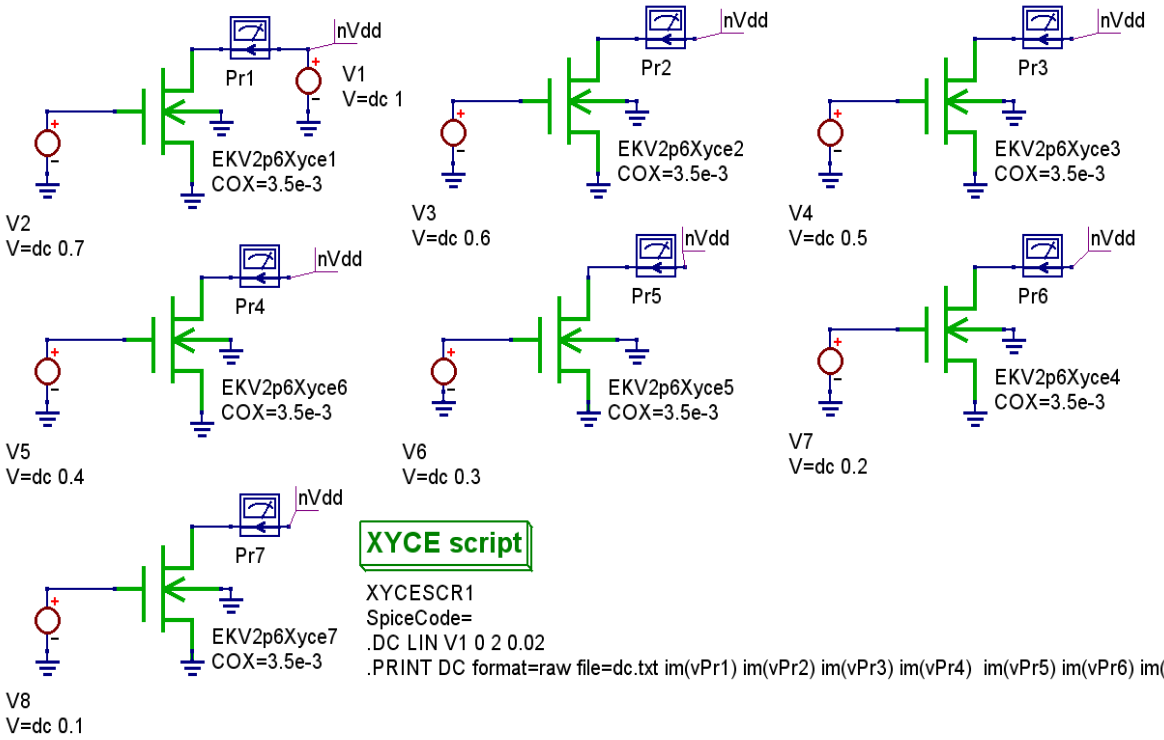
B15
I={V(nGdash)-PHI+GAMMA*(sqrt(V(nGdash)+GAMMAO2*GAMMAO2)-GAMMAO2)}

B16
I={(KP*W/L)/(1.0+THETA*V(nP))}

B17
I={1.0+GAMMA/(2.0*sqrt(V(nP)+PHI+4*VTH))}

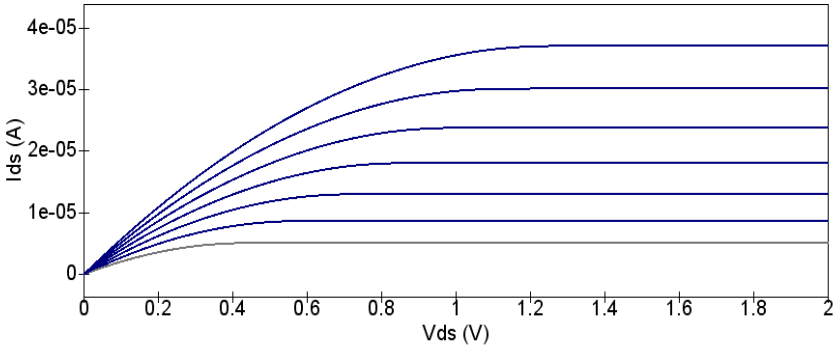
B18
I={LN(1.0+EXP( (V(nP)-V(nS))*VTHD2))*LN(1.0+EXP( (V(nP)-V(nS))*VTHD2))}

B19
I={LN(1.0+EXP( (V(nP)-V(nD))*VTHD2))*LN(1.0+EXP( (V(nP)-V(nD))*VTHD2))}




**XYCE script**

```
XYCESCR1
SpiceCode=
.DC LIN V1 0 2 0.02
.PRINT DC format=raw file=dc.txt im(vPr1) im(vPr2) im(vPr3) im(vPr4)  im(vPr5) im(vPr6) im(vPr7)
```




Ids versus Vds output characteristics generated by .DC scan using parallel test circuits with fixed values of Vgs

- Xyce 7.0/7.1 and beyond: Improved mixed-signal interface via the Verilog Procedural Interface (VPI); Increased Verilog-A simulation speeds; bug fixes (.MEASURE, .AC etc); Improved simulation data output. Current and future Verilog-A developments: Xyce/ADMS compiler modified to use analytic derivatives, new Xyce XML templates. Possible future Verilog-A additions – full implementation of ddx function and a new non-ADMS-based model compiler !.

- Tighter linking between QucsStudio/Qucs-S, Verilog-A/Xyce and device/circuit parameter measurements via an Octave "Toolkit".

- Introduction of multi-physics modelling via links to OPENMODELLICA: simulation of real world systems built from non-electrical and electrical components.

# Summary

- Low-cost high-performance PC engineering workstations have encouraged the development of compact device modelling and circuit simulation tools centered on a high-resolution graphics interface for schematic drawing, simulation control and output data visualization. This presentation outlined the capabilities of the Qucs/Qucs-S and QucsStudio series of circuit simulators and modelling tools.

- These tools allow interactive prototyping of compact device models and their testing using Qucs/QucsStudio and Qucs-S as a central platform in the construction of Verilog-A modules and Equation-Defined Device models.

- Each of these, when coupled with established, or new compact modeling techniques like mixed Equation-Defined Device and Verilog-A models (VAEDD), make the current software a highly flexible and innovative platform for compact modeling and circuit simulation.

- Future improvements to the Qucs software indicate that by merging device parameter measurements with circuit simulation for device parameter extraction, and the introduction of EM field simulation with openEMS  will significantly extend the scope of traditional circuit simulation.

# 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 circuit simulation with Qucs-S, International Journal of Electronics, Volume 105, Issue 3, pp. 412-425, http://dx.doi.org/10.1080/00207217.2017.1357764.

Brinson, Mike (2019) FOSS Compact Model Prototyping with Verilog-A Equation-Defined Devices (VAEDD). In: 26th International Conference Mixed Design of Integrated Circuits and Systems (MIXDES), pp. 92-97, 27-29 June 2019, Rzeszow, Poland. DOI: 10.23919/MIXDES.2019.8787063, ISBN: 978-1-7281-3408-6.

Grabinski Wladek, Pavanello Marcelo, De Souza Michelly, Tomaszewski Daniel, Brinson Mike, Malesinska Jola, Głuszko Grzegorz, Bucher Matthias, Makris Nikolaos, Nikolaou Aristeidis, Abo-Elhadid Ahmed, Mierzwinski Marek, Lemaitre Laurent, Lallement Christophe, Sallese Jean-Michel, Yoshitomi Sadayuki, Malisse Paul, Oguey Henri, Cserveny, Stefan, Enz Christian, Krummenacher Franço and Vittoz Eric, 2019, FOSS EKV2.6 Verilog-A Compact MOSFET Model. Proceedings of ESSDERC 2019 - 49th European Solid-State Device Research Conference (ESSDERC). pp. 190-193. ISSN ISBN: 978-1-7281-1539-9.

Mike Brinson, The Qucs/QucsStudio and Qucs-S Graphical User Interface: An Evolving "White-Board" for Compact Device Modeling and Circuit Simulation in the Current Era: Invited Paper, In: 27th International Conference Mixed Design of Integrated Circuits and Systems (MIXDES), pp. 23-32, 25-27 June 2020, Wroclaw, Poland, ISBN: 978-83-63578-10-9.

Mike Brinson, Qucs-S/QucsStudio/Octave Schematic Synthesis Tools for Device and Circuit Parameter Extraction from Measured Characteristics. In: 26th International Conference Mixed Design of Integrated Circuits and Systems (MIXDES), pp. 50-55, 27-29 June 2019, Rzeszow, Poland. ISBN: 978-83-63578-18-3.

- **Qucs -** Download version (Linux, Windows or Mac) as required from home page http://qucs.sourceforge.net/ .
- **QucsStudio** - Download Windows version QucsStudio-3.3.2.zip from http://www.dd6um.darc.de/QucsStudio/download.html.
- **Qucs-S (Qucs with SPICE) -** Download – version (Linux or Windows) as required from home page https://ra3xdh.github.io/ .
- **Xyce -** Download - version (Linux, Windows) as required from home page https://xyce.sandia.gov/ .